# Sensor Simulation and Evaluation for Infrastructure-Free Mobile Sensor Carrier Platforms

Artur Schütz(✉) and Maik Groneberg

Fraunhofer Institute for Factory Operation and Automation IFF,
Sandtorstr. 22, 39106 Magdeburg, Germany
{artur.schuetz,maik.groneberg}@iff.fraunhofer.de

**Abstract.** In this paper, we propose a highly accurate way to simulate Time-of-Flight sensors and their explicit characteristics, those enabling sensor evaluations for critical applications, like measurement scenarios, human-machine-collaboration and autonomous driving, as well as a way for estimation of the best-suited sensor for a specific use case. A. Schütz did the development of the simulation, while M. Groneberg was focused on the application of such simulation to logistics. Time-of-Flight Sensors are able to provide robust depth images of dynamic scenes at high framerates, which makes them interesting for movement analysis and logistics use cases and makes them advantageous for autonomous mobile robot navigation. Yet the depth images are disrupted by a number of systematic errors, which, to our knowledge, have not been fully simulated so far. These errors has to be taken into account in early development stages to avoid the destruction of hardware or injuring humans working with autonomous robots. We believe that simulations are suited to develop and to test algorithms for navigation under different environmental conditions to avoid errors, which are caused by misleading depth images. This paper examines the systematic errors of three different Time-of-Flight sensors and presents a physically motivated approach to simulate those sensors. We were able to simulate most of the systematic errors, which were observed in depth images of time-of-flight sensors by using a path tracing algorithm to calculate the influence of light propagation in a physically accurate manner.

**Keywords:** Mobile sensor platform · Infrastructure-free · Time-of-flight sensor · Lidar sensor · Global illumination · Path tracing · AMR · Drone · Movement analysis

## 1 Variety of Sensor Platform Types

We distinguish sensor platforms for scanning applications into four different categories, the infrastructure related/independent scan and manual/self-adapting scan. The characteristics on sensor platforms can met the requirements of multiple categories, so we can have manual or self-adapting infrastructure related or infrastructure independent scanning platforms, like shown in Fig. 1. Stationary platforms are infrastructure related, because the sensors need to be equipped at an existing infrastructure or the


M. Freitag et al. (Eds.): LDIC 2020, LNLO, pp. 101–113, 2020.
https://doi.org/10.1007/978-3-030-44783-0_10

infrastructure needs to be integrated. While the infrastructure is fixed, there is the need of manual adaption for moving the scanning system itself or the object, which needs to be measured, into the scanning area. We developed a stationary system for the verification of freight dimensions, called ScanSpecter. It was developed to measure the dimensions of an undefined freight loaded on a pallet, while a fork lifter crosses the scanning area. Another infrastructure related platform is the telescope scan, consisting of a mobile platform attached with a telescoping arm and various sensors used for positioning, environmental recognition and measurement as well as identification tasks. Infrastructure independent platforms can be used in undefined environments for a wider range of measuring tasks and a variety of objects. An example, for the category manual adoption, is the handheld scan. With its small size and dynamical use, it is well suited for measurement in dynamic and/or tight environments and for objects which can't be moved. Due to the movement of the user hard- and software image stabilization and exact positioning is unavoidable. Ultimately, the multi-platform scan enables autonomous scanning of undefined, large and moving objects. It shares the same requirements like the two previous forms, but with the addition of high security demands. The last three types can be summed up as mobile sensor carrier platforms. For now stationary sensor platforms are in common use, but with further development and improvement of sensor technologies, the mobile platforms become more important. In fact, of the growth of autonomous mobile systems like AMR's (autonomous mobile robots) and drones, one point has a special relevance in their usage. The safety issues are one of the last barriers that need to be crossed, until they find their place in our everyday lives. That's why our focus in this paper is fully on mobile sensor platforms and the collaboration with each other as well as with human beings and how we can enhance

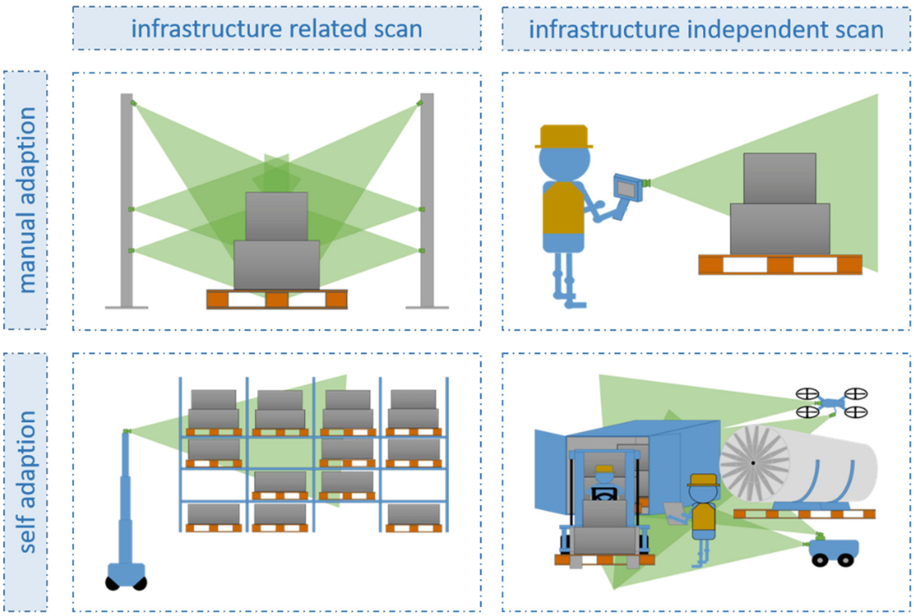



**Fig. 1.** Sensor platform classification for logistic measurement use cases [13]

safety by improved sensor data quality. All these systems place high demands on sensors, so it's absolutely necessary to verify a sensors suitability for a specific use case. While real world tests with hardware are often quite expensive, not reliable and comparable, due to environmental conditions, other solutions need to be developed to overcome these problems.

## 2 Benefits and Applications of Mobile Infrastructure-Free Sensor Platforms

The use of autonomous vehicles on our roads is currently restricted and obstructed by laws and regulations of individual countries. Until nowadays they have found their way exclusively into intralogistics, where they are mainly used for the transport of cargo and materials. Autonomous mobile robots differ from their predecessors, the automatic guided vehicles (AGV's), mainly in the way they navigate. Instead of just following predefined routes using a line-following algorithm, these robots perceive their surroundings in real time and can react to changing environmental conditions using intelligent sensors and algorithms to evade other vehicles, objects or even people by determining a new route. This greatly increases the safety and flexibility of the system. The collaboration between humans and mobile platforms can be distinguished into noncontact support and tactile steering. Noncontact support systems perform assistance tasks without an external control via a human being, while systems with a tactile steering are partly guided by tactile means, Fig. 2. We developed a tactile grip using force sensing resistors to enable such a functionality. A variety of tasks in logistics as well as in everyday life can accomplished, such as restocking good in a supermarket or warehouse and post parcels that automatically follow mail carrier. Another important application is the measurement of freight and the motion analysis of workers. This

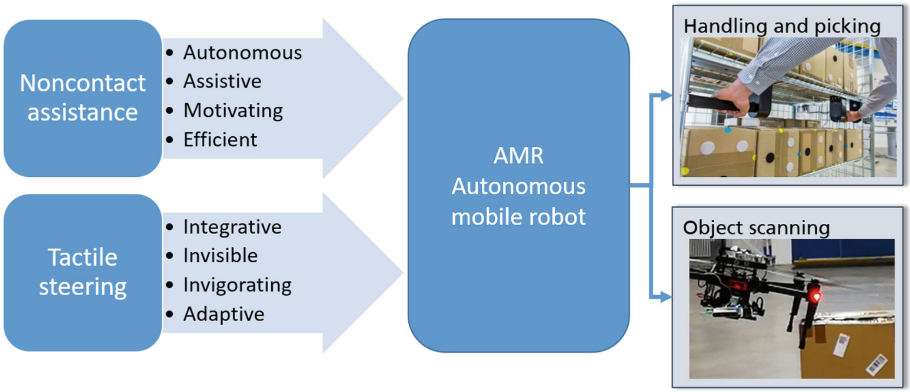



**Fig. 2.** Autonomous mobile robots [13]

ensures safe working as well as optimal storage and transport of goods.

## 3 Requirements in Human-Machine and Machine-Machine Collaboration

In order to fathom the limits of technology in multi-platform scanning systems, we changed the field of application to movement analysis in sports. Recreational and competitive sports demand high standards from analytical systems. High frame rates during data acquisition and low latencies during data processing are essential for a reliable analysis. The use of mobile systems is unavoidable due to the changing competition venues and the movement of athletes. "FAST – Fast Actuators, Sensors and Transceivers" is together with nine other project clusters part of the Zwanzig20 cluster program. They focus in times of upcoming 5G technologies on overall system latency in the millisecond range [14]. Within the project Fast Athletics, the movement of rowers and cross-country skiers are investigated by means of a mobile multimodal sensor system. Figure 3 shows the expired system setup at the different use cases.

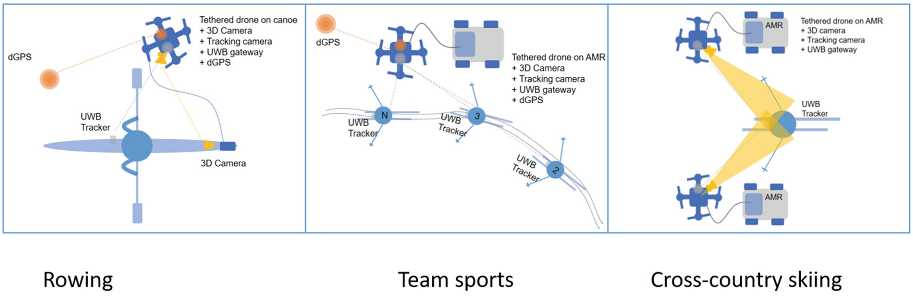



**Fig. 3.** Different measurement scenarios for rowing and cross-country skiing [13]

The proximity of AMR's and drones, as well as humans poses enormous challenges on technology to guarantee their safety. Therefore it is inevitable to carry out detailed simulations of the sensors and the entire system. Time-of-Flight (ToF) sensors, due to their high range and accuracy, can help to detect objects in time and accurately and enable safe control of autonomous vehicles. The analysis of the systematic and random errors, as well as the correction possibilities, must be analyzed in detail before the first real operation. Therefore, we developed a realistic, use case and environment independent simulation environment for ToF-sensors. The results can be used for sensor evaluation in terms of fitness and their errors to minimize the effect of them.

## 4 Sensor Analysis

Sensor simulations are important for hardware design and application development. Therefore, the simulation must be as accurate as possible and have to capture the major sensor characteristics to ensure similar results for real scenarios. As pointed out by Riestock et al. [15] depth sensors are an essential building block for applications which address the challenges of highly efficient transport and storage in logistics. Depth Sensors are used to measure freight sizes and movements to support loading processes and to observe storage usage. Autonomous robots which transport storage racks in warehouses, using sensors to avoid collisions, and robotic arms are already used in productive logistic systems. Using simulations allows an early access to a broad variety of data which can be used for development and a prototypical testing of applications in a virtual environment. Simulations for this purpose has already been done in literature for different types of depth sensors. However, the usage of Time-of-flight sensors introduces entirely new challenges in developing applications and simulations. Keller [16] developed a real-time simulation of time-of-flight sensors as a part of his PhD thesis. He used the OpenGL Pipeline to emulate common errors of time-of-flight sensors by augmenting a depth map, which was rendered as a part of the rendering process of the 3D scene, with common errors of time-of-flight sensors. Even though his approach can be simulated in real-time, Keller is not addressing errors which are caused by light emission and propagation. Meister et al. [2, 9] concentrate on errors which are introduced by light propagation by simulating time-of-flight sensors using physically inspired global illumination approaches. Meister et al. where able to simulate multipath errors of for the first time by using photorealistic rendering techniques. On the other hand, the simulation is done on CPU which results long rendering times. Also, they focused on errors caused by light propagation and exclusively simulates multiple path errors which are only one portion caused by time-of-flight sensors. Lambers et al. [10] concentrate on simulating time-of-flight sensors to evaluate chip layout variants. Therefore, most of errors caused by light emitting and propagation are not addressed by their approach. Our approach is able to simulate depth sensors almost in real-time while taking lens distortion and errors caused by light emission and propagation into account by using General Purpose Computation on Graphics Processing Units (GPGPU) and parallelizing the calculation on a large number of threads.

Time-of-flight sensors measure the distance to objects in a scene by emitting infrared light (IR) or near infrared light (NIR) light pulses, which are limited to a narrow frequency range. A distinction is made between pulsed and continuous wave modulation systems, where the difference can be found in the way of measurement. The Pulsed-Modulation measures the delay of the reflected light to determine the distances. In contrast, the continuous wave modulation estimates the traveled distance by estimating the phase shift between the wave function of the emitted and the reflected light [1].

In the following, time-of-flight sensors of the different operating modes are examined for possible errors, so that they can be taken into account in the simulation. The investigation focused on the industrial camera IFM O3D303, the Microsoft Kinect v2 and the Asus Xtion 2. The experimental results indicate that the Xtion 2 uses pulsed modulation whereas the other two systems use continuous wave modulation [3].

## 4.1 Time-of-Flight Sensor Errors

All observed sensors are affected by disruptive factors that cause deviations of the depth values from reality. We investigated several phenomena, which vary in strength and characteristics depending on the camera used. These disturbing factors can be divided into effects caused by the generation of the light pulse, optical issues during light propagation, as well as side effects caused by the sensor chip.

**Error Due to Emission of Modulated Light.** These relate to the operating temperature of the sensor, as well as the used LEDs and selected wave function. The basis of this investigation was a cold start of the sensors along with an orthogonal alignment of the optical axis to a planar surface. The pixel in the center of the image served as a reference object for the analysis. The corresponding depth value was observed for 13 min. Figure 4 illustrates the change in depth values according to the operation time. This effect can be reduced by adding active or passive cooling. Under steady ambient temperatures, it is also possible to correct the error mathematically, because the offset is constant and remains stable after about 15 min [4].

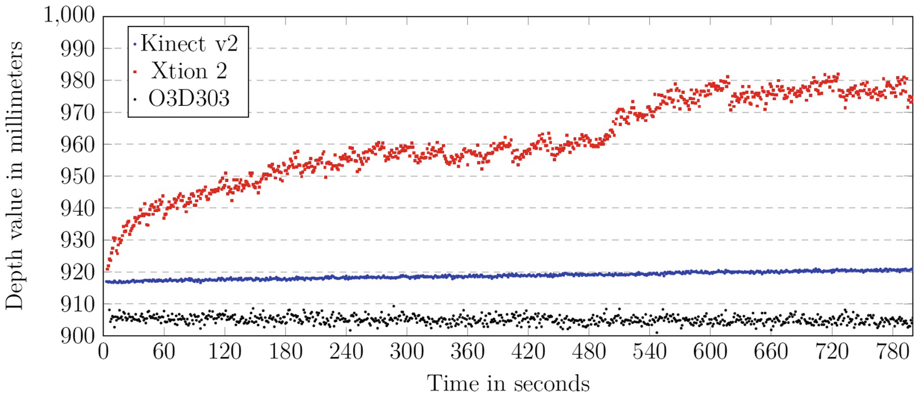



**Fig. 4.** Variation of the measured depth values of all sensors in millimeters over time

The deviates from a perfect sine function of the continuous wave modulation results in a distance-dependent systematic error affecting the depth values [4]. Often a square wave function is used, because it can be modeled more easily and accurately in digital systems. This leads to a sinusoidal systematic error. Figure 5 shows the systematic error of Kinect v2 in relation to distance.

The displayed error consists of a sum of superimposed sinusoidal functions, since three wave functions of different frequencies are used to calculate the depth value [4]. Using continuous wave modulation can also lead to incorrect estimates of the depth values if the maximum measurable distance is exceeded. Consequently, objects with a higher distance are interpreted as if they were closer [3].

**Optical Effects During the Light Propagation.** The LED that emits the modulated light is not exactly placed in the aperture, causing shadows in the image and artifacts in the depth image. Smooth and dark surfaces can also affect the depth values, as they enhance indirect reflections [2]. The light is directed through a lens system and an aperture to the sensor, resulting in scattering in the lens system and diffraction effects at the aperture.

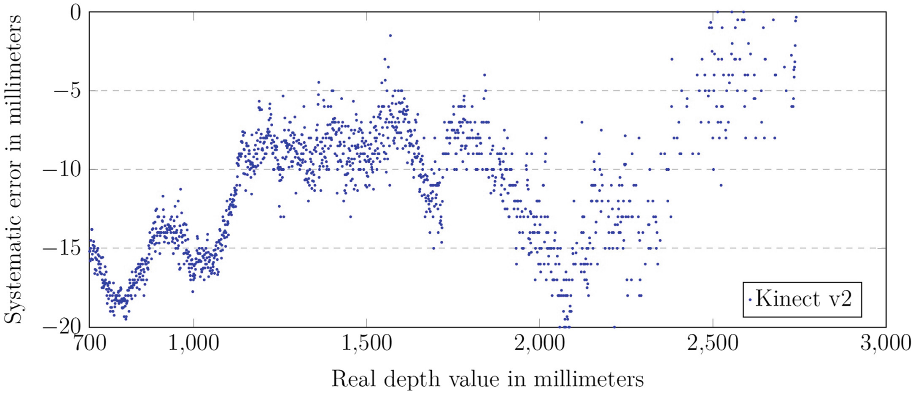



**Fig. 5.** Systematic error of Kinect v2 measuring depth values in relation to the actual distance

The three camera systems were investigated for the multipath error which is caused by indirect reflections. For this purpose, the cameras were positioned in front of a static corner whose two surfaces were positioned at an angle of 90° to each other. The deviations of the depth values are shown in Fig. 6 by means of a top-down view of the measured values compared to the Ground Truth distance of the middle horizontal section of the image.

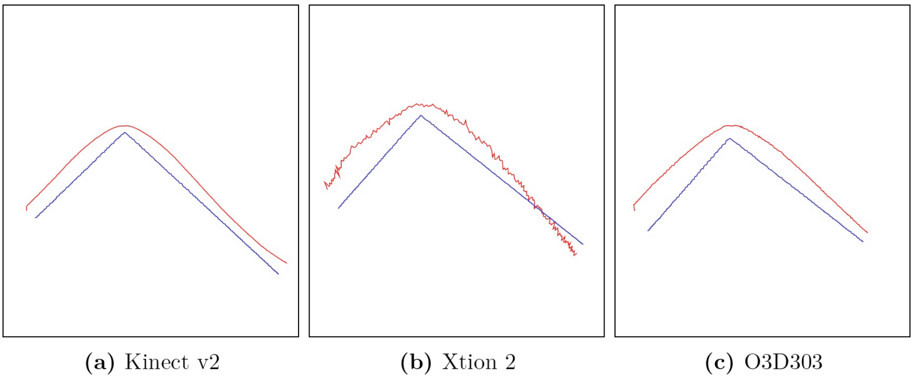
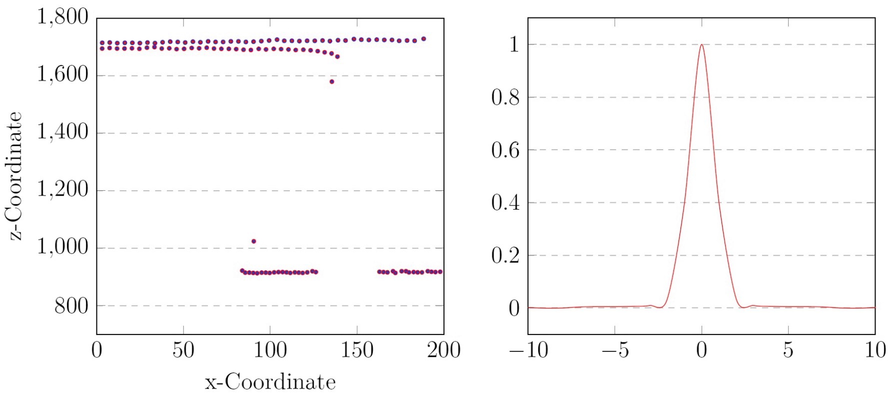



**Fig. 6.** Top-down view of depth value deviations

The experiment showed that at the majority of the points the cameras measured depth values that were too high, caused by the indirect reflections between the walls, because the light travelled a longer path back to the sensor. Lens scattering is an error which is already described in detail in several literatures. This error is caused by the fact that light incident on one point has an influence on the surrounding points. In addition to the diffraction of the light in the aperture, reflections and scattering within the lens system influence the neighboring pixels [5–8]. These effects are illustrated in Fig. 7(a). Intensely illuminated objects in the foreground influence the measured depth values of objects in the background.




(a) Depth value

(a) Point spread function of Microsoft Kinect v2

**Fig. 7.** (a) Top-down view of the depth values from the middle horizontal section of the image and (b) lens scattering error depending on the intensity of the infrared image

Fig. 7(b) shows the determined point spread function used in the simulation. On top of that, it was found that the flying pixel error is mainly caused by the lens scattering error [3].

**Light Detection via Sensor Chip.** Depending on the selected chip architecture, the detection of the incoming light also leads to errors that must be taken into account. The sensor chip can cause a random error depending on the measured intensity of infrared radiation. More distant and less illuminated objects might have a worser signal-to-noise ratio. The non-linearity in the integration of incoming light also impairs in particular the depth values of weakly reflecting surfaces [5, 9]. A movement of objects or the camera can, depending on the chip architecture, result in artefacts within the image [10]. Sensors operating with pulsed modulation do not show any systematic errors depending on the distance, but the depth values are influenced by ambient light [3]. On poorly illuminated surfaces, the depth values of the Asus Xtion 2 are usually underestimated. Figure 8 shows, analogous to Fig. 5, the error in the depth value depending on distance. In contrast to Kinect v2 and O3D303, this is not a sinusoidal error, but a steadily increasing error that amplifies with increasing distance.

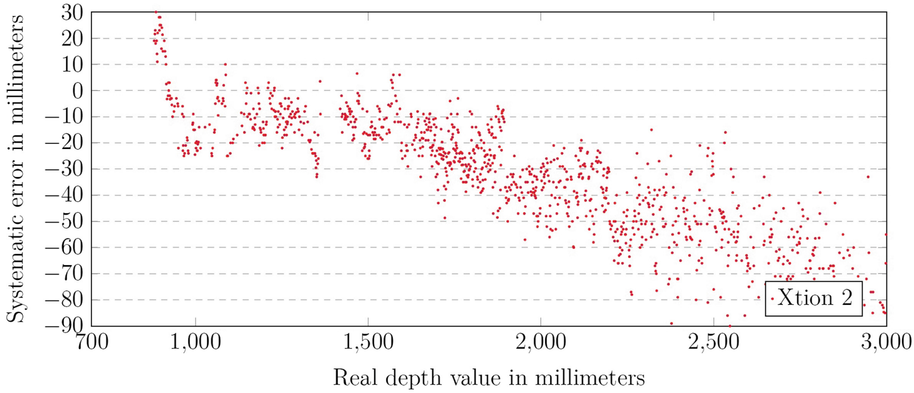



**Fig. 8.** Error of ASUS Xtion 2 depending on the distance

## 5 Simulation of Time-of-Flight Sensors

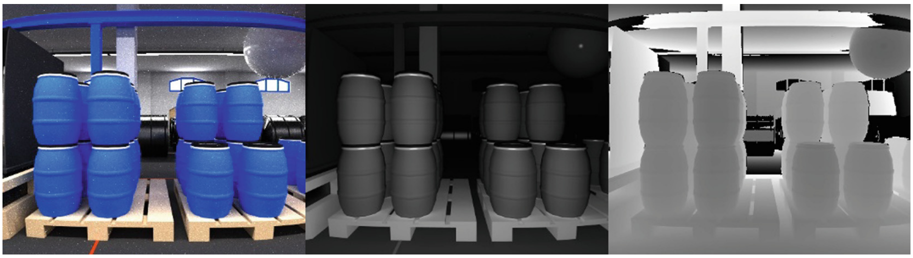
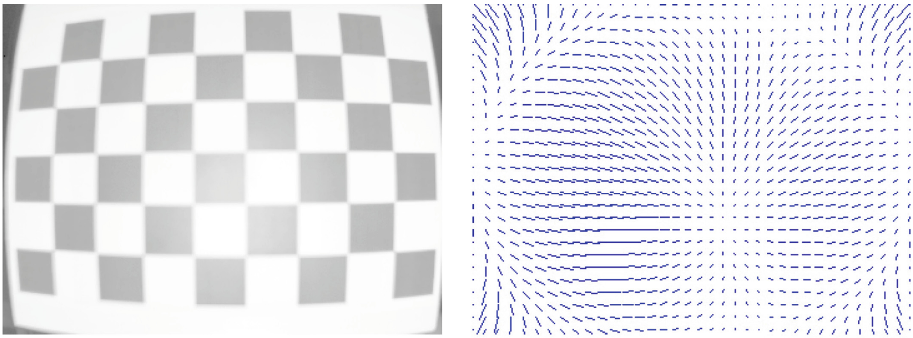


**Fig. 9.** Screenshots of the simulation environment with a color, infrared and depth image

A unidirectional path tracing algorithm using the NVIDIA OptiX Ray Tracing Engine was implemented to simulate the artifacts. The algorithm is an implementation after Kajiya [11], which was extended by a temporal component, in order to be able to consider the flight time of the light. For each pixel of the image, light beams are sent into the scene to determine whether the beam collides with an object. If this beam collides, we clarify wether this point is illuminated by light sources. The proportion of radiation reflected in the direction of the observer can now be calculated. For the indirect illumination by other surfaces, a new ray is then emitted in a random direction, which originates at the intersection of the surface with the first ray. Repeatedly it is determined whether this new ray collides with the scene and the resulting reflection toward the origin is calculated. This process is repeated until the maximum defined length of the path is reached or the ray leaves the scene. Figure 9 shows the resulting scene simulated using this procedure.



**(a)** Distorted image of a chessboard pattern **(b)** Result of the intrinsic calibration

**Fig. 10.** Distorted image of a chessboard pattern used for calibration and the result of the intrinsic calibration of the IFM O3D303

The simulation of the lens distortion was performed directly during ray generation. The camera matrix of the intrinsic calibration was used to calculate the inverse projection matrix, shown in Fig. 10. This is used to calculate a vector for each pixel, which is distorted accordingly by using the distortion coefficients. The calculation of the direct and indirect illumination is performed when the ray collides with the scene and the result is summed up for each pixel. The simulation provides four phase images for each frame, which can be used to calculate the depth values.

## 6 Evaluation of Simulation Results

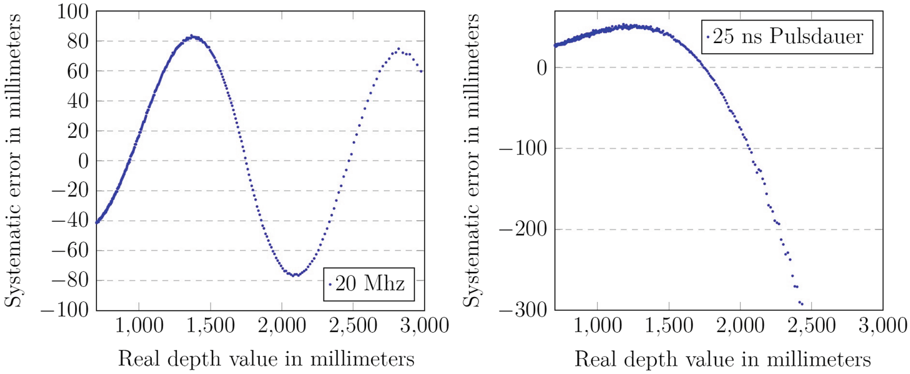
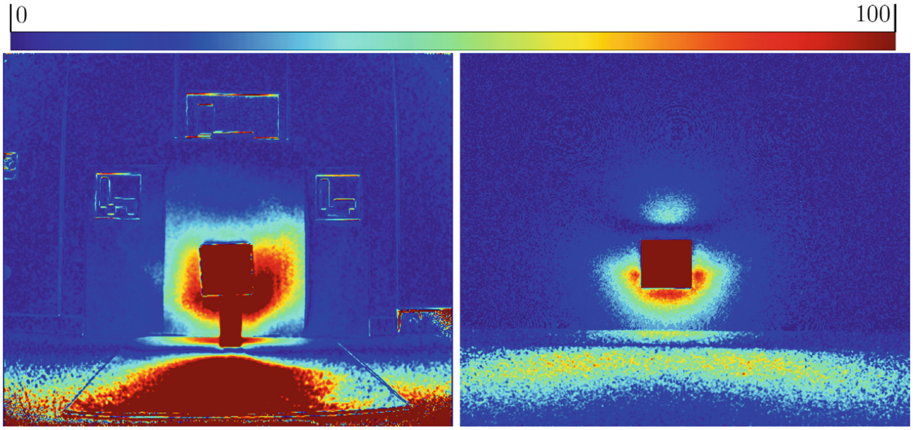



**Fig. 11.** Error in the simulation of the depth values using continuous wave modulation (left) and pulsed modulation (right) under the influence of ambient light relative to the actual distance.

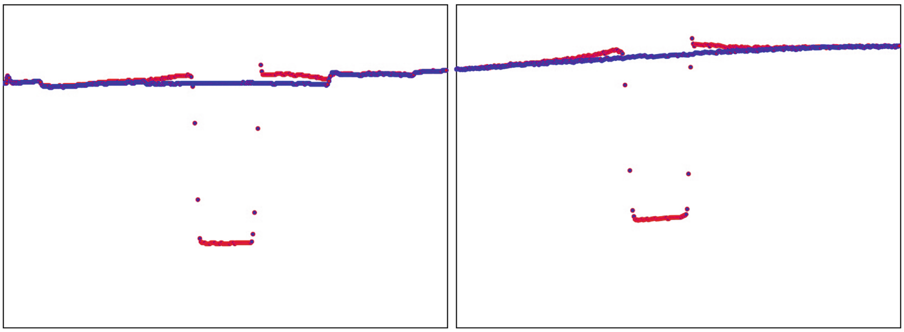
In the developed simulation environment for time-of-flight sensors either the simulation of continuous wave modulation or pulsed modulation can be performed. Both variants were tested under the influence of ambient light simulated by an environment map. Figure 11 shows the simulation error relative to the distance. We determined that the wavelength and amplitude of the systematic error occurring with the continuous wave sensors correlates with the frequency of the emitted square wave functions. Using sinusoidal wave functions, no systematic error was found in the depth values.




(a) Kinect v2 (b) Simulation

**Fig. 12.** Comparison of the errors caused by a Kinect v2 and our simulation

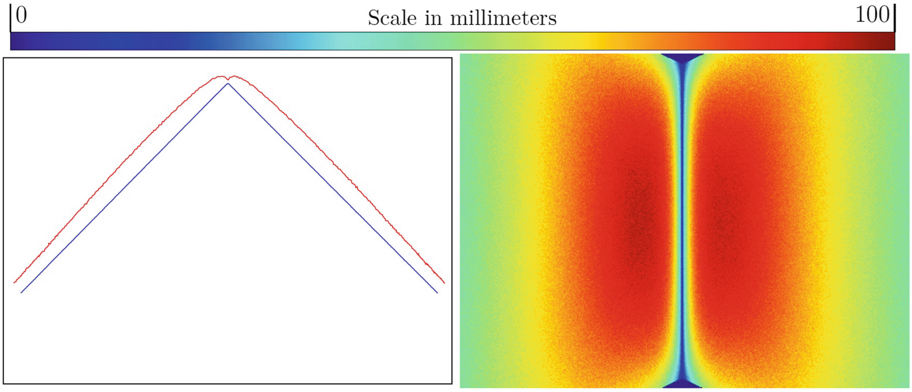
The propagation of light was investigated by placing a box in front of a bright background. This setup was simulated with a similar scenario. Figures 12 and 13 show the results of the comparison in which both the flying pixel error and the effects of lens scattering can be seen. In addition, in the background, to the right of the box, the effects of the shadows resulting from the positioning of the LEDs can be recognized.



(a) Deth values measured by Kinect v2 (b) Simulated depth values

**Fig. 13.** Comparison of the measured depth values obtained from the test setup versus the simulation

Finally, the effects of indirect lighting on depth values were investigated. A similar experimental setup as in the previous analysis was used, though several materials and surface structures were simulated this time. It was found that especially on smooth surfaces indirect reflections occurred, which falsified the depth values. In the simulation, rough surfaces exhibit a slighter multiple path error because they reflect a higher proportion of the light back in the direction of the light source [3, 12].




(a) Horizontal scanline of the depth values

(b) Error distribution

**Fig. 14.** Representation of the simulated depth values with multiple path errors at a path length of 16

Figure 14 shows the simulation of the error caused by indirect reflections. Thereby, it can be seen that the depth values at the edge showed a smaller deviation. No such observations were made in the analyses of the camera systems. Therefore, it is assumed that the simulation introduced this error. The cause of this error could be the use of a minimum distance between two ray intersections, which could lead to inaccurate simulation results at edges.

## 7 Conclusion

We presented a physically based simulation of time-of-flight sensors to support the development of software to correct systematic errors and test measuring algorithm under different environments to avoid unforeseen misleading depth images, which could cause the destruction of hardware, injure humans or just deliver wrong measurement results. Our simulation show a high agreement with measured depth images despite the fact, that not every observed error was simulated correctly yet. From a practical perspective, this enables the user to evaluate measuring and navigation algorithms under different environmental conditions, which speeds up software development and evaluation processes. In this essentially needed to develop mobile sensor platforms in a proper way.

**Acknowledgements.** The authors acknowledge the Federal Ministry of Education and Research Germany for financial support within the framework of the zwanzig20 program. https://de.fast-zwanzig20.de/.

## References

1. Li, L.: Time-of-flight camera - an introduction. Texas Instruments-Technical White Paper (2014)
2. Meister, S.: Photon mapping based simulation of multipath reflection artifacts in time-of-flight sensors (2012)
3. Schütz, A.: Analyse und Simulation von Time-of-Flight Sensoren mittels GPU Path Tracing (2019)
4. Giancola, S., Valenti, M., Sala, R.: A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Technologies. Springer, Heidelberg (2018)
5. Hertzberg, C., Frese, U.: Detailed modeling and calibration of a time-of-flight camera. In: Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics. SCITEPRESS - Science and and Technology Publications (2014)
6. Jamtsho, S., Lichti, D.: Modelling scattering distortion in 3D range camera. Int. Arch. Photogrammetry Rem. Sens. Spat. Inf. Sci. **38**, 299–304 (2010)
7. Mure-Dubois, J.C.C., Hügli, H.: Optimized scattering compensation for time-of-flight camera - Art. no. 67620 h, September 2007
8. Mure-Dubois, J.C.C., Hügli, H.: Time-of-flight imaging of indoor scenes with scattering compensation, January 2009
9. Meister, S., Nair, R., Kondermann, D.: Simulation of time-of-flight sensors using global illumination (2013)
10. Lambers, M., Hoberg, S., Kolb, A.: Simulation of time-of-flight sensors for evaluation of chip layout variants. IEEE Sens. J. **15**(7), 4019–4026 (2015)
11. Kajiya, J.T.: The rendering equation. In: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH) 1986. ACM Press (1986)
12. Oren, M., Nayar, S.K.: Generalization of lamberts reflectance model. In: SIGGRAPH (1994)
13. Groneberg, M., Brauckmann, N., Richter, K.: Infrastructure-Free Sensor Platform for Multidimensional Object and Motion Analysis. RelStat, October 2019
14. Fast. https://de.fast-zwanzig20.de/. Accessed 27 July 2019