UNIVERSITÄT SIEGEN

Computergraphik und Multimediasysteme

Calibration and Real-Time Processing of Time-of-Flight Range Data

---

Kalibrierung und Echtzeit-Verarbeitung von Time-of-Flight Distanz-Informationen

vom Fachbereich Elektrotechnik und Informatik
der Universität Siegen

zur Erlangung des akademischen Grades
Doktor der Ingenieurswissenschaften (Dr.-Ing.)

genehmigte Dissertation

von

Marvin Lindner
Siegen - 25. Mai 2010

1. Gutachter:	Prof. Dr. Andreas Kolb, Universität Siegen
2. Gutachter:	Prof. Dr. Reinhard Koch, Christian-Albrechts-Universität Kiel
Vorsitzender:	Prof. Dr. Udo Kelter, Universität Siegen

Tag der mündlichen Prüfung: 15. Oktober 2010

Printed on non-aging wood- and acid-free paper.
Gedruckt auf alterungsbeständigem holz- und säurefreiem Papier.

Acknowledgments First of all, I would like to thank my supervisor *Prof. Dr. Andreas Kolb* for his great support and the opportunity he gave me to work in his research group. Thanks to him, I was able to gain appealing insights into various topics in computer graphics and computer vision.

In this context, I also would like to thank *Prof. Dr. Reinhard Koch* for accepting the co-advisorship of my PhD thesis.

Furthermore, I wish to thank my colleague *Nicolas Cuntz* for his helpful tips and creative input. Thanks for the good time I had by sharing an office with me.

Thanks as well to the rest of my *colleagues* for their support and company.

Special thanks to the *German Research Foundation (DFG)* for supporting the research project that made this thesis possible.

And finally, the deepest love and gratitude to my *family* and all my *friends* for supporting me.

Thank you

– Marvin Lindner

Abstract

The following thesis addresses a new technology for active range estimation, so-called time-of-flight (TOF) cameras. Based on the runtime principle, time-of-flight cameras allow the parallel acquisition of multiple distance information and thus enable, in contrast to other approaches, the acquisition of an entire scene in real-time. Consequently, TOF cameras are most suitable for many real-time systems in the area of automatization and interaction, where they are used for, i.a., object or gesture recognition. Due to their novelty, however, the accuracy of time-of-flight senors has been studied barely up to the present.

Experiments in the context of this thesis revealed error sources whose characteristics result in distance deviations of several centimeters. Those error sources therefore have significant impact onto the accuracy of acquired distance information and the results of vision systems as well. In addition, current TOF cameras are of low resolution compared to other range sensing approaches. Although this circumstance does not represent a real error source, it might have negative influence on the accuracy of automatization algorithms and therefore gives reasons for appropriate pre-processing of the acquired information.

Dealing with basic research, the presented work covers the investigation of the accuracy of current camera models as well as the basic processing steps that are necessary for the enhancement of range images regarding further processing steps.

In the context of camera accuracy, the thesis primary focuses on the systematical error characteristics and discusses the design of phenomenological calibration models covering demodulation- as well as intensity-related deviations. Furthermore, it deals with the compensation of TOF-specific motion artifacts and describes a compensation approach, which is based on optical motion estimation as well as an theoretical axial motion model.

In the context of data processing, the present thesis deals with the reduction of noise effects as well as the algorithmic refinement of distance information. Regarding distance refinement, two approaches are discussed: explicitly surface approximation using *Moving Least Square* surfaces as well as edge preservative data upscaling in image space. Furthermore, it covers the fusion of range images with supplementary information as provided by additional imaging sensors, in order to provide multi-modal data for sophisticated vision systems.

Zusammenfassung

Die vorliegende Arbeit befasst sich mit einer neuartigen, kostengünstigen Technologie zur aktiven Entfernungsmessung, sogenannten Time-of-Flight (TOF) Kameras. Basierend auf dem Laufzeit-Prinzip, erlauben diese die parallele Aufnahme mehrerer Tiefeninformationen und ermöglichen somit, im Gegensatz zu bisherigen Technologien, die Akquisition einer kompletten Szene in Echtzeit. Infolgedessen, eignen sich TOF Kameras besonders gut für vielerlei Echtzeit-Systeme aus den Bereichen der Automatisierung und Interaktion, und finden dort ihren Einsatz u.a. zur Objekt- und Gestenerkennung. Aufgrund ihrer Neuheit, wurde die Genauigkeit von Time-of-Flight Kameras jedoch bisher kaum untersucht.

Durchgeführte Untersuchungen im Rahmen dieser Arbeit haben Fehlerquellen aufgezeigt, die in ihrer Ausprägung zu signifikanten Abweichungen in den Tiefeninformationen von mehreren Zentimetern führen. Diese haben somit relevante Auswirkung auf die Ergebnisse von Vision-Systemen. Des Weiteren weisen aktuelle TOF Kameras, im Vergleich zu anderen Verfahren zur Entfernungsmessung, eine geringere Bildauflösung auf. Auch wenn dieser Sachverhalt im eigentlichen Sinn keine Fehlerquelle darstellt, kann er doch entscheidenden Einfluss auf die Genauigkeit von Automatisierungs-Algorithmen haben und rechtfertigt somit die algorithmische Verfeinerung von Tiefeninformationen.

Im Rahmen von Grundlagenforschung, umfasst die vorliegende Ausarbeitung sowohl die Untersuchung potentieller Fehlerquellen von TOF Kameras und deren Korrektur, als auch die grundlegenden Vorverarbeitungsschritte, die nötig sind um aufgenommene Tiefeninformationen für die weitere Verarbeitung zu verbessern.

Im Kontext der Messgenauigkeit, werden primär die Charakteristiken systematisch auftretender Messfehler, sowie der Entwurf entsprechender, phänomenologischer Korrekturmodelle für Demodulations- und Intensitäts-abhängiger Abweichungen betrachtet. Darüber hinaus wird die Reduktion TOF Kamera spezifischer Bewegungsartefakte innerhalb dynamischer Szenen, basierend auf einer optischen Bewegungsschätzungen, sowie eines theoretischen axialen Bewegungsmodels behandelt.

Im Bereich der Datenverarbeitung behandelt die vorgelegte Ausarbeitung zunächst die Reduktion von Rauscheinflüssen sowie die algorithmischen Verfeinerung von Tiefeninformationen. In diesem Zusammenhang werden zwei Verfahren zur Tiefenverfeinerung erörtert: explizite Oberflächen-Approximationen mittels *Moving Least Square* Oberflächen und kantenerhaltendes Upsampling im Bildraum. Ferner befasst sich die Ausarbeitung mit der grundlegenden Fragestellung zur Fusion von Tiefenbildern mit weiteren Informationen zusätzlicher, bildgebender Sensoren zur Erstellung multi-modaler Daten.

Contents

Introduction

In the present days, a wide range of intelligent systems are used in the context of manufacturing processes, the automotive industry, remote exploration, medical settings as well as surveillance systems. Their functionality, for instance, cover application areas like quality control, driver assistance or autonomous navigation. With ongoing progress in technology, however, the demand for intelligent and autonomous systems still increases. For this reason, many engineers and researchers deal with the general challenge to build improved artificial systems that copy human abilities, e.g., visual perception, to interact with their environment without assistance or supervision.

Computer vision as complement to biological vision and essential part of most intelligent systems generally deals with the basic concept how artificial systems obtain and interpret visual information from images or image sequences of the real world. Common applications, for example, are given by tasks like

- *object and environment modeling* for, e.g., automated object inspection, topographical modeling or obstacle detection
- *event detection* for example in the context of visual surveillance
- *data analysis* like automated indexing and organization of images or video sequences
- contact-free *human-machine interaction*

While the actual design of a computer vision system is highly application dependent, most systems commonly deal with one or more aspects of

- object modeling and scene reconstruction,
- object recognition and tracking, or
- pose and motion estimation

and incorporate research topics from the field of artificial intelligence and machine learning. Most of these problems are rather complex and while conventional systems work on two dimensional image data only, enhanced systems also incorporate range information in order to increase accuracy and overcome ill-posed problems in the two-dimensional image domain. Classical sensing techniques and devices for range data acquisition, however, are rather time consuming, unhandy to use

and / or costs expensive. Consequently, they are usually inappropriate for most real-time applications, common computer vision systems usually aim at.

In contrast to that, a new type of range-sensing devices – so-called time-of-flight (TOF) cameras or range imaging sensors – has been developed recently [XSH*98, TBF*05, JG01]. Unlike other systems, TOF cameras are very compact and low-priced off-the-shelf alternatives, that are capable to acquire full scene distance information in real-time. Hence, by now a broad variety of research project already exist that investigate TOF cameras in the context of computer vision and computer graphics related applications [KBKL10].

However, current time-of-flight cameras are affected by systematic error sources as well as motion artifacts, which significantly influence the measuring accuracy. Beside that, all available cameras are also of low resolution compared to classical approaches, which also might have an considerable impact onto the recognition and reconstruction results of vision systems.

Contribution The presented thesis investigates the fundamental question about the accuracy of current TOF cameras and present adequate calibration models and processing techniques for systematic error sources, including motion artifacts. It also discusses refinement techniques for acquired range images as well as their fusion with supplemental imaging sensors in order to obtain multi-modal information for sophisticated vision systems. According to the outlined topics, the specific contribution consists of

- **phenomenological calibration models** for systematic wiggling and intensity-related sensing errors. All models basically consider a set of reference data that covers the unambiguous range and serves as basis for the approximation of an error correction function. Regarding intensity-related errors, actually two calibration models are presented, which either consider coupled or decoupled input parameters [LK06, LK07a, LSKK10].

- the investigation of an **alternate phase shift demodulation approach**, which is based on the observations that the detected input signal is actually a mixture of a sinusoidal and rectangular modulation. The alternate demodulation approach finally leads to a lightweight calibration model, which requires a significantly reduced number of reference data compared to the phenomenological models [LKR08].

- a compensation approach for TOF camera specific **motion artifacts** that is based on pixel-wise surface tracking and seletcs corresponding phase samples in subsequent phase images. In addition, the theoretical impact of axial motion along the viewing ray is deduced and discussed [LK09].

- the investigation of **polygonal surface approximation** for range data upsampling using *Moving Least Surfaces*.

- the extension of an **edge preserving upscaling filter** for image data in the context of range data refinement. Main objective of the filter design was the handling of flying pixels as well as the extrapolation of invalid distance information along object contours [LLK08].
- the **fusion of range images and supplemental image data** in order to obtain multi-modal range data. Unlike monocular hardware solutions, the presented approach is designed for more general, binocular setups [LK07b].

Main objective of all approaches, has been the real-time capability of the particular technique. Beside the presented work, investigations in the context of interactive range data segmentation and classification finally contributed to the collaborated work on parallel mesh clustering [CKCL09].

Overview The structure of the thesis consists of the following four chapters.

Chapter 1 provides necessary fundamentals about range sensing and TOF cameras in the context of computer vision. It gives a short overview about common range measurement techniques and explains the working principle of current TOF cameras, which is based on phase shift determination. Furthermore, it summarizes typical error sources for TOF cameras, that arise from both general time-of-flight as well as hardware-related issues. Afterwards, a definition of the traditional pinhole camera model is given along with an explanation of the classical camera calibration task known from traditional vision systems. The chapter is completed by a description of fundamental techniques for vision-based motion estimation as well as an overview about current graphics hardware and its advantages with respect to real-time capabilities.

Chapter 2 discusses the application of intrinsic calibration to TOF cameras, its shortcomings as well as current solutions. It also covers the contributed calibration models for systematic wiggling- and intensity-related distance deviations along with related calibration models published at the same time. Furthermore, it investigates an alternate phase shift demodulation approach and addresses a compensation approach for TOF camera specific motion artifacts.

Chapter 3 focuses on essential range image processing tasks like denoising, refinement and multi-sensor fusion. It gives general information for range image denoising and binocular data fusion, and introduces *Moving Least Square Surfaces* as one possible refinement technique in the context of explicit surface approximation. Furthermore, it describes an edge preserving upsampling technique for real-time range data refinement that works the image domain, and introduces a general processing framework for mixed CPU/GPU computation.

Chapter 4 finally gives a summary and discussion of achieved results along with an outlook of future work.

Chapter 1

Fundamentals

»All our knowledge has its origins in our perceptions.«

– Leonardo da Vinci

The following sections give a general introduction to the concept of range sensing. It explains the basic idea behind time-of-flight sensing and describes the working principle of recently developed time-of-flight cameras, their advantages over existing techniques as well as common drawbacks. It further introduces the theoretical camera model and mathematical notation used throughout the thesis as well as basic techniques related to camera calibration and motion estimation. Finally, in respect to real-time data processing, an overview about current graphics hardware is given to complete the fundamental background.

1.1 Range Measurement Techniques

Over the years, a various number of range measuring techniques has been introduced, covering contact (mechanical) as well as non-contact techniques. Both techniques basically differ in accuracy and suitability/usability. While contact scanners are most accurate but generally restricted to smaller distances, non-contact approaches generally allow a wider distance range at the expense of possible inaccuracy and computation effort. Due to their domain, most applications in computer vision and remote sensing therefore rely on non-contact techniques, whose most important concepts can be classified into *triangulation* and *time-of-flight*.

Triangulation In the context of triangulation, the location of a surface point is estimated by measuring the angles between the lines of sight from either end of a fixed baseline to the particular surface point. In doing so, the unknown point can be interpreted as the third point of a triangle with one known side and two known angles (cmp. Fig. 1.1).

The classic implementation of triangulation is the approach of stereo vision, which copies the human vision system by using a camera rig of two cameras. The main challenge in stereo vision is the complex task of finding point correspondences within both images. Based on epipolar geometry and image rectification (trans-

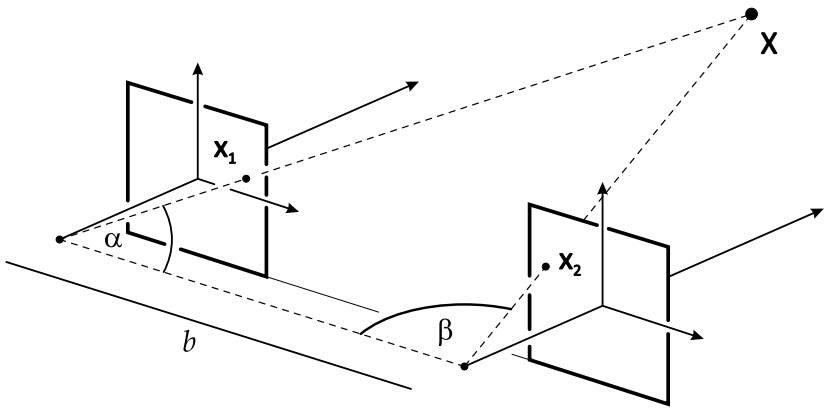


Figure 1.1: Triangulation in stereo vision. The point $\mathbf{X}$ can be reconstructed via the point correspondence $\mathbf{x}_1 \propto \mathbf{x}_2$, i.e. by implicitly estimating both angles α and β, and a given baseline length b [BF82].

formation), the two dimensional problem of finding correspondences is commonly reduced to finding the best match along a given scan line [BF82, Fau93].

Several approaches for the match finding exist that, for example, use intensity correlation [FP86], relaxation techniques [MP79, PMF85], dynamic programming [OK85] or prediction methods [AF87]. In general, all methods exploit constraints that preserve surface continuity (smoothness), match ordering or forbidden zones (where possible matches definitively cannot be found). Problems occur if homogeneous image regions do not provide enough textual information to avoid ambiguities, i.e. to find an unique match.

In contrast to passive stereo vision, active systems like laser or structured light scanners avoid the problematic task of finding point correspondences by replacing one camera by a light source. By doing so, the triangulation angles are well-defined through the exit angle of the emitted light and the column of the detected signal within the image sensor. Problems arises if the light signal is not reflected properly, which is the case if an object is either to far away to get illuminated or its material properties result in over- or under-exposure in the image detector.

In order to obtain range data for an entire scene, active systems either have to scan the scene row-by-row, as it is the case for laser scanners, or have to use temporal pattern sequences to binary encode the corresponding column of the projector and allow parallel detection (structured light). Both methods can be quite time consuming as lasers scanners typically involve mechanical setups to control the laser beam, whereas the processing time of structured light scanners

increases with the number of encoding patterns to project.

Regardless whether the system is active or passive, triangulation techniques have a general disadvantage as wide distance ranges and high accuracy requirements commonly require a large baseline. Thus, for large distances, triangulation might become completely impractical. Furthermore, due to the different viewing positions, occlusion and shadowing effects may occur. In this case, triangulation becomes impossible and yields incomplete range maps.

Time-of-Flight Another common concept to estimate range information is defined by the principle of time-of-flight. Being an adequate alternative to triangulation, time-of-flight (TOF) is already used by a wide range of automatization and sensing applications to estimate the distance to a given surface point. The main idea is described by the run time estimation between the emission of a given optical, electro-magnetic or acoustical signal and its arrival at a provided detector. If the signal's propagation velocity is known, the distance to the surface can be calculated from the time it took for the signal to travel between emitter and detector.

Generally, two types of TOF estimation exist: pulse and continuous modulation. In the context of *pulse modulation*, only a single light impulse is emitted, implying high demands on the detection accuracy in order to determine the exact time delay. *Continues modulation*, on the other hand, estimates the phase shift between the emitted signal and its response. The system is therefore less demanding, but the unambiguously range of the measurement gets limited due to the periodicity of the emitted signal.

Time-of-flight systems are – unlike triangulation setups – less affected by shadowing effects, but are mostly capable to treat only a single point at a time. Thus, in order to provide full range information, classical TOF devices have to mechanical sweep their signal across the scene for either a single row (2D scanner) or line-by-line (3D scanner), which makes classic TOF systems rather time consuming, expensive and/or unhandy to use.

Recently developed systems realize TOF sensing as an closed form, on-chip design. Time-of-flight cameras build on these chips are not only compact and cost-efficient, but also capable to estimate full scene range data in almost real-time. Unlike classical techniques, they do not rely on mechanical setups (like laser scanners) or expensive computations (as in stereo vision), making them very attractive for interactive applications.

At present, TOF cameras are produced by four manufacturers (cmp. Fig. 1.2): PMD Technologies/ifm electronics [PMD], MESA Imaging [MES], Canesta [CAN] and 3DV Systems [3DV].

While 3DV Systems offers the only TOF camera based on pulse modulation and realize a special on chip shutter technique to detect the light signal delay, all other manufacturers use continues intensity modulation and exploit the same working

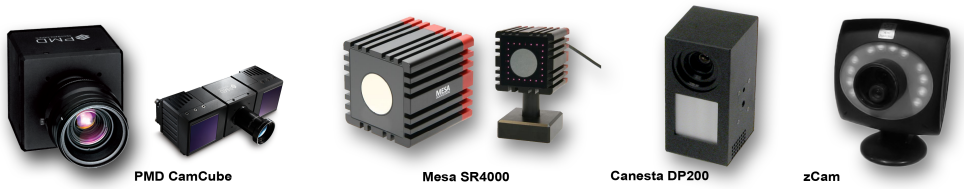


Figure 1.2: Current TOF camera models: PMD CamCube (204 × 204px), Mesa SR400 (176 × 144px), Canesta DP200 (64 × 64px) and zCam (320 × 240px).

principle, which is described next at the example of current PMD cameras.

1.2 Photonic Mixer Device

In the following, the working principle of continuous intensity modulation based TOF cameras is explained for current Photonic Mixer Devices (PMD). In this context, also a brief discussion of characteristical error sources for TOF cameras is given.

1.2.1 Working Principle

According to the time-of-flight principle, Photonic Mixer Devices (PMD) consist of two main components: a CMOS senor, which represents the detector for the time-of-flight measurement, as well as one or more active illumination units, emitting the optical signal to detect. Other than for classical systems like laser scanners or interferometry systems, the light source of TOF cameras does not have to be coherent or monochromatic. Therefore, during the measurement process, the entire scene gets illuminated with incoherent, intensity modulated near-infrared light (NIR), which is reflected at the observed surfaces and finally detected by the photo gates of the corresponding sensor pixel (cmp. Fig. 1.3).

Given the internal modulation signal s and its detected response r, each sensor pixel autonomously determines the corresponding cross correlation $c(\tau)$ for a fix internal phase delay τ, i.e.

$$c(\tau) = \lim_{T\to\infty} \int_{-T/2}^{T/2} r(t) \cdot s(t+\tau)\, \mathrm{dt}. \tag{1.1}$$

Assuming a sinusoidal signals [XSH*98, Lan00, KFM*04], i.e. $s(t) = \cos(\omega t)$ and $r(t) = k + a\cos(\omega t - \phi)$, the cross correlation sample is given by

$$c(\tau) = \frac{a}{2}\cos(\omega\tau + \phi) \tag{1.2}$$

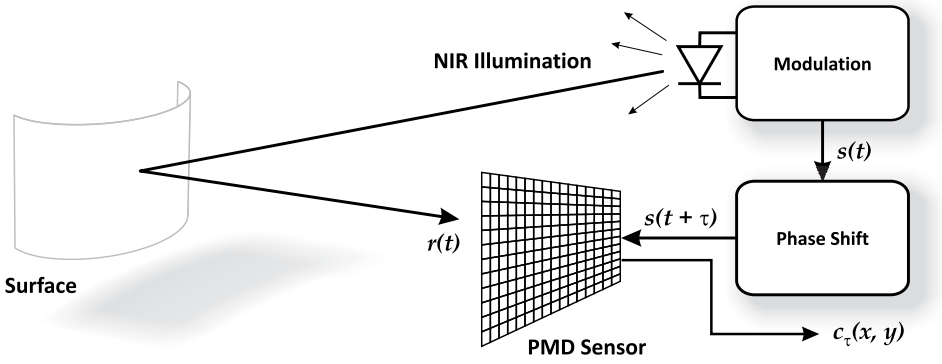


Figure 1.3: The principle of continues modulation based TOF cameras.

where $\omega = 2\pi f$ represents the angular modulation frequency, a is the amplitude of the incident optical signal and ϕ is the phase offset related to the object distance.

By sampling the correlation function, i.e. taking four subsequent phase images $I_i = c(\tau_i)$ with an internal phase delay of $\tau_i = i \cdot \pi/2\omega$, a pixel's distance-related phase shift $\phi \in [0, 2\pi]$ can be determined as[1]

$$\phi = \operatorname{atan2}(I_3 - I_1, I_0 - I_2) + \pi. \tag{1.3}$$

Consequently, the demodulation scheme assumes that a surface point is captured at the same pixel location in all phase images. Using the speed of light $c_0 \approx 3{\cdot}10^8\,\mathrm{m/s}$, the distance information is finally given by d

$$d = \frac{c_0}{2\omega} \cdot \phi, \tag{1.4}$$

where the factor $1/2$ is given by the fact that the light travels twice the distance between camera and surface point. Due to a commonly used modulation frequency of 20 MHz, distance information is typically clamped to an unambiguous distance range of 7.5 m.

Beside the distance information, most TOF cameras provide an additional intensity value h, which is comparable to a gray level image, as well as the signal's correlation amplitude a, giving information about the distance reliability. Both informations are obtained from the correlations samples via

$$h = \frac{1}{4}\sum_{i=0}^{3} I_i \quad \text{and} \quad a = \frac{1}{2}\sqrt{(I_3 - I_1)^2 + (I_0 - I_2)^2}. \tag{1.5}$$

[1] $\operatorname{atan2}(y, x)$, $(y, x) \mapsto [-\pi, \pi]$, computes the angle between the positive x-axis of a plane and the point (x, y).

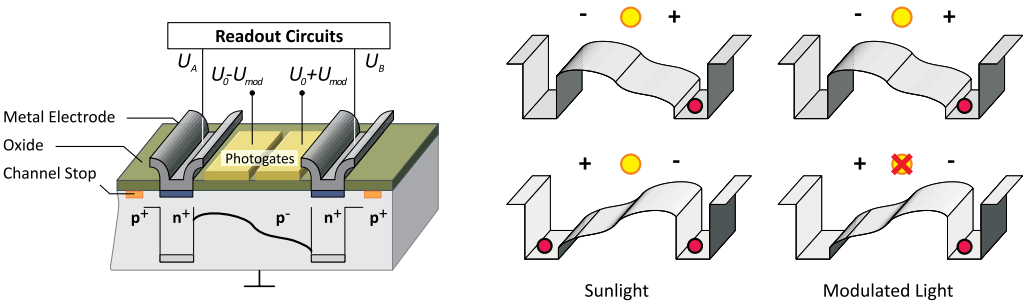


Figure 1.4: Simplified PMD pixel design (left) and constant light versus modulated light (right) [MKF*05].

Analog results can be derived by signal theory based on Fourier transformation.

The actual hardware implementation of PMD sensors is realized based on the CCD-principle as CMOS active pixel sensor [Lan00]. Each 2-tap sensor element (semiconductor) consist of two light conductive and transparent modulation electrodes as well as two readout diodes (commonly revered to as A and B) on the left and right side (see Fig. 1.4). Due to the modulation gates, the movement of generated charge carriers inside the semiconductor's substrate layer can be controlled by a reference signal to either the left or right side of the sensor element.

Assuming a rectangular reference signal, the generated charge carriers for constant incident light move to the left and to the right equally, whereas for modulated light all charge carriers will be moved to one of both readout diodes in cases where no phase delay between the modulation of the incident light and the detector is present. Other phase delays, however, will lead to a difference between the two output voltages of the readout diodes corresponding to the correlation of both signals (see Fig. 1.5). To be more precisely, both output voltages can be interpreted as contrary correlation samples $c(\tau_i)$ and $c(\tau_{i+2})$, which theoretically reduces the number of necessary sample images by two. However, most PMD cameras acquire the full quantity of phase images for $\tau_0 - \tau_3$, which is basically motivated by the fact that possible offset-voltages a_0 and b_0 for individually amplified phase samples

$$A_i = a_1 \cdot c(\tau_i) + a_0 \quad \text{and} \quad B_i = b_1 \cdot c(\tau_{i+2}) + b_0 \tag{1.6}$$

get canceled during the demodulation as can be seen by substituting $I_i = A_i - B_i$ into (1.3). However, individual amplification terms a_1 and b_1 remain. In this context, the demodulation as described in (1.3) has been shown to be insensitive to linear or quadratic distortions of the gait linearity [Lan00].

For outdoor applications, unfortunately, a huge amount of the sensor dynamics is occupied by uncorrelated sunlight. In order to avoid saturation effects due to excessive background light, latest PMD cameras exhibit a special suppression of background illumination (SBI) circuit that additionally adjusts the charge level in

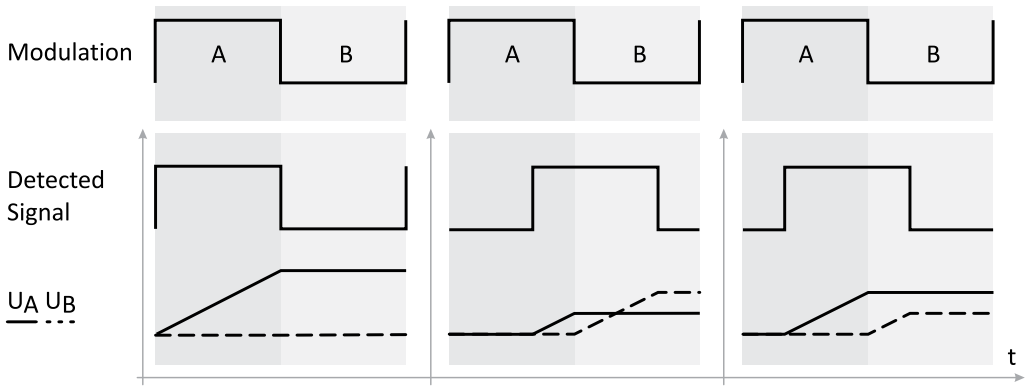


Figure 1.5: PMD output voltages ($U_A,\ U_B$) assuming a rectangular modulation.

both readout diodes in order to instantaneously reduce the unwanted background component. By doing so, the dynamic range of the PMD device can be essentially improved for correlated signal information. However, the scene's true intensity information ($A + B$) is significantly altered and therefore can not be recovered. An alternative NIR gray level image can be retrieved by considering the amplitude information, as both signals correlate and therefore are mostly interchangeable.

Current camera models already provide an enhanced resolution of 204×204 px. Basis for this thesis, however, has been a 19k PMD prototype without SBI circuit that provides a 160×120 px resolution.

1.2.2 Error Characteristics

Like almost every sensing device, the PMD as well is affected by several error sources that influence the accuracy of measured distance information. Basically, errors can be grouped into general TOF- as well as implementation-specific impacts.

Signal Quality Due to the underlying time-of-flight principle, accurate distance information highly depends on a correct detection of the emitted signal. For this reason, TOF cameras are generally quite sensitive to external influences affecting the emitted signal either in shape or intensity. Especially the IR-reflectivity of observed surfaces has strong impact on the estimation result and often leads to under- or overexposed pixels. While underexposed pixels suffer from bad signal-to-noise ratios, i.e. noise, overexposed pixels are not capable to provide any distance information at all. Both are generally detectable via the amplitude information. Over-exposed pixel, however, sometimes provide a misleading good amplitude value. A more reliable technique to detect oversaturation therefore is to check the individual phase samples I_i for saturation [Rap07].

Superposition Signal-related errors may also occur due to interferences with other existing NIR light sources as well as multiple reflections within the scene, e.g. within corners [GAL07]. In both cases, superimposed signals affect the phase demodulation, yielding falsified distance information. A special case of superposition is basically related to the solid angle of a PMD pixel. Analog to multi-reflections, distance jumps inside a solid angle result in superimposed signals leading to a false phase estimate, often referred to as *flying pixel*. Usually, the distance information of flying pixels lie in-between fore- and background (see Fig. 1.6), but can also tend towards the camera depending on the surface's true distance [KK09]. A simple segmentation of flying pixels using a pixel's amplitude is not possible, for which reason more sophisticated processing techniques are necessary, that for example consider a pixel's neighborhood relation.

Noise A general problem in the context of range sensing, especially for under-exposed pixel, is given by *noise* affecting the accuracy of the measured distance information. With respect to the underlying CMOS design, PMD-related noise can be generally classified into three categories [Lan00]: time variant and time invariant noise as well as signal noise.

Time Variant Noise covers thermal noise, reset noise, $1/f$ noise and dark current shot noise. All these error sources are signal independent and increase with rising temperature. Time variant noise can be significantly reduced or eliminated by proper cooling and signal processing techniques like correlated double sampling (CDS) [Tem96].

Time Invariant Noise, i.e. constant *fixed pattern noise*, can be classified into defect pixel (noticeable as static white or black pixel) and leaker (which are significant brighter than the neighborhood) as well as varying pixel offsets due to variations in oxide thickness, size of gate area and doping concentrations over the sensor.

Fixed pattern noise, including defect pixels, can be determined by taking a *black image*, i.e. keeping the optics shut while averaging over an sufficient amount of images. Beside techniques like CDS, time invariant noise can be mostly reduced by subtracting the black image from the camera output. Note that in the case of current PMD sensors, where each PMD pixel subtracts the number of generated electrons from an initial budget, an inverse subtraction has to be performed, i.e. the raw image has to be subtracted from the black image.

Signal Noise, emerging from photon shot noise, is the most dominant noise and has a great impact onto the effective signal-to-noise ratio. It cannot be suppressed and (more significantly) increases with the amount of incoming photons.

Shot noise is commonly modeled by Poisson-distributed arrival processes of independent events with occurrence rate λ, where $Var(X) = E(X) = \lambda$. However, for a high number of accumulated charges (which is the case for reasonable exposure times), the raw values distribution of both readout diodes A and B can be sufficiently approximated by a normal distribution. As a result, the

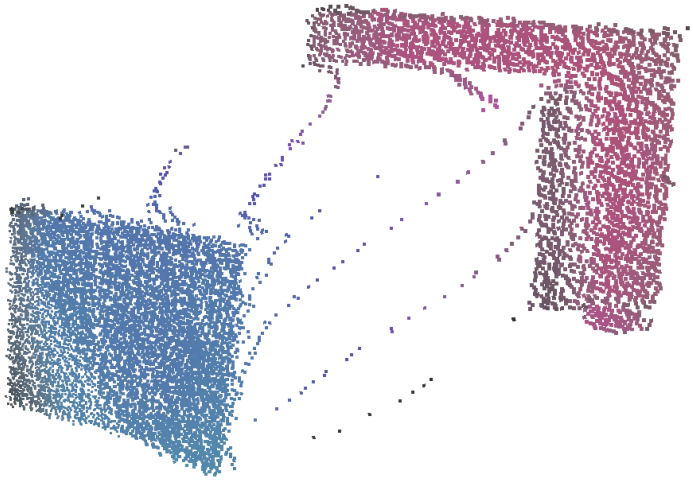


Figure 1.6: Flying pixel effect along object contours due to a superimposed signal, i.e. multiple distances, inside a single PMD pixel.

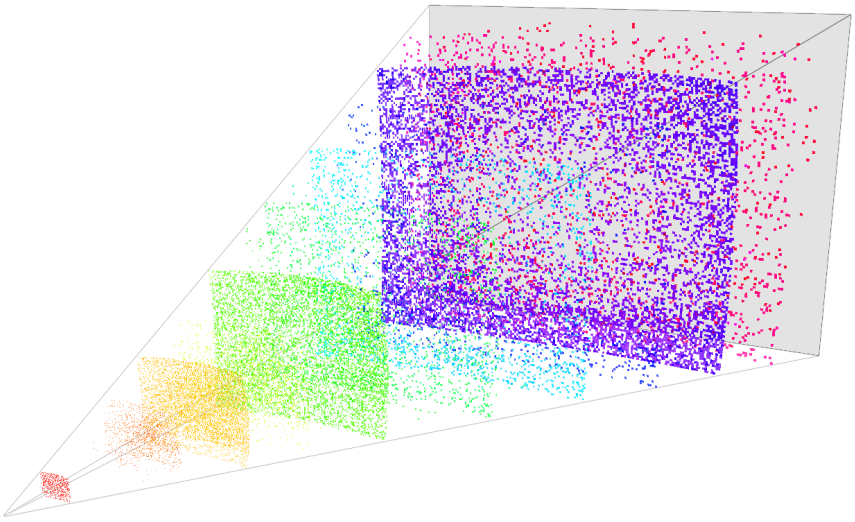


Figure 1.7: Quantization effects for low signal amplitudes. Here, a planar surface is captured with a short integration time.

standard deviation of distance information was found to be reciprocal to the modulation signal's amplitude a, i.e. $Var(\phi) \propto 1/a^2$ [FPR*09]. The signals amplitude therefore is not only a measure for the distance reliability, but can also be considered for noise reduction as discussed more in detail in Sec. 3.1.

Quantization Effects During the acquisition of subsequent phase images, an analog-digital conversion maps all phase samples I_n to integral values before (1.3), p. 9, is applied. Theoretical investigations and experiments by Frank et al. [FPR*09] show, that the corresponding quantization has a great impact on the phase demodulation for low amplitudes, i.e. large distances, and results in a sparse distribution of distance information as depicted in Fig. 1.7.

Systematic Errors

Beside their general dependencies to the signal quality, current PMD cameras are also affected by system specific error sources that additionally influence the distance accuracy in a negative way. In the following, a first overview about systematic errors will be given, that mainly describes their characteristics as well as their origins. In Chapter 2, a calibration model for each error source will be discussed as part of this thesis' contribution.

Demodulation Error Concerning modulation-based TOF cameras, a characteristical error is caused by the underlying demodulation scheme and its basic assumption of sinusoidal signals, which in practice is not met due to hardware and cost limitations. Current PMD cameras, for example, actually use a rectangular internal modulation which is altered by the LED's response time to a mixture between a sinusoidal and a rectangular shape. The result is a systematic wiggling that significantly alters the measured distance by shifting the distance information either towards or away from the camera depending on the surface's true distance (see Fig. 1.8). The exact shape depends on the additional harmonics included in the modulated signal, which is mainly influenced by the current-voltage characteristics of the LEDs used for illumination. However, only odd harmonics are proven to have a negative influence on the demodulation scheme stated in (1.3) on page 9 [Lan00].

In order to avoid wiggling artifacts, a more accurate demodulation scheme for non-harmonic signals has been frequently discussed [FPR*09, Rap07, Lua01]. Here, the main idea is based on a more precise representation of the correlation function that incorporates higher Fourier modes. Actually, by modeling the correlation function via a finite sum of superimposed cosine waves

$$c(\tau) = \sum_{k=0}^{l} c_k \cos(k(\omega\tau + \phi) + \theta_k),$$

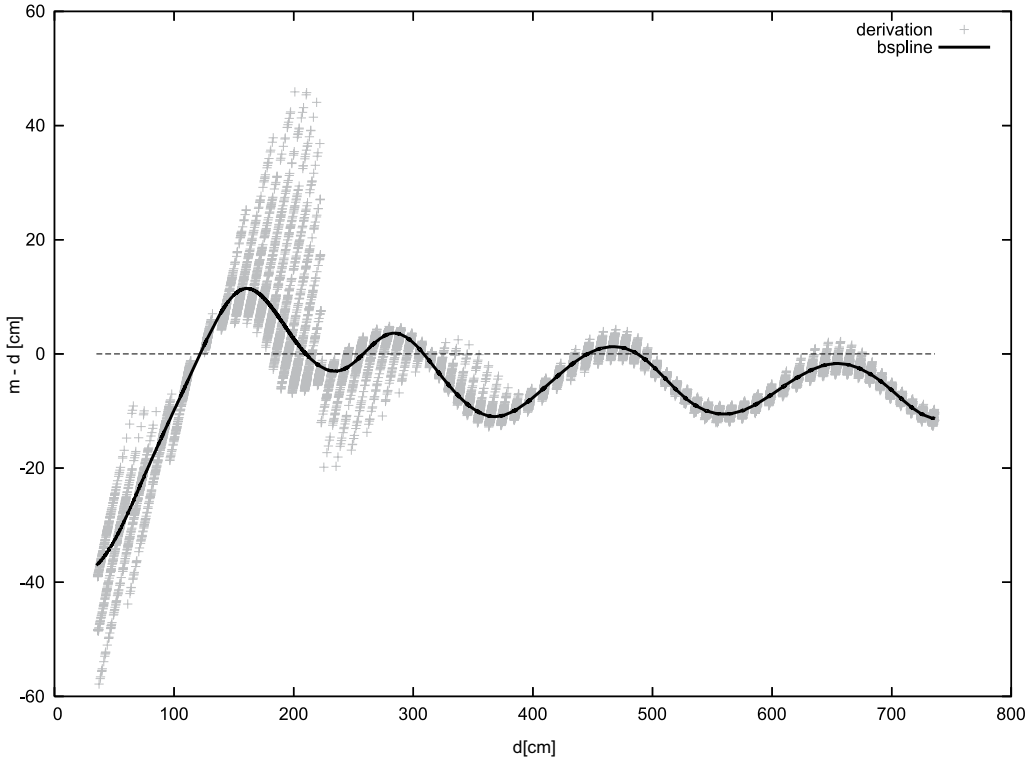


Figure 1.8: Systematic wiggling error for distance measurements between $1.0 - 7.5\,m$. A fitted error function is shown in black. Extreme outliers are explained by oversaturation in the near range.

a least square optimization over $N \geq 2l + 1$ samples leads to following phase demodulation scheme:

$$k\phi + \theta_k = \arg\left(\sum_{n=0}^{N-1} I_n e^{-2\pi i k \frac{n}{N}}\right)$$

where $I_n = c(\frac{2\pi}{\omega} \cdot \frac{n}{N})$. The distance-related phase-shift ϕ can be finally obtained by using a look-up table (LUT) for the fixed offsets θ_k of the additional modes.

However, extending the demodulation scheme for non-harmonically signals is rather impracticable as the number of required sample images I_n as well as the calculation effort for the demodulation increases. Especially the higher number of samples leads to further problems in respect to dynamic scenes and related motion artifacts (see below). For this reason, the simpler sinusoidal-based demodulation scheme is commonly used further on, which requires falsified distance information to be adjusted as described in Sec. 2.2.2.

Integration Time Depended Error The previously described demodulation error is additionally shifted by a constant, monotone increasing offset according to the integration time.

Intensity-Related Error Beside the systematic wiggling error, distance information is additionally altered by an intensity-related error inducing non-linear distance shifts (see Fig 1.9). The reason for the intensity-related false measurement is still unknown, but considered to be related to the semiconductor and the camera electronics. So far only a few phenomenological calibration approaches exist, which will be discussed further in Sec. 2.2.4.

Motion Artifacts Motion Artifacts as noticeable in Fig. 1.10 typically occur where objects or the camera itself moves, while consecutive phase images are taken. They arise from unmatching phase values during the demodulation process in cases where the sampling assumption is not satisfied (cmp. (1.3), p. 9). Motion artifacts are more extensive the faster the object moves or the longer the integration time is. In general, it can be distinguished between three error sources:

Lateral Motion which primary results in the mixture of foreground and background phase values at the boundary of moving objects.

Axial Motion which describes motion along the viewing direction and introduces additional phase changes due to non-constant object distance.

Texture Changes which occur for objects of varying reflectivity and result in unmatching phase values, even if the object distance does not change for a given pixel.

A theoretical investigation of discontinuity and texture-related motion artifacts has been published by Schmidt [Sch08]. He assumes that both even as well as odd correlation samples are taken at the same time, i.e. are related to the same reflectivity, and describes the theoretical impact of varying intensity onto the resultant distance information. A concret compensation model will be described in Sec. 2.3 as part of the thesis' contribution.

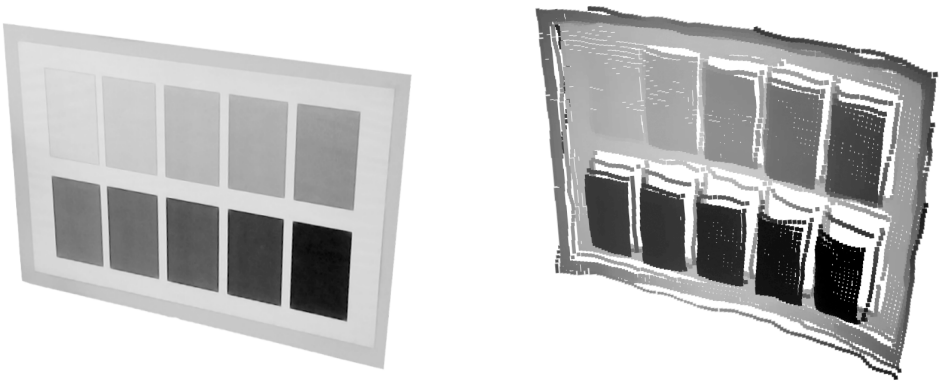


Figure 1.9: Intensity-related distance deviations due to varying object reflectivity, i.e. active light incident to the sensor. Original panel (left) and PMD image (right).

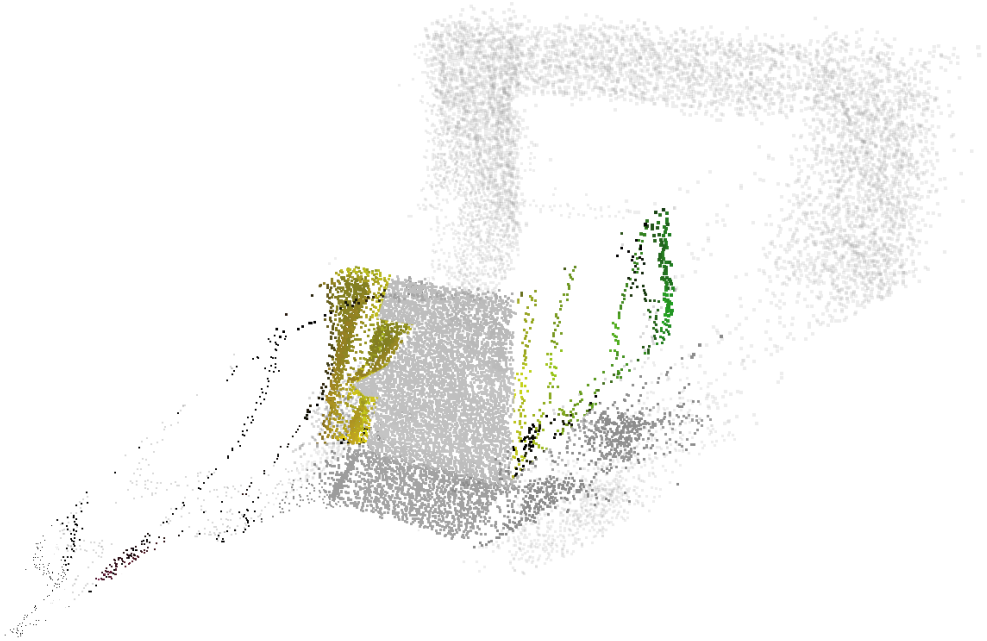


Figure 1.10: Motion artifacts (colored regions) for a textured box that moves from right to left in front of a wall.

1.3 Perspective Camera Model

As the scope of this work focuses on the calibration and processing of range images, the following sections will describe how 3D points and their according points on the image plane are basically related. Camera specific parameters will be introduced, which are part of the mapping process and thus are related to the correct back project of image points into 3D space.

The most common camera model in computer graphics and computer vision is the simple *pinhole model* [MSKS04] in which light, reflected from object surfaces, is passing through a tiny hole (the optical center) forming an image representation on a given image plane (see Fig. 1.11). The distance between the optical center and the image plane is commonly referred to as the *focal length* of the pinhole camera. The point where the perpendicular viewing ray (optical axis) intersects the image plane is referred to as *principal point*.

Following the theorem on intersecting lines, the projection of a 3D point with $\mathbf{X} = (X, Y, Z)$ onto the image plane is described in homogeneous coordinates by

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \Leftrightarrow \mathbf{x} = \Pi_f \mathbf{X} \tag{1.7}$$

yielding the image coordinate $\mathbf{x} = (x, y)$ for a given focal length f and a commonly unknown positive scalar $\lambda \in \mathbb{R}_+$ (the vertices distance value Z). By introducing a global *world coordinate frame* and incorporating the camera position and orientation into the mapping process, (1.7) extends to

$$\mathbf{x} = \Pi_f \begin{bmatrix} \mathrm{R} & \mathrm{t} \\ 0 & 1 \end{bmatrix} \mathbf{X} \; = \Pi_f \mathrm{M} \mathbf{X} \tag{1.8}$$

where $\mathrm{R} \in \mathbb{R}^{3\times3}$ represents the corresponding camera rotation and $\mathrm{t} \in \mathbb{R}^3$ the translation vector between the camera position and the origin of the world coordinate frame.

1.3.1 Intrinsic Parameters

According to (1.8), resulting image coordinates are still specified in units of the world coordinate frame relative to the principal point. Images, however, are generally specified as pixel arrays with their origin in the upper-left corner. Thus, an additional mapping is required that reflects camera specific characteristics, i.e. the pixel size $\mathbf{s} = (s_x, s_y)$ as well as the pixel position of the *principal point* $\mathbf{c} = (c_x, c_y)$ relative to the upper-left corner.

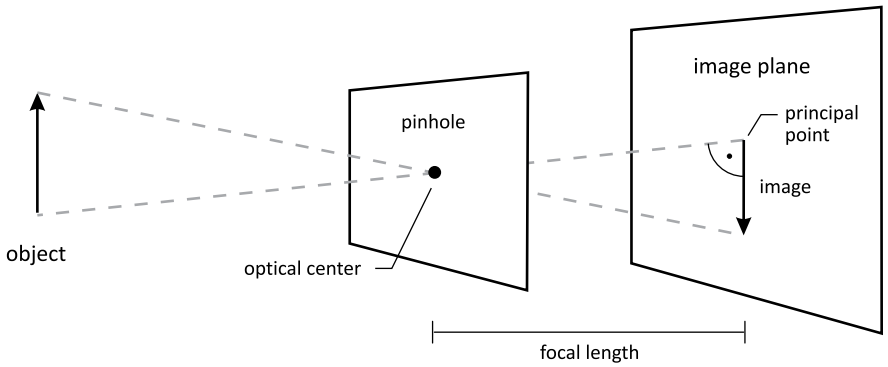


Figure 1.11: Image formation in a pinhole camera.

By defining the standard perspective projection as $\Pi_1 = \begin{bmatrix} \mathrm{I}, & 0 \end{bmatrix}^T$ and splitting the projection matrix Π_f into $\Pi_f = \mathrm{K}_f \Pi_1$, we are able to combine all camera-related parameters (including the focal length) into the *intrinsic parameter matrix*

$$\mathrm{K} = \mathrm{K}_s \mathrm{K}_f = \begin{bmatrix} s_x^{-1} & 0 & c_x \\ 0 & s_y^{-1} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1.9}$$

where f_x and f_y correspond to the focal length in pixel units along the x- or y-axis. Consequently, (1.8) can be extended to

$$\mathbf{p} = \mathrm{K}\Pi_1 \mathrm{M}\mathbf{X} = \mathrm{K}\mathbf{x} \tag{1.10}$$

yielding the final pixel coordinate $\mathbf{p} = (u, v)$ in pixel units. The intrinsic matrix K is commonly obtained through the process of camera calibration as described in Sec. 1.3.3.

1.3.2 Image Distortion

In practice, cameras differ from the pinhole model insofar as they use lenses to gather more light on the image plane. Thus, light rays do not pass straight through a tiny hole, but get bended and focused. Rays farther from the center of the lens thereby are bent more than those closer to the center.

Due to imperfections regarding today's lens systems, however, images taken by real cameras are often affected by nonlinear aberrations. These distortions are mainly characterized by a symmetric displacement along the radial direction from the principal point. Depending on the direction, the radial distortion causes either

an inward or outward displacement of image points (see Fig 1.12 left). Especially for cameras with a large field of view, the influence of the radial distortion becomes strongly noticeable. Thus, for computer vision systems, the correction of lens distortion is significant for accurate scene reconstruction.

In order to overcome limitations of the pinhole model, more sophisticated camera models like the *thin lens* or *thick lens* model exist, but mostly account to simulate depth of field effects [BW99]. Like the pinhole model, both yield perfectly undistorted image. Geometric lens models regarding the complete geometric description like the one published by Heidrich et al. [HSS97], on the other hand, are too complex to be used for interactive applications.

For most applications, therefore, the pinhole model is commonly extended by an additional distortion model, which is mainly based on the polynomial formulation

$$r_d = r \cdot \delta(r) = r \cdot \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots\right) \tag{1.11}$$

first introduced in the context of photogrammetry by Slama in 1980 [Sla80]. Here, $r = |\mathbf{x}|$ equals the undistorted distance between the principal point and the normalized coordinate $\mathbf{x} = \mathrm{K}^{-1}\mathbf{p}$. As (1.11) is basically dominated by the first term, radial distortion is commonly modeled by

$$\mathbf{x}_d = \mathbf{x} \cdot \left(1 + k_1 r^2 + k_2 r^4\right) = \mathbf{x} + L(\mathbf{x}, \mathbf{k}). \tag{1.12}$$

to avoid numerical instability [Zha00, Tsa87, WM94]. According to (1.10), the equivalent formulation in pixel coordinates is given by

$$\mathbf{p}_d = \mathbf{p} + (\mathbf{p} - \mathbf{c}) \cdot \left(k_1 r^2 + k_2 r^4\right) \tag{1.13}$$

where $r = |\mathbf{p} - \mathbf{c}|$.

To overcome the drawback of a missing analytical inverse, other models have been introduced, which are e.g. based on Taylor expansion or rational formulations [MCM04]. Nevertheless, polynomial models are still the most commonly used due to their accuracy and physical background. If necessary, the inverse of (1.12) can be obtained either by an iterative numerical scheme or recursively approximated by

$$\begin{aligned} \mathbf{x} &\approx \mathbf{x}_d - L(\mathbf{x}, \mathbf{k}) \approx \mathbf{x}_d - L(\mathbf{x}_d - L(\mathbf{x}, \mathbf{k}), \mathbf{k}) \\ &\approx \mathbf{x}_d - L(\mathbf{x}_d - L(\mathbf{x}_d - L(\mathbf{x}, \mathbf{k}), \mathbf{k}), \mathbf{k}) \approx \ldots \end{aligned} \tag{1.14}$$

where, in the first recursion step, $\mathbf{x}$ is approximated by $\mathbf{x}_d$ [Hei00]. Here, the distortion coefficients are typically small causing the model to be almost linear. Accordingly, the divergence between $\mathbf{x}$ and $\mathbf{x}_d$ in (1.14) gets smaller with every iteration. For strong lens distortions, at least three or four iterations are required.

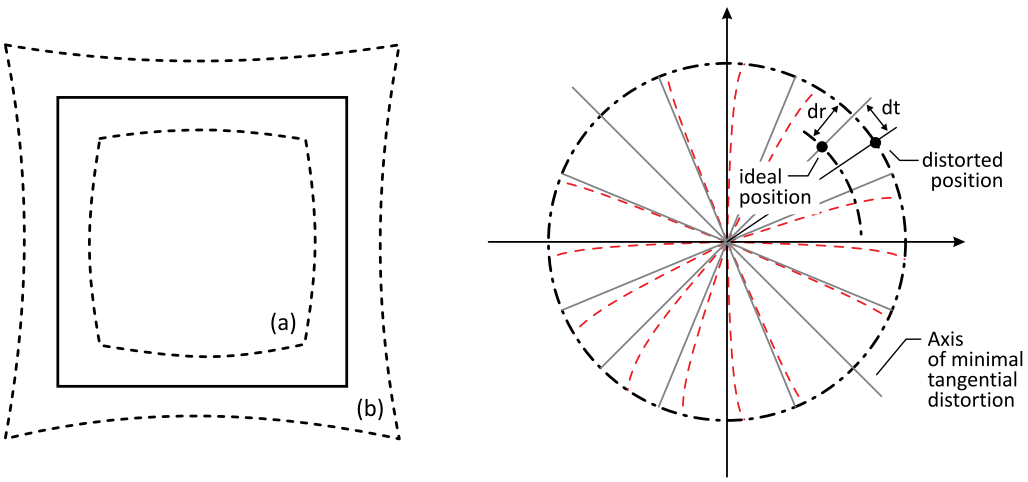


Figure 1.12: *Left:* Effect of radial distortions: undistorted view (solid line), negative displacement / barrel distortion (a) and positive displacement / pincushion distortion (b). *Right:* Effect of tangential distortion (dashed lines) consisting of radial (dr) and tangential (dt) displacements [WCH92].

In addition to radial distortion, optical systems often are affected by an asymmetric *tangential distortions* caused by decentering problems within multi-lens systems [Bro66]. The result is a supplemental geometric shift along and tangential to the radial direction (see Fig 1.12 right). In respect to (1.12), tangential distortion is commonly modeled by

$$\mathbf{x}_d = \mathbf{x} + L(\mathbf{x}, \mathbf{k}) + \left[2t_1xy + t_2\left(r^2 + 2x^2\right),\quad 2t_2xy + t_1\left(r^2 + 2y^2\right)\right]^T. \qquad (1.15)$$

In this case, the distortion coefficients $\mathbf{k} = (k_1, k_2)$ and $\mathbf{t} = (t_1, t_2)$ define the tangential profile along the axis of maximum tangential distortion.

1.3.3 Camera Calibration

Once the intrinsic parameters and distortion coefficients are known, 3D information can be inferred from 2D information and vice versa. Therefore, the calibration of intrinsic and extrinsic parameter is a necessary step in computer vision, which has great influence on the accuracy of the results.

Beside a few exceptions, camera calibration can be grouped into two categories: photogrammetric calibration and self-calibration.

For *photogrammetric calibration*, calibration is performed by observing a calibration object (rig) whose 3D geometry is a priori known – usually two or three planes orthogonal to each other [Fau93]. This way, the desired mapping parameters can be estimated by taking advantage of the distinctive correspondences between

predefined surface points and their counterparts on the image plane. Commonly checkerboard or circular patterns are used to provide easy detectable feature points.

For *self-calibration* no special calibration objects are used. Just by moving a camera in a static scene, the rigidity of the scene provides the correspondences necessary for intrinsic calibration [FLM92]. The concept behind self-calibration is basically motivated by the idea to overcome limitations of special calibration rigs and, meanwhile, to allow varying intrinsics of zooming/focusing cameras. Unlike photogrammetric methods, it cannot be applied as a single step, but has to be part of the image processing. Therefore, self-calibration always requires a complete series of images, taken from different viewpoints.

In our case, since current TOF cameras have a fixed focal length, i.e. have no auto-focus or zoom functionality, the simpler photogrammetric approach has been considered for calibration. The decision is based on the complexity of finding precise feature points in the low resolution TOF image as well as the fact that TOF cameras are often used to observe a steady scene providing an insufficient amount of camera poses for self-calibration.

A very popular approach for photogrammetric calibration has been published by Zhang [Zha00]. Its popularity is mainly founded on the fact, that it is based on a simple plane instead of a complex, three-dimensional calibration rig. It thus reduces the calibration requirements onto a minimum. An open-source implementation can be found in the Open Computer Vision Library (OpenCV) [OCV].

Zhang first approximates an initial guess for K and M by estimating the homography H between known 2D and 3D points, i.e.

$$\mathbf{p} = \mathrm{H}\mathbf{X} = \mathrm{K}\Pi_1\mathrm{M}\mathbf{X} \tag{1.16}$$

disregarding any distortion effects at first (cmp. (1.10), p.19). Assuming $Z = 0$, (1.16) can be simplified and provides linear constrains for an initial closed-form solution. After the mapping parameter has been determined, an initial solution for the radial and tangential distortion can be estimated, which together with the first results on intrinsic parameters serve as an initial guess for an overall, non-linear optimization.

Theoretically three views of a plane checkerboard of varying orientation (that avoid pure translation) are sufficient to estimate the camera's intrinsic parameters. However, due to noise, at least 10 images of a 7-by-8 checkerboard are recommend for a robustness estimation.

1.3.4 Undistortion and Back-Projection

In contrast to standard 2D imaging devices, where the mapping parameter $\lambda = Z$ of (1.7) on page 18 is generally unknown, TOF cameras allow a simple inversion

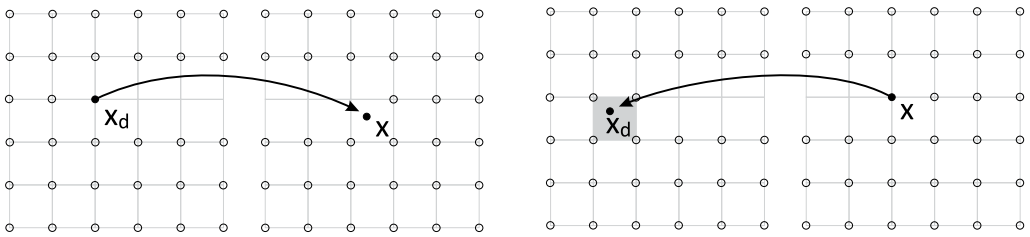


Figure 1.13: Forward (left) vs. Backward Mapping (right). While forward mapping results in a non-uniform data alignment, backward mapping requires data interpolation to obtain subpixel information.

of the perspective mapping process if the intrinsic parameters as well as the distortion coefficients are known. According to the inversion of the mapping process, the lens distortion has to be treated first.

Here, generally two types of image undistortion or image transformations can be considered, commonly referred to as *forward* and *backward mapping* (cmp. Fig. 1.13).

During *forward mapping*, given image data is relocated to new (undistorted) positions defined by the inverse of (1.15). Forward mapping therefore breaks up the original image structure and results in a non-uniform alignment of unmodified pixel values, i.e. an unstructured point cloud. This kind of undistortion has the advantage, that no interpolation is involved as the existing image data is only rearranged. However, if a regular image structure is desired, forward mapping leads to pixel snapping, i.e. multiple values per pixel and / or holes.

Backward mapping, in contrast, maintains the original data alignment by looking up corresponding image information in the distorted image, i.e. for each pixel of the undistorted image, the corresponding image coordinate in the distorted image is determined with respect to (1.15). As distorted sample positions most properly do not align with image pixels, according subpixel information is finally obtained by e.g. bilinear interpolation.

Due to the lack of an analytical inverse and in order to preserve the structure of uniformly sampled image data, the undistortion of TOF images is commonly done via backward mapping. Once the distortion coefficients are known, a distortion map can be precomputed to increase the overall performance. For vision systems that use more general spatial data structures to represent the scene, forward mapping may give the better alternative as no data interpolation is involved.

After lens distortion has been corrected, the corresponding 3D information can be reconstructed from the given depth image by projecting each pixel back into space. Again, by following the theorem of intersecting lines, the world coordinate for a given pixel $\mathbf{p} = (u, v)$ holding a radial distance information d can be obtain

by

$$\mathbf{X} = \frac{d}{|\mathbf{x}|} \cdot \mathbf{x} \tag{1.17}$$

where $\mathbf{x} = \mathrm{K}^{-1}\mathbf{p}$ represents the pixel's world coordinate on the normalized image plane with $f = 1$ and $|\mathbf{x}| = \sqrt{x^2 + y^2 + 1}$ equals the according radial distance to the projection center.

1.4 Optical Flow Estimation

Motion artifacts as described in Sec. 1.2.2 directly result from either object or camera motion during the acquisition of subsequent phase images. In consequence, many artifacts can be reduced by compensating for object displacements and deformations if the scene dynamics are known (see Sec. 2.3). However, for most scenes a detailed motion information is unfortunately unknown and therefore has to be estimated during runtime.

The estimation of motion from image sequences is a long studied key problem in computer vision. Due to the lack of depth information, motion estimation in computer vision classically focuses on the estimation of the displacement field between two image frames, i.e. the projected, two-dimensional path on the image plane. The estimation of such a motion fields is commonly referred to as *optical flow estimation*.

Unfortunately, motion estimation is a highly ill-posed problem and various approaches have been proposed differing in accuracy and time-complexity – most of them inapplicable for real-time tasks. Despite phase- or correlation-based methods [BB95], *variational methods* are still the most popular estimation techniques yielding the best results in terms of error measures [BFB94].

Neglecting principle problems like transparency, occlusion and shadowing, the basic principle behind all variational methods is based on the minimization of an energy functional that incorporates invariant image features, i.e. features that stay constant over time and are unaffected by the motion itself. The most prominent example is the classical assumption of constant grey values, i.e.

$$I(x + \delta_x, y + \delta_y, t + 1) - I(x, y, t) = 0 \tag{1.18}$$

where $\delta_y = (\delta_x, \delta_y)$ states the displacement at image position $\mathbf{x} = (x, y)$ between time steps the t an d $t + 1$. Other variants incorporate higher image derivatives like the gradient or Hessian, scalar values such as the norm of the gradient, the Laplacian or the determinant of the Hessian, as well as combination of those features [PBB*06].

By applying a Taylor extension to (1.18) and dropping the high order parts to get a linear system, i.e. convex problem, we obtain the well known optical flow

constraint

$$I_x\delta_x + I_y\delta_y + I_t = 0 \tag{1.19}$$

as formulated in the classical algorithms of Horn and Schunck [HS81] and Lucas and Kanade [LK81], where subscripts denote partial derivatives and $\delta_x(x, y, t)$ and $\delta_y(x, y, t)$ represents the unknown displacement field.

Unfortunately, (1.19) yields a single local constraint in two unknowns and thus yields no unique solution. This problem of motion ambiguity, the so-called *aperture problem*, is known in many disciplines and also applies to the human vision system. To overcome ambiguities, all optical flow approaches typically incorporate some kind of regularization, i.e. structural information, to derive additional constraints either locally or globally.

In the classical approach of Lukas and Kanade [LK81] regularization is based on the assumption of a constant flow inside a small neighborhood around the considered pixel. This way not only a sufficient number of constraints are given for a least square optimization, the flow estimation is also less vulnerable to noise. However, due to the local estimation and missing structural information inside homogeneous regions, local techniques are only capable to retrieve motion information at intensity boundaries. Furthermore, they tend to give poor results in presence of multiple motions violating the local constancy assumption.

For this reason, most flow estimation approaches nowadays apply a global regularization as introduced by the classical approach of Horn and Schunck [HS81]. Originally, they incorporate a homogeneous smoothness of the motion field by adding the regularization functional

$$E_{\text{reg}} = \int_\Omega |\nabla\delta_x|^2 + |\nabla\delta_y|^2 \, dx\, dy \tag{1.20}$$

This regularization term, however, has the disadvantage that it does not account for discontinuities in the motion field. Thus, in the last decades more sophisticated approaches has been introduced that can be classified into four basic strategies [BWKS06]:

– Image-Driven, Isotropic Regularization

$$E_{\text{reg}} = \int_\Omega \omega\left(|\nabla I|^2\right)\left(|\nabla\delta_x|^2 + |\nabla\delta_y|^2\right) \, dx\, dy \tag{1.21}$$

  downweightes the smoothness term at image location where intensity changes are high, assuming object contours to be coincident with changes in brightness.

– Image-Driven, Anisotropic Regularization

$$E_{\text{reg}} = \int_\Omega \nabla\delta_x^T D(\nabla I)\nabla\delta_x + \nabla\delta_y^T D(\nabla I)\nabla\delta_y \, dx\, dy \tag{1.22}$$

smooths only orthogonally along the local image gradient, i.e. $D(\nabla I)$ is an anisotropic projection matrix perpendicular to the image gradient.

– Flow-Driven, Isotropic Regularization

$$E_{\text{reg}} = \int_{\Omega} \Psi_S \left(|\nabla \delta_x|^2 + |\nabla \delta_y|^2 \right) \, dx \, dy \tag{1.23}$$

with $\Psi_S(x^2)$ is a positive increasing function with the property to increase less severely than a quadratic function, i.e. $\Psi(x^2) = \sqrt{x^2 + \varepsilon^2}$ corresponding to a total variation (TV) regularization [ZPB07]. Flow-driven, isotropic regularization reduces smoothing at those locations where edges in the flow field occur, i.e. penalizes deviations from smoothness less severely than in the quadratic setting.

– Flow-Driven, Anisotropic Regularization

$$E_{\text{reg}} = \int_{\Omega} \text{tr} \left(\Psi_T \left(\nabla \delta_x \nabla \delta_x^T + \nabla \delta_y \nabla \delta_y^T \right) \right) \, dx \, dy \tag{1.24}$$

Here, $\Psi_T(\text{V})$, $\text{V} \in \mathbb{R}^{2\times 2}$ is applied to the local flow tensor, which contains additional directional information [WS01].

By comparing the results of the different regularization terms, it can be observed that anisotropic regularization generally give slightly better results than the isotropic ones. Furthermore, nonlinear (flow driven) methods are able to overcome the problem of over segmentation that typically arises for image-driven techniques in the presence of textured scenes.

Flow estimation generally is a highly non-linear problem that is often linearized as in (1.19) in order to get a simplified, convex system. However, such a linearization generally holds only for displacements that account for at most one pixel. For this reason, optical flow estimation is commonly applied in a coarse-to-fine way. Here, an image pyramid is computed for which the optical flow of the coarser level serves as initial solution for the finer level (see Fig. 1.14). On each level the current image is pre-warped according to the scaled flow information of the subjacent level. The upscaled flow field is then refined by another optical flow estimation between the pre-warped and the reference image.

Generally, coarse-to-fine estimation offers two advantages: For *convex* (linearized) energy functionals, they allow to speed up the computation significantly [BWS05]. For *nonconvex* energy functionals they allow to improve the quality of the results significantly as local minima of the energy functionals disappear at sufficiently coarse resolutions [BWKS06]. A problem of coarser-to-fine techniques, however, is the possible propagation of estimation errors from coarser levels to finer levels.

The actual time-complexity of optical flow estimation varies with the underlying estimation technique, but has often been considered to be too slow for real-time

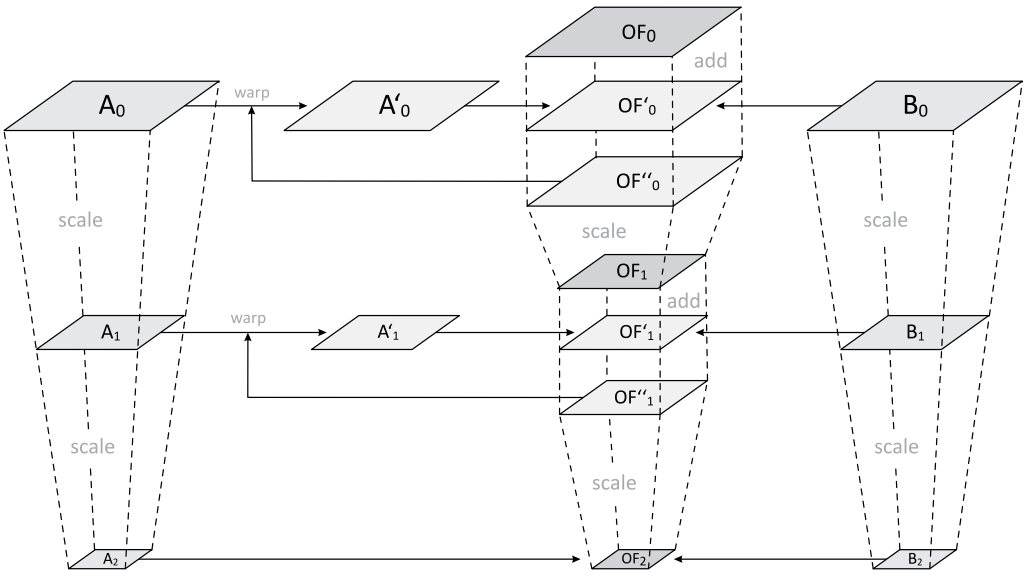


Figure 1.14: Coarse-to-fine Flow Estimation. The estimated flow field (middle) on level i serves as initial estimation for level $i - 1$.

purposes. Fortunately, deeper understanding of applied methods and parameters effect recently inspires the design of advanced and highly effective models.

A first accurate, real-time implementation has been published by Bruhn et al. [BWKS06], whose implementation of a flow-driven isotropic regularization achieves approximately 12 fps for an image resolution of 160×120 px. The basic idea of Bruhn's approach is the multi-grid (coarse-to-fine) solver for a linearized Euler-Lagrange equation system of the unlinearized regularization functional.

Zach et al. [ZPB07] later published a dual formulation of an isotropic TV regularization. Due to the dual formulation, the global optimization problem can be rewritten as iterative local statement, whose parallelized implementation on modern graphics hardware achieves 30 fps for an image resolution of 256×256 px. According to the Middlebury testbench [MDB], both implementations are currently among the best estimators for optical flow with respect to remaining angular and end-point errors.

1.5 Graphics Hardware

A various number of image and data processing techniques, like filters, are based on local restricted calculations and allow a high degree of parallelization. This parallelization, however, cannot be addressed by the main processing unit (CPU) of personal computers as it sequentially applies a single instruction to a single data element at a time. However, beside the main processing unit, modern systems nowadays also take advantage of specialized graphics hardware.

While first graphic cards in the 90s has been developed to accelerate parts of the so-called fixed graphics pipeline[2], modern hardware provides application developers with more flexible graphical processing units (GPU) that allow massive parallel processing of input data in terms of stream processing. Regarding memory bandwidth and the number of operation per seconds, the performance of such GPUs already outperforms that of comparable CPUs, while recent developments of graphics hardware towards a more generalized architecture provides computing capabilities for a wide range of applications. At present, a variate of complex problems like particle systems, optical flow computation already benefit from hardware acceleration and parallelization speed-ups [OLG*07].

The Graphics Pipeline Regarding modern graphics hardware, the classical graphics pipeline has been more and more replaced by a set of *shader programs*, where each shader program is assigned to one of three processing stages forming a single *renderpass* (cmp. Fig. 1.15):

In the first stage a so-called *vertex shader* takes over the modeling and viewing transformation as well as (perspective) projection of vertex data, yielding data in normalized, clipping coordinates with respect to the viewing frustum. Before the projection, a per-vertex lighting model like Blinn-Phong shading can be applied [Bli77].

With shader model 4.0, the vertex processing stage has been extended by a *geometry shader* that is applied after primitive assembly and allows stream alterations, i.e. the generation and removal of vertex data, based on a given input primitives (point, line or triangle).

After clipping and rasterization of the final primitives, the *fragment shader* is used to compute the final color of each affected pixel and allows per pixel processing like lighting, bump mapping or shadowing.

While first shader programs has been restricted to a limited number of instructions, recent graphic cards allow an unrestricted number of instructions.

In all three stages additional input data can be provided in form of uniform shader variables as well as a number of multi-dimensional image data (textures). Processing results can be stored and used as input data for a subsequent renderpass, i.e. rasterized fragments can be read back to serve as new texture data, while vertices processed in the geometry shader can be fed back into the pipeline via *transform feedback*. This way, iterative algorithms can be realized applying multiple renderpasses.

While early graphics hardware consisted of dedicated computing units, current graphics hardware draws advantages from an unified shading architecture allowing any of the several computing units to run any type of the three shader programs. This way, a dynamic workload balancing can be applied to avoid bottle necks,

[2]Synonym for a fixed sequence of basic processing steps for the image generation from a polygonal scene description.

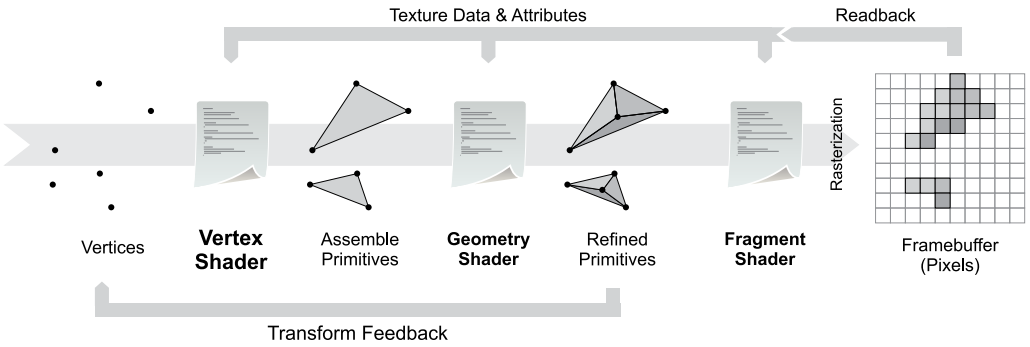


Figure 1.15: Modern render pipeline: A single renderpass from vertex data to rasterized screen information consists of three processing stages.

i.e. in situations of heavy geometry workload the balancing unit can allocate most computing units for vertex and geometry shader while in situations of heavy fragment computations most units are assigned to run the fragment program.

Programming Frameworks Modern shader programs are written using a specialized shading language. Currently three shader languages exist:

- OpenGL Shading Language (Open GLSL), which extends the open graphics library (OpenGL)
- High Level Shading Language as part of Microsoft's graphic framework DirectX
- Cg provided by the hardware manufacturer Nvidia

All languages are inspired by a C-like syntax and provide a subset of C functionality that allows loops and conditional branches, but generally excludes pointer arithmetic, dynamic memory allocation as well as text support. Furthermore, shader languages support hardware accelerated matrix and vector arithmetic as well as graphics-related functionality like texture access.

Beside graphic-related languages, the usage of the GPU as a more general purpose processing unit recently caused the development of more generalized computing languages that are not related to the original graphics pipeline concept. Compared to shader languages, these languages provide more generalized memory access as well as a synchronization mechanisms between the distinct processing units and remove the general pipeline restriction that a single stream unit is bound to write to a given output position only. This way, also graphic inexperienced developers are able to use the GPU as a parallel processor for their applications, avoiding limitations that basically emerge from the pipeline idea. Currently, two languages with similar structure exist:

- Nividia's Compute Unified Device Architecture (CUDA) as counterpart to Cg
- Open Computing Language (OpenCL) as counterpart the OpenGL framework

No matter which processing framework is finally used, utilizing the GPU always implies the adherence of the stream processing programming paradigm, which in turn makes strict demands on the program structure and data / memory layout. Therefore, the main challenge in utilizing the GPU is given by the appropriate design of effective algorithms. Here, range images already exhibits considerable advantages, as they innately comply with the data layout demands for parallel stream processing, i.e. satisfy an uniform, grid-like data alignment.

Thus, in order to archive real-time performance and preserve the original application area that distinguishes TOF cameras from other range sensing techniques, all contributions of this thesis are designed to run on the GPU.

Chapter 2

Calibration

»Accuracy of observation is the equivalent of accuracy of thinking.«

– Wallace Stevens

The following sections cover the accuracy of current TOF range cameras, and discusses the individual steps that are necessary to obtain accurate scene information in the specific case of PMD cameras. However, due to the same working principle, most techniques are also applicable to modulation-based TOF cameras of other vendors.

The first part of the chapter deals with the estimation of intrinsic and extrinsic camera parameters. This step is essential for all computer vision systems and guarantees a correct back projection of the provided distance information as described in Sec. 1.3.4. It further provides important pose estimates for, e.g., multi-sensor setups and image-based reference data acquisition as used in our distance calibration models.

The main part addresses the distance accuracy of TOF cameras with respect to systematic error sources (cmp Sec. 1.2.2). While the first sections cover the handling of systematic wiggling errors, the remaining sections are dedicated to the calibration of intensity-related errors. For each error source, a calibration model is presented that allows to significantly increase the distance accuracy.

The remaining section finally explains a compensation approach for TOF camera specific motion artifacts, whose characteristics have been outlined in Sec. 1.2.2.

Publications The intrinsic parameter estimation as well as the correction of demodulation errors has been presented in [LK06], while the correction of intensity-related errors are subject of [LK07a, LSKK10]. The alternative demodulation approach has been published in [LKR08]. The compensation of TOF camera-specific motion artifacts together with the simple axial motion model is discussed in [LK09].

2.1 Camera Parameter and Pose Estimation

The estimation of camera parameters is an essential task for all computer vision systems (cmp. Sec. 1.3.3). Here, most calibration approaches are typically based on one-to-one correspondences between a known calibration target and its projection

on the two dimensional image plane [Fau93, MSKS04, Zha00]. In order to avoid a manual processing of the input images, commonly patterns like checkerboards are used. By doing so, pattern features can be automatically detected, which allows a more accurate sub-pixel mapping. The latter is of great importance in order to reduce quantization errors that are introduced by the uniform pixel grid.

Compared to typical vision systems, however, TOF cameras currently provide only a low resolution along with a narrow opening angle. Both properties might have negative effects on a correct parameter estimation. In a first instance, we therefore investigated the general applicability of a conventional camera calibration to TOF range images. Here, we choose the calibration module of the Open Computer Vision Library (OpenCV) [OCV] to estimate the camera parameters of a PMD 19k camera model. The module is a straight forward implementation of Zhang's calibration approach as outlined in Sec. 1.3.3. Applied correctly, the module yields the intrinsic camera parameters as well as a pose estimate for each input image of a planar checkerboard.

In most of our tests, the calibration module estimated satisfying results for the camera intrinsics (see Tab. 2.1) as well as suitable rough pose estimates concerning the distance and orientation of the checkerboard. In addition to that, also the slight barrel distortions of the checkerboard has been adjusted correctly as can be seen in Fig. 2.1. Regarding the estimation results, tangential distortion can be mostly neglected, which can be ascribed to the simple camera optics.

In all tests, however, the main complexity of the parameter estimation turns out to be the accurate detection of checkerboard corners. Especially the pose estimation is strongly affected by noise and pixel quantization effects. Due to the low resolution, a minimum projection size of the checkerboard fields is required to achieve a correct corner detection inside the pixel grid. Especially large rotation angles between the camera and the checkerboard cause detection problems and therefore should be avoided. This and the necessity to acquire the complete pattern in each view, limits the number of possible input poses that are generally important for a robust parameter estimation.

For sensors smaller than the 19k model, the problem of accurate corner

	Pass 1	**Pass 2**	**Pass 3**
Focal Length (f_x, f_y) [mm]	12.39, 12.36	12.29, 12.28	12.30, 12.29
Image Center (c_x, c_y) [px]	77.49, 66.20	76.27, 68.75	78.93, 63.34
Radial Distortion (k_1, k_2)	-0.487, 1.131	-0.470, 1.284	-0.482, 1.701
Tangential Distortion (t_1, t_2)	0.001, 0.001	0.000, 0.005	0.002, 0.001

Table 2.1: Calibration results for a 160×120 px PMD camera using a conventional calibration framework. According to the camera specifications, the camera has a focal length of 12 mm.

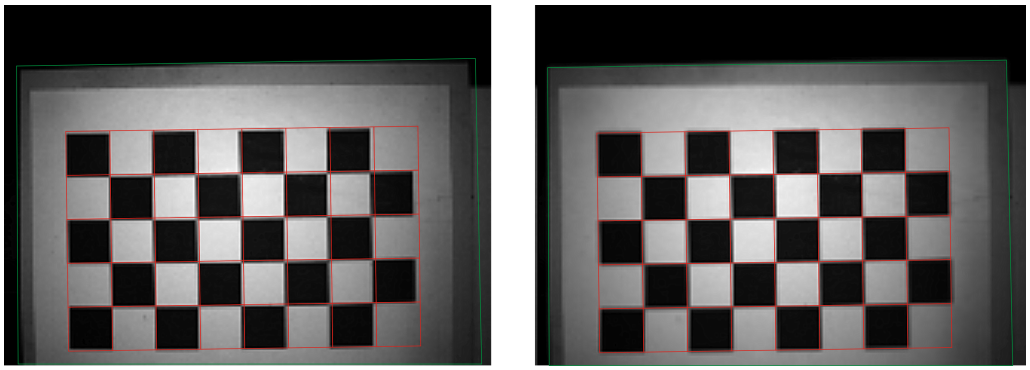


Figure 2.1: Example of an PMD range image before (left) and after an image undistortion (right).

detection even increases. Here, in most cases a manual selection is required that negatively affects the sub-pixel accuracy and the parameter estimation respectively. Consequently, either an alternative calibration target or calibration method must be considered, that allows a more accurate TOF camera calibration for low resolution input images.

Two alternative calibration patterns have been tested by Kahlmann et al., comprising circular features as well as a special calibration target consisting of NIR LEDs [KRI06]. While circular targets occupy too many pixels in the small image and limit the number of correspondence points, the best solution (according to Kahlmann et al.) is given by the calibration target made of NIR LEDs, as the reference points can be detected via simple thresholding. However, in contrast to a plotted pattern, this kind of target requires an accurate assembly of electronic components and also implies a certain target size.

A novel calibration approach has been published by Beder and Koch [BK08], who utilizes the available range information to synthesize an image of the according view. The camera parameters (including distortion coefficients) are iteratively optimized until the synthesized checkerboard matches the given reference image. By doing so, a highly accurate sub-pixel accuracy can be achieved. In contrast to other calibration models, a single reference image of the TOF camera (consisting of range and intensity information) is sufficient to estimate all parameters including intrinsic as well as extrinsic values. Furthermore and more importantly, the problem of correspondence detection is avoided by the synthesis process, which considers all pixels inside the projected checkerboard as reference points. However, in order to generate the synthesized view, the distance information is assumed to be accurate, which is a priori not the case due to systematic deviation errors (see Sec. 1.2.2). Also a strong correlation between rotation and translation of the checkerboard, caused by the narrow opening angle of the camera, has been reported.

For this reason, Schiller et al. [SBK08] proposed an extension of the calibration

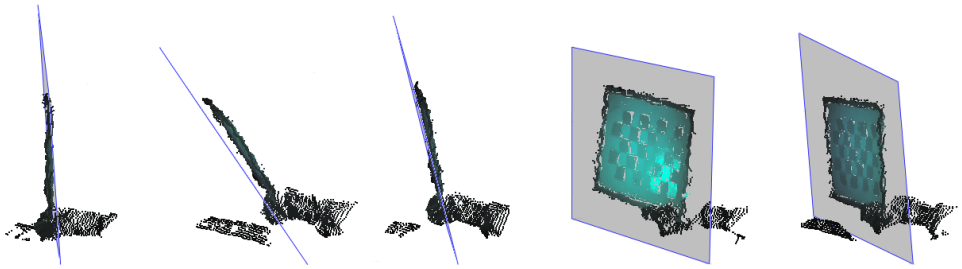


Figure 2.2: Vision-based reference data acquisition. The reference plane (blue) is calculated with respect to the extrinsic parameters of the checkerboard.

model that considers multiple images and incorporates an additional distance calibration to adjust false distance information (see Sec. 2.2). In contrast to the original approach, the new approach uses a multi-camera rig of at least one TOF and one CCD camera to reduce the correlation between the TOF camera's extrinsic parameters. Accordingly, the rig pose is estimated based on the high resolution CCD camera images, allowing high accurate results, whereas the TOF camera pose is estimated relative to the additional CCD sensor, using multiple pose images to stabilizes the whole estimation process.

While classical calibration techniques will gain in precision with increasing sensor resolution, Schiller's multi-cam approach currently gives the best results for an overall parameter estimation. Especially the high accurate pose estimates are of great importance for sensor fusion as discussed in Sec. 3.3. It also is the only high-precision calibration model that is currently able to handle TOF cameras with less resolution than the 19k model.

2.2 Distance Correction

Knowing the intrinsic parameters of a TOF camera, each pixel of the range image can be back projected by (1.17), p. 24, to provide a three dimensional point cloud of the acquired scene. The reconstruction results, thereby, mainly depend on the accuracy of the acquired distance information. Unfortunately, as stated in Sec. 1.2.2, the distance information of continues modulation-based TOF cameras is negatively affected by systematic error sources. In order to increase the distance accuracy and thus the reliability of the reconstruction result as well, a calibration of the systematic distance deviations is of great importance.

In the last three years, a number of calibration models have been introduced that cover the systematic demodulation error either alone or in combination with intensity-related deviations. The following sections will discuss the contributed calibration models for both, systematic demodulation (Sec. 2.2.2) as well as

intensity-related distance deviations (Sec. 2.2.4). They also depict similarities and differences to related models, which has been published during the same period.

Sec. 2.2.2 first describes a global adjustment approach, which is then further extended by a pixel-wise pre-adjustment in order to account for pixel inequalities. Furthermore, an alternative demodulation approach is investigated, which is based on a rectangular signal shape and accounts for the fact that the the true signal shape differs from the theoretical assumption of a cosine function.

Sec. 2.2.4 first describes an simple extension of the previous described wiggling model to handle the intensity-related error using a bivariate approach. Finally, a decoupled approach is presented that significantly reduces the number of required reference data.

2.2.1 Vision-Based Reference Data Acquisition

Considering a TOF camera to be a closed system, all discussed distance calibration models basically investigate potential deviations between measured and a priori known reference distances, commonly provided by, e.g., a track line [LK06, KRI06] or robots [FH08]. In addition, we suggest a vision-based alternative that utilizes the extrinsic parameter estimation from the previous section to avoid expensive equipment (see Fig. 2.2).

Knowing the pose estimate $\mathrm{M} = \begin{bmatrix} \mathrm{r}_1 & \mathrm{r}_2 & \mathrm{r}_3 & \mathrm{t} \end{bmatrix}$ of the calibration panel, the reference distance d of a pixel $\mathbf{p}$ can be easily computed by intersecting the pixel's viewing ray $\mathbf{r}(\alpha) = \alpha \cdot \mathrm{K}^{-1}\mathbf{p}$ with the reference plane $\langle \mathbf{X}, \vec{\mathrm{n}} \rangle = \zeta$. Both plane parameters are given by $\vec{\mathrm{n}} = \mathrm{M} \begin{bmatrix} 0, & 0, & 1, & 0 \end{bmatrix}^T = \hat{\mathrm{r}}_3$ and $\zeta = \langle \mathrm{t}, \vec{\mathrm{n}} \rangle = \langle \mathrm{t}, \hat{\mathrm{r}}_3 \rangle$. The reference distance d, i.e. the plane intersection point, is given by

$$d(\mathbf{p}) = \left\| \frac{\langle \mathrm{t}, \hat{\mathrm{r}}_3 \rangle}{\langle \mathbf{X}, \hat{\mathrm{r}}_3 \rangle} \cdot \mathbf{X} \right\|, \quad \text{with} \quad \mathbf{X} = \mathrm{K}^{-1}\mathbf{p}. \tag{2.1}$$

However, according to the stated dependency between extrinsic parameters for narrow opening angles in Sec. 2.1, we recommend the utilization of a camera rig to improve the required pose estimate. This way, the panel's transformation can be estimated more accurately via the additional, high resolution CCD camera and transformed into the TOF camera's coordinate system by using relative pose estimates between both sensors. Using the TOF camera for pose estimation alone, generally results in an unsteady reference plane, as sensor noise and missing sub-pixel accuracy affects the transformation results.

Tab. 2.2 shows the accuracy of a vision-based pose estimation using a high resolution CCD camera. Here, the true reference distance and rotation angle is compared to the estimated value. The experiment shows that both, the displacement error as well as the angular error, are less than two millimeters / degrees and therefore yield a suitable pose estimation. Regarding the distance accuracy, the constant offset in all measurements can be explained by the choice of the

Reference Distance	[cm]	220.0	160.0	140.0	120.0
Estimated Distance	[cm]	219.1	158.7	138.2	118.1
Reference Angle	[deg]	0.0	15.0	30.0	45.0
Estimated Angle	[deg]	1.5	16.8	30.0	46.5

Table 2.2: Pose estimation results regarding the displacement and rotation of the reference checkerboard. The experiment yields that both, the displacement error as well as the angular error, are less than two millimeters / degrees.

reference point that represents the camera position and differs from the sensor's true position.

2.2.2 Systematic Demodulation Error

Concerning the distance accuracy of current TOF cameras, the most significant error source is caused by the camera's phase shift demodulation itself (cmp. Sec. 1.2.2). By regarding the distance deviations between reference and measured distances, it gets obvious that the modulation mismatch with respect to the theoretical and real emitted signal results in a systematic error that can be best described as a sinusoidal wiggling (see Fig. 1.8, p. 15).

In order to improve the phase estimation, a modification of the demodulation scheme incorporating higher Fourier modes has been proposed, that has been already discussed in Sec. 1.2.2. In practice, however, the modification is rather impractical as the number of required sample images as well as the calculation effort for the demodulation significantly increases. Especially the number of sample images is crucial with respect to the real-time capability and the occurrence of motion artifacts (see Sec. 2.3). For this reason, the simpler sinusoidal-based demodulation is still preferred. Whereas, at the same time, an adequate error correction is mandatory to allow accurate range sensing in return.

In the following section we describe a phenomenological calibration approach, that is based on a mathematical approximation of the demodulation error. A comparable approach based on a look-up table (LUT) has been published simultaneous by Kahlmann et al. [KRI06]. In contrast to former models, which tried to describe the distance deviation by a linear approximation [SK06], both approaches are capable to express the wiggling error in a high accurate way. Due to their phenomenological nature, the input data for both models is given by a proper set of distance measurements covering the camera range as well as the according ground-truth data as, for example, determined by our vision-based acquisition system (see Sec. 2.2.1). While Kahlman's approach relies on one or more well chosen pixels in the image center, our model takes all reliable pixels into account.

In the case of Kahlmann et al., a LUT is build up for a well-chosen center pixel, yielding the distance deviation as a function of measured distance. If necessary,

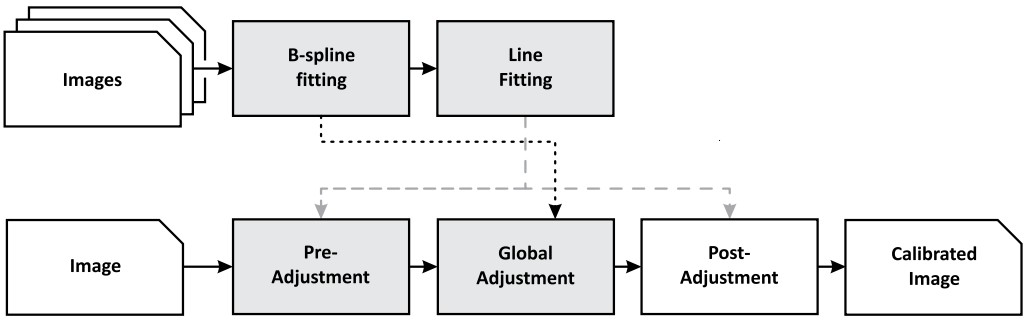


Figure 2.3: The calibration process for the correction of demodulation errors.

two table entries are linear interpolated to provide results for intermediated distance values. In addition to that, individual pixel offsets are handled by a *fixed pattern noise* matrix, which is obtained for a fixed distance by regarding each pixel's deviation with respect to the given reference pixel. The approach, however, has major drawbacks as it requires a dense sampling of the camera range to avoid interpolation errors – especially near inflection points of the wiggling error.

For this reason, we follow a different approach and approximate the demodulation error by a compact functional representation that finally serves as a correction function. In doing so, only a few fitting coefficients have to be stored, keeping the memory footprint low. Furthermore, it generally requires less input data compared to LUTs as the underlying B-spline estimation provides a smoother interpolation and thus is less vulnerable to interpolation errors. Similar to Kahlmann et al., the distance calibration is done in two distinguished steps (see Fig. 2.3):

1. Fitting a global deviation approximation for the entire image that represents the actual wiggling adjustment.

2. An additional estimation of a local per pixel pre- and (optional) post-adjustment to compensate pixel inequalities.

Here, the main idea behind the separation is to use a function of higher complexity for the global adjustment, whereas the per-pixel calibration uses simpler, i.e. constant or linear adjustment, thus being storage-efficient concerning the overall number of calibration parameters. The following sections will give more detailed information about each step.

Global Distance Adjustment Given the periodicity of the demodulation error (see Fig. 1.8, p. 15), a first attempt to approximate the distance deviations would be to use a superposition of sinusoidal base functions. However, the selection of a minimal set of proper base functions is rather complex. Thus, in order to reduce

complexity and to provide a more universal approach, an uniform, cubic B-spline

$$B(u) = \sum_{i=0}^{n} c_i \cdot N_i^3(u) \quad \text{where} \quad u_3 \leq u \leq u_n \tag{2.2}$$

is used instead, where n is the number of control points, N_i^3 represents the uniform cubic B-spline basis functions and $[u_0, \ldots, u_{n+4}]$ is the underlying knot vector. Compared to other possible approximations, B-splines exhibit a better local control and can be efficiently expressed by a simple matrix multiplication for uniformly distributed knot vectors. In the case of cubic B-splines with $u_i = i$, for example, the evaluation is alternatively defined by

$$B(t+j+3) = \frac{1}{6}\begin{bmatrix}1 & t & t^2 & t^3\end{bmatrix}\begin{bmatrix}1 & 4 & 1 & 0\\ -3 & 0 & 3 & 0\\ 3 & -6 & 3 & 0\\ -1 & 3 & -3 & 1\end{bmatrix}\begin{bmatrix}c_j\\ c_{j+1}\\ c_{j+2}\\ c_{j+3}\end{bmatrix} \tag{2.3}$$

with $0 \leq t < 1$ and $0 \leq j < n-3,\ j \in \mathbb{Z}$. The distance adjustment, therefore, can be implement quite efficiently on modern graphics hardware by using build-in data types for matrices and vectors.

Given a set of appropriate input images, the control points c_i for the desired B-spline correction $B() \mapsto d$ can be found via a least square fit, i.e. minimizing $\|\mathrm{A}\mathbf{c} - \mathbf{b}\|^2$ with

$$\mathrm{A} = \left[B_i^3(m_J)\right]_{i=0,\ldots,n}^{J=(\mathbf{p},k)}, \qquad \mathbf{c} = [c_i]_{i=0,\ldots,n}, \qquad \mathbf{b} = [d_J]_{J=(\mathbf{p},k)} \tag{2.4}$$

where m_J stands for the measured distance information at a pixel $\mathbf{p}$ of the k-th input image and d_J stands for the corresponding reference distance. The system of linear equations described by (2.4) can be finally solved using a pseudo inverse approach

$$\mathbf{c} = \left(\mathrm{A}^T\mathrm{A}\right)^{-1}\mathrm{A}^T\mathbf{b}. \tag{2.5}$$

Given the set of optimal B-spline control points and a measured distance $m(\mathbf{p})$, the global adjusted distance value $m_c(\mathbf{p})$ is simply given by

$$m_c(\mathbf{p}) = B(m(\mathbf{p})). \tag{2.6}$$

As the B-spline can only handle distance information that lies inside the calibration range, values outside the range have to be either clamped or leaved untouched – depending on the system requirements. In both cases, a special pixel flag might avoid that the system uses possibly invalid information.

The optimal number of control points strongly depends on the calibration range and has to be adjusted to the individual case. Insufficient control points

lead to inaccurate (even incorrect) correction results, where as too many might cause over fitting between individual input images. Beside manual adjustment, a possible solution is to apply an iterative fitting and increase the number of control points until a given fitting threshold is reached. In our case, where the periodicity of the demodulation error covers approximately two meters (see Fig. 2.4 and Fig. 1.8, p. 15), 12 control points are usually sufficient to cover the whole camera range of 7.5 m.

Figure 2.4 top, shows first results of the global correction for randomly selected pixels before and after the correction has been applied.

Per-Pixel Distance Adjustment Due to the applied optimization, the resulting B-spline actually yields the global correction for an optimized average pixel $\mathbf{p}_{\varnothing}$ with respect to a given set of input images. Due to individual pixel characteristics, however, the B-spline gets evaluated at slightly shifted locations, which in turn negatively affects the global correction result.

Instead of fitting a B-spline for each sensor pixel, a pixel-wise pre-adjustment can be applied that additionally adjust each sensor pixel towards to the optimized average pixel, i.e. maps $m(\mathbf{p}, k)$ to $m(\mathbf{p}_{\varnothing}, k)$, and is comparable to the fixed pattern noise matrix mentioned by Kahlmann et al. Assuming that individual pixel offsets are linear, the pre-adjustment is determined by pixel-wise fitting a line $l_{\mathbf{p}}$ into the image sequence minimizing

$$\sum_{k} \|m(\mathbf{p}, k) - m(\mathbf{p}_{\varnothing}, k)\|^2. \tag{2.7}$$

In contrast to the noise matrix, where the LUT's reference pixel is known, $m_{\varnothing} = m(\mathbf{p}_{\varnothing}, k)$ is a priori unknown, but can be determined by solving the inverse mapping $B(m_{\varnothing}) = d(\mathbf{p}, k)$, i.e. searching the root of

$$B(m_{\varnothing}) - d(\mathbf{p}, k) = 0 \tag{2.8}$$

via, e.g., Newton's method or bisection. In both cases, the initial solution should be as close to the reference distance as possible, i.e. $m_{\varnothing} = d(\mathbf{p}, k)$, to avoid local minima. Applying the additional pre-adjustment, the overall distance calibration is given by

$$m_c(\mathbf{p}) = B(m_{\varnothing}) \quad \text{with} \quad m_{\varnothing} = l_{\mathbf{p}}(m(\mathbf{p})). \tag{2.9}$$

In order to eliminate remaining distance deviations a second line fitting analog to the pre-adjustment can be performed. This time for the differences between the corrected distances $m_c(\mathbf{p}, k)$ and the reference distance $d(\mathbf{p}, k)$.

A final example for a wiggling adjustment is shown in the second plot in Fig. 2.4. By separating the calibration into two steps, a high accurate distance calibration can be achieved, reducing the amount of distance deviation to an average value less than 1 cm (cmp. Tab. 2.3).

However, as can be seen in the results, it turns out that an additional post-adjustment can be neglected as it leads to no significant improvements. Furthermore, it basically makes no difference whether a pre- or post-adjustment is applied, as both techniques lead to the same results. A pre-adjustment, however, is technically more sound.

Our experiments also show, that most coefficients of the pre-adjustment describe constant offsets, i.e. have a slope close to one. Consequently, the number of coefficients necessary for the pixel-wise adjustment can be reduced to one, which is comparable to a fixed pattern noise matrix.

2.2.3 Alternative Demodulation Approach

Phenomenological calibration models as well as enhanced demodulation schemes generally imply special effort by means of hardware modifications or reference data acquisition. Both methods are consequently either limited or rather time consuming with respect to their realizations. A more desirable approach, therefore, would use the standard four sample values, but leads to more accurate results.

The main reason for demodulation errors is the mismatch between theoretical and real signal modulation. Experiments show, that the reference signal for current PMD cameras actually correspond to a mixture between a rectangular and a sinusoidal signal [Rap07]. Consequently, we investigate an alternative demodulation approach that uses the standard four correlation samples, but considers a rectangular signal.

Assuming an ideal rectangular signal, the resulting cross correlation function $c(\tau)$ is triangular with its valley points displaced by the phase shift ϕ (see Fig. 2.5). Thus, for a shift smaller than $\frac{\pi}{2}$ as in Fig. 2.5a left, the phase shift ϕ can be obtained by fitting two intersecting lines

$$l_{1,2}(\theta) = m_{1,2} \cdot (\theta - \phi) + t \tag{2.10}$$

with contrary slopes $m_1 = -m_2$ and identical offset t through the sample points I_i. Here, l_1 is fitted through I_0 and I_1, whereas l_2 is fitted through I_2 and I_3 The equivalent least square optimization based on the resulting linear system

$$\begin{bmatrix} 0 & -1 & 1 \\ \pi/2 & -1 & 1 \\ -\pi & 1 & 1 \\ -3\pi/2 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} m \\ \phi m \\ t \end{bmatrix} = \begin{bmatrix} I_0 \\ I_1 \\ I_2 \\ I_3 \end{bmatrix} \tag{2.11}$$

leads to

$$\phi = \pi - \frac{\pi}{2} \cdot \frac{(I_3 - I_1)}{(I_0 - I_2) + (I_3 - I_1)}. \tag{2.12}$$

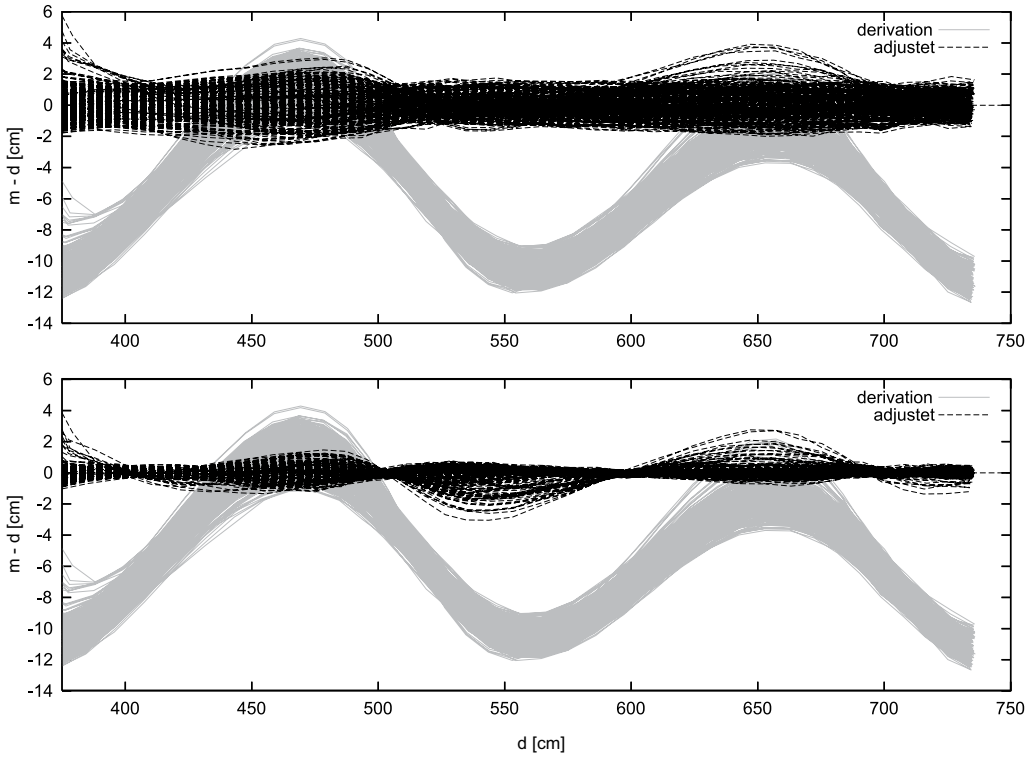


Figure 2.4: Original (light gray) and adjusted deviation (dashed, dark gray) between the measured and the expected reference distance for distances between $3.5 - 7.5\,\mathrm{m}$ (top: global correction only, bottom: additional pre-adjustment).

Correction	**Global**	**+ Pre**	**+ Pre / Post**	**+ Post**
Image 1	0.880074	0.277263	0.285939	0.316298
Image 2	0.842202	0.242508	0.237220	0.238037
Image 3	0.854546	0.369220	0.364661	0.376175
Image 4	0.824755	0.461143	0.463916	0.464283
Image 5	1.037930	0.556160	0.562826	0.554429
Pixel (80, 60)	1.277160	0.253993	0.253471	0.303678
Overall	0.928622	0.434910	0.434879	0.437099

Table 2.3: Calibration results for the demodulation adjustment using the global adjustment only (global) and together with a pixel-wise pre-adjustment (pre), pre- and post-adjustment (pre / post) or post-adjustment. The table states the mean deviation error in [cm] for selected reference images (row 1 - 5), for the center pixel over all reference images (row 6) as well as the overall mean error covering all pixels in all images (row 7).

Considering an additional shift for cases where $I_0 < I_1$ and $I_2 > I_3$, the real, distance-related phase shift $\phi_{\text{tri}} = \pi - \phi$ finally can be obtained by

$$\phi_{\text{tri}} = \frac{\pi}{2} \cdot \frac{(I_3 - I_1)}{(I_0 - I_2) + (I_3 - I_1)} + \begin{cases} \pi & I_0 < I_1 \wedge I_2 > I_3 \\ 0 & \text{else} \end{cases} \tag{2.13}$$

whereas the signal amplitude a is given by

$$a = \left| \frac{1}{4} \sum_{i=0}^{3} I_i - t \right|. \tag{2.14}$$

Special care must be taken for phase shifts where the valley points are located between the first two and the last two sample points (see Fig. 2.5b). In this case, the last sample point must be moved to the front in order to establish the correct fitting situation. This means that I_i becomes $I_{(i+1) \mod 4}$ whereas the intersection point ϕ is shifted by an additional amount of $\pi/2$.

Applying the new demodulation approach, the distance error unfortunately can not be reduced compared to the sinusoidal case as can be seen in Fig. 2.6. However, as the error trend turns out to be inverse to the systematic demodulation error for sinusoidal modulation, the new sampling approach can be used to attenuate the distance error by mixing the results of both demodulation schemes, i.e. a linear combination

$$\phi = k_1 \cdot \phi_{\text{sin}} + k_2 \cdot \phi_{\text{tri}} + k_3 \tag{2.15}$$

seems to be suitable to compute a new phase offset with higher accuracy than the one provided by either demodulation schemes alone. Analog to previously described calibration models, the optimal linear combination can be found by a least square optimization with respect to known reference data. In order to keep the number of calibration parameters as small as possible, we decided to let k_3 be a constant per-pixel offset comparable to fixed pattern noise (or pre-adjustment), whereas k_1 and k_2 correspond to global calibration parameters for the entire image.

Fig. 2.7 depicts the remaining derivations for the new alternative demodulation approach in comparison to results of the corresponding B-spline adjustment. As expected, experiments show that the combined demodulation approach actually can not keep up with the B-spline adjustment, but in contrast is fairly independent to the number of required reference images, as two reference images are already adequate to archive good results. The combined demodulation, therefore, is very effective in means of required reference data.

Compared to a constant or linear per-pixel adjustment of the original demodulation scheme (see Fig. 2.8), the combined demodulation model gives clearly the best results as these techniques are unable to cope with the systematic demodulation error. Improved results can be achieved by high-order combinations of ϕ_{sin}

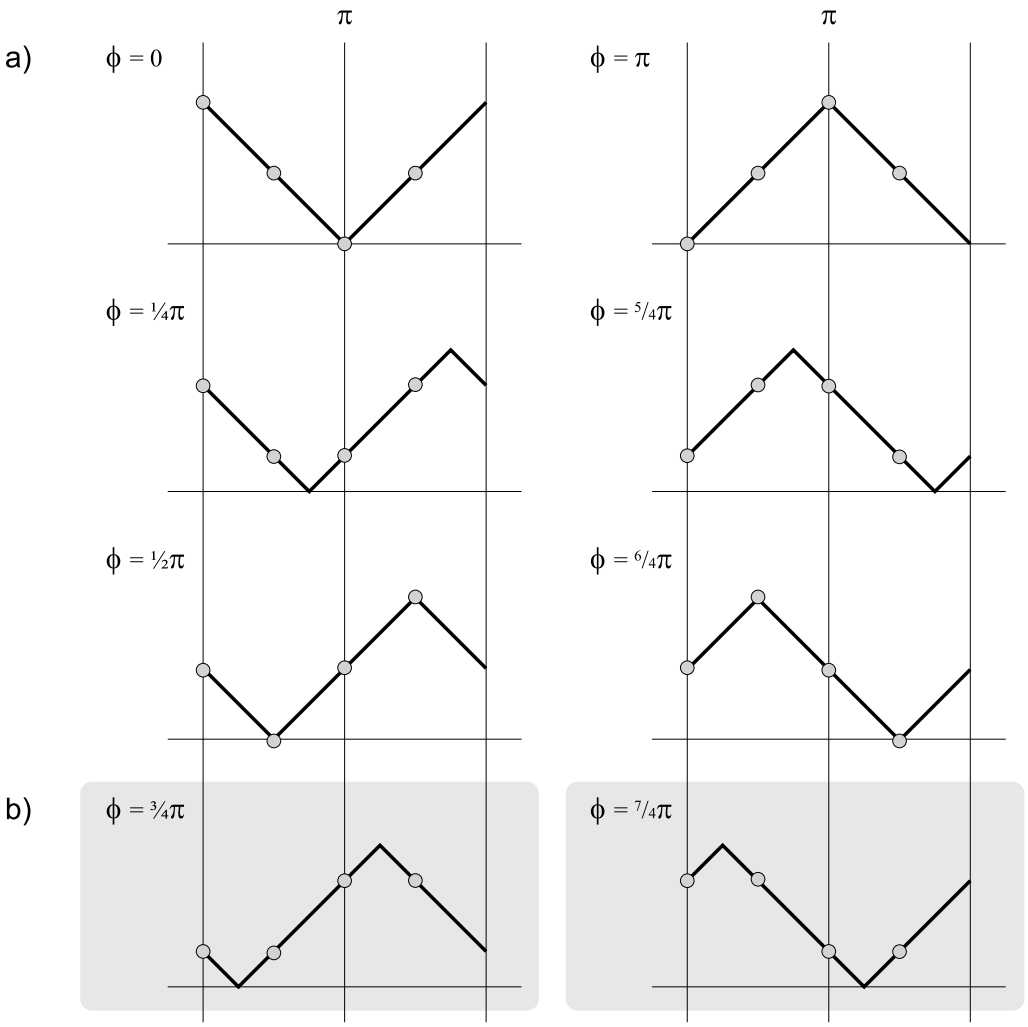


Figure 2.5: Sampling positions for a triangular correlation function and their corresponding phase offset. Special care must be taken for the gray shaded cases.

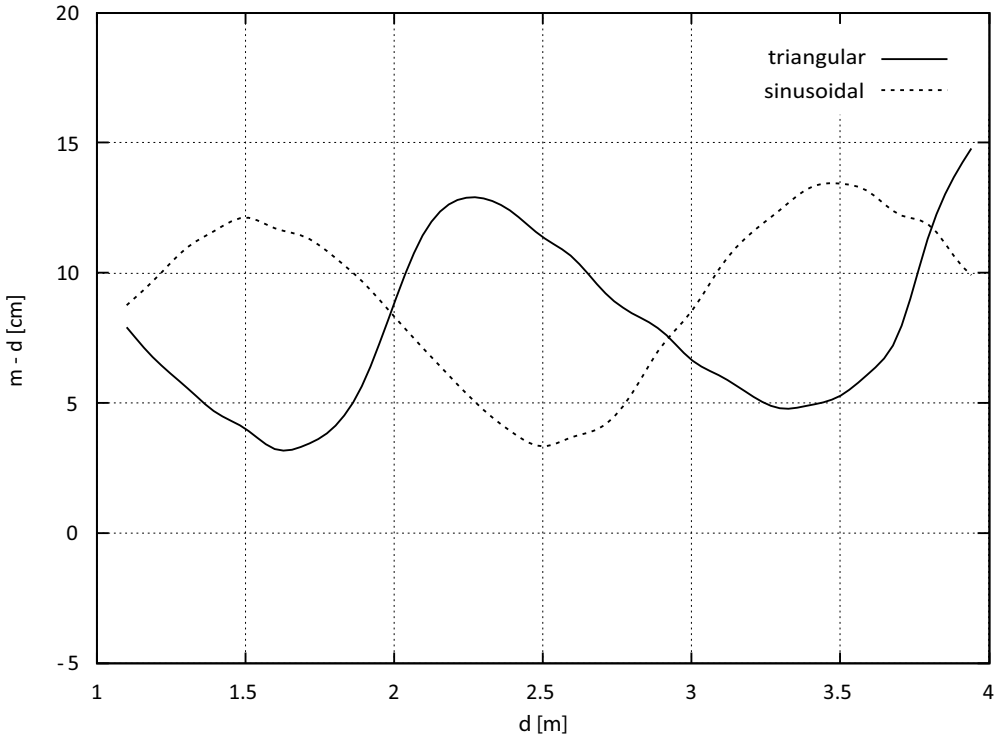


Figure 2.6: Mean distance error for the original (dashed) as well as the alternative, triangle-based demodulation (solid).

and ϕ_{tri} like

$$\phi = \sum_{i,j=0}^{2} k_{ij} \cdot \phi_{\text{sin}}^{i} \phi_{\text{tri}}^{j}. \tag{2.16}$$

Unfortunately, this way the number of parameters and therefore the number of necessary reference images would increase again, which in result brings no real advantages compared to the B-spline approach.

2.2.4 Intensity-Related Error

By adjusting distance information according to the compensation models discussed in Sec. 2.2.2, range images already provide increased accuracy with respect to both, systematic wiggling as well as integration time dependent errors. However, as all described models consider distance data only – and thus assume a single, fixed reflectivity during calibration –, intensity-related errors as described in Sec. 1.2.2 can not be handled in an adequate way and therefore remain.

While the actual origin of intensity-related errors are unknown, they are assumed to be caused by non-linearities of the semiconductor and pixel gains, and

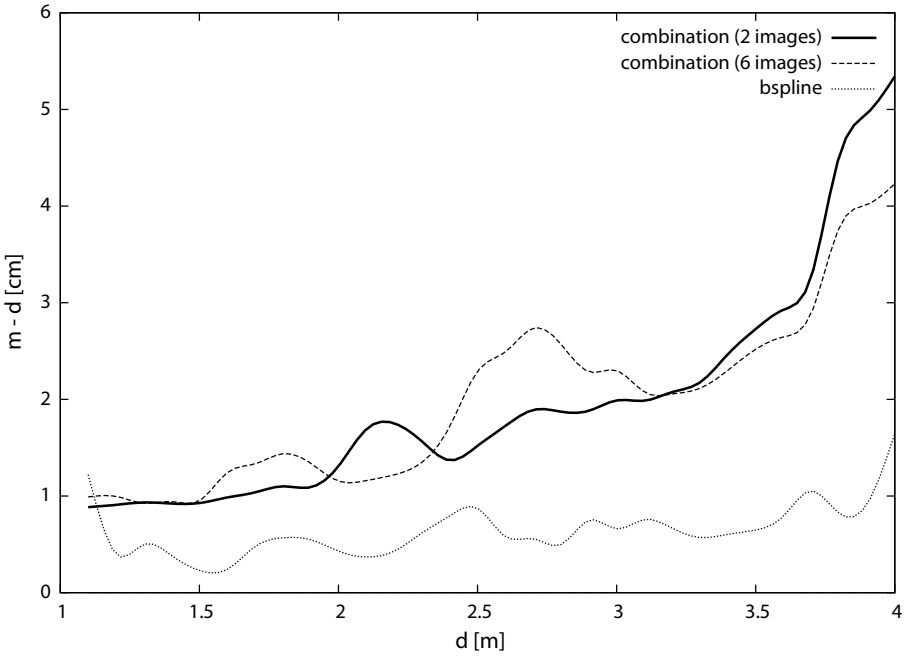


Figure 2.7: Mean distance error for the combined demodulation approach using two (solid) or six reference images (dashed line) compared to the optimal B-spline results.

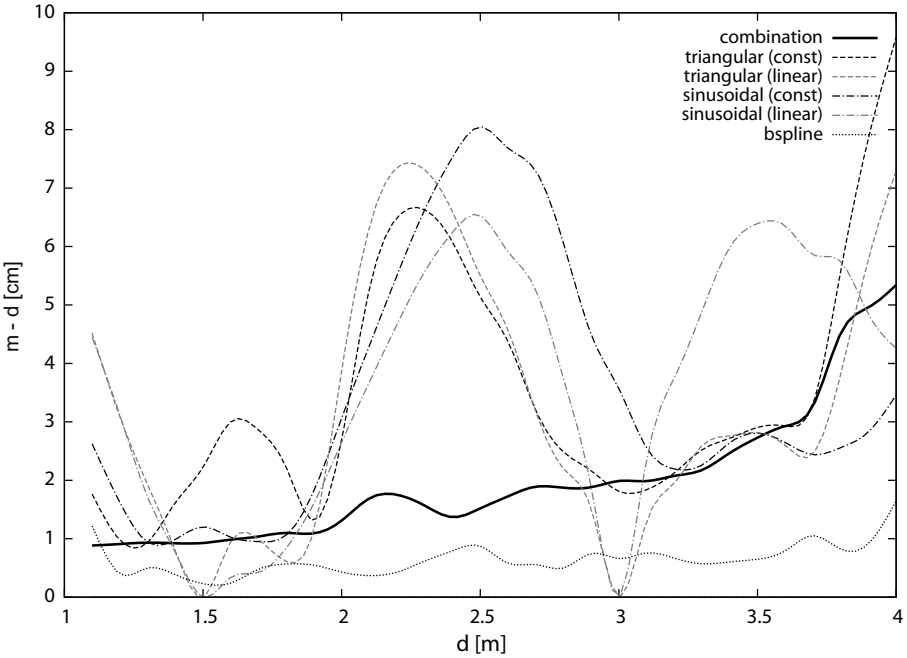


Figure 2.8: Mean distance error for the combined demodulation approach (solid line) compared to constant and linear per pixel adjustment (dashed lines) using two reference images only. The B-spline results are added for completeness.

therefore apply before the demodulation induced wiggling error, i.e.

$$m = \text{wiggling} \oplus \text{intensity} \oplus d \tag{2.17}$$

In all our experiments, low reflective regions tend to drift towards the camera, while the overall effect decreases with larger distance. Particularly the fade of the effect indicates that the error impact is strongly related to the absolute amount of the incident light. However, as the intensity error is additionally overlayed with wiggling deviations, a separate treatment is rather difficult. Recent calibration models therefore are commonly straight forward extensions of the phenomenological wiggling correction presented in Sec. 2.2.2.

In practice, the absolute amount of incident light can not be measured directly. The only information available instead is the intensity information h as well as the signal's amplitude a, which are additionally provided by the camera. Assuming homogeneous depth information, both values correlate and therefore are mostly interchangeable in practice. By considering one of both quantities as additional parameter, the deviation impact on the distance accuracy can be modeled with respect to the measured distance and intensity by considering different reflective reference targets at various distances.

In the following, two different calibration models will be discussed. While the first one incorporates the intensity information in a coupled distance-intensity model by using a two-dimensional error approximation, the second one aims for a more sophisticated, decoupled approach in order to reduce the large amount of necessary reference data. Two similar approaches have been recently published by Fuchs and Hirzinger [FH08] as well as Radmer et al. [RFSK08].

Analog to the coupled model, Fuchs and Hirzinger exploit a multiple B-spline approach by fitting a set of one-dimensional B-splines for a given number of intensities. The final adjustment is determined by linear interpolation between the two adjacent B-splines that enclose the pixels intensity. Again a dense set of intensity samples is necessary to avoid interpolation inaccuracies.

Radmer et al., in contrast, describe a decoupled approach that uses an additional three-dimensional LUT to further improve the already wiggling adjusted distance information with respect to the intensity value h and the camera's current integration time as well as the given distance itself. Like the wiggling adjustment by Kahlmann, Radmer uses linear interpolation between table entries, which again requires a dense set of reference data to avoid interpolation (linearization) errors. Apart from that, a lot of memory is wasted as the complete intensity range is not fully present for all distances due to the light attenuation.

Coupled Model

In the coupled approach, demodulation and intensity impacts are covered in combination, i.e. the measured distance information m and the intensity value h

form a bivariate domain for a combined distance adjustment $m_c(\mathbf{p}) = f(m_\mathbf{p}, h_\mathbf{p})$. It therefore can be considered as a straight forward extension of the demodulation model presented in the previous section. Accordingly, the distance correction is modeled by a bi-variate B-spline patch

$$P(u,v) = \sum_{i=1}^{p} \sum_{j=1}^{q} N_i^3(u) \cdot N_j^3(v) \cdot c_{ij} \tag{2.18}$$

analog to Sec. 2.2.2, where $N_i^3(u)$ and $N_j^3(v)$ represent the uniform cubic B-spline basis functions over a set of control points of size $p \cdot q$.

Given a set of input data $\Omega = \{(m, h, d)\}$, where each element consists of a measured distance m, the incident light h as well as the according reference distance d, a linear equation system can be derived that includes a fitting constraint

$$\sum_{i=1}^{p} \sum_{j=1}^{q} N_i^3(m_k) \cdot N_j^3(h_k) \cdot c_{ij} = d_k \tag{2.19}$$

for each pixel $k \in \Omega$, where m_k and h_k represent the measured distance and intensity information.

By arranging the fitting constraints such that the unknown coefficients c_{ij} form a vector $\mathbf{c} = (c_{11}, c_{12}, \ldots, c_{1m}, c_{21}, \ldots, c_{pq})^T$, the complete equation system can be expressed as $\mathrm{A}\mathbf{c} = \mathbf{b}$ with $\mathbf{b} = (d_1, \ldots, d_{|\Omega|})^T$ and

$$\mathrm{A} = \left[N_i^3(m_k) \cdot N_j^3(h_k) \right]_{\substack{(i,j) = (1,1),\ldots,(p,q) \\ k = 1,\ldots,|\Omega|}} \tag{2.20}$$

Due to the local support of the B-spline patch and the distance dependent light attenuation, however, $\mathrm{A}\mathbf{c} = \mathbf{b}$ is generally under constrained, and cannot be solved a priori as it has been the case for the systematic wiggling approximation. Especially for larger distances, no samples for high incident light exist. In order to avoid numerical instabilities and to guarantee a smooth extrapolation, additional smoothing constraints must be added. For this we apply the Laplace operator to form the following smoothing constrains

$$\nabla P(u,v) = \nabla \frac{\partial^2 P}{\partial u^2}(u,v) + \frac{\partial^2 P}{\partial v^2}(u,v) = 0. \tag{2.21}$$

Applying the derivation rules for B-splines and the fact, that B-splines form a function basis, (2.21) is equivalent to

$$4c_{ij} - (c_{i-1,j} + c_{i,j-1} + c_{i+1,j} + c_{i,j+1}) = 0 \tag{2.22}$$

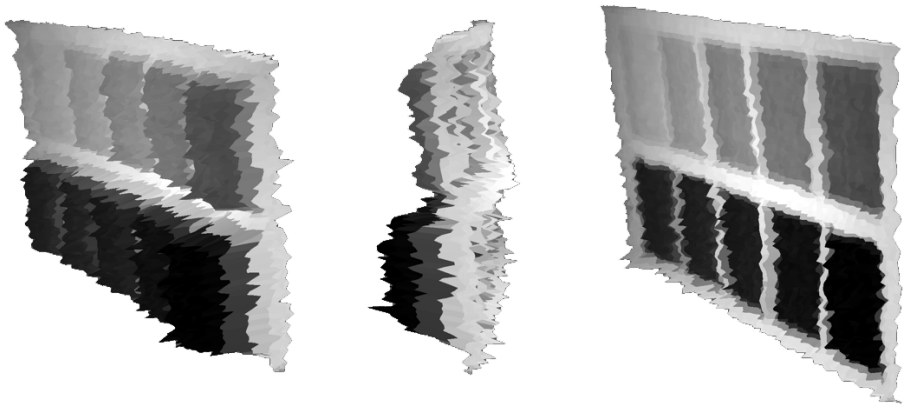


Figure 2.9: Distance deviation due to varying object reflectivity, i.e. active light incident to the sensor; front and side view (left), after a distance / intensity calibration (right).

for $(i, j) = (2, 2), \ldots, (p-1, q-1)$. Extending matrix A by this smoothing constraints, we get

$$\mathrm{A}_\nabla \cdot \mathbf{c} = \begin{bmatrix} \mathrm{A} \\ \lambda \mathrm{L} \end{bmatrix} \cdot \mathbf{c} = \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix} \tag{2.23}$$

where the sub-matrix L contains the Laplace-constraints described in (2.22) and $\lambda > 0$ controls the amount of smoothing. The extended linear equation system can be solved using a least square optimization, finally yielding a desired correction function for our coupled calibration model.

Analog to the original calibration model in Sec. 2.2.2, the distance adjustment can be further improved by likewise taking individual pixel characteristics into account. Originally, the reference distance has been used to calculate the deviation between each PMD pixel and the optimized average pixel $\mathbf{p}_\varnothing$. However, as no reference value for the intensity parameter is known, no canonically extension of the pre-adjustment for the intensity values exists. For this reason, an asymmetric pixel-wise pre-adjustment similar to the original approach is performed that takes distance information into account only, i.e.

$$m_c(\mathbf{p}) = P(m_\varnothing, h) \quad \text{with} \quad m_\varnothing = l_\mathbf{p}(m(\mathbf{p})). \tag{2.24}$$

In order to determine the improvement of the enhanced calibration model, both methods (distance only and coupled) have been used to adjust a set of distance information of the same set of input data. In the case of pure distance adjustment, the B-spline has been fitted to the deviation data of the high reflective panel, which possess the lowest noise interferences. This correction is applied to the complete input data, independent of the actual light incident to a sensor pixel.

Tab. 2.4 shows the mean deviation error for both methods with respect to different reference distances. For the coupled approach 10, respectively 5, control points for the distance and the intensity component as well as a smoothing factor of $\lambda = 1$ has been used. It can be seen, that the original calibration model, which handles distance information only, still adjusts the high reflective images correctly, but fail on the images with low reflectivity. In contrast, the enhanced calibration model, which takes distance and intensity into account, is able to correct all images independent from the incident light (cmp. Fig. 2.9).

Distance	[m]	**0.9**	**1.1**	**1.3**	**1.7**	**2.1**	**2.5**	**3.0**	**3.5**	**4.0**
Unadj.	⊘	15.77	15.76	13.14	8.88	6.53	11.02	10.64	3.08	3.54
Wiggling	⊘	1.94	2.82	2.10	2.89	1.35	1.73	1.59	2.18	2.67
	100%	0.10	0.21	0.30	0.21	–	–	–	–	–
	30%	2.53	3.50	3.24	3.72	–	–	–	–	–
Coupled Model	⊘	0.35	0.38	0.64	0.97	1.00	1.52	1.13	1.18	1.73
	100%	0.28	0.23	0.31	0.42	–	–	–	–	–
	30%	0.32	0.32	0.40	1.13	–	–	–	–	–

Table 2.4: Average distance deviation in [cm] for a number of reference distances before (Unadj.) and after a distance adjustment has been performed. In the near range, the distance error for 100 % and 30% IR reflectivity are additionally listed. In the uncorrected distance data, the systematic error due to the demodulation is clearly noticeable.

Decoupled Model

Coupled calibration models as previously discussed, generally require a large set of reference data with respect to intensity-distance pairs. In practice, however, the acquisition of such data is rather time consuming and impractical. In order to reduce the number of required reference images, a better solution therefore seems to be a separation of both calibration parameters, i.e. an individual treatment of systematic- and intensity-related deviations.

Regarding (2.17), the theoretical correct order of a separated distance correction is given by

$$\begin{aligned} d &= \text{intensity}^{-1} \oplus \text{wiggling}^{-1} \oplus m \\ &= \text{intensity}^{-1} \oplus m_w \end{aligned} \tag{2.25}$$

Due to the distance dependent light attenuation, however, the reference data for the

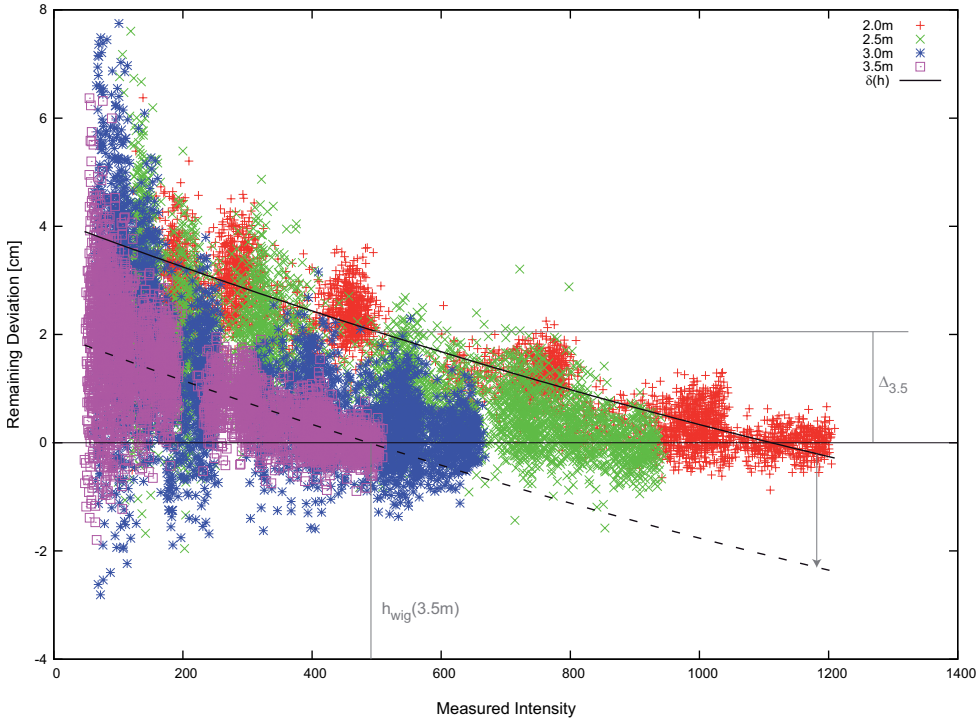


Figure 2.10: Remaining distance deviations after a wiggling adjustment has been applied. The plot is based on six different calibration targets, which has been captured at four distances.

wiggling correction undergoes intensity changes as well. For this reason, intensity-related deviations get already partially corrected by the wiggling adjustment, which becomes obvious by plotting the remaining distance deviations (see Fig. 2.10). Here, all deviations basically exhibit the same characteristics, but are vertically displaced by a offset Δ_d, reducing the deviation of the most reflective target to zero.

In the case that the offsets are known, the remaining intensity-related deviations can be generally corrected by an overall, one-dimensional correction term $\delta(h)$ that depends on the given intensity only. Fortunately, the same correction term also provides the required offsets, as for a given reference distance d, Δ_d is given by $\delta(h)$ regarding the intensity value $h_{\text{wig}}(d)$ that has been used for the wiggling calibration (cmp. Fig 2.10). The error adjustment is therefore finally given by

$$m_c(\mathbf{p}) = m^w_{\mathbf{p}} + \delta\Big(h_{\text{wig}}\left(m^w_{\mathbf{p}}\right)\Big) - \delta(h_{\mathbf{p}}). \tag{2.26}$$

where $\delta(h_{\text{wig}}(m^w_{\mathbf{p}}))$ accounts for the vertical pre-adjustment and $\delta(h_{\mathbf{p}})$ for the actual intensity-related deviations.

By pre-adjusting the according offsets, both error sources (wiggling- and

intensity-related) can be treated independently in a two step approach, which significantly reduces the number of required reference images: For the intensity-related distance adjustment term $\delta(h)$ it is sufficient to acquire the full intensity range only once for a fixed distance, whereas for the distance dependent offsets only the brightest intensity are required and has been already acquired for the wiggling adjustment. Note, that in order to cover a large intensity range, the chosen distance for the maximum intensity function $h_{\text{wig}}(d)$ should be preferably close to the camera.

Altogether, the decoupled calibration model for intensity-related deviations can be summarized as follows:

1. Determination of the distance-dependent intensity function $h_{\text{wig}}(d)$. For this purpose, it is important that the same reference data as for the demodulation adjustment is used.

2. Determination of the actual intensity-related distance adjustment $\delta(h) = m_w - d$ regarding a given set of reflectivities at a close range.

In our case, both functions , $h_{\text{wig}}(d)$ as well as $\delta(h)$, are modeled by polynomials of degree 2, i.e.

$$h_{\text{wig}}(d) = \sum_{k=0}^{2} a_k^{\text{wig}} \cdot d^k, \quad \delta(h) = \sum_{k=0}^{2} a_k^{\delta} \cdot h^k. \tag{2.27}$$

Light attenuation In practice, the PMD image (and therefore the intensity image as well) is unfortunately affected by a radial light attenuation caused by the optics, for which reason the maximum intensity function as stated in (2.27) cannot be determined globally.

Instead of determining the according function for each pixel individually, the intensity mapping can also be extended by an additional radial attenuation, i.e.

$$h_{\text{wig}}(d, r) = \sum_{k=0}^{n} a_k^{\text{wig}} d^k \sum_{l=0}^{m} b_l^{\text{wig}} r^l = \sum_{k=0}^{n} \sum_{l=0}^{m} \left(c_{kl}^{\text{wig}} d^k r^l \right) \tag{2.28}$$

where r is the euclidian distance to the projection center $\mathbf{c}$ on the image plane. Analogously, (2.26) extends to

$$m_c(\mathbf{p}) = m_{\mathbf{p}}^w + \delta\Big(h_{\text{wig}}\left(m_{\mathbf{p}}^w, \|\mathbf{p} - \mathbf{c}\|\right)\Big) - \delta(h_{\mathbf{p}}, \|\mathbf{p} - \mathbf{c}\|). \tag{2.29}$$

By doing so, both intensity mapping (pre-adjustment) as well as intensity-related distance adjustment can be expressed by a global set of coefficients, which finally reduces the overall number of internal calibration parameters.

Results Tab. 2.5 shows the average deviation error for the decoupled model with respect to different reference distances and reflectivities. As expected, the average deviation for the wiggling corrected data noticeably increases with decreasing reflectivity. Range data which, in contrast, has been adjusted by the decouple model, archives a much smaller average deviation in all cases.

Regarding the parameter coupling, the decoupled model achieves similar results as the coupled model. Only for low reflective targets at larger distances, the coupled model is able to adjust the distance information more accurate. In the near range, on the other hand, the coupled model performs less well due to the additional smoothing constraint.

		100%	**80%**	**60%**	**40%**	**20%**	**0%**
	Wiggling	0.1073	0.8905	1.6932	2.4432	2.7929	2.7218
1.0 m	Coupled	0.1973	0.1918	0.2714	0.2838	0.3498	0.7179
	Decoupled	0.1325	0.1239	0.1470	0.2032	0.3627	0.7572
	Wiggling	0.1088	0.7991	1.4507	1.6025	2.0964	2.5058
1.4 m	Coupled	0.3619	0.3220	0.3809	0.3715	0.6800	1.4711
	Decoupled	0.1095	0.1974	0.3848	0.4257	0.8659	1.7188
	Wiggling	0.2170	0.5018	0.7423	1.12713	2.1967	3.5027
1.8 m	Coupled	0.6699	0.5972	0.6131	0.7908	1.5728	3.4599
	Decoupled	0.2483	0.3338	0.4809	0.9292	1.8140	3.3225
	Wiggling	0.2438	0.6284	1.1121	1.7288	2.9523	5.0360
2.2 m	Coupled	0.9506	0.9824	0.8875	1.2031	2.0513	3.3471
	Decoupled	0.2257	0.5347	0.9104	1.5165	2.8597	4.9302

Table 2.5: Mean distance deviation in [cm] for a different number of reference distances and reflectivities regarding pure wiggling correction (⌀ 1.62 cm), coupled (⌀ 0.97 cm) as well as decoupled (⌀ 0.98 cm) intensity-related adjustment.

2.3 Motion Compensation

Regarding common vision systems, a large number of interactive applications deal with dynamic scenes consisting of moving objects or camera movements. However, due to the cameras' working principle based on subsequent phase images, fast motion during the acquisition typically leads to artifacts as classified in Sec. 1.2.2. Beside varying phase samples along the viewing ray, the most dominant impact can be ascribed to lateral motion and the fixed per-pixel sampling scheme, which assumes a steady scene during the integration process. For this reason, a compensation approach is discussed in the following section that breaks up

the fixed sample scheme and realigns corresponding phase images. By doing so, most motion artifacts due to lateral motion as well as texture changes can be reduced. The realignment is achieved by applying optical flow estimations to each individual phase image, i.e. by tracking individual surface points between phase images over time.

In addition, as pixel tracking is unable to deal with axial motion along the viewing direction, the impact of axial motion is discussed. Accordingly, based on the standard phase shift demodulation, a theoretical model for additional motion dependent offsets is proposed.

2.3.1 Optical Flow-Based Phase Image Registration

The problem of motion artifacts basically arises from unmatching phase images due to the demodulation of unrelated phase samples of different object or textual changes and the corresponding change in incident light. The main idea behind the proposed compensation approach therefore is to virtually discard the fixed per-pixel sampling and select correct phase samples by realigning corresponding phase images.

For the image realignment, a various number of registration approaches exist, which mainly consist of two basic steps [Zit03]:

- feature detection and mapping
- estimation of the according image transformation.

Both steps are crucial for the registration results and have to be adapted to the individual case.

Regarding the first step, registration methods can be classified into intensity-based or feature-based algorithms. Intensity-based approaches compare intensity patterns with respect to a given correlation metrics, while feature-based methods are designed to identify and map significant points, lines and / or contours in both images.

In the context of image transformation, registration approaches differ in global linear transformations models (like translation, rotation, scaling or other affine transformations) or local nonrigid transformations like radial basis functions (thin-plate or surface splines) and elastic deformation models including fluid- and optical-flow-based registration. As we want to archive an accurate pixel-wise mapping that is able to deal with object deformation and multiple motion, the most suitable approach in this context is given by optical-flow-based registration as it provides the most adequate results without explicit feature detection. An overview of optical flow techniques is given in Sec. 1.4.

Applying optical flow to range images in combination with variable sampling positions, however, requires the two following conditions to be fulfilled:

- As mentioned in Sec. 1.4, most variational optical flow implementations assume that corresponding surface points appear with the same intensity in subsequent images. This means, that moving objects have to appear with the same intensity in each phase image (Brightness Constraint).
- Applying the demodulation at different pixel locations requires unbiased phase samples, i.e. a homogeneous sensor behavior, in order to get the correct phase shift (Pixel Homogeneity).

Both conditions are a priori not met by standard phase images due to the sampling of a harmonical function itself as well as illumination and sensor inhomogeneities. In the following, we discuss how both pre-conditions can be satisfied.

Brightness Constraint An important precondition of optical flow is the assumption of constant intensity values between consecutive images. Unfortunately, by taking a look at the phase images I_i, it becomes obvious that objects appear differently in each phase image due to the internal phase shift and the applied convolution (see Fig. 2.11, top). However, as mentioned in Sec. 1.2, 2-tap PMD sensors actually measure two phase samples at a time

- the shifted reference signal yielding $A_i = c(\tau_i)$ as well as
- the inverted signal yielding $B_i = c(\tau_{i+2})$.

Commonly, both raw images are internally subtracted to form the actual phase image $I_i = A_i - B_i$ in order to reduce production-specific pixel behavior, whereas the pixel intensity is given analog to (1.5), p. 9, by

$$h = 1/8 \cdot \sum_{i=0}^{3} (A_i + B_i). \tag{2.30}$$

As both signals are inverse to each other, the absolute amount of incident light – and thus the total intensity – for a phase image I_i^+ can be computed by the sum of its raw images (see Fig. 2.11, bottom), making optical flow estimation generally applicable.

Pixel Homogeneity In practice, pixel gain differences as well as a radial light attenuation towards the image border affects the phase values (see Fig. 2.12, left column). Regarding the fixed sampling scheme, these individual pixel characteristics are simply ignored, as they have no impact on the local demodulation. For the realigned demodulation, however, the inhomogeneity not only influences the optical flow estimation in a negative way by violating the constant intensity assumption, it also leads to non-matching raw values during the realigned demodulation since differently biased pixel values are combined.

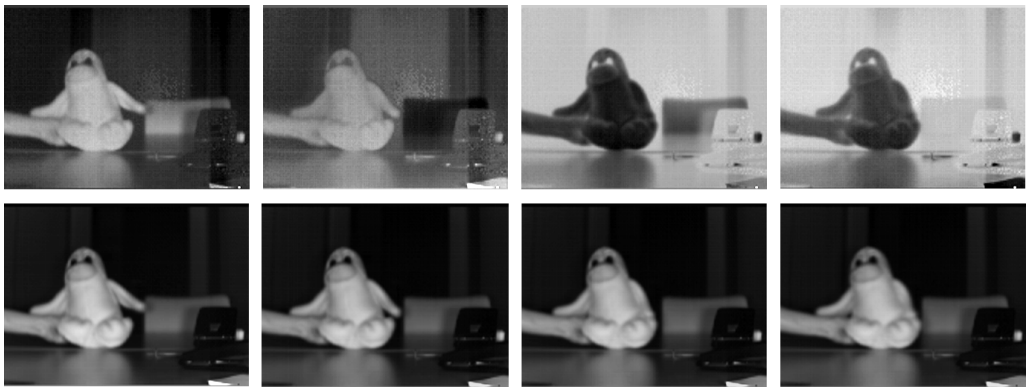


Figure 2.11: Visual difference between the original phase image $I_i = A_i - B_i$ (top row) and the phase image intensity $I_i^+ = A_i + B_i$ (bottom row).

In order to adjust pixel inhomogeneities and consequently improve the motion compensation, an intensity normalization published by Stürmer et al. [SPH08] has been adapted. Basis for the adjustment is a set of reference images that covers varying reflectivity at different distances. Analog to Stürmer's approach, which compares the intensity values of arbitrary image pixels with a reference pixel in the center of the image, our raw value adjustment is obtained by a pixel-wise fit of

$$f_A(A_i) = A_i', \;\; f_B(B_i) = B_i' \text{ with } i = 0 \ldots 3 \tag{2.31}$$

minimizing

$$\sum_{i=0}^{3} (A_i' + B_i') \stackrel{!}{=} h_{\text{ref}}, \tag{2.32}$$

where the reference intensity h_{ref} is given by the brightest pixel in the according reference image, i.e. the pixel concerning to the brightest pixel in the most reflective reference image. In presence of noise, the reference intensity should be determined over a small neighborhood.

Assuming a continuous behavior of the sensor gates, the correction term $f_X(X_i)$ should be smooth and monotonically increasing. This assumption has been confirmed by our experiments as shown in Fig 2.14. We therefore decided to fit a function of logarithmic form, i.e. $f_X(X_i) = c_1\sqrt{X_i + c_2} + c_3 X_i + c_4$, yielding the homogenization results as shown on the right side of Fig. 2.12.

Results The lateral motion compensation via flow estimation has been finally realized using the isotropic, flow-driven optical flow estimation of Zach et al. [ZPB07]. Zach's GPU-based implementation allows discontinuity preserving TV-L1 flow estimation in real time and currently holds the second place of the Middlebury's optical flow ranking [MDB]. As the quality of the motion compensation relies on

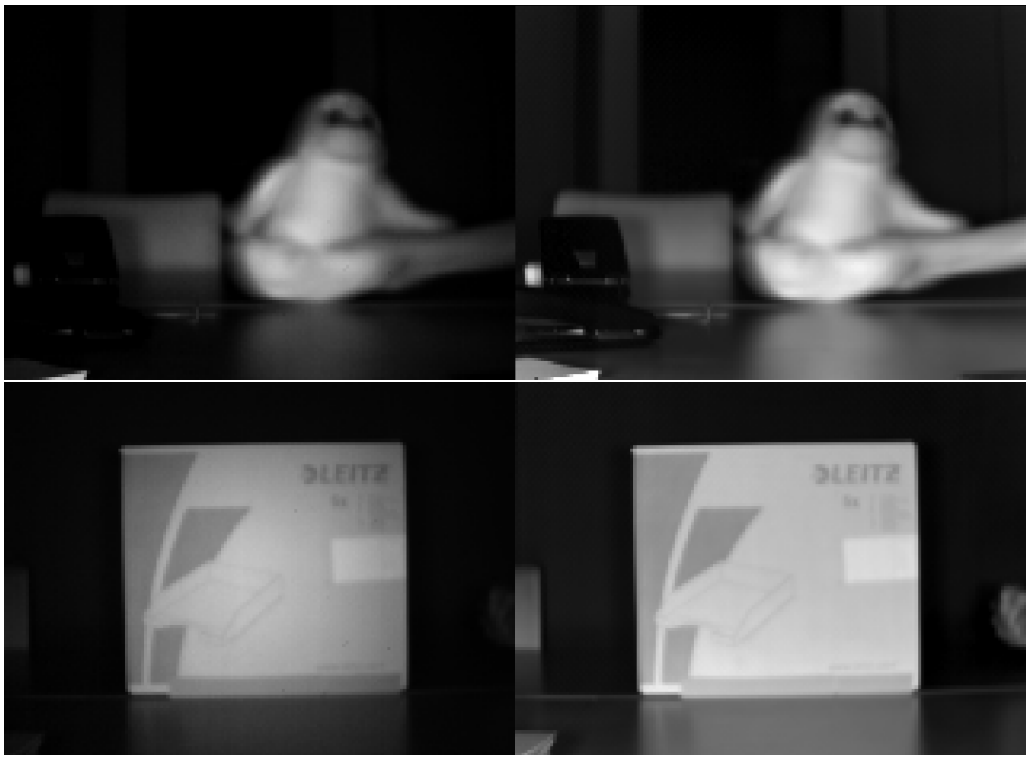


Figure 2.12: PMD intensity image before (left) and after (right) a pixel homogeneity adjustment. Note the strongly varying intensity towards the image border without correction.

the underlying flow estimation, our choice should give the best results in respect to the required accuracy and runtime currently possible.

The lateral compensation is actually done by estimating the optical flow between the phase intensity images I_1^+, I_2^+, I_3^+ and I_0^+ and an according resampling of the corresponding phase images. Before, all raw images are adjusted to eliminate brightness and pixel inhomogeneities. The resampled phase images are then further processed by the fixed standard demodulation scheme in order to calculate the final distance information. A complete system overview, including an optional axial motion compensation step (described in the next section), is given in Fig. 2.13.

The motion compensation has been tested on two exemplary scenes and achieved a frame rate of 10 fps on a Nvidia GeForce 8800 GTX. Whereas *Scene 1* consists of a simple moving box yielding only lateral motion artifacts (see Fig. 2.15), *Scene 2* consist of a more complex moving soft toy, allowing additional deformation (see Fig. 2.11 and 2.16). In both cases, motion artifacts occurred due to contour or texture changes and their corresponding phase mismatch.

By applying the lateral compensation approach, most of the arising artifacts

has been satisfactorily removed. Especially the comparison with the static scene, shows that the re-sampled box for example matches the reference scene very well in distance information and object size.

A statistical evaluation of the box scene is given in Tab. 2.6. The detection of box and background pixels (wall) is done using a clustering approach with respect to a region growing based on plane fitting. It can be seen, that the number of detected pixels is extremely close to the static situation. The variance in the data decreases by applying the motion compensation. No shift in the average distance for the detected box or background pixels occur. Note, that the texture-related errors in the object region are not captured by the variation, since these pixels are classified as outliers.

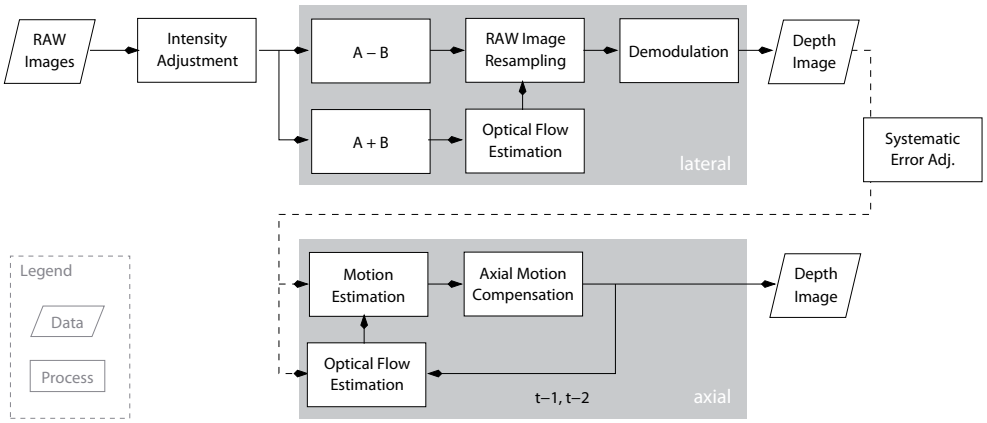


Figure 2.13: System Overview. The motion compensation consists of two consecutive modules: the lateral motion and an optional axial motion compensation. Note, that an additional systematic error correction is necessary to adjust demodulation errors before a correct velocity estimation can be performed.

		Static	Motion	Adjusted
Pixel Count	Background	7326	6904	7213
	Object	8521	6609	8458
	Outliers	3033	5367	3209
Mean Dist. [mm]	Background	2404	2405	2405
	Object	1271	1272	1272
Variance [mm]	Background	13.6	14.6	14.0
	Object	8.5	9.6	7.9

Table 2.6: Statistical analysis of *Scene 1* stating the number of object and background pixel as well as the corresponding average distance and its variance.

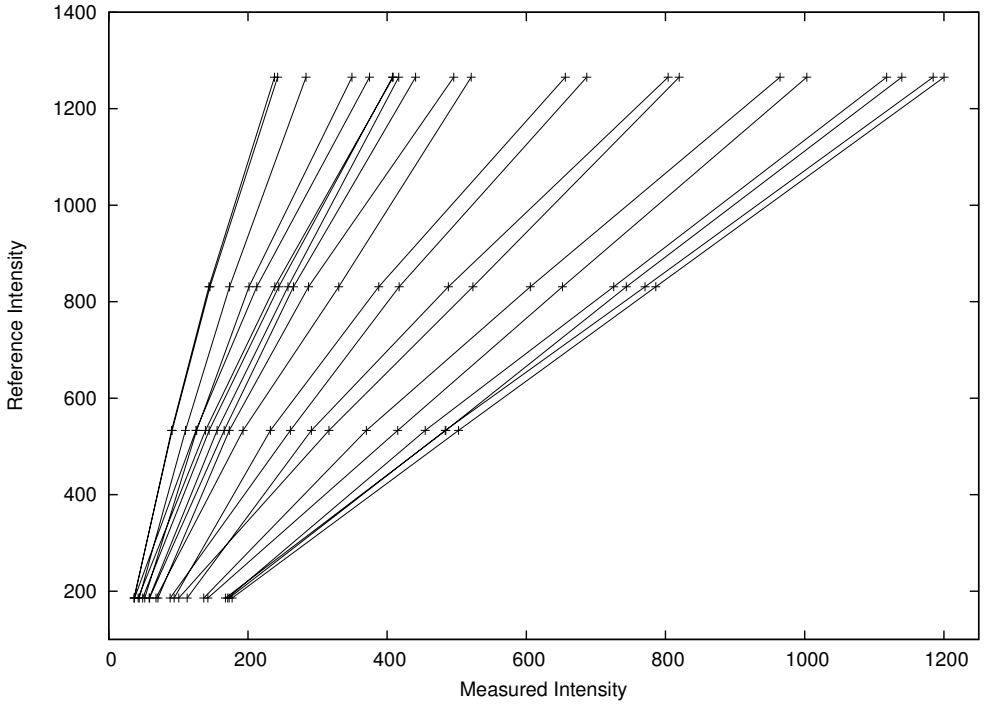


Figure 2.14: Intensity normalization for 20 randomly chosen pixels. The diagram shows the measured intensity ($\sum_{i=0}^{3} A_i$) compared to the reference intensity of the brightest pixel h_{ref}.

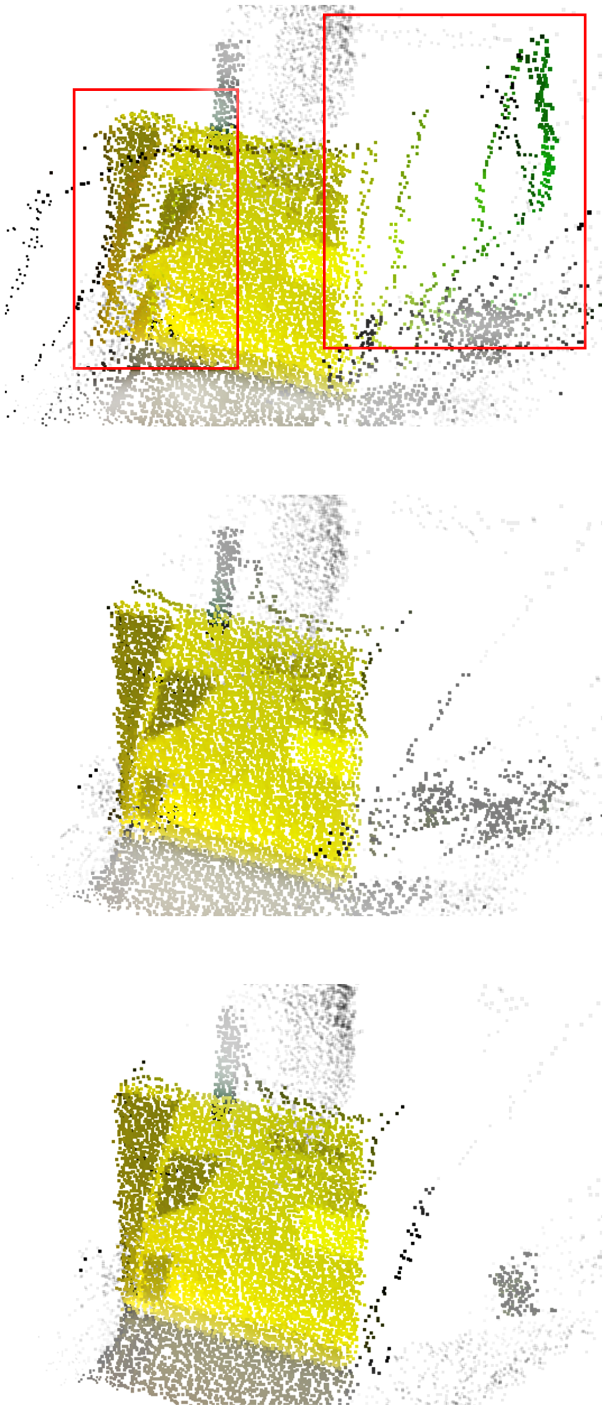


Figure 2.15: *Scene 1* before (top) and after motion compensation (middle) as well as the static reference scene (bottom). Note the motion artifacts due to texture changes of the box surface.

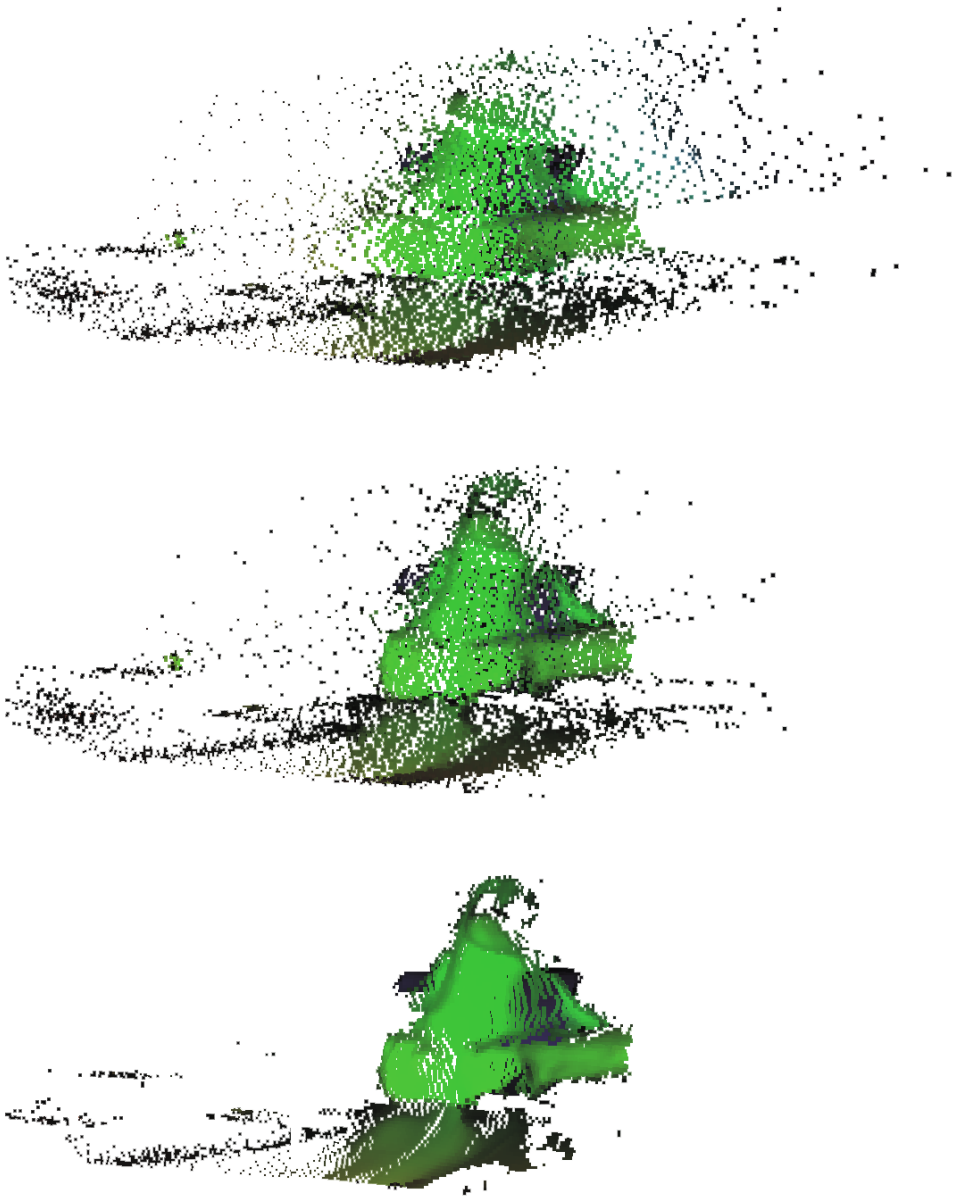


Figure 2.16: *Scene 2* before (top) and after motion compensation (middle) with additional outlier removal and bilateral filtering (bottom). Note the reduction of artifacts as well as the sharpened object features.

2.3.2 Axial Motion Impact

While optical flow is able to handle motion artifacts caused by lateral object shifts, it is unable to reduce the impact of axial motion along the viewing ray. Unlike lateral motion, which simply results in a displacement of corresponding phase values, axial motion introduces additional phase changes due to the varying object distance. The following section addresses the theoretical impact of axial motion onto the demodulation results. It first deals with a simple axial motion model that assumes a constant displacement between two phase images. The simple model is then further extended to account for an uniform, continuous motion during the integration series.

Assuming an uniform axial motion with an axial displacement Δ between the first and last phase image I_0 and I_3, the theoretical correlation samples are given analog to (1.2), p. 8, by

$$I_i = \frac{a}{2}\cos\left(i \cdot \frac{\pi}{2} + \phi_d + i \cdot \frac{\phi_\Delta}{3}\right) \tag{2.33}$$

where ϕ_d represents the initial distance at the start of the range image acquisition. Applying the demodulation scheme from (1.3) yields the phase offset

$$\begin{aligned}\phi_m &= \operatorname{atan2}\frac{\sin(\phi_d + \phi_\Delta) + \sin\left(\phi_d + \frac{1}{3}\phi_\Delta\right)}{\cos(\phi_d) + \cos\left(\phi_d + \frac{2}{3}\phi_\Delta\right)} \\ &= \operatorname{atan2}\frac{\sin\left(\phi_d + \frac{2}{3}\phi_\Delta\right)}{\cos\left(\phi_d + \frac{1}{3}\phi_\Delta\right)}.\end{aligned} \tag{2.34}$$

Consequently, if Δ and ϕ_m are known, the true phase offset ϕ_d can be calculated by

$$\phi_d = \operatorname{atan2}\frac{\sin(\phi_m)\cos\left(\frac{1}{3}\phi_\Delta\right) - \cos(\phi_m)\sin\left(\frac{2}{3}\phi_\Delta\right)}{\sin(\phi_m)\sin\left(\frac{1}{3}\phi_\Delta\right) + \cos(\phi_m)\cos\left(\frac{2}{3}\phi_\Delta\right)}. \tag{2.35}$$

However, instead of integrating over one period, current PMD cameras repetitively accumulated short-time integrated samples with an integration time of $T/2$, where $T = 1/f$. Accordingly, several distance shifts are summed up during the overall long-time integration.

For a more accurate model, the short-time integrated sample $A_{i,j}$ for τ_i of an incident signal with additional offset K and amplitude a can be formulated by

$$\begin{aligned}A_{i,j} &= \int_0^{T/2} K + a\sin\left(\omega t - \left(\phi_d + i\frac{\phi_\Delta}{3} + \kappa j\right) - i\frac{\pi}{2}\right) dt \\ &= \frac{a}{\pi f}\cos\left(\phi_d + i\frac{\phi_\Delta}{3} + \kappa j + i\frac{\pi}{2}\right) + \frac{K}{2f}\end{aligned} \tag{2.36}$$

where $i \cdot \Delta/3$ accounts for the distance offset at the beginning of each phase image analog to (2.34) and $\kappa j = j \cdot \phi_\Delta / 3N$ represents the additional offset for each short-time integration. Note that during each short-time integration, the distance is approximated to be constant. By accumulation all N short-time integrated sample $A_{i,j}$ we get the final phase sample

$$A_i = \sum_{j=0}^{N-1} A_{i,j} = \frac{a}{\pi f} \sum_{j=0}^{N-1} \cos\left(\phi_d + i\frac{\phi_\Delta}{3} + \kappa j + i\frac{\pi}{2}\right) + \frac{NK}{2f} \tag{2.37}$$

Regarding the final phase demodulation, all constant quantities ($NK/2f$ and $a/\pi f$) are of no relevance and thus can be neglected for further calculations, yielding

$$\begin{aligned} A_i &= \sum_{j=0}^{N-1} \cos\left(\phi_d + i\frac{\phi_\Delta}{3} + \kappa j + i\frac{\pi}{2}\right) \\ &= \sum_{j=0}^{N-1} \left(\cos\left(\phi_d + i\frac{\phi_\Delta}{3} + i\frac{\pi}{2}\right)\cos(\kappa j) - \sin\underbrace{\left(\phi_d + i\frac{\phi_\Delta}{3} + i\frac{\pi}{2}\right)}_{\varphi}\sin(\kappa j) \right) \\ &= \cos(\varphi) \sum_{j=0}^{N-1} \cos(\kappa j) - \sin(\varphi) \sum_{j=0}^{N-1} \sin(\kappa j) \\ &= c_1 \cos(\varphi) - c_2 \sin(\varphi) \end{aligned} \tag{2.38}$$

Finally, by applying the demodulation stated in (1.3), p. 9, we obtain:

$$\begin{aligned} A_3 - A_1 &= c_1 \cos\left(\frac{3}{2}\pi + \phi + \phi_\Delta\right) - c_2 \sin\left(\frac{3}{2}\pi + \phi + \phi_\Delta\right) \\ &\quad - c_1 \cos\left(\frac{\pi}{2} + \phi + \frac{\phi_\Delta}{3}\right) + c_2 \sin\left(\frac{\pi}{2} + \phi + \frac{\phi_\Delta}{3}\right) \\ &= 2\cos\left(\frac{\phi_\Delta}{3}\right)\left(c_1 \sin\left(\phi + \frac{2\phi_\Delta}{3}\right) + c_2 \cos\left(\phi + \frac{2\phi_\Delta}{3}\right)\right) \end{aligned} \tag{2.39}$$

and

$$\begin{aligned} A_0 - A_2 &= c_1 \cos(\phi) - c_2 \sin(\phi) \\ &\quad - c_1 \cos\left(\pi + \phi + \frac{2\phi_\Delta}{3}\right) + c_2 \sin\left(\pi + \phi + \frac{2\phi_\Delta}{3}\right) \\ &= 2\cos\left(\frac{\phi_\Delta}{3}\right)\left(c_1 \cos\left(\phi + \frac{\phi_\Delta}{3}\right) - c_2 \sin\left(\phi + \frac{\phi_\Delta}{3}\right)\right) \end{aligned} \tag{2.40}$$

yielding the finally measured phase offset

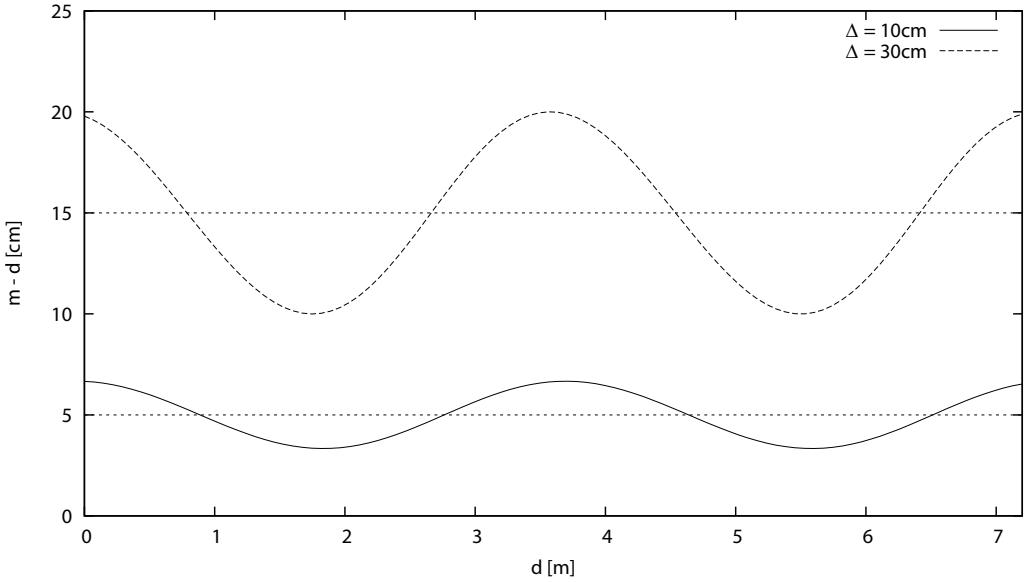
$$\phi_m = \frac{A_3 - A_1}{A_0 - A_2} = \frac{c_1 \sin\left(\phi + \frac{2}{3}\phi_\Delta\right) + c_2 \cos\left(\phi + \frac{2}{3}\phi_\Delta\right)}{c_1 \cos\left(\phi + \frac{1}{3}\phi_\Delta\right) - c_2 \sin\left(\phi + \frac{1}{3}\phi_\Delta\right)} \tag{2.41}$$

whose characteristics are similar to (2.34) but comprises an additional translation (see Fig. 2.17). Analog to (2.35), the inverse of (2.41) can be obtained applying the addition theorem.

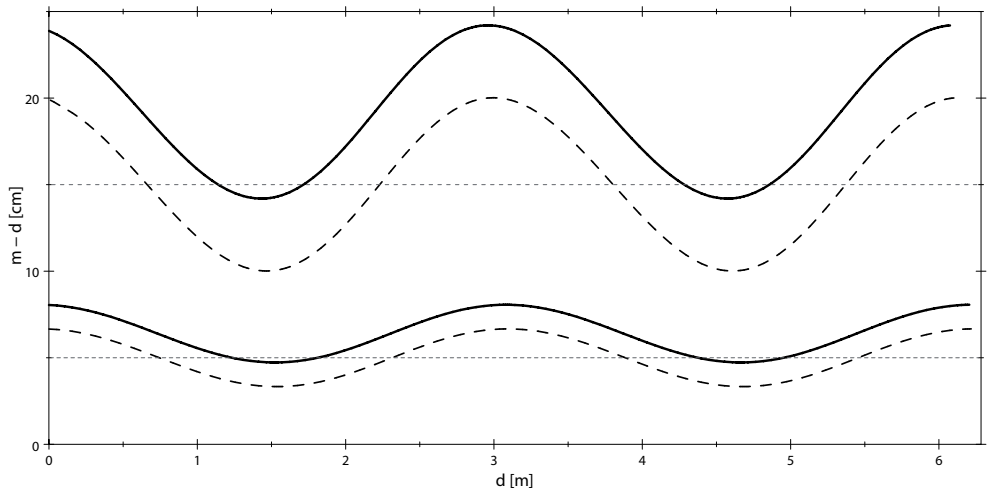
Results By taking a look on the deviation between the theoretically measured distance $d(\phi_m)$ and its ground truth distance d, it becomes obvious that the deviation smoothly fluctuates about $\Delta/2$ with an amplitude of $\Delta/6$ not causing any serious distance jumps up to $\Delta = 3.75$, i.e. half the unambiguousness range. The measured distance information therefore lies more or less half way between the initial and end distance of the integration process. Furthermore, regarding a common integration time of 15 ms per phase image, the displacements in Fig. 2.17 already imply a rather high velocity of 8 km/h, respectively 24 km/h. For most scenarios, axial motion consequently has less significant impact on the distance accuracy than lateral motion and therefore might be neglected, as the more critical resizing of object contours is already handled by the optical-flow-based pixel alignment.

However, for depth image sequences an optional compensation of axial motion errors based on the inverse of (2.41) can be applied if aspired. In this case, object velocities are estimated via the previous two corrected depth images. Knowing the velocity of a surface point, its theoretical deviation as well as its according correction can be derived from (2.41). The velocity estimation is done by using optical flow once more to track individual surface points between depth images. Note, that for proper estimations, the camera's systematic demodulation error must be corrected first (see Sec. 2.2.2).

For a single depth image, e.g. when motion estimation is not possible, a synthesized intensity image halfway between I_1^+ and I_2^+ rather than I_0^+ should be considered as optical flow reference. Otherwise, the object contour will not match its distance information due to the theoretical distance shift of approximately $\Delta/2$, causing the object to shrink or expand.



(a) Distance deviation for a total displacement of $\Delta = 10\,\text{cm}$ (solid line) and $\Delta = 30\,\text{cm}$ (dashed line) with respect to the simple model.



(b) Comparison between the simple (dashed lines) and the advanced axial-motion model (solid line), covering a displacement of $\Delta = 10\,\text{cm}$ (lower lines) as well as $\Delta = 30\,\text{cm}$ (upper lines).

Figure 2.17: Theoretical axial motion impact regarding (a) the simple and (b) the advanced model.

Chapter 3

Data Processing

»One and one is two, and two and two is four, and five will get you ten if you know how to work it.«

– Mae West

As mentioned in the Sec. 1.2.2, range images as acquired by current time-of-flight cameras are quite noisy and of rather low resolution. The following sections therefore covers essential processing techniques that can be used to enhance distance informations with respect to further processing steps. The overall structure is organized into three parts. The first two parts cover the general task of image denoising and the algorithmical refinement of distance information (Sec. 3.1 and Sec. 3.2). Part three deals with the fundamental process of data fusion for multi-modal imaging systems, i.e. systems that use more than one image sensor to acquire scene information / knowledge (Sec. 3.3).

All outlined algorithms has been implemented as part of a generic processing framework for mixed CPU / GPU computations. Conceptional details of the framework are given at the end of this chapter in Sec. 3.4.

Publications The edge preservative upsampling filter has been published in [LLK08], while the sensor fusion approach first has been presented in [LK07b].

3.1 Denoising

In Sec. 1.2.2, different noise sources has been classified that affected the accuracy of current TOF cameras. While most of these sources can be suppressed by proper cooling and signal processing techniques, the most prominent noise source depends on the number of incoming photons and can only be handled by adequate post-processing.

Image denoising, as one of the classical problems in signal processing, has been studied with intensive care. With introduction of range sensing techniques, most of the two-dimensional algorithms have been adapted to mesh and range data smoothing in the three-dimensional domain, but mostly really on supplemental normal information. In addition, so-called projection-based approaches have recently attracted the interest of many researchers. Here, range data denoising is done by fitting either implicit (so-called level sets) or explicit (e.g. polynomial

or B-spline-based) surface patches through the given, three dimensional point cloud [CBM*03, SBS05, LCLT07]. Unlike classical average filters projection-based techniques however generally involve non-linear or iterative optimization schemes. Furthermore, mostly all projection-based techniques suffer from over-smoothed object features due to the non-trivial task of defining a proper neighborhood relation.

Consequently, as the considered noise solely occures in viewing direction, range images are commonly denoised in the two-dimensional image space using classical noise filter (see below). The so far only sophisticated and in contrast non-averaging approach for TOF range images has been published by Böhme et al. [BHMB08]. The presented approach exploits the fact that TOF cameras provide both distance as well as intensity information that are not independent but linked by a shading constraint. Based on the Lambertian reflectance model, their smoothing approach iteratively optimizes the maximum likelihood estimate to observe a scene with distance information R and reflectivity (albedo) properties A for a given range image X_m and intensity values X_h, i.e. $p(X_m, X_h|R, A)p(R)p(A)$. Required normals are maintained by triangulating the range image in each iteration step. Even though the approach gives impressive results, it is generally computation intensive requiring approximately two seconds per frame. Consequently, it is inappropriate for real-time processing.

Considering the two-dimensional image space, range image denoising can be split up into temporal and spatial filtering as discussed next.

Temporal Denoising For temporal denoising, a sequence of subsequent images is processed by either linear averaging or non-linear median filtering. The according expectation value of the noise component is assumed to be zero and normal distributed.

Regarding the demodulation process of current TOF cameras, the temporal denoising can be applied to either the intermediate phase or the final range image. Compared to range image denoising, however, the processing of phase values I_i brings a clear advantage [Rap07]: By processing phase values, erroneous correlation samples have much less influence on the averaging process as they would have by simply averaging the falsified distance information – especially in the case of poorly exposed pixels. In addition, the computation effort is strongly reduced as (1.2), p. 8, has to be evaluated only once after the average determination instead of each frame.

While temporal denoising is well suited to obtain more accurate, static reference data for, e.g., distance calibration and fixed pattern noise, it is mostly impractical for dynamic scenes. Not only does the processing of image sequences reduce the frame rate, it also introduces additional ghosting and blurring artifacts.

For this reason, distance information is usually denoised with respect to a single image, while the averaging criteria is applied to a local neighborhood assuming the underlying signal to be constant or smooth.

Spatial Denoising As already mentioned, spatial denoising of image data has been studied for several decades, leading to a variety of well-established techniques, e.g., based on diffusion processes [Wei98], wavelet methods [RKN00] or total variation [BKP10, ROF92] as well as adaptive linear and non-linear filters. However, most of them are rather complex and consequently rather inapplicable for real-time processing tasks.

A simple but popular smoothing technique, is the Gaussian low-pass filter

$$G_\sigma(\mathbf{p},\mathbf{q}) = \frac{1}{2\pi\sigma^2} \cdot \exp\left(-\frac{\|\mathbf{p}-\mathbf{q}\|^2}{2\sigma^2}\right) \tag{3.1}$$

which, for efficiency, can be separated into two one-dimensional filters. However, convolution with a Gaussian kernel incorporates spatial information only, disregarding any confidence information about the given data (which in our case, for example, is given by the signal's amplitude). Erroneous pixel therefore contribute in the same way to the result as the rest of the neighborhood does.

For more accurate results, normalized convolution [KW93] can be used to incorporated the signal reliability into the average process. Here, the image data (distance information m) is additionally weighted by a supplemental confidence measure w and re-normalized to preserve its energy, i.e.

$$m' = \frac{G_\sigma * (m \cdot w)}{G_\sigma * w} \tag{3.2}$$

where $*$ stands for the convolution and $\cdot$ represents a scalar multiplication. This way, distance information that is more reliable maintains a greater influence to the average distance, while unreliable pixels (including flying pixels) and marked outliers have lower or no impact. Frank et al. [FPR*09] recently showed, that the appropriate confidence measure, i.e. distance deviation, is related to the inverse square of the amplitude, that is $w = a^2$.

Even though (3.2) gives already good results with respect to unreliable data (see Fig. 3.2, p. 70), the involved Gaussian filter is known to blur image features, i.e. distance discontinuities and contours in the range image. Furthermore, due to the dependency between signal amplitude and object distance, normalized convolution near distance discontinuities generally cause nearby object boundaries to grow, while boundaries of objects further away simultaneously shrink. A behavior that also has been marked out by Schmidt [Sch08]. For this reason, preferably edge preserving filter techniques should be applied.

A quite popular edge preserving alternative to Gaussian kernel is the *bilateral filter* [TM98], which extends (3.1) by an additional semantic/similarity term, i.e.

$$B_{\sigma,\nu}(\mathbf{p},\mathbf{q}) = \exp\left(-\frac{\|\mathbf{p}-\mathbf{q}\|^2}{2\sigma^2}\right) \cdot \exp\left(-\frac{\Psi^2(\mathbf{p},\mathbf{q})}{2\nu^2}\right) \tag{3.3}$$

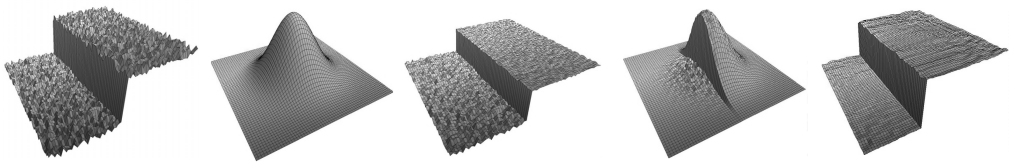


Figure 3.1: Bilateral filtering (from left to right): input, smoothing kernel, similarity measure, combined kernel, output [DD02].

where $\Psi(\mathbf{p}, \mathbf{q})$ is an appropriate similarity measure between both pixels $\mathbf{p}$ and $\mathbf{q}$. For range images, the pixel similarity can be defined, for example, via the pixel's intensity information and / or the distance information itself. Especially the latter prevents distinctive pixels (especially outliers) to be combined (robust estimation).

Compared to Gaussian filter, range features are much better preserved by bilateral filtering as can be seen in Fig. 3.1. Actually, it has been shown that bilateral filtering and the more sophisticated anisotropic diffusion are related through adaptive smoothing [Bar02] and belong to the same family of robust estimators [DD02]. However, in contrast to diffusion filtering, bilateral filtering is not energy preserving due to the asymmetric normalization term.

In the context of multi-modal data (cmp. Sec. 3.3), bilateral filtering can be further extended to cross-filtering, where the filter weights are based on the similarity determined on one or multiple different data sources, e.g., additional color information.

A more sophisticated extension of the bilateral filter is given by the *non-local means filter* [BCM05], where the idea of pixel similarity is extended to an according neighborhood similarity, i.e.

$$NL_{\sigma,\nu}(\mathbf{p}, \mathbf{q}) = \exp\left(- \sum_{k \in \Omega} \frac{G_\sigma(\|\mathbf{k}\|) \cdot \Psi^2(\mathbf{p} + \mathbf{k}, \mathbf{q} + \mathbf{k})}{\nu} \right) \tag{3.4}$$

where Ω defines a set of neighborhood pixels and $\Psi(\mathbf{p}, \mathbf{q})$ stats a similarity measure analog to (3.3). Thus, only pixels with the same distribution contribute to the smoothing result, i.e. for edge pixels only pixels along the edge are considered, while similar pixels close to the boundary are discarded.

An application of the NL-Means filter to range images has been published by Huhle et al. [HSJS08], who extended the original NL-Means filter by an iterative outlier identification. Essential for the extension is the observation that the NL-Means filter result equals the expectation value for an individual pixel given its neighborhood, i.e. $E(m(\mathbf{p}) \,|\, \Omega_\mathbf{p})$. Based on this observation, Huhle et al. define an inliner probability distribution with respect to the standard deviation σ_m inside the given neighborhood. Pixel whose probability is below 50% are concerned to be outliers. The classification process is iterated several times, while possible

outliers are excluded from the particular calculations, leading to improved outlier estimates with each iteration. Finally, a standard NL-Means average is performed, that excludes identified outliers. However, even with GPU acceleration, the whole filter process takes about two seconds for ten iterations.

Consequently, even though more sophisticated algorithms exist, iterative bilateral filtering currently gives the best, feature preserving results for real-time processing tasks. Nevertheless, as all averaging techniques are usually based on a constancy assumption, it should be regarded that filtering of radial distance information in image space generally introduces some small, mostly negligible bias. This behavior can be explained by the fact, that even the acquisition of a planar surface results in non-constant radial distance information.

Outlier and Flying Pixel Even though, most outliers and flying pixels are generally covered by weighted averaging, it is not guaranteed that the amplitude of these pixels always reflects their reliability in a proper way. Especially outliers due to oversaturation sometimes tend to exhibit a misleading high amplitude. Therefore, a special handling of oversaturated pixel as proposed by Rapp [Rap07] should be additionally performed. Here, the raw values of the readout diodes are considered to identify capacitor overflows. General outliers on the other hand can be additionally discarded if they are too far away from their neighboring pixels in terms of either absolute distances or relative measures regarding the neighborhood mean and variance.

In addition, most flying pixels can be classified by the number of valid neighbor pixels and discarded if the number of nearby pixels is below a given threshold. Alternatively, edge preserving sampling techniques as used in the distance refinement approach, presented in Sec. 3.2.2, can be used to avoid holes in the range image.

According to the system requirements, holes in the range image due to an outlier segmentation might be filled in by resampling techniques that use an explicit or implicit surface representation (see Sec. 3.2.1). For tasks like obstacle detection, however, hole filling is not a good choice as propagated distance information might lead to false assumptions in critical situations.

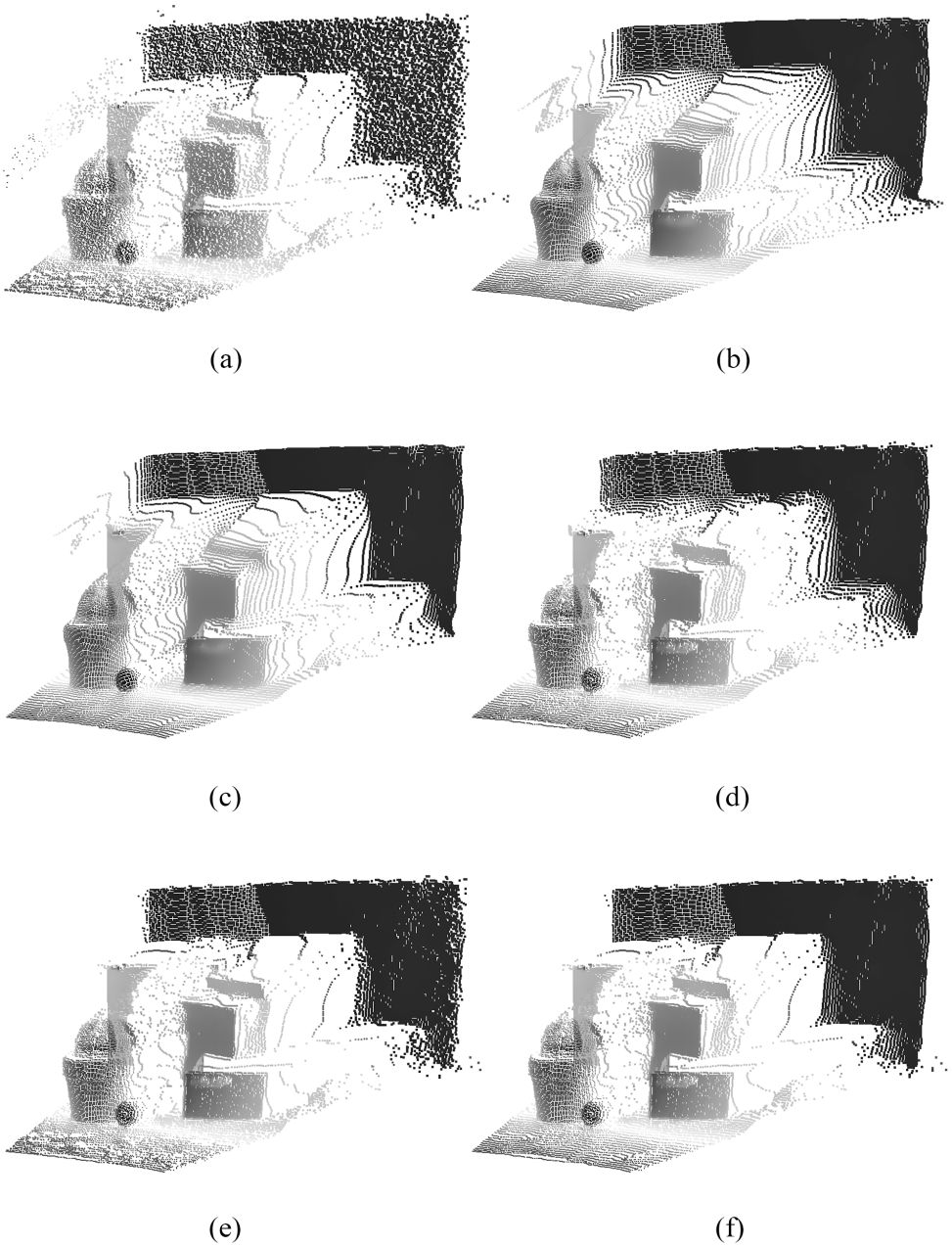


Figure 3.2: Comparison of denoising techniques: (a) unprocessed range image, (b) Gaussian Filter, (c) normalized convolution (weighted Gaussian), (d) Bilateral Filter (intensity-based), (e) and (f) Bilateral Filter (intensity- and distance-based) regarding a threshold of 5 cm, respectively 10 cm.

3.2 Distance Refinement

Even though TOF cameras are able to acquire full-scene information in an adequate time, they are still of low resolution compared to other scanning techniques and thus provide only a sparse point cloud of scene samples. Accordingly, further processing tasks like sensor fusion and even simple visualization of the data suffer from large gaps between the small number of disconnected points. Consequently, many systems incorporate an algorithmical refinement of the provided distance information by synthetically adding new surface points and fill in missing information, i.e. by upsampling the range image.

Image upsampling is a well-understood problem to upscale discrete image data. According to the Nyquist sampling theorem [Sha49] and assuming an ideal sampling rate, the original (one-dimensional) continuous signal can be retrieved from its discrete samples via convolution with the sinc function $\mathrm{sinc}(x) = \sin(x)/x$. In practice, however, images usually exhibits an insufficient sample rate, leading to ringing artifacts due to overlapping frequency spectra [MN88]. Therefore, due to the undersampling and the infinite extend of the sinc function, image upsampling is commonly done by local data interpolation.

In the simplest case, image upscaling is done by bilinear interpolation, which equals a simple quadrilateral mesh reconstruction of the according point cloud. Linear interpolation, however, is only a rough approximation and results in typical diamond artifacts (see Fig. 3.7, p. 82). The following section therefore will discuss two more advanced refinement techniques. At first, in Sec. 3.2.1, the application of a more general, explicit surface fitting approach to range images will be discussed, which is well-known from point-based rendering as *Moving Least Square* surfaces. Secondly, in Sec. 3.2.2, an extension of an edge preserving, biquadratical upscaling technique is introduced, which works in image space and takes advantage of the underlying grid structure of range images.

In contrast to the more general, distance-only techniques, two additional upsampling approaches for TOF range images have been introduced by [DT05] and [YYDN07] that use multi-modal data, i.e. incorporate additional, high-resolution color information. Both techniques exploit the fact that color or brightness discontinuities often co-occur with depth discontinuities. The general idea, therefore, is to split TOF pixel that span an edge in the color image, in order to allow sub-pixel distance information.

Diebel and Thrun [DT05] improve the range image resolution by first reprojecting distance information into the high resolution color image yielding a sparse distance distribution for scattered color pixel. The missing distance information is finally filled in by applying a Markov Random Field (MRF) to propagate distances information with respect to a global smoothness constraint. Here, the color information is used to add additional weights to the smoothness constraint that allow discontinuities between differently textured regions.

A different approach has been published by Yang et al. [YYDN07]. Here the

range image is first upscaled by a simple nearest neighbor interpolation to match the color image resolution. The upscaled range image is then used to build up a cost-volume with respect to a quadratical cost function and a number of uniquely spaced reference distances (buckets) along the TOF camera range. The individual volume layers are smoothed with a bilateral filter incorporating the additional color information to preserve edges. Finally, for each pixel the layer with the smallest error is selected to be the pixel's new distance information. In order to avoid quantization effects, the correct distance is determined by fitting the cost function into the enclosing layer data. The whole process is iterated several times, finally yielding colored, super-resolution range data.

Both approaches indeed give impressing results, but are too computation intensive and therefore inapplicable for most real-time applications. Furthermore, due to the absence of real sensor fusion (cmp. Sec. 3.3), both methods require that the projection centers of both sensors are arranged close to each other in order to reduce a false mapping between both sensors. Hence, the possibilities to arrange both cameras in a camera rig are quite limited. Furthermore, both approaches sometimes tend to produce distance clusters, i.e. preserve (sometimes emphasis) nonexistent discontinuities, where the assumption of co-aligned color and distance discontinuities is not met for highly textured regions.

3.2.1 Approximative Surface Reconstruction

Range images as provided by TOF cameras actually represent a set of surface sample points. For this reason, it is quite reasonable to interpret range images as point sets and apply well-established upsampling techniques for point-based geometry.

The basic concept of most re-sampling techniques for point clouds is an approximative surface reconstruction that recovers a smooth surfaces from (possibly noisy) point cloud data. The resultant surface representation is often re-sampled to provide a new set of sample points according to given requirements. In praxis, individual algorithms vary with respect to global or local approximation as well as implicit or explicit representations. A very popular approach in this context, however, are moving least squares (MLS) surfaces, which have been already mentioned in the context of range data denoising (see Sec. 3.1).

MLS surfaces have been originally proposed by Levin [Lev03, Lev98] and later adopted by Alexa et al. [ABC*03, ABCO*01] for object reconstruction. Since then, they have been widely used for modeling and rendering of point-based geometry. The original definition is based on an projection operator $\mathcal{P}$ (see below), that projects an arbitrary point $\mathbf{x}$ to a locally fitted polynomial. Given $\mathcal{P}$, the entire surface S for a given point cloud is implicitly given by the stationary points of the projection operator, i.e. $S = \{\mathbf{x} \,|\, \mathcal{P}(\mathbf{x}) = \mathbf{x}\}$.

MLS Surface Definition Regarding the original definition by Levin [Lev03], the projection operator $\mathcal{P}(\mathbf{x})$ for a set of points $\mathbf{p}_i \in \mathbb{R}^3, i = 1 \dots N$, is defined in two steps:

1. Estimate a local *reference plane* $H(\mathbf{x})$ close to $\mathbf{x}$ that minimizes the weighted square distance of all points to the approximated plane. Here, the weights are given with respect to the projection of $\mathbf{x}$ onto $H(\mathbf{x})$, i.e. $\mathbf{q} = \mathbf{x} + t\vec{\mathrm{n}}$. The reference plane is found by a moving least square (MLS) optimization of

$$\min_{\vec{\mathrm{n}},t} \sum_{i=0}^{N} \langle \mathbf{p}_i - \mathbf{q}, \vec{\mathrm{n}} \rangle^2 \, w(\|\mathbf{p}_i - \mathbf{q}\|) \tag{3.5}$$

   where $w : \mathbb{R}_+ \to \mathbb{R}_+$ is a smooth and monotone decreasing function, i.e. $w(d) = \exp\left(-d^2/h^2\right)$.

2. Given the local reference plane $H(\mathbf{x})$, and thus an orthonormal coordinate system with origin in $\mathbf{q}$, a local bivariate polynomial is approximated. Let $q_i = (x_i, y_i, z_i)$ be the coordinates of $\mathbf{p}_i$ in the local coordinate system. The coefficients of a polynomial approximation $g(x, y)$ are found by a weighted least square (WLS) optimization of

$$\min_{g} \sum_{i=0}^{N} \left(g(x_i, y_i) - z_i\right)^2 w(\|\mathbf{p}_i - \mathbf{q}\|). \tag{3.6}$$

Knowing $H(\mathbf{x})$ and $g(x, y)$, $\mathcal{P}(\mathbf{x})$ is finally defined as $\mathcal{P}(\mathbf{x}) = \mathbf{q} + g(0, 0)\vec{\mathrm{n}}$.

Computing the projection $\mathcal{P}(\mathbf{x})$ for a given point $\mathbf{x}$ generally implies non-linear optimization constraints, which will most likely have more than one local minimum. For this reason usually an iterative (still non-linear) scheme is applied, that descends toward the next local minimum by optimizing either $\vec{\mathrm{n}}$ or t in turns. Apart from that, an initial point $\mathbf{x}$ has to be a priori close enough to the surface, as the domain of the projection operator $\mathcal{P}$ is generally limited by the weighting term w.

An alternative definition of MLS surfaces has been given by Amenta et al. [AK04a], which leads to the generalization of MLS surfaces to extremal surfaces and proves that MLS surfaces are generally a subset of a manifold. The alternative extremal definition, furthermore, gives a more detailed understanding of the general domain of point set surfaces. It yields important insights into crucial shortcomings of the original definition [AK04b], including the projection step as well as the normal estimation by eigenvalue analysis. Here, both steps are shown to produce false results for points that are located too far away from the underyling surface.

The first prove for the approximation quality of an implicit MLS definition was given by Kolluri in the year 2005 [Kol05] by regarding uniform sampled surfaces. Kolluri was able to prove that for the adherence of a particular sample density,

the implicit MLS surface is a good approximation of the original surface and reconstructs the surface geometrically and topologically correct. Dey and Sun later extend this proof to non-uniform sampled data and introduced an adaptive MLS approach for a varying feature size [DS05].

Both approaches, however, imply either a priori known surface normals or incorporate an extensive normal and feature size approximation. Furthermore as both approaches are based on an implicit surface definition (root finding), they are generally more complicated to visualize than explicitly defined point set surfaces.

Ray Casting of MLS Surfaces Once the MLS surface is defined, new sample points can be added by projecting additional points on the point set surface. However, instead of inserting random sample points, the upsampling of TOF range images can be basically regarded as ray casting the MLS surface, i.e. explicitly adding sample points where the viewing ray through a pixel of the output buffer intersects the surface. Here, basically any arbitrary viewing position can be chosen, which beside image refinement also allows a resolution independent and (in principle) interactive visualization of the acquired range image.

The first ray casting approach for MLS surfaces has been published by Adamson and Alexa [AA03a, AA03b], which has been later ported to the GPU by Tejada et al. [TGN*06].

The main idea behind the ray casting of MLS surfaces is to converge towards the surface by iteratively projecting points $\mathbf{x}_i$ from the ray onto the surface S (see Fig. 3.3). In each step the current local surface representation g is approximated with respect to $\mathbf{x}_i$, and its intersection with the according ray $\mathbf{x}_{i+1}$ is determined. The whole process is repeated until the distance between the current intersection estimate $\mathbf{x}_i$ and its projection $\mathcal{P}(\mathbf{x}_i)$ is below a given threshold. Regarding the input domain of the projection operator $\mathcal{P}$ [AK04b], the initial intersection point $\mathbf{x}_0$ is determined by using an enclosing union of spheres with radius h. By doing so, $\mathbf{x}_0$ is assumed to be sufficiently close enough to the surface S.

As the polynomial g is only reliable within a sufficient proximity to $\mathcal{P}(\mathbf{x}_i)$, it is necessary to restrict the validity intersection points to a specific region of trust with radius r_T around $\mathcal{P}(\mathbf{x}_i)$. Here, Adamson and Alexa argue that the deviation between surface and approximation is naturally bounded by a sphere with radius h and thus $r_T \leq h$ is an adequate choice for a convergence criteria.

Otherwise, if $|x_{i+1} - \mathcal{P}(x_i)| > r_T$, no valid intersection is found and the iteration is proceeded using the next nearest intersection with the enclosing sphere structure along the viewing ray.

For the GPU implementation, the whole iteration process is decomposed into several render passes [TGN*06]:

1. **Initialization** In the initialization step a polynomial is approximated for each point of the point set. Due to the implicit grid structure of range images, the initialization is performed by rendering a quad in size of the input image,

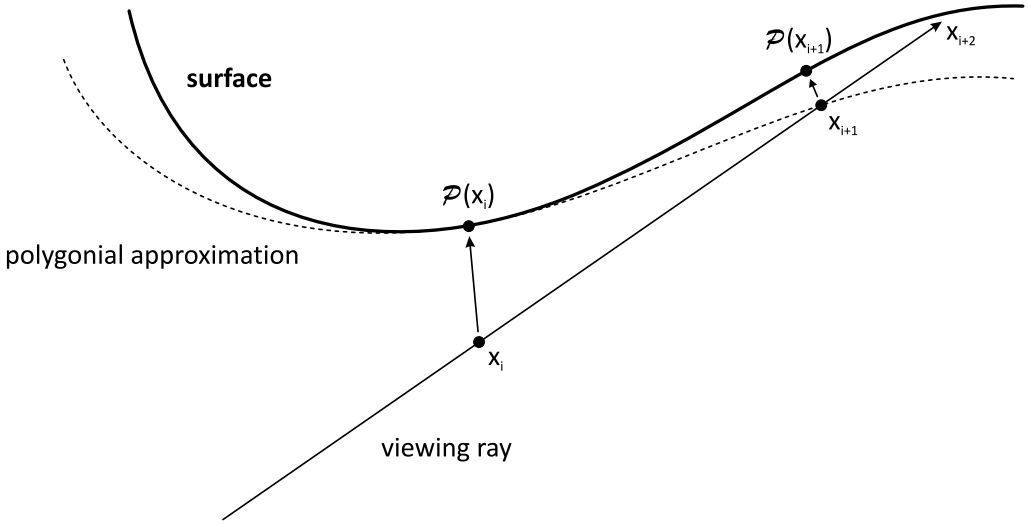


Figure 3.3: Iterative convergence of MLS ray casting by intersecting with the local polynomial approximation for $\mathbf{X}_i$ [AA03a].

such that each pixel corresponds to a point of the point set. For each rendered pixel $\mathbf{p}$, the coefficients of the polynomial $g(x, y) = ax^2 + by^2 + cxy + d$ are estimated in means of the projection operator explained above. In contrast to the general case, the initialization is strongly simplified, as explicit data structures for neighborhood relationships are not required.

The type of the polynomial is generally restricted by the number of available output (render) buffers, which are currently limited to eight buffers with at most four channels. However, in order to exploit the hardware support for matrices and vector, the number coefficients has been limited to four. By doing so, the intersection computation between viewing ray $\mathbf{r}(\alpha) = \alpha\vec{\mathbf{r}} + \mathbf{v}$ and locally fitted polynomial $g(x, y)$ is reduced to a quadratic functional with respect to the ray step parameter α, i.e.

$$\alpha_{1,2} = \frac{-f \pm \sqrt{f^2 - 4eg}}{2e} \tag{3.7}$$

with

$$\begin{aligned} e &= a \cdot r_x^2 + b \cdot r_y^2 + c \cdot r_x r_y \\ f &= 2(a \cdot r_x v_x + b \cdot r_y v_y) + c(r_x v_y + r_y v_x) - r_z \\ g &= a v_x^2 + b \cdot v_y^2 + c \cdot v_x v_y + d - v_z \end{aligned} \tag{3.8}$$

Here, $\mathbf{v}$ represents the viewer position with respect to the local coordinate system defined by H. For the case $e = 0$, (3.7) gets reduced to the linear solution $\alpha = -g/f$.

Assuming that $\mathbf{p}$ is already close to S, i.e. $t = 0$, step 1 of the projection operator degrades to a WLS normal estimation for the neighborhood $\Omega_{\mathbf{p}}$ (cmp. (3.5), p. 73) using eigenvalue analysis. In our situation, the neighborhood information is implicitly given by the range image structure. The normal estimation can be improved by building up the covariance matrix with respect to the average point of the neighbor points instead of the current pixel $\mathbf{p}$. Otherwise, if $\mathbf{p}$ is too far away from S, the eigenvalue analysis might fail (see Fig. 3.4).

2. **Find Intersection** For each point $\mathbf{x}_i$ of the point set a spherical point sprites of radius h is rendered to emulate the enclosing sphere structure. For each covered pixel $\mathbf{p}$, an intersection test of the according viewing ray with the local polynomial is performed. Due to the enabled z-buffer test, the closest intersection point $\mathbf{x}_p$ is finally selected.

3a. **Form Covariance Matrix** Again a spherical point sprite of radius h is rendered for each point $\mathbf{x}_i$. For each covered pixel, the entries of the covariance matrix $(\mathbf{x}_i - \mathbf{x}_p)(\mathbf{x}_i - \mathbf{x}_p)^T w(\|\mathbf{x}_i - \mathbf{x}_p\|)$ are accumulated using additive alpha blending and stored in two render buffers.

3b. **Normal Estimation** A full screen quad is rendered to perform a pixel-wise eigenvalue analysis like the *inverse power method* for the previous accumulated covariance matrix.

4a. **Form Linear System for Polynomial Fitting** Again a spherical point sprite is rendered for each point $\mathbf{x}_i$. This time, for each covered pixel its weighted contribution to the linear system, i.e. $\mathrm{A}^T\mathrm{A}$ and $\underline{z}_i\mathrm{A}$ with $\mathrm{A}^T = [\underline{x}_i^2 \quad \underline{y}_i^2 \quad \underline{x}_i\underline{y}_i \quad 1]$, is accumulated and stored. Here, $\underline{\mathbf{x}}_\mathbf{i}$ represents the coordinates of the point $\mathbf{x}_i$ with respect to the local coordinates system defined by $(\mathbf{x}_p, \vec{\mathrm{n}})$.

4b. **Estimate Polynomial Coefficients** Another render pass of a full screen quad is used to calculate the solution (coefficients) of the previous accumulated linear system.

5. **Projection** Each current intersection point $\mathbf{x}_p$ is projected onto the local approximated surface representation. If $|\mathcal{P}(\mathbf{x}_p) - \mathbf{x}_p| < \epsilon$, $\mathbf{x}_p$ is supposed to be the desired ray-surface intersection and the iteration process is terminated. Otherwise $\mathbf{x}_p$ is updated to the ray intersection with respect to the local approximation, and the iteration is continued in step 3a.

If step 3b or 4b fails, $\mathbf{x}_p$ is assumed to be false positive. In this case, step 5 is skipped and the next nearest intersection with the enclosing sphere structure is determined as new initial guess for the iteration process. Either way, according to Adamson and Alexa, a total number of 3 iterations is sufficient for most cases [AA03b].

Unlike general point sets, range images basically provide a varying sample density due to the perspective projection and different object distances. Thus, in

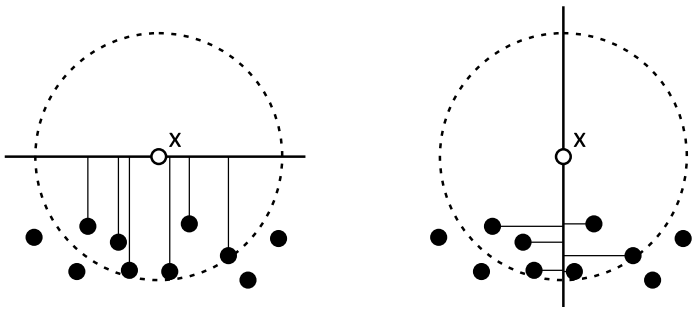


Figure 3.4: Planar surface estimation for the local neighborhood of $\mathbf{x}$. The sum of the distances of neighbor points to the plane parallel to the surface (left) are larger than those for the plane perpendicular to the surface (right), [AK04b]. A distance minimization leading to an eigenvalue decomposition therefore results in a wrong surface approximation for $\mathbf{x}$.

order to cope with sampling variations and avoid holes, the weighting parameter h is initially defined with respect to the image plane – e.g. covering a pixel neighborhood of $5 \times 5\,\text{px}$ – and accordingly scaled by the distance ratio $|\mathbf{x}_p| / f$.

Results The described ray casting has been applied to refine and display range images of $160 \times 120\,\text{px}$ resolution. For medium screen resolutions of $774 \times 486\,\text{px}$, we have been able to archive satisfying results (see Fig. 3.5) by a frame rate of $5 - 7\,\text{fps}$ on a Nvidia GeForce 8800 GTX. In all cases, the maximum number of intersection calculations has been restricted to 5 iterations, which turns out to be sufficient for most of our test scenes. The projection threshold has been varied between $0.1 - 1\,\text{cm}$

However, in all experiments, the small number of sample points as well as the high noise level of range images turned out to have notable negative impact onto to the normal approximation. Especially for small neighborhoods, the eigenvalue analysis is most likely to fail. Larger neighborhoods, on the other hand, result in undesired smoothing of object features, as the number of sample points compared to the feature size of acquired objects is usually quite small for TOF-based range images. Consequently, a pre-smoothing of the range image as described in Sec. 3.1 is absolutely vital.

Another problem arises from the typically non-closed surface structure of acquired range images. In contrast to common input data for a MLS surface reconstruction, featuring a single object, range images generally show multiple objects from a single viewing position and thus exhibit certain distance discontinuities. The locally fitted (unclipped) polynomials, however, are not capable to

reflect these discontinuities in an according way and therefore lead to undesired growing of object contours due to wrongly accepted ray-polynomial intersections inside a region of trust (see Fig. 3.6).

Furthermore, by definition, basic MLS-based techniques can reconstruct smooth surface only. For this reason, most approaches suffer the loss of sharp object features, especially edges. More sophisticated approaches for surface approximation exist, but are primary based on point cloud segmentation [JWS08, SDK09], robust statistics [OGG09, FCOS05] or complex implicit definitions, and therefore are rather inappropriate for real-time purposes.

3.2.2 Edge-Preservative Image Upscaling

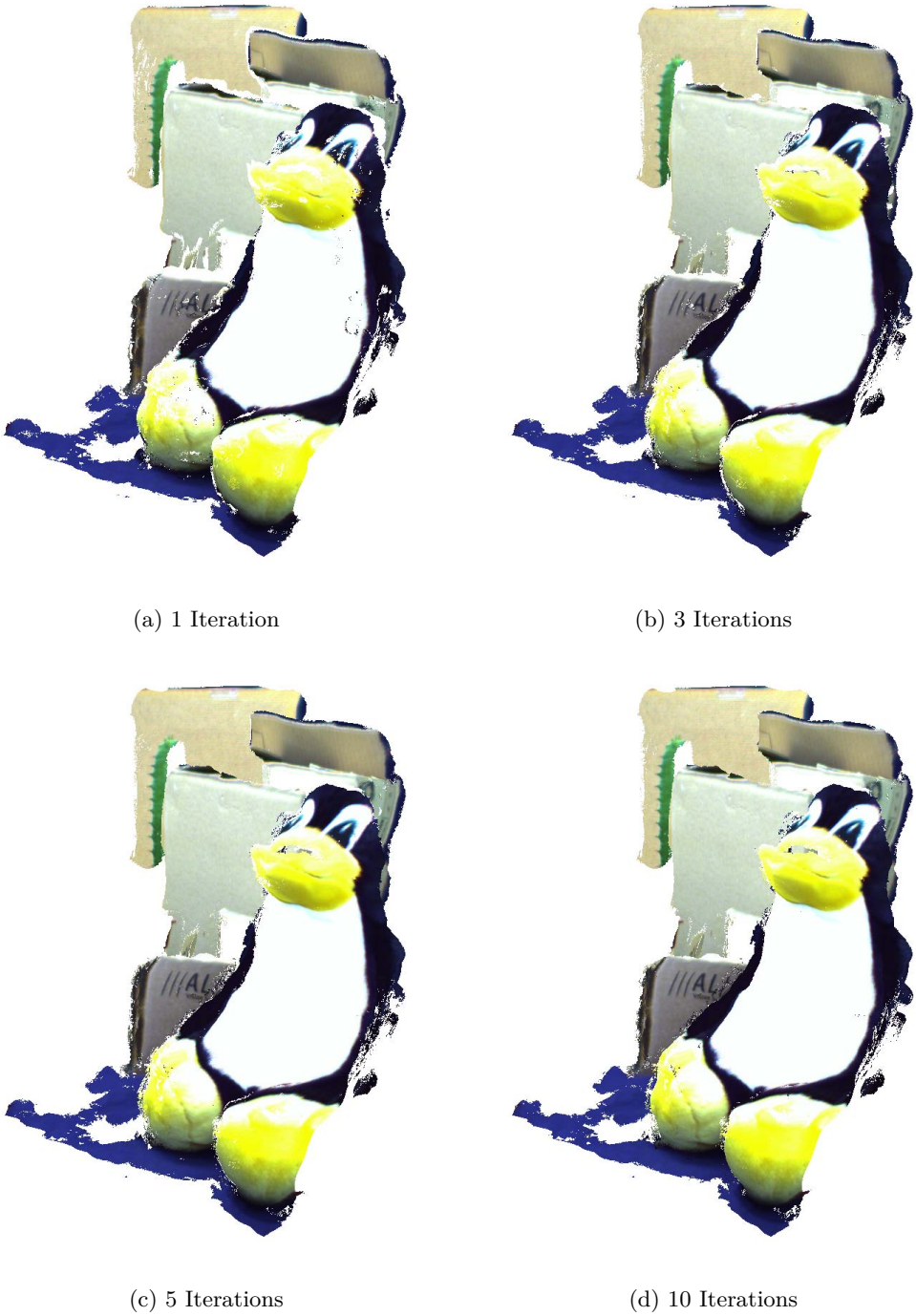
Due to the explicit surface fitting, MLS-based reconstruction as presented in Sec. 3.2.1 generally implies noise reduction along the surface normal. For range images, however, noise occurs along the viewing ray, which is not necessarily orthogonal to a pixel's according surface patch. This section therefore describes a mono-modal refinement of distance information in image space. The presented approach is based on an advanced image upsampling technique published by Kraus et al. [KES07], which has been extended to range images in order to account for invalid pixels as well as different occurrences of edges, i.e. object contours.

In the simplest case, image upscaling is done by bilinear interpolation, which equals a simple quadrilateral mesh reconstruction of the provided point cloud. Linear interpolation, however, is only a rough approximation and results in typical diamond artifacts. For more accurate refinement, usually higher-order filters like biquadratic or bicubic bspline interpolation are used, that guaranty a more precise data interpolation at the expense of an increased amount of input pixels (cmp. Fig. 3.7, p. 82).

High order schemes, however, are not directly supported by the current graphics hardware and therefore must be implemented by means of special fragment programs. Commonly, such programs imply a large number of texture fetches, which significantly reduces the frame rate. For this purpose, Sigg and Hadwiger [SH05] have introduced a sampling scheme that exploits hardware supported bilateral interpolation to reduce texture fetches and enables fast bicubic bspline filtering for real-time applications.

An alternative approach has been published by Strengert et al. [SKE06], whose biquadratic b-spline interpolation is based on iterative pyramid-like upscaling, but requires less texture fetches than the technique presented by Sigg and Hadwiger. Later Kraus et al. [KES07] extended the pyramid upscaling scheme by an technique to preserve edges in the upscaled result (see below).

In the following, we adopt the upscaling technique presented by Kraus et al. for range images. We therefore adjust the sampling scheme to account for invalid data and perform an (optional) extrapolation of distance information if possible. Thus, during the distance refinement, two main processing steps are performed:



(a) 1 Iteration

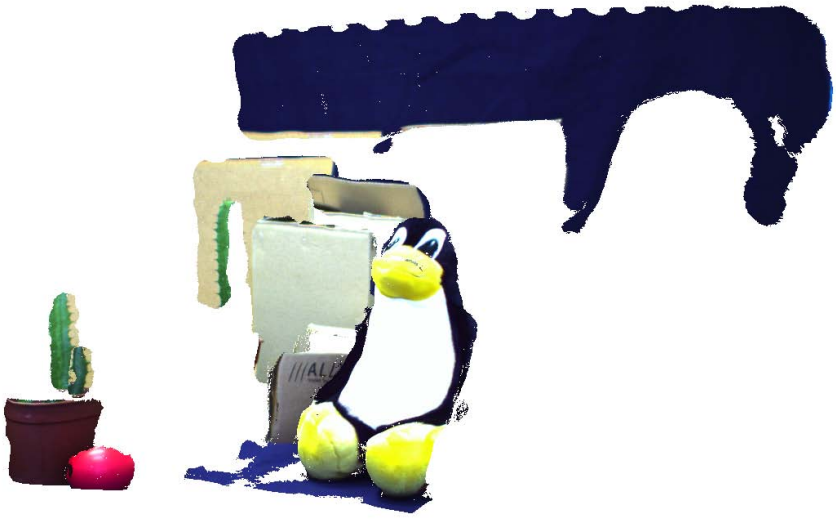
(b) 3 Iterations

(c) 5 Iterations

(d) 10 Iterations

Figure 3.5: MLS ray casting results for a different number of iterations. Note that viewing rays close to object contours need about five iterations to converge. After that the surface approximation does not change significantly.

(a)



(b)

Figure 3.6: Comparison between (a) piecewise linear interpolation and (b) MLS upsampling. Note the hole filling effect with regard to the soft toy's eyes and the undesired growing of object contours (e.g. regarding the cactus).

1. An extrapolation of valid pixels to fill in missing data for the biquadratic interpolation scheme and thus avoid shrinking of the valid pixel areas.
2. The upscaling of the valid regions incorporating edge preserving as well as special treatment of still incomplete grid cells.

Generally, interpolation schemes of higher order are possible, but have been discarded due to the size of their filter masks and the large number of special cases with respect to invalid pixel configurations.

Biquadratic Interpolation In the context of pyramid-based upscaling, two general possibilities for new samples positions exist (see Fig. 3.8):

– *Primal Scheme:* Add new samples between old sample locations while preserving existing information.

– *Dual Scheme:* Place new samples symmetrically between old samples locations and discard existing information.

While the primal scheme is the more traditional scheme, image interpolation according to the dual scheme, as depicted in Fig. 3.8, results in a C^1-continuous biquadratic bspline filtering [KES07]. By making use of the bilinear interpolation capabilities of graphics hardware, the pyramid-based biquadratic filtering can be implemented easily yielding linear time complexity.

Preserving Discontinuities Instead of applying the dual sampling scheme directly, Kraus et al. [KES07] suggest a gradient-based displacement of the sampling position as follows.

Assuming an ideal edge Θ at x_0 that has been blurred by applying a convolution with the normal distribution $\mathcal{N}$ of standard deviation σ, i.e.

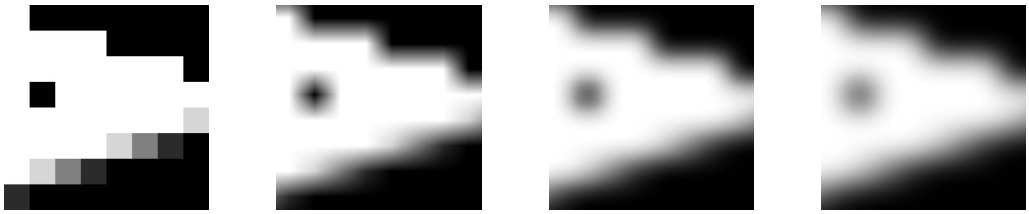
$$\begin{aligned} f(x) &= (y_{\min} + (y_{\max} - y_{\min}) \cdot \Theta(x - x_0)) \otimes \mathcal{N}(x, \sigma^2) \\ &= \frac{y_{\min} + y_{\max}}{2} + \frac{y_{\max} - y_{\min}}{2} \operatorname{erf}\left(\frac{x - x_0}{\sqrt{2}\sigma}\right), \end{aligned} \tag{3.9}$$

the ideal edge transition x_0 is related to the root of the scale invariant expression

$$d(x) \overset{def}{=} \frac{-\sigma^2 f''(x)}{f'(x)} = x - x_0. \tag{3.10}$$

Intending the blurred edge for the doubled resolution to be twice as sharp, i.e. using $\sigma/2$ for the convolution in (3.9), the sharpened edge signal is alternatively

erf() stands for the Gaussian error function with $\mathcal{N}(x, \sigma^2) = 0.5\left(1 + \operatorname{erf}\left(x/\sqrt{2}\right)\right)$



(a) (b) (c) (d)

Figure 3.7: Overview of interpolation schemes: (a) piecewise constant, (b) piecewise bilinear interpolation (C^0 continuous), (c) biquadratic B-splines (C^1 continuous), (d) bicubic B-splines (C^2 continuous) [SKE06].

expressed by

$$f(2(x - x_0) + x_0), \tag{3.11}$$

which can be rewritten as

$$f(2(x - x_0) + x_0) = f(x + (x - x_0)) = f(x + d(x)). \tag{3.12}$$

Thus, the sharpening of an ideal edge can be achieved be resampling the original edge signal $f(x)$ at the sample position displaced by $d(x)$.

In the two-dimensional case, $f''(x)$ is commonly approximated by the Laplacian $\Delta f(\mathbf{x})$, whereas $f'(x)$ is given by the gradient $\nabla f(\mathbf{x})$. Choosing the normalized gradient for the direction of the second derivative, the coordinates of each sample location are displaced by an offset vector

$$\vec{\mathrm{d}}(\mathbf{x}) = \frac{-\sigma^2 \cdot \Delta f(\mathbf{x})}{|\nabla f(\mathbf{x})|^2} \nabla f(\mathbf{x}). \tag{3.13}$$

Accordingly, the parameter σ controls the maximum scale of edges that are sharpened, i.e. for $\sigma = 0$ no edges are sharpened and the method is equivalent to the original biquadratic interpolation.

As the displacement $\vec{\mathrm{d}}(\mathbf{x})$ depends on the gradient $\nabla f(\mathbf{x})$ and the Laplacian $\Delta f(\mathbf{x})$, the input range image should be denoised before these values are estimated. However, it is important that the final image interpolation at sample position $f(\mathbf{x} + \vec{\mathrm{d}}(\mathbf{x}))$ accesses the unfiltered range image to avoid unintended blurring of the magnified image. In the context of TOF range images, the gradient and Laplacian are approximated based on valid depth values only. Invalid range values were simply discarded from the respective filtering masks.

In order to avoid excessive sharpening due to numerical instabilities, $\vec{\mathrm{d}}(\mathbf{x})$ should be additionally clamped to an user-specified maximum t_d, i.e.

$$\hat{\mathrm{d}}(\mathbf{x}) = \min(t_d, |\vec{\mathrm{d}}|) \cdot \frac{\vec{\mathrm{d}}}{|\vec{\mathrm{d}}|} \tag{3.14}$$

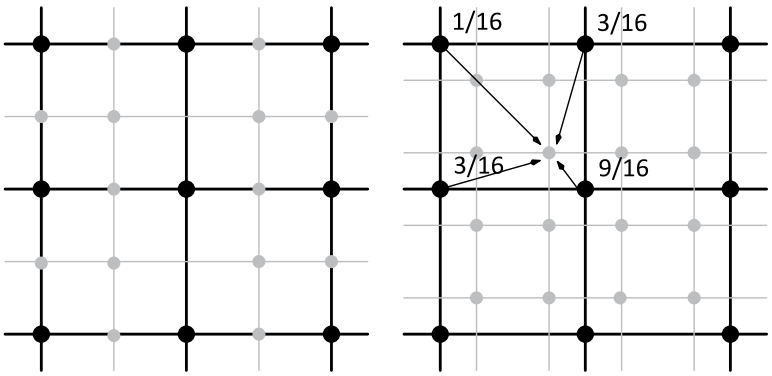


Figure 3.8: Left: Primal scheme reusing existing pixel; Right: Dual scheme based on the subdivision of biquadratic B-spline patches [SKE06].

Regarding the sampling scheme, Kraus et al. suggest a value of $t_d \approx 0.25$, which worked fine in our experiments, but generally does not account for flying pixels. Here, t_d must be large enough to sample the range data one or more pixels away.

Instead of considering a single sample displacement $t_d > 0.25$, which might lead to a susceptibility to noise, the original edge-directed approach has been further extended by applying the offset correction iteratively. Therefore, the sampling position is iteratively displaced by $\hat{\mathrm{d}}(\mathbf{x})$ until the displacement is smaller than a given threshold t_d^m or the maximum number of iterations has been reached. By doing so, the sample location is moved away from the edge center until a position on the adjacent region has been reached. In contrast to the original approach, which considers only a fixed smoothing magnitude σ for all edges, the iterative approach is able to cope with differently smoothed edges (including flying pixels), while an over-sharpening of smaller edges is avoided.

In order to reduce noise and edge sharpening in rather homogeneous regions, an additional, second threshold t_g is introduced, that allows the location correction to be omitted if the length of the gradient $\nabla f(\mathbf{x})$ is smaller than t_g.

Handling of Invalid Pixels In contrast to standard images, range images usually comprise inaccurate distance information for outliers and pixels covering depth inhomogeneity (flying pixels) as well as areas with insufficient incident active light. During interpolation, it is very important to avoid the propagation of data from both inaccurate as well as invalid pixels. In consequence, the dual interpolation scheme should be applied to cells with four valid pixels only. Unfortunately, skipping the interpolation of incomplete cells finally results in a shrinking of valid regions.

For this reason, an extrapolation of the boundary depth values is performed in order to expand the valid pixel region artificially. Note, that the extrapolation is applied without changing the original validity of a pixel. The detailed extrapolation

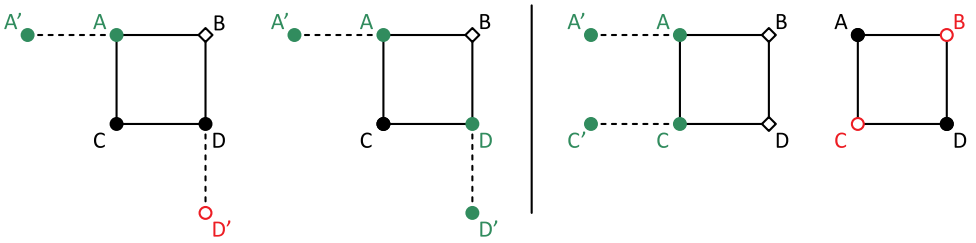


Figure 3.9: Extrapolation of invalid pixel (• valid, ∘ invalid and ⋄ extrapolated pixel). Left: Extrapolation in case of one invalid sample: $B = 2A - A'$ or $B = \frac{1}{2}(2A - A' + 2D - D')$; Right: Extrapolation in case of two invalid samples: $B = 2A - A'$, $D = 2C - C'$. In the second case, no extrapolation is performed.

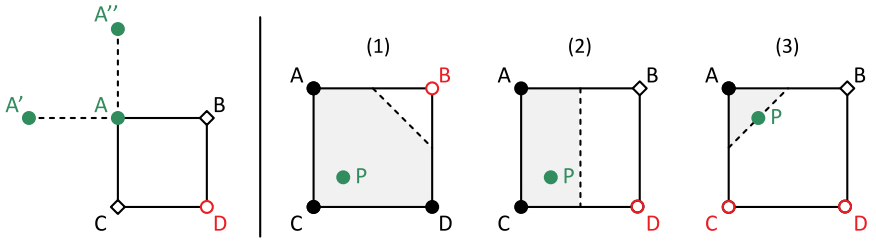


Figure 3.10: Left: Extrapolation in case of three invalid samples: $B = 2A - A'$, $C = 2A - A''$; Right: Interpolation using valid (•) and extrapolated (⋄) samples only: (1) $P = \frac{1}{4}A + \frac{1}{2}C + \frac{1}{4}D$, (2) $P = \frac{1}{4}B + \frac{3}{4}C$, (3) $P = \frac{3}{4}A + \frac{1}{4}B$. Dashed lines indicate object boundaries.

schemes for one and two invalid pixels are depicted in Fig. 3.9, whereas the three pixel case is shown in Fig. 3.10 left.

After extrapolation, the actual interpolation is additionally restricted to meaningful sample locations with respect to the original object boundary. By doing so, smooth object boundaries can be preserved while staircase artifacts are avoided.

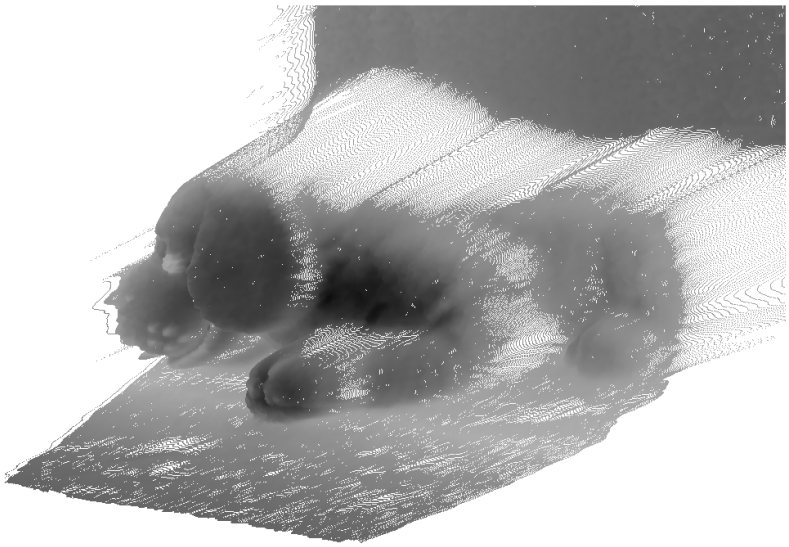
In cases where the extrapolation has failed, simpler interpolation schemes like linear interpolation or barycentric coordinates are applied in order to obtain smooth boundary contours (see Fig. 3.10 right).

Results The distance refinement has been tested with a range images of 160 × 120 px resolution. Choosing a scaling factor of 16 and a maximal number of of 250 iterations, we have been able to achieve a frame rate of 60 fps on a Nvidia GeForce 8800 GTX. The test results are shown in Fig. 3.11 - 3.13:

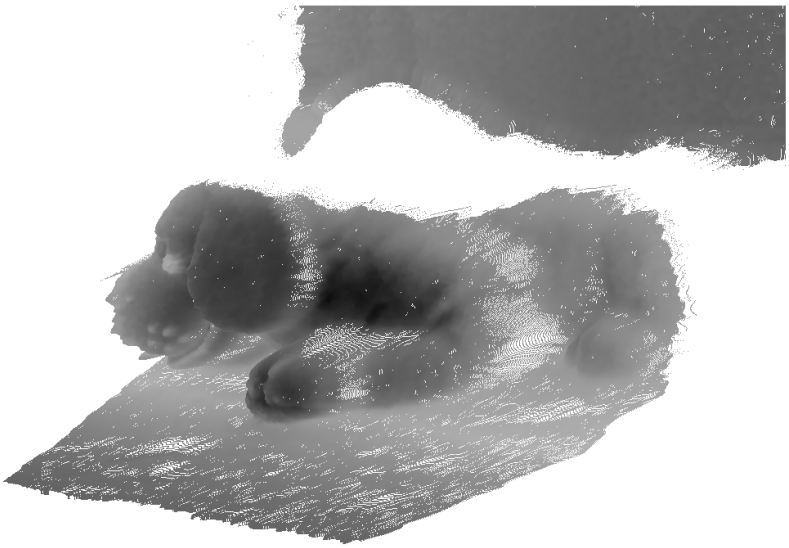
Fig. 3.11 demonstrates the difference between general biquadratic upscaling and our iterative, edge-directed approach. While biquadratic upsampling leads to interpolated pixels along object contours, iterative edge-directed upsampling provides a well defined separation of foreground and background.

Fig. 3.12 shows the impact of both, extrapolation of invalid pixels and reduced interpolation of incomplete grid cells, on the upscaling results. As can be seen in Fig. 3.12 a, biquadratic interpolation of complete grid cells only clearly results in staircase and contour shrinking effects. While extrapolation already appreciable improves the upscaling results (Fig. 3.12 b), additional reduced interpolation of incomplete grid cells adds further details (see Fig. 3.12 c).

Fig. 3.13 demonstrates the difference between unrestricted and boundary strict interpolation. While unrestricted interpolated leads to lacerated contours, strict interpolation guarantees clean outlines.

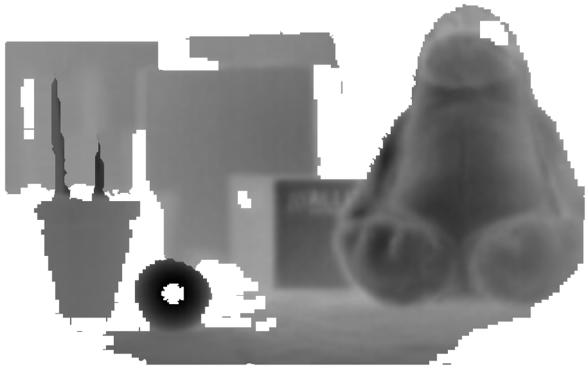


(a) Biquadratic interpolation.

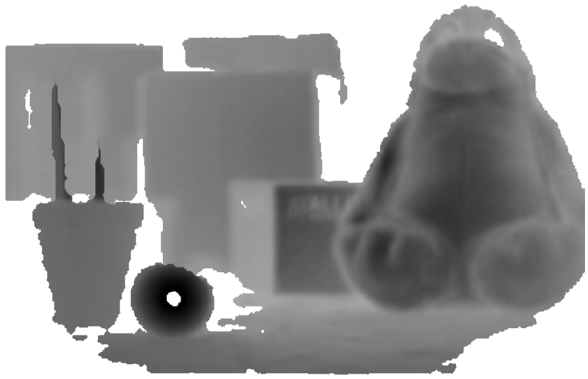


(b) Iterative, edge-directed interpolation.

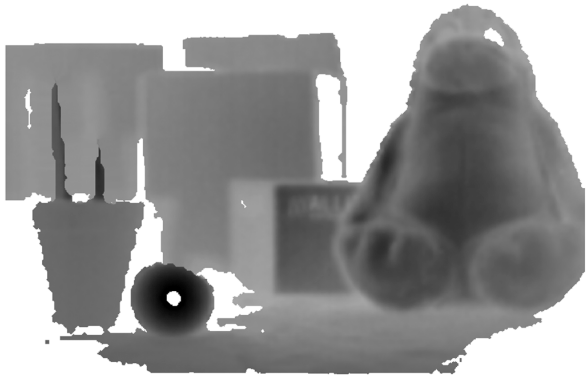
Figure 3.11: Comparison of standard biquadratic upsampling and the iterative, edge-directed approach.



(a) Original, edge-directed upsampling approach without extrapolation.



(b) Extended, edge-directed upsampling with extrapolation only.



(c) Extended, edge-directed upsampling with additional bilinear interpolation.

Figure 3.12: Comparison of the extended, edge-direted upsampling results with respect to the original, edge-directed approach.

(a) Original, edge-directed upsampling approach.

(b) Extended, edge-directed upsampling using non-strict interpolation.

(c) Extended, edge-directed upsampling using strict interpolation.

Figure 3.13: Comparison of interpolation results with respect to different boundary handling.

3.3 Image Sensor Fusion

In general, a given sensor provides only the specific type of information it has been designed for, like for example range data, color information or near-infrared samples. Depending on the complexity of a given task, the information of a single sensor might not be sufficient to archive satisfying results.

For this reason, a wide range of vision systems usually consist of a variety of distinct sensors to benefit from combined, multi-modal data. Range sensing systems, for example, often incorporate additional high resolution image sensors to provide supplemental (color) information for general processing tasks like range image segmentation and refinement (see Sec. 3.2). Even in the simplest case, TOF cameras already benefit from the additional color information, as it can be used to enhance the visual quality by simply replacing the NIR intensity image with corresponding color information.

Nevertheless, due to the varying viewing perspective between two or more image sensors and the involved non-affine transformations, input data from different devices cannot be simply overlaid to get multi-modal information. Instead, sensor fusion is commonly achieved by hardware solutions, combining two or more sensors in a monocular setup, or software solutions, implementing sophisticated mapping algorithms.

A first hardware solution for combing a color and a TOF sensor has been published by Lottner et al. [LHLW07]. Here, a monocular camera consisting of two sensors is described, that uses a beam splitter to redirect the particular wave bands to the according image respectively TOF sensor. By doing so, both image sensors share the same perspective, which in this case allows a simple image overlay yielding combined RGBZ data.

However, the suggested hardware solution generally limits the process of sensor fusion as it disallows the incorporation of further sensor data without extensive hardware modification. The complexity of hardware solutions thereby increases, the more image sensors are supposed to be integrated. Especially the involved beam splitter limits the number of maximal sensors as it reduces the power of the incoming light signal.

Software-based data fusion, in contrast, usually allows a more flexible combination of image sensors (and sensing techniques) as each device a priori operates independently and can be added or removed as required. In contrast to hardware solutions, however, software approaches are commonly based on advanced mapping algorithms, which sometimes can be quite complex.

In the following, we discuss a simple software-based mapping between a given TOF and color image sensor and its extension to projective texture mapping. A similar approach has been presented by Reulke [Reu06], where the data fusion is done by an algorithm called *orthophoto generation*. Here, the 2D image is distorted in order to eliminate the perspectivity of the image by taking the 3D information into account. The result is an orthographic image, where the optical

rays are parallel. After the image rectification, the data fusion is straight forward applying a parallel mapping of the color information onto the appropriate distance data.

Simple Color Fusion The simplest approach for software-based sensor fusion that involves depth information, is commonly implemented by re-projecting each pixel of the range image onto the image plane of the additional image sensor, resulting in a single color value per pixel (see Fig. 3.16 a and 3.17 a, p. 94f).

In the first step, the observed scene is initially reconstructed through a back-projection of each pixel of the range image $\mathbf{p}$ into the world coordinate frame of the according image sensor (see Sec. 1.3.4). The resultant point cloud of 3D sample points $\mathbf{X}_i$ is then transformed into the coordinate system of the color camera and finally projected onto the color sensor frame to obtain the desired image coordinates $\mathbf{p}_i^{\mathrm{B}}$ by

$$\mathbf{p}_i^{\mathrm{B}} = \mathrm{K_B} \Pi_1 \mathrm{M_B} \mathbf{X}_i \tag{3.15}$$

where $\mathrm{M_B}$ represents an affine viewing transformation determined by the sensor registration step and $\mathrm{K_B}$ consists of the intrinsic parameters of the color camera.

Undersampling As a re-projected pixel of a range image most certainly covers a larger area than that of a color pixel, the simple selection of nearest (possibly linear interpolated) neighbor information based on (3.15) generally leads to visible undersampling artifacts.

A common approach to reduce undersampling has been invented in the year 1983 by Lance Williams and is called *mip mapping* [Wil83]. Here, a series of downscaled (averaged) versions of the input image provides the most appropriate sampling texture, whose texels in screen space cover at least one pixel of the output buffer. Starting with the original input image at level 0, the mipmap texture on level l is downscaled by a factor of two to half the size of the mipmap texture on level $l-1$.

In the case of projective texturing, where the texture is not attached to the geometry directly, the appropriate mipmap level for an output pixel (u, v) is given by the scale factor $\lceil \log_2 \rho(\mathbf{X}(u,v)) \rceil$, where $\rho(\mathbf{X})$ states the pixel ratio between the output buffer and the texture, regarding the according surface point $\mathbf{X}$. Due to possibly rectangular pixels, i.e. $s_x \neq s_y$, the pixel ratio $\rho(\mathbf{X})$ is actually given by

$$\rho(\mathbf{X}) = \max\{\rho_x(\mathbf{X}), \rho_y(\mathbf{X})\} \tag{3.16}$$

with

$$\rho_i(\mathbf{X}) = \frac{S_i^{\mathrm{v}}(\mathbf{X})}{S_i^{\mathrm{B}}(\mathbf{X})} = \frac{Z_{\mathrm{v}}(\mathbf{X})}{f_i^{\mathrm{v}}} \cdot \frac{f_i^{\mathrm{B}}}{Z_{\mathrm{B}}(\mathbf{X})} = \frac{Z_{\mathrm{v}}(\mathbf{X})}{Z_{\mathrm{B}}(\mathbf{X})} \cdot \left(\frac{f_i^{\mathrm{B}}}{f_i^{\mathrm{v}}} \right)_{\mathrm{const}} \tag{3.17}$$

where f_i^{B}, f_i^{v} correspond to the particular focal length in pixel units and $Z_{\mathrm{B}}(\mathbf{X})$, $Z_{\mathrm{v}}(\mathbf{X})$ represent the euclidean distance of the surface point to the projection

center of the according camera. Through substitution, (3.16) can be rewritten as

$$\rho(\mathbf{X}) = \max\left\{\frac{f_x^{\mathrm{B}}}{f_x^{\mathrm{v}}}, \frac{f_y^{\mathrm{B}}}{f_y^{\mathrm{v}}}\right\} \cdot \frac{Z_{\mathrm{v}}(\mathbf{X})}{Z_{\mathrm{B}}(\mathbf{X})}. \tag{3.18}$$

For a fixed camera rig, also $Z_{\mathrm{v}}(\mathbf{X})/Z_{\mathrm{B}}(\mathbf{X})$ can be considered to be constant. In this case $\rho(\mathbf{X})$, and therefore the according mipmap level as well, can be precomputed in advance. Accordingly, only a single level of the mipmap pyramid has to be calculated, resulting in a simple texture downscaling.

By applying averaging techniques like mipmapping, fusion artifacts can be significantly reduced as notable in Fig. 3.16. Nevertheless, relevant details of the color image might still get lost due to involved averaging of pixel information. For some applications, however, this loss of information might be curcial for the results of further processing steps.

Projective Mapping In order to maintain the full, unaltered information of a secondary image sensor, it is quite insufficient to interpret a reconstructed pixel of the range image as a single point in space. Thus, beside refined distance information (see Sec. 3.2), it is rather necessary to store the information covered by a range pixel and represent each pixel by an accordingly scaled, surface-oriented quad. The orientation of the quad can be either determined explicitly by means of an approximated surface normal (cmp. Sec. 3.2.1) or implicitly with respect to the interpolated distance of the quad vertices. We recommend the second technique in order to avoid extensive normal approximation as well as holes in the reconstructed surface geometry.

Given the more sophisticated representation of the scene, the additional color information for each quad can be retrieved by linear interpolation of the reprojected color image coordinates stated in (3.15). However, due to the involved perspective projections between TOF camera and the additional image senors, simple linear interpolation leads to distorted mapping results (cmp. Fig. 3.14). Instead, the general solution is to apply projective texture mapping as proposed by Segal [SKvW*92]. By doing so, the distortion of the color image inside each quad can be totally avoided.

Given two points $\mathbf{p}_1^{\mathrm{A}} = (u_1, v_1, w_1)_{\mathrm{A}}$ and $\mathbf{p}_2^{\mathrm{A}} = (u_2, v_2, w_2)_{\mathrm{A}}$ with $\mathbf{p}_i^{\mathrm{A}} = \mathrm{K_v}\Pi_1\mathbf{X_i}$ in clip coordinates of the range camera (before normalization), the determination of any intermediate point on the image plane using linear interpolation

$$\mathbf{p}_{\mathrm{A}} = (1-t)\cdot\mathbf{p}_1^{\mathrm{A}} + t\cdot\mathbf{p}_2^{\mathrm{A}} \tag{3.19}$$

requires the proper computation of the associate location $\mathbf{p}_{\mathrm{B}}$ in the RGB image plane (cmp. Fig. 3.14). According to Segal [SKvW*92], the color image plane coordinates $\mathbf{p}_{\mathrm{B}}$ are given by

$$\mathbf{p}_{\mathrm{B}} = (1-t)\cdot\mathbf{p}_1^{\mathrm{B}}/w_1^{\mathrm{A}} + t\cdot\mathbf{p}_2^{\mathrm{B}}/w_2^{\mathrm{A}} \tag{3.20}$$

where $\mathbf{p}_i^{\mathrm{B}} = \mathrm{K_B}\Pi_1\mathrm{M_B}\mathbf{X_i} = (u_i, v_i, w_i)_{\mathrm{B}}$ represent the corresponding clip coordinates with respect to the color camera. Thus, perspective correct interpolated color values can be achieved by a simple linear interpolation of $u_{\mathrm{B}}/w_{\mathrm{A}}$, $v_{\mathrm{B}}/w_{\mathrm{A}}$ and $w_{\mathrm{B}}/w_{\mathrm{A}}$ and a subsequent normalization.

Final results for projective mapping are shown in Fig. 3.17 clearly stating the improvement compared to simple mapping approach.

Occlusion Detection Due to the different viewing positions of both cameras, an incorrect mapping of occluded surface regions, e.g. in concave object regions, may occur (see Fig. 3.15). In this case, the additional color sensor is unable to provide proper image information, as the interpolation of re-projected image coordinates can only return false color information from the occluding surface. For setups where the projection centers of both cameras are positioned close to each other, occlusions effects predominantly occur in the near distance range.

In order to prevent a false mapping, we adopt a render approach comparable to shadow maps [RSC87, SD02]. Here, the main idea is to store the closest geometry with respect to the color sensor in a distance buffer. For this purpose, the geometry representation (surface aligned quads) is first transformed into the coordinate system of the color camera, using the transformation matrix $\mathrm{M_B}$, and projected onto the image plane.

During the rasterization of the projected geometry, the distance of each rasterized fragment with respect to the color camera's origin is written in an off-screen buffer $\mathcal{B}$ that stores the minimal per-pixel distance. Using the functionality of modern graphics hardware, this can be easily achieved by storing the z-Buffer usually used for hidden surface removal.

During the color mapping step, the currently transformed (and possibly interpolated) distance information with respect to the color sensor $z(\mathbf{p}_{\mathrm{B}})$ is determined and compared to the corresponding z-Buffer entry $\mathcal{B}(u_{\mathrm{B}}, v_{\mathrm{B}})$. In those cases where the interpolated distance is farther away from the color sensor than the distance information stored in the frame buffer, i.e.

$$z(\mathbf{p}_{\mathrm{B}}) > \mathcal{B}(u_{\mathrm{B}}, v_{\mathrm{B}}) + \varepsilon \tag{3.21}$$

the surface point is hidden and the color assignment is omitted. Otherwise, the color assignment is performed using either the simple or the projective approach described in Sec. 3.3. The ε-offset is required to account for z-Buffer quantization errors due to numerical inaccuracies and noise in the range data. Fig. 3.18 shows the mapping with additional occlusion detection.

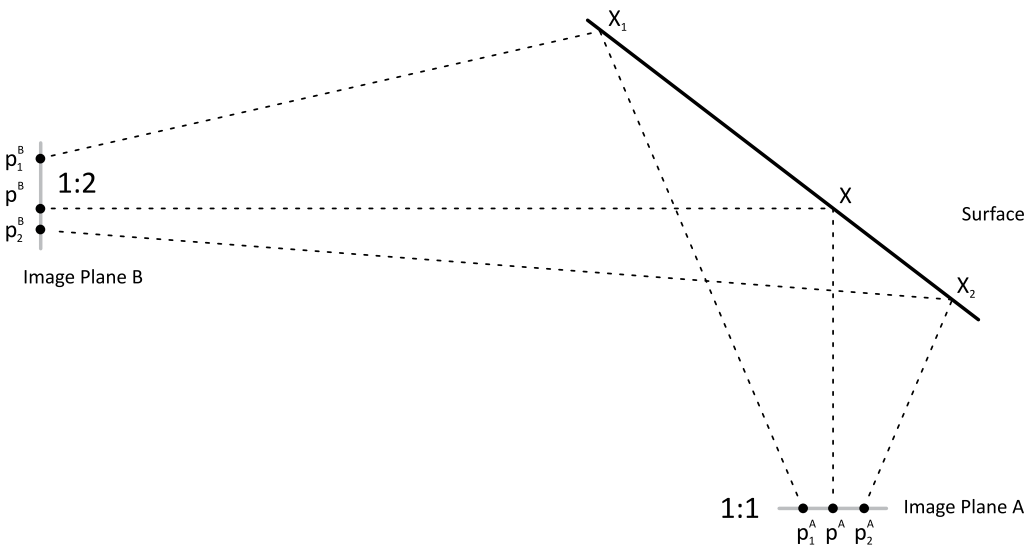


Figure 3.14: Texture distortion introduced by linear interpolation. The midway interpolated point $\mathbf{p}^A$ on the image plane A corresponds to a division ratio of 1:2 in the second image plane B.

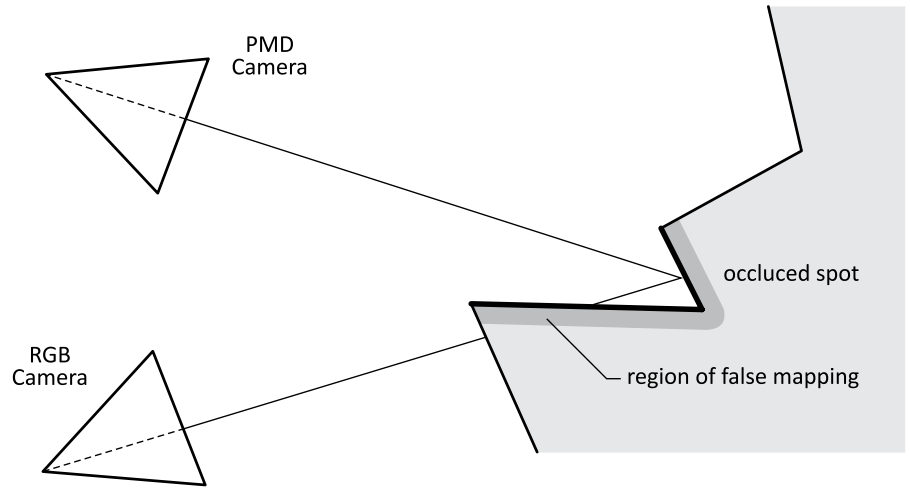
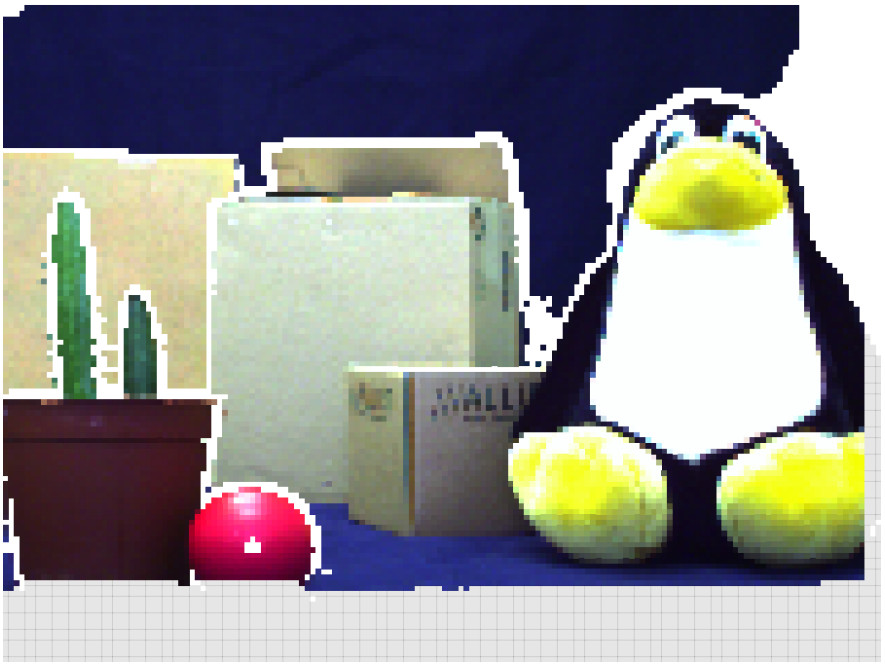
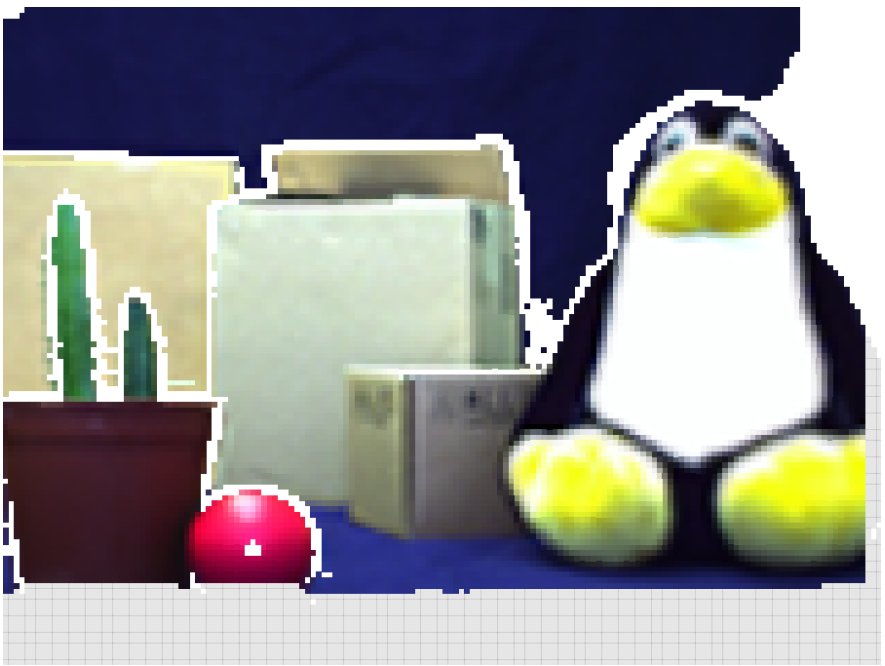


Figure 3.15: False color mapping in occluded regions with respect to the RGB sensor.

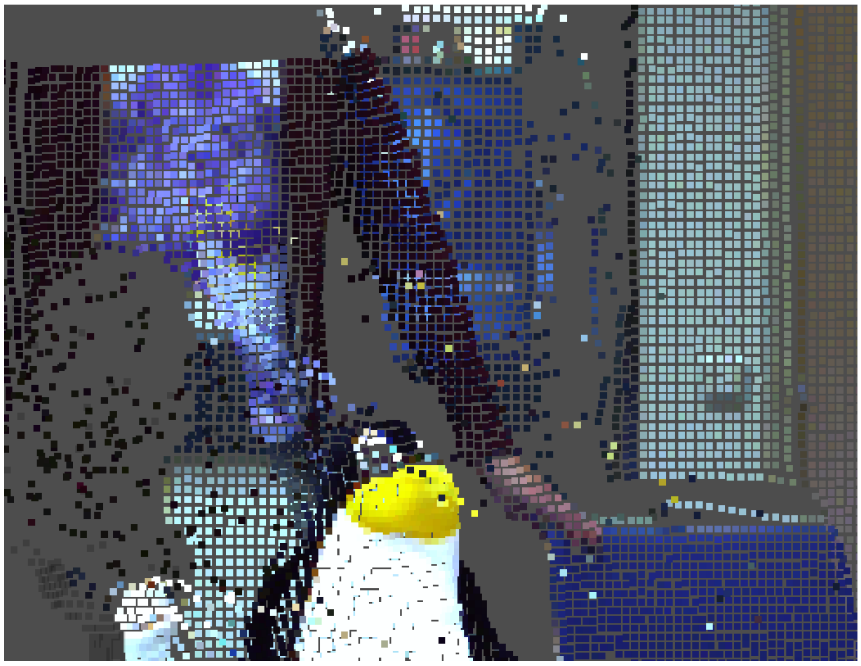


(a) Without Mip Mapping

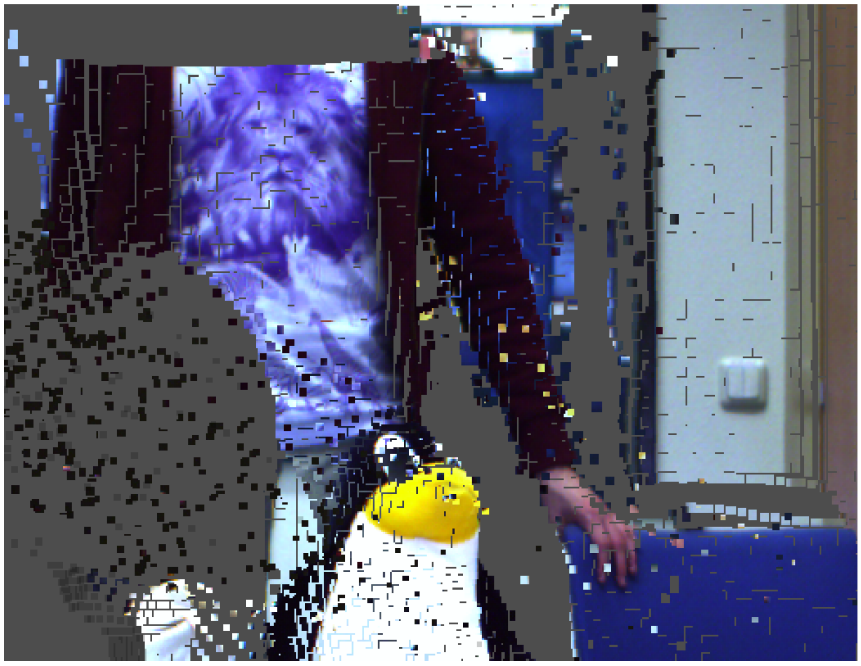


(b) With Mip Mapping

Figure 3.16: Applied mipmapping fusion in order to avoid undersampling. The noticeable blurring is introduced by linear interpolation and the mipmap-related scale factor of 2^3, as the real sensor scale ratio is 6.4.



(a)



(b)

Figure 3.17: Projective Mapping Results (b) versus simple mapping (a). Note the additional pattern details on the shirt.

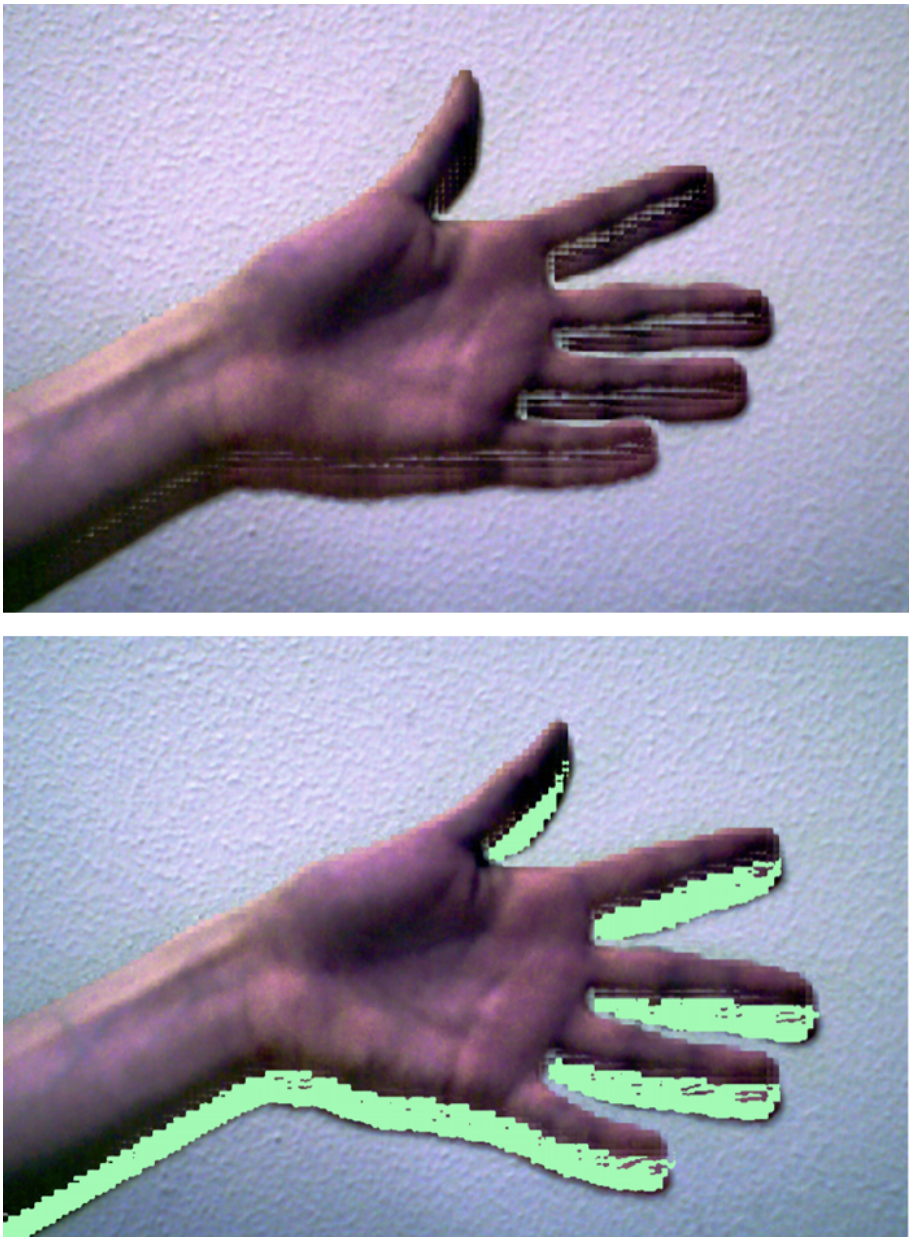


Figure 3.18: Occlusion detection (bottom) versus false mapping (top).

3.4 Generic Processing Framework

All algorithms discussed in this thesis have been implemented as part of a generic, pipeline-based processing framework, that can be easily extended by new modules. Due to its flexibility, it provides certain advantages compared to other solutions of fixed structure and functionality. Beside automatic memory management, it guarantees that only modified data is updated in order to avoid unnecessary recalculations.

The concept of a pipeline framework is not new and similar implementations can be found, for example, as part of:

- DirectShow (Microsoft)
- Quartz Composer (Apple)
- Insight Segmentation and Registration Toolkit (ITK, OpenSource)
- Visualization Toolkit (VTK, OpenSource)

The framework implementation described next features a graphical user interface (GUI), which is completely detached from the logical layer and thus can be easily adapted to particular requirements (e.g. replaced by a command line interface).

Similar to the underlying structure, the GUI provides the user with the three different types of pipeline modules: sources, sinks and filters. All modules can be interactively arranged and connected. Module parameters can be accessed via dynamically configured setup dialogs, which use the individual module descriptors to query information about parameter types as well as their according getter and setter methods. Additional control panels are used to access internal module states like e.g. the capture state of a camera source module.

Prototyping and Runtime Type Information The essential part for most generic systems is given by a framework that allows the user to create new objects by name or ID without modification and recompilation of the entire system. Therefore, the framework has to encapsulate knowledge about which concrete classes the system finally provides and how these classes are instantiated and combined. Here, generally two approaches exist [GHJV03]:

- the usage of specialized factory objects implementing a common interface (abstract factory)
- the creation of new instances by cloning a given prototype object (prototyping)

While the *abstract factory* pattern requires the implementation of additional classes, *prototyping* only depends on the implementation of a clone operation,

which means that factory and prototype are the same object. Beside reduced subclassing, prototyping also benefit from specifying new objects by

- simply varying the initial prototype parameters or
- combining objects to a new composite prototype, which is instantiated using deep copies.

Regardless of which pattern is used, either the factory object or the prototype has to be registered by a manager instance that allows the system to access the factory/prototype by an unique identifier. In the described processing framework, therefore, a special runtime type information (RTTI) class has been designed.

Each public class of the framework owns a static RTTI object that provides an unique ID as well as a user friendly name, e.g., the class name. While the ID can be efficiently used for internal purposes, but might change between program starts, the string constant is fix for each class and provides an identifier for purposes like user interaction or serialization. Every time a class is dynamically loaded, its static RTTI member is initialized, which registers itself at the factory passing the associated prototype object.

Furthermore, with respect to dynamic type checking and sub-classing, each RTTI object also provides information about the class hierarchy by storing a pointer to the RTTI object of its base class. This way, it can be verified if certain RTTI objects represent a subclass of a given base classed identifier.

Reference Counting and Smart-Pointer Regarding a generic program that allows the interactive combination of modules, dynamically allocated objects are often shared between several owners that do not necessarily know each other. In such situations, it is sometimes difficult to decide if an object is no longer in use and who is responsible for the deallocation. For this reason, modern object-oriented runtime environments like Java or $C_{\#}$ provide *garbage collection* where the framework periodically checks the heap for unreferenced objects – so-called tracing garbage collection.

In traditional programming languages like C or C++, where garbage collection is not a priori available, explicit reference counting (deterministic garbage collection) can be used to recreate a similar automatic memory management. Here, an attached counter is incremented or decremented every time an object is referenced or dereferenced. In contrast to garbage collection, where objects stay in memory until the next deallocation cycle is invoked, objects are reclaimed as soon as their reference counter drops to zero. In object oriented languages, incorrect reference counting can be avoided by encapsulating standard pointer references into *smart pointer* structures that provide the necessary functionality, i.e. a basic interface that increment a reference counter during creation and copy operations, and decrement the counter as soon as the deconstructor is called.

Unfortunately *reference cycles* can emerge if objects revere directly or indirectly to themselves. In such situations, the reference counters of all involved object will never be decremented to zero and the according objects remain on the heap. Here, weak pointer (which do not increase the reference counter, but keep track of the validity of the referred object) can be used to break up cyclic reference structures. If such differentiation between strong and week pointer can not be accomplished, the system can either prevent the user to create cycles completely or apply cycle-collection techniques that are similar to garbage collection.

Special care should be taken to avoid an additional performance overhead due to unnecessary assignments and deassignments of references regarding, e.g., frequent function calls or loops. Here, smart pointer references can be used to avoid the invocation of the copy-constructor during function calls.

Implicit casting of smart pointers can be realized either based on the original implicit casting functionality of the programming language, using a separate smart pointer hierarchy or by defining template based casting operators between the inherited class and the current base class.

Pipeline Concept The provided processing framework is based on a generic pipeline concept that allows the interactive combination of processing algorithms by means of filter modules. Accordingly, it provides basic programming interfaces for pipeline objects like data sources, sinks (for example render modules or file writer) and filters, and implements the underlying update mechanism.

Each pipeline module owns an input, respectively output, descriptor that consists of a list of RTTI string identifiers. Beside internal runtime type checks, these identifiers are primary used to provide the user with information about the input and output types (channels) of each module. Furthermore, it can be used by the framework to create output container for source and filter modules automatically. An additional attribute descriptor allows the system to provide the user with according getter/setter methods for available module parameters.

To improve the overall performance, the naive recalculation of intermediate results is avoided by applying an automatic update mechanism. This mechanism allows the systematic update of pipeline modules whose input data has changed. The according update cycle is designed as a two-step approach:

- In the first step, all sources are sequentially checked for updates. If a source provides new information, it passes a modification flag to those modules whose input channel is connect to one of its output channels. The modification flag is then recursively passed further through the pipeline until a sink or already modified module is reached.

- In the second step all sinks are checked for a positive modification flag and updated if necessary. In this case, a filter module passes the update call further through the pipeline until a source is reached. After all necessary

input data has been updated, the filter finally calculates the new output results. A status flag is used to prevent loops in the recursive update calls.

Basically running on the CPU, the pipeline framework has been extended to utilize the GPU for accelerated parallel computing if possible. In this case, special modules upload the required data to the memory of the graphics hardware, while GPU-based processing modules interchange intermediate results via texture buffers by passing texture ID objects that additionally provide information about texture size and type. During the execution, GPU processing results remain in the memory of the graphics hardware, unless they are used as input for CPU modules. In this case, the data has to be transfered back to a data container stored in the PC's main memory first.

Chapter 4

Discussion and Outlook

With the growing demand of user interactivity and autonomous systems, range information and therefore range sensing technology itself gains more and more in importance. During the last decades, different approaches and devices for the acquisition of range information have been proposed, that vary in accuracy, production costs and usability as well as real-time capability.

The list of sensing techniques recently has been extended by a new type of acquisition device. While former approaches are either computation intensive or have to sequentially scan a given scene, the newly developed time-of-flight cameras allow to acquire the range information in parallel for a whole scene in real-time. At the same time, the production costs are low due to their CMOS realization. Altogether, TOF cameras present a promising alternative for real-time vision application.

However, in comparison to classical techniques, current TOF cameras are still of low resolution as well as affected by several error sources. As the acquisition of distance information using TOF cameras is a rather new and unexplored technique, the subject of this thesis has been the investigation of the range sensing accuracy of current TOF cameras as well as the algorithmical improvement of distance information focusing on noise reduction and range data upsampling.

The first part has addressed the classical task of intrinsic parameter estimation and presented phenomenological calibration models for the two major error sources that deal with demodulation- as well as intensity-related errors. In this context, also an alternative demodulation approach has been investigated for error reduction, which assumes a rectangular modulation. The chapter is concluded by an outline of the compensation of TOF camera specific motion artifacts.

The second part has covered the general discussion of image noise reduction and investigated *Moving Least Square Surfaces* as one possible solution for range data upsampling. Furthermore, it contributed an edge preserving filter for range data upsampling in the image domain. The chapter concludes with the discussion of image sensor fusion in order to obtain multi-modal range information.

Discussion In the context of intrinsic calibration, we successfully applied the intrinsic parameter estimation to TOF camera models with a sensor resolution greater than $160 \times 120\,\text{px}$. The extrinsic pose estimation, however, is affected by sub-pixel inaccuracy of detected feature points as well as a parameter correlation due to a narrow opening angle. Classical pose estimation based on checkerboard

detection therefore turned out to be useful for rough pose only. Schiller et al. [SBK08], however, have shown that pose estimates can be significantly improved by the utilization of multi-camera rigs including at least one high-resolution standard imaging sensor. By doing so, the rigs absolute pose with respect to a single input image can be estimated using the additional high-resolution sensor, whereas the TOF camera's pose is stabilized by using multiple input images for an overall relative pose estimation.

Concerning the investigations of range sensing accuracy, experiments have revealed that current TOF cameras are systematically affected by a phase shift demodulation error as well as an intensity-related distance deviation. While the former produces a wiggling like distance error, the later causes lower reflective areas to significantly drift towards the camera. Compared to each other, the demodulation error, however, turns out to be the more dominant and causes distance deviations of several centimeters. For accurate range sensing, a proper wiggling calibration therefore is absolutely vital. In this context, the presented calibration model for wiggling adjustments is able to reduce wiggling errors to deviations smaller than one centimeter, while individual pixel impacts can be satisfactorily described by constant offsets. Regarding intensity-related deviations, two models have been presented. While the coupled approach is a simple extension to the wiggling model that leads to a bivariate approximation, further investigation showed that the effects can be decoupled in order to reduce the number of necessary reference data. However, the integration time dependent error remains due to the change in incident light intensity. For accurate range sensing, therefore, also a separate calibration of the integration time dependent offset is required.

Beside the two phenomenological calibration models, also an alternative phase shift demodulation has been proposed, that assumes a rectangular signal modulation. Experiments have shown that the wiggling error remains, but exhibits contrary wiggling behavior, which can be used to realize a light-weight calibration based on 3-4 reference images only. However, compared to the B-spline-based approach, results of the alternative demodulation approach are less accurate and merely render advantages in the reduced number of required reference images.

Regarding motion compensation, an axial-motion model leads to the assumption that, for moderate velocities, lateral displacement in form of mismatching phase values and intensities has the most influence onto acquired distance information. Hence, artifacts have been significantly reduced by applying optical flow-based image registration. The accuracy and computation speed of the motion compensation thereby depends on the utilized optical flow algorithm.

In the context of range image refinement, two feasible upsamling techniques for range images have been discussed – *Moving Least Square Surfaces* as an example of an explicit surface approximation as well as an enhanced *edge-directed upscaling filter*. Compared to each other, each approach has its individual advantages and disadvantages.

MLS surfaces originate from point-based rendering and provide a high order

reconstruction of surfaces defined by general point sets. Due to the interactive ray casting of the reconstructed surface, range image upsampling/visualization benefits from resolution and view independent insertion of visible surface points only. Outlier and flying pixels are implicitly removed if the number of neighbor pixels is below a given threshold. Small holes, in contrast, are automatically filled in. However, due the underlying assumption of a smooth and continuous surfaces, and the low sample-to-feature ratio of current TOF cameras, range image reconstruction / upsampling generally suffers from contour bleeding and undesirably smoothed features. More sophisticated approaches exist, but are computationally more intensive and are therefore mostly applicable for post-processing tasks.

Edge-directed upscaling on the other hand exploits the uniform sampling of range images and applies two dimensional filtering in image space. By applying a biquadratic upsampling scheme as well as an edge-directed adjustment of the sampling position, it provides comparable high order upscaling results and preserves sharp object contours. However, due to the pyramid-based upscaling, only scaling factors to the power of two are possible. Thus, especially for scaling factors greater than 2^3, the difference between desired and next possible scaling factor might be too large – which in most cases might result in wasted computing time.

The fusion technique outlined in the context of this thesis performs a re-projection of the reconstructed distance information. Unfortunately, due to the involved viewing transformation, re-projection-based techniques highly rely on accurate sensor registration as well as correct distance information, particularly with regard to the distance calibration of the involved TOF camera. Thus, as long as correct distance information cannot be assured, all sensors should be placed close to each other in order to align corresponding viewing rays as good as possible and avoid false mapping. A usual disadvantage of software-based sensor fusion arises from occlusion artifacts due to the different viewing position of the individual sensors. Here, software-based approaches are only capable to provide incomplete multi-modal information, which has to be handled accordingly.

Future Directions While the next generation of TOF cameras will provide higher resolutions and thus allow an automated and more precise estimation of intrinsic parameters and camera pose, the extensive task of distance calibration remains. Here, unfortunately no further improvements can be achieved without possible hardware modifications.

However, concerning motion artifacts, the presented approach still allows improvements as both, the lateral as well as axial motion compensation, do not account for mismatching phase values due to both, the distance-related light attenuation as well as the change of viewing rays.

In the context of accurate distance refinement, alternate approaches might lead to even better results. Considering a point cloud representation, for example,

fitting of basic geometrical primitives comprising planes, cylinders, cones and spheres provides a more accurate surface representation than interpolation or polynomials. Furthermore, primitive fitting allows a more accurate approximation of surface normals, while missing range information (holes) can be easily filled in. Secondly, primitive fitting typically involves range data segmentation and classification, which in turn can be used for tasks like object tracking.

Regarding software-based data fusion, alternative approaches might further improve fusion results by avoiding the error-prone geometry reconstructions and considering more sophisticated data mapping in image space instead. However, these approaches are rather complex and basically involve non-affine image deformations as well as adequate detection of multi-modal features between each sensor.

One general solution for multi-modal feature estimation is the consideration of mutual information and the according joined histogram between both images. The actual image registration is performed by minimizing the dispersion of the histogram, i.e. by aligning homogeneous regions, with respect to a given set of possible image transformations. Other approaches detect significant image features in all input images like points or edges, which accordingly have to be mapped onto each other. The main problem of sensor fusion in the image domain, however, is to find an adequate image transformation that handles view dependent disparities and occlusion in a correct way. Unfortunately, currently established image transformations are either inadequate or computational too intensive and therefore rather suitable for post-processing tasks only.

The ideal solution to the mentioned processing steps would be an efficient, overall denoising-refinement-fusion approach for sensors with varying modality. Hence, denoising and refinement can take advantage of the additional high resolution data, while data fusion benefit from implicit distance correction.

Bibliography

[3DV] 3DV Systems, http://www.3dvsystems.com.

[AA03a] Adamson A., Alexa M.: Approximating and intersecting surfaces from points. In *Proc. of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (2003), Eurographics Assoc., pp. 230–239.

[AA03b] Adamson A., Alexa M.: Ray tracing point set surfaces. In *Shape Modeling International* (2003), IEEE Computer Society, pp. 272 – 282.

[ABC*03] Alexa M., Behr J., Cohen-Or D., S. L., D. L., Silva C. T.: Computing and rendering point set surfaces. *IEEE Trans. on Visualization and Computer Graphics 9*, 1 (2003), 3 – 15.

[ABCO*01] Alexa M., Behr J., Cohen-Or D., Fleishman S., Levin D., Silva C.: Point set surfaces. In *Proc. of the Conference on Visualization* (2001), IEEE Computer Society, pp. 21 – 28.

[AF87] Ayache N., Faverjon B.: Efficient registration of stereo images by matching graph descriptions of edge segments. *Intl. J. of Computer Vision 1*, 2 (1987), 107 – 131.

[AK04a] Amenta N., Kil Y.: Defining point-set surfaces. *ACM Transactions on Graphics 23*, 3 (2004), 264–270.

[AK04b] Amenta N., Kil Y.: The domain of a point set surface. In *Eurographics Symposium on Point-based Graphics* (2004), pp. 139–147.

[Bar02] Barash D.: A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Trans. on Pattern Anal. and Mach. Intell. 24*, 6 (2002), 844–847.

[BB95] Beauchemin S. S., Barron J. L.: The computation of optical flow. *ACM Computing Surveys 27*, 3 (1995), 433 – 466.

[BCM05] Buades A., Coll B., Morel J.-M.: A non-local algorithm for image denoising. In *IEEE Conf. on Computer Vision and Pattern Recognition* (2005), vol. 2, pp. 60–65.

[BF82] Barnard S. T., Fischler M. A.: Computational stereo. *ACM Computing Surveys 14*, 4 (1982), 553–572.

[BFB94] Barron J. L., Fleet D. J., Beauchemin S. S.: Performance of optical flow techniques. *Intl. J. of Computer Vision 12*, 1 (1994), 43–77.

[BHMB08] Böhme M., Haker M., Martinetz T., Barth E.: Shading constraint improves accuracy of time-of-flight measurements. In *IEEE Conf. on Computer Vision and Pattern Recognition; Workshop on TOF Camera based Computer Vision* (2008), pp. 1 – 6.

[BK08] Beder C., Koch R.: Calibration of focal length and 3D pose based on the reflectance and depth image of a planar object. *Int. J. on Intell. Systems Techn. and App., Issue on Dynamic 3D Imaging* (2008), 285–294.

[BKP10] Bredies K., Kunisch K., Pock T.: Total generalized variation. *SIAM Journal on Imaging Sciences* (2010). accepted preprint.

[Bli77] Blinn J. F.: Models of light reflection for computer synthesized pictures. In *Proc. of the 4th Annual Conference on Computer Graphics and Interactive Techniques* (1977), pp. 192–198.

[Bro66] Brown D.: Decentering distortion of lenses. *Photometric Engineering 32*, 3 (1966), 444–462.

[BW99] Born M., Wolf E.: *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light.* Cambridge University Press, 1999.

[BWKS06] Bruhn A., Weickert J., Kohlberger T., Schnörr C.: A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. *Intl. J. of Computer Vision 70*, 3 (2006), 257–277.

[BWS05] Bruhn A., Weickert J., Schnörr C.: Lucas/kanade meets horn/schunck: combining local and global optic flow methods. *Intl. J. of Computer Vision 61*, 3 (2005), 211–231.

[CAN] Canesta Inc., http://www.canesta.com.

[CBM*03] Carr J. C., Beatson R. K., McCallum B. C., Fright W. R., McLennan T. J., Mitchell T. J.: Smooth surface reconstruction from noisy range data. In *Proc. of the 1st Intl. Conf. on Comp. Graphics and Interactive Techniques* (2003), ACM, pp. 119–126.

[CKCL09] Chiosa I., Kolb A., Cuntz N., Lindner M.: Parallel mesh clustering. In *Proc. Eurographics Symp. on Parallel Graphics and Visualization* (2009), pp. 33–40.

[DD02] Durand F., Dorsey J.: Fast bilateral filtering for the display of high-dynamic-range images. In *Proc. of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (2002), ACM, pp. 257–266.

[DS05] Dey T., Sun J.: An adaptive MLS surface for reconstruction with guarantees. In *Proc. of the 3rd Eurographics Symposium on Geometry Processing* (2005), Eurographics Association, pp. 43–52.

[DT05] Diebel J., Thrun S.: An application of Markov random fields to range sensing. In *Proc. of Advanced Neural Information Processing Systems* (2005), pp. 291–298.

[Fau93] Faugeras O.: *Three-Dimensional Computer Vision*. The MIT Press, 1993.

[FCOS05] Fleishman S., Cohen-Or D., Silva C. T.: Robust moving least-squares fitting with sharp features. *ACM Trans. Graph. 24*, 3 (2005), 544–552.

[FH08] Fuchs S., Hirzinger G.: Extrinsic and depth calibration of TOF-cameras. In *IEEE Conf. on Computer Vision and Pattern Recognition* (2008), pp. 1–6.

[FLM92] Faugeras O. D., Luong Q.-T., Maybank S. J.: Camera self-calibration: Theory and experiments. In *Proc. of the 2nd European Conference on Computer Vision* (1992), pp. 321–334.

[FP86] Forstner W., Pertl A.: Photogrammetric standard methods and digital image matching techniques for high precision surface measurements. *Pattern Recognition in Practice II* (1986), 57 – 72.

[FPR*09] Frank M., Plaue M., Rapp H., Köthe U., Jähne B., Hamprecht F. A.: Theoretical and experimental error analysis of continuous-wave time-of-flight range cameras. *Optical Engineering 48*, 1 (2009).

[GAL07] Gudmundsson S., Aanæs H., Larsen R.: Environmental effects on measurement uncertainties of time-of-flight cameras. In *Intl. Symposium on Signals, Circuits and Systems* (2007), pp. 113–116.

[GHJV03] Gamma E., Helm R., Johnson R., Vlissides J.: *Design Patterns*. Addison-Wesley Prof. Comp. Series. Addison-Wesley, 2003.

[Hei00] Heikkilä J.: Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22* (2000), 1066–1077.

[HS81] Horn B. K. P., Schunck B. G.: Determining optical flow. *Artificial Intelligence 17* (1981), 185–203.

[HSJS08] Huhle B., Schairer T., Jenke P., Strasser W.: Robust non-local denoising of colored depth data. In *IEEE Conf. on Computer Vision and Pattern Recognition* (2008), pp. 1–7.

[HSS97] Heidrich W., Slusallek P., Seidel H.-P.: An image-based model for realistic lens systems in interactive computer graphics. In *Proc. of the Conference on Graphics Interface* (1997), Canadian Information Processing Society, pp. 68–75.

[JG01] J. I. G., G. Y.: 3D imaging in the studio. In *Proc. of SPIE* (2001), vol. 4298, pp. 48 —- 56.

[JWS08] Jenke P., Wand M., Straßer W.: Patch-graph reconstruction for piecewise smooth surfaces. *13th International Fall Workshop Vision, Modeling, and Visualization 2008* (2008), 3–12.

[KBKL10] Kolb A., Barth E., Koch R., Larsen R.: Time-of-flight cameras in computer graphics. *Computer Graphics Forum 29*, 1 (2010), 141–159.

[KES07] Kraus M., Eissele M., Strengert M.: GPU-based edge-directed image interpolation. In *Proc. of the Scandinavian Conference on Image Analysis* (2007), vol. 4522/2007 of *Lecture Notes in Computer Science*, Springer, pp. 532–541.

[KFM*04] Kraft H., Frey J., Moeller T., Albrecht M., Grothof M., Schink B., Hess H., Buxbaum B.: 3D-camera of high 3D-frame rate, depth-resolution and background light elimination based on improved PMD (photonic mixer device)-technologies. In *OPTO* (2004).

[KK09] Keller M., Kolb A.: Real-time Simulation of Time-Of-Flight Sensors. *Simulation Practice and Theory 17* (2009), 967–978.

[Kol05] Kolluri R.: Provably good moving least squares. In *ACM SIGGRAPH Courses* (2005), ACM, pp. 213–222.

[KRI06] Kahlmann T., Remondino F., Ingensand H.: Calibration for increased accuracy of the range imaging camera SwissRanger^{TM}. In *Image Engineering and Vision Metrology* (2006).

[KW93] Knutsson H., Westin C.-F.: Normalized convolution: A technique for filtering incomplete and uncertain data. In *Proc.of the 8th Scandinavian Conference on Image Analysis* (1993), SCIA, Norwegian Society for Image Processing and Pattern Recognition, pp. 515–523.

[Lan00] Lange R.: *3D Time-Of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology.* PhD thesis, University of Siegen, 2000.

[LCLT07] Lipman Y., Cohen-Or D., Levin D., Tal-Ezer H.: Parameterization-free projection for geometry reconstruction. *ACM Trans. on Graphics 26*, 3 (2007), 22–27.

[Lev98] Levin D.: The approximation power of moving least-squares. *Mathematics of Computation 67*, 224 (1998), 1517–1531.

[Lev03] Levin D.: Mesh-independent surface interpolation. *Geometric Modeling for Scientific Visualization* (2003), 37–49.

[LHLW07] Lottner O., Hartmann K., Loffeld O., Weihs W.: Image registration and calibration aspects for a new 2D/3D camera. In *EOS Conf. on Frontiers in Electronic Imaging* (2007), pp. 80–81.

[LK81] Lucas B. D., Kanade T.: An iterative image registration technique with an application to stereo vision. In *Proc. of the 7th Intl. Joint Conference on Artificial Intelligence* (1981), pp. 674–679.

[LK06] Lindner M., Kolb A.: Lateral and depth calibration of PMD-distance sensors. In *Intl. Symposium on Visual Computing* (2006), vol. 2 of *LNCS*, Springer, pp. 524–533.

[LK07a] Lindner M., Kolb A.: Calibration of the intensity-related distance error of the PMD TOF-camera. In *Intelligent Robots and Computer Vision XXV* (2007), vol. 6764, SPIE.

[LK07b] Lindner M., Kolb A.: Data-fusion of PMD-based distance-information and high-resolution RGB-images. In *Intl. Symposium on Signals, Circuits and Systems* (2007), vol. 1, pp. 121–124.

[LK09] Lindner M., Kolb A.: Compensation of motion artifacts for time-of-flight cameras. In *Dynamic 3D Imaging* (2009), vol. 5742 of *LNCS*, Springer, pp. 16 – 27.

[LKR08] Lindner M., Kolb A., Ringbeck T.: New insights into the calibration of TOF sensors. In *IEEE Conf. on Computer Vision and Pattern Recognition* (2008), pp. 1–5.

[LLK08] Lindner M., Lambers M., Kolb A.: Data fusion and edge-enhanced distance refinement for 2D RGB and 3D range images. *Intl. J. on Intell. Systems Technologies and Applications 5*, 1 (2008), 344 – 354.

[LSKK10] Lindner M., Schiller I., Kolb A., Koch R.: Time-of-flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding* (2010). Accepted for publication.

[Lua01] Luan X.: *Experimental Investigation of Photonic Mixer Device and Development of TOF 3D Ranging Systems Based on PMD Technology.* PhD thesis, Department of Electrical Engineering and Computer Science, 2001.

[MCM04] Ma L., Chen Y., Moore K. L.: Rational radial distortion models of camera lenses with analytical solution for distortion correction. *Intl. Journal of Information Acquisition 1*, 2 (2004), 135–147.

[MDB] Middlebury database, http://vision.middlebury.edu/flow/.

[MES] Mesa Imaging, http://www.mesa-imaging.ch.

[MKF*05] Möller T., Kraft H., Frey J., Albrecht M., Lange R.: Robust 3D measurement with PMD sensors. In *Proc. of the 1st Range Imaging Research Day at ETH Zurich* (2005).

[MN88] Mitchell D. P., Netravali A. N.: Reconstruction filters in computer-graphics. *SIGGRAPH Computer Graphics 22*, 4 (1988), 221–228.

[MP79] Marr D., Poggio T.: A computational theory of human stereo vision. In *Proc. of the Royal Society of London* (1979), vol. 204, The Royal Society, pp. 301–328.

[MSKS04] Ma Y., Stoatto S., Kosecka J., Sastry S.: *An Invitation to 3D Vision*. Springer, 2004.

[OCV] OpenCV, http://sourceforge.net/projects/opencvlibrary.

[OGG09] Oztireli C., Guennebaud G., Gross M.: Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum 28*, 2 (2009).

[OK85] Ohta Y., Kanade T.: Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence 7*, 2 (March 1985), 139–154.

[OLG*07] Owens J. D., Luebke D., Govindaraju N., Harris M., Krüger J., Lefohn A. E., Purcell T.: A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum 26*, 1 (2007), 80–113.

[PBB*06] Papenberg N., Bruhn A., Brox T., Didas S., Weickert J.: Highly accurate optic flow computation with theoretically justified warping. *Intl. J. of Comput. Vision 67*, 2 (2006), 141–158.

[PMD] PMD Technologies, http://pmdtec.com.

[PMF85] Pollard S., Mayhew J., Frisby J.: PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14 (1985), 449 – 470.

[Rap07] Rapp H.: *Experimental and Theoretical Investigation of Correlating TOF-Camera Systems*. Master's thesis, University of Heidelberg, Germany, 2007.

[Reu06] Reulke R.: Combination of distance data with high resolution images. In *ISPRS Commission V Symposium on Image Engineering and Vision Metrology* (2006).

[RFSK08] Radmer J., Fuste P., Schmidt H., Kruger J.: Incident light related distance error study and calibration of the PMD-range imaging camera. In *Computer Vision and Pattern Recognition Workshops* (2008), pp. 1–6.

[RKN00] Recknagel R., Kowarschick R., Notni G.: High-resolution defect detection and noise reduction using wavelet methods for surface measurement. *Journal of Optics A: Pure and Appl. Optics 2* (2000), 538 – 545.

[ROF92] Rudin L. I., Osher S., Fatemi E.: Nonlinear total variation based noise removal algorithms. *Physics D 60*, 1-4 (1992), 259–268.

[RSC87] Reeves W. T., Salesin D. H., Cook R. L.: Rendering antialiased shadows with depth maps. In *Proc. of the 14th Annual Conference on Computer Graphics and Interactive Techniques* (1987), ACM, pp. 283–291.

[SBK08] Schiller I., Beder C., Koch R.: Calibration of a PMD camera using a planar calibration object together with a multi-camera setup. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (2008), vol. XXXVII, B3a, pp. 297–302.

[SBS05] Schall O., Belyaev A., Seidel H.-P.: Robust filtering of noisy scattered point data. In *IEEE/Eurographics Symposium on Point-Based Graphics* (2005), Eurographics Association, pp. 71–77.

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

[SD02] Stamminger M., Drettakis G.: Perspective shadow maps. In *Proc. of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (2002), pp. 557–562.

[SDK09] Schnabel R., Degener P., Klein R.: Completion and reconstruction with primitive shapes. *Computer Graphics Forum (Proc. of Eurographics) 28*, 2 (2009), 503–512.

[SH05] Sigg C., Hadwiger M.: Fast third order texture filtering. *GPU Gems 2* (2005).

[Sha49] Shannon C. E.: Communication in the presence of noise. In *Proc. Institute of Radio Engineers* (1949), vol. 37, pp. 10–21. Reprint as classic paper in: Proc. of the IEEE, vol. 86, no. 2 (1998).

[SK06] Stommel M., Kuhnert K.-D.: Fusion of stereo-camera and PMD-camera data for real-time suited precise 3D environment reconstruction. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems* (2006), pp. 4780–4785.

[SKE06] Strengert M., Kraus M., Ertl T.: Pyramid methods in GPU-based image processing. In *Workshop on Vision, Modelling, and Visualization* (2006), pp. 169–176.

[SKvW*92] Segal M., Korobkin C., van Widenfelt R., Foran J., Haeberli P.: Fast shadows and lighting effects using texture mapping. In *Proc. SIGGRAPH* (1992), pp. 249–252.

[Sla80] Slama C.: *Manual of Photogrammetry.* 1980.

[SPH08] Stürmer M., Penne J., Hornegger J.: Standardization of intensity-values acquired by time-of-flight-cameras. *IEEE Society Conf. on Computer Vision and Pattern Recognition Workshops* (2008), 1–6.

[TBF*05] T. O., B. B., F. L., G. B., B. R., A. H.: Swissranger SR 3000 and first experiences based on miniaturized 3d-tof cameras. In *Proc. of the First Range Imaging Research Day at ETH Zurich* (2005).

[Tem96] Temes G. C.: Autozeroing and correlated double sampling techniques. 45–64.

[TGN*06] Tejada E., Gois J., Nonato L. G., Castelo A., Ertl T.: Hardware-accelerated extraction and rendering of point set surfaces. In *Proc. of Eurographics/IEEE VGTC Symposium on Visualization* (2006), pp. 21–28.

[TM98] Tomasi C., Manduchi R.: Bilateral filtering for gray and color images. In *Proc. of the 6th Intl. Conference on Computer Vision* (1998), pp. 839–846.

[Tsa87] Tsai R. Y.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation 3*, 4 (1987), 323–344.

[WCH92] Weng J., Cohen P., Herniou M.: Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. on Pattern Analysis and Machine Intelligence 14*, 10 (1992), 965–980.

[Wei98] Weickert J.: *Anisotropic Diffusion in Image Processing.* PhD thesis, University of Copenhagen, 1998.

[Wil83] Williams L.: Pyramidal parametrics. *SIGGRAPH Computer Graphics 17*, 3 (1983), 1–11.

[WM94] Wei G. Q., Ma S. D.: Implicit and explicit camera calibration: Theory and experiments. *IEEE Trans. on Pattern Analysis and Machine Intelligence 16*, 5 (1994), 469–480.

[WS01] Weickert J., Schnörr C.: A theoretical framework for convex regularizers in PDE-based computation of image motion. *Intl. J. of Computer Vision 45*, 3 (2001), 245–264.

[XSH*98] Xu Z., Schwarte R., Heinol H., Buxbaum B., Ringbeck T.: Smart pixel – photonic mixer device (PMD). In *Proc. of Intl. Conf. on Mechatron. & Machine Vision* (1998), pp. 259–264.

[YYDN07] Yang Q., Yang R., Davis J., Nistér D.: Spatial-depth super resolution for range images. In *IEEE Conf. on Computer Vision and Pattern Recognition* (2007), IEEE Computer Society.

[Zha00] Zhang Z.: A flexible new technique for camera calibration. In *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2000), pp. 133–1334.

[Zit03] Zitova B.: Image registration methods: a survey. *Image and Vision Computing 21*, 11 (October 2003), 977–1000.

[ZPB07] Zach C., Pock T., Bischof H.: A duality based approach for realtime TV-L1 optical flow. In *Proc. of the DAGM Symposium on Pattern Recognition* (2007), pp. 214–223.

List of Figures

List of Tables

List of Symbols

A, B	2-tap PMD readout diodes	10
a	cross correlation amplitude	8, 9
$\mathbf{c} = (c_x, c_y)$	principal point	18
$c(\tau)$	cross correlation sample	8
c_0	speed of light	9
d	distance information	9
f	focal length in world coordinate units, i.e. millimeter	18
f_x, f_y	focal length in pixel units along the x- respectively y-axis	19
h	pixel intensity	9
I_i	ith phase image	9
K	intrinsic parameter matrix	19
$\mathbf{k} = (k_1, k_2)$	radial distortion coefficients	21
λ	mapping parameter	18
M	modelview matrix	18
m	measured distance information	38
ω	angular modulation frequency	8
$\mathbf{p} = (u, v)$	pixel coordinate in pixel units	19
ϕ	distance-related phase offset	8, 12
Π_1	standard perspective projection matrix	18
Π_f	projection matrix	18
r	reflected responce signal	8
$\mathbf{s} = (s_x, s_y)$	pixel size in world coordinate units	18
s	reference signal	8
$\mathbf{t} = (t_1, t_2)$	tangential distortion coefficients	21
τ	internal time shift	8
$\mathbf{X} = (X, Y, Z)$	3D world coordinate	18
$\mathbf{x} = (x, y)$	image coordinate in world coordinate units	18
$\mathbf{x}_d$	distorted image coordinate	20

Index