

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Real-time simulation of time-of-flight sensors

Maik Keller *, Andreas Kolb

Computer Graphics Group, Institute for Vision and Graphics, University of Siegen, 57076 Siegen, Germany

ARTICLE INFO

Article history:

Received 23 June 2008

Received in revised form 23 December 2008

Accepted 6 March 2009

Available online 19 March 2009

Keywords:

Sensor simulation

Time-of-flight

Photonic Mixing Device

Real-time simulation

GPU programming

ABSTRACT

Today's time-of-flight (TOF) sensors measure full-range distance information by estimating the elapsed time between emission and receiving of active light in real-time. Such sensors are inexpensive, compact, and they have a high performance, which especially fits real-time applications, e.g. in the fields of automotive, robotics, 3D imaging, and visualization. The simulation of such sensors is an essential building block for hardware design and application development. Therefore, the simulation data must capture the major sensor characteristics.

This paper introduces a simulation approach, which is motivated by physics, for the Photonic Mixing Device (PMD) sensor which is a specific type of time-of-flight sensor. Dynamic motion blurring and resolution artifacts such as flying pixels as well as the typical deviation error are prominent effects of real world systems. Flying pixels arise when an area of inhomogeneous depth is covered by a single PMD-pixel whereas the deviation error is based on the anharmonic properties of the optical signal. The modeling of these artifacts is essential for an authentic simulation approach. We present a detailed comparison between a real PMD-device and the simulation data regarding the sensor characteristics.

The proposed algorithms are implemented in a hardware accelerated solution which makes use of the programmability of modern Graphics Processing Units (GPUs). This way, an interactive simulation feedback is provided for applications and further data processing. The simulation takes place in real-time and thus all required control mechanisms are accessible in real-time, too.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

There are several well-known image-based sensor techniques for measuring distances between objects and the sensor location. Examples based on triangulation and/or image correspondences are laser triangulation, structured light and stereo vision [1,2]. Time-of-flight (TOF) sensors, on the other hand, compute the distance to an object by estimating the elapsed time between the emission of light from an active illumination and the arrival at the pixel of a sensor chip. Relatively new camera-like approaches, like the Photonic Mixing Device (PMD) [3,4] and the Z-Cam [5], are still evolving.

A simulation of TOF-sensors is very helpful in numerous of use cases. On the one hand, the sensor parameters such as resolution, focal-length and frequency can easily be modified and distortion effects can be evaluated in detail. On the other hand, experiments can be carried out under reproducible conditions, especially with regard to dynamic scene setups. The simulation approach should not be based on an ideal type of sensor, of course, the simulation results must rather reflect the major sensor characteristics instead in order to produce results comparable to real sensor data. Thus, the sensor simulation is influenced by the current research concerning the sensor model such as the PMD-device. The simulation data effects the development of down-stream algorithms significantly, e.g. in the fields of object recognition and image analysis as well

* Corresponding author. Tel.: +49 271 740 3452.

E-mail address: maik.keller@uni-siegen.de (M. Keller).

as correction of calibration errors. A real-time simulation is indispensable since the algorithms themselves handle the sensor data also in real-time within the data processing pipeline. This means that the sensor simulation needs to be as fast as a real world sensor.

The presented work is based on our Sensor Simulation Framework [6] which can handle TOF-sensors based on the PMD technology in principle. The current paper focuses on a fully equipped PMD-sensor simulation including major temporal and spatial artifacts, and the systematic deviation error. These effects are essential while simulating this kind of sensor (details see in Section 3.2). The capability of real-time simulation is another main aspect—apart from the precision of the simulated data—which is achieved by a parallelized simulation approach using the programmability of modern Graphics Processing Units (GPUs). Thus, the data can be calculated highly parallel which allows for interactive simulation feedback and real-time processing of down-stream algorithms.

The structure of this paper is as follows: in Section 2, an overview of the related work on PMD-sensor simulations is given, followed by a review of the theoretical basis about the PMD technology in Section 3. Section 4 proposes the physical sensor model which is the basis for the simulation approach. In Section 5, a realization of the theoretical sensor model is described. The results of the sensor simulation are presented in Section 6 and their evaluation is discussed in Section 7. Finally, Section 8 concludes the presented sensor simulation approach.

2. Related work

Developers often apply their algorithms on both real sensor data and simulated sensor data. Peters et al. [7] use synthetic test data for the localization of mobile robots. Their TOF-simulator application is MATLAB-based and not suitable for real-time simulation. The approach which they favor has its origin in the measurement of the time-of-flight of the transmitted signal, which results in the final distance to the object. In [8] they extend their application in order to model the bistatic effects which are caused by the spatially separated illuminator and receiver positions in 3D cameras. All these sensor characteristics are also covered by our approach. Furthermore, we use the rendering pipeline of the graphics card which automatically provides the relevant scene geometry with respect to the current view instead of calculating it laboriously manually. Apart from that, our simulation approach is suitable for real-time applications. Streckel et al. [9] use synthetic ideal depth data to test their structure from motion algorithms in connection with TOF-sensors. Their simulation of depth noise is based on results presented by Kuhnert and Stommel [10] who assume a quadric relation between the distance from the sensor to the object and the standard deviation of the PMD-sensor's depth data. The latter approaches neglect sensor characteristics like flying pixels (see Fig. 12) and distance deviation (see Fig. 13). These artifacts are covered by the approach we present in this paper.

3. Photo mixing device technology

3.1. Signal theory

The Photonic Mixing Device (PMD) camera illuminates the scene with modulated, incoherent near infrared (NIR) light (see Fig. 1). Smart pixel sensors [3,4,11] gather the reflected light. The devices are of a low lateral resolution, e.g. current models are available from 64×48 up to 160×120 pixels. One pixel samples and correlates the incoming optical signal with

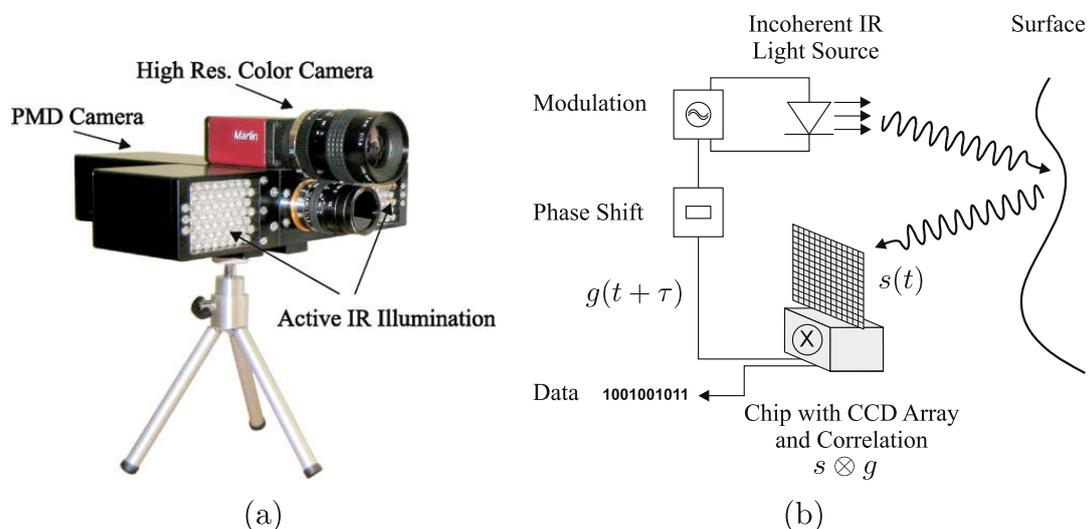


Fig. 1. PMD-device and time-of-flight principle. In (a), a PMD camera (19 k model) with an additional high resolution color camera is shown. The principle of the PMD technology is illustrated in (b).

the internal reference signal of the modulated illumination which results in an image of a per-pixel sampling of the correlation function with an additional internal phase delay. This image is called phase image throughout this paper. The PMD takes four of these phase images to determine the distance related phase shift and thus the distance to the respective object region can be calculated.

This process can be expressed by the following equations: given a reference signal $g(t)$ and the optical signal $s(t)$ incident to a PMD-pixel, the pixel samples the correlation function $c(\tau)$ for a given internal phase delay τ :

$$c(\tau) = s \otimes g = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} s(t) \cdot g(t + \tau) dt. \quad (1)$$

Assuming the reference signal to be sinusoidal, i.e. $g(t) = \cos(\omega t)$ and no additional non-linear signal deformation, the optical response signal is given by $s(t) = b + a \cos(\omega t + \varphi)$, with ω being the modulation frequency, a the amplitude, b the correlation function's bias, and φ the phase shift relating to the object distance. It should be noted that the modulation frequency defines the distance unambiguity for the distance sensing, e.g. $\omega = 20$ MHz results in an unambiguous distance range of about 7.5 m. Basic trigonometric calculations yield:

$$c_\tau(\varphi) = b + \frac{a}{2} \cos(\tau + \varphi). \quad (2)$$

To be more precise, the demodulation of the correlation function is made by using several samples of $c_\tau(\varphi)$ which are obtained by four sequential PMD phase images $A_i = c_{\tau_i}(\varphi)$ which use internal phase delays $\tau_i = i \cdot \frac{\pi}{2}$, $i \in \{0, 1, 2, 3\}$.

The final distance d to the object's surface (see Eq. 4), the intensity values b (see Eq. 5) as well as the amplitude a (see Eq. 6) can be calculated based on the four phase images:

$$\varphi = \arctan \left(\frac{A_3 - A_1}{A_0 - A_2} \right), \quad (3)$$

$$d = \frac{c_{\text{light}}}{4\pi\omega} \varphi, \quad (4)$$

$$b = \frac{A_0 + A_1 + A_2 + A_3}{4}, \quad (5)$$

$$a = \frac{\sqrt{(A_3 - A_1)^2 + (A_0 - A_2)^2}}{2}, \quad (6)$$

with $c_{\text{light}} \approx 3 \times 10^8 \frac{\text{m}}{\text{s}}$ being the speed of light. For more detailed information about the technological background of the PMD-sensor see Lange [4].

3.2. Real sensor behavior

The outcome of a real world PMD-sensor has a couple of artifacts coming along with the implementation of this kind of technology. The distinctive features of the resulting data are still the topic of active research particularly in the area of sensor calibration. This knowledge forms the basis of our simulation approach.

3.2.1. Flying pixels

First, a reconstructed depth image of a real PMD-device provides inaccurate distance information for pixels which cover depth inhomogeneities, yielding so-called flying pixels. The typical artifacts where the object's boundaries tend to drift either towards the background or the camera sensor are illustrated in Fig. 2.

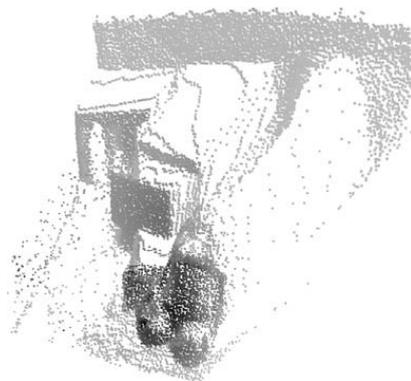


Fig. 2. Flying pixel artifacts. The pixels of the objects' boundaries drift in the direction of the background. The image shows real sensor data.

3.2.2. Wiggling

The wiggling error is the error which arises from the higher harmonics in the optical and the reference signal, i.e. the signal is not perfectly sinusoidal. This error is also known as the systematic deviation error (see Fig. 13). Fig. 3 shows the sensor data which is measured by capturing a planar surface. However, the actual distance of the surface is marked by the position of the reference plane. Rapp et al. [12] investigated the systematic errors and statistical uncertainties of the PMD-sensors. The distance based errors of the PMD-sensors are also examined by Lindner et al. [13,14] who presented an approach of the lateral and depth calibration of such sensors.

3.2.3. Motion blur

The sampling of the sinusoidal correlation function depends on phase delay and phase shift. This temporal integration leads to a motion blurring effect in dynamic scene setups since the phase images A_i which are based on a sequential exposure lead to varying object points observed by the respective pixels (see Fig. 4).

3.2.4. Noise

Since the cameras use active illumination (IR light) to measure the depth, they are sensible to errors from reflections in their surroundings. Moreover, the electronic components of a camera react sensitively to a variation in temperature. In addition to the thermic noise the signal is also influenced by dark current noise and reset noise [15], for example. Thus, the noise produced by a PMD-sensor is still subject of current research. In Fig. 3 the noise becomes visible in the cloud of points while capturing a planer surface.

4. Physical sensor model

In this section a sensor model, which is motivated by physics, is presented which is the basis for the simulation approach. The model must include the sensor characteristics described in the previous section in order to be comparable to real sensor data.

4.1. Illumination and radiosity

Radiosity is defined as the density of the total emitted and reflected radiation leaving a surface. The radiosity B is calculated for a point P of a lambertian surface which is illuminated by an area light source L as follows:

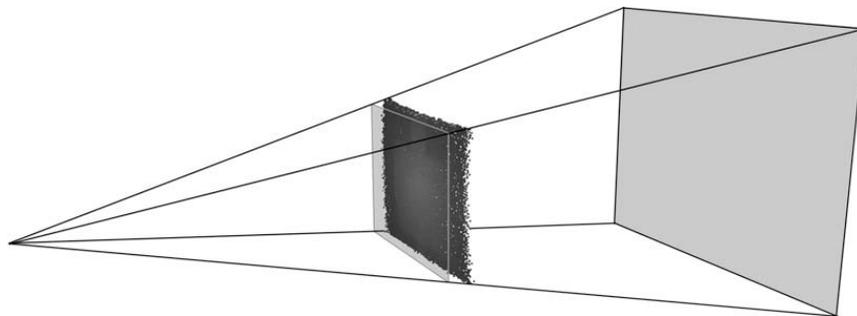


Fig. 3. Systematic deviation error. The wiggling error is indicated by the reference plane which marks the actual distance from the surface. The image shows real sensor data.

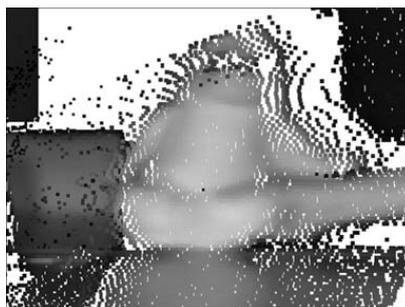


Fig. 4. Motion artifacts. The object is moving from the right to the left side. This results in motion artifacts which are especially visible at the boundary of the object. The image shows real sensor data.

$$B(P) = \int_{Q \in L} f(-\hat{\omega}_Q) \frac{1}{d_{p,Q}^2} (\hat{\omega}_Q \cdot \hat{n}_p) \cdot k_p \, dQ, \quad (7)$$

where we assume f to be the emission distribution function of L . Then $B(P)$ is the outgoing intensity at point P (see Fig. 5). The term $d_{p,Q}$ stands for the distance from P to a point Q of L . The brightness depends on the light's angle of incidence which is calculated by the dot product of the light direction $\hat{\omega}_Q$ and the surface normal \hat{n}_p of P . k_p is a constant which expresses the surface's diffuse reflectivity at P .

4.2. Point light illumination

The ideal demodulation of the correlation function c has been presented in Section 3.1 (see Eq. 1). The attenuation of the signal in proportion to a point's distance d_p to the sensor is already included in the optical function s , with $s \propto \frac{1}{d_p^2}$. Furthermore, s includes the radiosity in point P of the object's surface which means $s \propto B(P)$. If it is assumed that s equals the reference function g with an additional phase shift φ_p , then this can be expressed by:

$$s(t) = g(t + \varphi_p) \cdot \frac{B(P)}{d_p^2}. \quad (8)$$

In this first approach the illumination of the scene is restricted to a point light source. This simplification limits the light source L in Eq. 7 so that it consists of one sample Q which means it is not integrated over the area of the light source. Furthermore, if it is assumed that a PMD-pixel is affected by just a single point P of the surface, this results in the equation:

$$A_i = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} g(t + \varphi_p) \cdot \frac{B(P)}{d_p^2} \cdot g(t + \tau) \, dt. \quad (9)$$

A closer look at the physical setup of the TOF-sensors shows that a modification of Eq. 9 is still required. A single PMD-pixel covers not only the distance to one point of an object's surface, it rather represents the information which is covered by its solid angle Ω which is illustrated in Fig. 6. Therefore, it is necessary to integrate over Ω :

$$\begin{aligned} A_i &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} \left[\int_{\hat{\omega}_p \in \Omega} g(t + \varphi_p) \cdot \frac{B(P)}{d_p^2} \, d\hat{\omega}_p \right] \cdot g(t + \tau) \, dt \\ &= \int_{\hat{\omega}_p \in \Omega} \frac{B(P)}{d_p^2} \left[\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} g(t + \varphi_p) \cdot g(t + \tau) \, dt \right] \, d\hat{\omega}_p. \end{aligned} \quad (10)$$

Now \bar{c} is introduced as the *normalized* correlation function which is in contrast to c in Eq. 1 independent of the illumination and the attenuation of the signal, then Eq. 10 can be written as

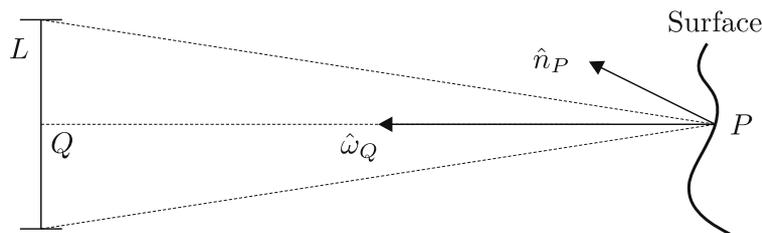


Fig. 5. The surface is illuminated by an area light source L . The brightness of the light which is emitted in P depends on $\hat{\omega}_Q$ and \hat{n}_p . These vectors form the light's angle of incidence.

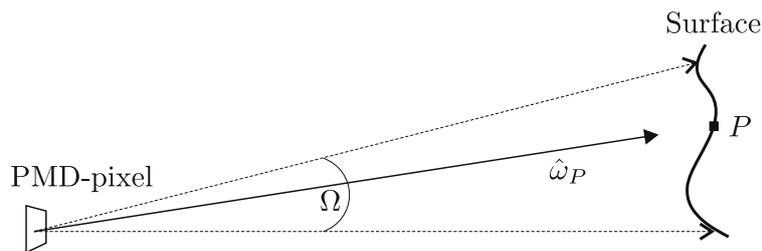


Fig. 6. A single PMD-pixel covers not just the distance to one point of the surface, it rather represents also depth inhomogeneities which are covered by its solid angle.

$$A_i = \int_{\hat{\omega}_p \in \Omega} \frac{B(P)}{d_p^2} \cdot \bar{c}_\tau(\varphi_p) \, d\hat{\omega}_p = \int_{\hat{\omega}_p \in \Omega} f(-\hat{\omega}_Q) \frac{1}{d_{p,Q}^2} (\hat{\omega}_Q \cdot \hat{n}_p) \cdot k_p \cdot \frac{1}{d_p^2} \cdot \bar{c}_\tau(\varphi_p) \, d\hat{\omega}_p. \quad (11)$$

If the position of the point light source is set exactly to the position of the sensor, which means that $d_{p,Q} = d_p$, a discretization of Eq. 11 yields:

$$A_i = \sum_{\hat{\omega}_p \in \Omega} f(-\hat{\omega}_Q) \frac{\hat{\omega}_Q \cdot \hat{n}_p}{d_p^4} \cdot k_p \cdot \bar{c}_\tau(\varphi_p). \quad (12)$$

4.3. Area light illumination

The approach of a point light illumination which is described in Section 4.2 can be extended to an illumination with area lights which represents the realistic light model of a real world device. If there is an area light source L , then all quantities which are related to the light position have to be varied. Then Eq. 11 leads to:

$$A_i = \int_{\hat{\omega}_p \in \Omega} \frac{k_p}{d_p^2} \left[\int_{Q \in L} f(-\hat{\omega}_Q) \frac{1}{d_{p,Q}^2} (\hat{\omega}_Q \cdot \hat{n}_p) \cdot \bar{c}_\tau(\varphi_{p,Q}) \, dQ \right] d\hat{\omega}_p. \quad (13)$$

It should be noted that the phase shift φ depends on the distance $d_{p,Q}$ from the point Q of the light source to the point of the surface P as well as on the distance d_p from the surface to the sensor, which means that $\varphi_{p,Q} \propto \|P - Q\| + d_p$. This formula is discretized by:

$$A_i = \sum_{\hat{\omega}_p \in \Omega} \frac{k_p}{d_p^2} \sum_{Q \in L} f(-\hat{\omega}_Q) \frac{1}{d_{p,Q}^2} (\hat{\omega}_Q \cdot \hat{n}_p) \cdot \bar{c}_\tau(\varphi_p). \quad (14)$$

5. Sensor simulation

In this section the theoretical sensor model of Section 4.2 is implemented which includes a point light illumination. The goal is the simulation of the generation process of the four phase images A_i . These phase images also form the basis for further data output of the PMD-sensor such as phase shift, depth, intensity and amplitude information.

5.1. Graphics hardware and simulation framework

We particularly aim at the generation of synthetic sensor data and the manipulation of camera parameters in real-time. The high performance can be achieved by using programmable Graphics Processing Units (GPUs) of modern graphic cards which automatically provide depth information of a synthetic scene. The GPU is programmed by implementing so-called vertex and fragment programs. Thus, the image data is directly written to the GPU memory with the help of so-called frame-buffer objects. This enables the advantages of modern graphics hardware which is fast, flexible and highly configurable [16,17]. These parallel stream processors offer a way to outsource a considerable part of the sensor simulation to the graphics card and thus a sensor-pixel can be processed very fast.

However, using standard graphics rasterization techniques means that there is a limitation to per-pixel data only which is available at the GPU. Thus, the flying pixels effect of a real smart pixel sensor is elaborate to implement, because no information about the surrounding pixels exists while processing a single pixel at the GPU. A super-sampling approach [6] based on a higher resolution rendering of the scene to deal with this problem is described in Section 5.2.1. The complete sensor model of Section 4.2 which is motivated by physics is implemented on the GPU. Therefore, the other characteristics of a real world sensor, which are summarized in Section 3.2, are realized.

Thus, the usage of an adequate correlation function (see Section 5.2.2) automatically results in the wiggling error. The motion blur effect is also taken into consideration by rendering the scene four times at different points of time according to the four phase images. Our approach of the incorporation of noise is explained in Section 5.2.3.

The concept of the simulation of a PMD-sensor is embedded in a Sensor Simulation Framework [6] which imports 3D scenes and provides tools for object and camera manipulation as well as animation. The framework (see Fig. 7) is hardware accelerated to allow for interactive simulator feedback, making use of the programmability of GPUs. Thus, the presented simulation algorithms are integrated easily.

5.2. Algorithm

As standard graphics rasterization techniques of the graphics cards are used it is possible to take advantage of the depth information about the scene which is automatically provided by the hardware. The depth data is only available within the range of $[0, 1]$ and includes a projective depth deformation, i.e. non-linear.

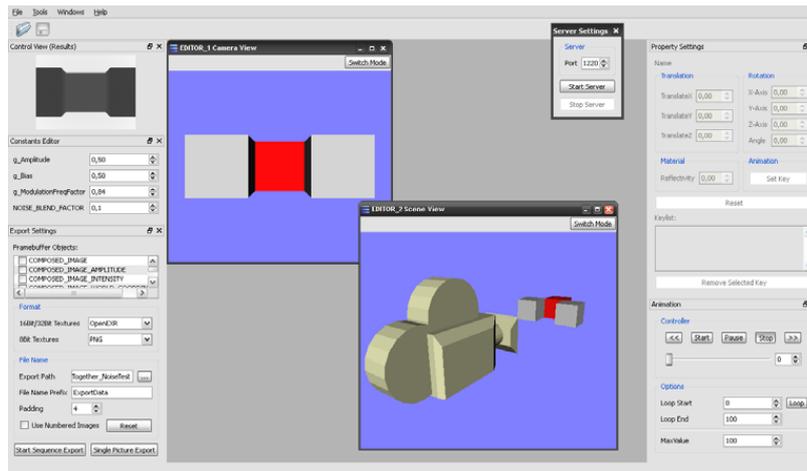


Fig. 7. The user interface of the simulation framework including manipulation and animation modules.

The ideal distance information d_{ideal} is then calculated by a perspective correction of the depth data. d_{ideal} also contains the transformation from cartesian coordinates into radial coordinates in order to fit the TOF-approach where light source and sensor are located in the same position.

First, the scene is rendered four times according to the phase delay τ_i in order to generate the four phase images A_i . The resulting distance d_{ideal} is then used for the calculation of the ideal phase shift:

$$\varphi_{ideal} = \frac{4\pi\omega}{c_{light}} d_{ideal}. \quad (15)$$

Second, the phase images A_i are calculated by sampling the normalized correlation function \bar{c} and incorporating the object's IR reflectivity and its orientation as well as the intensity attenuation. Thus, the adaption of Eq. 12 results in

$$A_i = f(-\hat{\omega}_Q) \frac{\hat{v} \cdot \hat{n}}{d_{ideal}^A} \cdot k \cdot \bar{c}_{\tau_i}(\varphi_{ideal}), \tau_i = i \cdot \frac{\pi}{2}, i \in \{0, 1, 2, 3\}, \quad (16)$$

with the intensity of the lambertian light source being $f(-\hat{\omega}_Q) = 1$. k is the object's reflectivity constant, \hat{v} is the vector from the object point towards the camera position and its illumination, and \hat{n} is the surface normal. \hat{v} and \hat{n} determine the brightness of the camera's illumination by the light's angle of incidence which is calculated by the dot product of the two vectors (see Eq. 7).

5.2.1. Spatial and temporal super-sampling

In Eq. 12, a PMD-pixel is influenced by the information which is covered by its solid angle Ω . Since only per-pixel data is available at the GPU a spatial super-sampling algorithm is used which is applied to a higher resolution rendering of the scene. Thus, a simulated pixel will also contain the information which relates to several directional samples within its solid angle (see Fig. 6). In Fig. 8, a simple super-sampling algorithm is illustrated which simulates the cover by a pixel's solid angle and thus leads to a satisfactory simulation of a real world sensor's flying pixel effects. The algorithm splits a pixel into several sub-pixels, and a sample is taken for each of the sub-pixels. The resulting pixel's phase value A for a certain phase image i is then defined as the mean value of all m sub-pixels j :

$$A_i = \frac{1}{m} \sum_{j=0}^{m-1} A_i^j. \quad (17)$$

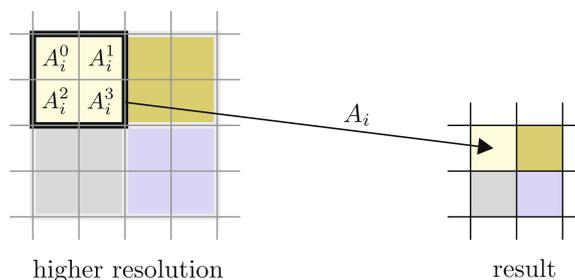


Fig. 8. The higher resolution image is sampled several times (here: four times). The value for the resulting pixel is based on the mean value. Thus, an approximation of the flying pixel effect of a PMD-device is implemented.

Note that in contrast to Peters et al. [8] the sampling of the normalized correlation function \bar{c} is applied to each sub-pixel (see Eq. 16), whereas Peters first computes the phase φ_{ideal} by superpositioning individual point responses and applies the correlation function afterwards. Peters et al. [8] do not give results regarding the flying pixel effect.

The motion blurring effect of a real PMD-device is implemented by rendering the scene several times according to the quantity of phase images A_i . Here the original scene is rendered four times one after the other at four different points of time. This means that during an animation sequence the geometric setup of the scene, e.g. camera and object position as well as orientation, will differ between the phase images. The left part of Fig. 9 shows the various images which are taken during a camera movement. The four phase images A_i which are calculated on the basis of the rendered images are displayed in the middle. Since phase images are used to calculate the resulting data such as the depth information, every single phase image contributes to the simulation result (figure on the right). Thus, the implementation of the motion blur is achieved by temporal super-sampling.

5.2.2. Correlation function

The actual function can not be measured exactly since the precise form of the correlation function c of a PMD-device is influenced by numerous physical and electronic effects. Although a measurement of the signal response of a real PMD-pixel can be achieved by directly illuminating it while applying all values of the internal phase delays τ_i , the simulation result differs from the real sensor behavior, i.e. the systematic deviation error is not reproducible (see Section 3.2.2). The simulation framework is flexible and highly configurable and thus offers a comfortable way of testing the influence of the correlation function's various implementations on the results. The values of the measured correlation function, which are displayed in Fig. 10, are written to a texture which afterwards is available for a texture look-up at the GPU. Thus, the correlation function's values are integrated into the calculation of the phase images A_i .

More realistic simulation results of the systematic deviation error are achieved by an approximation of the sinusoidal signal by various Fourier-modes which has also been examined in detail by Rapp et al. [12,11]:

$$c(\tau_i, \varphi_{\text{ideal}}) = \frac{1}{2} \cdot (1 + \cos(\varphi_{\text{ideal}} + \tau_i) - 0.041 \cdot \cos(3 \cdot (\varphi_{\text{ideal}} + \tau_i))), \quad (18)$$

with $\tau_i = i \cdot \frac{\pi}{2}$, $i \in \{0, 1, 2, 3\}$. This signal form results from a phenomenological approach, the plot of Eq. 18 is shown in Fig. 10.

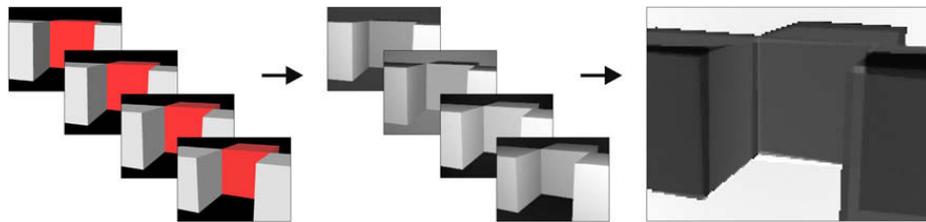


Fig. 9. Temporal super-sampling in a dynamic scene. The figure on the left shows the rendered images of a dynamic simulation scenario which contain color and depth information. The computation of the phase images (middle) is based on the rendered images. The temporal super-sampling leads to artifacts in the depth values (figure on the right) which are calculated on the basis of the simulated phase images.

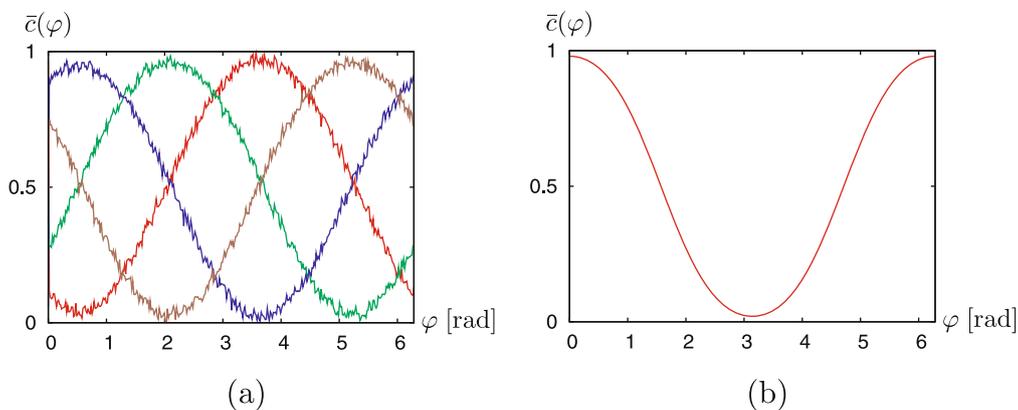


Fig. 10. Correlation functions. The measured correlation function for all values of the phase shift φ and the internal phase delay τ_i is displayed in (a). The correlation function of Eq. 18 is shown in (b).

5.2.3. Noise approach

A correct depth noise model for PMD-sensors is still a topic of current calibration research. Thus, a simple noise model is used in the sensor simulation which results in the plausible behavior of the simulated sensor. The idea is to include the noise even on the level of phase images. To be more precise, a certain amount of randomly chosen noise values is added to the values of the phase image A_i . Due to the fact that the noise is a per-pixel noise it is added to the resulting phase images after super-sampling the higher resolution image. This means, A_i of Eq. 17 is modified in the following way:

$$A_{\text{noisy},i} = A_i + \alpha \cdot n_{\text{rand}} + A_i \cdot \beta \cdot n_{\text{rand}}, \quad (19)$$

with $\alpha, \beta \geq 0$ being the noise coefficients, $n_{\text{rand}} \in [-1, 1]$ is a random Gaussian number, and $A_{\text{noisy},i}$ is the resulting pixel's phase value A for a certain phase image i . The signal-to-noise ratio is described by $\alpha \cdot n_{\text{rand}}$ which results in a constant pixel noise. The term $A_i \cdot \beta \cdot n_{\text{rand}}$ models intensity related noise effects. Since it is tricky to generate random numbers on graphics hardware, a texture is sampled which contains random numbers. The influence of the noise is then controlled by adjusting α and β . Noisy phase images then result in depth images which contain also noisy values.

6. Results

The simulation of the phase images of the PMD-device enables us to reconstruct also depth, intensity and amplitude data by using known demodulation formulas (see Section 3.1).

Some of the results are shown in Fig. 11. A PMD-device with a resolution of 160×120 pixels is simulated, the spatial super-sampling is computed on a higher resolution rendering of the scene with 1920×1440 pixels with down-sampling of 40×40 sub-pixels. The correlation function which is used in this scene setup is based on Fourier-modes (see Section 5.2.2) and the noise model uses $\alpha = 0.000035$ and $\beta = 0.01$. The depth information is coded in color in Fig. 11a. The temporal artifacts of the PMD-sensor are only visible during an animation sequence as shown in Fig. 11c. The flying pixels which become particularly visible in Fig. 12 are caused by the spatial super-sampling of the scene. The side-by-side comparison of the real data and the simulation data shows correspondence to a great extend regarding this sensor behavior. In the first sce-

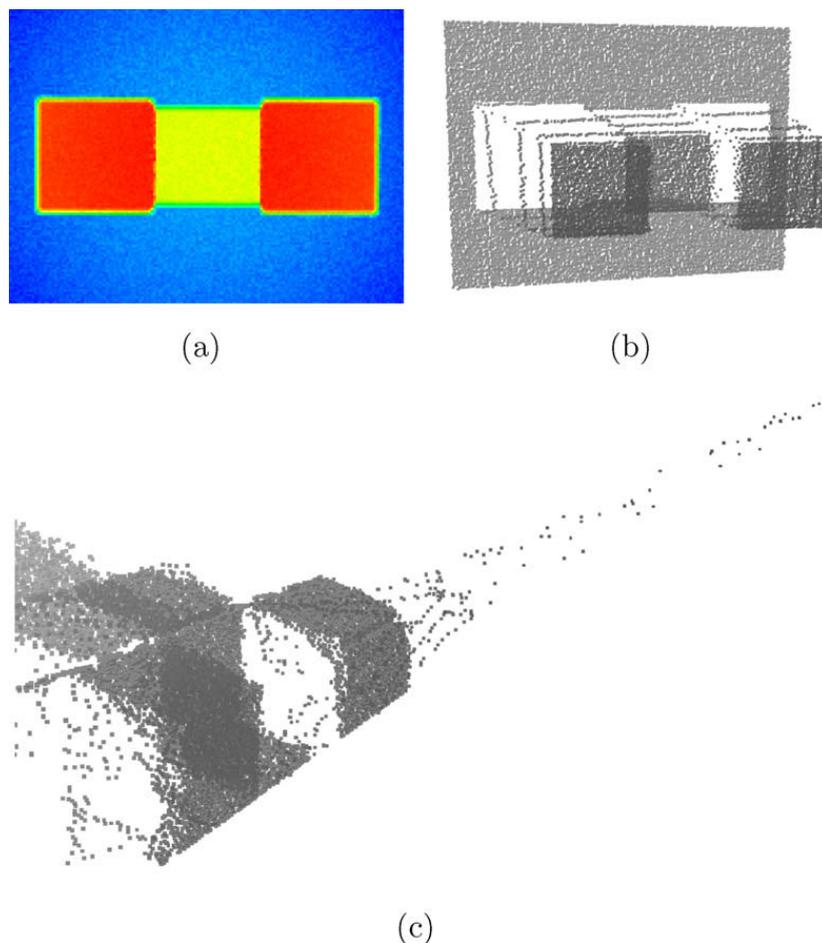


Fig. 11. Simulation results. The simulated depth information of a PMD-sensor is displayed in (a). In (b), the image of (a) is shown in a 3D-world-coordinate view. Its calculation is based on simulated phase images. The temporal artifacts of a PMD-sensor become visible during an animation sequence (see (c)).

nario, the object is positioned in a short distance from the wall in order to demonstrate the flying pixels which tend to drift from the object towards the wall. The distance between object and camera sensor is 1.67 m. The object itself is placed in front of the wall at a distance of 0.36 m (see Figs. 12c and d). On the other hand, a great distance of 4.13 m between the object and the wall results in flying pixels which are located in front of the object and drift in the direction of the camera sensor (see Figs. 12e and f).

7. Data evaluation

The validation of the simulation data is quite elaborate. The wiggling errors of the sensor simulation and a real PMD-device are displayed in Fig. 13. This simulation uses the Fourier-mode correlation function which is described in Eq. 18, since the correlation function which is based on a measured signal response (see Fig. 10a) results in slightly noisy data due to the specific measuring technique. The graph is plotted as a function of the measured distance for the image depth information of distances from 0.9 to 4.0 m. Fig. 13b shows this function measured by the sensor simulation. As the simulation results in relative intensity values, the data is linearly transformed on the y-axis which depends on the internal camera design with its latency time. However, the simulated data clearly indicate the typical deviation error of the real world sensor in Fig. 13a.

Although images of static scenes are relatively easy to verify, the validation of the data of an animation sequence is quite difficult because of the spatial and temporal artifacts which occur at the same time. The comparison of real dynamic PMD-sensor data with simulated data on a quantitative basis is very difficult. Therefore, we present only a qualitative comparison in Fig. 14. This shows a moving box in front of a wall with a lateral velocity of approximately 2 m/s.

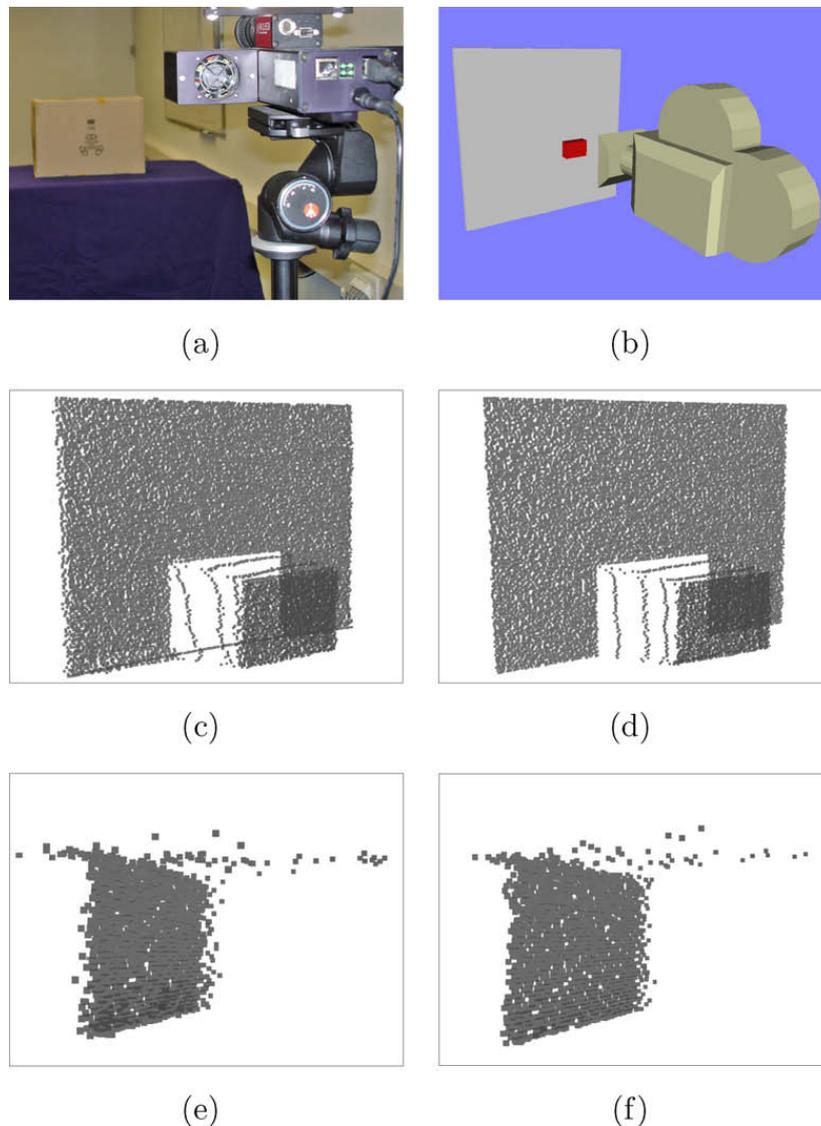


Fig. 12. Real PMD-data and simulation: comparison of flying pixel behavior. The first column shows the data of the real PMD-sensor while the second column represents the simulation data. (a) and (b) illustrate the scene setup. The object's width is 0.34 m and its height is 0.24 m. The flying pixels' behavior of the object with a short distance to the background is shown in (c) and (d). (e) and (f) display the effect of a great distance between object and background.

In Section 5 the theoretical sensor model of Section 4.2 was implemented which supports a point light source at the camera position.

The advantage of this approach is that the graphics hardware needs less computation time for the illumination of the scene and thus the simulation feedback is achieved in real-time with interactive frame rates. With regard to the model of the graphics card, the frame rates are between 15 and 30 frames per second. The approach of an scene illumination with area lights (see Section 4.3) was implemented, too. According to Eq. 14, an area light illumination was simulated which is approximated by up to 112 single lights which are located next to the sensor and are regularly distributed onto an area

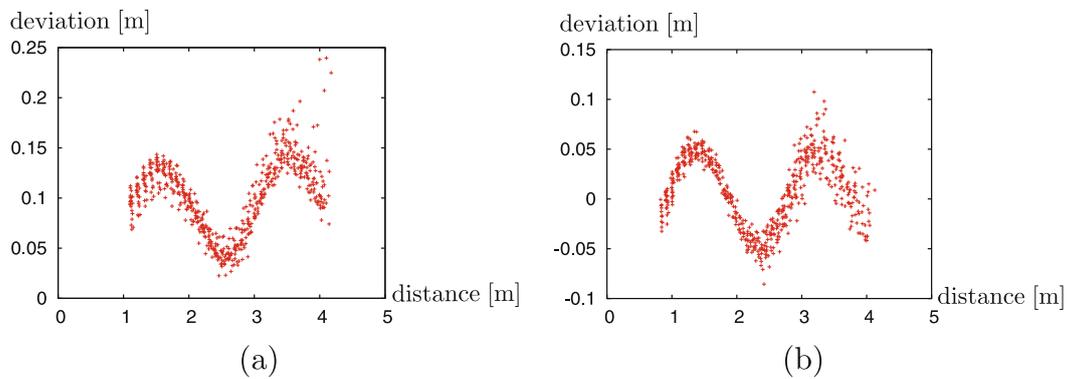


Fig. 13. Deviation error. The graph (a) of a real PMD-device with a resolution of 160×120 pixels is compared to the simulation data in (b). On the x -axis the measured distances are shown and the deviation error is marked on the y -axis (both in meters). The simulation uses a spatial super-sampling on a higher resolution rendering with 640×480 pixels and with a down-sampling of 4×4 sub-pixels. The noise model uses $\alpha = 0.000164$ and $\beta = 0.047$.

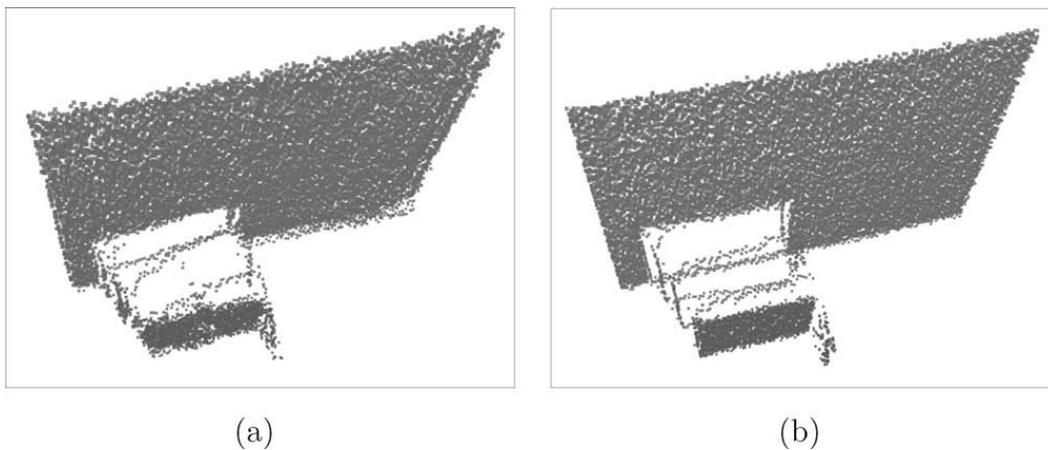


Fig. 14. Comparison of a depth image of a dynamic scene. (a) Shows the depth image acquired by a real PMD-device during a horizontally movement from the left to the right of the box which is shown in Fig. 12. The simulated data during the same object movement is displayed in (b). The typical temporal artifacts on the left and the right end of the object are highly comparable to the real sensor data as well as the spatial artifacts.



Fig. 15. Difference between the image of area lights approach and point lights approach. The error values are calculated by subtracting the phase image A_0 produced by the simplified point lights approach from the phase image A_0 generated by the area lights approach. It should be noted that the black regions differ less than 0.005% from the area lights approach and the white regions contain even smaller differences. This means that the resulting values are almost zero and thus the two approaches produce nearly the same simulation results.

of 19×4.5 cm. This corresponds to the specification of a real PMD-device (19 k model). The distance from the camera and its illumination to the center of the scene is 2 m. The difference of the two illumination models is marginally perceptible as illustrated in Fig. 15. However, the frame rate of the area light illumination decreases to less than one frame per second because of the high computational costs.

8. Conclusion

We have presented a fully equipped simulation model of a Photo Mixing Device (PMD). The simulation is based on a physical model and resembles the sensor's phase image generation. All relevant sensor effects like flying pixels, wiggling and motion blurring have been reproduced by applying this model in connection with the use of spatial and temporal super-sampling techniques. Furthermore, a noise model is integrated into the sensor simulation. The proposed algorithms are implemented in a hardware accelerated solution which uses the programmability of GPUs. This results in an interactive simulation feedback for the user who controls the simulation in real-time. The resulting data is comparable to real world sensor data including the data of dynamic scenes which contain camera and object movements.

The simulation concept is based on current research results concerning time-of-flight sensors and the Photo Mixing Device technology. The implementation of the sensor model, which is motivated by physics, can easily be adapted to latest research and we are interested in new correlation functions as well as appropriate noise models.

Acknowledgement

This work is partially funded by Grant V3DDS001 from the German Federal Ministry of Education and Research (BMB+F). In addition to this, we would like to thank Jens Orthmann, Marcel Piotraschke and Marvin Lindner of the Computer Graphics Group of the University of Siegen, Germany, for providing support concerning the implementation of the sensor model.

References

- [1] O. Faugeras, *Three-Dimensional Computer Vision*, The MIT Press, 1993.
- [2] R.I. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, second ed., Cambridge University Press, 2004, ISBN:0521540518.
- [3] Z. Xu, R. Schwarte, H. Heinol, B. Buxbaum, T. Ringbeck, Smart pixel – photonic mixer device (PMD), in: *Proceedings International Conference on Mechatronics Machine Vision*, 1998, pp. 259–264.
- [4] R. Lange, *3D Time-of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology*, Ph.D. Thesis, University of Siegen, 2000.
- [5] 3DV Systems, 2007. <<http://www.3dvsystems.com>>.
- [6] M. Keller, J. Orthmann, A. Kolb, V. Peters, A simulation framework for time-of-flight sensors, in: *Proceedings of the International IEEE Symposium on Signals, Circuits and Systems (ISSCS)*, vol. 1, 2007, pp. 125–128.
- [7] V. Peters, F. Hasouneh, S. Knedlik, O. Loffeld, Simulation of PMD based self-localization of mobile sensor nodes or robots, in: *Proceedings of the Symposium on Simulation Technique (ASIM)*, 2006.
- [8] V. Peters, O. Loffeld, K. Hartmann, S. Knedlik, Modeling and bistatic simulation of a high resolution 3d pmd-camera, in: *Proceedings of the Congress on Modelling and Simulation (EUROSIM)*, 2007.
- [9] B. Streckel, B. Bartczak, R. Koch, A. Kolb, Supporting structure from motion with a 3D-range-camera, in: *Proceedings of the Scandinavian Conference Image Analysis (SCIA)*, 2007, pp. 233–242.
- [10] K.-D. Kuhnert, M. Stommel, Fusion of stereo-camera and PMD-camera data for real-time suited precise 3D environment reconstruction, in: *Proceedings of the International Conference on Intelligent Robots and Systems*, 2006, pp. 4780–4785.
- [11] H. Rapp, *Experimental and Theoretical Investigation of Correlating TOF-Camera Systems*, Master's Thesis, University of Heidelberg, Germany, 2007.
- [12] H. Rapp, M. Frank, F. Hamprecht, B. Jahne, *A Theoretical and Experimental Investigation of the Systematic Errors and Statistical Uncertainties of Time-of-Flight Cameras*, 2007.
- [13] M. Lindner, A. Kolb, Lateral and depth calibration of PMD-distance sensors, *Advances in Visual Computing*, vol. 2, Springer, 2006, pp. 524–533.
- [14] M. Lindner, A. Kolb, Calibration of the intensity-related distance error of the PMD TOF-camera, in: *SPIE, Intelligent Robots and Computer Vision XXV*, vol. 6764, 2007, p. 67640W..
- [15] S. Klein, *Entwurf und Untersuchung von integrierten Ausleseschaltungen für hochauflösende 3D-Bildsensoren auf PMD-Basis zur Unterdrückung von Fixed-Pattern-Noise (FPN)*, Master's Thesis, University of Siegen, 2008.
- [16] R. Strzodka, M. Doggett, A. Kolb, Scientific computation for simulations on programmable graphics hardware, *Simulation Practice and Theory* 13 (8) (2005) 667–680.
- [17] J. Owens, *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, Addison-Wesley Professional, Ch. Streaming Architectures and Technology Trends, 2005, pp. 457–470.