Characterization of Pulse-Based Time-of-Flight Sensor Data

Bachelor Thesis

in Computer Science

submitted by

Markus Plack

born on 26th of Feburary, 1992 in Kirchen (Sieg)

Written at

Computer Graphics and Multimedia Systems Group
Faculty 12
University of Siegen, Germany.

in Cooperation with

Delphi Deutschland GmbH
Delphiplatz 1, 42119 Wuppertal

Advisors:

Prof. Dr. Andreas Kolb, Computer Graphics and Multimedia Systems, University of Siegen

M. Sc. Hamed Sarbolandi, Computer Graphics and Multimedia Systems, University of Siegen

Started on: 13th of Janurary 2016

Finished on: 29th of April 2016

Charakterisierung von pulsbasierten Time-of-Flight Sensor Daten

Bachelorarbeit

im Fach Informatik

vorgelegt von

Markus Plack

Geboren am 26. Februar 1992 in Kirchen (Sieg)

Angefertigt am

Lehrstuhl für Computergraphik und Multimediasysteme
Naturwissenschaftlich-Technische Fakultät
Universität Siegen

in Zusammenarbeit mit

Delphi Deutschland GmbH
Delphiplatz 1, 42119 Wuppertal

Betreuer:

Prof. Dr. Andreas Kolb, Computergraphik und Multimediasysteme, Universität Siegen

M. Sc. Hamed Sarbolandi, Computergraphik und Multimediasysteme, Universität Siegen

Beginn der Arbeit: 13. Januar 2016

Abgabe der Arbeit: 29. April 2016

Eidesstattliche Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Siegen, den 29. April 2016

Zusammenfassung

Diese Arbeit analysiert Daten einer pulsbasierten ToF Kamera. Der Fokus der Analyse liegt bei dem Fixed Pattern Noise eines solchen Systems und fehlerhaften Messergebnissen aufgrund von Bewegungs- oder Mehrpfadeffekten. Zusätzlich werden verschiedene Methoden zur Rauschentfernung angewendet.

Um das Verständnis des Systems und seiner Mängel zu erleichtern, wurde eine Simulation basierend auf dem in dieser Arbeit hergeleitetem detailliertem mathematischem Modell programmiert. Diese Simulation wurde einem Programm hinzugefügt, das ursprünglich zur Bedienung der Kamera und Visualisierung ihrer Daten programmiert wurde. Letztlich werden verschiedene Verbesserungen und Erweiterungen dieses Programms beschrieben.

Abstract

This thesis analyzes data of a pulse based ToF camera. The focus of this analysis is the fixed pattern noise of such a system and defect measurement results due to movement and multipath effects. Additionally some denoising methods are performed.

To facilitate understanding of the system and its defects a simulation was programmed based on a detailed derivation of a mathematical model performed in this thesis. This simulation was added to a programm that was originally designed to operate the camera and visualize its data. Lastly some improvements and additions to this programm are described.

Acknowledgements

This thesis was written in cooperation with *Delphi Deutschland GmbH*. I want to thank *Diethelm Noll* and *Albrecht Haack* for the chance to write this thesis and for their support.

Special thanks go to my university supervisors *Prof. Dr. Andreas Kolb* and *M. Sc. Hamed Sarbolandi* for their help and support.

I also would like to thank my *family* and *friends* who were always there for me and helped me along my way.

Markus Plack

Curiously enough, the only thing that went through
the mind of the bowl of petunias as it fell was
Oh no, not again.
Many people have speculated that if we knew
exactly why the bowl of petunias had thought that
we would know a lot more about the nature
of the Universe than we do now.

- Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

Table of Contents

Abbreviations

ToF	time of flight
PMD	photonic mixing device
DSNU	dark signal non-uniformity
PRNU	photo response non-uniformity

Frame Notation

$VTX_{1,Light,Full}$	raw frame of the first shutter with active illumination before acquisition
$VTX_{1,Light,After}$	raw frame of the first shutter with active illumination after acquisition
$VTX_{2,Light,Full}$	raw frame of the second shutter with active illumination before acquisition
$VTX_{2,Light,After}$	raw frame of the second shutter with active illumination after acquisition
$VTX_{1,Dark,Full}$	raw frame of the first shutter without active illumination before acquisition
$VTX_{1,Dark,After}$	raw frame of the first shutter without active illumination after acquisition
$VTX_{2,Dark,Full}$	raw frame of the second shutter without active illumination before acquisition
$VTX_{2,Dark,After}$	raw frame of the second shutter without active illumination after acquisition

$VTX_{1,Light}$	frame of the first shutter with active illumination $VTX_{1,Light} = VTX_{1,Light,Full} - VTX_{1,Light,Afer}$
$VTX_{2,Light}$	frame of the second shutter with active illumination $VTX_{2,Light} = VTX_{2,Light,Full} - VTX_{2,Light,After}$
$VTX_{1,Dark}$	frame of the first shutter without active illumination $VTX_{1,Dark} = VTX_{1,Dark,Full} - VTX_{1,Dark,After}$
$VTX_{2,Dark}$	frame of the second shutter without active illumination $VTX_{2,Dark} = VTX_{2,Dark,Full} - VTX_{2,Dark,After}$

VTX_1	final frame of the first shutter without the ambient light $VTX_1 = VTX_{1,Light} - VTX_{1,Dark}$
VTX_2	final frame of the second shutter without the ambient light $VTX_2 = VTX_{2,Light} - VTX_{2,Dark}$

Chapter 1

Introduction

The automotive industry offers a large array of applications for time of flight systems. Especially with the development of driver assistance systems and self-driving cars the demand for reliable and fast distance measurement devices is increasing.

While radar and lidar systems, which have been used for several years, provide good distance measurements, they are limited to certain directions. To use such a device to capture multiple directions mechanical systems are needed, which are prone to malfunctioning in the harsh conditions of automotive mobility.

This disadvantage does not apply to time of flight cameras, which are not only very compact, but also low priced relative to comparable systems. They are able to produce full 2.5D images of acceptable resolutions. Also the introduction of fast data processing systems makes harnessing the additional information provided by such cameras easier and enables them for real time applications.

Since those systems need an active illumination unit the topic of eye safety must not be unmentioned. In this regard pulse based systems offer an avantage, since they only use very short pulses of light that carry much less energy than the continuous illumination of phase based systems.

However those systems have some limitations. Aside from systematic errors influencing the accuracy the requirements on the hardware are high and thus rarely met by any system.

The goal of this thesis is to evaluate a prototype of a pulse based time of flight system.

The structure of the following five chapters is as follows.

Chapter 2 introduces the concepts of depth cameras with a focus on pulse based and phase based time of flight devices. For the pulse based technique a detailed mathematical model is derived, that was used to implement a simulation of this sensor.

Chapter 3 gives an overview of the work this thesis is based on.

Chapter 4 presents an analysis of the camera's fixed pattern noise, two experiments to analyse movement artefacts and multipath effects and a short overview and comparison of various denoising methods.

Chapter 5 introduces the software used and extended in the course of this thesis, including the above mentioned simulation.

Chapter 6 sums up the results of this thesis.

Chapter 2

Foundations

Section 2.1 focuses on the two ToF ideas and the two techniques used in camera systems today. In section 2.2 several non ToF techniques for range measuring and depth-cameras are presented. Section 2.3 concludes this chapter with a presentation of the Hamamatsu Demo Kit used in this thesis.

2.1 Time-of-Flight

The ToF principle incorporates the correlation between the runtime of a signal and its traveled distance due to a constant speed. While electromagnetic waves are used in ToF camera systems, the same principle is used by sonar systems which use sound waves. Equation 2.1 states this relationship as the traveled distance D of a signal is equal to its speed c times the traveled time ΔT.

$$D = c \cdot \Delta T \tag{2.1}$$

ToF cameras are called active cameras, meaning that they emit electromagnetic light, usually in the near infrared spectrum, which travels to the scene and back. So for those systems the signal covers the distance twice, yielding equation 2.2, where c_0 is the speed of light. [Wik16c]

$$D = c_0 \cdot \frac{1}{2} \Delta T \tag{2.2}$$

$$c_0 = 299.792.458 \frac{m}{s}$$

Note that this equation assumes that both sensor and illumination unit are at the same physical location, which is not possible. This induces an error into the distance computation.

Two principles to measure ΔT will be presented in the following sections. First a phase modulation based approach, where a modulated light is sent into a scene and its reflection is phase shifted depending on the distance. Second a pulse based approach, where a single - usually very short - pulse of light is emitted and its reflection captured in two shutters of the camera.

2.1.1 Phase Modulation Based

Current Photonic Mixer Devices (PMD) use a CMOS sensor capturing the reflection of an active illummanation unit illuminating the entire scene with incoherent, intensity modulated near-infrared light (NIR). The modulation of the captured signal is shifted to the emitted signal relative to the distance and the camera aims to measure this shift ϕ as seen in figure 2.1 to retrieve the distance based on knowledge of the period T of the modulation frequency. [Lin10]

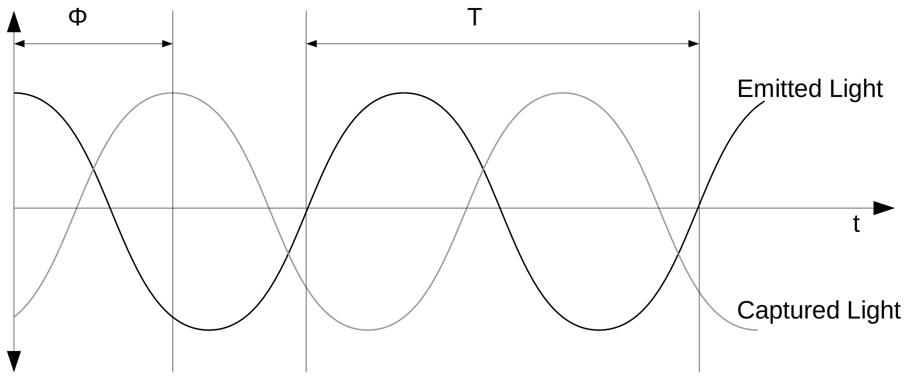


Figure 2.1: Principle idea of a phase based modulation measurement

Working Principle

For the distance measurement each pixel implements the correlation function $c(t)$ which is defined as

$$c_\tau(t) = r(t) \otimes o_\tau(t) = \lim_{T \to \infty} \int_{-T/2}^{T/2} r(t) \cdot o_\tau(t) dt. \tag{2.3}$$

$$\tau = i \cdot \frac{\pi}{2}, i = 0, 1, 2, 3$$

$o_\tau(t)$ describes the emitted signal shifted by an offset τ and $r(t)$ is the captured signal. Both funtions can be expressed as cosinusoidal functions:

$$o_\tau(t) = cos((\omega + f_m\tau) \cdot t) \tag{2.4}$$

$$r(t) = I + A \cdot cos(\omega t + \phi) \tag{2.5}$$

Here ω is the angular frequency of the emitted light, I an offset due to background light, which is assumed to be constant, A is the amplitude and ϕ represents the phase shift of the signal, which stems from the distance to the object. Using those functions equation 2.3 can be simplified as

$$c_\tau = \frac{A}{2} cos(\tau + \phi) + I. \tag{2.6}$$

Since there are three unknowns in this equation (A, ϕ and I) at least three different points need to be sampled. Usually four samples, denoted as A_i as described in equation 2.3 are sampled. Now the phase shift ϕ is given by

$$\phi = \arctan(\frac{A_3 - A_1}{A_0 - A_1}). \tag{2.7}$$

From the phase shift ϕ and the modulation frequency of the light source f_m the total runtime of the signal ΔT can be calculated as

$$\Delta T = \frac{\phi}{2\pi f_m}. \tag{2.8}$$

Using the previously direved relation between time of flight and distance from equation 2.2 the distance can be computed as

$$D = c_0 \cdot \frac{\phi}{4\pi f_m}. \tag{2.9}$$

Note that due to the wave character of the light only distances within the range $[0, 2\pi]$ can be measured unambiguously. This means that the maximum distance that can be measured D_{max} depends on the modulation frequency f_m as stated in equation 2.10. [LNL$^+$13]

$$D_{max} = c_0 \cdot \frac{1}{2f_m} \tag{2.10}$$

Distributors

Infineon Technologies is one major distributer of phase based ToF Sensors. Their current family of depth sensors is called REAL3^{TM} and consists of three versions, that succeed the IRS10x0C family. Resolutions range from 160 x 120 pixel in the IRS1615C to 352 x 288 pixel in the IRS1125C. [AG15]

A reference camera for these sensors is provided by pmdtechnologies and called CamBoard pico flexx, providing a resolution of 224 x 172 pixel at 5 to 45 fps in the range from 0.1 to 4 meters. The camera is connected via USB 2.0/USB 3.0 and runs under Windows 7/8, Ubuntu 14.04, Mac OS X ($\leq$10.9.2) and ARM Android. [pmd15]

(a) Infineon Technologies: IRS1125C [AG15]

(b) pmdtechnologies: CamBoard pico flexx [AG16]

(c) Heptagon Taro Tof Camera[Hep16]

Figure 2.2: Phase Modulation Based Sensors

Another distributor is Heptagon, offering a ToF camera named TARO, which is based on their so called Time-of-Flight 3DRanger^{TM} technology. It offers a resolution of 120 x 80 pixel covering a 90 degree field of view up to a maximum distance of 5 meters. [Hep16]

2.1.2 Pulse Based

A pulse based ToF camera emits only a short light pulse. This pulse is reflected in the scene and again captured by a CMOS sensor. The sensor contains two shutters for each pixel. When accumulating the light in the sensor first one and then the other shutter of the camera is active, as is shown in figure 2.3. Due to the distance to an object the reflected light pulse is shifted by ΔT and therefore the values of both shutters change.

Working Principle

In the following a mathematical model of such a camera will be derived. For simplicity only one pixel will be considered. In the course of this model description some assumptions are made that may or may not hold. These assumptions, as well as some simplifications, induce an error to the depth computation. These assumptions and their errors are mentioned whenever possible. Some of them will be discussed later.

Let $i(t) : \mathbb{R} \rightarrow \mathbb{R}$ describe the illumination generated by the camera's illumination unit in direction of the scene at time t. Now let $s(t) : \mathbb{R} \rightarrow \mathbb{R}$ describe the illumination hitting the sensor of the camera chip in a static scene.

It can be assumed that $s(t)$ shifts the signal $i(t)$ corresponding to the constant distance to an object by ΔT and changes its intensity proportional by a factor $\gamma \in \mathbb{R}_+$ depending on the objects reflectivity and its distance to the camera according to the inverse square law. Note that the change of intensity does not need to be proportional, but only strictly monotone and not dependent on the time t. The constantness of both time shift ΔT and intensity change can be violated in a dynamic scene.

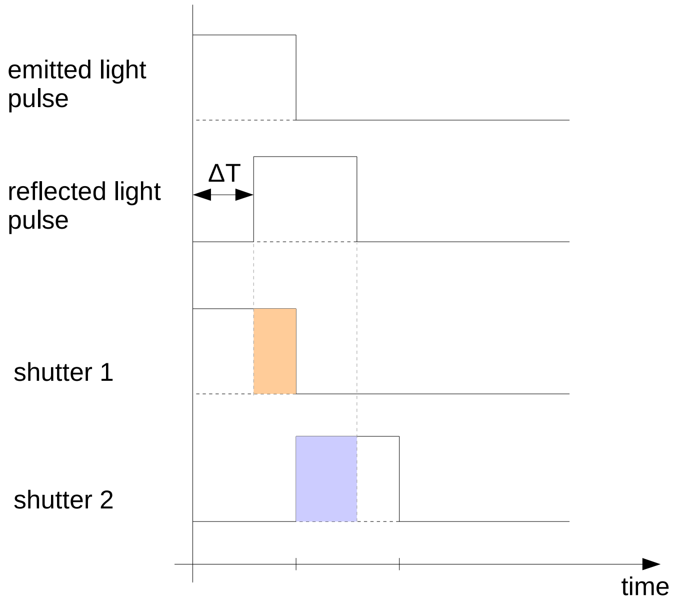


Figure 2.3: Pulse based camera timings

Also ambient light $a(t)$ of the scene is added to the signal. Using equation 2.2 we can now define the incident illumination of the sensor $E_v(t)$ as

$$E_v(t) = s(t) = \gamma \cdot i(t - \Delta T) + a(t) = \gamma \cdot i(t - 2D\frac{1}{c_0}) + a(t). \tag{2.11}$$

The response of the sensor is optimally proportional to the illumination over the total acquisition time. In detail, before starting the capture the current on the photodiode is set to a certain voltage U_f. When light hits the sensor the voltage decreases proportional to incident illuminance and time [Wik16a]. This process is not proportional in reality due to some errors induced in the electric circuits of the sensor. Let $R_{T_0,\theta}(t) : \mathbb{R} \to [0, \alpha], \alpha \in \mathbb{R}_+$ describe the response curve of the sensor for a measurement starting at T_0 and of duration θ. α is constant for a specific temperature. This is described in equation 2.12.

$$U_a(T_0, \theta) = \lim_{T\to\infty} U_f - (\int_{-T}^{T} R_{T_0,\theta}(t) \cdot E_v(t)\mathrm{d}t + \beta_\theta) \tag{2.12}$$

Here $\beta_\theta \in \mathbb{R}$ is constant for a specific acquisition time θ. Subtracting U_f and $U_a(T_0, \theta)$ yields the response $U(T_0, \theta)$ of the pixel:

$$U(T_0, \theta) = U_f - U_a(T_0, \theta) = \lim_{T\to\infty} \int_{-T}^{T} R_{T_0,\theta}(t) \cdot E_v(t)\mathrm{d}t + \beta_\theta \tag{2.13}$$

To simplify the model the emitted light pulse is optimally a rectangular function. So now we define $i_r(t)$ as a pulse emitted at $t = 0$ with intensity c and duration/width w.

$$i_r(t) = \begin{cases} 0 & ,t < 0 \\ c & ,0 \leq t \leq w \\ 0 & ,t > w \end{cases} \tag{2.14}$$

We assume that in the short time where the measurement is done the ambient light stays constant as $a(t) = A$. Then $E_v(t)$ becomes

$$E_v(t) = s_r(t) = \begin{cases} A & ,t < 2D\frac{1}{c_0} \\ \gamma c + A & ,2D\frac{1}{c_0} \leq t \leq 2D\frac{1}{c_0} + w \\ A & ,t > 2D\frac{1}{c_0} + w \end{cases} \tag{2.15}$$

To help the later following derivation we already split $E_v(t)$ in a background part $E_b(t)$ originating from the background illumination and pulse part $E_p(t)$ originating from the active illumination unit of the camera:

$$E_v(t) = E_b(t) + E_p(t) \tag{2.16}$$

$$E_b(t) = A, \qquad E_p(t) = \begin{cases} 0 & ,t < 2D\frac{1}{c_0} \\ \gamma c & ,2D\frac{1}{c_0} \leq t \leq 2D\frac{1}{c_0} + w \\ 0 & ,t > 2D\frac{1}{c_0} + w \end{cases} \tag{2.17}$$

Also the response curve of the sensor is assumed to be a rectangular function.

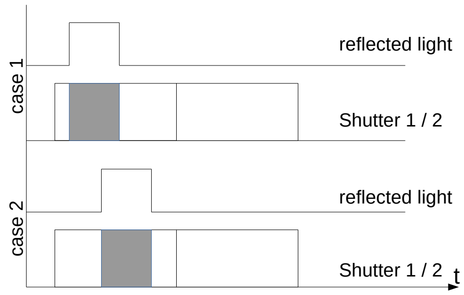
$$R_{T_0,\theta}(t) = \begin{cases} 0 & ,t < T_0 \\ \alpha & ,T_0 \leq t \leq T_0 + \theta \\ 0 & ,t > T_0 + \theta \end{cases} \tag{2.18}$$

These assumptions do not hold in reality. To study the error caused by this a simulation was programmed which offers the ability to use arbitrary functions as light pulses and sensor responses and study their effects.

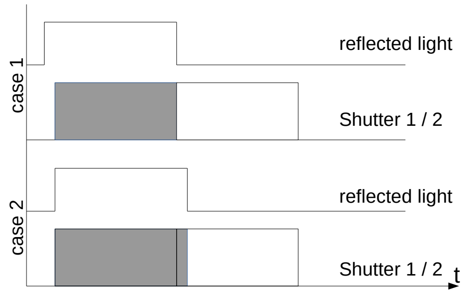
The choice of the right parameters for the light pulse length (w) and the shutters (T_0, θ_1, θ_2) depends mainly on the distance range that needs to be covered by the camera. Chosing $\theta_1 > w$ or $\theta_2 > w$ results in a situation, where the depth can no longer be unambigiously recovered over the whole possible range. See figure 2.4(a).

Chosing $\theta_1 < w$ or $\theta_2 < w$ does not yield ambigious cases, but unnecessarily complicates distance calculations. See figures 2.4(b) and 2.4(c).

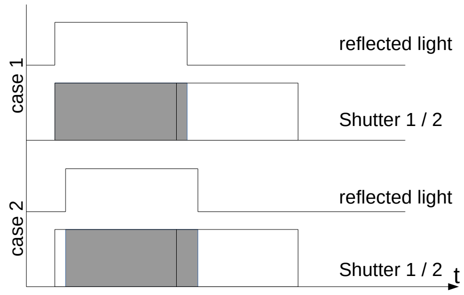
So given the range D_{max} that should be covered by the camera these values can be computed using equation 2.2 as in equation 2.19.



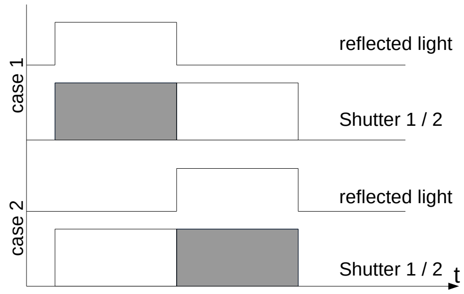
(a) When $\theta > w$ the unambigious range shrinks, as case 1 and case 2 can not be distinguished.



(b) When $\theta < w$ the quotient of $\frac{VTX_1}{VTX_1+VTX_2}$ does not change proportianal to the distance. The time shift between case 1 and case 2 is the same as in figure (c), but the quotient does not change by the same factor.



(c) See figure (b)



(d) For $\theta = w$ these are the minimum and maximum measurable distances. Between them $\frac{VTX_1}{VTX_1+VTX_2}$ changes proportional to the distance from 0 to 1.

Figure 2.4: Pulse Based Camera Parameter Choices

$$\theta = \theta_1 = \theta_2 = w = 2D_{max}\frac{1}{c_0} \tag{2.19}$$

Now we can change equation 2.13 into

$$U(T_0, \theta) = \lim_{T \to \infty} \int_{-T}^{T} R_{T_0,\theta}(t) \cdot E_v(t) \mathrm{d}t + \beta_\theta \tag{2.20}$$

For the depth measurement two responses VTX_1 and VTX_2 are needed, which are directly consecutive. Since both of them contain the ambient light Term A, a total of two such pairs needs to be captured, one with active camera illumination, which is called $VTX_{1,Light}$/$VTX_{2,Light}$, and one without, called $VTX_{1,Dark}$/$VTX_{2,Dark}$ respectively. Those are defined as follows.

$$\begin{aligned} VTX_{1,Light} &= U(T_0, \theta_1), \quad VTX_{2,Light} = U(T_0 + \theta_1, \theta_2) \\ VTX_{1,Dark} &= U(T_1, \theta_1), \quad VTX_{2,Dark} = U(T_1 + \theta_1, \theta_2) \end{aligned} \tag{2.21}$$

The choice of T_1 has to be such that no more light from the emitted light pulse is reaching the sensor. Formally speaking equation 2.22 must hold.

$$E_v(t) = A, \quad t \geq T_1. \tag{2.22}$$

By subtracting the light and dark frames the ambient light is removed from the equation.

$$\begin{aligned} VTX_1 = VTX_{1,Light} - VTX_{1,Dark} &= U(T_0, \theta_1) - U(T_1, \theta_1) \\ VTX_2 = VTX_{2,Light} - VTX_{2,Dark} &= U(T_0 + \theta_1, \theta_2) - U(T_1 + \theta_1, \theta_2) \end{aligned} \tag{2.23}$$

Using the above equations $VTX_{1,Light}$ can be evaluated as

$$\begin{aligned} U(T_0, \theta_1) &= \lim_{T \to \infty} \int_{-T}^{T} R_{T_0,\theta_1}(t) \cdot b(t) + \lim_{T \to \infty} \int_{-T}^{T} R_{T_0,\theta_1}(t) \cdot p(t) \mathrm{d}t + \beta_{\theta_1} \\ &= \theta_1 \alpha A + \begin{cases} (\theta_1 - 2D\frac{1}{c_0})\alpha\gamma c & , 2D\frac{1}{c_0} < \theta_1 \\ 0 & , else \end{cases} + \beta_{\theta_1} \end{aligned} \tag{2.24}$$

and $VTX_{1,Dark}$ as

$$\begin{aligned} U(T_1, \theta_1) &= \lim_{T \to \infty} \int_{-T}^{T} R_{T_1,\theta_1}(t) \cdot b(t) + \lim_{T \to \infty} \int_{-T}^{T} R_{T_1,\theta_1}(t) \cdot p(t) \mathrm{d}t + \beta_{\theta_1} \\ &= \theta_1 \alpha A + \beta_{\theta_1}. \end{aligned} \tag{2.25}$$

With this VTX_1 can be written as

$$\begin{aligned} VTX_1 &= \theta_1 \alpha A + \begin{cases} (\theta_1 - 2D\frac{1}{c_0})\alpha\gamma c & , 2D\frac{1}{c_0} < \theta_1 \\ 0 & , else \end{cases} + \beta_{\theta_1} - (\theta_1 \alpha A + \beta_{\theta_1}) \\ &= \begin{cases} (\theta_1 - 2D\frac{1}{c_0})\alpha\gamma c & , 2D\frac{1}{c_0} < \theta_1 \\ 0 & , else \end{cases}. \end{aligned} \tag{2.26}$$

Similarly VTX_2 can be evaluated using

$$\begin{aligned} U(T_0+\theta,\theta) &= \lim_{T\to\infty}\int_{-T}^{T} R_{T_0+\theta,\theta}(t)\cdot b(t) + \lim_{T\to\infty}\int_{-T}^{T} R_{T_0+\theta,\theta}(t)\cdot p(t)\mathrm{d}t + \beta_\theta \\ &= \theta\alpha A + \begin{cases} 2D\frac{1}{c_0}\alpha\gamma c & ,2D\frac{1}{c_0} < \theta \\ 2\theta - 2D\frac{1}{c_0}\alpha\gamma c & ,\theta_1 \leq 2D\frac{1}{c_0} < 2\theta + \beta_\theta \\ 0 & ,else \end{cases} \end{aligned} \tag{2.27}$$

and

$$\begin{aligned} U(T_1+\theta,\theta) &= \lim_{T\to\infty}\int_{-T}^{T} R_{T_1+\theta,\theta}(t)\cdot b(t) + \lim_{T\to\infty}\int_{-T}^{T} R_{T_1+\theta,\theta}(t)\cdot p(t)\mathrm{d}t + \beta_\theta \\ &= \theta\alpha A + \beta_\theta \end{aligned} \tag{2.28}$$

as

$$\begin{aligned} VTX_2 &= \theta\alpha A + \begin{cases} 2D\frac{1}{c_0}\alpha\gamma c & ,2D\frac{1}{c_0} < \theta \\ 2\theta - 2D\frac{1}{c_0}\alpha\gamma c & ,\theta_1 \leq 2D\frac{1}{c_0} < 2\theta + \beta_{\theta_1} - (\theta\alpha A + \beta_\theta) \\ 0 & ,else \end{cases} \\ &= \begin{cases} 2D\frac{1}{c_0}\alpha\gamma c & ,2D\frac{1}{c_0} < \theta \\ (2\theta - 2D\frac{1}{c_0})\alpha\gamma c & ,\theta_1 \leq 2D\frac{1}{c_0} < 2\theta \\ 0 & ,else \end{cases} . \end{aligned} \tag{2.29}$$

Finally the measured distance can be defined as

$$D = \frac{1}{2}c_0\theta\frac{VTX_2}{VTX_1 + VTX_2}. \tag{2.30}$$

Proof:

For $2D\frac{1}{c_0} < \theta$ the equation holds:

$$\begin{aligned} D &= \frac{1}{2}c_0\theta\frac{2D\frac{1}{c_0}\alpha\gamma c}{(\theta - 2D\frac{1}{c_0})\alpha\gamma c + 2D\frac{1}{c_0}\alpha\gamma c} \\ &= \frac{1}{2}c_0\theta\frac{2D\frac{1}{c_0}\alpha\gamma c}{\theta\alpha\gamma c} \\ &= \frac{1}{2}c_0\theta\frac{2D\frac{1}{c_0}}{\theta} \\ &= D \end{aligned} \tag{2.31}$$

For $\theta_1 \leq 2D\frac{1}{c_0} < 2$ the equation yields

$$\begin{aligned} D &= \frac{1}{2}c_0\theta\frac{(2\theta - 2D\frac{1}{c_0})\alpha\gamma c}{0 + (2\theta - 2D\frac{1}{c_0})\alpha\gamma c} \\ &= \frac{1}{2}c_0\theta. \end{aligned} \tag{2.32}$$

In this distance range all distances get clamped to the maximum measurable distance $D_{max} = \frac{1}{2}c_0\theta$, compare equation 2.10.

In the other cases the distance to be measured lies outside the range of the camera and the equation yields

$$D = \frac{1}{2}c_0\theta\frac{0}{0}. \tag{2.33}$$

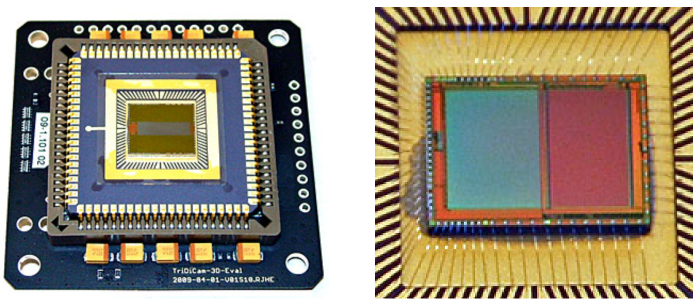
There is no information regarding the distance in the data and such pixels should be handled with care.

It should be noted, that the measurable range $[0, D_{max}]$ can easily be shifted by a factor $\delta > 0$ to $[\delta, D_{max} + \delta]$, by increasing T_0 by $\frac{\delta}{c_0}$, effectively introducing a delay between the light emission and the start of the capture. In this case equation 2.30 can be written as

$$D = \frac{1}{2}c_0\theta\frac{VTX_2}{VTX_1 + VTX_2} + \delta. \tag{2.34}$$

Distributors

The Japanese company Hamamatsu Photonics is a manufacturer of pulse based ToF sensors. They offer three different sensors, two of which are area sensors, and one line sensor, the S11961-01CR, with a total of 256 effective pixels. The area sensors have resolutions of 64 x 64 (S11962-01CR) and 160 x 120 (S11963-01CR). For more information on the S11963-01CR see section 2.3. [Ham16]



(a) TriDiCam: Line Sensor [Tri16b]

(b) TriDiCam: Area Sensor [Tri16a]

Figure 2.5: Pulse based sensors from TriDiCam

Another distributor is Tridicam, offering a 64 x 2 Pixel sensor and an area sensor. The second has been developed at the Fraunhofer IMS and uses Lateral Drift-Field Photodetectors offering a resolution of 128 x 96 pixel. [Tri16b] [Tri16a]

2.2 Non-ToF Techniques

In this section several other methods for range measurement are presented. Opposite to the ToF approach to range measuring, which derives the distance to an object from the time that an emitted light takes to travel to an object and back, most other methods rely on triangulation.

2.2.1 Triangulation

The aim of triangulation is to compute the distance to a surface point X from one end of a fixed baseline $\overline{AB}$ using the angles α at A, β at B of the triangle $\triangle ABX$ and the length $\|\overline{AB}\|$. [Lin10]

After computing γ as $\gamma = \pi - \alpha - \beta$ one can compute the desired distances $\|\overline{AX}\|$ and $\|\overline{BX}\|$ using the law of sines as

$$\|\overline{AX}\| = \frac{\|\overline{AB}\| \cdot \sin\beta}{\sin\gamma} \quad and \quad \|\overline{BX}\| = \frac{\|\overline{AB}\| \cdot \sin\alpha}{\sin\gamma} \tag{2.35}$$

2.2.2 Structured Light

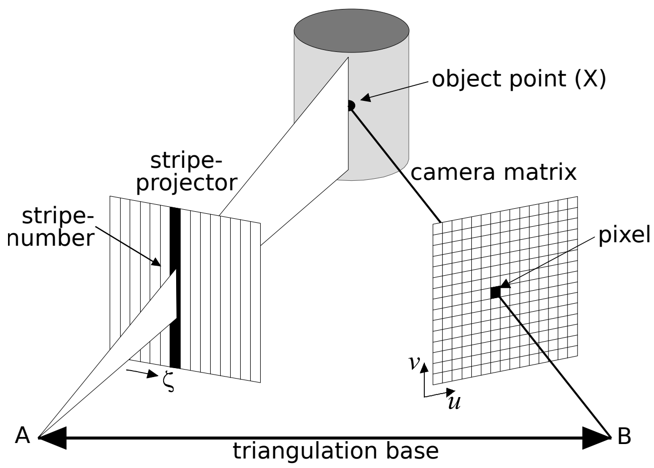


Figure 2.6: Structured Light Principle. Translated from [Wik16b]

A structured light technique, as has been used for example in the original Kinect of Misrosoft, incorporates the principle of triangulation using a light pattern that is projected on the object. The projector and camera form the baseline for the distance triangulation. The light pattern can be of various forms, ranging from line patterns to complex dot patterns. The task of the camera is to match the segments in the picture with the patterns of the projector. [Zha12]

2.2.3 Stereo Vision

A similar approach is stereo vision which uses two or more cameras and relies on pairing pixels of both cameras and deriving the distance based on their disparity. This method is inspired by the natural stereo vision. The task of the image processor is matching pixels from both images that show the same point of the scene. From the relative positions of the pixels and the two cameras the position of this point in 3D can be derived by triangulation. [Mat13]

2.3 Hamamatsu Demo Kit

In this thesis the evaluation kit for the pulse based S11963-01CR Distance Image Sensor was used.

2.3.1 Imager

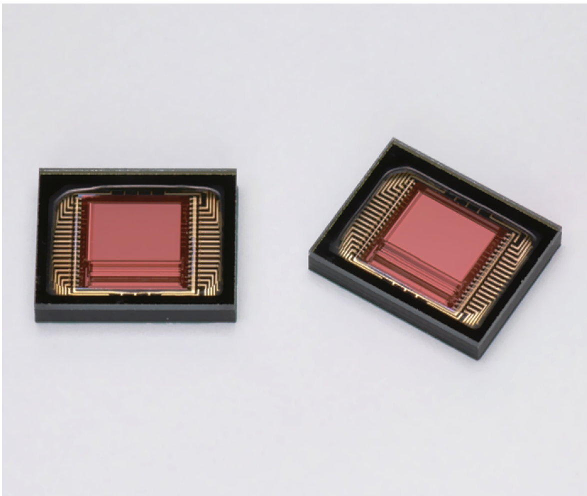


Figure 2.7: S11963-01CR Distance Image Sensor [K.K14]

The imager has a total number of 168 x 128 pixels of which 160 x 120 pixels are usable at photosensitive area of 4.8 x 3.6 mm. It covers the visible spectrum and near infrared light as can be seen in figure 2.8, though the evaluation kit uses a HOYA IR83N IR-transmission filter limiting the wavelengths to $830 \pm 10nm$, which is at the sensors peak sensitivity. [K.K14]

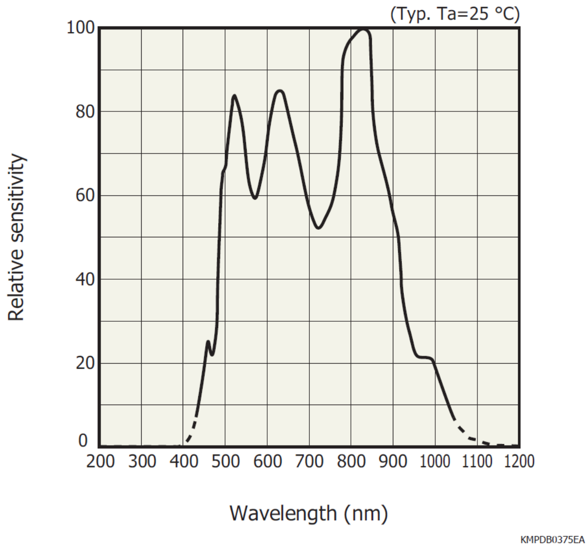


Figure 2.8: Spectral Response of the S11963-01CR imager [K.K14]

2.3.2 Evaluation Kit

The evaluation kit is powered by two power supplies. The original illumination unit of the kit was replaced because it did not illuminate the image homogeneously and had strong vignetting. The optical lens has a $37,5^{\circ} \times 27.7^{\circ}$ angle of view with an F-number of 1.2 focal length of $8mm$. [K.K14]

Communication

Communication with a computer is achieved via a gigabit ethernet interface. UDP is used as a protocol for both data and instruction transport. The payload of an UDP package starts always with an one byte transmission header followed by one or two bytes specifying the command. This is followed by a command depending payload. [PHO13a]

Parameters

Several parameters can be adjusted to control the behaviour of the camera, foremost the in section 2.1.2 discussed pulse width and the shutter widths. The most important parameters are listed in table 2.1. For a full list refer to [PHO13b].

Software

With the evaluation kit comes a software that offers basic functionality of communication with the camera and capturing data, but was not used in this thesis, since the more advanced Toffylizer is preferred.

Name	Unit	Default Value	Description
tacc	ms	30000	Total acquisition time for a single frame Must be equal to $VTX_1 + VTX_2 + VTX_3$
light pulse width	ns	30	Duration of the light pulse
light pulse delay	ns	75	Delay between light pulse emission and opening of the first shutter to incorporate the hardware delay of the illumination or to shift the measurement range
VTX_1	ns	30	Opening time of the first shutter
VTX_2	ns	30	Opening time of the second shutter
VTX_3	ns	9940	Time at which both shutters are closed between pulses

Table 2.1: Camera Parameters [PHO13b]

Chapter 3

Prior Work

This thesis builds upon the work of Simon Theiss in his master thesis “Analysis of a pulse-based ToF camera for automotive application”. The first version of the Toffylizer, a software tool to view, capture and manipulate ToF data, was programmed in the course of this thesis. Also a first camera analysis with the Hamamatsu Evaluation Kit and a correction was done there. The results should be summarized in section 3.1.

Some of the experiments in this thesis are based on “Kinect Range Sensing: Structured-Light versus Time-of-Flight Kinect”. These will be summarized in section 3.2.

3.1 Camera Analysis

3.1.1 Parameter Calibration

Light Pulse Delay

The light pulse that is emitted by the camera does not happen instantly when the acquisition starts, but after a certain delay. To measure this delay a simple setup was used. Using a certain pulse width and assuming equal values for both shutters a distance can be computed.

$$D = \frac{1}{2} \cdot c_0 \cdot w \cdot \frac{VTX_2}{VTX_1 + VTX_2} = \frac{1}{4} \cdot c_0 \cdot 40ns = 2.998m \tag{3.1}$$

By setting the camera in front of a wall that is $2.998m$ away several measurements with different delays were done and then the data was used to find the delay at which $\frac{VTX_1}{VTX_2} = 1$ holds. The best match was found to be a delay of $75ns$, which is not perfect, but the closest option, since only values that can be divided by 5 are accepted by the camera.

Timeshift

To test if the recorded distance shifts over time the camera was placed in front of a wall and left running for several hours. The change is most significant in the first hour and on average a total shift of $10cm$ from about $1.1m$ to $1.2m$ within three hours was measured.

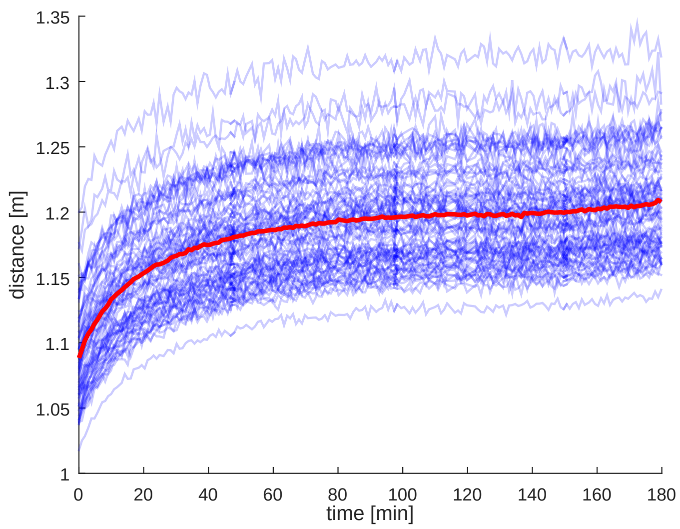


Figure 3.1: Timeshift

Oversaturation

Oversaturation of a pixel can happen for multiple reasons. Most obvious is the limited charge that the photosensitive element gets charged with before the shutter opens which can then only drop to zero. But also the AD conversion can cut off very high values or other parts in the processing of the images. It is desirable to measure values close to the oversaturation threshold due to the higher resolution they can offer, but having oversaturated pixels means a loss of information and therefore the oversaturation threshold was measured.

For that the acquisition time was increased and the highest possible values for both shutters were read. These were around 25000. [The15]

3.1.2 Distance Calibration

Measurement setup

For the distance calibration the camera was mounted on a movable rail facing a wall. In the range of $0.5m$ to $5m$ 150 frames were recorded for each step, with a stepsize of $2cm$. The pose of the camera needs to be known for a correct estimation of the ground truth. Since the resolution of the ToF camera is too low for a precise measurement, a CCD camera was rigidly attached to the ToF camera to record a checkerboard on the wall. Using this dual camera approach and the MultiCameraCalibration software the poses and the intrinsic camera parameters were obtained.

Polynomial Correction

The error of the measured distances regarding the ground truth can be approximated by a fifth degree polynomial as in 3.2.

$$Error_{reference}(d) = a_1x^5 + a_2x^4 + a_3x^3 + a_4x^2 + a_5x^1 + a_6 \tag{3.2}$$

Since correcting the error pixelwise with this polynomial would be too memory intensive it is only estimated for one reference pixel and applied over the whole image.

$$d_{u,v} = m_{u,v} - Error_{reference}(m_{u,v}) \tag{3.3}$$

Pixelwise linear correction

To further improve the results a pixelwise linear correction is applied to the data using a polynomial as in 3.4.

$$Error_{pixel}(u, v, d) = b_1x + b_2 \tag{3.4}$$

With this equation 3.3 can be extended to

$$d_{u,v} = m_{u,v} - Error_{reference}(m_{u,v}) - Error_{pixel}(u, v, m_{u,v}). \tag{3.5}$$

With this correction the error was reduced to $-50mm$ to $50mm$ in the near range.

Adaptive intensity calibration

In order to improve the error in the range beyond $3m$ an adaptive intensity calibration was performed. Since this is currently not supported by the cameras hardware, three different intensity settings were manually taken to cover the near ($0.5 + m$), medium (2.0 + m) and far (4.0 + m) range and the parameters of the cameras were set such that the values are close to, but below, the oversaturation threshold. For each of those ranges the above calibration was performed.

When using the best fit of those three sets for calibration the distance error of the camera was reduced to $-50mm$ to $50mm$ even in the far range. [The15]

3.2 Experiments

In their article „Kinect Range Sensing: Structured-Light versus Time-of-Flight Kinect“ Sarbolandi et al. introduce seven test scenarios to evaluate depth cameras. Those scenarios were each designed to yield information regarding one error source of such systems and get comparable results. Two of those experiments - the Reflective Board and Turning Siemens star - were used in this thesis for evaluation of the Hamamatsu Demo Kit.

For a detailed description of the eperimental setups and a comparison of the results with the Hamamatsu system see sections 4.2 and 4.3.

Reflective Board

This experiment was designed to measure the influence of strongly reflective objects on the distance measurement. In a scene containing such an object the emitted light of a camera may reach the sensor on more than one way, hence such effects are called multi-path effects.

Turning Siemens star

The turning siemens star experiment aims to identify defective distance measurements that result from movement in the scene also known as flying pixels.

[SLK15]

Chapter 4

Camera Analysis

In this chapter three different aspects of the Hamamatsu camera system are examined. First it was attempted to measure the fixed pattern noise of the sensor and try to improve image quality based on this results. Second a movement analysis is made and multiplath effects of the camera are studied. This chapter concludes with an overview over various denoising algorithms.

4.1 Fixed Pattern Noise

Fixed pattern noise refers to noise that occures in images of digital sensors that is temporally constant and results in visual patterns in the final image. It usually consists of two parts, the dark signal non-uniformity (DSNU) also called dark noise/ dark current and the photo response non-uniformity (PRNU) also called gain noise In this section an identification and correction of the FPN of the Hamamatsu camera is described.

4.1.1 Model Description

First we rephrase the derived model for the FPN in the camera system based on the model described in section 2.1.2. Since the variation of the noise over the sensor is important, now the model needs to be extended from one pixel to the whole sensor.

Let $\hat{U}_i(T_0, \theta, \overrightarrow{p}) : \mathbb{R} \times \mathbb{R}_+ \times I \rightarrow \mathbb{R}$ define the response of one shutter i of the sensor for all pixels $\overrightarrow{p} \in I = (1, ..., n) \times (1, ..., m)$ as follows.

$$U_i(T_0, \theta, \overrightarrow{p}) = \lim_{T \to \infty} \int_{-T}^{T} R_{i,T_0,\theta,\overrightarrow{p}}(t) \cdot E_v(t) \mathrm{d}t + \beta_{i,\theta,\overrightarrow{p}} \tag{4.1}$$

With $R_{i,T_0,\theta,\overrightarrow{p}}(t)$ being defined as

$$R_{T_0,\theta,\overrightarrow{p}}(t) = \begin{cases} 0 & ,t < T_0 \\ \alpha_{i,\overrightarrow{p}} & ,T_0 \leq t \leq T_0 + \theta \\ 0 & ,t > T_0 + \theta \end{cases} \tag{4.2}$$

Now $\alpha_{i,\overrightarrow{p}}$ and $\beta_{i,\theta,\overrightarrow{p}}$ are pixel dependent and correspond to the PRNU ($\alpha_{\overrightarrow{p}}$) and the DSNU($\beta_{\theta,\overrightarrow{p}}$). Since it is not known whether these values are the same for all shutters the model is extended to have possibly different values for each shutter as follows.

$$\begin{aligned} VTX_{1,Light} = U_1(T_0, \theta_1), \quad VTX_{2,Light} = U_3(T_0 + \theta_1, \theta_2) \\ VTX_{1,Dark} = U_2(T_1, \theta_1), \quad VTX_{2,Dark} = U_4(T_1 + \theta_1, \theta_2) \end{aligned} \tag{4.3}$$

4.1.2 Experimental Setup

The camera is placed in a dark room facing a diffuse white wall with its lens disattached and the illumination unit disabled. By removing the lens minor irregularities in the wall and its illumination are no longer recognized by the sensor and the capture can be seen as homogeneous. A separate illumination is used to control the light flow to the sensor and the exact illumination is measured as a reference.

For ten light intensities 300 frames are captured each, along with the measured illumination. Additionally 300 frames are captured with the lens covered to get the DSNU.

4.1.3 Results

In the case of the covered sensor equation 4.1 can be written as

$$\begin{aligned} U_i(T_0, \theta, \overrightarrow{p}) &= \lim_{T \to \infty} \int_{-T}^{T} R_{i,T_0,\theta,\overrightarrow{p}}(t) \cdot 0 \mathrm{d}t + \beta_{i,\theta,\overrightarrow{p}} \\ &= \beta_{i,\theta,\overrightarrow{p}}. \end{aligned} \tag{4.4}$$

This means that values of $\beta_{i,\theta,\overrightarrow{p}}$ can be estimated by taking the average of the 300 taken frames. Figure 4.1 shows their histograms. The derivation of the depth formula assumes that the pair $\beta_{1,\theta,\overrightarrow{p}}$ and $\beta_{2,\theta,\overrightarrow{p}}$ as well as the pair $\beta_{3,\theta,\overrightarrow{p}}$ and $\beta_{4,\theta,\overrightarrow{p}}$ are equal and therefore cancel out. It can be seen in figures 4.1(e) and 4.1(f) that their differences are close to zero with standard deviations of around 2.7 which is small enough to verify the assumption.

Having the illumination of the sensor time independent as $E(T) = E_j$ for $j = 1, ..., 10$ equation 4.1 can be written as

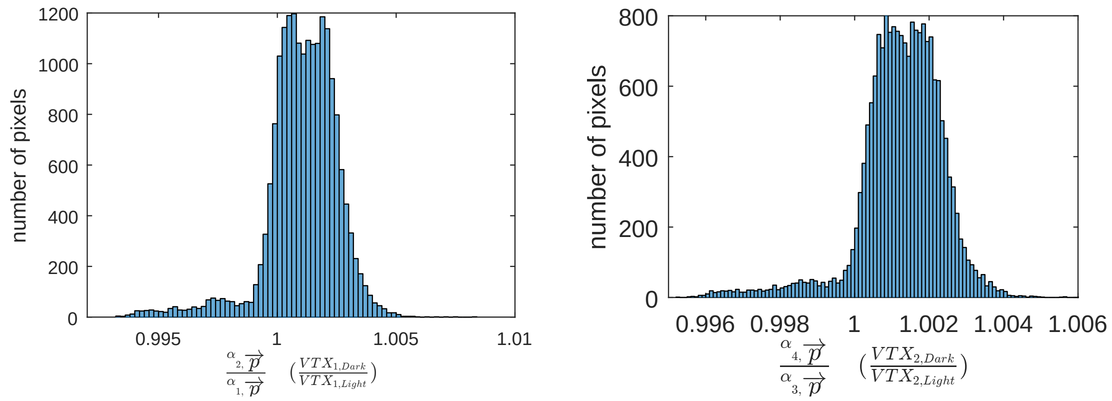
$$\begin{aligned} U_{i,j}(T_0, \theta, \overrightarrow{p}) &= \lim_{T\to\infty} \int_{-T}^{T} R_{i,T_0,\theta,\overrightarrow{p}}(t) \cdot E_j \mathrm{d}t + \beta_{i,\theta,\overrightarrow{p}} \\ &= \lim_{T\to\infty} E_j \cdot \int_{-T}^{T} R_{i,T_0,\theta,\overrightarrow{p}}(t) \mathrm{d}t + \beta_{i,\theta,\overrightarrow{p}} \\ &= E_j \cdot \theta\alpha_{i,\overrightarrow{p}} + \beta_{i,\theta,\overrightarrow{p}} \end{aligned} \tag{4.5}$$

Given the ten measured pairs of $U_{i,j}$ and E_j for each shutter and the previously found $\beta_{i,\theta,\overrightarrow{p}}$ values for $\alpha_{i,\overrightarrow{p}}$ can be estimated using regression. Their mean, minimum, maximum and standard deviation are listed in table 4.1.

Shutter	**Mean**	**Std. Dev.**	**Min**	**Max**
$\alpha_{1,\overrightarrow{p}}$ $(VTX_{1,Light})$	3060.79	88.74	2218.90	3406.24
$\alpha_{2,\overrightarrow{p}}$ $(VTX_{1,Dark})$	3064.07	89.07	2223.15	3416.53
$\alpha_{3,\overrightarrow{p}}$ $(VTX_{2,Light})$	4239.43	190.47	2949.99	4850.83
$\alpha_{4,\overrightarrow{p}}$ $(VTX_{2,Dark})$	4244.82	190.88	2957.08	4861.54

Table 4.1: PRNU Values of $\alpha_{i,\overrightarrow{p}}$

The table suggests that $VTX_{1,Light}$ and $VTX_{1,Dark}$ as well as $VTX_{2,Light}$ and $VTX_{2,Dark}$ are very similar. This can be seen by plotting the histograms of $\frac{\alpha_{2,\overrightarrow{p}}}{\alpha_{1,\overrightarrow{p}}}$ and $\frac{\alpha_{4,\overrightarrow{p}}}{\alpha_{3,\overrightarrow{p}}}$ as in figure 4.2.



(a) $\sigma = 0.00151 \quad E = 1.00106$

(b) $\sigma = 0.00112 \quad E = 1.00127$

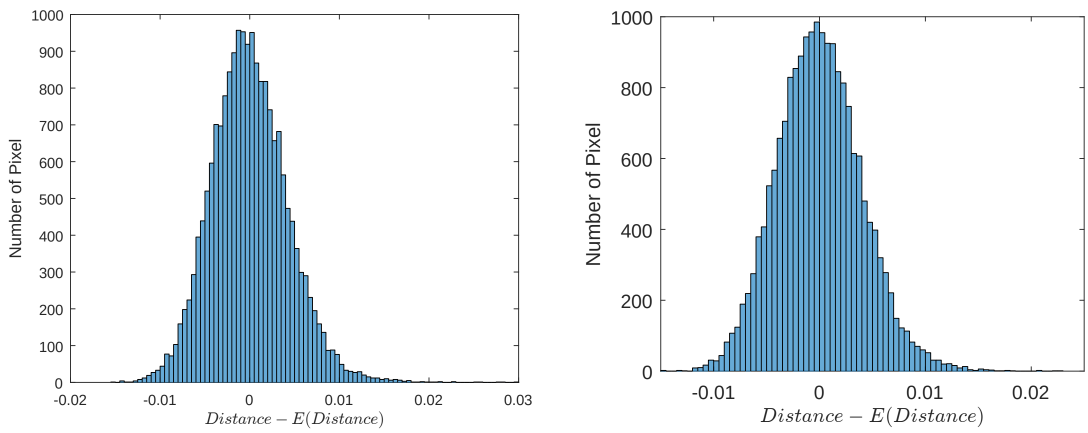
Figure 4.2: Histograms of the relative PRNU for both shutters.

4.1.4 Correction

A correction of the data was attempted to apply the found values of $\alpha_{i,\overrightarrow{p}}$ and $\beta_{i,\theta,\overrightarrow{p}}$ to improve the fixed pattern noise of the sensor.

To evaluate the correction a capture of a wall was taken. For both the original distance values and the distance values from the corrected data a plane was fitted through the backprojected points and the difference from the plane and each pixel was computed. If the correction was successfull the magnitude of the distances of the corrected data should be small compared to the original data.

Figure 4.3 shows the histograms of the distances for both original and corrected data. Unfortunately σ did only decrease insignificantly and therefore the correction can not be seen as successfull.



(a) $OriginalData, \sigma = 0.0043$ (b) $CorrectedData, \sigma = 0.0040$

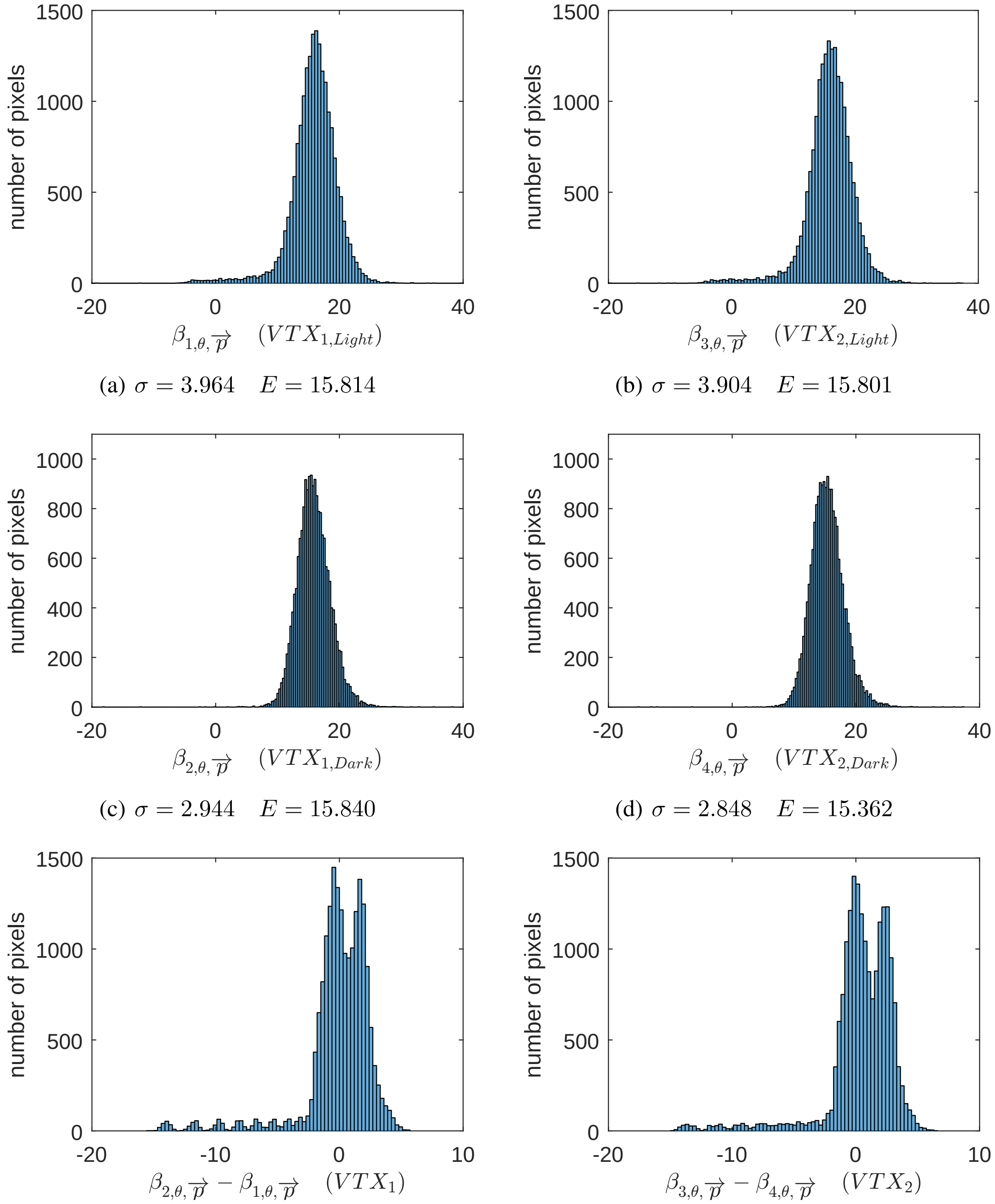
Figure 4.3: FPN Correction Results

4.2 Movement Analysis

When observing a dynamic scene while recording a depth image various errors can occure. In this section an analysis of the influence of lateral movement on the measurement result is presented.

4.2.1 Experimental Setup

The Camera is set up in front of a Siemens star mounted to a stepper motor to control the rotating speed of the star. The background is a homogeneous white wall. For several speeds ranging from 0 to 120 rotations per minute the camera records 200 frames of depth data.



(a) $\sigma = 3.964 \quad E = 15.814$

(b) $\sigma = 3.904 \quad E = 15.801$

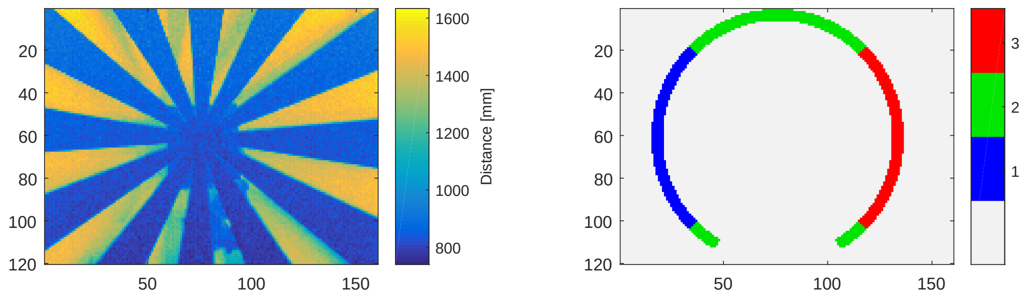
(c) $\sigma = 2.944 \quad E = 15.840$

(d) $\sigma = 2.848 \quad E = 15.362$

(e) $\sigma = 2.698 \quad E = -0.025$

(f) $\sigma = 2.731 \quad E = 0.440$

Figure 4.1: Histograms of the measured DSNUs



(a) Rawdata sample with a static star (b) Segmentation of the data in sections one to three

Figure 4.4: Siemens Star Setup

A section of interest is manually selected in the images such that the velocity is constant for all pixels. By plane fitting the back projected points using the data without movement a segmentation between front and back pixels is achieved. This approach is more reliable than a simple threshold using a fixed depth.

With this setup 1 rpm relates to 6 pixel swept in the defined area per second. We also define arcs 1,2 and 3 which relate to horizontal and vertical movement to test for differences as shown in figure 4.4(b).

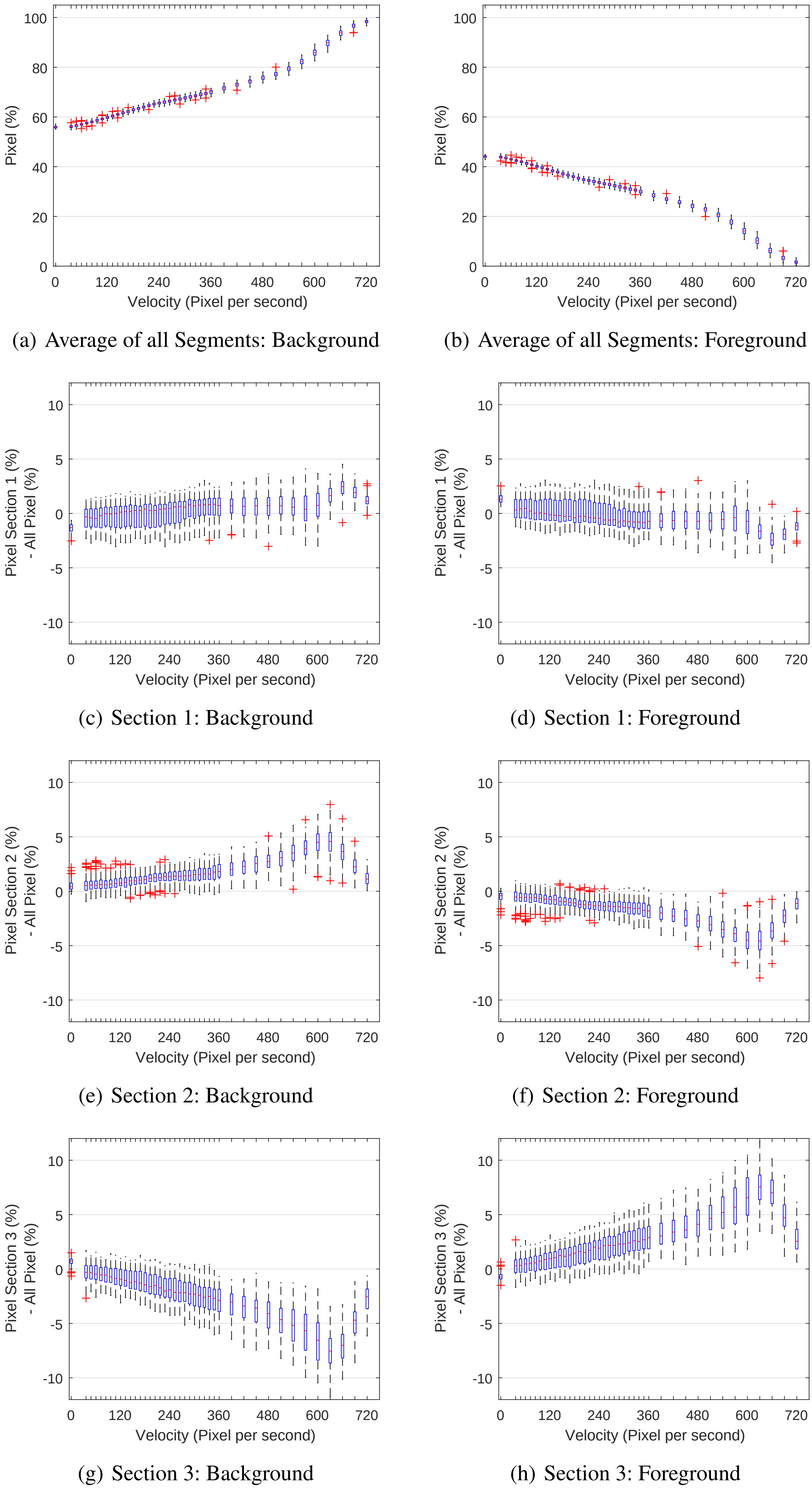
4.2.2 Results

The results are shown in figure 4.5. With no movement roughly 44% are classified as foreground and 56% are classified as background pixel. The distribution is the same at a velocity of 6 pixel per second. For higher speeds the distribution starts to spread. Starting at 480 pixel per second, where 24% are foreground and 76% are background pixel, the spread starts to increase faster until at the last measurement at 720 pixel per second only 1.6% of the section are foreground pixel.

Since the illumination unit of the camera is at the side of the lens, one would expect similar results in the left and right sections of the selection and different behavior in the other section. Figures 4.5(c) to 4.5(h) show the difference between each section and the average measurement for foreground and background classification. The measurement shows that section one (left side) has values close to the average, while section two has more background pixel than the average, especially for higher velocities with a peak at 630 pixel per second. Section three shows opposite behavior with more foreground pixel, again peaking at 630 pixel per second.

4.3 Multipath Effects

The distance computation is based on the assumption that the light that is captured by a single pixel originates from an area that has the same distance to the camera at all points. While this assumption is



(a) Average of all Segments: Background
(b) Average of all Segments: Foreground
(c) Section 1: Background
(d) Section 1: Foreground
(e) Section 2: Background
(f) Section 2: Foreground
(g) Section 3: Background
(h) Section 3: Foreground

Figure 4.5: Siemens Star Results

violated at curved sufaces, where a smoothing appears, and at edges, where flying pixel will occure, it is also violated in the case of multipath effects. That is a surface that not only reflects the direct incident light from the illumination unit, but also light that reaches the surface indirectly. This effect is studied in this section.

4.3.1 Experimental Setup

The Camera is placed in front of a 60x40cm Whiteboard that is vertically mounted on a turning table in front of a white projector screen. The projector screen can be removed to have a non-reflective black curtain in the background which is used for the reference measurement. This setup is shown in figure 4.6. For each angle from $0°$ to $90°$ in steps of $1°$ between the camera ray and the surface tangent $\vec{t}$, given as $\vec{t} = (\vec{n} \times \vec{c}) \times \vec{n}$, 100 frames are captured. This is done for both the white projector screen background and the black curtain. One vertical line in the center of the board is examined. This region is chosen at the pivot point of the rotation of the board such that the distance to the camera is constant for each angle.

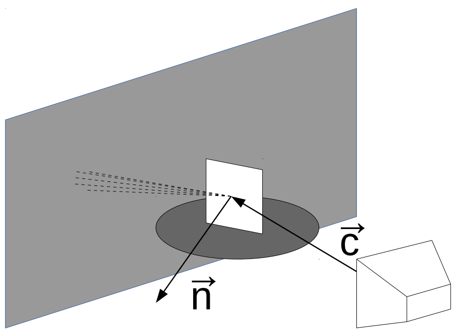


Figure 4.6: Setup of camera and reflective board for the multipath measurement

4.3.2 Results

Figure 4.7 shows a boxplot of the measured distances for all angles. The distance is almost constant at about $1.2m$ for both measurements until about $35°$. After that point the distances increases for both measurements. While the standard deviation for the multipath measurement roughly doubles the standard deviation of the reference measurement gets more than ten times as big up to $60cm$.

Additionally a section of the background was plotted the same way for both measurements. It can be seen, that the distribution of the distances converges to the distribution of the background for lower angles. At about $5°$ the whiteboard functions almost as a mirror, resulting in a measurement that is very close to the background.

The only difference between the highly reflective whiteboard and the white background is a smaller variance on the background. Otherwise they can not be differentiated. This can be seen as a result of the whiteboard not being a total reflector and therefore still influencing the measured values.

In the second case the background has a high standard deviation because the black curtain reflects little light and therefore decreases the signal-to-noise ratio. This effect can also be seen on the whiteboard as the angle decreases.

Altogether it can be said that with increasing reflectivity the measurement converges to the reflected background leaving almost no indication of the reflection and thus yielding wrong distance values.

4.4 Denoising

In this section several denoising filters and their application for depth data as well as their placement in the data processing pipeline is discussed. First the algorithms of the filters, namely gaussian, bilateral and NL-Means filter, are described and their benefits and disadvantages are layed out. This is followed by a discussion of some of the possiple placements between raw data from the camera and final depth data. This section is concluded by a visual comparison and discussion of the results as seen in the implementation of the filters using the Toffylizer.

4.4.1 Description of the algorithms

Gaussian Filter

The Gaussian filter is very easy to implement and fast. It is based on a convolution of the two dimensional image $\hat{I}$ by a two dimensional gaussian function. This process eliminates high frequencies such as noise from the image, but sharp edges are also smoothed. For a numerical implementation a finite kernel of a size usually between $3x3$ and $7x7$ is used. For an efficient implementation the kernel can be separated in a $nx1$ and a $1xn$ kernel that are applied in succession.

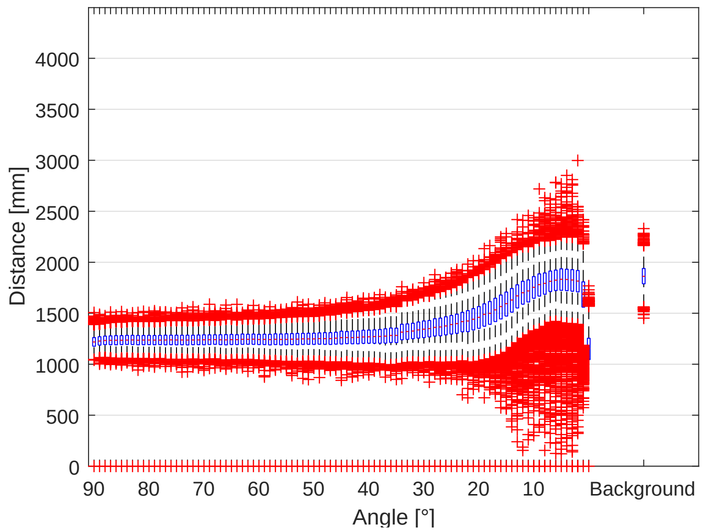
The gaussian filter is the fastest of the three discussed filters, but lacks the ability to keep edges in an image.

Bilateral Filter

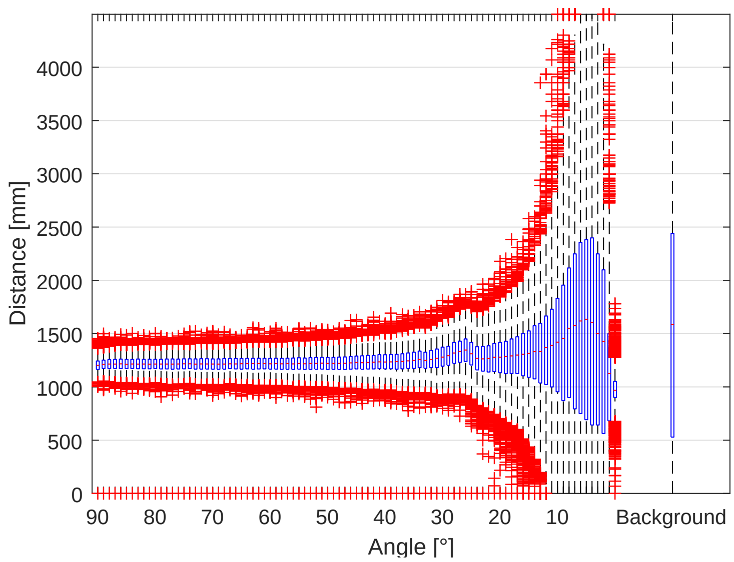
The bilateral filter aims to preserve borders in an image while removing the noise. To do so it computes a weighted average of the neighbourhood of a pixel using the distance applied to a 2D-Gaussian Kernel and the difference of intensity applied to a 1D-Gaussian Kernel as denoted in equation 4.6.

$$BF[I]_p = \frac{1}{W_p} \sum q \in S G_{\sigma_s}(||p-q||) G_{\sigma_r}(|I_p - I_q|) I_q \tag{4.6}$$

The Term W_p is used to normalize the weights and is defined as



(a) Multipath



(b) Reference

Figure 4.7: Multipath Results

$$W_p = \sum q \in SG_{\sigma_s}(||p-q||)G_{\sigma_r}(|I_p - I_q|). \tag{4.7}$$

Since the bilateral filter is also relatively fast it should be preferred to the Gaussian Filter whenever possible because of its edge preserving properties. [PKT09]

NL-Means Filter

The idea of the NL-Means Filter is to compute a pixel value again as the weighted average of other pixels. The weights are computed by the similarity of the neighbourhoods of both pixels. This is described in equation 4.8.

$$NL[v](i) = sumj \in Iw(i,j)v(j) \tag{4.8}$$

The weights $w(i,j)$ are computes as follows.

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{||v(N_j)-v(N_i)||^2_{2,a}}{h^2}} \tag{4.9}$$

Here $v(N_j)$ and $v(N_i)$ are the vectors of all values in the neighbourhood of pixel i/j, h controls the degree of filtering and $Z(i)$ is a normalizing factor defined as

$$Z(i) = \sum j e^{-\frac{||v(N_j)-v(N_i)||^2_{2,a}}{h^2}}. \tag{4.10}$$

This filter is more memory and time intensive than the before mentioned filters, but has good properties regarding noise removal and obtaining edges and (repeating) patterns in an image. [BCM05]

4.4.2 Placement of the Filters

The filters can be used at different steps in the data processing pipeline. For this thesis three different version of each filter have been programmed for each filter using the following data.

- **Option 1:** Processing $VTX_{1,Light}$, $VTX_{1,Dark}$, $VTX_{2,Light}$ and $VTX_{2,Dark}$ each
- **Option 2:** Processing VTX_1 and VTX_2 each
- **Option 3:** Processing the final distance image

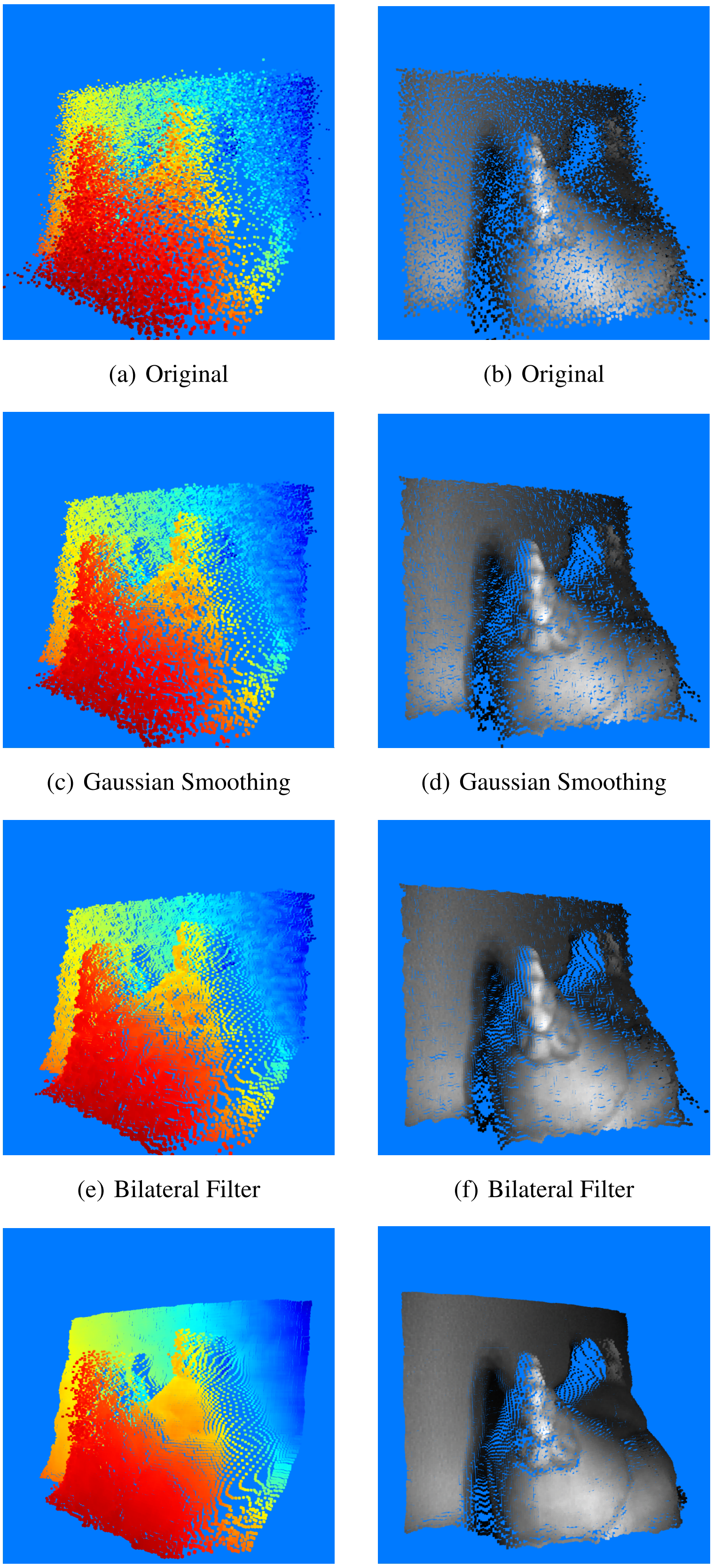
In terms of computational complexity option one is the most complex, followed by option two and lastly three. On the other hand the additional information provided might help improve the quality of the final image.

4.4.3 Comparison

Both option one and two did not produce depth images of improved quality for the bilateral and the NL-Means filter. For the Gaussian filter the differences were neglectable and thus option three should be chosen for a better performance. Figure 4.9 shows a comparison of the filters applied to the depth image with the raw data. The scene shows a clay figure of a chicken in front of a wall.

On a visual comparison the best denoising results were achieved with the computationally expensive NL-Means filter, followed by the bilateral filter and the gaussian filter, though in regions of fine details like the head the bilateral and the gaussian filter deliver even better smoothing results. Nevertheless the background is best smoothed by the NL-Means filter.

Figure 4.8: Image of the object used to demonstrate the denoising filters



(a) Original (b) Original

(c) Gaussian Smoothing (d) Gaussian Smoothing

(e) Bilateral Filter (f) Bilateral Filter

(g) NL-Means Filter (h) NL-Means Filter

Figure 4.9: Denoising Results: left column with color coded distance, right column with illumination intensity

Chapter 5

Toffylizer

The Toffylizer was first programmed by Simon Theiss in the course of his Master Thesis. In the course of this thesis multiple changes and additions have been made to the software. Those additions will be presented in this chapter.

Section 5.1 is about a simulation that has been added to facilitate understanding of and visualize different errors resulting from imperfect camera parameters. The integration of CUDA into the software is discussed in section 5.2, while section 5.3 presents the improved OpenGL renderer. The impoved communicationw with the camera is shown in section 5.4.

For a guide on installation and operation of the Toffylizer see appendix B. For the original description of the software, which discusses some aspects in depht see [The15].

5.1 Simulation

A simulator based on the derived mathematical model of the sensor in section 2.1.2 has been included in the Toffylizer. The shape of the functions of the light pulse ($i(t)$) and the shutter responses ($R_i(t)$) can be controlled. They are displayed in the upper half of the widget. It is possible to load them from a formatted text file or to generate them from scratch using an integrated tool based on Catmull-Rom-Splines using the create function. After that the time delay of all functions can be adjusted in the GUI, as well as the offset and sensitivity of the shutters with results being displayed instantly.

The final results are displayed at the lower half of the widget based on the assumed θ given by shutter time. The Tabs "Integral 1" and "Integral 2" plot $R_{T_0,\theta}(t) \cdot E_v(t)$ as part of $U(T_0, \theta)$ but without the given offset. In tab "VTX1/VTX2" both functions are plotted alongside with the offset. Finally "Distance" shows the simulated distances over the given range, as well as the optimal distances and the error. For rendering the graphs the free QCustomPlot widget was used.

Upon initialisation all curves are set to be optimal rectangular functions with a pulse width of $30ns$.

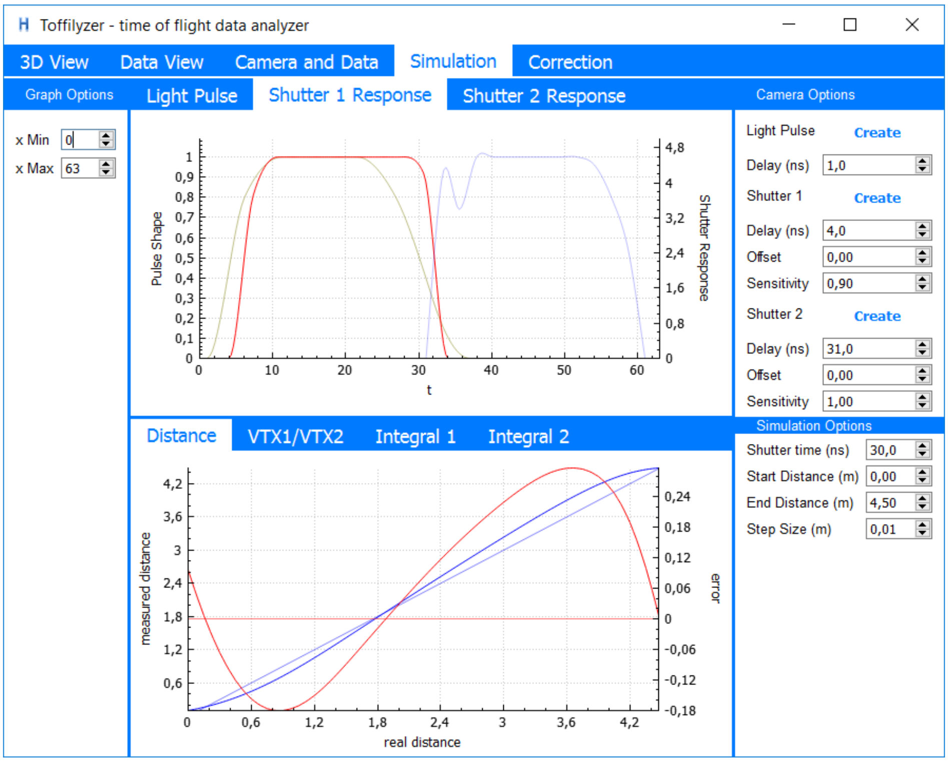


Figure 5.1: Screenshot of the Simulation Widget

5.2 Cuda Integration

Cuda has been chosen for the data processing because of its reasonably good availability on modern computers [Pir15] and the seamless integration into an OpenGL rendering Pipeline. Also the application of Nvidia processors in the automotive environment is rapidly increasing [Nvi14].

The raw data of the camera is stored in the ram upon capturing or loading. When necessary the data is sent to the GPU in an OpenGL vertex buffer. By binding this buffer to cuda arbitrary kernels can be run on it to compute the depth from the raw data and also to filter the images. Additionally thrust methods are used to retrieve meta data like mean, minimum, maximum and quantiles useful for data analysis and rendering (see figure 5.3).

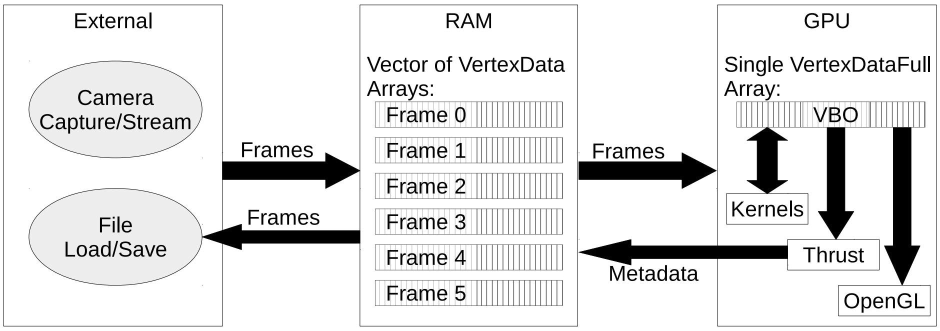


Figure 5.2: Data Flow

Raw data is stored in a struct named VertexData containing an array of 10 floats. For simpler acces the [] operator is overloaded to return the wanted values based on an enumeration for both host and device code. Processed data is stored in a similar struct called VertexDataFull, which contains 17 floating point numbers.

```
struct VertexData
{
        float data[10];
        __CUDA_CALLABLE__ float& operator[](size_t idx) {
                return data[idx]; }
        __CUDA_CALLABLE__ const float& operator[](size_t idx) const {
                return data[idx]; }
};
```

Listing 5.1: Data Structures in Types.h

5.3 New 2D/3D Renderer

The visualisation has been split into two separate widgets, one for 2D visualisation of the data and one for a full 3D projection of the captured frames. Both widgets can show either capured data or a live stream of a connected camera. Also both of them contain a small widget to control the processing and correction of the data on the fly. Metadata, such as extrema, average and quantiles are also displayed.

To improve the visual quality of the 3D visualisation the use of GL_POINTS for each vertex was replaced with a real geometry cube. This was done in the geometry shader of the rendering pipeline, where each point of input produces a triangle strip of 12 vertices forming the cube.

```
#version 150

layout (points) in;
layout (triangle_strip, max_vertices = 12) out;

in lowp float intensity[];
```

```
in lowp float opacity[];

out lowp float intensityOut;
out lowp float opacityOut;

uniform mat4 matrix;

 void main()
{
        float size = 0.99f;

        gl_Position = matrix * (gl_in[0].gl_Position + vec4(0,0,0,0));
        intensityOut = intensity[0];
        opacityOut = opacity[0];
        EmitVertex();

        gl_Position = matrix * (gl_in[0].gl_Position + vec4(size,0,0,0));
        intensityOut = intensity[0];
        opacityOut = opacity[0];
        EmitVertex();

        gl_Position = matrix * (gl_in[0].gl_Position + vec4(0,size,0,0));
        intensityOut = intensity[0];
        opacityOut = opacity[0];
        EmitVertex();

        [...]
}
```

Listing 5.2: Geometry Shader: PointShader.gs

5.4 Stable Camera Communication

The communication with the camera is based on UDP packages. While this makes for an easy implementation of a software controlling the camera the downside is the reliability of the communication. To avoid any problems with missing packages and provide the user with information regarding a failing communication some measures have been implemented.

Each UDP command send to the camera is acknowledged by the camera and some for of answer send back to the computer. This means that to be sure that a command reached the camera the software has recognize missing replies and react accordingly. Since it was found, that the camera sometimes doesn't react on commands at first try, the software resends the command several times. The number of maximum resends per command and waiting time before a resend is defined in CameraConnection.h. After that it is likely that the connection was lost and an exception is thrown, which is then show to the user as either a Message Box in the Toffylizer, or a console output in the CameraCapture program.

Also incoming frames that are not expected are disregarded and a stopping signal is send to the camera. This can happen when a previous instance of the application started a capture but did not stop it, or when the stop signal is lost.

Chapter 6

Conclusion

In the course of this thesis a mathematical model of a pulse based ToF camera was derived and a simulation built upon it. The simulation offers valuable insights into the various effects that can occure in such a system. An Attempt to reduce the fixed pattern noise of the sensor based on this model did unfortunately not yield satisfying results. A reason for that was not found.

An analysis of the effects of lateral movement on the distance image was performed with quite satisfying results regarding the low variation of distance values at higher velocities.

The multipath effects have been analysed and it was found that at a certain reflectance of an object the camera yields the depth values of the reflected scenery.

Various denoising algrithms were successfully implemented and tested in the Toffylizer. Due to the integration of cuda to the software with an tight interaction between the data processing and the rendering these filters can be run in real time on the camera data.

Appendix A

Attachment

The attached DVD contains the following data:

- **index.txt:** An index of the contained files and folders
- **thesis.pdf:** A pdf version of this thesis
- **src:** Source code of the Toffylizer
- **bin:** Binaries of the Toffylizer
- **sample:** Sample captures for the Toffylizer
- **bib:** Literature cited in this thesis
- **matlab:** Matlab code used in this thesis
- **hamamatsu:** Files provided by Hamamatsu

Appendix B

Toffylizer Manual

B.1 System Requirements

The software can be compiled for both 32 and 64 bit systems. A Nvidia GPU that supports GLSL version 4.50 is required.

It was build and tested on Windows 10 using Visual Studio 2013. Building on other operating systems/IDEs is possible, but changes in the CMake files may be required.

Also CMake version 2.8.11 or newer, Qt 5.5 and Cuda 7 are required.

B.2 Building

Use CMake to generate project files for Visual Studio. CMake will automatically detect Qt - when its environment variable is configured correctly - and the cuda libraries. The following targets will be generated within the project:

- CmakeTargets:
  - INSTALL
  - ZERO_CHECK
- ext:
  - libQCustomPlot
- lib:
  - libCamera
  - libConnection
  - libCore
  - libCorrection

  - libRenderer
  - libSimulator
  - libTools

- Toffilyzer:
  - CameraCapture
  - Toffylizer

- ALL_BUILD

Toffylizer should be manually set as the start project.

Building the INSTALL target in release or debug mode will build the Toffylizer and the command line tool CameraCapture and automatically copy the binaries along with the required libraries to a folder bin in the same directory as the top level CMakeLists for deployment.

B.3 Usage

B.3.1 Toffylizer

The Toffylizer offers five different tabs that can be selected at the top.

3D View

This widget offers the option to select a 3D view of either the current dataset or a live stream of the connected camera. Under Projection the user can chose between automatic and manual projection of the selected value on one of three offered heatmaps.

A small widget offers a quick access to the most important functions for a correction. Below this various informations of the currently rendered frame are displayed.

The Viewport can be rotated with a pressed left mouse button, zoomed with the mouse wheel and the scene can be moved with the right mouse button.

Data View

This widget offers the same options as the 3D View but for a two dimensional visualisation that can be zoomed with the mouse wheel and moved with the left mouse pressed.

Camera and Data

This tab shows a list of the currently loaded frames. It is possible to load, save and delete this list. Also a connection to a the camera can be established here.

Simulation

The simulation offers the possibility to change the light pulse and shutter functions and study the effects on the distance.

Correction

This widget is a bigger version of the small correction widget included in the 3D View and Data View.

B.3.2 CameraCapture

CameraCapture is a command line tool used to capture data from the camera. The possible parameters are listed below.

Parameter	**Description**
-ip X	set the ip of the camera to X
-port X	set the port of the camera to X
-verbose	the program will be verbose
-nostop	the camera will not stop sending data after the end of the program
-outfile X	set a prefix X for the outfiles to save the frames to. It is 'out' by default
-frames X	set program will capture X frames and save them to disk
-overwrite	the program will overwrite the output frame files. By default it does not
-help	print this help text
-Tacc	set the corresponding camera parameter
-Light_pulse_width	set the corresponding camera parameter
-Light_pulse_delay	set the corresponding camera parameter
-VTX1_width	set the corresponding camera parameter
-VTX2_width	set the corresponding camera parameter
-VTX3_width	set the corresponding camera parameter
-Vref	set the corresponding camera parameter

Table B.1: CameraCapture Parameters

Literature

[AG15] Infineon Technologies AG. Real3^{TM} image sensor family. Brochure, 2015. [http://www.infineon.com/dgdl/Infineon-REAL3+Image+Sensor+Family-PB-v01_00-EN.PDF?fileId=5546d462518ffd850151a0afc2302a58 Online; accessed 4-Feburary-2016].

[AG16] Infineon Technologies AG. Real3^{TM} image sensor family - infineon technologies, 2016. [http://www.infineon.com/cms/en/product/sensor/3d-image-sensor-real3/3d-image-sensor-for-consumer/real3-image-sensor-family/channel.html?channel=5546d4614937379a0149382f21d6007a Online; accessed 4-Feburary-2016].

[BCM05] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 60–65, Washington, DC, USA, 2005. IEEE Computer Society.

[Ham16] Hamamatsu. Distance image sensors — hamamatsu photonics, 2016. [http://www.hamamatsu.com/jp/en/product/category/3100/4005/4148/index.html Online; accessed 30-March-2016].

[Hep16] Heptagon. Sensing through light products — heptagon, 2016. [http://hptg.com/product/#sensing Online; accessed 16-Feburary-2016].

[K.K14] HAMAMATSU PHOTONICS K.K. Distance area image sensor s11963-01cr. Brochure, 2014. [http://www.hamamatsu.com/resources/pdf/ssd/s11963-01cr_kmpd1142e.pdf Online; accessed 30-March-2016].

[Lin10] Marvin Lindner. *Calibration and Realtime Processing of Time-of-Flight Range Data*. Phd thesis, Computer Graphics Group, University of Siegen, December 2010.

[LNL$^+$13] D. Lefloch, R. Nair, F. Lenzen, H. Schaefer, L. Streeter, M. J. Cree, R. Koch, and A. Kolb. *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, volume 8200, chapter Technical Foundation and Calibration Methods for Time-of-Flight Cameras, pages 3–24. Springer Berlin Heidelberg, September 2013.

[Mat13] Stefano Mattoccia. Stereo vision: Algorithms and applications. January 2013. [http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf Online; accessed 5-March-2016].

[Nvi14] Nvidia. Nvidia automotive driving innovation. Brochure, 2014. [https://www.nvidia.com/content/tegra/automotive/pdf/automotive-brochure-web.pdf Online; accessed 24-Janurary-2016].

[PHO13a] HAMAMATSU PHOTONICS. Command specification of evaluation kit for distance image sensor(s11963-01cr). Brochure, 2013.

[PHO13b] HAMAMATSU PHOTONICS. S11963-01cr distance image sensor evaluation kit (k3e-b90734). Brochure, 2013.

[Pir15] Usman Pirzada. Nvidia gained market share in q2 2015, now holds 81jon peddie research report, 2015. [http://wccftech.com/nvidia-gained-market-share-q2-2015-holds-81-market-jpr/ Online; accessed 24-Janurary-2016].

[PKT09] Sylvain Paris, Pierre Kornprobst, and Jack Tumblin. *Bilateral Filtering*. Now Publishers Inc., Hanover, MA, USA, 2009.

[pmd15] pmdtechnologies. Reference design brief camboard pico flexx. Brochure, 2015. [http://pmdtec.com/picoflexx/downloads/PMD_RD_Brief_CB_pico_flexx_V0201.pdf Online; Stand 04.02.2016].

[SLK15] H. Sarbolandi, D. Lefloch, and A. Kolb. Kinect range sensing: Structured-light versus time-of-flight kinect. *Journal of Computer Vision and Image Understanding*, 13:1–20, 2015.

[The15] Simon Theiss. Analysis of a pulse-based tof camera for automotive application. Master thesis, University of Siegen, July 2015.

[Tri16a] TriDiCam. Area sensor, 2016. [http://www.tridicam.de/en/products/array-sensor.html Online; accessed 30-March-2016].

[Tri16b] TriDiCam. Line sensor, 2016. [http://www.tridicam.de/en/products/line-sensor.html Online; accessed 30-March-2016].

[Wik16a] Wikipedia. Active pixel sensor — wikipedia, die freie enzyklopädie, 2016. [https://de.wikipedia.org/w/index.php?title=Active_Pixel_Sensor&oldid=150307254 Online; accessed 21-Janurary-2016].

[Wik16b] Wikipedia. Datei:principle of 3d fringe projection.svg - wikipedia, 2016. [https://de.wikipedia.org/wiki/Datei:Principle_of_3d_fringe_projection.svg Online; accessed 30-March-2016].

[Wik16c] Wikipedia. Time-of-flight camera — wikipedia, the free encyclopedia, 2016. [https://en.wikipedia.org/w/index.php?title=Time-of-flight_camera&oldid=710008356 Online; accessed 30-March-2016].

[Zha12] Zhengyou Zhang. Microsoft kinect sensor and its effect. *MultiMedia, IEEE*, 19(2):4–10, Feb 2012.