Vision, Modeling, and Visualization (2013)
Michael Bronstein, Jean Favre, and Kai Hormann (Eds.)

Simulation of Time-of-Flight Sensors using Global Illumination

S. Meister[1] and R. Nair[1] and D. Kondermann[1]

[1]Heidelberg Collaboratory for Image Processing, University of Heidelberg, Germany

Abstract

Time-of-Flight (ToF) cameras use specialized sensors and modulated infrared light to simultaneously obtain depth, amplitude and intensity images. Depth images from such cameras suffer from various errors which exhibit a more complex behavior than traditional intensity images. Of these errors, the phenomenon of multi-reflection or multi-path interference poses the biggest challenge to researchers. It is caused by indirect light paths between camera and light source and is therefore dependent on scene geometry. While simulated data can be used for ground truth evaluation and whitebox testing, current simulators do not model multipath effects. The method we present is capable of simulating all scene-dependant effects by taking global illumination into consideration. This is accomplished by modifying a bidirectional path tracing algorithm such that it takes the time-dependent propagation of modulated light in a scene into consideration. Furthermore, by combination of the proposed method with a previous hardware simulator we are capable of reproducing all effects in ToF cameras. The system was validated both on test targets with known real Time of Flight camera responses as well as qualitatively on a more complex room scene. The simulator as well as the source code is available at `http://hci.iwr.uni-heidelberg.de/Benchmarks/`.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—

1. Introduction

Time of Flight (ToF) imaging is a depth acquisition modality with many interesting properties such as independent, per pixel depth measurements, well localized object contours and robustness towards ambient light. It is interesting for various applications ranging from 3D-reconstruction over HCI to driver assistance systems. However, depth maps obtained with these cameras show a variety of systematic and statistical errors such as depth noise, flying pixels or multi-path interference. Therefore, the development of methods to compensate for these effects is an active field of research.

Validation and benchmarking of such algorithms is a challenging task. Ground truth comparisons are usually limited to images taken in controlled lab conditions, as only here sufficiently correct ground truth can be acquired. Recent research by Wulff et al. [WBSB12] or Meister et al. [MK11] suggest that computer generated images are in many cases suitable alternatives for algorithm validation. Unlike measured ground truth where the challenges lie in the measurement accuracy and reference data alignment, simulation requires all observable effects to be modeled properly by the system. For ToF imaging, none of the current methods are capable of simulating all ToF related errors simultaneously.

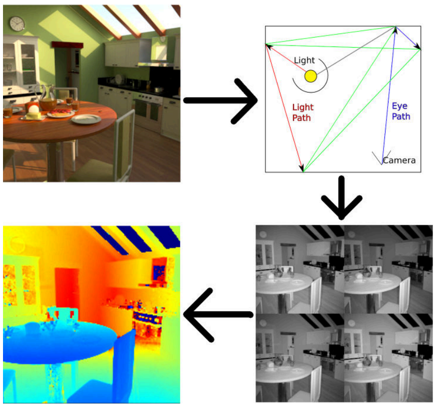

Figure 1: *Processing Pipeline of our algorithm: Starting from a scene description path tracing based global illumination is used to generate four raw phase frames. The scene depth can then be recomputed from these phase images.*

DOI: 10.2312/PE.VMV.VMV13.033-040

The method we present here is capable of simulating all scene dependent effects in ToF data including multipath effects caused by global lighting. To accomplish this, we borrow from methods known in the computer graphics community to correctly simulate light propagation.

There, global illumination algorithms are used to simulate both direct and indirect light as well as material properties like reflection, transmission and scattering. However, these methods assume that light propagation is instantaneous and each frame is in a steady state. Our method extends a path tracing global illumination method to also consider the finite speed of light. A rundown of the pipeline is given in Fig.1.

We will focus on the simulation of multipath interference as well as artifacts caused by indirect illumination, as they have not been addressed in literature yet. The rest of this paper is organized as follows. After giving a brief overview of related work in Section 2, we will give an introduction to ToF imaging and error sources in Section 3. We will then introduce our simulator in Section 4 and discuss evaluation experiments in Section 5.

2. Related Work

Multiple methods for the classification, measurement and removal of errors inherent in time-of-flight images have been described, for example by Plaue [Pla06], Schmidt [Sch08], Lindner and Kolb [LK07] or Falie and Buzuloiu [FB08].

The estimation and compensation of multipath interference in time-of-flight cameras has been the focus of recent investigation e.g. by Fuchs [Fuc10], Jiménez et al [JPMP12] or Dorrington et al. [DGC*11]. Most of the described methods assume that the involved materials have lambertian reflectance properties or take only simple one-time reflections into consideration. Scattering inside the camera casing and lens is related to this problem and has been investigated by Karel et al. [KGP12].

A method for simulating the *sensor* of time-of-flight cameras has been described by Schmidt [Sch11].

In contrast we simulate the *light propagation* in a scene. In computer graphics Smith et al. [SSD07] were the first to postulate a generic time-dependent form of the rendering equation. Regarding time-of-flight cameras, this has only been performed by Keller, Kolb et al. [KOKP07] [KK09]. In their work they presented an extensive framework for time-of-flight simulation based on scanline-rendering.

Our proposed method enhances previous methods by simulating multipath interference via path tracing. The usage of Monte Carlo methods to simulate multiple light scatterings is not new as it has already been employed for the evaluation of LiDAR systems. Examples are the works of Gordon [Gor82] describing the effect of laser reflections on oceanic surfaces or the works by Kunkel et al. [KW76] who investigated multiple atmospheric scatterings.

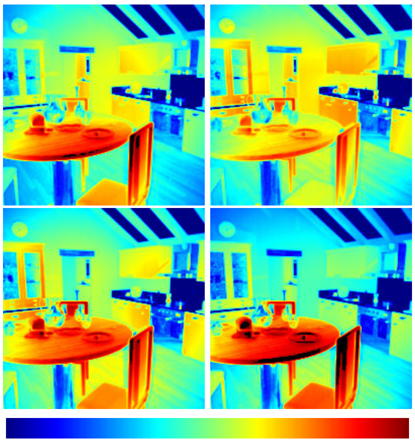

Figure 2: *Raw phase images of synthetic* ***Kitchen*** *scene (colorcoded logarithm). Well visible are the intensity changes at different depths. Reflecting surfaces such as the chrome on the oven door appear very dark as they reflect only indirectly illuminated surfaces.*

In the following two sections, we will summarize the working principles of time-of-flight cameras as well as the bidirectional path tracing algorithm and our modifications to it.

3. Time-of-Flight Cameras

Continuous-wave time-of-flight depth cameras use a variation of the following principle: the intensity of an active light source (LEDs mounted coaxial to the camera optic) is modulated with a Megahertz-range frequency. The image sensor correlates the input signal created by the reflected light with the modulation reference signal to calculate the phase shift ϕ. This phase-shift is proportional to the traveled light distance modulo the ambiguity range ($7.5m$ for a modulation frequency of 20 MHz) due to the periodicity of the modulation signal. Some laser range finders use a similar principle although a time-of-flight camera can perform the depth calculation for a whole image simultaneously.

At least 4 raw frames (I_{0-3}) measured at four different phases are necessary to compute the depth. An example for a synthetic **Kitchen** scene can be seen in Figure 2 (see section 5 for details).

For phase shifts λ of $\frac{1}{2}\pi$, π and $\frac{3}{2}\pi$, a modulation frequency f and light speed c the distance d of an object to the camera can be computed as:

$$d \propto atan\left(\frac{I_3 - I_1}{I_0 - I_2}\right) \cdot \frac{c}{f}$$

Different sensor models may capture more raw frames to compensate for noise or background illumination. More details on the CamCube sensor and a detailed error analysis can be found in [Pla06] or [Sch08].

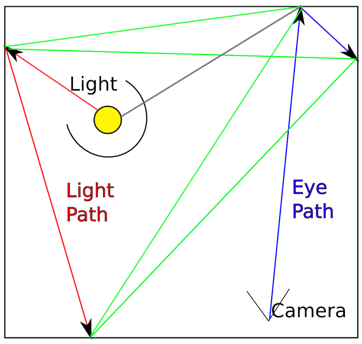

Figure 3: *Principle of Bidirectional Path Tracing: eye paths are blue, light path are red. At each intersection point the paths are connected (green lines). Also the algorithm tries to connect eye path intersections directly with the light sources (gray path, here invalid due to shadowing).*

There are multiple sources of errors which cause the result of this equation to differ from the true object-to-camera distance. Apart from different random noise sources (electronics, photon noise, etc.) the most prevalent error is multipath or interreflection. It results in flying pixels at object corners or overestimated depths. Multipath error is caused by light which did not travel a direct path from the light into the scene and back to the camera but was reflected multiple times. In a computer graphics context such effects could be summarized as global illumination. Due to these errors the sensor reports a depth value which is typically higher than the true depth as well as dependent on intensity. Additional errors e.g. the wiggling error, which arise from the fact that the modulation signal is not perfectly sinusoidal can be removed by calibration. See the works by Rapp [RFHJ08] or Erz and Jähne [EJ09].

4. Phase Modulated Bidirectional Path Tracing

Bidirectional Path Tracing as introduced by Lafortune and Willems [LW93] and Veach and Guibas [VG95] solves the global illumination problem of the *rendering equation*, which was first described in detail by Immel et al. [ICG86] and Kajiya [Kaj86]. It synthesizes an image in the following way: A view ray from the camera is sampled and traced into the scene. Once it intersects a polygon a secondary ray is sampled recursively according to the local bidirectional reflectance distribution function (BRDF) until a maximum recursion depth is reached. In a similar manner light rays are sampled from all light sources in the scene. All view ray intersection points and all light ray intersections are then connected to evaluate the light contributions along the view ray. These contributions are then added up to acquire the final light intensity at the pixel the view ray was sampled from. See Figure 3 for a visualization.

Our modification of the bidirectional path tracer is based on the BiDir Integrator of the open source LuxRender project [L], which itself is based on the pbrt render engine by Pharr and Humphrey [PH10].

It computes the path lengths for all view and light rays and then weights the contributions based on the complete path length, and the light modulation. A single light contribution L_i is then modulated with a factor m_i as follows:

$$m_i = \left(0.5\cos\left(\frac{d\,f\,2\pi}{c} + \lambda\right) + 0.5\right) * I_m$$

- d distance
- f modulation frequency
- c speed of light
- λ phase shift between modulation and sensor signal
- $I_m = [0..1]$ modulation intensity
- $O_m = [0..1]$ modulation offset with $O_m + I_m = 1$

The final intensity L_r observed in a pixel is then:

$$L_r = \sum_i L_i \cdot (m_i + O_m)$$

Usually, for each pixel multiple paths and light contributions are accumulated and the final response for a pixel is computed by averaging the contributions. This also involves the use of a spatial filter which can mimic the point-spread-function of a real camera system. This is similar to a supersampling approach and allows us to correctly simulate flying pixels due to mixed depth cues at a single sensor pixel.

Like in most rendering algorithms, it is possible to shift priority between speed of execution, memory consumption and physical realism. The most influential parameters of the path tracer are the maximum recursion depth for light and eye paths as well as the number of samples per pixel. As exspected, at a eye recursion depth of one no multipath effects can be observed. When the light path depth is set to zero, the algorithm is reduced to a unidirectional path tracer. The optimal value is scene dependent but as sensor and lightsource on real ToF cameras are usually very close together direct illumination dominates and path depths of approximately 8 are sufficient for simple test scenes. We draw this conclusion as the simulation in Figures 8 and 9 with path length 8 was indistinguishable from a simulation with path length 16. However, more complex setups, especially with transparent or mirror like materials may necessitate higher path lengths. Considering this a bidirectional tracer may seem needlessly complex for the given problem when a unidirectional tracer may be sufficient. But as the implementation is very generic, one can always reduce the light path length for faster convergence when the scene geometry allows it while keeping the advantages of a bidirectional tracer when needed.

As most global illumination methods, path tracing is a statistical process and will converge to a correct solution as the number of samples per pixels increases. As each of the phase images is rendered independently, the statistical image noise can influence the produced depth maps significantly. Usually, a few hundred raytracing samples per pixel are enough for visually convincing results, but in our case a the number of samples should be one order of magnitude higher to create accurate depth maps.

Caching of intermediate results and direct computation of all required phase shifts could reduce this problem in future versions of the algorithm, although in that case it would not be possible anymore to simulate motion artifacts as each frame would be limited to exactly the same geometry.

5. Experiments

In this work we focus on the simulation of the PMDTec CamCube 3 sensor, although the simulation principles can easily be applied to other cameras as well. To evaluate our algorithm we compared the generated depth maps with results from a real PMDtec CamCube 3, the simulation method described by Schmidt [Sch11], the simulator by Keller et al. [KK09], as well as ground truth data. The comparison with ground truth is important as it allows us a separate quantitative statistical and systematic error analysis of all involved imaging systems (synthetic and real).

We used two real scene setups as well as two purely synthetic ones to test the algorithms. The first synthetic scene, labeled **Corner** is intended as a proof of concept. The displayed object is a 90-degree corner with three different specular reflecting materials which only differ in the brightness of their diffuse channel (see Figure 5 left). We used Luxrenders 'glossy' shader which simulates a lambertian base material with a coating based microfacet model with a schlick distribution (See Luxrender documentation for details).

The second synthetic scene, labeled **Kitchen** [Cen] shows a typical household kitchen with various reflecting and transparent objects. It is mainly used to demonstrate that the algorithm can deal with complex setups. A photorealistic rendering is display in Figure 13 and raw phase images of the scene are displayed in Figure 2. The scene **RealCorner** is similar to the **Corner** scene, but a real target of the same geometry was used to obtain and compare data from a real ToF camera (Figure 5 right). The scene **Box** which was also captured with a real camera contains a wooden box with a more complex geometry and different materials (wood as well a differently colored paper) (Figure 4).

Ground truth was either directly available from the synthetic scene description or created by using objects with known geometry and 3d meshes whose position relative to the camera was estimated using manually annotated 2d-to-3d correspondences. The 2d-to-3d pose estimation and camera calibration was performed with an Levenberg-Marquardt optimization from the OpenCV image processing library [Bra00]. The meshes have an accuracy of $\approx 1mm$ and reprojection errors of the internal and external camera calibration were below ≈ 0.5pixel. Hence we assume that the error in the ground truth depth maps is lower than $\approx 1mm$ which is well below the standard error ranges of most time-of-flight cameras. Unless stated otherwise we converted the radial depth values (distance from the camera center) from the real and synthetic cameras to z-depth values, which is defined as the distance of a point to the sensor plane.

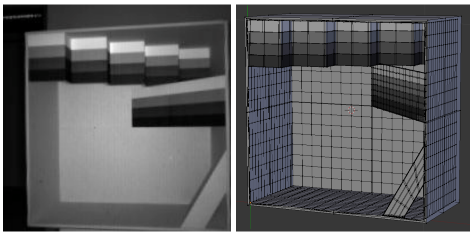

Figure 4: ***Left:*** *intensity image of target* ***Box*** *taken with real ToF camera.* ***Right:*** *polygon mesh of target box.*

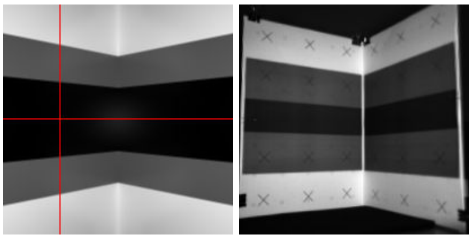

Figure 5: ***Left:*** *ToF Intensity Image of synthetic* ***Corner*** *scene (depth profiles along the red lines are shown in Figures 8 and 9,* ***Right:*** *Intensity image of* ***RealCorner***

We did not consider additional light sources in the scenes as their non-modulated light contributions can be rendered separately and added as constant offset to all phase images.

5.1. Basic Evaluation

To test the basic properties of the algorithm mostly the synthetic **Corner** scene was used. Multiple reflections of the light between both walls causes the planes to appear curved in the top down view (Figure 8). When the path tracing is interrupted after the first intersection (green dotted line in Figure 8) no interreflection takes place and the resulting depth corresponds exactly to the ground truth depth. For maximum light and eye path lengths of two (blue solid line) multireflection takes place, but also image artifacts occur in the corner. In this case the depth variance in the center over multiple realizations of the experiment reach a maximum (see Figure 7). For higher recursion depths (solid red line) these artifacts disappear again. Recursion depths higher than 8 would only be necessary in complex scenes where the illumination is purely indirect and increasing this value has little to no effect in typical ToF setups.

As error metric we computed the pixelwise difference between the computed and ground truth depth images and calculated the standard deviation over the image, save for bor-

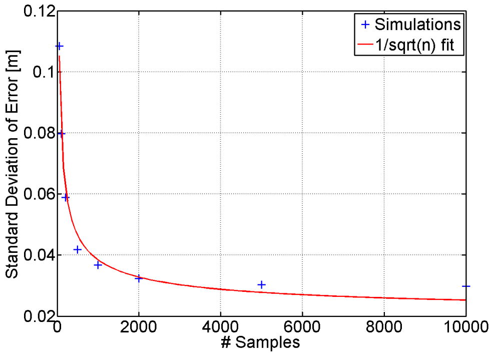

Figure 6: *Depth noise wrt number of pixel samples. The error follows a $\frac{1}{\sqrt{N}}$ curve as expected for a monte-carlo method.*

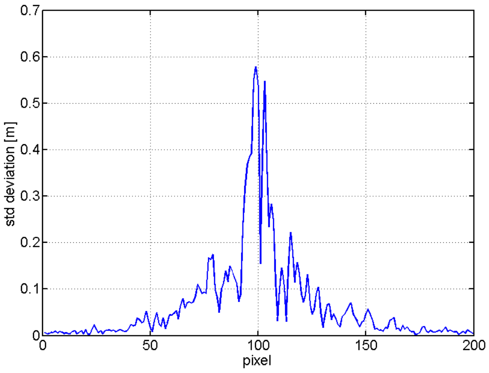

Figure 7: *Depth variance for simulation of **Corner** scene with recursion depth 2. Rendering artifacts in the center suggest that this depth is insufficient.*

der pixels if known. As expected from a Monte-Carlo algorithm, the error decreases with $\frac{1}{\sqrt{N}}$ as the number of samples N increases (see Figure 6). The error however does not approach zero but the fixed offset imposed by the multipath effect.

We also investigated the effect the scene geometry has on the multipath artifacts. For that matter we changed the corners opening angle in steps in between 110 and 50 degree and rerun the simulation each time. In Figure 10 we plotted the true corner form and the simulated form while the deviation between both values is displayed in Figure11. This was done by fitting a line to each side of the simulated form and calculating the angle between them. For a true opening angle of 80 the simulated form deviates most strongly from the ground truth one as in this case the angles between the light, camera and centers of the walls fulfill the reflection law almost perfectly. As these effects can only be explained by light/scene interactions we conclude that our algorithm does indeed simulate multipath effects correctly.

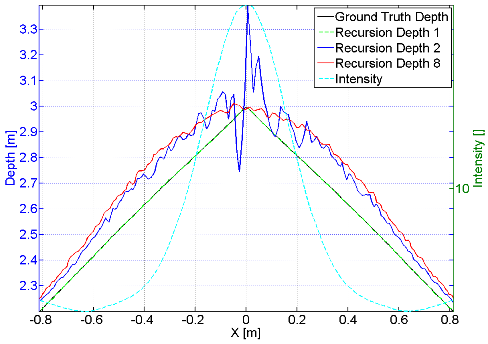

Figure 8: *Horizontal depth profiles along the lines in Figure 5 for different recursion depths. For recursion depth one, no multipath effects occur, recursion depths of two show strong render artifacts in the center.*

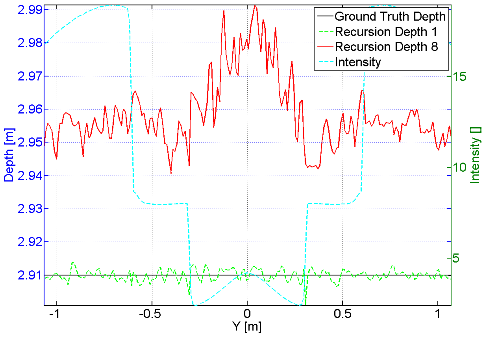

Figure 9: *Vertical depth profiles along the lines in Figure 5 for different recursion depths, The measured depth changes with the observed intensity. Multipath effects are strongest in the center as the relative contribution of direct light is weaker.*

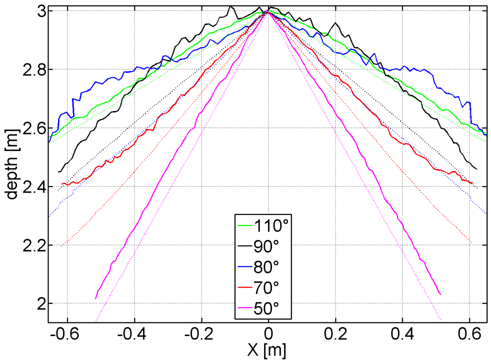

Figure 10: *Horizontal depth profile for different corner angles (solid lines: simulated, dotted lines: ground truth). Multipath effects are most distinct for an* 80° *opening angle.*

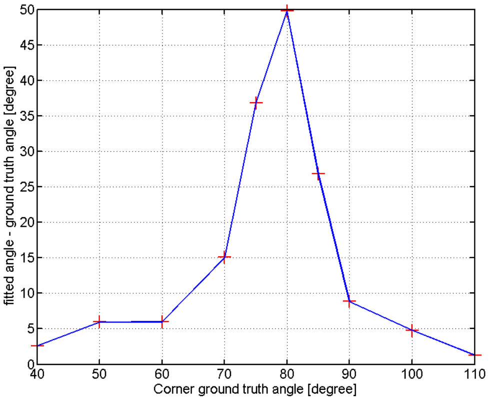

Figure 11: *Ground truth corner angle vs. difference between ground truth and fitted line. At* 80° *the reflection of the light source appears in the middle of the side walls which increases the distortions caused by multipath effects.*

5.2. Influence of Material Properties

Apart from regular reflection and multireflection, different materials can have different brightness levels in the resulting ToF images. ToF cameras are known to exhibit a intensity dependent depth offset which is mainly caused by the non-linear photo response function of the sensor ([Sch11]). Examples can be seen in the vertical depth profile of the **Corner** scene (Figure 9) and the slope in the **Box** scene (region A in Figure 12). The vertical slice in the former shows a higher measured depth in the center where dark material was used while the darker part of the slope in the latter rises at a more shallow angle. In these regions the light which was reflected from brighter regions and has taken a longer path has a higher relative contribution over the direct light.

Both our method as well as Schmidt's method reproduce this effect to a certain degree, although based on different physical principles. For Schmidt's simulator the offset is mostly caused by simulating the non-linear pixel response curves, which changes the relative intensity differences between the raw phase images. In our method multipath effects are more distinct in dark areas as here the relative intensity contribution between direct and indirect light is different than for brighter materials. Additional material based effects are for example the bulge in region C of Figure 12. This is probably caused by strong interreflection at the grazing angle and could not be reproduced reliably by any method.

The **Kitchen** scene shows examples of more effects present in more complex scene setups and real life conditions. Clearly visible are artifacts caused by reflections or transparent materials such as the glass cups or the chrome surfaces on the oven. The cooker hood for example does reflect parts of the back wall and the side of the white cupboard. The cupboard side is not illuminated directly by the ToF illumination and is therefore darker (See also raw phase images in Figure 2).

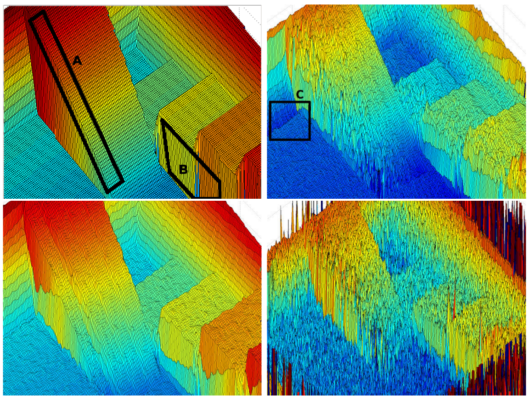

Figure 12: *Surface plots of target box.* ***Top left:*** *Ground truth;* ***Top right:*** *real ToF output;* ***Bottom left:*** *our simulation;* ***Bottom right:*** *simulation by Schmidt. Visible are the intensity dependent slop (A), flying pixel (B) and the bulge in the real image caused by material properties (C).*

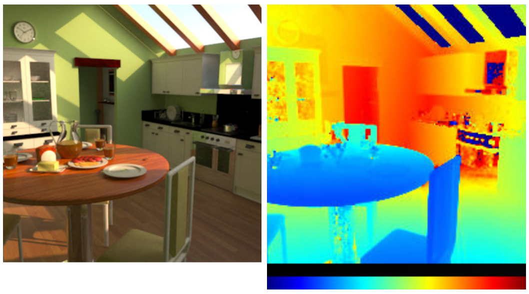

Figure 13: ***Left:*** *Photorealistic rendering of scene.* ***Right:*** *Simulated ToF depth map. Depth range from 0 m(blue) to 7.5 m(red). Materials like glass or chrome create artifacts in ToF depth images. Examples are the oven door or the glass cabinet.*

5.3. Comparison and combination of different simulation methods

Multiple global illumination methods have been described in computer graphics and many of them could be modified in the same manner as path tracing to allow ToF simulations. For example we investigated whether the photonmapping algorithm by Jensen [Jen96] can be used to simulate multipath effects. An example depth profile of a corner scene obtained with Photonmapping can be seen in Figure 14. The difference between ground truth and simulated depth is much smaller for pure diffuse illumination (magenta line) than for direct and specular components (red line). Photonmapping handles diffuse indirect illumination differently than specular illumination (which is treated in a similar way as our path tracing approach). As the former is less important in typical ToF scenarios we consider the photon shooting process as less suited for ToF simulation, although it may be faster under certain conditions as the light distribution must only be computed once.

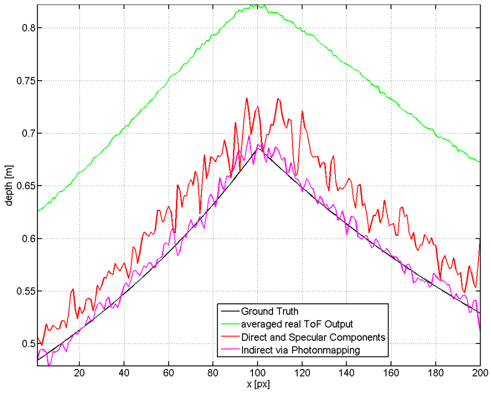

Figure 14: *Horizontal depth profile simulated with photonmapping algorithm instead of Path Tracing. The multipath effect caused by diffuse indirect illumination (magenta) is much smaller than the effect caused by specular reflection (red).*

To compare our method to existing ToF simulators we primarily used the **RealCorner** and **Box** scenes (Figures 15 and 4). For the visualization we shifted the depth values to their common mean. This is justified as all ToF cameras and simulation methods introduce a different bias into the data, however these can be nullified by known depth calibration techniques.

The method of Schmidt et al. can simulate depth offsets caused by different intensities while Keller's method is realtime capable due to a sophisticated GPU implementation (the bright dot is caused by partial overexposure). We were also able to combine methods by using the depth maps of our algorithm as input for Schmidt's method. The result is a simulation whit included multipath artifacts combined with intensity based offsets and a physical noise model.

To perform a statistic comparison of the algorithm results we first calculated the error images between the ground truth and the real ToF output as well as the difference between ground truth and each individual simulation. We then computed the spearman correlation between each pair of error images (Table 1). The closer the correlation between the real ToF output and a given simulation method, the higher the realism of the method. Our method (bold) is slightly worse than Schmidt's method (green), mainly because the later one simulates the intensity based offset more accurately. Combining both methods (blue) significantly improves the results as now nearly all existing physical effects are handled.

6. Conclusion

Based on a modification of the bidirectional path tracing algorithm, in this paper we solve the problem of physically correct multipath interference for phase modulated time-of-flight camera simulation. We have shown that the method can correctly simulate flying pixels, depth errors due to multiple interreflections (multipath) as well as transparent or

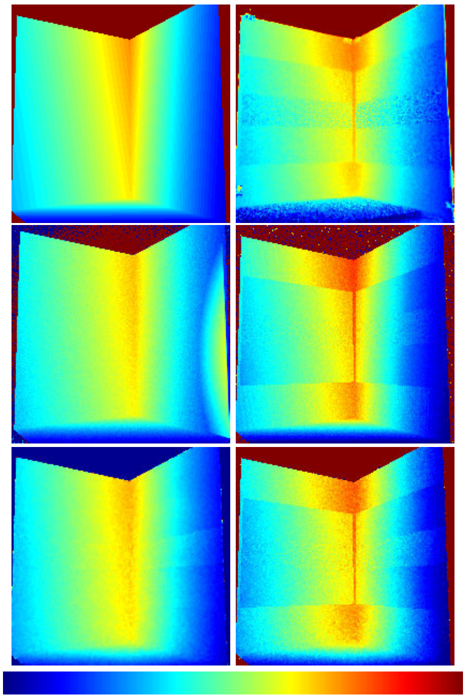

Figure 15: *Colorcoded Depth Maps for **RealCorner** scene, **top left:** ground truth, **top right:** real ToF camera, **middle left:** Keller's method, **middle right:** Schmidt's method, **bottom left:** our method, **bottom right:** combination of our and Schmidt's method. All depth values were shifted to the same mean depth: 0.5 m (blue) to 0.95 m (red).*

	Real	Schmidt	Keller	Our	Combi
Real	1.00	0.25	−0.20	**0.19**	**0.42**
Schmidt	0.25	1.00	−0.78	−0.38	0.67
Keller	−0.20	−0.78	1.00	0.38	-0.61
Our	**0.19**	−0.38	0.38	1.00	0.09
Combi	**0.42**	0.67	-0.61	0.09	1.00

Table 1: *Spearman correlation matrix for real time-of-flight data, Schmidt's method, Keller's method, our simulation and the combi-method. Higher values indicate a simulation result closer to the real camera output.*

strongly reflecting materials based on the scene geometry. Furthermore, it is able to simulate time-of-flight depth distortions caused by different materials in the scene. Our simulation is so far the only one which accounts for all these effects, although it does not yet address all problems like intensity dependent offsets or wiggling errors. However we have shown that our simulation method can be combined with other sophisticated sensor simulation methods to cover a wider field of ToF related problems. By taking into consideration the physical and geometric setup of a scene we can create more realistic images and depth maps for statistical analysis, evaluation, denoising and test of time-of-flight cameras. This enables researchers and practitioners to create synthetic test datasets for various environmental conditions, scenes or resolutions without the need of a real camera setup.

Future work will concentrate on optimizing the computational efficiency of the rendering process. An option would be to create all phase images simultaneously instead of using four distinct renderings at the cost of simulating motion artifacts. Additionally, more complex material shaders could simulate the physically correct behavior under infrared illumination or effects such as the bulge in Fig. 12 region C.

The modified LuxRender code will be made available on our website (http://hci.iwr.uni-heidelberg.de/Benchmarks/) according to the GNU General Public License.

References

[Bra00] BRADSKI G.: The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000). 4

[Cen] CENOBI, BLENDSWAP COMMUNITY: Kitchen scene. Distributed under the Creative Commons Attribution-ShareAlike 3.0 license http://www.blendswap.com/blends/view/62683. 4

[DGC*11] DORRINGTON A., GODBAZ J., CREE M., PAYNE A., STREETER L.: Separating true range measurements from multipath and scattering interference in commercial range cameras. In *IS&T/SPIE Electronic Imaging* (2011), International Society for Optics and Photonics. 2

[EJ09] ERZ M., JÄHNE B.: Radiometric and spectrometric calibrations, and distance noise measurement of ToF cameras. *Dynamic 3D Imaging* (2009), 28–41. 3

[FB08] FALIE D., BUZULOIU V.: Distance errors correction for the time of flight (ToF) cameras. In *Imaging Systems and Techniques, 2008. IST 2008. IEEE International Workshop on* (2008), IEEE, pp. 123–126. 2

[Fuc10] FUCHS S.: Multipath interference compensation in time-of-flight camera images. In *Pattern Recognition (ICPR), 2010 20th International Conference on* (2010), IEEE. 2

[Gor82] GORDON H. R.: Interpretation of airborne oceanic lidar: effects of multiple scattering. *Applied Optics 21*, 16 (1982), 2996–3001. 2

[ICG86] IMMEL D., COHEN M., GREENBERG D.: A radiosity method for non-diffuse environments. In *ACM SIGGRAPH Computer Graphics* (1986), vol. 20, ACM, pp. 133–142. 3

[Jen96] JENSEN H.: Global illumination using photon maps. *Rendering Techniques 96* (1996), 21–30. 6

[JPMP12] JIMÉNEZ D., PIZARRO D., MAZO M., PALAZUELOS S.: Modelling and correction of multipath interference in time of flight cameras. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (2012), IEEE. 2

[Kaj86] KAJIYA J.: The rendering equation. *ACM SIGGRAPH Computer Graphics 20*, 4 (1986), 143–150. 3

[KGP12] KAREL W., GHUFFAR S., PFEIFER N.: Modelling and Compensating Internal Light Scattering in Time of Flight Range Cameras. *The Photogrammetric Record 27*, 138 (2012). 2

[KK09] KELLER M., KOLB A.: Real-time simulation of time-of-flight sensors. *Simulation Modelling Practice and Theory 17*, 5 (2009), 967–978. 2, 4

[KOKP07] KELLER M., ORTHMANN J., KOLB A., PETERS V.: A simulation framework for time-of-flight sensors. In *Signals, Circuits and Systems, 2007. ISSCS 2007. International Symposium on* (2007), vol. 1, IEEE, pp. 1–4. 2

[KW76] KUNKEL K. E., WEINMAN J.: Monte carlo analysis of multiply scattered lidar returns. *Journal of Atmospheric Sciences 33* (1976), 1772–1781. 2

[L] LuxRender: GPL Physically Based Renderer. http://www.luxrender.net. URL: http://www.luxrender.net. 3

[LK07] LINDNER M., KOLB A.: Calibration of the intensity-related distance error of the PMD ToF-camera. In *Proc. SPIE, Intelligent Robots and Computer Vision* (2007), vol. 6764, p. 67640W. 2

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. In *Proceedings of CompuGraphics* (1993), vol. 93, pp. 145–153. 3

[MK11] MEISTER S., KONDERMANN D.: Real versus realistically rendered scenes for optical flow evaluation. In *Electronic Media Technology (CEMT), 2011 14th ITG Conference on* (2011), IEEE, pp. 1–6. 1

[PH10] PHARR M., HUMPHREYS G.: *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2010. 3

[Pla06] PLAUE M.: *Analysis of the PMD Imaging System*. Tech. rep., Interdisciplinary Center for Scientific Computing,University of Heidelberg, 2006. 2

[RFHJ08] RAPP H., FRANK M., HAMPRECHT F., JAHNE B.: A theoretical and experimental investigation of the systematic errors and statistical uncertainties of Time-Of-Flight-cameras. *International Journal of Intelligent Systems Technologies and Applications 5*, 3 (2008), 402–413. 3

[Sch11] SCHMIDT M.: *Analysis, Modeling and Dynamic Optimization of 3D Time-of-Flight Imaging Systems*. PhD thesis, University of Heidelberg, 20011. 2, 4, 6

[Sch08] SCHMIDT M.: *Spatiotemporal Analysis of Range Imagery*. PhD thesis, University of Heidelberg, 2008. 2

[SSD07] SMITH A., SKORUPSKI J., DAVIS J.: *Transient rendering*. Tech. rep., School of Engineering, University of California, 2007. 2

[VG95] VEACH E., GUIBAS L.: Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*. Springer, 1995, pp. 145–167. 3

[WBSB12] WULFF J., BUTLER D. J., STANLEY G. B., BLACK M. J.: Lessons and insights from creating a synthetic optical flow benchmark. In *ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation* (oct 2012), A. Fusiello et al. (Eds.), (Ed.), Part II, LNCS 7584, Springer-Verlag, pp. 168–177. 1