

---

# Hardware Design and Accurate Simulation for Benchmarking of 3D Reconstruction Algorithms

---

**Sebastian Koch\***

TU Berlin

s.koch@tu-berlin.de

**Yuri Piadyk\***

New York University

ypiadyk@nyu.edu

**Markus Worchsel**

TU Berlin

m.worchsel@campus.tu-berlin.de

**Marc Alexa**

TU Berlin

marc.alexa@tu-berlin.de

**Claudio Silva**

New York University

csilva@nyu.edu

**Denis Zorin**

New York University

dzorin@cs.nyu.edu

**Daniele Panozzo**

New York University

panozzo@nyu.edu

## Abstract

1 Images of a real scene taken with a camera commonly differ from synthetic images  
2 of a virtual replica of the same scene, despite advances in light transport simulation  
3 and calibration. By explicitly *co-developing* the scanning hardware and rendering  
4 pipeline we are able to achieve negligible *per-pixel* difference between the real  
5 image taken by the camera and the synthesized image on geometrically complex  
6 calibration object with known material properties. This approach provides an ideal  
7 test-bed for developing data-driven algorithms in the area of 3D reconstruction,  
8 as the synthetic data is indistinguishable from real data and can be generated at  
9 large scale. Pixel-wise matching also provides an effective way to quantitatively  
10 evaluate data-driven reconstruction algorithms. We introduce three benchmark  
11 problems using the data generated with our system: (1) a benchmark for surface  
12 reconstruction from dense point clouds, (2) a denoising procedure tailored to  
13 structured light scanning, and (3) a range scan completion algorithm for CAD  
14 models. We also provide a large collection of high-resolution scans that allow our  
15 system and benchmarks to be used without having to reproduce the hardware setup.

## 16 1 Introduction

17 Reconstruction of 3D geometry from real world measurements is a fundamental task in computer  
18 vision and data-driven methods are emerging as a promising approach to increase the reliability and  
19 quality of the reconstruction by leveraging data priors to compensate for noise or missing data. Due  
20 to the wide availability of low cost and low accuracy scanners, most of the benchmarks, datasets and  
21 algorithm development in 3D reconstruction rely on this type of data. In this work, we propose the  
22 first large scale data generator, dataset, and benchmark for high resolution and high accuracy 3D  
23 reconstruction: with rapid improvements in consumer sensors and 3D reconstruction methods the  
24 demand for high-quality 3D reconstruction will also grow rapidly. The high resolution regime poses  
25 a different set of challenges compared to low-accuracy 3D reconstruction, and its objective evaluation  
26 is more challenging, as it is not possible to rely on the data from a high resolution scanner as the

---

\*Both authors contributed equally to this work.



Figure 1: From left to right: the image of the *Pawn* acquired by a camera is faithfully reproduced by our scanning simulator. The pixel-wise difference is small after hardware calibration, and can be further reduced using our gradient descent optimization on the calibration parameters and material properties. The color represents the normalized pixel-wise difference in luminance (from 0 to 1). The height of the model is 152.4 mm.

27 ground truth, as done in some existing benchmarks (Section 2). As an alternative, we propose an  
 28 approach in which, instead of attempting to acquire ground truth data at yet higher resolution, which  
 29 is increasingly impractical, we use objects for which ground truth is known, as these are fabricated  
 30 at high accuracy from CAD models. As the number of objects that can be precisely fabricated and  
 31 scanned is necessarily limited, we couple this fabrication-based approach with an advanced simulator,  
 32 calibrated using these high-accuracy real-world objects, capable of producing unlimited amount of  
 33 synthetic data closely approximating what our hardware setup would produce. We have three major  
 34 contributions:

35 **1) Data Generator.** We introduce a hardware design for a structured-light 3d scanner paired with a  
 36 simulator that produce pixel-wise accurate replicas of the images acquired by the scanner camera after  
 37 calibration (Figure 1). This is achieved combining careful hardware choices, modern light transport  
 38 algorithms, carefully fabricated calibration geometry, and restricting the material and lighting setting.  
 39 The outcome is a system that allows to create 3D scanning dataset for structured light (a popular  
 40 choice for both high-end photogrammetry scanners, and for the low cost real-time scanners commonly  
 41 used in vision and robotics applications such as the Kinect sensor and Intel RealSense) targeting both  
 42 the high-resolution and high-accuracy regime, or low-accuracy settings, where the high accuracy of  
 43 the simulator can then be used to measure quantitatively reconstruction errors and capture real-world  
 44 source of noise.

45 **2) Dataset of Real Scans.** We provide a dataset of scans equipped with ground-truth geometry,  
 46 which enable to use our data generator and benchmarks without requiring to reproduce the hardware  
 47 setup. Any algorithm developed, trained, and evaluated with our system, can then be tested on this  
 48 dataset of real scans to evaluate its generalization to real-data with minimal efforts. The dataset  
 49 contains 3 precisely-machined calibration objects and 7 color-textured 3D printed objects scanned  
 50 with 30 degrees rotating stage intervals, for a total of  $\approx 0.5$  terabytes of data.

51 **3) Benchmark Tasks.** We combine our data generator and real scans to introduce three benchmarks  
 52 for different surface reconstruction tasks of increasing complexity: a. filtering of scanner noise from  
 53 a single range scan, b. completion of missing parts from a single range scan, and c. conversion of a  
 54 collection of range scans into a triangulated surface. The first two tasks are ideal for convolutional  
 55 neural networks that can exploit the regularity of the range scans and treat them as images, while the  
 56 last tasks is a much more challenging task that requires varying input and output sizes, and has both  
 57 (a) and (b) as subtasks. For each benchmark we provide synthetic training and test data, real-world  
 58 test data (to evaluate generalization), a procedure for evaluation, and the implementation of a baseline  
 59 method. We are planning to organize public challenges for these tasks to enrich our selection of  
 60 methods for each task.

61 The hardware blueprint for the scanner, the manufacturing protocol for the calibration objects,  
 62 the reference implementation of the simulator/calibration procedure, the real scans dataset, and  
 63 the benchmark tasks and baselines are available at [https://geometryprocessing.github.io/  
 64 scanner-sim](https://geometryprocessing.github.io/scanner-sim) to foster widespread adoption.

65 **2 Related Work**

66 We refer to the additional material for an overview of structured light scanning hardware and  
67 reconstruction algorithms. We focus here specifically on datasets and benchmarks for 3D scanning.

68 **Stereo Reconstruction.** A popular dataset in stereo reconstruction has been introduced in [18] by  
69 constructing a stereo scanner and combining it with a projector to collect ground truth annotations  
70 using structured light. This approach however is challenging to scale to the generation of large  
71 datasets as the acquisition time and effort is high. This motivated the development of the SyB3R  
72 data generators by [12] which uses a photorealistic rendering system to synthesize annotated data.  
73 However, these synthetic datasets still can not replace real world datasets but complement them by  
74 guiding the design of the physical setup. A number of setups use additional mechanical components,  
75 such as robotic arms [9] or a spherical gantry [19]; while substantially expanding the size of the  
76 dataset that can be obtained, these approaches might introduce additional sources of errors [4]. Our  
77 work aims to combine the best of both worlds, allowing acquisition of real data and the synthetic  
78 generation of data that is *indistinguishable* from the one acquired in the real world, opening the door  
79 to training models on synthetic data and test them on real one. Extending our work to generate stereo  
80 data is one of the avenues of future work, as it will only require us to add an additional camera to  
81 our system. In fact, a variety of configurations can be tested without the need to build the actual  
82 error-prone physical replicas because of the high quality simulation of these setups.

83 **Laser and Structured Light.** [1] introduced a synthetic benchmark for surface reconstruction  
84 algorithms, using a simplified rendering model simulating a laser scanner. The approach is validated  
85 by 3d printing and scanning an object with a NextEngine laser scanner and qualitatively comparing the  
86 results. They do not *calibrate* their synthetic scanner to exactly match the NextEngine scanner, in part  
87 due to the lack of details on the internals of the NextEngine scanner. [13] present a similar synthetic  
88 benchmark for structured light reconstruction algorithms with the goal of measuring the effects of  
89 illumination artifacts, including projector defocus, inter-reflections and subsurface scattering. They  
90 build a synthetic simulator using photo-realistic rendering (implemented with PBRT) that takes an  
91 object and its BRDF into account. They show that some of the discoveries made with the simulator  
92 apply to a physical scanner, but there is no attempt to match the results of the simulator and of the  
93 real scanner. To the best of our knowledge, our work is the first proposing a simulator that creates  
94 indistinguishable results from the physical scanner, thus allowing to generate large, faithful training  
95 datasets without having to scan manually thousands of objects.

96 **Synthesis of Realistic Images.** In terms of side-by-side comparisons between rendered and pho-  
97 tographed images, Phong's [16] seminal paper was first in using visual comparison of the rendered  
98 image of a sphere to a photograph to highlight the quality of his shading model. Meyer et al. [14]  
99 performs two detailed studies: comparing radiometric measurements between physical and rendered  
100 models, and a perceptual study comparing rendered images shown on a color TV monitor to the  
101 physical model using the Cornell Box [5]. Pattanaik et al. [15] calibrates a CCD camera to compare  
102 real and synthetic imagery of the Cornell Box, and attribute image differences to "mismatch between  
103 the numerical description of the scene geometry and the actual geometry". We are not aware of any  
104 existing work able to achieve a faithfulness comparable to our approach, especially on geometrically  
105 complex objects. The problem of designing a perceptual model to compare real and synthetic images  
106 has been pioneered by Rushmeier et al. [17]. In our setting, we opt for direct pixel-wise difference  
107 as our goal is to generate replicas of images to faithfully simulate a 3D reconstruction instead of  
108 producing perceptually similar images.

109 **3 Hardware Setup and Simulator**

110 A structured light scanning setup is composed of 3 main components: a camera  $C$ , a projector  $P$ , and  
111 the object being scanned  $O$ , which can be optionally placed on a rotating stage  $S$  (Figure 2). We refer  
112 to the technical appendix and to [11] for a more detailed introduction to structured light scanning.

113 **SLS Primer.** Assuming that the position of all the objects in the scene is known to sufficient  
114 accuracy, a 3D object can be reconstructed by illuminating a single pixel of the projector at a time,  
115 detecting the location of the illuminated point on the camera sensor and then triangulating. This

116 procedure is impractically slow. By projecting a set of coded patterns it is possible to establish  
 117 correspondences between camera and projector pixels from a small set of images [2, 6]. The  
 118 accuracy of the reconstruction depends on many factors, including the resolution of the camera  
 119 and projector, the lenses used, and the accuracy of the estimation of the relative position of camera  
 120 and projector. Finding this set of parameters (calibration of the scanner) is usually performed by  
 121 ‘scanning’ objects of known geometry. We carefully analyze each component of the scanning system,  
 122 selecting hardware components to minimize the noise that we cannot simulate, and we propose a  
 123 corresponding calibration procedure with the goal of minimizing reconstruction errors. Different  
 124 from existing approaches, we do not strive to make the calibration procedure simple and/or efficient,  
 125 our goal is solely on minimizing effects that cannot be recovered by optimization methods in the  
 126 computational part of the system.

127 **Hardware.** To build a highly accurate structured  
 128 light scanner, we use the following parts:  
 129 (1) A *CharuCo board* [3] for accurate calibration  
 130 of the camera and projector geometry; (2) A  
 131 *linear stage* capable of accurately reproducing  
 132 positions for the CharuCo calibration board with  
 133 known intervals to measure the focus distance  
 134 and the aperture for both the camera and the  
 135 projector; (3) A *Spectralon®* for the radiometric  
 136 calibration; (4) A set of precisely-machined  
 137 calibration objects; (5) An *Atlas 31.4 MP Camera*  
 138 by *Lucid Vision Labs* with *Edmund Optics*  
 139 *APS-C 50 mm lens* and (6) a *Texas Instruments DLP4710EVM-LC projector*.

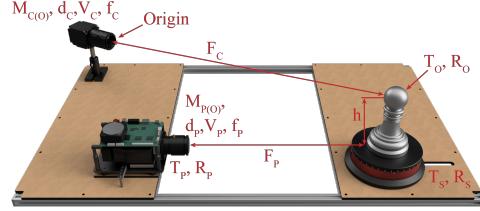


Figure 2: A diagram illustrating our structured light scanner setup and the parameters associated with each component (see supplemental material for a detailed explanation).

140 **Simulation.** We base our rendering system on the physically based renderer MITSUBA [8], extending  
 141 it to accurately model our specific motorized stages, camera, sensor, lenses, and to simulate our  
 142 projector light source. Due to space limitations, we refer to the technical supplement for a complete  
 143 description of how each hardware component in our system is parametrized (there are 31 parameters,  
 144 listed in Table 1 in the technical supplement) in Mitsuba and how each parameter is calibrated.

## 145 4 Quantitative Evaluation of Virtual and Physical Setups

146 We quantitatively evaluate the accuracy of our simulation with regards to both geometry and radiometry  
 147 in a series of controlled experiments. We compare first direct pixel-wise difference between  
 148 images (as this is a direct measurement of the faithfulness of our data generator), and then study how  
 149 this difference affects correspondence computation and reconstruction error for 3D scanning.

150 **Simulation Accuracy.** For the first experiment, we compute a pixel-wise difference between the  
 151 image acquired by the scanner of our *Pawn* calibration object and the simulated image. Figure 1  
 152 shows that the images are extremely similar, with only minor mismatches on the boundary. By  
 153 optimizing for material and projector properties, we can increase the matching even further. The  
 154 difference is more visible when we switch to a 3D printed object having a different material and a  
 155 larger fabrication tolerance (Figure 3). After reconstruction of the point clouds, the geometry of the  
 156 scanned and simulated version match closely - with a median error of 0.1 mm and a mean error of  
 157 0.17 mm (Figure 4).

158 **Correspondence Accuracy.** As an indirect validation, we compute correspondences from a full  
 159 stack of coded images for both simulated and real images, and show the pixel-wise difference in  
 160 Figure 5. Similarly to before, the difference is small.

161 **Reconstruction Accuracy.** Finally, we test reconstruction errors after triangulation. We use the  
 162 CAD model from which the object was fabricated as the ground truth, and we can thus generate a  
 163 perfect range scan via ray tracing. As before, we evaluate how the pointcloud reconstructed from our  
 164 simulation matches the pointcloud reconstructed from the physical scanner. For this, we calculated  
 165 the Hausdorff distances between the properly positioned reference geometry and the reconstructed  
 166 pointclouds. The results can be seen in Figure 6.



Figure 3: The image of the colored object (left) is very similar to the simulated image (middle). We show the normalized pixel-wise differences in luminance on the right. The colors are different due to the lack of calibration during 3D printing.



Figure 4: Point clouds from real and simulated scans and their error. The mean error is 0.17 mm. Note that some parts of the reconstructed object differ due to lighter and less saturated colors of the 3D-printed object.



Figure 5: Color-coded horizontal correspondence indices retrieved from the physical setup (left), the simulation (middle) and the difference between the indices for Gray code scanning (right).

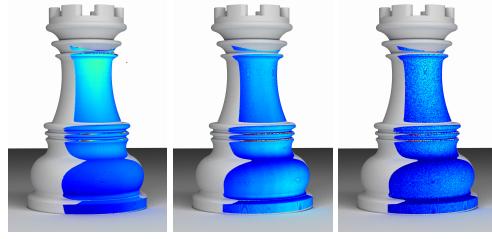


Figure 6: Distances between the reference geometry and the reconstructed point clouds from the physical scanner (left), simulated scanner before (middle) and after (right) optimization of material and projector parameters. We use the same colormap as Figure 4.

## 167 5 Data Acquisition and Synthetic Dataset Generation

168 **Data Acquisition.** We scanned a collection of objects using our hardware setup: 3 calibration  
 169 objects and 7 colored 3D printed objects. For each object, we acquired 12 views, rotating the object  
 170 around its up-axis in steps of 30 degrees, at a resolution of 6464x4852. A typical acquisition time per  
 171 object is several hours depending on the exposures used for HDR image capture. For all objects we  
 172 also provide the reference geometry, allowing to use this data and mix it with synthetic data generated  
 173 using our simulator. With this collection of acquired data, it will be possible to benchmark algorithms  
 174 without having to reproduce the hardware setup. For further details about the acquired data, see the  
 175 full description in the supplementary material.

176 **Software and Hardware Specifications.** The calibrated simulator is distributed as source code  
 177 and a Docker container. It supports custom coded patterns, editing of scanner geometry (e.g. the  
 178 baseline between camera and projector), editing of camera and/or projector resolution, and scanning  
 179 of arbitrary objects provided as densely triangulated surfaces. It is divided into three parts: (1) The  
 180 rendering pipeline that reads in geometry, projection patterns, material and transformation type (fixed  
 181 pose for matching, turntable rotation for data generation and scanning, and random rotation for  
 182 calibration purposes and data generation). (2) The decoding pipeline that takes the rendered images  
 183 and decodes the patterns with a decoding function (we supply decoding for Gray code, unstructured  
 184 light [2], and micro-phase shifting patterns [6]). (3) A reconstruction pipeline that reads in the  
 185 rendered images from step 1 and the correspondences from step 2 and produces the reconstructed  
 186 point cloud (with normals and colors) and depth map as well as the ground truth point cloud and  
 187 depth map. In addition, we provide the hardware blueprints and the code for rerunning the scanner  
 188 calibration, to allow the complete reproduction of our system and results.

189 **Large Scan Dataset.** Our dataset contains the physical scans of 4 calibration objects (including  
 190 flat plane) and 7 textured 3D printed objects (see Figure 8). We enrich this dataset with simulated  
 191 scans from 1000 mechanical objects from the ABC dataset (see Figure 7). Each object is scanned  
 192 with Gray code patterns in 10 random rotations in simulation and from 12 different directions on the  
 193 rotating stage for the physical setup. In total, the dataset contains 192 physical (some calibration  
 194 objects are scanned in multiple configurations, e.g. with or without ambient illumination) and 10k

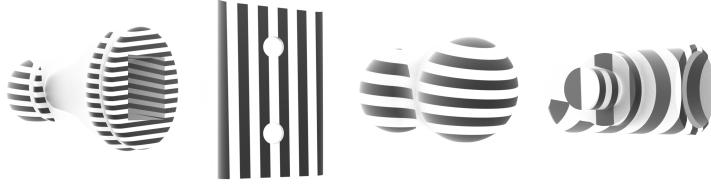


Figure 7: Four example models illuminated with Gray code patterns from the LSD.



Figure 8: The 3D-printed textured objects (left) and the respective simulated renderings (right).

195 synthetic scans, annotated with ground truth depth and 3D geometry. For convenience, the synthetic  
 196 dataset is split into multiple chunks of 250 scans each. For one chunk, the processing time on a 6-core  
 197 machine was roughly 50 hours and is mostly depending on the render-resolution and the amount  
 198 of samples per pixel. To the best of our knowledge, this dataset is unique and offers a next level  
 199 benchmark for the evaluation of traditional and data-driven algorithms for processing 3D geometry  
 200 (Section 6). We also envision this dataset and the scan simulator to become useful for developing  
 201 postprocessing, correspondence, and texture reconstruction algorithms. Our data does not contain  
 202 personally identifiable information or offensive content.

## 203 6 Benchmark

204 The scanning datasets generated using our simulator contain ground truth and, at the same time, are  
 205 ‘faithfully’ reproducing the errors introduced by a real scanner, such as camera/projector distortion  
 206 and defocus, radiometric errors, and decoding errors. We introduce a set of three benchmarks,  
 207 two targeting low-level processing of individual range scans (which due to their regularity are a  
 208 good match for cnn-based approaches) and the last one targeting surface reconstruction, a problem  
 209 requiring unstructured approaches to scale to high resolution.

### 210 6.1 Data-Driven Processing of Depth Images

211 The data we generated, containing both real-world scan and highly faithful simulated data (both with  
 212 ground truth geometry), is ideal for benchmarking low-level processing of depth scans. We propose  
 213 benchmarks for two common tasks: 1) scan denoising to remove artifacts that arise from calibration  
 214 problems and noise in the scanning process and 2) hole filling to complete gaps in the depth images.  
 215 We also provide the results of baseline methods for each task.

216 **Denoising.** Minor inaccuracies in the calibration and errors in triangulation result in subtle errors  
 217 in the reconstructed depth (see Figure 9). In contrast to outliers that can occur when scanning  
 218 highly specular objects, these imperfections are located on a much smaller scale (millimeter to  
 219 submillimeter), a challenging problem that is very relevant for metrology applications. Despite the  
 220 large body of work in machine learning and depth map processing, we are unaware of any related  
 221 work in the structured light scanning context that tries to correct errors at this scale in a data-driven  
 222 way, likely due to the lack of a data generator or acquisition setup for this problem. We use our  
 223 approach to build such a dataset and propose a baseline method that casts the task as an image-to-  
 224 image translation problem and train a convolutional neural network on pairs of reconstructed and  
 225 ground truth depth maps. The dataset is composed of 250 samples of reconstructed and ground truth  
 226 depth maps. We propose three baselines for this task: a simple bilateral filter whose parameters are  
 227 tuned with a sweep on the training data, a mesh-based Laplacian smoothing that is constrained to  
 228 move the depth map vertices parallel to the camera’s optical axis, and a novel data-driven approach  
 229 inspired by other depth denoising [21] and image-to-image translation approaches [7]. The details on  
 230 the latter approach are in the additional material.

Table 1: The trained denoising model reduces the mean absolute error for synthetic and physical test scans and outperforms the non data-driven methods. Since the loss function doesn't consider normal orientations, the traditional methods outperform CNN with respect to this metric.

	Normal Angle Difference					
	$\downarrow$ MAE [mm]	$\downarrow$ RMSE [mm]	$\downarrow$ Mean [ $^\circ$ ]	$\uparrow$ 3 [%]	$\uparrow$ 5 [%]	$\uparrow$ 10 [%]
Scan	0.6906	6.6553	26.20	2.73	6.48	19.85
CNN	<b>0.6199</b>	6.6111	19.69	5.92	14.16	38.29
Bilateral	0.6851	6.6374	<b>3.79</b>	<b>77.69</b>	<b>88.53</b>	<b>95.29</b>
Laplacian	0.6874	<b>6.6072</b>	6.72	45.86	67.40	86.94

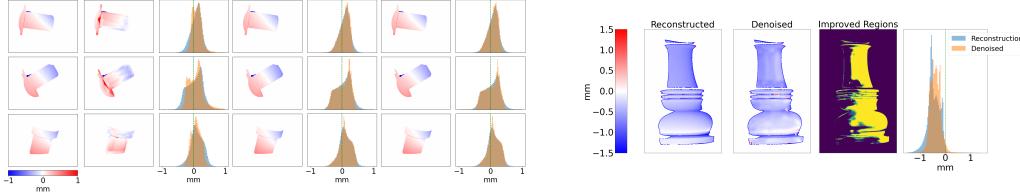


Figure 10: Errors of virtual scans before and after denoising. From left to right: signed error of the depth map recovered by SLS, signed error and distribution after depth denoising with our machine learning model, a bilateral filter, and Laplacian smoothing.

231 Table 1 and Figure 10 show the benefit of  
 232 the data-driven approach compared to the  
 233 two baseline methods. Figure 11 shows  
 234 denoising results on a real scan from the  
 235 physical scanner. As for the virtual test  
 236 objects, the model improves the error dis-  
 237 tribution by reducing its standard devia-  
 238 tion. Note that we can compute the error  
 239 of the real *Rook* scan because for the cal-  
 240 ibration objects we have established an  
 241 accurate alignment to the simulated envi-  
 242 ronment. The point clouds in Figure 11  
 243 show the reprojected depth maps in 3D  
 244 space. Typical reconstruction artifacts  
 245 like the staircase pattern are removed or  
 246 partially attenuated in the denoised scan,  
 247 which confirms that the model learned to  
 248 correct specific artifacts of the real scan-  
 249 ning process by analyzing statistics of the  
 250 virtual data.

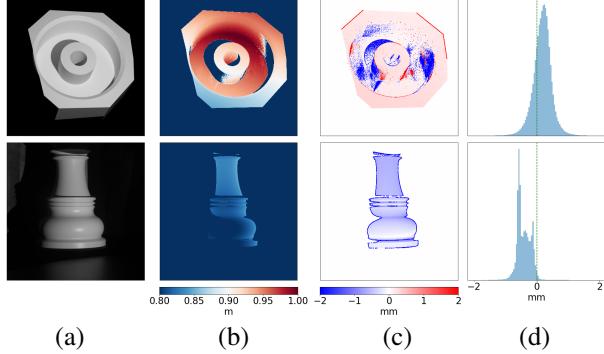


Figure 11: Results of our denoising model on a real depth map from the physical scanner. The model trained purely on synthetic data improves most parts of the scan.

Figure 9: Small-scale depth error in the scans of a virtual sample from the LSD (top row) and the real *Rook* cal-  
 bration object (bottom row). Camera image with white  
 projector light (a), the reconstructed depth map (b), its  
 signed error to the ground truth depth (c), and the error  
 distribution (d). Errors larger than 2 mm are clamped in  
 the error maps and are not considered in the distributions.

251 **Shape Completion.** Shape Completion (sometimes also referred to as shape inpainting, hole filling,  
 252 or depth completion) is a common postprocessing task in 3D scanning. We generate 100k crops of  
 253 size  $320 \times 256$ . Each sample consists of an image crop from the diffusely illuminated rendering,  
 254 the corresponding ground truth depth map as well as the reconstructed depth map from the virtual  
 255 scanning process. In contrast to images with projector illumination, the diffusely illuminated ones  
 256 provide the network with information about areas not hit by projector light rays. See Figure 13 for  
 257 examples of ambient and directed illumination and their difference. We used the data-driven approach  
 258 by [20] for depth completion of indoor scenes as a baseline, see the additional material for more  
 259 information on the training regime and architecture. For the training of their model, we modify the  
 260 data loader to read samples from our synthetic dataset in the form of cropped patches. The training is  
 261 run for 100 epochs. We did not further modify their approach, so we refer to the original paper for  
 262 more details on the model and training process [20].

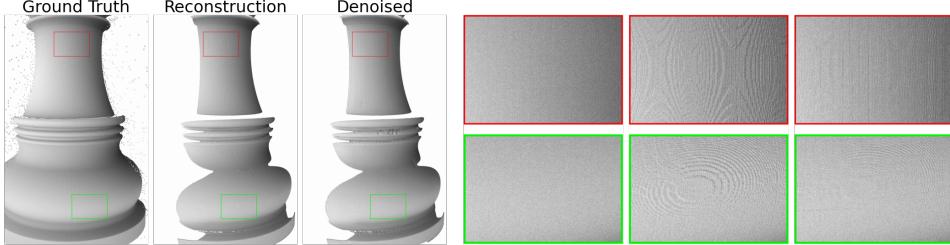


Figure 12: Point clouds of real scans before and after denoising with our network trained purely on synthetic data. The staircase artifacts in the SL scan are significantly reduced.

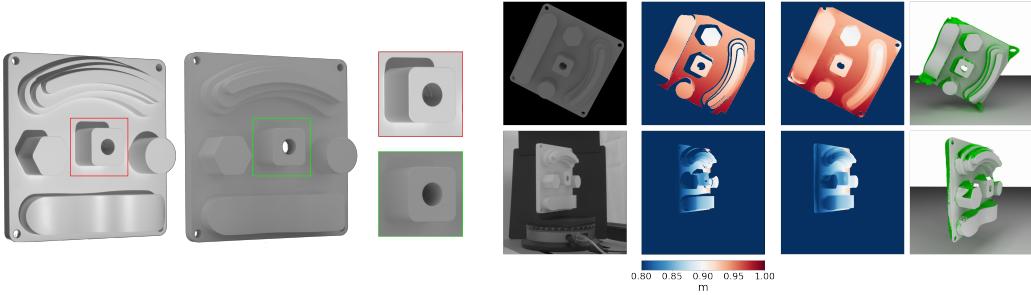


Figure 13: From left to right: the *Shapes* object illuminated with white projector light and ambient light as well as two close up views. Both images are acquired with the hardware scanner.

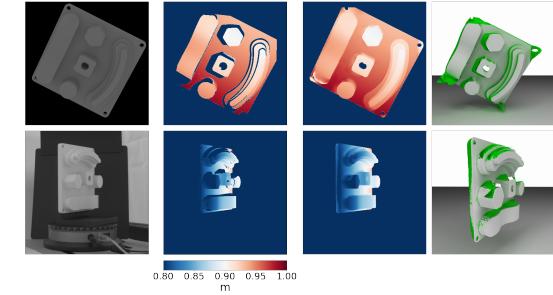


Figure 14: Holefilling results for a virtual scan (top rows) and a real scan (bottom row). From left to right: Ambient image, reconstructed depth map, reconstructed depth map with filled holes, and corresponding rendering with green indicating filled areas.

263 For inference on full resolution input, we apply the trained model on overlapping tiles and merge the  
 264 results. The final completed depth map is a hybrid of the reconstructed depth map from the scanning  
 265 process and the output of the trained model: We preserve valid depth values from the scan and use the  
 266 model to fill in only the missing areas. To quantify which areas to fill specifically, we use an object  
 267 mask extracted from the ambient image. Figure 14 shows shape completion results for both virtual  
 268 scans from the simulated scanner and a real scan from the hardware scanner. The preliminary results  
 269 suggest that the model trained purely on synthetic data can generalize to our real world 3D scans.  
 270 The resulting depth maps have less holes from occlusion and are better suitable, for example, for  
 271 surface reconstruction tasks. While these results still rely on an algorithm introduced for indoor depth  
 272 processing, our datasets open the door to future methods specifically designed for shape completion  
 273 in the structured light scanning context.

## 274 6.2 3D Reconstruction

275 A common downstream task in structured  
 276 light scanning is assembling a full model  
 277 of a captured object. For this task, mul-  
 278 tiple depth maps are fused into one point  
 279 cloud and a consistent surface is extracted  
 280 from this unstructured representation. Our  
 281 system can be used to simulate the scan of  
 282 any 3D object. We provide scripts to au-  
 283 tomatically generate an arbitrary number  
 284 of range scans and fuse them into a single  
 285 point cloud that can be processed with a  
 286 3D reconstruction algorithm. The recon-  
 287 structed geometry can then be compared  
 288 with the reference geometry using Haus-  
 289 dorff distance to quantitatively measure the  
 290 reconstruction error. We provide the script  
 291 to generate the data and evaluate the reconstructions, and as an example we use them to evaluate the

Table 2: One-sided Hausdorff distance (in mm) for different parameterizations of the Poisson surface reconstruction.  $D$  is maximum octree depth and  $N_S$  is the minimum number of samples per node.

		$N_S$				
		1	5	10	15	20
$D$	5	3.600	3.607	3.610	3.611	3.601
	6	18.937	2.334	2.333	2.334	2.324
	7	18.222	13.888	1.576	1.570	1.572
	8	16.531	13.667	1.365	1.394	1.380
	9	17.871	13.869	1.334	1.326	<b>1.308</b>
	10	18.390	15.828	1.337	1.342	1.354
	11	18.416	16.187	1.345	1.348	1.328

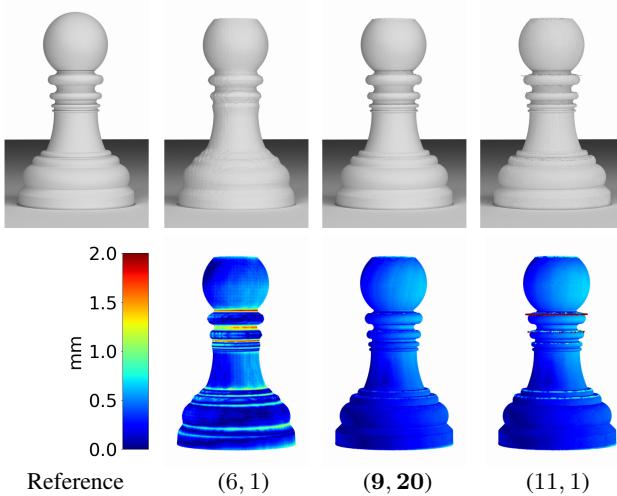


Figure 15: Error of reconstructed surfaces for selected parameter combinations. Reference mesh next to surfaces reconstructed from virtual scans (top row) and their error with respect to the reference (bottom row). The parameters  $(D, N_S)$  with lowest error are highlighted.

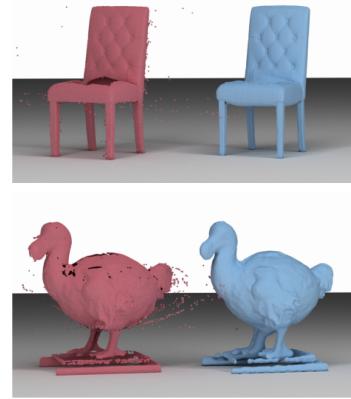


Figure 16: Point clouds (red) next to reconstructed surfaces (blue) for a couple of test objects from the LSD. The point clouds are from real scans. The Poisson surface reconstruction parameters are from parameter sweep on virtual scans.

292 screened Poisson surface reconstruction (SPSR) [10] algorithm. This is not a data-driven approach,  
293 but its parameters are context dependent and have to be adapted according to the input data. We use  
294 our system to automatically find optimal parameters for a given object.

295 We perform a parameter sweep for the maximum depth  $D$  of the octree over which the equation  
296 is solved and the minimum number of samples  $N_S$  in each octree node over the *Pawn* and *Rook*  
297 calibration objects. Table 2 shows the one-sided Hausdorff distances in millimeters from the meshes  
298 recovered with SPSR to the ground truth surface. The values are averaged over the runs for both  
299 test objects. Figure 15 provides qualitative insights for some parameter combinations and the error  
300 distribution over the surface. Note that we trimmed the reconstructed meshes above and below a  
301 certain height to avoid limitations imposed by the capture setup (e.g. we cannot reconstruct the  
302 bottom of the object as it is invisible). The parameters that achieve the best score in the sweep  
303 produce a surface that is very detailed, yet is not dominated by the noise in the input point cloud.  
304 We show that the parameters discovered in the virtual environment generalize to the real world by  
305 applying them to a set of point clouds acquired by the real scanner (Figure 16).

## 306 7 Concluding Remarks

307 We built an accurate structured light scanner and a corresponding simulation pipeline that, after  
308 calibration, can reproduce pixel-accurate replica of the images acquired by the scanner’s camera. The  
309 acquired data and the simulator allow us to construct 3D scanning datasets with annotated ground  
310 truth, which are ideal for developing datasets to support the development of data-driven algorithms  
311 for surface reconstruction, providing a quantitative criteria to evaluate their performance. We do not  
312 foresee any potential negative societal impact of our work.

313 **Limitations.** The high accuracy of our system has been achieved by controlling the lighting setup  
314 and restricting the choice of materials. Despite its apparent simplicity, we believe this is an important  
315 setting, as it allows to establish a baseline performance for 3D reconstruction methods. Extending  
316 our work to more realistic lighting setups and to materials with non negligible subsurface scattering  
317 is an exciting avenue for future work.

## 318 References

- 319 [1] Matthew Berger, Joshua A. Levine, Luis Gustavo Nonato, Gabriel Taubin, and Claudio T. Silva.  
320 A benchmark for surface reconstruction. *ACM Trans. Graph.*, 32(2), April 2013.

- 321 [2] V. Couture, N. Martin, and S. Roy. Unstructured light scanning to overcome interreflections. In  
 322 *2011 International Conference on Computer Vision*, pages 1895–1902, Nov 2011.
- 323 [3] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic  
 324 generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*,  
 325 47(6):2280 – 2292, 2014.
- 326 [4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving?  
 327 the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern  
 328 Recognition*, pages 3354–3361, 2012.
- 329 [5] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling  
 330 the interaction of light between diffuse surfaces. *SIGGRAPH Comput. Graph.*, 18(3):213–222,  
 331 January 1984.
- 332 [6] Mohit Gupta and Shree K. Nayar. Micro phase shifting. In *CVPR*, pages 813–820, 2012.
- 333 [7] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional  
 334 adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition  
 (CVPR)*, pages 5967–5976, 2017.
- 335 [8] Wenzel Jakob. Mitsuba renderer, 2010. <http://www.mitsuba-renderer.org>.
- 336 [9] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale  
 337 multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern  
 338 Recognition*, pages 406–413, 2014.
- 340 [10] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans.  
 341 Graph.*, 32(3), July 2013.
- 342 [11] Douglas Lanman and Gabriel Taubin. Build your own 3d scanner: optical triangulation for  
 343 beginners. In *International Conference on Computer Graphics and Interactive Techniques,  
 344 SIGGRAPH ASIA 2009, Yokohama, Japan, December 16-19, 2009, Courses Proceedings*. ACM,  
 345 2009.
- 346 [12] Andreas Ley, Ronny Hansch, and Olaf Hellwich. *SyB3R: A Realistic Synthetic Benchmark for  
 347 3D Reconstruction from Images*, pages 236–251. Springer International Publishing, 2016.
- 348 [13] Esdras Medeiros, Harish Doraiswamy, Matthew Berger, and Claudio T. Silva. Using physically  
 349 based rendering to benchmark structured light scanners. *Computer Graphics Forum*, 33(7):71–  
 350 80, 2014.
- 351 [14] Gary W. Meyer, Holly E. Rushmeier, Michael F. Cohen, Donald P. Greenberg, and Kenneth E.  
 352 Torrance. An experimental evaluation of computer graphics imagery. *ACM Trans. Graph.*,  
 353 5(1):30–50, 1986.
- 354 [15] Sumanta N. Pattanaik, James A. Ferwerda, Kenneth E. Torrance, and Donald P. Greenberg.  
 355 Validation of global illumination simulations through CCD camera measurements. In *5th Color  
 356 and Imaging Conference, CIC 1997, Scottsdale, Arizona, USA, November 17-20, 1997*, pages  
 357 250–253. IS&T - The Society for Imaging Science and Technology, 1997.
- 358 [16] Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317,  
 359 1975.
- 360 [17] Holly E. Rushmeier, Gregory J. Ward, Christine D. Piatko, Phil Sanders, and Bert W. Rust.  
 361 Comparing real and synthetic images: Some ideas about metrics. In Pat Hanrahan and Werner  
 362 Purgathofer, editors, *Rendering Techniques '95, Proceedings of the Eurographics Workshop in  
 363 Dublin, Ireland, June 12-14, 1995*, Eurographics, pages 82–91. Springer, 1995.
- 364 [18] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nesic, Xi Wang,  
 365 and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In  
 366 Xiaoyi Jiang, Joachim Hornegger, and Reinhard Koch, editors, *GCPR*, volume 8753 of *Lecture  
 367 Notes in Computer Science*, pages 31–42. Springer, 2014.

- 368 [19] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation  
 369 of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on*  
 370 *Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528, 2006.
- 371 [20] Dmitry Senushkin, Mikhail Romanov, Ilia Belikov, Anton Konushin, and Nikolay Patakin.  
 372 Decoder modulation for indoor depth completion. *arXiv preprint arXiv:2005.08607*, 2020.
- 373 [21] Shi Yan, Chenglei Wu, Lizhen Wang, Feng Xu, Liang An, Kaiwen Guo, and Yebin Liu. Ddrnet:  
 374 Depth map denoising and refinement for consumer depth cameras using cascaded cnns. In  
 375 Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer*  
 376 *Vision – ECCV 2018*, pages 155–171, Cham, 2018. Springer International Publishing.

## 377 A Paper Checklist

- 378 1. For all authors...
  - 379 (a) Do the main claims made in the abstract and introduction accurately reflect the paper's  
   380 contributions and scope? **[Yes]**
  - 381 (b) Did you describe the limitations of your work? **[Yes]** See Section 7.
  - 382 (c) Did you discuss any potential negative societal impacts of your work? **[Yes]** We do not  
   383 foresee any potential negative societal impact of our work. (Section 7).
  - 384 (d) Have you read the ethics review guidelines and ensured that your paper conforms to  
   385 them? **[Yes]**
- 386 2. If you are including theoretical results...
  - 387 (a) Did you state the full set of assumptions of all theoretical results? **[N/A]**
  - 388 (b) Did you include complete proofs of all theoretical results? **[N/A]**
- 389 3. If you ran experiments (e.g. for benchmarks)...
  - 390 (a) Did you include the code, data, and instructions needed to reproduce the main experi-  
   391 mental results (either in the supplemental material or as a URL)? **[Yes]** Code, data, and  
   392 instructions are available at: <https://geometryprocessing.github.io/scanner-sim/>
  - 393 (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they  
   394 were chosen)? **[Yes]**
  - 395 (c) Did you report error bars (e.g., with respect to the random seed after running experi-  
   396 ments multiple times)? **[N/A]**
  - 397 (d) Did you include the total amount of compute and the type of resources used (e.g., type  
   398 of GPUs, internal cluster, or cloud provider)? **[Yes]** Yes, in Section 5.
- 399 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
  - 400 (a) If your work uses existing assets, did you cite the creators? **[Yes]**
  - 401 (b) Did you mention the license of the assets? **[Yes]**
  - 402 (c) Did you include any new assets either in the supplemental material or as a URL? **[Yes]**  
   <https://geometryprocessing.github.io/scanner-sim/>
  - 403 (d) Did you discuss whether and how consent was obtained from people whose data you're  
   404 using/curating? **[N/A]**
  - 405 (e) Did you discuss whether the data you are using/curating contains personally identi-  
   406 fiable information or offensive content? **[Yes]** Our data does not contain personally  
   407 identifiable information or offensive content. (Section 5).
- 409 5. If you used crowdsourcing or conducted research with human subjects...
  - 410 (a) Did you include the full text of instructions given to participants and screenshots, if  
   411 applicable? **[N/A]**
  - 412 (b) Did you describe any potential participant risks, with links to Institutional Review  
   413 Board (IRB) approvals, if applicable? **[N/A]**
  - 414 (c) Did you include the estimated hourly wage paid to participants and the total amount  
   415 spent on participant compensation? **[N/A]**