Dissertation

submitted to the

Combined Faculties for the

Natural Sciences and for Mathematics

of the Ruperto-Carola University of Heidelberg, Germany

for the degree of

Doctor of Natural Sciences

put forward by
Diplom-Physiker Martin Otmar Schmidt
born in Nürnberg

Oral examination: 5. November 2008

Spatiotemporal Analysis of Range Imagery

Referees: Prof. Dr. Bernd Jähne

Prof. Dr. Ulrich Platt

Zusammenfassung

Die vorliegende Arbeit befasst sich mit der Fragestellung, wie aus einer Tiefenbildsequenz das zugehörige dreidimensionale Bewegungsfeld bestimmt werden kann. Wir untersuchen das Signal von Tiefenkameras, die auf dem Laufzeitverfahren basieren und sich eines neuartigen optoelektronischen Bauelements bedienen, dem Photomischdetektor (PMD). Dieser liefert neben der Tiefe auch Informationen zur mittleren Strahlungsleistung und deren Modulationsamplitude. Wir erörtern wie dieser erweiterte Informationsgehalt genutzt werden kann.

Die Rekonstruktion eines Bewegungsfeldes aus einer Bildsequenz ist ein schlecht gestelltes inverses Problem und kann allgemeingültig nicht gelöst werden. Überdies enthält das raumzeitliche Signal einer PMD-Kamera diverse, teilweise sehr spezifische, systematische und statistische Fehler von explizit räumlicher wie zeitlicher Abhängigkeit (z.B. *Bewegungsartefakte*).

Wir analysieren die unterschiedlichen Fehler und entwickeln ein Verfahren zur Korrektur systematischer Tiefensignalfehler. Mit einem neuartigen *Two-State-Channel-Smoothing* verbessern wir von Rauschen und Ausreißern verfälschte Tiefenkarten. Wir erweitern das *Strukturtensorverfahren*, um damit erstmals den erweiterten Informationsgehalt der PMD-Kameras zur Verbesserung der Bewegungsschätzung zu nutzen und Aussagen zur Güte der Schätzung zu ermöglichen. Bei den entwickelten Algorithmen wurde darauf geachtet, dass deren Berechnungskomplexität eine Verwendung in eingebetteten Systemen nicht ausschließt. Die Algorithmen werden anhand von synthetischen und realen Einzelbildern wie auch Bildsequenzen überprüft.

Abstract

The present thesis handles the topic of how to determine the three dimensional motion field from a corresponding sequence of range images. We investigate signals given by range cameras that are based on the time-of-flight principle for which they employ the novel optoelectronic photonic-mixer-device (PMD). Its signal comprises information about the range, the mean radiant flux and its modulation amplitude. We discuss how to take advantage of this wealth of information.

The estimation of a motion field from image sequences is an ill-posed inverse problem which can not be solved in general. Moreover, the spatiotemporal signal of a PMD-camera is corrupted by several kind of, partially rather specific, errors of systematic and statistical nature depending explicitly on time and space (*e.g. motion-artifacts*).

We analyze those errors and develop a method to correct for systematic errors in the range signal. By means of a novel *two-state-channel-smoothing* we improve range images corrupted by noise and outliers. We use and extend the *structure tensor* approach to come for the first time to an improved motion estimate that exploits the PMD-signal and provides an inherent measure for its confidence. The presented algorithms were developed under the premise to be of a computational complexity that not forbids their application within an embedded system. They are tested on synthetic and real images and image sequences.

Acknowledgments

I gratefully acknowledge the support of many people who contributed in various ways to the completion of this thesis.

First of all I would like to thank Prof. Dr. Bernd Jähne for giving me the opportunity to work on various interesting topics of computer vision and for supervising my thesis. I am grateful for his kind support in both scientific and organizational issues. I thank Prof. Dr. Ulrich Platt for agreeing to act as the second referee.

Thanks go to the staff of the IWR and HCI that do an excellent job in keeping things running, especially to Barbara Werner and Karin Kubessa-Nasri for making bureaucracy less painful and Dr. Hermann Lauer, Markus Riedinger and Dr. Ole Hansen letting the data streams flow right were they should.

I am grateful to Pavel Pavlov, the most suave person I know, for giving work at the office a congenial feel and being a inexhaustible source of mathematical knowledge.

I would like to thank Dr. Michael Klar for giving me an introduction to camera calibration and support in various related algorithmic issues. A big thank-you goes to PD Dr. Ullrich Köthe, who always took the time to answer my questions on various image processing topics. Thanks to PD Dr. Christoph Garbe for giving suggestion and tips on various topics.

For proof-reading and comments on the thesis I am deeply grateful to Dr. Achim Falkenroth, Roland Rocholz, Claudia Kondermann, Zhuang Lin, Andreas Schmidt and Marion Zuber.

I enjoyed working at the lab, which I blame mostly the *Windis* for and in particular Dr. Kai Degreif for introducing me to the small wind-wave-flume, Dr. Achim Falkenroth, Roland Rocholz for the various discussion on water-wave-measurements, Dr. Uwe Schimpf, Alexandra Herzog offering always some tea, Kerstin Richter, Florian Huhn, Rene Winter, Steffen Haschler, and last but not least Dr. Günther Balschbach for giving excellent administrative support - thank you all.

With respect to the research done for the PMD-cameras I would like to thank Holger Rapp for all his work with the experimental setup, Mario Frank giving me access to his range measurements, Matthias Plaue for discussions about the PMD's working

principle, Dr. Markus Jehle for experimental and theoretical support, Michael Erz for the demodulation measurements and Dr. Hagen Spies for advice on range flow algorithms.

Many thanks go to Dr. Björn Menze and Dr. Michael Kelm (telling me what the prior does in the monastery and why he ROCks under the trees of a random forest), Dr. Linus Görlitz, Daniel Kondermann (helping Charon over the Styx), Dr. Ralf Küsters, Christoph Sommer, Dr. Nikolaos Gianniotis, Dr. Marc Kirchner, Prof. Dr. Fred A. Hamprecht, Björn Andres, Bernhard Renard, Michael Hanselmann, Frederik Kaster, Sebastian Boppel, Bjoern Voss, Jörg Greis, Stephan Kassemeyer, Lars Feistner and Natalie Müller.

I would like to thank all the people of the IWR, HCI and IUP that gave me a cheerful time in Heidelberg, particularly the members of DIP, MIP and IPA group.

Along my time at the IWR I worked together with numerous external collaborators on various projects whom I would like to thank as well.

Thanks to all members of the LOCOMOTOR team for interesting discussions and friendly collaboration; especially Dr. Ingo Stuke for giving me support with his multiple motion algorithms, Dr. Hanno Scharr for illuminating talks and Dr. Kai Krajsek for a crash course in Kriging.

I enjoyed working together with the members of the Bosch corporate research team in Hildesheim (CR/AEM5), in particular Henning Voelz; very likely this was the first and last time in my life, that I can say *my job* is to drive a BMW through sun, rain, and snow around Heidelberg.

I appreciate the support of PD Dr. Michael Felsberg by giving me access to his channel smoothing algorithm.

I would like to thank all collaborators within the Smartvision project, in particular Hermann Hoepken for the fruitful and pleasant cooperation on the demonstrator.

Working within the LYNKEUS project was a pleasant experience. Thanks to all collaborators and in particular to Sandra Stecher for her friendly and straight cooperation, Prof. Dr. Andreas Kolb and Maik Keller, giving me support for the TOF-Simulator and trying to find solutions for my application specific problems, and Stefan Fuchs for the egomotion sequences. I gratefully acknowledged the financial support of the BMBF within the project LYNKEUS (promotional reference: 16SV2296).

Last but not least, I would like to thank my parents, my brothers and my friends (in particular the *Kumperla*) for their words of encouragement and emotional support.

Contents

Chapter 1

Introduction

1.1 Motivation

There is a vast number of tasks in science and industry that involve the analysis of *image data*. Generally, these image data are images of two or three dimensional kind, *i.e.* projections of *features* of some physical object or scene on a two or three dimensional (regular) grid. These features are physical properties like the spectral reflectance of a surface (*i.e.* its color) or the absorbance of body tissue. The data is acquired with some imaging device like a (digital) camera, microscope, computer tomography scanner or magnetic resonance scanner, to name only a few. The sampled information can be scalar, vector valued (*e.g.* multispectral imaging) or even of tensorial kind.

Sometimes it is sufficient to analyze single images. But with the advances in computing power and storage capacity more and more one wants to utilize the additional information embedded in the temporal domain. Or spoken differently, some problems can be tackled properly only if their inherent dynamic nature is caught in image sequences.

Often the motion of single or multiple objects is of interest, like for time-to-collision estimation in automotive industry. Sometimes the temporal evolution of (non rigid, deformable) surfaces is studied, *e.g.* the motion of waves on the water-surface. Even if the dynamics itself are not of interest they still might need to be accounted for, because they introduce perturbations that need to be corrected. As for instance image registration techniques in medical imaging try to compensate the motion of a patient during the (long time) acquisition of an image series. Particularly for

many scientific applications an exact measurement of the motion field is of major importance (*e.g.* calculation of growth rates of plant leaves).

The calculation of a three dimensional physical motion field from a corresponding (2D) image sequences is however not an easy problem to solve, as we will discuss in the context of this thesis. In fact it can not be solved generally.

The recent development of the so called *photonic-mixer-device* (PMD) marks a substantial progress towards that goal to reconstruct the physical motion field from an image sequence. For with them not only the irradiance information (known as *gray value* of conventional image sensors) but additionally the distance to the observed (object-) surface can be acquired at the same time.

A PMD-sensor is an integrated circuit of an array of single PMD-pixels. It measures the distance based on the time-of-flight principle using an active illumination. The observed scene is illuminated by infrared, modulated, incoherent light which is reflected and gathered in an array of solid-state image sensors comparable to those used in common digital (CMOS/APS) cameras. The major benefits in comparison with range measurement techniques like stereo vision or laser scanners are

- real-time range- and brightness-image acquisition
- no correspondence problems and low algorithmic effort
- inexpensive (because conventional) manufacturing process
- compact and robust "off-the-shelf"-cameras are available

Before we continue, we need to state some words about the denotation of some terms we use throughout this thesis. If we are talking of the *PMD-technique* or simply the *PMD*, we mean sensors or cameras that are based on the principles of optoelectronic modulation based time-of-flight measurement, that we present in section 2.2. While the acronym PMD (photonic-mixer-device) relates to a specific realization of this technique protected by patent (held by *PMDTec*hnologies GmbH), we still use it for all similar realization as there is no other common acronym for this technique. With *gray* value sensors we denote conventional imaging devices (digital cameras) that measure the irradiance on the sensor pixels. By *3D motion field* we mean a two dimensional field of three dimensional velocity vectors that corresponds to the physical motion of surface patches projected on a two dimensional array (the sensor) by some optical receiver (an objective).

The image-signal of the PMD-sensor comprises information about the range, the mean radiant flux and its modulation amplitude. In the present thesis we show how to utilize this extended information content for the improvement of the range measurement itself and for the estimation of an exact 3D motion field corresponding to an acquired image sequence. We illuminate the PMD-measurement technique in the context of image processing, particularly motion estimation.

The signal of a current PMD-sensor bears several kind of errors. Some of them are rather specific to the PMD. For example *motion-artifacts* or a bias in the range-measurement that is modulated in its magnitude with increasing distance. The error types are of systematic and statistical nature and depend explicitly on time and space. We analyze those errors and develop a method to correct for the bias in the range signal.

The work presented in this thesis partially evolved under collaboration within the LYNKEUS project, funded by the *Federal Ministry of Education and Research* (BMBF). The project addresses problem solutions by means of 3D-vision for applications in automation engineering and robotic, autonomous vehicles, man-machine interaction, safety engineering, medical engineering and quality control. Therefore the computational efficiency was an important constrain for the algorithms to develop. For the solution of tasks that might be implemented on an embedded system (like *e.g.* motion estimation), we tried to come up with algorithms of not more than linear complexity in time and space and we avoided iterative algorithms whenever possible.

1.2 Outline

The single chapters of this thesis are not self-contained. The subject is just too complex for that to be possible. However, we tried to always refer to the respective sections in the thesis when a concept or term is used for the first time within another chapter and we do not believe it to be common knowledge. A reader who just wants to have a look at a specific section and is not about to read the thesis from beginning, is advised to use the electronic version of the thesis, as following the references via the hyperlinks is possible with ease. For the introduction to motion estimation in chapter 4 we recommend the electronic version too, because there is an illustrating animation embedded that (naturally) can be viewed only using the electronic version (with *Acrobat Reader*).

The thesis is split in two parts: *Theory* and *Experiments and Applications*. In the former we give the theoretical fundament for the proposed methods and analyze errors under simplified, idealized assumptions. In the latter we present and discuss results of the experimental analysis regarding data acquisition, still image processing and motion estimation. The content of the chapters is the following

Chapter 2 We give a short overview of common range measurement techniques and describe the relevant details of the PMD measuring principle for image processing. We specify the features of the PMD, both the advantageous and the problematic ones. We analyze its errors of common and specific nature and present a new method to correct for a specific kind of systematic error in the range measurement.

Chapter 3 We describe image processing particularly in the context of filtering. We explain the basics of linear filtering and introduce more advanced concepts of non-linear kind, like robust estimation. We show how the PMD-signal can be exploited by means of non-linear filtering and present an extension to *B-spline channel smoothing*, that we named *two state channel smoothing*, to improve range imagery in the presence of noise and outliers.

Chapter 4 We give an introduction to motion estimation and illuminate the general problem that it poses. We describe how a three dimensional motion field can be calculated from the PMD-signal, by extending the structure tensor approach to motion estimation for the specialties of an active vision system like the PMD. We describe a specific error of PMD-range measurement that occurs in the presence of motion, so called *motion artifacts*.

Chapter 5 We describe the experimental setup that was used to acquire image sequences and calibration data. We exemplify diverse errors of the PMD and correct for fixed-pattern-noise and systematic range deviations. Particularly we study the interdependence of range and amplitude, important for motion estimation.

Chapter 6 We present and discuss results for still image processing and motion estimation on synthetic and real-world data. We systematically investigate by various examples the basic difficulties in 3D motion estimation from range imagery. We give a proof-of-concept of the method, demonstrate its successful application on real-world data and discuss its limitations.

Chapter 7 We give a résumé of the achieved results and an evaluation of the PMD-technology w.r.t. motion estimation. We try to highlight important topics that should be addressed regarding the sensor technology as well as algorithmic aspects of motion estimation.

In the appendix III the reader may find a list of acronyms and abbreviations and a description of the notation used throughout the thesis. The author hereby apologizes if he should have failed to adhere this notation style anywhere and thereby diminished the comprehensibility of the text.

1.3 Contribution

The following is a list of what the author believes to be the novel contributions of this thesis

- a largely self contained presentation of 3D motion field estimation using the structure tensor approach extended for PMD-technology
- a concise overview on the most prominent error types of PMD-sensors, particularly those introducing interdependencies in the range and amplitude measurement
- a novel method to correct for a specific kind of systematic error in the range measurement
- an extension for B-spline channel smoothing to improve range imagery in the presence of noise and outliers, which exploits the technicalities of the PMD-signal
- an extension for the structure tensor approach to motion estimation that takes advantage of the wealth of information embedded in the PMD-signal, achieving betters motion estimates w.r.t. the aperture problem and in the presence of noise
- first time application of the structure tensor approach to 3D motion estimation on PMD-data

Part I

Theory

Chapter 2

Range Data and Time-of-Flight Measuring Principle

2.1 Optical Range Measurement Techniques

There is an abundance of techniques for measuring distances or *ranges** to objects. Objects can be punctiform like a star in the sky or spread-out like the ground of the sea below a ship. What kind of range or object is of interest depends on the specific application, just like the method that is suitable for doing the measurement. Range measurements can be contactless (via sound or light) or tactile. Tactile measuring devices can be very simple like a plumb line or fancy like an atomic force microscope.

Computer vision implies the use of contactless optical techniques for range measurement; in most cases used to find an accurate description of surfaces. The wealth of methods there are, demonstrates the importance of range measurement for all kinds of applications.

The most important optical range measurement techniques can be divided into three categories: *Triangulation*, *interferometry* and *time-of-flight* based methods.

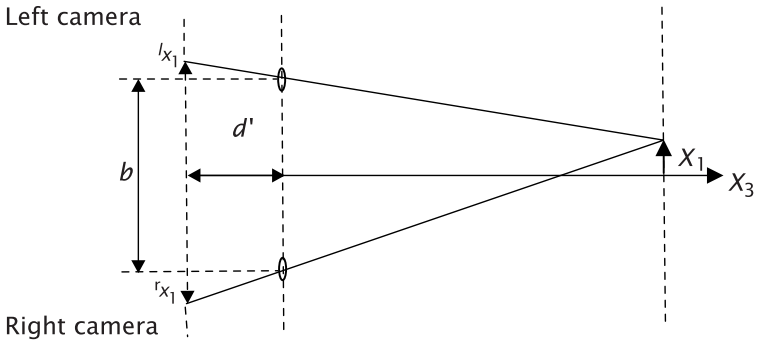


Figure 2.1: A stereo camera setup (from [Jäh02])

2.1.1 Triangulation

Triangulation is based on the fact that a scene is seen under a different viewing angle if the viewing position changes. The difference in the viewing angle causes a shift in the projected image, from which the distance to the elements of the scene can be inferred, if specific conditions are met.

In *Stereoscopy*, which is a specific realization of the triangulation principle, depicted schematically in figure 2.1, a point X is projected onto two different positions on the image plane of two different cameras with parallel optical axes, separated by a distance b, the *stereoscopic basis*. The difference in the position is called *disparity* or *parallax* p and is inverse proportional to the range X_3, as can be derived from geometrical optics:

$$p = {}^{r}x_1 - {}^{l}x_1 = d' \left(\frac{X_1 + b/2}{X_3} - \frac{X_1 - b/2}{X_3} \right) = b \, \frac{d'}{X_3} \tag{2.1}$$

Assuming the noise in the parallax measurement to be Gaussian, with a standard deviation σ_p, and applying the laws of error propagation, we deduce that the absolute precision of a range estimate decreases with the squared distance.

$$z \equiv X_3 = \frac{bd'}{p} \rightsquigarrow \sigma_z = \frac{b\,d'}{p^2}\,\sigma_p = \frac{z^2}{b\,d'}\,\sigma_p \tag{2.2}$$

*The commonly used term *range*, can be somewhat confusing, as it is also a synonym for domain; at passages where the meaning might be ambiguous, we will use the terms *depth range* and *distance* synonymous with range to clarify. Besides, from a physical point of view, a distance measurement is, because of Heisenberg's uncertainty principle, always associated with a range of distances (or rather a probability distribution) and never with a precise single value

Another variant of triangulation is *Active Triangulation*, which replaces one of the cameras by a light projector.

Triangulation suffers from the fact, that the observed object needs to be textured, because corresponding points in two images need to be identified. If this is not possible this is typically due to the so called *aperture problem* that can not be solved (see section 4.1.3 for a discussion of the problem in the context of motion estimation); then the parallax, which is essential for triangulation based range estimation, can not be determined. Moreover, one needs at least two, preferably calibrated, optical devices if the observed objects are moving.[†]Those systems are typically neither cheap nor easy to maintain, due to necessary calibrations, if not operated in an ideal environment.

2.1.2 Time-of-Flight Based Methods

The basic idea of time-of-flight based methods (TOF) is to determine the delay on a signal (typically on an electromagnetic carrier wave) induced by the time it needs to travel a certain distance. This delay is directly given as time τ needed to cover twice the distance z between a detector and an object, for a light pulse that travels to an object, is (diffusely) reflected from it and then detected on the same position from which it was sent:

$$\tau = \frac{2z}{c} \text{ , where } c \text{ is the speed of light.} \tag{2.3}$$

With the upper equation for an ideal time-of-flight measurement, we infer that in contrast to triangulation the precision of the distance measurement is independent of the distance but direct proportional to the precision of the time measurement τ:

$$z = \frac{c}{2}\tau \rightsquigarrow \sigma_z = \frac{c}{2}\sigma_\tau \ . \tag{2.4}$$

Using a light pulse or pulses (*i.e. pulse modulation*) and measuring the time delay is rather demanding regarding the speed of light and typical distances to be measured, because frequencies in the order of GHz and above need to be handled properly.

[†]If the objects are not moving while the image acquisition takes place, one could also move the camera to generate the parallax.

Using *continuous wave modulation* (CW), the carrier wave is modulated periodically with a frequency ν. Here, not a time delay but a phase shift ϕ between outgoing and incoming signal is determined:

$$z = \frac{c}{4\pi\,\nu}\,\phi \rightsquigarrow \sigma_z = \frac{c}{4\pi\,\nu}\,\sigma_\phi \ . \tag{2.5}$$

The phase shift can be determined by correlating the signals in time. This is less demanding and more robust with respect to tolerances of the used measuring device components compared to pulse modulation.

The chosen modulation frequency ν determines the distance range $\Delta z = c/2\nu$ that can be measured uniquely and which is known as *unambiguity range*. Because the phase shift ϕ used to calculate z is a cyclic variable, distances z' above Δz yield an erroneous range $z = z' \bmod \Delta z$.

2.1.3 Interferometry

Interferometry can be regarded as a special case of time-of-flight using continuous wave modulation, where the modulation is given by the frequency of the electromagnetic wave itself. The radiation needs to be coherent as otherwise the cross-correlation of out- and incoming signals can not be used to determine the phase shift, *i.e.* the waves do not interfere. For an electromagnetic wave, c is $\nu \cdot \lambda$. Substitution in equation (2.5) yields:

$$z = \frac{\lambda}{4\pi}\,\phi \rightsquigarrow \sigma_z \sim \frac{\lambda}{4\pi}\,\sigma_\phi \text{ and } \Delta z = \frac{\lambda}{2} \tag{2.6}$$

Detailed analysis shows that for classical interferometry, as realized in a Michelson interferometer, σ_z is proportional to the inverse of the distance or the aperture of the observation, as it features an optical averaging over the micro-topology of the object under investigation: $\sigma_z \sim z^{-1}$ ([Häu+99]).

To overcome limitations given by the small, wavelength-determined unambiguity range Δz, *multiwavelength interferometry* can be used. Due to speckle noise that occurs when coherent light is reflected from a rough surface, classical interferometry is only suited well for smooth surfaces.

White light interferometry, more precisely *coherency radar*, uses a radiation that has a coherence length of only a few wavelengths, thus interference patterns arise only for

distances of this coherence length. The range is found while scanning over a distance range and looking for interference of maximum contrast. As the interference contrast is used as the basis for the range measurement, unlike phase information in classical interferometry, speckle noise has no influence and rough surfaces can be measured.

There are numerous interferometry methods for all kinds of applications reaching the highest possible depth-resolution and competitive measuring ranges of several meters, at the cost of a sensitive, expensive, highly specialized and therefore inflexible setup.

2.2 The Photonic Mixer Device

The following section is a condensed description of the *photonic mixer device* (PMD) based on the work of Lange [Lan00]; Heinol [Hei01]; Justen [Jus01]; Schneider [Sch03] and our theoretical and experimental findings (see also chapter 5). The focus is on highlighting aspects that are relevant for the tasks of image processing, especially motion estimation, avoiding the very details of technical realization. Furthermore, we describe a new way to correct for systematic errors in the phase measurement and give a new formula for calculating the phase shift of purely rectangular modulated signals.

The *photonic mixer device* realizes TOF-range measurement by *continuous wave modulation*. Compared to more conventional systems, that do the necessary detection of the optical signal and its cross-correlation with the reference signal separately from each other, the PMD integrates both in one semiconductor based circuit. By moving the process of mixing and correlation of the signals from a separate electronic component to the integrated optoelectronic interface, the major sources of errors of conventional systems are avoided. Moreover, the parallelization of range point measurements on a sensor matrix, as it is essential for a 3D-camera system, is significantly simplified [see Sch+98]. Figure 2.2 illustrates the principle of concurrent detection and mixing, *i.e.* the simultaneous generation of photoelectrons and mixing with the electronic reference signal, which is the essential new feature of the PMD.

Today's PMD-sensors combine the functionality of the CCD-principle with the benefits of a realization in a CMOS-process see [Lan00]. The CCD principle offers an almost noise-free addition of optically generated charge carriers and defined local and temporal charge separation in the charge domain, being crucial for an optimal performance of a PMD. As the PMD successively adds short-time integrated sampling

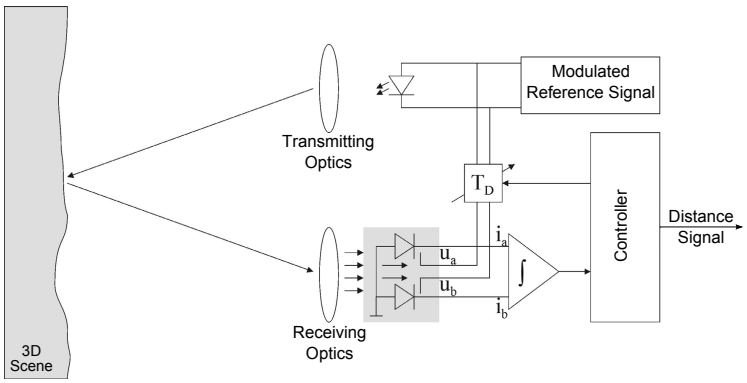


Figure 2.2: Principle of concurrent detection and mixing in the PMD (based on [Sch03])

points, a reasonable signal-to-noise ratio (SNR) can only be achieved, if this works practically free of noise. Furthermore, only the numerous addition of the short-time samples (*i.e.* the cross-correlation) suppresses the higher-order and non-harmonic frequencies in the modulation signal (that are unavoidable due to technical constraints) and allows the pixels to act as *lock-in pixels*, insensitive to frequencies other than the modulation frequency.

CMOS-processes are widely available and as such relatively cheap. They allow the implementation of the PMD as an active pixel sensor (APS), what means pixels with an active stage [Fos93]. An APS permits random access pixels (and therefore application specific performance enhancements) and an improved SNR. The CCD-principle can not only be realized within a CCD-process, but also within a CMOS-process. The maximization of the *charge transfer efficiency* CTE, that is one of the major benefits of the CCD-process, is of minor importance with respect to the PMD, as only a few charge transfers are needed (for details see [Lan00, chap. 5]). Using CMOS circuitry, first signal processing steps (like the temporal sampling, *i.e.* demodulation) can already be realized in the pixel, while maintaining reasonable fill factors. The fill factor is of major importance, because the performance of the PMD-camera depends on the modulated signal from an active illumination. As the intensity of back-scattered light decreases with the square of the target's distance to the camera, again the active illumination accounts for the need of a high dynamic range of the PMD-pixels.

The pixel's working principle is exemplary illustrated in figure 2.3a: The potential gradient in the semiconductor is controlled by applying proper gate voltages ($u_{mod,A}$

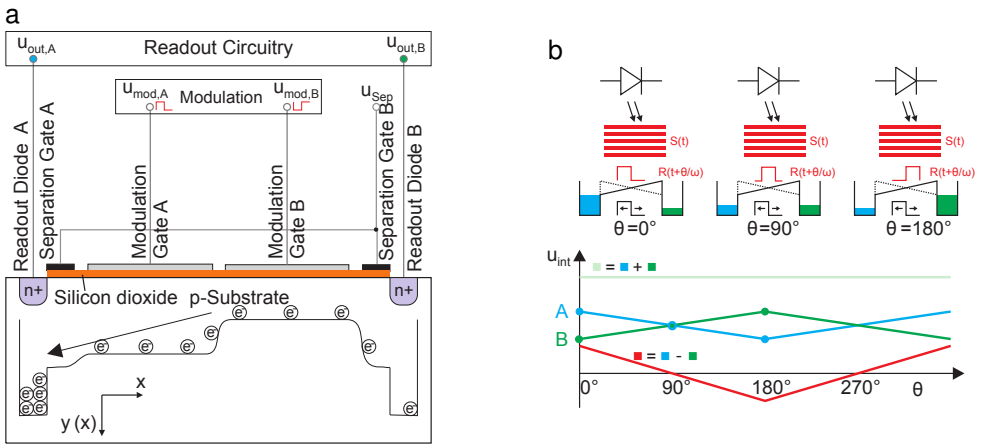


Figure 2.3: PMD-pixel's working principle: a schematic structure of the circuit b sampling of the cross-correlation function (adapted from [Alb07])

and $u_{mod,B}$) to the photogates A and B. The potential gradient is changed synchronously with the modulation of the incoming light, so the optically generated charge carriers are either driven into the left or the right integration gate (under the readout diodes A and B), each for half of the modulation period. For a typical modulation frequency of *e.g.* 20MHz and an integration time of 5ms these short time integrations repeat a hundred thousand times.

It is worth noting that within one short time integration (for the upper example, half a period is 25ns) only a few electrons (or none, depending on the optical power) are generated and transported to the integration gates. As the accumulation of single electrons under the integration gate is essentially noise free, this is actually an advantage, because *self-induced drift*[‡], which would lower the charge transfer to the integration gates, has no influence for the PMD [Lan00].

The accumulated charge in the capacity under one integration (or readout) gate, results in a measurable voltage u_{out} at the readout diodes. This voltage approximates a cross-correlation of the optically generated input signal $S(t)$ with the phase-shifted sampling signal $R(t)$, the phase-shift θ being the correlation variable. Figure 2.3b illustrates the sampling for 3 different phase-shifts θ at 0°, 90° and 180° and the

[‡] *self-induced drift* is induced by the coulomb forces between free charge carriers of same polarity, making them repel one another. It's of practical significance only, if many free charge carriers are located close to each other, and as such depends heavily on the number of generated electrons.

resulting voltages at the readout gates A and B. The electrooptical input-signal $S(t)$, indicated by the red bars, is always the same (because the scene does not change). Only the sampling (or reference) signal is phase shifted. If the readout gates A and B are identical w.r.t. the manufacturing process then $u_{out,A}(\theta + 0°) = u_{out,B}(\theta + 180°)$.

The output signal is only an approximation of the correlation, because not the sampling signal but some not necessarily linear function, dependent on the sampling signal and properties of the semiconductor circuit, describes what fraction of the photoelectrons are deposited in the integration capacity. For example a generated photoelectron may recombine with its hole before it reaches the capacity, because it was generated too far from it. In the ideal case, the fraction would be one, as soon as the potential drives the electrons to the corresponding integration gate, and zero otherwise. This way the correlation would be that of a square wave (its range being $\{0, 1\}$) with the electrooptical signal of the same period. The phase of the square wave is shifted by 180° for the two different gates. For the ideal case of a sinusoidal modulated electrooptical signal, the correlate is sinusoidal too: $\int_0^{2\pi} \sin(x + \varphi) \cdot H(\sin(x + \theta))\mathrm{d}x = 2\cos(\varphi - \theta)$ with H being the Heaviside step function.

2.2.1 Demodulation

The described process of demodulation shall now be put in mathematically formalism, making some idealizing assumptions.

The irradiance $E(t)$ seen by the PMD-sensor may be modeled as

$$E(t) = {G_0}' + A' \cdot M(t), \quad \text{with } M(t) = M(t + T). \tag{2.7}$$

$M(t)$ is a periodic function with (time-) period T (and angular frequency $\omega = 2\pi/T$) that originates from the modulated illumination of a spatial scenery. ${G_0}'$ is a constant irradiance offset and A' is the amplitude associated with the (normalized) modulation $M(t)$. If we assume the photosensitive semiconductor to have a linear responsivity, the induced electrooptical signal $S(t)$, which is proportional to the generated charge carriers, is

$$S(t) = G_0 + A \cdot M(t), \tag{2.8}$$

with G_0 and A just being linearly scaled versions of ${G_0}'$ and A'. Of course this is only an approximation as both G_0 and A are influenced by various optical and

electrooptical properties of the system (*e.g.* transmission of optical filters or quantum efficiency of the PMD). The details of these properties are discussed in [Lan00; Lua01; Jus01; Sch03] and are out of the scope of this thesis.

Because of the time it takes for the light to travel to the object and back again, the modulation $M(t)$ is delayed for a phase-difference φ with respect to a reference modulation M': $M(t) = M'(t-\varphi/\omega)$. Under ideal conditions, the PMD demodulates the signal $S(t)$ by correlating it with a discretized version of M', namely $R(t) = H(M'(t))$, H being the Heaviside step function. The cross-correlation function $I(\theta)$ reads

$$I(\theta) = \int_0^{mT} S(t) \cdot R(t + \frac{\theta}{\omega}) \mathrm{d}t. \tag{2.9}$$

2.2.1.1 Sinusoidal Modulation

Assuming the modulation being sinusoidal $M'(t) = \sin(\omega t)$, we find for a correlation range of m ($\in \mathbb{N}^+$) periods

$$\begin{aligned}
I(\theta) &= \int_0^{mT} (G_0 + A\,\sin(\omega t - \varphi)) \cdot H(\sin(\omega t + \theta)) \mathrm{d}t && \text{(2.10a)} \\
&= \int_0^{mT} (G_0 + A\,\sin(\omega t - \varphi - \theta)) \cdot H(\sin(\omega t)) \mathrm{d}t \\
&= m \int_0^{\frac{T}{2}} (G_0 + A\,\sin(\omega t - \varphi - \theta)) \mathrm{d}t \\
&= m \int_{\frac{\varphi+\theta}{\omega}}^{\frac{T}{2}+\frac{\varphi+\theta}{\omega}} (G_0 + A\,\sin(\omega t)) \mathrm{d}t && \text{(2.10b)} \\
&= m\,T \left(\frac{A}{\pi}\,\cos(\varphi + \theta) + \frac{G_0}{2} \right), && \text{(2.10c)}
\end{aligned}$$

considering that $H(\sin(\omega t))$ is one for half a period $[0, T/2]$ and zero otherwise.

Equation (2.10c) describes the idealized demodulation for a sinusoidal modulation that is done by the PMD. A generalized version of it is given by Plaue [Pla06] for modulations $M(t)$ that can be approximated by a Fourier series. $I(\theta)$ corresponds to the output of the PMD, and θ is the variable of the cross-correlation function that the PMD implements. Choosing a specific θ corresponds to sampling the cross-correlation function. According to equation (2.9) we may select a specifc θ by phase-shifting the reference signal $R(t)$ respectively.

With respect to a TOF-camera, we are interested in the unknowns φ, A and G_0. φ corresponds to the range r by $\varphi = 2\omega r/c$, c being the speed of light (see equation (2.5)). A is proportional to the amplitude of the modulation of the active illumination of the camera. G_0 sums up background illumination and the DC-component of the active illumination.

As we have three independent unknowns, we need at least 3 equations to find a solution. By taking samples of $I(\theta)$ for several θ between $[0, 2\pi]$, we obtain those equations. One equation (2.10c) for each sample. As the output of the PMD is subject to noise, it is reasonable to take more samples to find the optimal solution in a least squares sense; anyhow the implementation of the PMD, which was described above, returns at its two output gates in one shot two samples shifted by 180°. Taking another shot with 90° (and therefore also 270°) phase-shift gives us enough information to solve for the unknowns of equation (2.10c) optimally in a least square sense. Of course, one can take more than 4 samples to improve the variable estimation with respect to noise. For N equidistant sampling points, the optimal solution in a least squares sense is given by:

$$\begin{aligned} \varphi &= \arg\left[\sum_{n=0}^{N-1} I_n e^{-\imath\theta_n}\right] \\ A &= \frac{2\pi}{N}\left|\sum_{n=0}^{N-1} I_n e^{-\imath\theta_n}\right| \\ G_0 &= \frac{2}{N}\sum_{n=0}^{N-1} I_n \end{aligned} \qquad \text{with} \quad \begin{aligned} I_n &= \frac{I(\theta_n)}{m\,T}\ , \\ \theta_n &= 2\pi\frac{n}{N}\ . \end{aligned} \tag{2.11}$$

A proof for an equal formulations of equation (2.11) can be found in [Pla06] or [Xu99]. It simplifies for $N = 4$ to:

$$\begin{aligned} \varphi &= \arg\left[(I_0 - I_2) + \imath(I_3 - I_1)\right] \\ A &= \frac{\pi}{2}\sqrt{(I_0 - I_2)^2 + (I_3 - I_1)^2} \\ G_0 &= \frac{I_0 + I_1 + I_2 + I_3}{2} \end{aligned} \qquad \text{with} \quad \begin{aligned} I_n &= \frac{I(\theta_n)}{m\,T}\,, \\ \theta_n &= n\,\frac{\pi}{2}\,. \end{aligned} \tag{2.12}$$

Most PMD-based TOF-cameras use essentially these equations to determine phase and amplitude. However, the amplitude A calculated here is that of the electrooptical input signal, while formulas given in literature (*e.g.* [LS01]) often calculate the amplitude of the correlation signal. It is important to be aware of the difference, if it comes to the interpretation of A as a physical property of the optical signal: equation (2.12) compensates for the so called *demodulation contrast*. In particular, if the demodulation contrast depends on the measurement itself (*i.e.* it is not a constant) the difference is of relevance, as we will discuss in section 2.2.1.3 on page 20.

2.2.1.2 Rectangular Modulation

Now let us suppose that the modulation is not sinusoidal but rectangular. In equation (2.10b) we then have to replace sin by $\operatorname{sgn} \circ \sin$, *i.e.* a square wave with range $\{-1, 1\}$. The resulting correlation function is a triangle wave:

$$\begin{aligned} I(\theta) &= m \int_{\frac{\varphi+\theta}{\omega}}^{\frac{T}{2}+\frac{\varphi+\theta}{\omega}} (G_0 + A \cdot \operatorname{sgn}(\sin(\omega t)))\mathrm{d}t \\ &= m\,T\left(\frac{A}{\pi}\,\operatorname{tri}\left(\varphi + \theta\right) + \frac{G_0}{2}\right), \end{aligned} \tag{2.13}$$

where $\operatorname{tri}(\phi) = \frac{\pi}{2} - \arccos(\cos(\phi))$, *i.e.* a triangle wave with range $[-\frac{\pi}{2}, \frac{\pi}{2}]$ and $\operatorname{tri}(0) = \frac{\pi}{2}$.

If we apply equation (2.12) to correlation samples (2.13) that result from a rectangular modulation, there is a systematic error in the phase estimation, as the model assumption of a sinusoidal modulation do not apply. We will discuss this in detail

in section 2.2.2. Here we just want to state an exact solution for the unknowns in equation (2.13) given 4 equidistant sampling points:

$$\begin{aligned} A &= \max\left[\left|(I_0 - I_2) + (I_1 - I_3)\right|, \left|(I_0 - I_2) - (I_1 - I_3)\right|\right] \\ \varphi &= \operatorname{sgn}(I_1 - I_3) \cdot \left(\frac{I_0 - I_2}{4\,A} + \frac{1}{4}\right) + \frac{1}{2} \\ G_0 &= \frac{I_0 + I_1 + I_2 + I_3}{2} \end{aligned} \qquad \text{with} \quad \begin{aligned} I_n &= \frac{I(\theta_n)}{m\,T} \\ \theta_n &= n\,\frac{\pi}{2} \end{aligned} \tag{2.14}$$

The author found the solution by analyzing the symmetries in the 4 correlation signals $I(\theta_n, \varphi)$ (2.13) and checked it by inserting these in the solution (2.14) using a computer algebra system. However, he is not sure if it is really new, as the problem seems to be a quite common; but he could find nothing similar in PMD-related literature.

2.2.1.3 Demodulation Contrast

In optics, the *modulation transfer function* MTF is of major importance for the description of an optical system. It is based on the modulation (or *modulation contrast*) M, which typically refers to a spatial pattern:

$$M = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} \tag{2.15}$$

$$MTF = \frac{M_{image}}{M_{object}}. \tag{2.16}$$

L is the radiance (or luminance) and M_{object} and M_{image} the modulation of an object and its image. If the object has a modulation of 100% (if $L_{min} = 0$) the MTF cancels to M_{image}. Equation (2.16) is correct only if we assume there is only a single frequency (of e.g. a spatial pattern), as the MTF depends on the frequency. More precisely, the MTF is the magnitude of the (optical) *transfer function* of a (optical) linear system (see section 3.1.4), and describes the response of an optical system to an image decomposed into sine waves.

The *demodulation contrast* is defined similarly but refers to a modulation in time:

$$C_{demod} = \frac{\text{measured amplitude}}{\text{measured offset}} \tag{2.17}$$

It quantifies the PMD's performance of charge separation. The amplitude of the sampled correlation $I(\theta)$ (2.10c), assuming a perfect charge separation, is with respect

to the integration time $m\,T$ attenuated by a factor of $1/\pi$ relative to the original modulation of $S(t)$; the offset is $G_0/2$. If we assume a modulation contrast of 100% of the electrooptical signal $S(t)$ then $G_0 = A$. So the demodulation contrast of this idealized PMD is:

$$C_{demod} = \frac{\frac{A}{\pi}}{\frac{G_0}{2}} = \frac{\frac{A}{\pi}}{\frac{A}{2}} = \frac{2}{\pi} \approx 64\%.$$

The various realizations of the PMD, however, neither have a demodulation contrast of this magnitude nor it is constant. It tends to be below 40%. Moreover, C_{demod} depends on the modulation frequency and the radiant energy deposit on a PMD-pixel (and other hardwired system features, that are of no interest in the scope of this thesis) during the exposure [Lan00; Lua01].

While the dependence on frequency is negligible for motion estimation, as typically the modulation frequency is not changed during image acquisition, the dependence on radiant energy is of major importance: We want to use the amplitude of equations (2.12) or (2.14) as a measure for the radiance emitted from a (moving) object-patch in the scene. We then can use this information in a physical motivated model of the scene for motion estimation, as will be discussed in section 4.1. If however the demodulation contrast itself depends on the radiance, we have to compensate for this, as (2.12) and (2.14) only apply for constant C_{demod}.

This can be achieved by doing a calibration measurement C_{demod} subject to G_0 and use it to correct the measured amplitude for the varying contrast. Then for example the amplitude calculated for a sinusoidal modulation in (2.12) becomes:

$$A \sim \frac{\sqrt{(I_0 - I_2)^2 + (I_3 - I_1)^2}}{C_{demod}(G_0)}. \tag{2.18}$$

We dropped for the sake of simplicity any constants of proportionality because they are marginal within the context of motion estimation. Furthermore, equation (2.18) only holds true if there is no background illumination during calibration and image acquisition, as the demodulation contrast depends on background illumination itself.

2.2.2 Error Analysis

Current PMD camera types show various errors — systematic as well as random ones — in their range signal. We first investigate the systematic errors, which result

from deviations of the technical realization from the model assumptions that were used to derive equation (2.11). Most of them may be corrected by an appropriate calibration. To do a proper calibration, we need at least a model for the errors and a way to measure them. Hence, in the following we will describe the errors, model them, and show ways to correct them; how the errors are measured is part of chapter 5.

2.2.2.1 Systematic Errors

The investigated systematic errors are

- a periodic, sinusoidal deviation of the phase measurement over the unambiguity range
- a constant phase deviation per pixel, which corresponds to the dark-response nonuniformity (DRNU) of classical CMOS sensors (often laxly called "fixed pattern noise") but has quite different reasons
- overexposure of individual pixels
- an exposure time dependent constant phase deviation

There are some more known systematic errors that will not be addressed here but are nevertheless of some importance:

- Phase drift due to thermal effects
- Near field errors due to the extended, non-punctiform, non-radial (in respect to the object lens) modulated illumination

Periodic Phase Error If we assume a periodic modulation of the illumination which induces a modulation of the number of emitted photoelectrons and assume the reference signal to be modulated with the same frequency, then the samples the PMD-sensor returns are those of a periodic signal too. The reason for this is that the PMD performs an operation that corresponds to cross-correlating the light- and reference signal as discussed in section 2.2.1. The sampled signal however is not necessarily a sinusoidal one. If for example the light and reference signal are assumed to be rectangular, then the cross-correlation is a triangle wave (see equation (2.13)).

Applying

$$\varphi_{\sin}(\vec{I}) = \arg\left[(I_0 - I_2) + \imath(I_3 - I_1)\right]$$

on the vector $\vec{I}_{\text{tri}}$ of cross-correlation samples

$$I_{\text{tri}}(\theta_n) \sim \frac{A}{\pi}\ \text{tri}\left(\varphi + \theta_n\right) + \frac{G_0}{2},$$

yields a periodic error in the phase estimation:

$$E_\varphi(\varphi) = \varphi_{\sin}(\vec{I}_{\text{tri}}) - \varphi \sim \arg\left[\arcsin\left(\cos(\varphi)\right) + \imath \arcsin\left(\sin(\varphi)\right)\right] - \varphi \tag{2.19}$$

$$\approx (3\frac{\pi}{8} - \arctan(3))\sin(4\varphi) \approx \frac{1}{14}\sin(4\varphi) \tag{2.20}$$

Thus the maximum absolute error is $1/14/2\pi = 1.14\%$ of the unambiguity range. The error of the approximation using $\sin(4\varphi)$ is less then 10% of the phase error. Because motion estimation deals with derivatives of range measurements, more important than the absolute error, is the relative error with respect to the slope of $\varphi_{\sin}$:

$$E_{rel} = \frac{\partial_\varphi E_\varphi}{\partial_\varphi \varphi} = \partial_\varphi E_\varphi \approx \frac{4}{14}\cos(4\varphi).$$

This means that we introduce a maximum error of $4/14 \approx 29\%$ if we calculate range-slopes based on measurements that assume that the modulation is sinusoidal, while it is actually rectangular.

Fourier Approximation If we have a look at the testbench range measurements (see section 5.2.2), we find that the systematic error has indeed a major component going with $\sin(4\varphi)$. However, there are additional smaller components of higher and lower harmonics of the angular base frequency $n = 1$. Therefore, we may approximate the error by a Fourier series:

$$E'(\varphi) = \mathit{offset} + \sum_{n=1}^{k}(a_n \sin(n\varphi + \theta_n)) \tag{2.21}$$

The Fourier coefficients can be found numerically by doing a least squares fit. If data can be acquired for the whole phase range of 2π one may use also a FFT. The erroneous phase measurement φ_{err}, then is described by

$$\varphi_{err}(\varphi) = \varphi + \mathit{offset} + E(\varphi), \quad \text{where} \quad E(\varphi) = E'(\varphi) - \mathit{offset}, \tag{2.22}$$

and by defining $\phi(\varphi) = \varphi_{err}(\varphi) - \mathit{offset}$, we get rid of the constant *offset* error:

$$\phi(\varphi) = \varphi + E(\varphi).$$

We need to find the inverse function $\varphi(\phi)$, if we want to correct the measured data ϕ to become the true value φ. As there is no analytic exact solution for the inverse function we may take $E(\varphi)$ to be a small perturbation and approximate the inverse function as the inverse of its Taylor polynomial.

The first order Taylor series of $\phi(\varphi)$ at $\varphi = \varphi_0$ is

$$\phi(\varphi) = \varphi_0 + \mathrm{E}(\varphi_0) + \partial_\varphi E(\varphi_0) \cdot (\varphi - \varphi_0) + O\left((\varphi - \varphi_0)^2\right) \ .$$

The inverse of the Taylor series (given by *Mathematica*) is

$$\phi^{-1}(\phi(\varphi)) = \varphi(\phi) = \varphi_0 - \frac{\varphi_0 - \phi + E(\varphi_0)}{\partial_\varphi E(\varphi_0) + 1} + O\left((\varphi_0 - \phi + E(\varphi_0))^2\right)$$

Choosing φ_0 to be ϕ we obtain [§]

$$\begin{aligned} \varphi(\phi) &= \phi - \frac{\phi - \phi + E(\phi)}{\partial_\varphi E(\phi) + 1} + O\left(E(\phi)^2\right) \\ &\approx \phi - \frac{E(\phi)}{\partial_\varphi E(\phi) + 1}. \end{aligned} \tag{2.23}$$

The inverse Taylor polynomial (2.23) is a good approximation for $\varphi(\phi)$ if $|E(\phi)| \ll 1$ and $E(\phi) \approx E(\varphi + \partial_\varphi E(\varphi) \cdot E(\varphi)) \overset{!}{\approx} E(\varphi)$ (which conditions $\partial_\varphi E(\varphi)$ to be small too), because only then the remainder indicated by $O(E(\phi)^2)$ is small compared to $E(\varphi)$ (the error that needs to be corrected). Additionally $\phi(\varphi)$ is invertible only if it is monotonic, implying that $\partial_\varphi E > -1$. All requirements are fulfilled if the Fourier-coefficients a_n and the number of modes k needed to describe the error are small, which is in good agreement with the measurements.

Equation (2.23) is a compact, analytic solution for the problem of correcting a phase error that can be described by equation (2.21). If necessary an approximation by a

[§] mathematical more precise we take $\varphi(\varphi_0)$ in the limit of ϕ: $\lim_{\varphi_0 \to \phi} \varphi(\varphi_0)$

higher order Taylor polynomial can be derived with ease, at the cost of increasing the resources needed to calculate the inversion. The necessary data for calculating the Fourier coefficients has to be acquired during calibration of the camera. Typically $k = 4$ Fourier modes are sufficient to suppress the error considerably (see chapter 5 for results). Compared to a lookup table or B-spline approximations [LK06; KRI06] (in extreme cases for every pixel) this is very efficient with respect to needed memory resources and acceptable regarding processing time.

Constant Phase Error per Pixel Typical images of conventional CCD- or CMOS-cameras show two types of pixel specific systematic errors (both being more prominent for CMOS sensors): *dark-response nonuniformity* (DRNU) and *photo-response nonuniformity* (PRNU), that can be directly related to the pixels' varying offset and gain (due to *e.g.* variations in oxide thickness and doping concentrations over the sensor). Sometimes these (in respect to the measurement *systematic*) errors are somewhat sloppy called *fixed pattern noise*. Range imagery appears to have the same kind of nonuniformity errors. But differences in offset and gain (assuming a linear gain) should actually have no influence on the phase measurement, because offset and gain just cancel out of equation (2.12).

The explanation for the pixel nonuniformity is that the reference signal $R(t)$ (of equation (2.9)) connected to each pixel receives a phase delay due to the slightly varying capacitance of the individual pixels and other hardware-design and -processing related reasons that may affect the phase of R. As the error is constant over time it can be corrected using appropriate calibration methods.

Let $\boldsymbol{K}$ denote the matrix of fixed pattern phase errors (where the matrix elements correspond to image pixels) and $\boldsymbol{N}$ the (temporal) noise corresponding to a field of random variables, such that the expectation value $\langle \boldsymbol{N} \rangle$ is $\mathbf{0}$. Then $\overline{\boldsymbol{N}}^t \approx \mathbf{0}$, the bar denoting the (temporal) arithmetic mean over a sequence of acquired data. With the true range at each pixel given by $\boldsymbol{T}$, we may model a taken range (or phase) image $\boldsymbol{R}$ as:

$$\boldsymbol{R} = \boldsymbol{T} + \boldsymbol{K} + \boldsymbol{N}. \tag{2.24}$$

Taking the arithmetic mean over a sequences of frames of a fixed view we get:

$$\overline{\boldsymbol{R}}^t = \overline{\boldsymbol{T}}^t + \overline{\boldsymbol{K}}^t + \overline{\boldsymbol{N}}^t \approx \boldsymbol{T} + \boldsymbol{K}.$$

If we are able to create a homogeneous incident illumination with respect to phase (and preferably irradiance), *i.e.* $\boldsymbol{T} = \mathfrak{I} \cdot T$, the estimation of $\boldsymbol{K}$ is easy; we regard the

fixed pattern error $\boldsymbol{K}$ as a sample (*taken* once during manufacturing of the sensor) of a field of i.i.d. random variables that have an expectation value of zero, then the spatial average over $\boldsymbol{K}$ is approximately zero and we just have to subtract the spatiotemporal average $\overline{\boldsymbol{R}}^{st}$ from $\overline{\boldsymbol{R}}^{t}$:

$$\overline{\boldsymbol{R}}^{t} - \overline{\boldsymbol{R}}^{st} \approx (\boldsymbol{T} + \boldsymbol{K}) - (\overline{\boldsymbol{T}}^{st} + \overline{\boldsymbol{K}}^{st} + \overline{\boldsymbol{N}}^{st}) \approx (\boldsymbol{T} + \boldsymbol{K}) - \boldsymbol{T} = \boldsymbol{K}. \tag{2.25}$$

A homogeneous phase may be achieved by modifications to the camera hardware using a telecentric illumination, but involves a complex and somewhat expensive experimental setup. Also the assumption of i.i.d. may be violated as the variations in the manufacturing process are not necessarily spatially uncorrelated.

If we use a simple whiteboard-like target for the calibration, a paraboloid-like phase pattern is irradiated on the sensor. If we remove the lens from the camera this improves the homogeneity of the data, and neighborhoods of the image pixels may be approximated by planar surface patches. The average over a symmetric neighborhood of a central pixel then is the value of the central pixel itself. So we may estimate $\boldsymbol{K}$ via

$$\overline{\boldsymbol{R}}^{t} - \mathcal{B}\overline{\boldsymbol{R}}^{t} \approx (\boldsymbol{T} + \boldsymbol{K}) - (\mathcal{B}\boldsymbol{T} + \mathcal{B}\boldsymbol{K}) \approx (\boldsymbol{T} + \boldsymbol{K}) - \boldsymbol{T} = \boldsymbol{K}. \tag{2.26}$$

$\mathcal{B}$ denotes binomial convolution of an appropriate mask size, which is large enough to let $\mathcal{B}\boldsymbol{K} \approx \boldsymbol{0}$, while small enough that the approximation of a pixel neighborhood as a planar surface patch is still valid and $\mathcal{B}\boldsymbol{T} \approx \boldsymbol{T}$. Image borders may be treated with respect to convolution by mirroring the data at the image borders. For results we refer to section 5.2.1.

The demodulation images $\boldsymbol{I}_n$ (2.11), from which $\boldsymbol{R}$ is calculated, are subject to DRNU and PRNU just like conventional image sensors. The data may be corrected for both errors by doing a calibration analog to conventional systems, if the modulated illumination is replaced by an unmodulated one, as then the PMD acts essentially like an irradiance sensor. The channels $\boldsymbol{I}_n$ may be corrected individually in a preprocessing step and its result used for calculating an improved phase and amplitude estimate. As explained before, offset and gain variations are of minor importance for the phase estimation, but the amplitude estimate is essentially influenced by gain. Looking at equation (2.11) we find that if the single demodulation images depend on a common image of gain factors $\boldsymbol{\alpha}$, the same is true for the amplitude estimate:

$$\text{if } \boldsymbol{I}_n \sim \boldsymbol{\alpha} \xrightarrow{(2.11)} \boldsymbol{A} \sim \boldsymbol{\alpha}$$

For a description of methods like *photon-transfer technique*, *flat-fielding* or statistical approaches and their application to correct for DRNU and PRNU we refer to [MF81; Fow+98; TRK01; Wag03; Grö03; Kla05].

Overexposure and Saturation For a gray value sensor overexposure occurs if the full well capacity (*i.e.* the saturation level) of a sensor pixel is exceeded. Then one can no longer map from the measured signal on the irradiance or the physical property of interest (in our case the distance to an object) even if the sensor response is known. If we want to correct for overexposure we first have to detect it.

A simple but somewhat unreliable method to detect heavy overexposure, even if one does not have access to the cross-correlation samples I_n, but only to the calculated amplitude A (2.11), is to check if A is zero; because for heavy or *total* overexposure, all capacities are saturated and all I_n are equal and thus A calculates to zero. However, this only works if really all samples I_n are saturated, which is not the case for *partial* overexposure, as we may see. Furthermore, $A = 0$ is ambiguous w.r.t. underexposure which may lead to an amplitude of zero too. So both *false positive* and *false negative* rate with respect to detection of overexposure may be high.

Overexposure of a PMD-pixel depends on both radiance *and* distance of an imaged object. Using equation (2.10c) and the inverse-square law for a punctiform light source, we come to a simplistic approximation of the demodulation signals I_n dependent on the distance r of an object of constant reflectivity:

$$I_n(r) \sim \frac{A(r)}{\pi}\cos(r\frac{2\pi}{R} + n\frac{\pi}{2}) + \frac{A(r)}{2C_{mod}} \sim \frac{\frac{2C_{mod}}{\pi}\cos(r\frac{2\pi}{R} + n\frac{\pi}{2}) + 1}{r^2}. \tag{2.27}$$

We assume the modulation contrast C_{mod} of the light source to be 100%, the unambiguity range $R = 7.5\text{m}$ ($\equiv f_{mod} = 20\text{MHz}$). Normalizing the demodulation signals by I_0 yields figure 2.4. Taking *e.g.* $I_2(4\text{m})$ to be exactly the saturation level of a sensor pixel, we find that the other demodulation samples are not in saturation yet. Or vice versa only if $I_0(4\text{m})$ is at saturation level, we can be sure to detect overexposure by testing if all samples are equal. We may denote this two kinds of overexposure specific to the PMD as *partial* and *total* overexposure. Furthermore, a maximum signal ratio of more than 4 indicates that irradiance needs to be high, such that an overexposure can be detected by testing on $A = 0$.

If the raw data is technical accessible, it is more reliable to check if any I_n is saturated and then, if positive, to flag the measurement as overexposed. Detected single

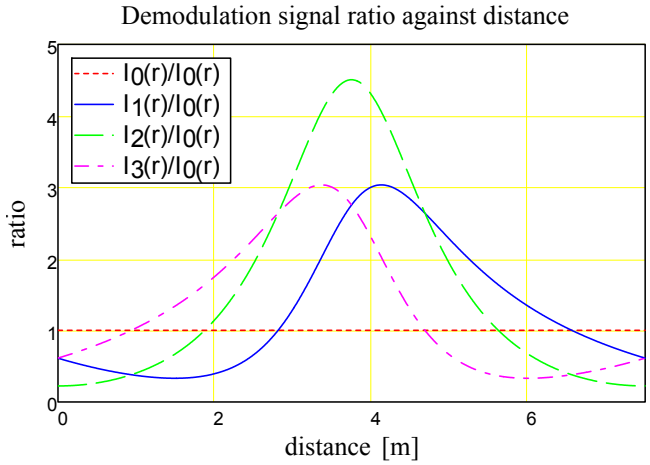
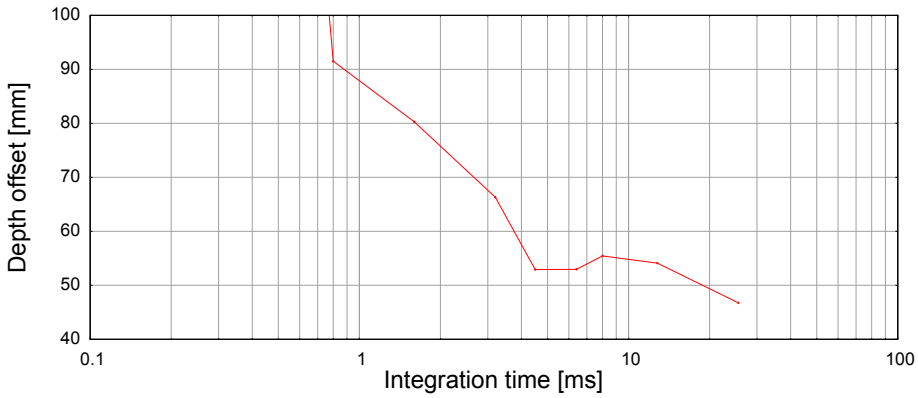


Figure 2.4: Demodulation signal ratio against distance [m]

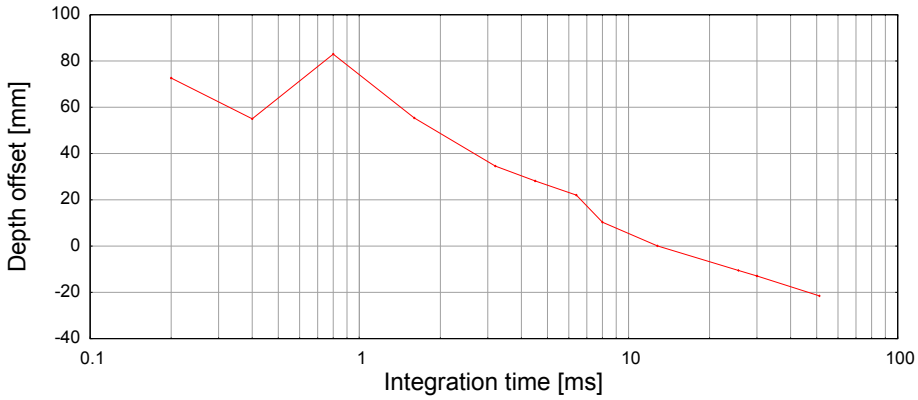
pixels or pixel regions then may be corrected (under specific assumption about the neighborhood) using techniques like *inpainting* (see *e.g.* [Tsc06] for an elaborated example) or simpler interpolation techniques, or may be excluded from further processing if possible. Overexposure may lead to effects like *blooming* (see [Jäh04]) so that the confidence in the information content of the neighbor-pixels shall be reduced.

Exposure Time / Amplitude Dependent Phase Deviation Experimental data shows that current PMD-camera types bear a systematic error in phase measurement which is constant for a specific exposure time (or *integration time* with respect to the process of cross-correlation) and independent of the measured range. Figure 2.5 gives an overview of the error for some PMD-type range cameras.

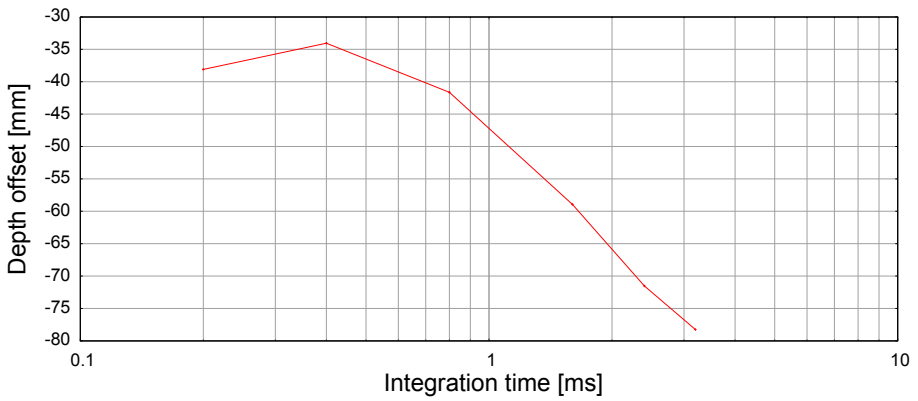
The simplistic models discussed so far, can not give an explanation for this dependency. A probable explanation could be that the electronics that control the phase shifting of the reference signal are somehow correlated to the integration time circuit. We are not fully convinced that the error explicitly (and exclusively) depends on the integration time. [Rap07] argues that the error is not related to amplitude as otherwise it would change with depth (while the measurements show that it is constant). However, we observed a similar constant offset that is rather constant with increasing depth and only depends on the reflectivity of the observed surface (see section 5.2.3). Therefore exposure time, amplitude A and also offset G_0 (the latter both depending on the reflectivity) are candidates for being the source of the



(a) PMD 19k



(b) SR-3000



(c) IFM-O3D

Figure 2.5: Range error against integration time (at r = 2.5m) [Rap07]

error. The author thinks that there is some kind of bias in the phase-calculation formula, that turns up if the idealized model assumptions are not met.

A exposure time dependent error is of special interest for cameras that use adaptive or multiple exposure times (like the IFM O3D) to increase the effective measurable range. Then an offset correction needs to be applied specific for every exposure time. For details regarding the experimental investigation we refer to [Rap07]. One may subsume this error under fixed pattern error if it is extended for exposure dependence.

2.2.2.2 Random Errors

The essential noise sources for PMD-sensors are the same as for conventional CCD- or CMOS-image sensors. Theses are electronic and photon *shot noise*, *thermal noise*, *reset noise*, $1/f$ *noise* and *quantization noise* (for details see [JGH99]). For a detailed analysis of *quantization noise* (in context of a TOF-camera system) that is of some importance in weak illumination environments we refer to [Fra07]. Except photon shot noise, all of these random errors can be significantly reduced or eliminated by respective signal processing or hardware related techniques (like cooling) [LS01]. Therefore, the influence of shot noise on the range measurement resolution shall be analyzed.

Shot noise is a fundamental property of the quantum nature of light and arises from statistical fluctuations in the number of photons emitted from a light source. The same is true for the generation process of electron-hole pairs, which is discrete as well. Shot noise is unavoidable and always present in imaging systems. In terms of signal-to-noise ratio, the best a detector can do is to approach the shot noise limit. The pseudo signal X produced by shot noise can be described by Poisson statistics for which applies $\mathrm{Var}(X) = \langle X \rangle =$ *rate of charge carrier generation.*

From the basic law of error propagation we know that the uncertainty in the phase calculation (2.11) is given by:

$$\mathrm{Var}(\varphi) = \sum_{n=0}^{N-1} \left[\left(\frac{\partial \varphi}{\partial I_n} \right)^2 \mathrm{Var}(I_n) \right]. \tag{2.28}$$

Assuming I_n (measured in units of electrons) to be a Poisson distributed random variable, $\mathrm{Var}(I_n) = I_n$ applies. Using the 4 sample algorithm (2.12) and (2.10c) we find the variance to be:

$$\mathrm{Var}(\varphi) = \sum_{n=0}^{3} \left[\left(\frac{\partial \varphi}{\partial I_n} \right)^2 I_n \right] = \frac{\pi^2}{4mT} \frac{G_0}{A^2} \sim \frac{\pi^2}{4} \frac{G_0}{A^2} \tag{2.29}$$

We dropped integration time mT (assumed to be constant) with the proportionality. For the amplitude and offset calculation we determine by analogical reasoning:

$$\mathrm{Var}(A) \sim \frac{\pi^2}{4} G_0 \qquad \mathrm{Var}(G_0) \sim \frac{1}{2} G_0$$

If we drop the assumption of an optimal demodulation contrast of $2/\pi$ and express equation (2.29) using modulation contrast and demodulation contrast and G_0 by $DC + B$, B being the background illumination and DC the DC-component of the modulated illumination, we find:

$$\mathrm{Var}(\varphi) \sim \frac{DC + B}{(C_{mod}\, C_{demod}\, DC)^2}$$

The additional noise sources, $1/f$-, reset- and thermal noise, may be summarized as dark noise and modeled by an additional number of electrons D that contribute exclusively to the constant background illumination as this noise does not correlate with the modulation. The standard deviation of the range measurement error is then given by

$$\sigma_\varphi \sim \frac{\sqrt{DC + B + D}}{C_{mod}\, C_{demod}\, DC} \tag{2.30}$$

Figure 2.6 shows qualitative how the uncertainty in the range measurements depends on various parameters. Only the denoted parameter was changed while the others were kept constant: mild background illumination and dark noise equivalent to 30000e$^-$ generated during exposure time, electrooptical signal of $2 \cdot 10^5$e$^-$, modulation contrast of 90% and demodulation contrast of 50%. For the range curve the inverse square law of irradiation was applied and 10^5e$^-$ were assumed to be integrated in a distance of 1m. These values are in magnitude those of a realistic setup (see [Lan00, chap. 4.2]). The range curve was cut at 50cm, because the pixel would go into saturation below this limit (and the linear model employed for demodulation would no longer be valid).

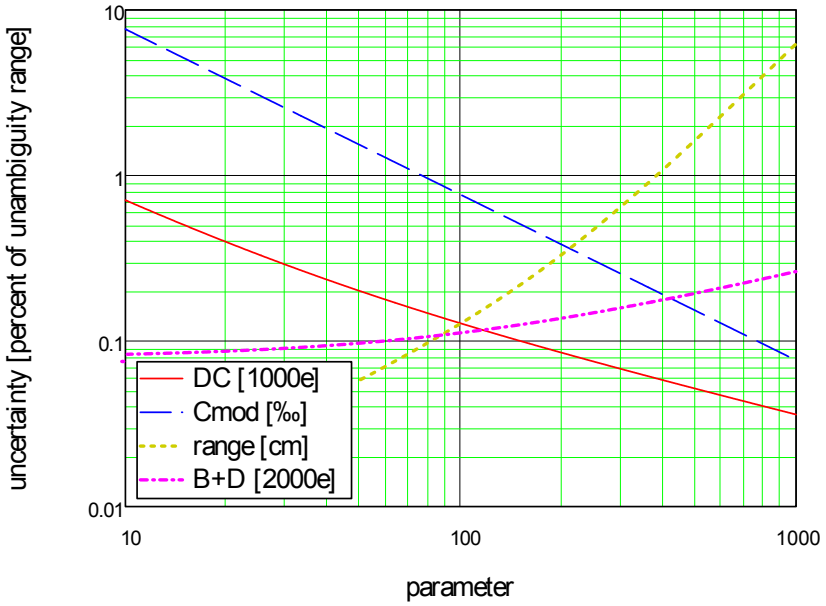


Figure 2.6: σ_φ given in percent of the unambiguity range in dependence of optical power (DC), modulation contrast, range and background illumination

---

We find that the range dependence has the most prominent impact on the measurement accuracy. Due to limited capacitance of a PMD-pixels it is not possible to achieve a decent range accuracy over the entire unambiguity range at constant exposure time (or optical power) and constant reflectivity. Multiple exposure times or adaptive illumination is needed to achieve a good accuracy.

Once image acquisition is completed, we can improve range measurement with respect to noise, by making assumptions about the smoothness of the range data in a local neighborhood; we may apply simple linear smoothing filters, which will however blur image features at edges. Using more advanced (nonlinear and/or robust) filters, we are able to preserve edges in range imagery both at intensity and range edges. Its important to notice in this context, that we can use equation (2.29) to determine an uncertainty (or confidence) measure from the measurements A and G_0 , from which appropriate filters can take advantage of and improve their denoising performance (see section 3.1.5 and section 3.2).

Chapter 3

Image Processing and Filters

3.1 Basics

3.1.1 Discretization and Sampling

Dealing with PMD-imagery and motion estimation involves handling of several kinds of discretization: We have discretization in space due to the sensor grid. Discretization in time, as we can only take a finite number of image-frames during a specific time-slice. And a discretization in the range of image-data, *i.e.* quantization, due to the quantization on sensor-level.

Discretization is closely related to the term *sampling*, which denotes the reduction of a continuous signal to a discrete signal. And a *sample* refers to a value or set of values at a point in time and/or space. For signal and image processing the *Nyquist–Shannon sampling theorem* is of utmost importance. It states that if a function $f(t)$ contains no frequencies higher than or equal to ν, then it is completely determined by giving its ordinates at a series of points t_n spaced $\frac{1}{2\nu}$ apart.

For the different kinds of discretization different models apply, or rather the continuous physical models, which are the basis for the interpretation of the data, need to be transformed into their discrete counterparts.

Derivatives and Gradient For example in context of motion estimation derivatives on signals are of major importance, because these are basic to compound operators like the gradient which is involved in estimating orientation (or equivalently velocity)

in an image sequence. Unfortunately derivatives are defined exactly for continuous functions only and not for discrete signals.

One might try to approximate the derivative-operator by a discrete counterpart *e.g.* a finite difference. Finite differences are successfully applied in *finite difference methods* for solving (partial or ordinary) differentials equations on a digital computer (on an analog computer things are treated quite different). Fornberg [For98] gives a very compact, analytic solution for calculating derivatives of any order, approximated to any level of accuracy on equispaced grids. In the context of finite difference methods, the consistency of an operators is of major importance, *i.e.* that the discrete approximation converges towards the continuous operator for vanishing difference h between grid points. Well known examples of these approximations are *forward*, *backward* and *central difference quotient*.

However, for image processing these operators, though applicable, are not first choice. The basic reason is that typically the grid of the data-samples is fixed, while it is adaptable with respect to the solution of a differential equation. So even though consistency of the operator is given it is not necessarily a good measure for the performance of the operator with respect to image processing.

For image processing other features are for various reasons of practical relevance. For instance the separability of an operator is of relevance w.r.t. its speed , *i.e.* necessary floating point operations. The *isotropy* of the gradient (or rather the viewpoint invariance of its result in the sense of a tensor) is important to an unbiased orientation estimate. Isotropy w.r.t. a vectorial quantity like the gradient has two aspects: isotropy in magnitude and in direction. For motion estimation the isotropy in direction is of major importance as the flow field can be calculated from the spatiotemporal orientation of structures in an image sequence.

Given a plane wave

$$w(\boldsymbol{k}) = A\,\exp(\imath(\boldsymbol{k}\boldsymbol{x}+\theta)) = A\,\exp\left(\imath\left(\begin{bmatrix} m\cos(\phi) \\ m\sin(\phi)\end{bmatrix}^{\mathrm{T}}\cdot\boldsymbol{x}+\theta\right)\right) =: w(m,\phi) \quad (3.1)$$

and the gradient $\boldsymbol{\nabla}_{\boldsymbol{x}} = [\partial_{x_1}\ \partial_{x_2}]^{\mathrm{T}}$ in two dimensions, the anisotropy in magnitude of the gradient w.r.t. the wavenumber $\boldsymbol{k}$ may be described by

$$\Delta(m) := |\boldsymbol{\nabla}_{\boldsymbol{x}} w(m,\phi)| - |\boldsymbol{\nabla}_{\boldsymbol{x}} w(m,0)| \quad (3.2)$$

and anisotropy in direction by

$$\Delta(\phi) := \arctan\left(\frac{\partial_{x_2} w(m,\phi)}{\partial_{x_1} w(m,\phi)}\right) - \phi \tag{3.3}$$

which are zero for the continuous gradient applied at coordinates of equal phase, *e.g.* $\boldsymbol{x} = 0$.

An approach, that allows a detailed analysis of an operator w.r.t. anisotropy and other features, is to interpolate a continuous signal from the discrete samples, do the continuous operation and sample the result at the grid points. As we will see, these tasks can be realized by means of digital filters without leaving the discrete domain. However, for the analysis of digital filters we need to change from spatial domain to frequency domain, or rather project the spatial data into the Hilbert space of plane waves (3.1), what corresponds to *Fourier transformation* that we will introduce in the following.

3.1.2 Fourier Transform

A function $g : \mathbb{R}^n \longmapsto \mathbb{C}$ is called *Fourier transformable* if the *Cauchy principal value*

$$\hat{g}(\boldsymbol{k}) \;:=\; \mathcal{F}(g(\boldsymbol{x})) \;:=\; \frac{1}{(2\pi)^n} \int_{-\infty}^{\infty} \exp(-\imath \boldsymbol{k}\boldsymbol{x}) g(\boldsymbol{x}) \; \mathrm{d}^n x \tag{3.4}$$

for all $\boldsymbol{k}$ exists. $\hat{g}(\boldsymbol{k})$ is called the (multidimensional, forward) *Fourier transform* (FT) of g (adapted from [MV06]). The inverse Fourier transform of $\hat{g} : \mathbb{R}^n \longmapsto \mathbb{C}$ is

$$\mathcal{F}^{-1}(\hat{g}(\boldsymbol{k})) \;:=\; \int_{-\infty}^{\infty} \exp(\imath \boldsymbol{k}\boldsymbol{x}) \hat{g}(\boldsymbol{k}) \; \mathrm{d}^n k \quad (\;=\; g(\boldsymbol{x})) \tag{3.5}$$

if the integral exists as Cauchy principal value. $g(\boldsymbol{x})$ and $\hat{g}(\boldsymbol{k})$ are called a *Fourier transform pair*, denoted abridged as $g(\boldsymbol{x}) \circ\!\!-\!\!\bullet \hat{g}(\boldsymbol{k})$. Due to the finite energy and range and the continuity of physical processes the Cauchy principal value always exists in the context of image processing.

There are several other common and equivalent definitions for the Fourier transform, which differ in to which kind of frequency domain the spatial domain is mapped and how the factor $(1/2\pi)^n$ is distributed among the (forward) Fourier transform and

its inverse. We have chosen the frequency domain to be the vectorial wavenumber $\boldsymbol{k} := {}^{2\pi}/_{\boldsymbol{\lambda}}$. One needs to be very careful about which definition was used, if looking up formulas in that context in textbooks, and how these shall be applied in one's own calculations, especially because various textbooks show inconsistencies with their own definitions throughout the text.

Features of the Fourier Transform Some important features of the (multidimensional) Fourier transform, that are used throughout the thesis are depict here via their respective Fourier transform pairs:

Linearity	$ag(\boldsymbol{x}) + bh(\boldsymbol{x})$	$\circ\!\!-\!\!\bullet$	$a\hat{g}(\boldsymbol{k}) + b\hat{h}(\boldsymbol{k})$	(3.6)
Separability	$\prod_{p=1}^{n} f(x_p)$	$\circ\!\!-\!\!\bullet$	$\prod_{p=1}^{n} \hat{f}(k_p)$, where $f : \mathbb{R} \mapsto \mathbb{C}$	(3.7)
Similarity	$g(s\,\boldsymbol{x})$	$\circ\!\!-\!\!\bullet$	$\hat{g}({}^{\boldsymbol{k}}/_{s})/\lvert s\rvert$	
	$g(\boldsymbol{A}\boldsymbol{x})$	$\circ\!\!-\!\!\bullet$	$\hat{g}(\boldsymbol{A}^{\mathrm{T}\,-1}\boldsymbol{k})/\lvert\boldsymbol{A}\rvert$	(3.8)
Rotation	$g(\boldsymbol{U}\boldsymbol{x})$	$\circ\!\!-\!\!\bullet$	$\hat{g}(\boldsymbol{U}\boldsymbol{k})$, $\boldsymbol{U}$ is unitary	(3.9)
Convolution	$(g{*}h)(\boldsymbol{x})$	$\circ\!\!-\!\!\bullet$	$(2\pi)^n(\hat{g}\cdot\hat{h})(\boldsymbol{k})$	(3.10)
Multiplication	$(g{\cdot}h)(\boldsymbol{x})$	$\circ\!\!-\!\!\bullet$	$(2\pi)^n(\hat{g}*\hat{h})(\boldsymbol{k})$	(3.11)
Translation	$g(\boldsymbol{x}-\boldsymbol{x}_0)$	$\circ\!\!-\!\!\bullet$	$\hat{g}(\boldsymbol{k})\exp(-\imath\boldsymbol{k}\boldsymbol{x}_0)$	
	$g(\boldsymbol{x})(\exp\imath\boldsymbol{k}_0\boldsymbol{x})$	$\circ\!\!-\!\!\bullet$	$\hat{g}(\boldsymbol{k}-\boldsymbol{k}_0)$	(3.12)
Derivatives	$\partial_{x_p} g(\boldsymbol{x})$	$\circ\!\!-\!\!\bullet$	$\imath k_p \hat{g}(\boldsymbol{k})$	(3.13)
δ-impulse	$\delta(\boldsymbol{x})$	$\circ\!\!-\!\!\bullet$	$({}^{1}/_{2\pi})^n$	(3.14)
δ-comb	$\sum_m \delta(x - m\Delta x)$	$\circ\!\!-\!\!\bullet$	$\frac{2\pi}{\Delta x}\sum_n \delta(k - n\frac{2\pi}{\Delta x})$	(3.15)
Gaussian	$\exp(-\frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}{2\,\sigma^2})$	$\circ\!\!-\!\!\bullet$	$\left(\frac{\sigma}{\sqrt{2\pi}}\right)^n \exp(-\frac{\sigma^2\boldsymbol{k}^{\mathrm{T}}\boldsymbol{k}}{2})$	(3.16)
Box	$\prod_{p=1}^{n} H(w - \lvert x_p\rvert)$	$\circ\!\!-\!\!\bullet$	$\prod_{p=1}^{n} \frac{\sin(w\,k_p)}{\pi\,k_p} = \prod_{p=1}^{n} \frac{w}{\pi}\operatorname{sinc}(\frac{w}{\pi}k_p)$ where $\operatorname{sinc}(x) := {}^{sin(\pi x)}/_{\pi x}$	(3.17)

The features given embed also those of the discrete Fourier transform (DFT), that is implemented on computers typically via a Fast Fourier Transform (FFT). The DFT

is the FT of a function with a finite extension and a finite bandwidth (FEF). Finite extension functions can always be extended to a periodic function by concatenating it to infinity with the range-values of its domain (we may call this process *periodization*).

According to the *Nyquist–Shannon sampling theorem* such a function may be represented without loss of information by a set of samples of this function, if these are taken as ideal samples (*i.e.* via convolution with a *Dirac delta function*) and the sampling frequency is bigger than two times the highest frequency in the function-signal (*i.e.* the onesided *baseband-bandwith* B): $freq_{samp} > 2B$. Vice versa a signal that is ideally sampled with a sampling distance Δx must not contain any frequency equal or above the Nyquist frequency k_{ny}, if the sampling shall neither loose information nor introduce artifacts (called *Moiré pattern* for 2D-imagery or in general *aliasing*):

$$k_{max} \overset{!}{<} k_{ny} = \frac{\pi}{\Delta x} = \pi \, freq_{samp} \; . \tag{3.18}$$

We denote the wavenumber which is scaled to the Nyquist frequency as $\tilde{k}$:

$$\tilde{k} := \frac{k}{k_{ny}} = \frac{k \, \Delta x}{\pi} = \frac{2\Delta x}{\lambda} \quad \Longleftrightarrow \quad k = \pm k_{ny} \quad \equiv \quad \tilde{k} = \pm 1 \; . \tag{3.19}$$

The DFT corresponds to FT applied on a FEF function that is periodizated (*i.e.* convolution with a delta-comb with a spacing as big as the domain of the function) and sampled (multiplication with a delta-comb of spacing smaller then half of the smallest wavelength of the periodizated signal). In Fourier domain this corresponds to a sampling of the frequency space (multiplication with delta-comb of spacing inverse to the extension of the function's domain) and periodization of it with a period of $freq_{samp}$, due to equations (3.10) and (3.15).

3.1.3 Interpolation

If we want to interpolate a sampled FEF function (*e.g.* an image), we may do this by reversing the effect of sampling in the Fourier domain, which is the periodization; by multiplication with a box-function of half width $w = \pi \, freq_{samp}$ we can achieve this. In the spatial domain this corresponds to a convolution with a scaled sinc function (3.17). While theoretically the sinc function is the ideal interpolation function, from a practical perspective it does not help too much. First of all the support of sinc is unbounded (*i.e.* it has no compact support) and the function decreases only linear

with its variable, which makes it a poor candidate to be used in a numerical convolution, even if small errors are acceptable. Furthermore it is not direction isotropic (or rotational-invariant), *e.g.* in a multidimensional image the interpolation result depends on the orientation of structures in the image.

Another candidate for interpolation are properly scaled Gaussian functions: What we need to do for a proper interpolation, is to set the signal samples in Fourier space to zero outside the signal's original baseband-bandwidth. Or in analogy to solid-state physics: All but the first Brillouin zone have to be zero. Additionally for a lossless reconstruction the signal must not be suppressed within the first Brillouin zone. A properly scaled Gaussian multiplied with the Fourier transform approximates those requirements sufficiently well, and corresponds to a convolution with a Gaussian of inverse width (3.16). Moreover the multidimensional Gaussian is the only function that is both separable and rotation-invariant.

Interpolation using Gaussian functions, while not ideal, still works well for digital imagery that complies to the sampling theorem, because real world image data typically has a low signal to noise ratio for wavenumbers near to the Nyquist frequency. So even though the Gaussian becomes approximately zero close to the Nyquist frequency, it tends to suppress more noise than signal. Furthermore the sampling of a digital camera is not ideal but influenced by the MTF (see section 3.1.4) of the optics involved and typically acts as a low pass filter which narrows the effective bandwidth of the signal additionally. Such the missing flank of the Gaussian (compared to the box-function) is less problematic.

Interpolation of a continuous function g_c from sampled data $g(\boldsymbol{x_n})$ on a grid $\boldsymbol{x_n}$ is realized via a discrete convolution with a continuous function $h(x)$:

$$g_c(\boldsymbol{x}) = \sum_{\boldsymbol{n}} g(\boldsymbol{x_n})\, h(\boldsymbol{x} - \boldsymbol{x_n}). \tag{3.20}$$

The interpolation function $h(x)$ needs to fulfill the interpolation condition:

$$h(\boldsymbol{x}) = \begin{cases} 1 & \text{for } \boldsymbol{x} = 0 \\ 0 & \text{for } \boldsymbol{x} = \boldsymbol{x_m} - \boldsymbol{x_n} \quad \text{where } m \neq n \end{cases}. \tag{3.21}$$

Applying a partial derivative along the grid dimensions x_p yields due to linearity of the derivative:

$$\begin{aligned} g'_c(\boldsymbol{x}) :=\ & \partial_{x_p} g_c(\boldsymbol{x}) = \partial_{x_p} \sum_{\boldsymbol{n}} g(\boldsymbol{x_n})\, h(\boldsymbol{x} - \boldsymbol{x_n}) \\ =\ & \sum_{\boldsymbol{n}} g(\boldsymbol{x_n})\, \partial_{x_p} h(\boldsymbol{x} - \boldsymbol{x_n}) = \sum_{\boldsymbol{n}} g(\boldsymbol{x_n})\, h'(\boldsymbol{x} - \boldsymbol{x_n}) \end{aligned}$$

Resampling on the original grid positions $\boldsymbol{x_m}$ equals:

$$g'(\boldsymbol{x_m}) := g'_c(\boldsymbol{x_m}) = \sum_{\boldsymbol{n}} g(\boldsymbol{x_n})\, h'(\boldsymbol{x_m} - \boldsymbol{x_n}) \tag{3.22}$$

So for calculating the derivatives at the original grid points we only need to do a discrete convolution of the signal with a sampled version of the (partial) derivative of the interpolation function. If the interpolation function is chosen to be a Gaussian, $h'(\boldsymbol{x_m} - \boldsymbol{x_n})$ may be approximated by a filter mask $\boldsymbol{H}$ with a finite number of filter coefficients. It is only an approximation, because h' has no compact support, and we need to truncate it at its ends. But as h' decreases rapidly with increasing $|\boldsymbol{x_m} - \boldsymbol{x_n}|$, the error introduced is only a small one. The filter or convolution mask $\boldsymbol{H}$ is completely independent of the signal and the position that it is applied on. The operation of applying such a mask via convolution belongs to the class of *linear shift-invariant* filters, that we introduce more formally in the following section. For the sake of simplicity we will stick to a formulation for 2D scalar images; anyhow things may be generalized to multidimensional, spatiotemporal (image-)data or tensor valued data see [Big06, chap. 3.6].

3.1.4 Convolution, Point Spread Function and Transfer Function

Filtering in the context of image processing is realized for the class of linear shift-invariant (LSI) filters as convolution. Convolution of a 2-dimensional image $\boldsymbol{G}$ of size $M \times N$ with a square convolution mask $\boldsymbol{H}$ of $(2R+1)^2$ elements h_{mn} is given by

$$g'_{mn} = \sum_{m'=-R}^{R} \sum_{n'=-R}^{R} h_{m'n'} g_{m-m',n-n'} =: [\boldsymbol{H} * \boldsymbol{G}]_{mn}$$

The filter is by definition *linear* and *shift invariant*, as it has the properties of a linear operator and does not depend on the position (m, n) at which it is applied.

The *point spread function* (PSF) is defined as the filter-response on a point image $\boldsymbol{P}$ ($p_{m,n} = \{1 \text{ for } m = n = 0, 0 \text{ otherwise}\}$) and is identical to the convolution mask $\boldsymbol{H}$

$$\text{PSF}_{mn} := \sum_{m'=-R}^{R} \sum_{n'=-R}^{R} h_{m'n'} p_{m-m',n-n'} = h_{mn} = [\boldsymbol{H} * \boldsymbol{P}]_{mn} \tag{3.23}$$

It fully describes a LSI filter, as its response to an arbitrary image is just a linear combination of shifted PSFs, with the coefficients being the pixel values of the image.

The *optical transfer function* (OTF) is defined as the Fourier transform of the PSF.* It is the wavelength dependent multiplication factor of a LSI filter in the Fourier domain. This is easy to see, regarding the convolution theorem (3.10) that states, that convolution in the spatial domain corresponds to a multiplication in the Fourier domain (and vice versa). The discrete delta peak image $\boldsymbol{P}$ of equation (3.23) transforms due to equation (3.14) to a constant value for all Fourier domain pixels. Thus, the filter is, not surprisingly, described completely by $\hat{\boldsymbol{H}}$. The magnitude of the optical transfer function is called the *modulation transfer function* (MTF) and describes the attenuation of the sinusoidal waveforms as a function of their spatial frequency.

$$\text{OTF} := \mathcal{F}(\text{PSF}) = \mathcal{F}(\boldsymbol{H} * \boldsymbol{P}) = \hat{\boldsymbol{H}} \cdot \hat{\boldsymbol{P}} = \hat{\boldsymbol{H}} \cdot const. \tag{3.24}$$

$$\text{MTF}_{mn} := |\mathcal{F}(\text{PSF})_{mn}| = |\hat{\boldsymbol{H}}_{mn}| \cdot const. \tag{3.25}$$

Eventually, all a LSI filter does, is to attenuate sinusoidal waves and to translate their positions (where the translation vector is encoded in the argument of the respective complex Fourier transform entry $\hat{\boldsymbol{H}}_{mn}$). The (continuous) transfer function $\hat{h}$ of a LSI-filter $\boldsymbol{H}$ on an orthogonal grid is

$$\hat{h}(\tilde{\boldsymbol{k}}) = \sum_{m=-R}^{R} \sum_{n=-R}^{R} h_{mn} \exp(-\pi \imath \begin{bmatrix} n \\ m \end{bmatrix}^{\mathrm{T}} \cdot \tilde{\boldsymbol{k}}) \tag{3.26}$$

From Euler's formula $\exp(\imath x) = \cos(x) + \imath \sin(x)$ we derive for filter masks of even symmetry

$$\hat{h}(\tilde{k}) = h_0 + \sum_{n=1}^{R} 2\, h_n \cos(\pi\, n\, \tilde{k}), \tag{3.27}$$

and for odd filter masks

$$\hat{h}(\tilde{k}) = \imath \sum_{n=1}^{R} 2\, h_n \sin(\pi\, n\, \tilde{k}). \tag{3.28}$$

*The term *transfer function* has a more general definition, but is used sometimes synonymously with OTF

Due to the separability of the Fourier transform (3.7), the transfer function of a multidimensional, separable filter - composed by convolving one dimensional filters (of specific symmetry) - is just the product of the individual filters.

Filter Design and Optimization Now that we introduced LSI filtering by means of convolution we come back to discrete operators. We have shown on page 37 that interpolating the discrete data, taking the derivative and resampling on the original grid can be done approximatively by applying a single discrete filter by means of convolution. If we do as proposed we arrive at the filter family of so called *Derivatives of Gaussian*. The anisotropy of a gradient operator composed by these filters is w.r.t. its magnitude (3.2) definitively lower than those of the central difference quotient, but identical w.r.t. the direction estimation (3.3). A very well known derivative filter is the *Sobel operator*. Compared to the previously mentioned filters its anisotropy is more than a factor 2 lower for the angle estimate and also smaller w.r.t. the magnitude of the gradient for details see [JH00, chap. 9.7]. This is achieved by introducing an asymmetry in the width of the interpolating Gaussians in direction of the derivative and normal to it. But the maximum angle anisotropy is still around 20° and is independent of the filter-size (which improves magnitude anisotropy only).

To further reduce anisotropy one can treat the filter design as an *optimization problem*. This means that we look for a filter that differs from the ideal (continuous) filter as less as possible under given constraints, arising from the discrete and finite extension of the applicable filter mask. The measure for the difference between the ideal and optimized filter and how the ideal filter actually should look like is based on the problem and its specific requirements. *E.g.* a discrete derivative filter can due to its finite extension never have a transfer function, that is both *ideal* w.r.t. equation (3.13) such that $\hat{h}(\tilde{k}) = \imath\tilde{k}$ for $\tilde{k}$ within]-1,1[and zero outside, as a discontinuity in Fourier space would require an infinite extension of the filter. Therefore, its a matter of design in which frequency band the so called *reference function* should approximate the ideal best, and which of the desired features (like isotropy) may be violated at which cost, within the optimization. A suitable optimization strategy then returns the filter coefficients, minimizing the cost or error between the *ansatz function* (basically equation (3.26)) and the reference function, in compliance with the given constraints. For details we refer to [JSK99] and in closing would like to point out that the derivative filters used in context of motion estimation for this the-

sis were optimized w.r.t. a maximum precision in orientation estimation as described by Scharr [Sch00].

3.1.5 Normalized Averaging

The PMD-data we are dealing with is affected by errors, statistical and systematic ones. Here we show a simple method to improve the range data with respect to the statistical errors.

As we have seen in section 2.2.2.2 the amplitude and offset of the PMD signal gives us a measure for the reliability of the range measurement. With equation (2.29) we find the variance of the range signal as proportional to G_0/A^2. However, most of the PMD-camera models we know do not give direct access to the offset G_0 in their standard configuration. The *PMD19k* for example returns besides range R and amplitude A also a third channel denoted as *intensity* I. But this *intensity* is not the DC-offset of the electrooptical signal but the amplitude signal weighted by the distance in some (unknown) manner.[†]

Only if we have access to all raw channels, we might calculate G_0 by equation (2.12). If this is not possible one might approximate G_0 as proportional to A, if we assume no background illumination. Then the variance in the range measurement is approximated by $\mathrm{var}(R) \sim A^{-1}$.

If we want to denoise the data correctly by averaging over a specific neighborhood, we know from elementary statistics that appropriate averaging requires the weighting of each data value with the inverse of the variance, *i.e.* using the upper approximation we just need to multiply by A.

As it is well known that box filters do not have very good properties from a signal processing point of view (due to their infinite and slowly decreasing transfer function), the neighborhood itself needs to be weighted too. So we need to incorporate another set of weights in the averaging procedure by using a filter such as a Binomial. Both weightings can be achieved with a technique that is known as normalized averaging [GK95].

[†]The relative error of I is even higher than that of A and R: suppose $I = AR^b$, then error propagation leads to $\sigma_I/I = \sqrt{(\sigma_A/\bar{A})^2 + (b\,\sigma_R/\bar{R})^2}$.

Normalized averaging is a special case of a more general filtering technique called normalized convolution that is described in detail by [KW93; Wes94; Far03]. The employed filter (*e.g.* a Binomial) is called the *applicability* $\boldsymbol{B}$. If the measurement data are denoted with $\boldsymbol{R}$ and the weighting image with $\boldsymbol{W}$ (*e.g.* the amplitude image $\boldsymbol{A}$) normalized averaging reads:

$$\boldsymbol{R}' = \frac{\boldsymbol{B} * (\boldsymbol{W} \cdot \boldsymbol{R})}{\boldsymbol{B} * \boldsymbol{W}} \ . \tag{3.29}$$

The weighting image is not necessarily associated with an error. It can be used to exclude or amplify pixels with certain features. In this way, normalized averaging becomes a versatile operator, that was used for various tasks in the context of this thesis.

However, it should be noticed that applying normalized averaging as described, leads to a bias toward smaller range values at depth-edges, as the confidence measure (or weighting image) is correlated with the (physical) quantity to denoise, *i.e.* the amplitude decreases with increasing depth: In the neighborhood of a depth edge, surface patches of the same reflectivity near to the camera (denoted in the following as *near surfaces*) will be weighted stronger than those far away. This leads to an anisotropic, biased blurring of image features, such that the near surfaces tend to grow while those away shrink. So normalized averaging should not be applied at surface edges.

Band Enlarging Operators Normalized averaging is a potentially band enlarging operation, because it involves multiplication of two images $\boldsymbol{W} \cdot \boldsymbol{R}$, which corresponds to a convolution in the wavenumber domain (3.11). If the sum of the bandwidths of $\hat{\boldsymbol{W}}$ and $\hat{\boldsymbol{R}}$ is larger than $\tilde{k} = 1$ in any dimension, aliasing occurs. Thus, it is important to adapt the bandwidth of the images w.r.t. the Nyquist wavenumber *before* multiplying the images, either by upsampling the images, which is a lossless but somewhat expensive operation (w.r.t. processing time and memory consumption) or by pre-smoothing with *e.g.* a binomial filter, which is fast but potentially lossy. The same rules apply for operations where rotations (3.9) are involved, as a FT-image that is rotated exceeds the Nyquist borders - the corner areas lie outside the first Brillouin zone - and if the corresponding Fourier coefficients are not zero, aliasing will occur.

Köthe [Köt03] points out, that the influence of band enlarging operators was frequently neglected in computer vision literature in conjunction with more complex

operators like *e.g.* the Canny edge detector and the structure tensor. With modern high resolution image sensors of several million pixels resolution however, the aspect is of less importance, because typically the camera's optics act as a low-pass filter with respect to the sensors resolution, especially in the field of consumer market products. Not so however for current PMD-sensors, as they have a low sensor resolution compared to the resolution of the optics yet.

3.2 Edge Preserving Smoothing

The methods to denoise range imagery discussed so far, all lack the ability to denoise or smooth the data without blurring image features like edges or corners. This is due to the fact that the models they are based on assume a planar neighborhood or at least one with a very specific symmetry, and thus are violated around the mentioned features. There are basically two ways to handle this problem. One is to extend the model to explain the data better in a specific neighborhood. The other is to improve the estimate on the model parameters, by means of a robust estimator, *i.e.* one that gives a correct estimate in the presence of a minority of data points that do not fit to the model, so called outliers. Both approaches may be combined and transitions are smooth. Applying robust methods of statistics to the field of computer vision is not trivial. Taking for example the simple case of a corner of a cube seen from atop with a range camera: In the vicinity of the corner there are 3 planes, thus the *majority* of the pixels in a neighborhood of any pixel near to the corner, will violate a single planar model, and therefore cannot be treated as classical outlier w.r.t. this model.

3.2.1 Robust Estimators

Robust estimation is concerned with the accurate estimation of model parameters in the presence of data that violates the model for which the parameters shall be determined and/or the assumptions about which errors the measurements show (*i.e.* the employed noise model): the data may contain classical, gross outliers that are not consistent with an assumed data model exposed to *e.g.* Gaussian noise. For a low-level model of PMD-data this might stem from specular reflections of the modulated illumination, leading to saturation of the capacities and in turn to a completely arbitrary depth measurement. Defective pixels or interreflection of light from multiple surfaces are other sources of outliers. Another class of outliers consists

of pixels belonging to a minority of the data, a population that is compatible with a different[‡], potentially unknown data model; *e.g.* in the case of a planar surface model every step- or roof-edge of an object or partial occlusion will give rise to these kind of outliers. With respect to image processing the same pixel can be either an outlier or an inlier, depending on the position of the model to be estimated.

As each pixel measurement is subject to small-scale random variations, the parameter estimation problem is heavily overconstrained (for both low-level and high-level models), which suggests that a maximum likelihood estimation technique should be employed to solve the problem. Under the assumption of normal (Gaussian) distributed, additive noise, least squares estimation is a maximum likelihood estimator [LP02, chap. 20.2.6], *i.e.* the probability for the observed measurements is maximal for the estimated parameters.

Let y_i be the measurements at the independent (or *control* or *explanatory*) variables $\boldsymbol{x_i}$, *e.g.* the sensor grid coordinates, of the model m for which the (vector of) parameters $\boldsymbol{p}$ are to be estimated, *e.g.* the surface normal and intercept of a planar surface model. Then the ordinary least squares (OLS) estimate $\hat{\boldsymbol{p}}$ is given as

$$\hat{\boldsymbol{p}} = \underset{\boldsymbol{p}}{\operatorname{argmin}} \sum_i \left(\frac{y_i - m(\boldsymbol{x_i}, \boldsymbol{p})}{\sigma_i} \right)^2 \tag{3.30}$$

$$= \underset{\boldsymbol{p}}{\operatorname{argmin}} \sum_i \left(\frac{r(y_i, \boldsymbol{x_i}, \boldsymbol{p})}{\sigma_i} \right)^2 , \tag{3.31}$$

leading to the more general formulation known as *M-estimator*:

$$\hat{\boldsymbol{p}} = \underset{\boldsymbol{p}}{\operatorname{argmin}} \sum_i \rho\left(\frac{r_{i,\boldsymbol{p}}}{\sigma_i} \right) . \tag{3.32}$$

The expression to be minimized is called the *objective function*, and $\rho(r)$ is known as the *loss function* (or *error norm*), which is $\rho(r) = r^2$ for the least squares estimate. The *residual function* r describes the (error) distance between a measurement and the model determined by $\boldsymbol{p}$ (and $\boldsymbol{x_i}$). The residuals need to be normalized to the scale (noise level) σ_i associated with the measurements y_i; in the simplest case the measurements are i.i.d. thus $\sigma_i = \hat{\sigma} \ \forall \ i$, which in turn can be neglected for least squares estimation.

[‡]different means, either an instance of the same model with different parameters or a completely different model

An advantage of the least squares estimation problem (3.30) is, that it can be solved efficiently for models m, which are linear in their parameters $\boldsymbol{p}$ (but possibly nonlinear w.r.t. the independent variables), by means of the LSI-filters introduced in the previous section (for details see [JHG99]). However, most real world problems cannot be described sufficiently by a single model under a Gaussian noise assumption, and because the least squares loss function grows unlimitedly with increasing $|r|$, a single outlier can corrupt the estimation seriously; this is why the *breakdown point* of least squares is 0. The breakdown point is the minimum fraction of outlying data that can cause an estimate to diverge arbitrarily far from the sought value. The theoretical maximum breakdown point of any "general purpose" estimator is 0.5, because with more than 50% outliers, these can be arranged in a way that, in terms of regression analysis, a fit through them will minimize the objective function.

The breakdown point of an estimator does not say anything about its *efficiency*, which is defined as the minimum possible variance for an (unbiased) estimator divided by its actual variance, with the minimum possible variance being determined by a target distribution (*e.g.* a Gaussian one). Typically robust estimators with a high breakdown point tend to have a low efficiency, thus the estimates have a high variance and require a big number of measurements to gain a reasonable (statistical) precision.

The least squares estimator belongs to the class of *M-estimators* ("M" for "maximum likelihood type" [Hub81, page 43]). These are of the form (3.32), with $\rho(r)$ being a function of even symmetry ($\rho(r) = \rho(-r)$) with an unique minimum at zero and monotonically increasing for $r > 0$. The robustness against outliers is achieved by a loss function that grows subquadratically. This becomes clearer if we look at the solution of equation (3.32) which is determined by the root of its derivative w.r.t. $\boldsymbol{p}$. If $\boldsymbol{\nabla_p}$ denotes the vector of partial derivative operators (∂/∂_{p_n}) of the $n = 1 \dots N$ parameters of m, then we find a system of N equations for the vanishing derivatives at the minimum of the objective function:

$$\boldsymbol{\nabla_p} \sum_i \rho\left(\frac{r_{i,\boldsymbol{p}}}{\sigma_i}\right) = \sum_i \frac{1}{\sigma_i} \psi\left(\frac{r_{i,\boldsymbol{p}}}{\sigma_i}\right) \boldsymbol{\nabla_p} r_{i,\boldsymbol{p}} \stackrel{!}{=} 0, \text{ where } \psi(r) = \partial_r(\rho(r)) \tag{3.33}$$

For a model m linear in its parameters $\boldsymbol{p}$, $\boldsymbol{\nabla_p} r_{i,\boldsymbol{p}}$ simplifies to $\boldsymbol{x_i}$:

$$\sum_i \psi\left(\frac{r_{i,\boldsymbol{p}}}{\sigma_i}\right) \frac{\boldsymbol{x_i}}{\sigma_i} = 0 \tag{3.34}$$

and for the simplest model $m = 1 \cdot p$ it is

$$\sum_i \psi\left(\frac{r_{i,\boldsymbol{p}}}{\sigma_i}\right) \frac{1}{\sigma_i} = 0. \tag{3.35}$$

If the model m is not linear in $\boldsymbol{x}$ then $\boldsymbol{x_i}$ in equation (3.34) may denote a vectorial function dependent on the explanatory variables only. The derivative of the loss function is known as the *influence function* ψ. The name is reasonable as it is plain to see from system (3.34) that ψ directly determines the influence of a single residual on each constraint equation to become zero. For least squares the influence function is identical to the normalized residual and thus, its absolute value can become arbitrarily large. Robust estimators use an influence function that is bounded above and below and which may become zero for large residuals. Influence functions tending to zero most quickly (known as *hard redescenders*) permit the most aggressive rejection of outliers. This feature is of major importance if the outliers have small residuals in the range of 4 to 10 σ [Ste99]. Redescending influence functions however make $\sum_i \rho(r_{i,\boldsymbol{p}}/\sigma_i)$ *nonconvex*, such that solvers for equation (3.33) may converge to local minima, if the initial guess is not close to the optimum.

Iteratively reweighted least squares (IRLS) is such a solver, which deduces from equation (3.33), where ψ is substituted by $w(r)\,r$, with $w(r) = {}^{\psi(r)}/_{r}$ known as the *weight function*:

$$\sum_i \frac{1}{\sigma_i^2} w\left(\frac{r_{i,\boldsymbol{p}}}{\sigma_i}\right) r_{i,\boldsymbol{p}}\, \boldsymbol{\nabla_p} r_{i,\boldsymbol{p}} = 0 \tag{3.36}$$

This can be iteratively solved by means of common weighted least squares solvers (*e.g.* SVD or *Gaussian elimination* for m linear in $\boldsymbol{p}$ and *Gauss–Newton* or *Levenberg-Marquardt* for a non-linear model), if for each iteration the weights $w(r)$ are calculated for the current guess of $\boldsymbol{p}$ and then fixed for the least squares solver; a least squares solver is applicable because the term $r_{i,\boldsymbol{p}}\, \boldsymbol{\nabla_p} r_{i,\boldsymbol{p}}$ in (3.36) is just the derivative of the least squares problem (3.31), while the other terms are kept constant.

Black and Rangarajan [BR96] give a survey of the various, in statistics and computer-vision literature proposed influence/loss functions in the light of related *outliers processes*; one example of a redescending influence function is the Leclerc function, depicted in Figure 3.1:

$$\begin{aligned} \rho_\eta(r) &:= 1 - \exp(-\frac{r^2}{\eta^2}), \\ \psi_\eta(r) &:= \partial_r \rho_\eta(r) = \frac{2\,r}{\eta^2} \exp(-\frac{r^2}{\eta^2}) \quad \text{and} \quad w_\eta(r) := \frac{\psi_\eta(r)}{r} = \frac{2}{\eta^2} \exp(-\frac{r^2}{\eta^2}) \end{aligned} \tag{3.37}$$

A second look at equation (3.34) tells us that standard M-estimators still have a breakdown point of zero, because an erroneous measurement at a point $\boldsymbol{x_i}$, which is

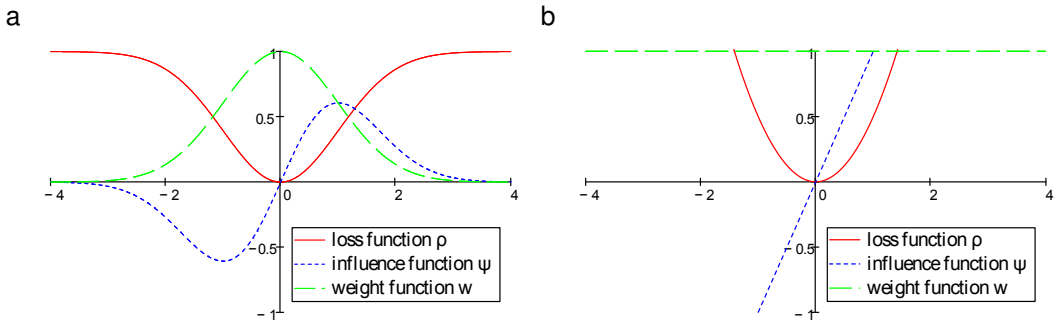


Figure 3.1: Robust and nonrobust M-estimators: a The robust Leclerc functions for $\eta^2 = 2$ and b the respective functions for least squares ($\psi(r) := r$)

far away from the bulk of the data may still corrupt the whole measurement. An alternative to M-estimators is the *least median of squares* (LMS) estimator, which has the maximum possible breakdown point of 0.5; compared to least squares the objective function is not the sum, but the median of the squared residuals:

$$\hat{\boldsymbol{p}} = \underset{\boldsymbol{p}}{\operatorname{argmin}} \underset{i}{\operatorname{median}} \left(\frac{r_{i,\boldsymbol{p}}}{\sigma_i} \right)^2 \tag{3.38}$$

For a simple linear regression model the LMS solution corresponds to the "narrowest strip covering half of the observations" [RL87]. LMS buys its excellent robustness against outliers at the cost of a less efficient (*random sampling*) search technique, because the median is not differentiable and thus *gradient descent* or *Newton's method* are not applicable; moreover, LMS has an abnormally slow convergence rate [RL87]. A robust estimator between OLS and LMS is *least trimmed squares* (LTS) which minimizes like OLS the sum of squared residuals, but excludes (at most) 50% of the residuals of larger magnitude from summation, which leads to an improved convergence rate while maintaining a high breakdown point.

Figure 3.2 illustrates some of the properties of robust estimators for a simple linear model (a straight line with unknown slope and intercept). While the LMS estimate finds the majority population model independent of the initial guess, the Leclerc M-estimator finds local minima which might belong to the majority population (the increasing straight) or the minority (the decreasing straight) or are completely wrong. The LMS estimate tends to be worse than the M-estimate, if both succeed and use the same initial guesses and the same convergence tolerance (the limiting difference in the objective function of two succeeding guesses to stop the minimization), indicating that LMS has a lower convergence rate. All optimizations were done with

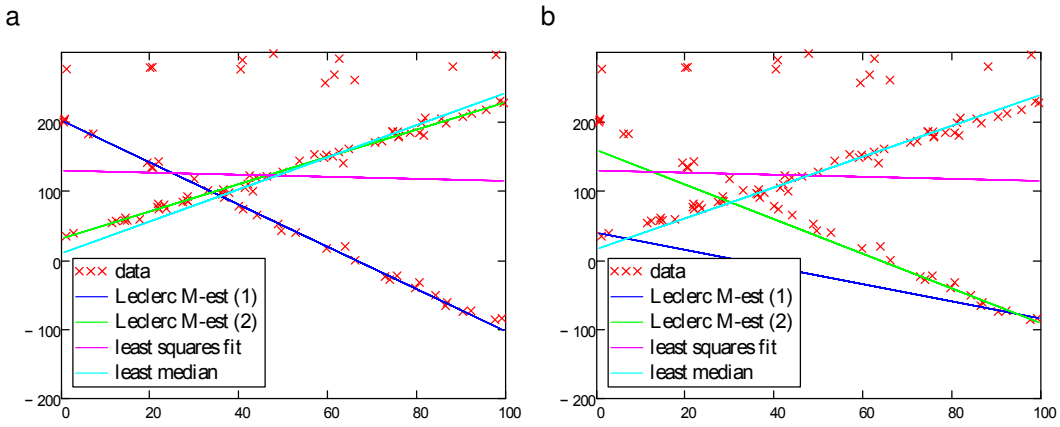


Figure 3.2: Illustration of sensitivity of M-estimators to local minima and slow but robust convergence of the LMS estimator. a two populations with fractions of 56% and 33% and 11% gross outliers; M-est (1) and (2) just differ in the inital guess. b same as a but with different initial guesses.

---

a nonlinear conjugate gradient solver. η^2 was chosen to be 2 for the Leclerc function and the residuals were scaled to the noise level ($\sigma = 5$) of the Gaussian i.i.d. model populations. The gross outliers stem from an uniform distribution in the range [250,300].

Figure 3.3 illustrates a problem more typical for image processing, in the context of robust estimators: a step edge. The model is the same as for figure 3.2 but the data contains no gross outliers and the two populations (constant lines with different offset) hardly overlap. Again Leclerc finds local minima and LMS gives a worse estimate. Furthermore, we see that the robust estimators break down if the residuals of the outliers w.r.t. the larger population only have a magnitude of some σ (the step of the edge for figure b is only 4σ).

3.2.2 Bilateral and Diffusion Filtering

Bilateral and diffusion filtering are very popular image processing methods for the task of denoising image data. Black et al. [Bla+98] show that anisotropic diffusion as introduced by Perona and Malik [PM90] may be regarded as a robust estimator. Durand and Dorsey [DD02] point out the that *bilateral filtering* as introduced by Tomasi and Manduchi [TM98] and what they call *0-order anisotropic diffusion*

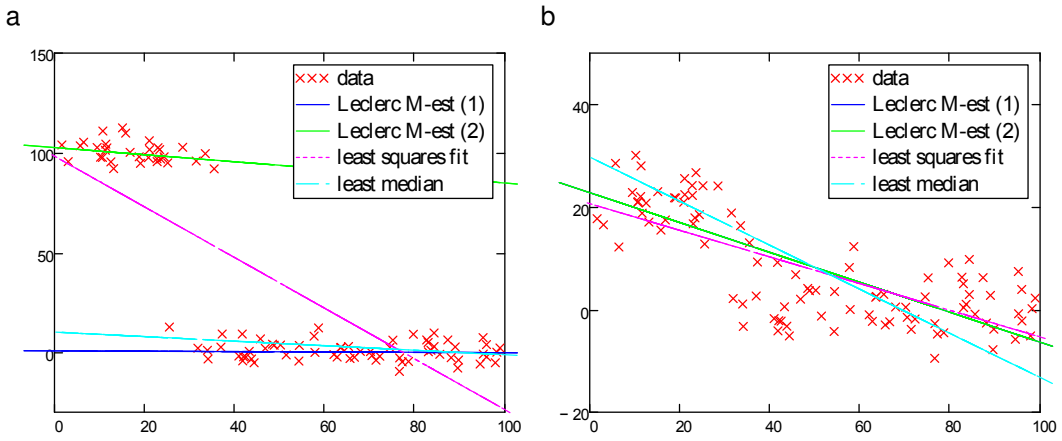


Figure 3.3: Illustration of breakdown of robust estimators with decreasing difference in magnitude between residuals of outliers and model samples, at a step edge. a the outliers (minority population) have a distance of more than 20σ from the majority; b the distance is only 4σ and all estimators fail, leading to a "bridging" estimate between the two population models.

(while *inhomogeneous diffusion* would be more appropriate) belong to the same family of robust estimators, the major difference being, that inhomogeneous diffusion filtering is *energy preserving*, while bilateral filtering is not (due to an asymmetric normalization term w.r.t. single pixels); energy preserving for a gray value image means that the arithmetical mean of its pixel's gray values does not change due to the applied filter.

Diffusion filtering is motivated by a physical observation expressed by *Fick's law*

$$\boldsymbol{j} = -\boldsymbol{D}\,\boldsymbol{\nabla} u, \tag{3.39}$$

which states, that a concentration (or temperature) gradient $\boldsymbol{\nabla} u$ causes a flux $\boldsymbol{j}$ that tries to compensate the gradient, in a way that is determined by the *diffusion tensor* $\boldsymbol{D}$, which is a positive definite, symmetric matrix. If $\boldsymbol{j}$ and $\boldsymbol{\nabla} u$ are parallel we speak of *isotropic diffusion*. Then $\boldsymbol{D}$ degenerates to D, the diffusivity. If D depends on (local) features of the field u and therefore is not constant, we are speaking of *inhomogeneous (but isotropic) diffusion*. Only if $\boldsymbol{j}$ is in general not parallel to the gradient this shall be called *anisotropic diffusion*.

For a closed system, mass (or heat) do not vanish, *i.e.* $\frac{\mathrm{d}u(x,y,t)}{\mathrm{d}t} = 0$. Applying the chain rule and identifying $\boldsymbol{j}$ as $u \cdot \left[\begin{smallmatrix}\partial_t x \\ \partial_t y\end{smallmatrix}\right]$ gives us the continuity equation

$$\partial_t u + \boldsymbol{\nabla} \boldsymbol{j} = 0. \tag{3.40}$$

Substituting $\boldsymbol{j}$ from Fick's law (3.39) yields the *diffusion equation*

$$\partial_t u = \boldsymbol{\nabla}(\boldsymbol{D}\, \boldsymbol{\nabla} u). \tag{3.41}$$

For image processing the local concentration may be replaced by *e.g.* the gray value of an image pixel, implying a discretization of the diffusion equation in space. Using a constant diffusivity is only appropriate, if we assume a constant gray value, such that the inhomogeneity introduced by (Gaussian) noise in the data is distributed and therefore leveled out by the homogeneous diffusion. At an edge in an image this assumption is clearly not fulfilled and homogeneous diffusion introduces errors (*e.g.* a blurring of the gray value edge). If image structures shall not become corrupted, the diffusion tensor needs to depend on the local structure in the evolving image; if so, the time dependence leads to a feedback, which indicates the nonlinearity of such a diffusion filter. Discretization in time and approximation of the derivatives by finite differences leads to an iterative solver. Perona and Malik [PM90] proposed a discretization for an inhomogeneous diffusion (which they called, somewhat sloppy, *anisotropic*):

$$I_s^{t+1} = I_s^t + \frac{\lambda}{|\mathfrak{n}|} \sum_{p \in \mathfrak{n}} w(\nabla I_{s,p}) \nabla I_{s,p} \approx I_s^t + \frac{\lambda}{|\mathfrak{n}|} \sum_{p \in \mathfrak{n}} w(I_p - I_s^t)(I_p - I_s^t), \tag{3.42}$$

where $w(x)$ was proposed to be $\exp(-x^2/\sigma^2)$. I_s^t denotes the (gray) value of a sampled image pixel s at time step (or rather iteration) t and $\mathfrak{n}$ a neighborhood of $|\mathfrak{n}|$ pixels p around s. $\nabla I_{s,p}$ indicates the directional derivative of I at s in the direction of p. If we look back at the definitions of the M-estimator equation (3.32) we may identify the model m as that of a constant gray value $p = I_s$, the measurements y_i as the pixel values I_p, and $w(x)$ as the Leclerc weight function (3.37). Comparing equation (3.42) with (3.36) while remembering that $\boldsymbol{\nabla_p} r_{i,\boldsymbol{p}} = 1$ for $m = p$, we find that equation (3.42) is just the gradient descent solution of (3.36) (*i.e.* IRLS).

Thus, inhomogeneous isotropic diffusion is a robust M-estimator for the very simple model of a constant neighborhood. If one extends equation (3.42) for a weighting of the addends by their distance (*e.g.* $w(p - s)$) this corresponds to a *generalized M-estimator* (GM). While the solution of the homogeneous isotropic diffusion equation

converges to an image of a single constant value (where the iteration steps are well approximated by Gaussian or binomial filtering), inhomogeneous diffusion converges to segments of constant value if hard redescenders are used as influence functions (the number and size of the segments depends on the noise level and structure of the image as well as the chosen influence function). Anisotropic diffusion allows to smooth along edges but not perpendicular to these, achieving edge-enhancing smoothing. A thorough discussion of (anisotropic) diffusion filtering, relations to *curvature-preserving PDEs* and their application is given in [Tsc02; Tsc06].

A similar reasoning as above is possible for bilateral filtering, which is motivated by introducing a weighting of the addends not only w.r.t. their spatial distance (as Gaussian filtering does) but also w.r.t. their distance in range to the pixel s:

$$I_s := \frac{1}{norm(s)} \sum_{p \in \mathfrak{n}} w_s(p-s)\, w_r(I_p - I_s)\, I_p \tag{3.43}$$

with the normalization term $norm(s) := \sum_{p \in \mathfrak{n}} w_s(p-s)\, w_r(I_p - I_s)$.

For details regarding the relation to robust estimators and diffusion filtering we refer to [DD02] and just want to annotate that the formal similarity to equation (3.42) already suggests their close relation and that the performance of the methods heavily depend on the chosen weight- or respective influence function. Jones, Durand, and Desbrun [JDD03] developed an interesting extension of bilateral filtering to (3D) surface meshes, that can be used to estimate the position of mesh vertices in a robust manner. The extension introduces the concept of predictors, which incorporate shape information in the filtering process, by means of normals on the (non-robustly) smoothed surface.

We realized an optional bilateral filtering, dependent on the range information, for the robust regularization of the structure tensor used in range flow estimation. For the task of denoising single PMD-frames however, we employed another robust estimation technique, an extended version of *channel smoothing*; it exhibits the advantage of being computationally very efficient compared to bilateral and (even more) anisotropic diffusion filtering and is more relaxed about its exact parametrization.

3.3 Two State Channel Smoothing

Channel smoothing, or more precisely w.r.t. this work *B-spline channel smoothing* is a technique introduced by Forssén, Granlund, and Wiklund [FGW02] and thoroughly discussed in [FSF02; FFS06], that allows robust smoothing of low-level signal features without the main drawback of conventional robust smoothing concerning its applicability in image processing: the high computational complexity, arising from the typically employed iterative solvers for finding the (local) minimum of the objective function.

Channel smoothing uses a *channel representation*[NGK94] of the signal to be smoothed. Channel representation is closely related or analogous to concepts in other fields of research, *e.g. population coding* (computational neurobiology), *radial basis functions* (neural networks) or *fuzzy membership functions* (control theory). From a viewpoint of classical statistics, averaging of the channel representation can be regarded as a regularized sampling of the *probability density function* (*pdf*) of the signal measurements, by means of a kernel density estimator (for a detailed discussion see [For04, chap. 4]).

A (nonlinear) decoding of the averaged channel representation allows to extract the *modes* of the distribution, *i.e.* the local maxima of the distribution. The modes correspond in terms of section 3.2.1 to the different model instances or populations comprised in the signal. It is essentially the decoding step, what makes channel smoothing a robust estimator. We extend regular channel smoothing for PMD-range data, by applying a weighting of the single channel vectors w.r.t. the confidence in the single pixel-measurements and using a new smoothing technique that differentiates between pixels for which the weighting is taken into account and those that use the unweighted channel entries.

The steps involved in the application of our extended B-spline channel smoothing to PMD-data are:

Encoding creation of the B-spline channel representation from the PMD-range data

Two State Smoothing smoothing the channels with a technique we named *two state smoothing*, which allows to weight the range measurement w.r.t. some confidence measure, without the tendency to enlarge the near surfaces as observed for common normalized averaging

Decoding extracting the mode that approximates maximum likelihood from the averaged channel representation, yielding a robust estimate of the surface distance

In section 6.1 you can find an application of this novel extension to B-spline channel smoothing, that we will describe in the following paragraphs in detail.

Encoding The range signal is transformed to the B-spline channel representation, *i.e.* (sparse) vectors of B-spline values at every pixel position. The channel representation for a bounded signal $f(\boldsymbol{x}) \in [1.5, N - 0.5]$ is given by an encoding into N channels at pixel positions $\boldsymbol{x}$

$$c_n(\boldsymbol{x}) = B_2(f(\boldsymbol{x}) - n), \quad n = 1 \ldots N, \tag{3.44}$$

where the quadratic B-spline $B_2(f)$ is given by convolving the rectangle function $\Pi(x) = H(1/2 - |x|)$ two times with itself, yielding the explicit piecewise definition

$$B_2(f) := \begin{cases} 3/4 - f^2 & |f| < 1/2 \\ 1/2\,|f| - 3/4 \quad \text{for} & 1/2 \leq |f| < 3/2 \\ 0 & 3/2 \leq |f| \end{cases}. \tag{3.45}$$

As the signal needs to be bounded between $[1.5, N - 0.5]$ we have to scale the range data accordingly. If $r(\boldsymbol{x})$ is the range signal bounded to $[A, B]$, it may be transformed as

$$f(\boldsymbol{x}) = \frac{N - 2}{B - A}(r(\boldsymbol{x}) - A) + 1.5. \tag{3.46}$$

As the depth information of a PMD-sensor is based on a phase measurement, implying a specific unambiguity depth-range, and phase corresponds to a periodic domain, one might think about adapting channel representation to this circular topology. Felsberg, Forssén, and Scharr [FFS06] show that this is easily done, by adding the

lower two and upper two channels into two single channels (as they are the same for a periodic domain). However, for PMD-data this would not be of much help, because while phase is periodic, range is originally not, *i.e.* the periodicity of the PMD-sensor's depth-range is only a technical shortcoming.

Two State Smoothing We want to weight the range data according to the confidence measure we derived for the PMD-signal. As described above the (averaged) channel representation may be interpreted as an estimate of the *pdf*. Multiplying the individual channel vectors by the respective pixel confidence, does not change the depth value but only the weighting of the vector with respect to the *pdf*-estimate, similar to the weighting done by a GM-estimator. The mathematical proof that such a weighting is sound from a statistical point of view w.r.t. the validity of the *pdf* is outstanding, but experimental results show a good performance of the proposed method w.r.t. robustness and noise suppression.

Lets suppose the weight image is $w(\boldsymbol{x})$, then the weighted channels vectors (c'_n) are given by

$$c'_n(\boldsymbol{x}) = c_n(\boldsymbol{x})\, w(\boldsymbol{x})$$

We need to average the data in each channel to come to a reasonable estimate for the *pdf* w.r.t. the range of the signal. With a model that assumes local constancy (or smoothness) this can be achieved by Gaussian (or binomial) convolution, as it respects the locality of the model by weighting distant pixels less. However, a pure binomial filtering tends to bias the estimate toward nearer values, if the channel vector are weighted, similar to the case of normalized convolution in section 3.1.5. In the resulting image near surfaces tend to grow, but different to normalized convolution the edges are not blurred but sharp.

The reason for this is that binomial smoothing of the channels creates a nonzero probability estimate for zero-value channel pixels (and the respective value range), if the neighboring channel-pixels are nonzero. Furthermore a nonzero channel pixel will be diminished if it is in the neighborhood of zero-valued pixels, *i.e.* an edge. Because the weighting has a bias to weight the near surfaces more than those far away, the probability estimate for the near value, which has been zero before smoothing, tends to become bigger than that of a channel pixel farther away.

Thus, we propose a new smoothing algorithm for a channel representation of range data that is going to be weighted and which differentiates between zero and nonzero

channel pixels, and therefore was named *two state channel smoothing*. For zero valued pixels we use the unweighted (w.r.t. the confidence measure) neighborhood to find a *pdf*-estimate, while for the nonzero pixels we use the weighted neighborhood. The nonzero pixel estimates need to be normalized w.r.t. the weighting to be comparable with the zero pixel estimates. Eventually, we calculate the estimate $\boldsymbol{c'}(\boldsymbol{x})$ of the *pdf* for pixel $\boldsymbol{x}$ as follows :

$$\boldsymbol{C_{nz,n}} = \frac{\boldsymbol{B} * (\boldsymbol{W} \cdot \boldsymbol{C_n})}{\boldsymbol{B} * \boldsymbol{W}} \tag{3.47}$$

$$\boldsymbol{C_{z,n}} = \boldsymbol{B} * \boldsymbol{C_n} \tag{3.48}$$

$$c'_n(\boldsymbol{x}) = \begin{cases} c_{nz,n}(\boldsymbol{x}) & \text{if} \quad c_n(\boldsymbol{x}) \neq 0 \\ c_{z,n}(\boldsymbol{x}) & \quad\quad\; c_n(\boldsymbol{x}) = 0 \end{cases}, \tag{3.49}$$

where small letter variables denote the functional representation of the corresponding matrices $\boldsymbol{C}$, with indices z for zero-pixel, nz for nonzero-pixel and n indicating the channel number. Equation (3.47) is normalized averaging with a binomial applicability $\boldsymbol{B}$, while (3.48) is just plain binomial smoothing.

Decoding The decoding of the encoded signal $c_n(\boldsymbol{x}) = B_2(f(\boldsymbol{x}) - n)$ can be achieved by the linear interpolation

$$f(\boldsymbol{x}) = \sum_{n=1}^{N} n\, c_n(\boldsymbol{x}) \ . \tag{3.50}$$

This is a result from describing a function $P(f)$ by a B-Spline approximation

$$P(f) = \sum_n \alpha_n B_2(f - n),$$

and requiring that $P(f) = f$, *i.e.* the identity function. For this case one obtains the approximation coefficients to be $\alpha_n = n$ [FSF02].

If we interpret $f(\boldsymbol{x})$ as a random variable and $\boldsymbol{c'}$ as a kernel density estimate of its *pdf*, we come to an estimate of the first moment of f, by replacing $\boldsymbol{c}$ with $\boldsymbol{c'}$ in (3.50). Thus, (3.50) gives us an estimate of the expectation of f, because the first moment about zero of a probability distribution is the expectation value of the corresponding random variable. We may reformulate (3.50) as the first central moment, which is zero for a *pdf*

$$\sum_n (n - \hat{f})\, c'_n = 0 \ , \tag{3.51}$$

where $\hat{f}$ is an estimate of the unperturbed signal. This formulation corresponds to the constraint equation (3.35), but $\sum_i$ is hidden in the channel vector. To understand this, we take a look at the continuous formulation of the optimization problem we are dealing with in the limit of an infinite number of measurements:

$$\hat{f} \;=\; \underset{f_0}{\operatorname{argmin}}\, E(f_0)\;, \tag{3.52}$$

$$\text{where} \quad E(f_0) \;:=\; \int \rho(f - f_0)\,\mathrm{pdf}(f)\,\mathrm{d}f = (\rho * \mathrm{pdf})(f_0)\;. \tag{3.53}$$

The condition of a vanishing derivative at the minimum gives us the continuous formulation of (3.35):

$$\begin{aligned} 0 \;&=\; \partial_{f_0} E(f_0)|_{f_0=\hat{f}} \;=\; -\int \rho'(f-\hat{f})\,\mathrm{pdf}(f)\mathrm{d}f \\ &= (\psi * \mathrm{pdf})(\hat{f})\;, \quad \text{where } \rho'(r) = \partial_r \rho(r) = \psi(r)\;. \end{aligned} \tag{3.54}$$

Looking back at equation (3.51) and comparing it with equation (3.54), we may identify the first central moment as a discrete convolution at $\hat{f}$ of the sampled identity function (n) with the sampled *pdf* (c'_n), and we conclude that the influence function ψ of channel smoothing with linear decoding (3.50) is the identity function $\psi(r) = r$. As we know, this corresponds to a least squares estimate and is therefore not robust.

We need to make the decoding robust, as the *pdf* estimated by $\boldsymbol{c'}$ may be multimodal or contain outliers. This can be achieved by making ψ a hard redescender, doing a windowed reconstruction about the mode of $\boldsymbol{c'}$. The window size is chosen to be three, because we need to keep the window size as small as possible, to achieve a minimum computational effort; and three channels are the minimum to reconstruct a measurement f encoded via (3.44) without errors. Instead of changing the width of the influence function (corresponding to the decoding window size), the degree of robustness of channel smoothing can be controlled by adjusting the number of encoding channels N. This is because the robustness is determined by the width of the influence function relative to the number of channels N.

The appropriate number of encoding channels N depends on the (Gaussian) noise level σ_f of the signal $f(\boldsymbol{x})$. In order to reject not more than 5 percent of the inlier samples, the distance between two channels must be greater than $4\sigma_f$ [FFS06].

The robust reconstruction reads

$$\hat{f}_{n_0(\boldsymbol{x})} = \frac{1}{E(n_0)} \sum_{n=n_0-1}^{n_0+1} n\, c'_n(\boldsymbol{x}) = n_0 + \frac{c'_{n_0+1}(\boldsymbol{x}) - c'_{n_0-1}(\boldsymbol{x})}{E(n_0)}\ , \tag{3.55}$$

where $E(n_0(\boldsymbol{x})) = c'_{n_0+1}(\boldsymbol{x}) - c'_{n_0-1}(\boldsymbol{x})$ is the probability for the estimate $\hat{f}$ to be within the value-range of the corresponding decoding window about n_0. The channel window center n_0 should be near to the mode of *pdf*, *i.e.* the location of its global maximum, such that the decoded signal value becomes a maximum likelihood estimate of the unperturbed signal. There are several possibilities to choose n_0 with the given kernel density estimate $\boldsymbol{c'}$. We decided for the computational efficient, but not necessarily best method to choose $n_0 = \text{argmax}_{n_0}(E(n_0))$, such that the determined channel window has the largest sum of channel values. For a multimodal *pdf* the modes may be located near to each other, such that n_0 might be chosen to lie in between the modes, what leads to a wrong estimate.

Based on the windowed reconstruction of the signal value equation (3.55), taking into account the definition of the channel vector entries c_n (3.44) and assuming an infinite number of samples, the effective influence function of channel smoothing can be calculated analytically [FFS06]:

$$\psi(\Delta f) = B_2(\Delta f - 1) - B_2(\Delta f + 1), \text{ where } \Delta f := f - n_0. \tag{3.56}$$

The depicted function $\psi(\Delta f)$ in figure 3.4 is however not precisely the influence

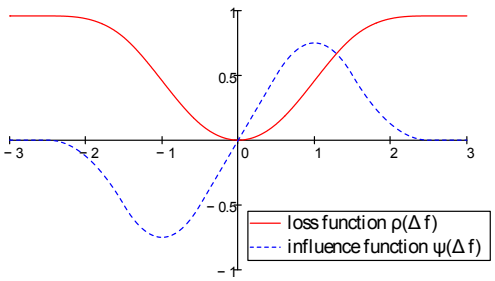


Figure 3.4: Influence and loss function of robust channel smoothing

function, because therefore Δf had to be the residual of the measurement, which is defined as $f - \hat{f}$. Thus, the true influence function is shifted for the rounding difference $\left|n_0 - \hat{f}\right|$. Therefore, the influence function is no longer ideal w.r.t. the

broken (odd) symmetry about zero, what introduces a (minor) quantization error for all estimates $\hat{f}$ that are no integer values.

Felsberg, Forssén, and Scharr [FFS06] propose a method called *virtual shift decoding* that resolves this problem of channel smoothing. However, this method is somewhat involved and expensive w.r.t. computation time. We compared the results of both methods on PMD-data and found no significant improvements, given a high number N of encoding channels, as it is appropriate for PMD-data. For signals that have a high noise variance (low SNR), implying a low number of encoding channels, virtual shift decoding is of more interest, because its computational effort scales with the number of channels and the quantization errors are the more prominent the smaller the number of channels is.

Chapter 4

Motion Estimation

Motion estimation has become an important discipline in computer vision. There is hardly any complex computer vision task, that has not to deal with motion. In various industrial and scientific applications the movement within a scene needs to be accounted for. There is a multitude of tasks that are obviously related to motion estimation, like time-to-collision estimation or pedestrian detection in automotive industry. Another example is particle tracking for the visualization of flow fields of liquids or gases or more general the analysis of dynamical processes in scientific applications. The calculation of displacement fields for motion-based compression of video sequences involves motion estimation too. For many other tasks, however, the link to motion estimation is not so obvious, although it is still inherent. For instance image registration or disparity estimation in stereo vision. Even for still image processing the visual systems of mammals employ the motion analysis pathways of the brain. For example, while humans are looking at a picture, their eyes perform so-called (micro-)saccades for the analysis of the scene. These micro saccades introduce artificial motion on the retina which is then processed by parts of the visual cortex sensitive to the direction of motion and spatial structures (see [Big06]).

Typically the motion estimate is not the actual target of real-world applications. Most times the specific motion estimation algorithm is only one link in a process chain or chains. Therefore, the input and output, as well as the computational efficiency and qualitative performance are subject to various constraints and limitations. This might be one reason why there is such a vast number of different approaches, algorithms and specific implementations for motion estimation. Another reason is that motion estimation from image sequences, which is the topic of this chapter, is in general an ill-posed inverse problem. We will see in the following sections, that various assumptions have to be made in order to make the problem a well-posed one.

This is also the reason why we are talking of an *estimate*. It is only a guess that is true (or rather approximately correct) only if the various necessary assumptions are met. Basically the different assumptions that are made lead to the various algorithms proposed in computer vision literature. Often it turns out that one concept is equivalent to the other or just a special case, formulated in a different manner; this is no wonder, since computer vision is an interdisciplinary research field incorporating the jargon and concepts of various scientific disciplines.

4.1 Optical Flow and Range Flow

4.1.1 Optical Flow and Motion Field

Before we start discussing our approach of motion estimation, we first need to clarify in which kind of motion estimate we are interest and what a motion *estimate* is. We are interested in the motion of objects or rather their surfaces in three dimensional space. This physical motion is partially captured by an optical device, *e.g.* a camera, by taking several images of it over time. Taking an image, typically means projection of the 3D scenery on a 2D plane. Thus the physical 3D vector field of velocity vectors associated with the motion, is projected to the image plane and becomes a 2D vector field known as the *motion field*. The basic motion estimation algorithms try to estimate this motion field from the sequence of images. Horn [Hor87] showed that the reconstruction of the physical 3D motion field from the 2D motion field is possible in most cases if the optical characteristic and the external parameters of the setup (especially the parameters of the projection) are known. However, what the camera (or the human eye) sees is not necessarily as closely related to the motion field as one might think.

The apparent motion at the image plane that is based on the visual perception is known as the *optical flow* or *image flow*. An extreme example of the potential disagreement between optical flow and motion field is given by Horn [Hor86] and depicted in figure 4.1. The figure shows an ideal sphere with an uniform surface. It may rotate around any axes through its center of gravity without any apparent motion. Therefore the optical flow field of the *rotating* Horn sphere is *zero* everywhere. In contrast, a moving light source that illuminates the sphere will change the brightness distribution on the sphere over time, inducing an apparent motion.

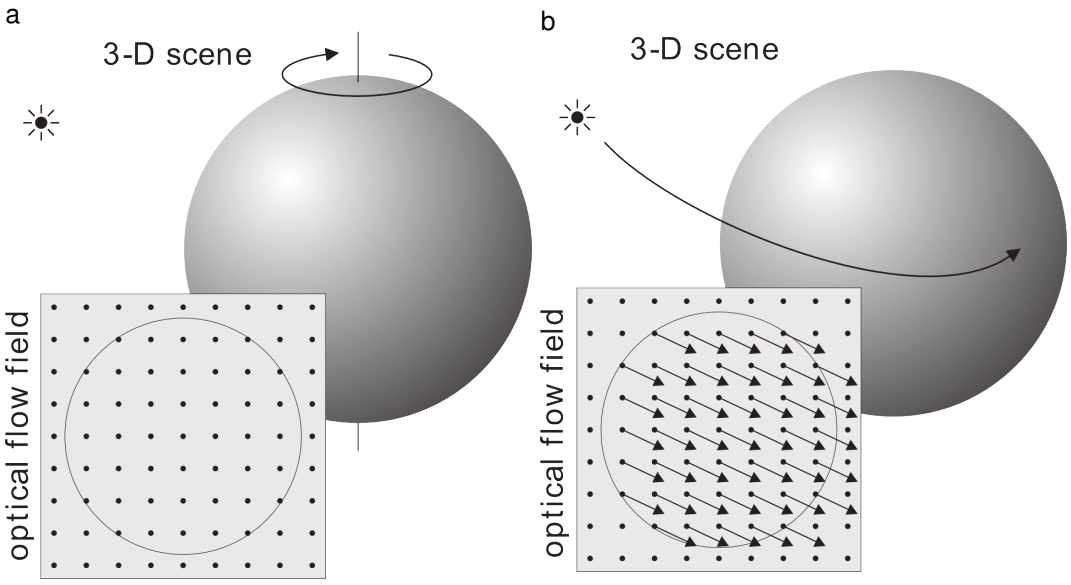


Figure 4.1: Disagreement of physical motion field and optical flow field: a a spinning ideal sphere with fixed illumination shows no apparent motion; b a moving illumination causes an apparent motion of the brightness distribution at no motion of the sphere (images from [JH00])

Therefore we find a nonzero optical flow field while actually the sphere might be at rest.

4.1.1.1 Barber's pole illusion and complex motion

Because human vision is very good in motion estimation, one might presume that the problems described so far are somewhat academic and only problematic because of technical weaknesses or a lack of intelligence in the employed algorithms. We want to stress that, while additional intelligence might help to come to a better motion estimate or rather to a guess that has a higher probability to be correct, it does not solve the general problems. Therefore, we want to present another less academic example. It demonstrates the weaknesses in human vision and illuminates some further problems of associating the optical flow with the motion field. Moreover, it demonstrates that the optical flow is potentially ambiguous because of the so-called *aperture problem*.

The barber's pole on the right[a] is a good example for a situation that appears to be simple to analyze but features various aspects of *complex motion* that are demanding w.r.t. motion estimation and only to handle if specific, rather restrictive, assumptions are made. The barber's pole is a cylinder with diagonal running colored stripes (of a single orientation) rotating around its axis of symmetry. A human observer has the illusion of a motion upward despite the fact that he knows that the cylinder is spinning to the left. This phenomenon is a manifestation of the aperture problem. To put it simply (we will discuss the details later), it describes the following: If we have access only to a limited field of view (the *aperture*) of a moving object and this object has a texture that exhibits only a single orientation, then we can not say anything about motion along this orientation.

Figure 4.2: The barber's pole illusion demonstrates the general ambiguity of optical flow

[a]The electronic version (PDF with JavaScript enabled) of this thesis is needed to see the described optical flow estimation phenomena; zoom in to focus easier on the different apertures; click on the pole to temporarily stop the animation

Thus the motion estimate becomes ambiguous, as only the vector component of the motion normal to the orientation can be determined. Hildreth [Hil82] found an elegant rule that the human vision system applies to come to an unique estimate: The constructed motion field is that which is compliant with the apparent brightness changes and of maximum uniformity within the aperture. Focusing on one of the small rectangular apertures on the right side of figure 4.2, we construct a flow field pointing from right to left, because it is more uniform: there is a discontinuity in the direction of flow only at the shorter vertical edges, while along the longer horizontal there is no discontinuity in the direction of the flow field. If we focus on a larger aperture centered between the pole and the small clippings, our visual system constructs a mixture of both motions which runs normal to the orientation of the stripes. This type of flow field corresponds to a motion estimate known as *normal flow* which we will discuss in section 4.1.3.

It is important to understand that this ambiguity is not a problem of human vision only and not specific to the rather seldom cases of optical illusions. With only the sequence given on the current page one just can not be sure which motion is the true one. Also physical assumptions about possible motions do not help. While a rotation is a good explanation for the horizontal motion, the same is true for the

vertical one, if one assumes a striped, colored ribbon moving like a belt drive. While the single orientation of the barber's pole is rather unnatural, weak textured regions and step edges in images occur quite often, and the scale and size of the analyzed region determines if an aperture problem exists, given the unavoidable uncertainty (*i.e.* noise) in the measurements.

The barber's pole features some other typical problems of motion estimation. There are specular reflections on the cylinder. In the upper part of the pole they are of a magnitude such that the color of the stripes is occluded. Within a small neighborhood along the occlusion boundary there exist two motions. The one of the stripes and that of the fixed specular highlight (a zero-motion). A similar situation we have in the lower part of the pole. The specular reflections are of less magnitude and are transparent superimposed on the moving stripes. This time the two motions are not along the boundary but spread over the area of specular reflection. Remembering the discussion about robust estimation, we realize that this could be handled by a two motion model or a robust estimator. However, both approaches will fail for 3 motions: Imagine the pole protected by some glass tube. The reflections on it might transparently superimpose an additional layer of motion in the surrounding scenery.

Another problem is that we might not be interested in the rotation of the pole but only in translations of the pole's position. For example a computer vision system installed on an automobile, might not know anything about barber poles. How to decide if the apparent motion is of relevance or not, if all information accessible is a sequence of gray or color valued images?

Now that we have illustrated some of the problems in recovering the physical motion field from optical flow we will show how the PMD-signal can be used to come to a motion estimate that is more robust with respect to the motion field we are interested in. We will use an optical flow based approach to motion estimation. The advantage of optical flow compared to correspondence based methods is that they are inherent continuous. Continuous problems can be tackled in a profound way with the mathematical apparatus of analysis and in particular calculus.

4.1.2 Brightness Change Constraint Equation

How to describe optical flow mathematically? Let the point $\boldsymbol{p}$ belong to a small image patch in the 2D-image plane g. If the patch moves at constant velocity $\boldsymbol{f} = \begin{bmatrix} u \\ v \end{bmatrix}$ along

a line and does not rotate, *i.e.* does a translation, then the motion of $\boldsymbol{p}$ is described by

$$\boldsymbol{p}(t) := \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \boldsymbol{p}(0) + \boldsymbol{f}t \ . \tag{4.1}$$

If the motion is different, *i.e.* in some way accelerated (like for a rotational or curved motion), $p(t)$ shall be a first order approximation of the true motion, valid for small t. Lets assume that the image patch just changes its position over time but not its appearance, *i.e.* its gray values (if g is a gray value image). The constancy of the gray value texture along with the motion of the image patch may be stated as

$$g(\boldsymbol{p}(t), t) = const \ . \tag{4.2}$$

If we take the (total) derivative in time of equation (4.2), applying the chain rule and considering (4.1) we find:

$$\begin{aligned} \frac{d}{dt} g(\boldsymbol{p}(t), t) = \frac{\partial g}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial g}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\partial g}{\partial t}\frac{\mathrm{d}t}{\mathrm{d}t} = g_x u + g_y v + g_t = 0 \\ \longrightarrow \quad (\boldsymbol{\nabla} g)^{\mathrm{T}} \cdot \boldsymbol{f} + g_t = 0 \qquad (4.3) \\ \text{or} \quad (\boldsymbol{\nabla}^{st} g)^{\mathrm{T}} \cdot \tilde{\boldsymbol{f}} = 0 \quad \text{, where } \tilde{\boldsymbol{f}} = [u \ v \ 1]^{\mathrm{T}} \ . \qquad (4.4) \end{aligned}$$

Equations (4.3) and (4.4) are equivalent formulations of the well known *brightness change constraint equation* (BCCE). $\boldsymbol{f}$ is the velocity of a point $\boldsymbol{p}(t_0)$ in the image patch, $\boldsymbol{\nabla} g$ and $\boldsymbol{\nabla}^{st} g$ are the spatial and spatiotemporal image gradient at $\boldsymbol{p}(t_0)$ at time t_0, and g_t the partial derivative in time at the same point. The BCCE is valid only inside the patch: At the borders there is a velocity discontinuity as well as a potential discontinuity in the image signals; the spatial and temporal derivatives on g are not defined on both sides of the border. This is of relevance, because the continuous formulation of the BCCE is going to be discretized and the patch border in general does not coincide with a pixel border and thus has a "spatial extension".

The BCCE belongs to the inverse problem of finding the model parameters $\boldsymbol{f}$ for given data $\boldsymbol{\nabla}^{st} g$. It relates the spatiotemporal image structure with the sought optical flow velocity vector $\boldsymbol{f}$, which consists of two unknown scalar values. One equation is not enough to solve the problem uniquely, but it constrains the solution to a line in the (u, v) flow space. However, if the image is constant in the neighborhood of $\boldsymbol{p}$, $\boldsymbol{\nabla}^{st} g$ is a null vector, such that the BCCE is fulfilled for arbitrary $\boldsymbol{f}$ and therefore gives no constraint at all. To come to an unique solution we may take additional points

$\boldsymbol{p_n}$ of the moving patch into account, each one related to a BCCE (4.4), yielding a system of linear equations

$$\boldsymbol{G}_\nabla \cdot \boldsymbol{f} = 0 \ , \tag{4.5}$$

where the matrix $\boldsymbol{G}_\nabla$ contains in its row vectors the respective spatiotemporal derivatives of g at the points $\boldsymbol{p_n}$. It depends on the signal g, if we succeed in finding an unique solution: if $\boldsymbol{\nabla} g$ is linear dependent on g_t, *i.e.* there is only a single orientation in the data, all constraining equations are equivalent and the system is underdetermined. Only if there are exactly two linear independent equations in the system, the solution will be unique. Due to noise this will never be the case for real world data, and typically the system is overconstrained; we may compensate for the given uncertainty in the data by writing $\boldsymbol{G}_\nabla \cdot \boldsymbol{f} \approx 0$. A solution can then be found only from a probabilistic point of view, trying to minimize the error in the estimate $\hat{\boldsymbol{f}}$ of $\boldsymbol{f}$, by *e.g.* a (total) least squares approach (we will discuss this in section 4.1.6).

We realize that in general, motion estimation from image sequences is an *ill-posed* inverse problem, *i.e.* a problem that does not fulfill the postulates of Hadamard [Had02] about well-posedness: it might not have a solution in the strict sense (*i.e.* only a probabilistic estimate can be given), the solutions might not be unique and/or might not depend continuously on the data, in some reasonable topology. This is both true for estimation of optical flow and even more for the estimation of the motion field. Only if one can assure that the various implied assumption are true (like the rigidity of observed objects, which move in conformance to a specific motion model) either by a specific experimental setup or by a sophisticated analysis of the image content, the problem may become partially well-posed. In general any quantitative motion estimate is associated with an uncertainty and therefore motion estimation algorithms should supply a confidence measure, describing the accuracy of the estimate.

In the following we will not address the additional problems involved with noise explicitly, but always keep in mind that there is noise and that all real world flow estimates are subject to noise. For example if we speak of two *equal* BCCE (4.4) for two different points, equality is to be understood as relative to the SNR of the data.

4.1.3 Aperture Problem

We have noted that the BCCE (4.4) is underdetermined. The system (4.5) is underdetermined too, if the single equations correspond to samples of an image patch g of

a single orientation. To clarify this we look at the time evolution of such an image patch, which corresponds to a *rank one signal*: Given the vector $\boldsymbol{n} = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$ normal ($\|\boldsymbol{n}\|_2 = 1$) to the single oriented texture in the image patch g, and the differentiable function $s : \mathbb{R} \longmapsto \mathbb{R}$, we define

$$g^1(x, y, t) := s([x\ y\ t] \cdot \tilde{\boldsymbol{n}}) = s(l) \ ,$$

where $\tilde{\boldsymbol{n}} \in \mathbb{R}^3$ is $\boldsymbol{n}$ extended for the temporal dimension. We find $\tilde{n}_3$ by substituting g^1 for g in the BCCE (4.3)

$$(\boldsymbol{\nabla} g^1)^{\mathrm{T}} \boldsymbol{f} + g_t^1 = n_1\, s'\, u + n_2\, s'\, v + \tilde{n}_3\, s' = s'(\boldsymbol{n}^{\mathrm{T}} \boldsymbol{f} + \tilde{n}_3) = 0 \tag{4.6}$$

such that we may cancel the derivative $s' = \dfrac{\mathrm{d}s}{\mathrm{d}l}$ and solve for $\tilde{n}_3 = -\boldsymbol{n}^{\mathrm{T}} \boldsymbol{f} = -f_n$, if s' is not zero, *i.e.* $s(l)$ must neither be constant nor at an extremum.

All equations for all points at all times within the image patch are of the form (4.6) and therefore equivalent. This is the *aperture problem* of optical flow. We can only determine the flow component f_n normal to the orientation of the image, *i.e.* in the direction of $\boldsymbol{n}$. We may express the *raw normal flow* vector $\boldsymbol{f_n} = f_n\, \boldsymbol{n}$ in terms of the spatiotemporal image derivatives by the following reasoning

$$\begin{aligned} \tilde{n}_3\, s' = -\, f_n\, s' = g_t^1 \qquad &\rightsquigarrow \quad f_n = -\frac{g_t^1}{s'} \\ \left\|\boldsymbol{\nabla} g^1\right\|_2 = \left\|s'\, \boldsymbol{n}\right\|_2 = s'\, \|\boldsymbol{n}\|_2 = s' \qquad &\rightsquigarrow \quad s' = \left\|\boldsymbol{\nabla} g^1\right\|_2 \end{aligned}$$

$$\implies \quad f_n = -\frac{g_t^1}{\|\boldsymbol{\nabla} g^1\|_2} \quad , \quad \boldsymbol{n} = \frac{\boldsymbol{\nabla} g^1}{\|\boldsymbol{\nabla} g^1\|_2} \quad \text{and} \quad \boldsymbol{f_n} = -\frac{g_t^1\, \boldsymbol{\nabla} g^1}{\|\boldsymbol{\nabla} g^1\|_2^2} \ . \tag{4.7}$$

If the patch is not of single orientation we might find the full flow $\boldsymbol{f}$ by taking other points into account. If we take other points into account we have to be sure that they are inside the image patch and not on or beyond the motion boundary, as otherwise the BCCE is not valid anymore. Thus we can not just blindly extend the region around our point of interest, to solve for the aperture problem.

The problem that we need to take additional points into account, but are restricted to a region to choose this points from, which is in general unknown and depends on the flow itself, is referred to as *generalized aperture problem*. And while we would like to take as many data points into account as possible, to achieve a good estimate w.r.t. noise in the data, we are restricted in doing so for the same reasons. There are several ways to deal with the generalized aperture problem, and to find the best flow

estimate under random, or more precisely, generic conditions is a topic of ongoing research [Pap+06; Tel+06; Gov06; Xia+06].

Equation (4.4) is the simplest formulation of the BCCE, which may be extended for more complex motion models like affine flow (see *e.g.* [BA96]) and for multiple motions (see [MSB01; Bar+03; Stu+03]). So far we totally neglected the specific properties of the PMD-signal. For the amplitude-signal of the PMD, the BCCE is a rather bad approximation. A PMD-camera uses an active illumination and therefore motion in depth will involve a major change in the optical power irradiated on each sensor pixel and therefore violates the BCCE. We will discuss later how this can be handled, but first show how to use the range-signal of the PMD, because this also helps to understand how to deal with the amplitude-signal.

4.1.4 Range Flow Constraint Equation

A time varying surface may be viewed as a depth function $Z(X, Y, t)$, with the Cartesian coordinates (X, Y, Z). The coordinate of a point of interest on this surface may be described by

$$\boldsymbol{P}(t) = [X(t), Y(t), Z(X(t), Y(t), t)]^T \ .$$

The function $Z(t) := Z(X(t), Y(t), t)$ with one argument is the Z-coordinate of the moving point on the surface, while the function Z with three arguments describes the time evolution of the surface.

If we take the derivative of $P(t)$ in time and assume a pure translation with constant velocity

$$\begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} = \begin{bmatrix} X_0 + U \cdot t \\ Y_0 + V \cdot t \\ Z_0 + W \cdot t \end{bmatrix} = \boldsymbol{P}(0) + \boldsymbol{f}\, t \ , \text{ where } \boldsymbol{f} \text{ is the velocity vector of } \boldsymbol{P},$$

we yield by applying the chain rule on $Z(t)$:

$$\frac{d}{dt}\boldsymbol{P} = \boldsymbol{f} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} U \\ V \\ U\, Z_X + V\, Z_Y + Z_t \end{bmatrix} \ ,$$

where Z_X and Z_Y are the partial derivatives of $Z(X, Y, t)$. The herein embedded equation

$$W = U\, Z_X + V\, Z_Y + Z_t \tag{4.8}$$

is called the *range flow constraint equation (RFCE)* [Yam+93]; an analogon to the BCCE, that deals with range instead of brightness values.

It constrains the sought solution $\boldsymbol{f}$ for a given range-data-set $Z(X, Y, t)$ at (X_{t_0}, Y_{t_0}, t_0). However, the constraint may only be applied if the surface is smooth with respect to the spatial resolution of the data set, as otherwise the partial derivatives of Z can not be calculated properly (on depth-edges they are not defined at all). Furthermore, with respect to the temporal resolution of the data set, the motion of the surface patch must be well approximated by a translation.

In order to evaluate the RFCE, the partial derivatives of the depth function $Z(X, Y, t)$ with respect to world coordinates X and Y have to be computed. Range data, as delivered by the PMD-camera, is given in sensor coordinates $r(x, y, t)$, with r being the radial distance $|pos_{camera} - pos_{surface}|$ and x, y being the sensor-pixel coordinates*. After applying the transformation from sensor to world coordinate system the range-data is unevenly sampled.

Thus computing the derivatives is no longer straight forward. One can apply TLS (total least squares) or OLS (ordinary least squares) estimation from a local first-order approximation of the surface or resample the data on a Cartesian grid [SG02]. However, both methods have the disadvantage of being rather slow and in the case of resampling the necessary interpolation may introduce additional errors.

In the following we will employ fast derivative filters to compute the world coordinate derivatives; Spies and Barron [SG02] showed that these have competitive accuracy when applied to real-world range data sequences. Because derivative filters are applied via convolution, which implicitly assumes an evenly sampled grid, we have to find a way to compensate for the deviation due to uneven sampling.

Here the objects of interest are 2D surfaces in the 3D world $Z = Z(X, Y, t)$. The data points are sampled at locations on the sensor array, which in turn depend on

*this is a simplified description of the sensor coordinates, for a more precise description see [Jus01, pp. 61 ff]

the 3D data points observed: $x = x(X, Y, Z)$; $y = y(X, Y, Z)$. The transformation from $r(x, y, t)$ to world coordinates yields one data set for each of X, Y and Z on a sampling grid (x, y, t) (*e.g.* $X = X(x, y, t)$).

For the total differential of the three data sets we obtain

$$\begin{aligned} dX &= X_x\,dx + X_y\,dy + X_t\,dt, \\ dY &= Y_x\,dx + Y_y\,dy + Y_t\,dt, \\ dZ &= Z_x\,dx + Z_y\,dy + Z_t\,dt\ . \end{aligned} \tag{4.9}$$

Eliminating dx and dy from equation (4.9) results in:

$$dZ \quad = \quad \frac{1}{-\frac{\partial(Y,X)}{\partial(x,y)}} \left(\frac{\partial(Z,Y)}{\partial(x,y)}\,dX + \frac{\partial(X,Z)}{\partial(x,y)}\,dY + \frac{\partial(X,Y,Z)}{\partial(x,y,t)} \right).$$

The expressions of type $\frac{\partial(A_1,\cdots,A_n)}{\partial(a_1,\cdots,a_n)}$ denote the determinant of the Jacobian matrix of the functions $A_1, \cdots, A_n$ with respect to their arguments $a_1, \cdots, a_n$, which we may abbreviate as *the Jacobian* hereinafter:

$$\frac{\partial(A_1, \cdots, A_n)}{\partial(a_1, \cdots, a_n)} := \left| \begin{bmatrix} \frac{\partial A_1}{\partial a_1} & \cdots & \frac{\partial A_1}{\partial a_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial A_n}{\partial a_1} & \cdots & \frac{\partial A_n}{\partial a_n} \end{bmatrix} \right|$$

Differentiation in time and rearranging $\frac{dZ}{dt} = W$ yields

$$0 \;=\; \frac{\partial(Z,Y)}{\partial(x,y)}\,U + \frac{\partial(X,Z)}{\partial(x,y)}\,V + \frac{\partial(Y,X)}{\partial(x,y)}\,W + \frac{\partial(X,Y,Z)}{\partial(x,y,t)} \tag{4.10}$$

This is the RFCE using derivatives on the (evenly sampled) sensor coordinates x, y (and time t) and thus can be evaluated by convolving the range-data with derivative kernels. Using equation (4.10) implies having transformed the radial range data $r(x, y, t)$ to Cartesian world coordinates.

Instead of applying the filters on the transformed data, it is possible to substitute X, Y and Z with the analytic expressions from the sensor model, so that the derivative filters are applied directly on $r(x, y, t)$.

We use a pinhole camera model, thus

$$X(x,y,t) = \frac{x\,r(x,y,t)}{\sqrt{e}},\; Y(x,y,t) = \frac{y\,r(x,y,t)}{\sqrt{e}},\; Z(x,y,t) = \frac{f\,r(x,y,t)}{\sqrt{e}}$$
$$\text{with} \quad e := x^2 + y^2 + f^2 \quad \text{and } f \text{ being the focal length.} \tag{4.11}$$

After substituting X, Y and Z in equation (4.10) we obtain a somewhat bulky expression, which we simplify by further substitutions and rearrangements to

$$0 = U\,(r\,x - r_x\,e) + V\,(r\,y - r_y\,e) + W\,d - r\,r_t\,\sqrt{e} \tag{4.12}$$
$$\text{where} \quad d = f\,r + \frac{e\,(r_x\,x + r_y\,y)}{f}$$

This new variant of the RFCE reduces the number of necessary filter operations and simplifies error analysis regarding noise in $r(x,y,t)$ and systematic errors introduced by the derivative kernels.

4.1.5 Aperture Problem Revisited

The RFCE poses only one constraint in three unknowns. It describes a plane C in (U,V,W)-space with surface normal $[Z_X\,Z_Y\,1]^T$. The best solution, given this constraint, is the minimal vector $\boldsymbol{f_r}$ between the (U,V,W)-space-origin and C (see figure 4.4a). The raw normal flow for range data is analogous to that of the BCCE equation (4.7)

$$\boldsymbol{f_r} = \frac{-Z_t}{{Z_X}^2 + {Z_Y}^2 + 1}\,[Z_X\,Z_Y\,1]^T = -\frac{Z_t\,\boldsymbol{\nabla} Z}{\|\boldsymbol{\nabla} Z\|_2^2}\;, \tag{4.13}$$

where $\boldsymbol{\nabla} Z$ denotes the 3D spatial gradient of the range measurement Z at the surface point of interest in Cartesian coordinates. Different to the BCCE however a constant neighborhood (or rank null signal) is already sufficient to determine a normal flow, because the RFCE (4.8) contains the additional velocity term W.

Three characteristic types of neighborhoods for range data are illustrated in figure 4.3. Depending on the neighborhood only a specific flow type can be estimated:

plane flow If the neighborhood is of planar structure all constraints are linearly dependent and only the *plane flow* can be calculated (see figure 4.4a). This

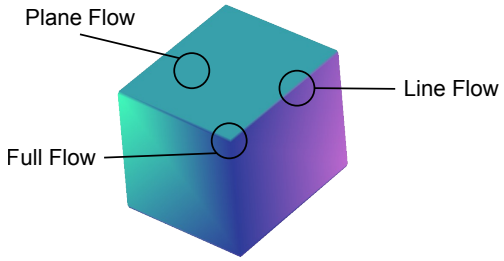


Figure 4.3: Illustration of the three characteristic types of neighborhoods encountered in range data and the corresponding flow types that can be estimated in the respective neighborhood.

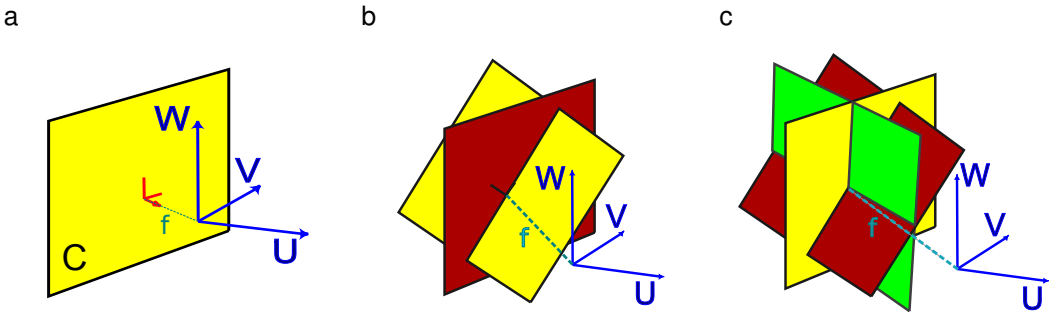


Figure 4.4: The number of independent constraint planes in velocity-space (U, V, W) associated with the RFCE's within an aperture determine the type of flow that can be estimated: a plane flow, b line flow or c full flow.

---

means that if the considered field of view (the *aperture*) shows a pure plane (with no additional structure), only the movement perpendicular to this plane can be detected.

line flow Linear structures in position space such as intersecting planes correspond to two distinct classes of constraint planes in the examined aperture (see figure 4.4b). The point on the common line closest to the origin gives the appropriate line flow. This *line flow* lies in the plane perpendicular to the linear structure. Any movement along the direction of the structure, *e.g.* an edge, can not be resolved by a local analysis; this corresponds to a rank one signals (that we introduced with the aperture problem for optical flow), where $s'(l)$ is not constant.

full flow On corner- or peak-like structures clearly all three components of the movement can be determined locally. In such a neighborhood three linearly independent constraint equations can be found. These correspond to three mutually distinct, *i.e.* non-parallel, constraint planes in the velocity space (see figure 4.4c). The full 3D-flow is readily computed by the intersection of the constraint planes, assuming that the flow is constant in the neighborhood.

The existence of plane and line flow, being the only possible local estimates within specific neighborhoods, is the manifestation of the aperture problem for range flow.

4.1.6 Local and Global Flow Estimation

4.1.6.1 Local Total Least Squares Estimation

In the following we will describe how a local estimate is obtained by means of a total least squares (TLS) technique. We use TLS because it can be computed efficiently and is the appropriate choice if not only the measurement vector $\boldsymbol{b}$ but also the model (or "data") matrix $\boldsymbol{M}$ of an overdetermined (linear) system of equations $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{b}$ is contaminated by noise.

Ordinary least squares (see section 3.2.1) minimizes the residual $\boldsymbol{r} = \boldsymbol{M}\boldsymbol{x} - \boldsymbol{b}$

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{M}\boldsymbol{x} - \boldsymbol{b}\|_2 \ ,$$

which is appropriate if only $\boldsymbol{b}$ is subject to noise; in other words OLS corresponds to perturbing the measurements $\boldsymbol{b}$ by a minimum amount $\boldsymbol{r}$ such that $\boldsymbol{b} + \boldsymbol{r}$ can be explained by $\boldsymbol{M}$ for the model parameters $\boldsymbol{x}$.

Total least squares minimizes

$$\underset{\boldsymbol{p}}{\operatorname{argmin}} \|\boldsymbol{D}\boldsymbol{p}\|_2 \ , \quad \text{subject to } \boldsymbol{p}^{\mathrm{T}}\boldsymbol{p} = 1 \text{ and where } \boldsymbol{D} = [\boldsymbol{M}, \boldsymbol{b}] \ ,$$

which corresponds to perturbing both $\boldsymbol{M}$ and $\boldsymbol{b}$. This is the appropriate problem description for both optical and range flow because, as we will see, the entries of $\boldsymbol{D}$ correspond to the single Jacobians in the RFCE (4.10); the Jacobians, while they dependent on the explanatory variables (x, y, t), are also subject to noise because they are based on the PMD-signal. For a detailed study of TLS we refer to [VHV91] or to [GL80] for a less extensive discussion.

Assuming constant flow in a neighborhood of the point of interest, we get n constraint equations (4.10) if we take n neighboring samples into account. These can be written as

$$
\begin{aligned}
& {}^{k}\boldsymbol{d}^{\mathrm{T}}\tilde{\boldsymbol{f}} = 0\ , \quad k = 1 \dots n \qquad (4.14)\\
\text{where} \quad & {}^{k}\boldsymbol{d} = \left[\frac{\partial(Z,Y)}{\partial(x,y)}\ \frac{\partial(X,Z)}{\partial(x,y)}\ \frac{\partial(Y,X)}{\partial(x,y)}\ \frac{\partial(X,Y,Z)}{\partial(x,y,t)}\right]^{T}\Bigg|_{(x_k,y_k,t_k)}\\
\text{and} \quad & \tilde{\boldsymbol{f}} = \left[\boldsymbol{f}^{T}\ 1\right]^{T} = [UVW1]^{T}
\end{aligned}
$$

Stacking up all equations in the (spatiotemporal) neighborhood gives analogous to equation (4.5)

$$\boldsymbol{D}\tilde{\boldsymbol{f}} = 0 \quad \text{where data matrix } \boldsymbol{D} = [{}^{1}\boldsymbol{d}, \dots, {}^{n}\boldsymbol{d}]^{T} \qquad (4.15)$$

For real world data the rank of $\boldsymbol{D}$ is due to noise typically greater three; at least if more than three samples ($n > 3$) were taken to build the system of equations. It follows that the system (4.15) is overdetermined and there exists no exact solution. One way to deal with this problem is to recast it to an optimization problem and find a solution $\tilde{\boldsymbol{f}}$ in a total least squares sense, *i.e.*:

$$\left\|\boldsymbol{D}\tilde{\boldsymbol{f}}\right\|_2 = \tilde{\boldsymbol{f}}^{T}\boldsymbol{D}^{T}\boldsymbol{D}\tilde{\boldsymbol{f}} = \tilde{\boldsymbol{f}}^{\mathrm{T}}\boldsymbol{S}\tilde{\boldsymbol{f}} \longrightarrow \min. \qquad (4.16)$$

Be aware that $\tilde{\boldsymbol{f}}_4 = 1$ imposes a constraint. As the more generic (but equivalent) constraint $\boldsymbol{p}^{\mathrm{T}}\boldsymbol{p} = 1$ is easier to handle, we replace $\tilde{\boldsymbol{f}}$ by the generic parameter vector $\boldsymbol{p}$. Restating the upper minimization problem in a continuous form gives

$$\hat{\boldsymbol{p}} = \underset{\boldsymbol{p}}{\operatorname{argmin}} \int_{-\infty}^{\infty} w(\boldsymbol{x}-\boldsymbol{x}', t-t')(\boldsymbol{d}^{T}\boldsymbol{p})^{2}\mathrm{d}\boldsymbol{x}'\mathrm{d}t' \quad \text{subject to} \quad \boldsymbol{p}^{\mathrm{T}}\boldsymbol{p} = 1\ . \qquad (4.17)$$

The weighting function $w(\boldsymbol{x}-\boldsymbol{x}', t-t')$ defines the data points $\boldsymbol{d} = \boldsymbol{d}(\boldsymbol{x}', t')$ that are taken into account for the estimate and allows to weight these according to their position relative to the point of interest $(\boldsymbol{x}, t)$. A common choice for w is a three dimensional Gaussian function or rather Gaussian *pdf*, if we require the weighting to correspond to a probability distribution, such that w is normalized, *i.e.* $\int_{-\infty}^{\infty} w\,\mathrm{d}\boldsymbol{x}'\mathrm{d}t' = 1$. A Gaussian weighting pays tribute to the generalized aperture problem, in that it implies the reasonable assumption that far from the point of interest the probability to find the same flow is lower than near to it. From a signal

processing point of view it additionally has the advantage, in contrast to a box filter, that it will not introduce any aliasing because of its limited bandwidth. Nevertheless, it is important that the sampling theorem was not violated by prior signal processing steps like pixel-wise multiplication, as explained in the previous chapter in the context of band enlarging operators.

The requirement $\boldsymbol{p}^{\mathrm{T}}\boldsymbol{p} = 1$ can be incorporated by means of a Lagrangian multiplier λ in the objective or *energy function* E we need to minimize:

$$E = \int_{-\infty}^{\infty} w(\boldsymbol{x} - \boldsymbol{x}', t - t') \left[(\boldsymbol{d}^T \boldsymbol{p})^2 + \lambda (1 - \sum_{i=1}^{n} p_i^2) \right] \mathrm{d}\boldsymbol{x}' \mathrm{d}t' \ . \tag{4.18}$$

Taking the derivatives with respect to the parameter vector $\boldsymbol{p}$ we find the constraint equations for a minimum of E to be

$$\frac{\partial E}{\partial p_i} = \ 2 \int_{-\infty}^{\infty} w(\boldsymbol{x} - \boldsymbol{x}', t - t') \left[d_i (\boldsymbol{d}^T \boldsymbol{p}) - \lambda p_i \right] \mathrm{d}\boldsymbol{x}' \mathrm{d}t' \stackrel{!}{=} 0 \quad \forall \ i = 1 \ldots 4 \ . \tag{4.19}$$

We may take the p_i out of the integral, as they are assumed to be constant:

$$p_1 \int_{-\infty}^{\infty} w \, d_i \, d_1 \mathrm{d}\boldsymbol{x}' \mathrm{d}t' \ + \ldots + \ p_4 \int_{-\infty}^{\infty} w \, d_i \, d_n \mathrm{d}\boldsymbol{x}' \mathrm{d}t' = \lambda \, p_i \quad \forall \ i = 1 \ldots 4 \ . \tag{4.20}$$

The right hand side follows if we require the weight function to be normalized.

The 4 equations (4.20) can be written in matrix form:

$$\boldsymbol{S} \, \boldsymbol{p} = \lambda \, \boldsymbol{p} \quad \text{where} \quad \boldsymbol{S} = \int_{-\infty}^{\infty} w \, (\boldsymbol{x} - \boldsymbol{x}', t - t') \ (\boldsymbol{d} \, \boldsymbol{d}^T) \ \mathrm{d}\boldsymbol{x}' \mathrm{d}t' \ . \tag{4.21}$$

The real symmetric (and positive semidefinite) 4×4 matrix $\boldsymbol{S}$ is an extension of the *structure tensor* [HS99] for range flow introduced by Spies et al. [Spi+99]. Equation (4.21) is the eigenvalue equation for $\boldsymbol{S}$. Each of the 4 eigenvectors corresponds to an extremum of the objective function E, and the (always positive) value of the corresponding eigenvalue is a measure for how close to zero the extremum is (the smaller the closer).

In a discrete implementation, the components of $\boldsymbol{S}$ can be computed using standard image processing operations:

$$\boldsymbol{S}_{i,j} = \langle d_i \, d_j \rangle \ , \ i, j = 1 \ldots 4 \ ,$$

where $\langle \cdots \rangle$ denotes an averaging operator like normalized averaging or plain binomial smoothing.

Gradient Based Weighting While binomial (or Gaussian) averaging is an optimal choice for a smooth flow field and data terms subject to i.i.d. noise, it basically ignores motion boundaries. Because a large spatial gradient in range data typically coincides with a motion boundary (at the border of an object) the author proposes an additional weighting dependent on the magnitude of the spatial gradient to reduce the influence of occlusion on the estimate:

$$w_o(m, \sigma, \mu) = \exp\left(-\left(\frac{m}{\sigma} - \mu\right)^2\right) \quad , \tag{4.22}$$

where m is the magnitude of the spatial range gradient ($\sqrt{d_1^2 + d_2^2}$), while σ and μ control width and center of the function. While the weighting does not solve the aperture problem, it can exclude data points near to an edge, which are likely to bear information that contradicts the motion model due to occlusion. If the aperture of the spatial weighting is small, probability is high that no data points corresponding to a different motion are integrated.

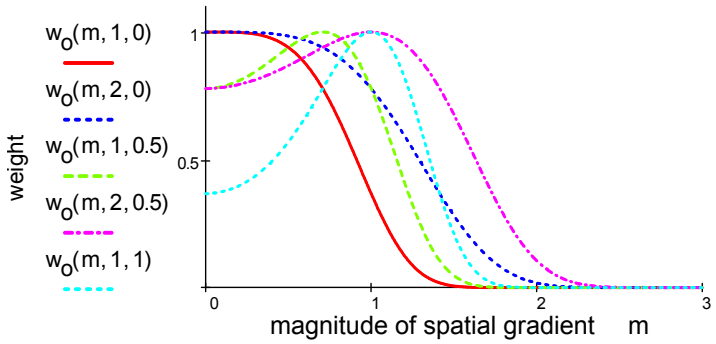


Figure 4.5: Weight function to suppress influence of data at a motion boundary

Furthermore, if the parameters are chosen correspondingly (see figure 4.5), w_o can attenuate the influence of data points with a small gradient, which typically are more critical to integrate w.r.t. the noise in data. To understand this, let us look at the BCCE: $(\nabla g)^{\mathrm{T}} \cdot \boldsymbol{f} + g_t = 0$. If ∇g is small and g_t is large and both are due to noise rather bad measurements, then the BCCE implies an incommensurable large and erroneous normal flow, *i.e.* an outlier. The same is true for range flow. So it is reasonable to attenuate data points with a small gradient. If the neighborhood

contains only gradients of similar magnitude, the attenuation will not influence the estimate. On simulated test data we achieved improved flow estimates in the vicinity of occlusion boundaries using the above weighting. However, the parameters σ and μ were tuned manually and a detailed analysis based on real world data is outstanding. We also used 1-step bilateral filtering for weighting the rows of the data matrix $\boldsymbol{D}$ based on the the difference in depth between central pixel and neighbors (analog to equation (3.43), whereas we replaced $w_r(I_p - I_s)$ by $w_r(Z_p - Z_s)$) but achieved no satisfactory results. The resulting flow estimates were very sensitive to the parametrization of the weighting function w_r and the local structure of the data.

Minimum Norm Solutions The structure tensor $\boldsymbol{S}$ contains all necessary information to determine the local spatiotemporal structure of the data. The estimate $\hat{\boldsymbol{p}}$ is found as the eigenvector to the smallest (or *vanishing*) eigenvalue λ_4 of $\boldsymbol{S}$, if and only if the aperture problem was solved by pooling over an adequate neighborhood, *i.e.* one that corresponds to a full flow. The sought solution is then given by $f_i = \hat{p}_i/\hat{p}_4$ with $i = 1 \dots 3$ or equivalently by

$$\boldsymbol{f}_f = \frac{1}{e_{44}} \begin{bmatrix} e_{41} \\ e_{42} \\ e_{43} \end{bmatrix} \quad . \tag{4.23}$$

The e_{mn} are the entries of the matrix of eigenvectors of $\boldsymbol{S}$:

$$\boldsymbol{E}_{\boldsymbol{S}} = [\boldsymbol{e_1} \cdots \boldsymbol{e_4}] = \begin{bmatrix} e_{11} & \dots & e_{41} \\ \vdots & \ddots & \vdots \\ e_{14} & \dots & e_{44} \end{bmatrix} ,$$

where the eigenvector $\boldsymbol{e_n}$ belongs to the eigenvalue λ_n and the eigenvalues are sorted in descending order, *i.e.* $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4 (\approx 0$, if model assumption were met).

In the ideal case λ_4 is zero. Deviations from this are either contributed to noise in the data or, if the eigenvalue is large relative to the noise, indicate a violation of the model assumptions of a local constant flow, like in the case of occlusion (corresponding to at least two motions) or multiple transparent motion. However, if the aperture problem was not solved by integrating over a set of data points, because for example the surface we are looking at is a pure plane, multiple eigenvalues will be rather small. For this case only the plane flow or the line flow can be estimated locally, as explained in the section 4.1.5. The normal flow is determined by taking the

eigenvectors of either all vanishing eigenvalues or of all non-vanishing eigenvalues into account.

For line flow two eigenvalues vanish and the flow estimate calculated from the eigenvectors of the two non-vanishing eigenvalues is

$$\boldsymbol{f}_l = \frac{1}{1 - e_{14}^2 - e_{24}^2} \left[e_{14} \begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \end{bmatrix} + e_{24} \begin{bmatrix} e_{21} \\ e_{22} \\ e_{23} \end{bmatrix} \right] . \tag{4.24}$$

For plane flow three eigenvalues vanish and the flow estimate calculated from the eigenvector of the single non-vanishing eigenvalue is

$$\boldsymbol{f}_p = \frac{e_{14}}{1 - e_{14}^2} \begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \end{bmatrix} = \frac{e_{14}}{e_{11}^2 + e_{12}^2 + e_{13}^2} \begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \end{bmatrix} . \tag{4.25}$$

The equality $1 - e_{14}^2 = e_{11}^2 + e_{12}^2 + e_{13}^2$ above holds true, because we optimized under the requirement $\boldsymbol{p}^{\mathrm{T}}\boldsymbol{p} = 1$, which every eigenvector needs to fulfill.

Spies [Spi01] derives the general formula for *minimum norm* solutions[†] of TLS range flow estimates which is given by

$$\boldsymbol{f} = \frac{\sum_{i=q+1}^{n} e_{in} \left[e_{i1} \dots e_{i(n-1)} \right]^{\mathrm{T}}}{\sum_{i=q+1}^{n} e_{in}^2} = \frac{\sum_{i=1}^{q} e_{in} \left[e_{i1} \dots e_{i(n-1)} \right]^{\mathrm{T}}}{1 - \sum_{i=1}^{q} e_{in}^2} , \tag{4.26}$$

where q is the number of non-vanishing eigenvalues of a $n \times n$ structure tensor $\boldsymbol{S}$. The left expression calculates the flow based on the vanishing eigenvalues (and corresponding eigenvectors), while the right one does it based on the non-vanishing eigenvalues.

4.1.6.2 Regularization of Local Flow Estimates

If we want to estimate the full flow for a neighborhood that allows only plane or line flow to be determined locally, we needs to make further assumptions, which give further constrains, in a *global* sense. This is typically done in a variational framework

[†] With respect to TLS also the full flow estimate is a minimum norm solution as the structure tensor is rank deficient for all flow types. Only if the model assumptions are violated the structure tensor is of full rank.

that uses a data and a smoothness term, which together make up an energy that is to be minimized globally. The data term derives itself from constraints (or vice versa) of a kind like those we used for the local TLS estimation. The smoothness term is motivated by assumptions of global nature, like for example that the motion of a rigid object is smooth.

Spies and Garbe [SG02] present a variational approach that is based on the local TLS-estimates. Restated for our problem, the regularized motion vector $\hat{\boldsymbol{v}}$ is found as the solution to the following minimization problem

$$\hat{\boldsymbol{v}} = \underset{\boldsymbol{v}}{\operatorname{argmin}} \int_A \left(\omega_c (\boldsymbol{P}\,\boldsymbol{v} - \boldsymbol{f})^2 + \alpha \sum_{i=1}^{3} (\boldsymbol{\nabla} v_i)^2 \right) \mathrm{d}x\mathrm{d}y \tag{4.27}$$

The projection matrix $\boldsymbol{P}$ projects the sought parameter vector $\boldsymbol{p}$ on the solution-subspace to which the minimum norm solution $\boldsymbol{f}$ of the local TLS estimate belongs, such that solutions $\boldsymbol{v}$ that are distant to $\boldsymbol{f}$ w.r.t. this subspace increase the cost within the objective function. The parameter ω_c describes the confidence in the local TLS-solution, and will be defined in the next section. The parameter α controls the overall smoothness of the regularized flow field $\hat{\boldsymbol{v}}(x, y)$; it controls the influence of the smoothness term which penalizes flow fields that have large gradients in the vector components, *i.e.* are not smooth.

The projection matrix $\boldsymbol{P}$ is calculated from the *reduced* eigenvectors $\boldsymbol{b_i}$ of $\boldsymbol{S}$ (which are a basis of the minimum norm solution (4.26)):

$$\boldsymbol{P} = \boldsymbol{B_q} \boldsymbol{B_q}^{\mathrm{T}} \;, \quad \boldsymbol{B_q} = [\boldsymbol{b_1} \ldots \boldsymbol{b_q}] \;, \quad \boldsymbol{b_k} = \frac{1}{\sqrt{\sum_{i=1}^{3} e_{ki}^2}} \begin{bmatrix} e_{k1} \\ e_{k2} \\ e_{k3} \end{bmatrix} \;, \tag{4.28}$$

where q is the number of non-vanishing eigenvalues. For further details on such (TLS) regularization techniques we refer to the work of [SG02; GHO99].

4.1.6.3 Performance Issues

As $\boldsymbol{S}$ is real and symmetric, the eigenvalues and -vectors can easily be computed using the Jacobi eigenvalue algorithm (*Jacobi rotations*), which has the advantage to be inherently parallel (and therefore a parallel implementation is possible [GL96]). However, the method calculates all 4 eigenvalues and corresponding eigenvectors of

the 4×4 matrix, while for a full flow it would be sufficient to calculate the eigenvector to the smallest eigenvalue. This implies that there is potential for a reduction of the computational effort.

Barth [Bar00] shows that the minors of the structure tensor of optical flow (a 3×3 matrix) can be used to calculate a flow estimate with only a fifth of flops needed for conventional structure tensor analysis. However, its questionable if the algorithm can be efficiently extended for 4×4 matrices. Moreover, no normal flows may be calculated with this method (as it assumes only a single eigenvalue to vanish).

If not a complete eigenvalues analysis is of interest, then *partial total least squares* (PTLS) [VHV91] may be used to directly calculate the minimum norm solution. Depending on the structure of the data a performance increase of up to 50% compared to a complete eigenvalue analysis of the structure tensor seems achievable.

Computational most expensive however, is the regularization of the local flows with a variational method, if there is an aperture problem in the local neighborhood. The aperture problem can partially be solved by taking the amplitude information of the PMD-signal into account, as will be discussed in section 4.1.8. This increases the density of full flow estimates that can be achieved by the local method. If the further processing depends on a dense flow field, one may calculate the regularized flow field on a (spatially) downsampled grid, because the spatial resolution of a flow estimate is due to the employed aperture (to overcome the aperture problem) always lower than the original resolution of the data. The original resolution may be regained after regularization, by a cheap bicubic or B-spline interpolation on the original grid. A further speed improvement could be achieved by using multigrid methods similar to those Bruhn et al. [Bru+06] employed for accelerating various variational optical flow techniques.

4.1.7 Confidence and Type Measure

So far we explained how to calculate a flow estimate appropriate to the specific neighborhood. But we also need to decide which neighborhood exists so we may choose the right flow type. Furthermore, we would like to get a measure for the likelihood of the estimate. For this we may exploit the spectrum of the structure tensor $\boldsymbol{S}$.

The smallest eigenvalue λ_4 of $\boldsymbol{S}$ corresponds to the residual of the TLS estimate. Therefore, if it is large relative to the noise level of the data samples ${}^k\boldsymbol{d}$ of equation (4.14), it indicates a violation of the model assumptions and we need to reject the estimate. This may be accomplished by introducing a threshold τ which λ_4 must not surpass if the flow estimate shall be accepted. Spies, Jähne, and Barron [SJB00] propose a *confidence measure* based on λ_4:

$$\omega_c = \begin{cases} 0 & \text{if } \lambda_4 > \tau \text{ (or } \operatorname{tr}(\boldsymbol{S}) < \eta) \\ \left(\dfrac{\tau - \lambda_4}{\tau + \lambda_4}\right)^2 & \text{otherwise} \end{cases} . \tag{4.29}$$

Since the trace of the structure tensor $\boldsymbol{S}$ is essentially the sum over the squared magnitude of the spatiotemporal gradients in the neighborhood and the trace of a symmetric matrix is rotation invariant, $\operatorname{tr}(\boldsymbol{S})$ is a measure for structure in the data independent of its orientation. This is why in [SJB00] pixels are excluded from a further eigenvalue analysis if the trace falls below a specific threshold. While this is reasonable for the optical flow, this is not necessarily so for range data; the plane flow can be calculated for any kind of neighborhood, as depth information is an unambiguous feature of an observed object (compared to brightness information that is somehow fuzzy w.r.t. its significance to describe features of an object).

The author proposes to use the amplitude information A of the PMD-signal to exclude pixels from a local flow estimate, because only if the range information itself is not reliable a flow estimate seems unreasonable.

$$\omega_c(\lambda_4, \tau) = \begin{cases} 0 & \text{if } \lambda_4 > \tau \text{ or } A < \kappa \\ \left(\dfrac{\tau - \lambda_4}{\tau + \lambda_4}\right)^2 & \text{otherwise} \end{cases} . \tag{4.30}$$

Only if we are not interested in the plane flow, we might use additionally or alternatively the trace of $\boldsymbol{S}$ to reject flow estimation. If the trace is small or equal compared to the noise level in the data, one might use it, without further analysis, to assume a plane flow which is approximately zero parallel to the optical axis (*i.e.* Z), such that the subspace of a probable flow vectors is the plane spanned by velocity vectors perpendicular to the optical axis (*i.e.* $(U, V, W = 0)$).

The confidence measure is 1 if there is no residual in the estimate and quickly decreases to zero towards $\lambda_4 = \tau$ as depicted in figure 4.6. Because least squares estimation is a maximum likelihood estimator if the model assumptions (Gaussian noise, constant flow) are met, the residual (or λ_4) corresponds to the width of the

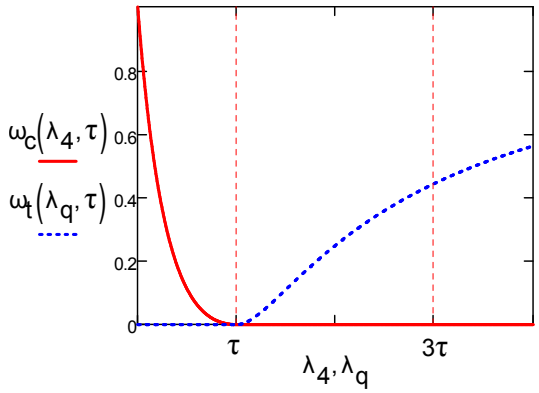


Figure 4.6: Confidence and type measure for range flow

likelihood function and is therefore also a measure for the likelihood of the estimate. However, because a likelihood function is generally not normalized, the author doubts that ω_c is a consistent measure of the likelihood over the various structures that an image sequence may exhibit.

The *type measure* [SG02] allows to measure how well the dimensionality of the nullspace of $\boldsymbol{S}$, *i.e.* the space spanned by the eigenvectors of vanishing eigenvalues, has been determined. This can be done by examining how much the smallest non-vanishing eigenvalue ($\lambda_p > \tau$) is above the noise-level dependent threshold τ. A normalized measure for each encountered type then is

$$\omega_t(\lambda_q, \tau) = \left(\frac{\lambda_q - \tau}{\lambda_q}\right)^2 , \tag{4.31}$$

The type measure ω_t depicted in figure 4.6 increases slowly from $\omega_t(\tau, \tau) = 0$ converging to 1 in the limit of $\lambda_q \to \infty$.

4.1.8 Combining Range and Intensity Data

Apart from the range information discussed so far, the PMD sensor also returns light intensity information that is proportional to the amplitude of the backscattered modulated illumination. Do not confuse this amplitude intensity with the intensity information of an ambient illumination, like it is typical for conventional video systems, as this intensity signal is directly related to the distance *sensor - reflecting surface*.

As for common optical flow we can derive a constraint on the solution from this kind of intensity information. This intensity constraint is deviating from the classical BCCE (4.3), in that it depends on the depth coordinate. Haußecker and Fleet [HF01] showed that the classical BCCE can be augmented in a way, that the brightness can change according to a parametrized time-dependent function h:

$$g(\boldsymbol{p}(t), t) = h(g_0, t, \boldsymbol{l}) \ ,$$

where g_0 denotes the gray value at time $t = 0$ and $\boldsymbol{l}$ represents the parametrization. $\boldsymbol{p}$(t) is the point of interest as defined in equation (4.1), but we substitute $\boldsymbol{v}$ for $\boldsymbol{f}$ to differentiate between the optical flow $\boldsymbol{v}$ and the Cartesian 3D-flow $\boldsymbol{f}$. The total derivative in time on both sides yields the generalized brightness change constraint equation:

$$\frac{d}{dt} g(\boldsymbol{p}(t), t) = g_x u + g_y v + g_t = (\boldsymbol{\nabla} g)^{\mathrm{T}} \cdot \boldsymbol{v} + g_t = \frac{d}{dt} h(g_0, t, \boldsymbol{l}).$$

Substituting the flow vector $\boldsymbol{f}$ for $\boldsymbol{l}$ and modeling the dependence of brightness on distance according to a power law we find

$$h(g_0, t, \boldsymbol{f}) = g_0 \cdot \left(\frac{|\boldsymbol{r}_0|}{|\boldsymbol{r}(t, \boldsymbol{f})|} \right)^a \ , \tag{4.32}$$

where $\boldsymbol{r}(t, \boldsymbol{f}) = \boldsymbol{r}_0 + \boldsymbol{f}\, t$ is the point of interest in Cartesian camera coordinates and $r = |\boldsymbol{r}|$ the radial distance measured by a PMD-camera. For *e.g.* $a = 2$ equation (4.32) corresponds to the inverse square distance law of a punctual light source. Differentiation with respect to t and approximation by a first order Taylor series valid for small t yields:

$$\frac{dh}{dt} \approx -a\, g_0 \frac{\boldsymbol{r}_0^T}{{r_0}^2} \boldsymbol{f} \ , \quad \text{where } \boldsymbol{r}_0 = [X\ Y\ Z]^T|_{t=0}$$

Taking into account the necessary coordinate transformation from sensor- to camera-coordinates and renaming g_0 to $A(x, y)$ (for the measured amplitude at a specific pixel $[x, y]$) the extended BCCE is found analogous to the RFCE (4.10) and (4.14) as

$$\left[\frac{\partial(A, Y)}{\partial(x, y)} + \frac{a\, A\, X}{r^2}, \frac{\partial(X, A)}{\partial(x, y)} + \frac{a\, A\, Y}{r^2}, \frac{\partial(Y, X)}{\partial(x, y)} + \frac{a\, A\, Z}{r^2}, \frac{\partial(X, Y, A)}{\partial(x, y, t)} \right] \cdot \tilde{\boldsymbol{f}} = 0 \tag{4.33}$$

This equation, in contrast to the classical BCCE, constrains all three velocity components and takes into account the brightness change due to a change in the radial distance. Analog to equation (4.12) it may be transformed to a formulation where

the derivative filters are applied directly on the measured range values $r(x, y)$ using the pinhole camera model (4.11)

$$\begin{aligned} 0 = \; & A_t - \frac{r_t\, e \cdot (A_x\, x + A_y\, y)}{f^2\, r + e \cdot (r_x x + r_y y)} \\ & + U \cdot \left(x \cdot d + \frac{A_x \big(f^2 + x^2\big) + A_y\, x\, y + \frac{y \cdot b \cdot e}{r}}{c} \right) \\ & + V \cdot \left(y \cdot d + \frac{A_y \big(f^2 + y^2\big) + A_x\, x\, y - \frac{x \cdot b \cdot e}{r}}{c} \right) \\ & + W \cdot (f \cdot d) \end{aligned} \tag{4.34}$$

where

$$b := \; A_x\, r_y - A_y\, r_x \;, \quad c := \; \sqrt{e}\; (r_x\, x + r_y\, y) + \frac{f^2\, r}{\sqrt{e}}$$

$$d := \; A\, \frac{a}{r\sqrt{e}} \;, \quad e := \; x^2 + y^2 + f^2 \;,$$

power law exponent a and focal length f.

The exponent a may depend on r itself. While we would expect it to be constant $a = 2$ for a Lambertian scatterer, radiometric analysis for the *PMD19k* showed, that it depends on depth (see section 5.2.3). Therefore, if the observed depth range is large, it may be necessary to replace a by $a(r)$. The function can be determined by a radiometric calibration of the specific camera.

While the upper equation looks rather complicated and computational costly, it contains various terms like for instance e or $f^2 + x^2$ that are constant over time and therefore need to be calculated only once. And while we need to calculate 9 different derivatives by convolution (expensive compared to multiplication and addition) for the Jacobians in equation (4.33), there are only 6 needed for equation (4.34). Furthermore, if we are interested in the motion field only, an explicit calculation of the Cartesian range coordinates is not necessary anymore.

The outer form and the sought flow $\boldsymbol{f}$ of equation (4.33) is identical to that of equation (4.14). And thus we can combine both information channels to estimate $\boldsymbol{f}$ in the manner of equation (4.16):

$$\tilde{\boldsymbol{f}}^T (\boldsymbol{S_R} + \beta\, \boldsymbol{S_A}) \tilde{\boldsymbol{f}} \longrightarrow \min. \quad \Longrightarrow \quad \boldsymbol{S}\, \boldsymbol{p} = \lambda\, \boldsymbol{p}, \quad \text{with } \boldsymbol{S} = \boldsymbol{S_R} + \beta\, \boldsymbol{S_A} \;, \tag{4.35}$$

where R and A subscript the *r*ange and *a*mplitude based extended structure tensors. β is used to weight the two data channels with respect to possible differences in signal-

to-noise ratio or other reasons that affect confidence in the data. This way both data channels can be joined in a single, combined structure tensor on which the eigenvalue analysis is applied. Thus we can increase the accuracy of the flow estimate, because we take more samples for a single estimate into account. And what is even more important, we increase the probability to estimate locally a full flow, because the additional structure of the intensity channels may solve the aperture problem.

4.1.9 Equilibration

So far we were a little bit sloppy about the noise in the data. We assumed it to be i.i.d. in the components of the data vector $\boldsymbol{d}$. If we assume the d_i $(i = 1 \dots 4)$ to be i.i.d. random variables with variance σ^2, then their covariance matrix is simply

$$\mathrm{Cov}(\boldsymbol{d}) = \sigma^2 \mathfrak{I}$$

and the structure tensor ${}^n\boldsymbol{S}$ subject to noise is approximately given by[‡]

$${}^n\boldsymbol{S} \approx \boldsymbol{S} + \mathrm{Cov}(\boldsymbol{d}) = \boldsymbol{S} + \sigma^2 \mathfrak{I} \ ,$$

where $\boldsymbol{S}$ is the ideal structure tensor given there is no noise in the data matrix $\boldsymbol{D}$. Under this premise the determined eigenvectors of ${}^n\boldsymbol{S}$ are identical to that of $\boldsymbol{S}$, because the addition of the scaled unity matrix only affects the eigenvalues but not the eigenvectors of the matrix. Therefore a structure tensor corrupted by i.i.d. noise yields an unbiased flow estimate if the model assumptions are met.

Unfortunately in general the noise in the data vector entries is neither independent nor identical distributed, but correlated and of different variances, as a glance at equations (4.10) and (4.33) suggests and as it was discussed in [FPA99]. We will not handle the case of correlated noise in the data vector elements but rather stick to the less involved case of different variances and refer to [MM01; Müh04] for details on this topic.

If the errors in the data entries are uncorrelated and of zero mean but different variance, then the covariance matrix is given by

$$\mathrm{Cov}(\boldsymbol{d}) = \mathrm{diag}(\,(\sigma_i^2)\,) \text{ and } {}^n\boldsymbol{S} \approx \boldsymbol{S} + \mathrm{diag}(\,(\sigma_i^2)\,)$$

The eigenvector of ${}^n\boldsymbol{S}$ are no longer identical to those of $\boldsymbol{S}$ but biased by the noise variance (for details see [VHV91]). We can correct for this if we multiply the data

[‡]only in the limit of an infinite number of data samples the relation becomes an identity

matrix $\boldsymbol{D}$ by a right hand equilibration matrix $\boldsymbol{W}$, which is just the square root of the inverse covariance matrix *i.e.*

$$\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}} = \mathrm{Cov}(\boldsymbol{d})^{-1} = \mathrm{diag}(\,(1/\sigma_i^2)\,)$$

The equilibrated data matrix is then given as

$${}^{e}\boldsymbol{D} = \boldsymbol{D}\,\boldsymbol{W} = \boldsymbol{D}\,\mathrm{diag}(\,(1/\sigma_i)\,)$$

i.e. the columns of the data matrix are weighted by the inverse of the respective standard deviation. The covariance matrix of the respective data vector ${}^{e}\boldsymbol{d}$ is then the identity matrix. This is easy to see if we model the equilibrated data vector entries as

$${}^{e}d_i = \frac{d_i + n_i}{\sigma_i},$$

where the d_i are the unperturbed entries and n_i a noise term of variance σ_i. The expectation values of the structure tensor entries S_{ij} are then

$$\langle {}^{e}d_i \cdot {}^{e}d_j \rangle = \left\langle \frac{d_i\,d_j + n_i\,n_j + d_i\,n_j + n_i\,d_j}{\sigma_i\sigma_j} \right\rangle = \frac{\langle d_i\,d_j\rangle + \langle n_i\,n_j\rangle + \langle d_i\,n_j\rangle + \langle n_i\,d_j\rangle}{\sigma_i\sigma_j}$$

Because the noise terms n_i are of zero mean and uncorrelated all expectation values with a noise term n_i are zero except for $\langle n_i\,n_i\rangle = \sigma_i^2$ and therefore the covariance is just the identity matrix.

Thus we come to an unbiased eigenvector estimate. However, we need to scale the found parameter vector entries ${}^{e}p_i$ of the equilibrated structure tensor, to find the parameter vector that belongs to the original (unequilibrated) problem:

$$p_i = {}^{e}p_i/\sigma_i$$

A flow vector to a full flow is then given by $f_i = {}^{e}p_i\,\sigma_4/({}^{e}p_4\,\sigma_i),\ \forall\ i = 1\ldots 3$.

Finding the correct σ_i for the data vector is not trivial. For example the 4 data vector entries of the extended BCCE (4.34) correspond to 4 nonlinear scalar function $d_i(\boldsymbol{v})$ in the random variables $\boldsymbol{v}$ being A, r and their derivatives. It should be possible to calculate the variance of the resulting random variables from (see [Jäh04, chap. 7.3.2])

$$\sigma_i^2 \approx (\boldsymbol{\nabla} d_i)^{\mathrm{T}}\ \mathrm{Cov}(\boldsymbol{v})\,\boldsymbol{\nabla} d_i\ ,$$

or if we take $\boldsymbol{d}$ as a vector valued function $\boldsymbol{d}(\boldsymbol{v})$

$$\mathrm{Cov}(\boldsymbol{d}) \approx \boldsymbol{J}\ \mathrm{Cov}(\boldsymbol{v})\,\boldsymbol{J}^{\mathrm{T}}\ ,$$

where the Jacobian matrix $\boldsymbol{J}$ (of $\boldsymbol{d}$ w.r.t. $\boldsymbol{v}$) is to be taken at the expectation value of $\boldsymbol{d}$. The covariance matrix $\mathrm{Cov}(\boldsymbol{v})$ needs to be determined from the filter-masks of the employed derivative filters and an estimate about the noise level of the PMD-signals A and r.

However, to speak of an expectation value of $\boldsymbol{d}$ within a necessarily structured and therefore not constant spatiotemporal neighborhood is dubious. The underlying *pdf*'s are multimodal. Because (σ_i) needs to be defined only up to a scaling factor, a normalization of the involved functions might help to solve the problem. However, the author is doubtful about how to handle this topic and a final examination is outstanding.

Therefore we determined the equilibration factors empirically. We used a test pattern subject to Gaussian noise of specific variance σ^2 as input and determined the resulting noise σ_i in the data elements d_i of the RFCE. The constant equilibration factors are then $1/\sigma_i$, because the input noise is only a scaling factor that can be ignored. This approach is however dependent on the specific test pattern and does not consider the local structure of real data. Therefore the presented results in this thesis are improvable w.r.t. equilibration.

4.2 Motion Artifacts

Motion artifacts are a PMD-specific error that occurs around moving reflectivity or distance edges. In figure 4.7 we see two cylinders rotating about their symmetry axis. The cylinders are painted in black and white such that regions of articulate different reflectivity are side by side. Around these moving reflectivity edges we identify heavy errors in both amplitude and range image. The artifact regions have a clear border. It is not blurred like in the case of motion blur of conventional cameras. The faulty distance measurements span the whole unambiguity range of the sensor.

Motion artifacts are a technical weakness of current PMD-sensor. The four cross-correlation samples $I_n = I(\theta_n)$ (2.9), that are used to calculate the range estimate (in a least square sense) based on equation (2.12) (for sinusoidal modulation), are not acquired in parallel (*i.e.* at the same time) but serially. The technical details of the several camera models differ, but non to the author known and commercially available camera model acquires all 4 samples at once. The *PMD19k* and the O3D

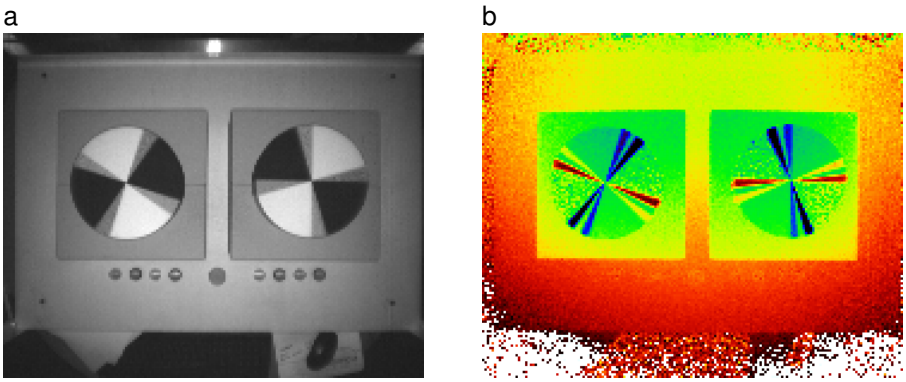


Figure 4.7: Motion artifacts at an irradiance edge. The rotating cylinders show articulate errors around the step edges in reflectivity: a amplitude image, b range image.

(see figure 5.4) take 2 samples at one time[§], the SR3000 is documented to take only 1 sample per time. Therefore, the sensor pixels take cross-correlation samples of different surface patches if the observed surface is moving. Depending on the reflectivity texture and the spatial structure of the surface the calculation (2.12) might yield results that are very different from an averaged measurement (an average would be the "best" possible measurement in such a situation).

Starting from equation (2.10c)

$$I(\theta) = m\,T\left(\frac{A}{\pi}\cos\left(\varphi+\theta\right)+\frac{G_0}{2}\right)\ ,$$

we might model the correlation samples I_n as being proportional too

$$I_n \sim mag_n\left(C\cos\left(\varphi_n+\theta_n\right)+1\right)\ , \tag{4.36}$$

where mag_n is proportional to the irradiance at the pixel and C is the modulation contrast of the illumination. If we assume that two samples of the cross correlation I_n and I_{n+2} are taken at the same time, then the phase φ_n and magnitude mag_n are identical for the samples at $\theta_n = \theta + n\pi/2$ and θ_{n+2} (the addition in the index n shall be in modulo arithmetic mod4).

With the upper equation we can easily calculate the results for different kind of edges (at reflectivity and/or distance edges) if we substitute mag_n and φ_n by values of our

[§]the two samples correspond to the outputs A and B of figure 2.3 that are phase shifted for 180°. Ultimately, the sensor needs to take all in all 8 samples (*i.e.* 4 snapshots) to compensate for manufacturing variations in the capacities of gate A and B.

choice. The results can be calculated from equation (2.12) or equally by doing a least squares fit of $I(mag, \varphi, C, \theta) = mag\,(C\cos(\varphi + \theta) + 1)$ to the sampling points I_n at $\theta = \theta_n$ (as this is exactly what (2.12) corresponds to).

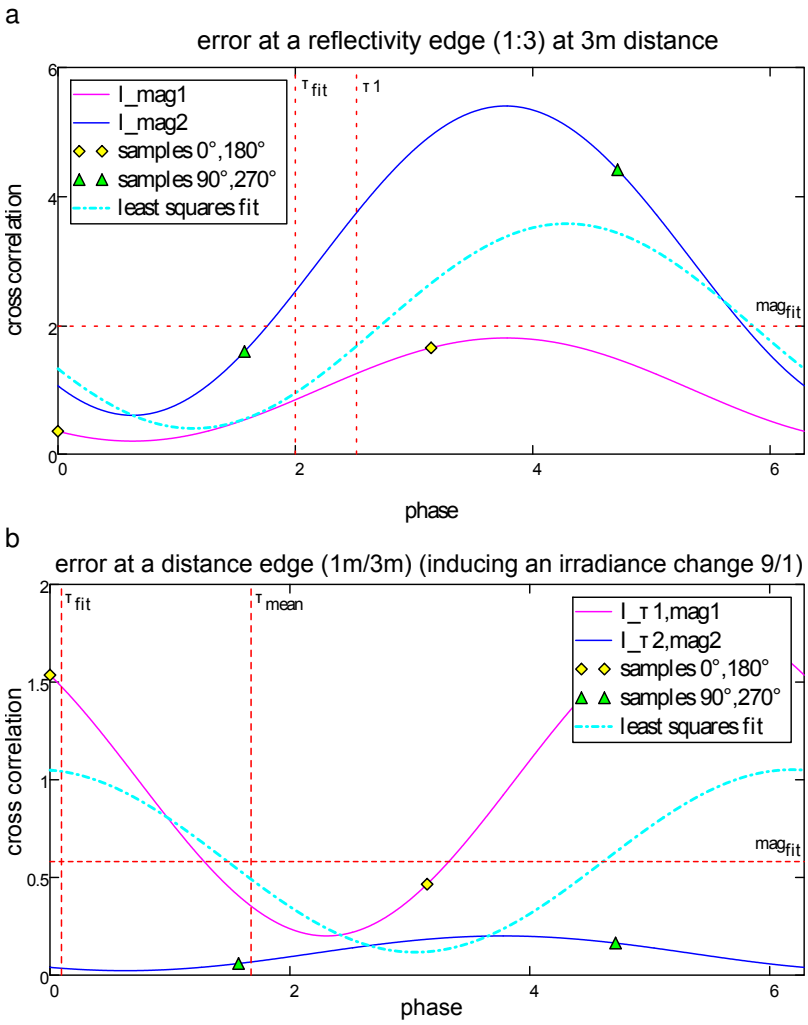


Figure 4.8: Illustration of how motion artifacts occur due to a least squares fit to sampling points of different populations at a a reflectivity edge and b at a distance edge (where the irradiance was modeled according to the inverse square power law and constant reflectivity)

Figure 4.8 shows how the least squares fit produces the motion artifacts. $\tau 1$ in 4.8a is the phase corresponding to the distance of 3m at the reflectivity edge. τ_{fit} ($\equiv \varphi$) is the phase calculated from the fit. Notice that a positive phase/distance $\tau 1$ induces a

shift of the correlation function to the left, such that the maximum of the correlation function is not at $\tau 1$ but at 2π-$\tau 1$. Figure b shows how arbitrary such a least squares fit can be, if the used samples are of two different populations. The calculated value $\tau 1$ (near to zero) is far from the mean value at 2m.

Figure 4.9 gives insight to the quite interesting structure of the motion artifact errors at reflectivity edges: a plots the calculated range against the phase (or distance) and the ratio of the two irradiance magnitudes at the edge. b shows the resulting error in meters. c and d are analogous but show the calculated amplitude. Most interesting is that there are distances where hardly any error occurs.

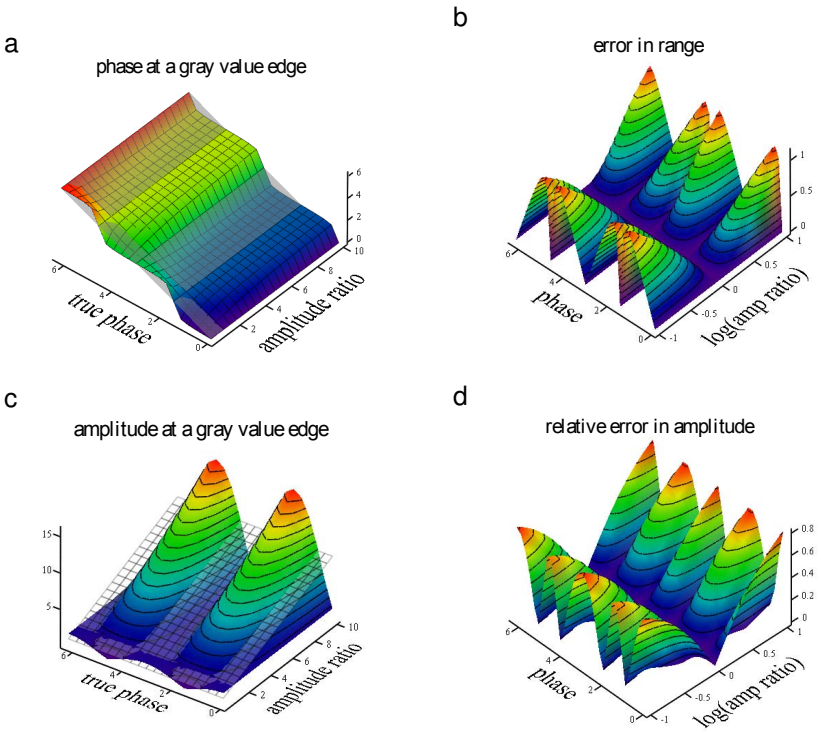


Figure 4.9: Insight into the structure of motion artifacts at reflectivity edges

Figure 4.10 shows the even more complex structure of the errors at distance edges. Again the irradiance was modeled according to the inverse square power law, while the reflectance of the surface was assumed to be constant. The relative error depicted in Figure 4.10b is defined as the quotient $|\varphi_{fit} - \varphi_{mean}| \,/\, |\varphi_1 - \varphi_2|$, where φ_1, φ_2 are the phases (or distances) at the edge and φ_{mean} their mean value. Also at distance edges we can find phase regions where the motion artifact error is less prominent.

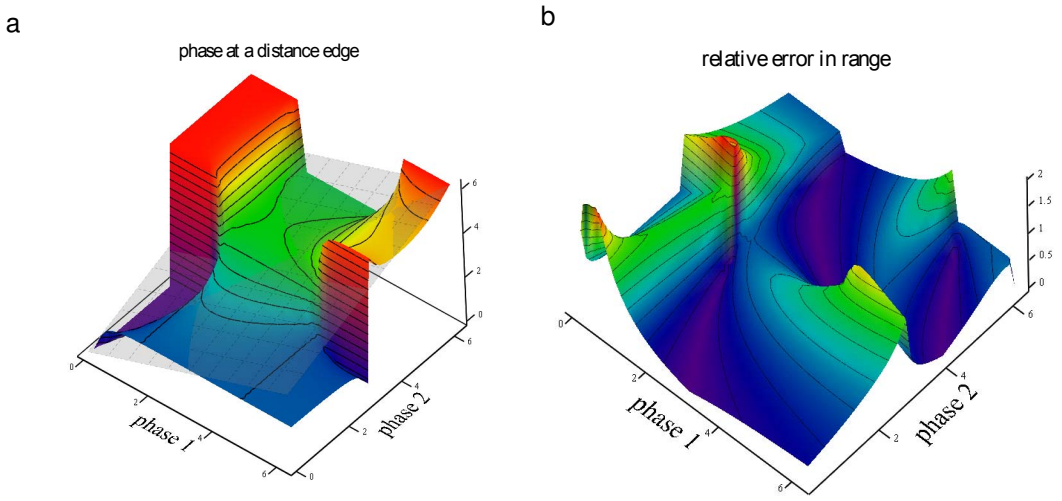


Figure 4.10: Insight into the structure of motion artifacts at distance edges: a calculated range, b relative error in range

---

We showed the plots above, because they may be of interest regarding the planning of experimental setups that incorporate PMD-cameras. If there is a degree of freedom in the choice of the distance or reflectivity range that one plans to use for the setup, one might reduce the errors introduced by motion artifacts by choosing the distance and/or reflectivity according to the regions in the upper (or adequate) plots, where the errors are less dominant.

If we have a sensor that is rather ideal in its correlation measurements, we might detect the presence of motion artifacts by exploiting the symmetry of the correlation signal: For an ideal correlation signal (2.9) the differences in the correlation samples $D_1 = I_0 - I_1$ and $D_2 = I_3 - I_2$ should be identically. Therefore we might just test on $D = D_1 - D_2 < \tau$, where τ is a noise dependent threshold. If D is above the threshold it indicates a motion artifact, and we might process the respective pixel in an appropriate way.

So far, we have not found a way to correct for the motion artifact errors properly. But actually we think that the best way to solve this problem is technologically within the sensor, just by taking all 4 correlation samples at once. While we know that this introduces other problems (like varying capacities of the necessary 4 readout gates) and might be in conflict with constrains to the manufacturing process, we think that it is essential for a robust and uncomplicated processing of sequences which image processes of (highly) dynamic content, *e.g.* structured objects that move rapidly.

Part II

Experiments and Applications

Chapter 5

Testbench Measurements

5.1 Experimental Setup

Most of the sequences we analyzed in the context of this thesis, were acquired with an experimental setup that we will describe in the following and to which we will refer as *testbench*. The setup consists of two motor-driven linear positioner tables mounted on top of standard industry tables. The industry tables are mobile and can be locked with a spoke. Both positioner tables hold object carriers. One carrier holds a TOF-camera the other a target.

Between camera and target a zig-zag shader of black photo pasteboard has been installed, to avoid spurious reflections of the modulated IR light at the glossy positioner table rails; light emitted from the modulated illumination but reflected by other surfaces before it is reflected back by the target to the sensor pixels, has to pass a longer distance then light of direct illumination. The phase information of the irradiated light on the sensor pixels would no longer be consistent and the range measurement becomes corrupted.

We used two arrangements of the tables to acquire the image sequences: a linear arrangement and a T-shaped one. The linear arrangement depicted in figure 5.1 was chiefly used for the calibration measurements as it allows to capture a depth range of 6m, as each positioner has a range of 3m. The depth range acquired was approximately from 30cm to 630cm . Besides calibration measurements also a pure translatory motion in depth can be acquired with this setup.

The second, T-shaped arrangement depicted in figure 5.2 was used for motion measurements. It allows to acquire sequences of motion in the plane within an area equal

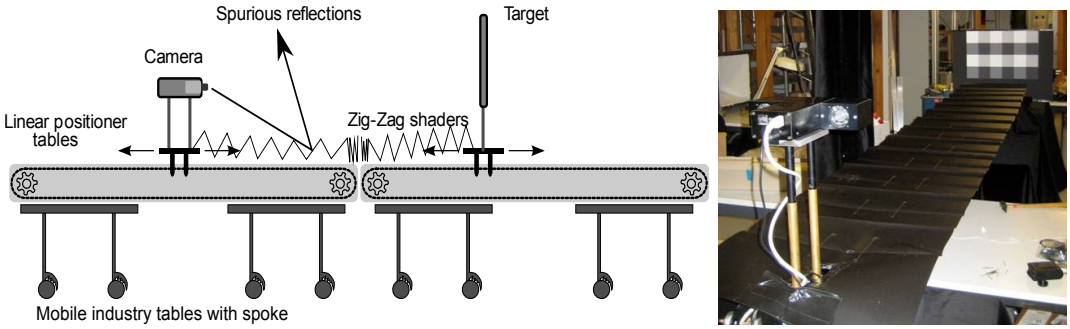


Figure 5.1: Linear arrangement of the positioner tables for calibration measurements

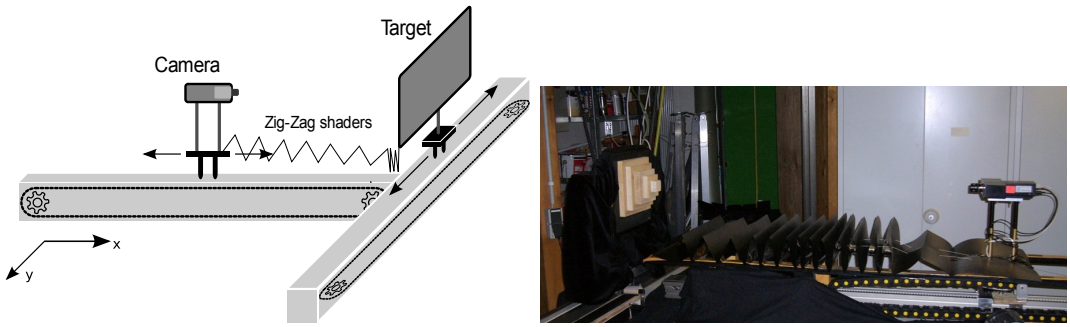


Figure 5.2: T-shaped arrangement of the positioner tables for motion measurements

to that of an isosceles triangle with 3m base and variable height (depending on the basic distance between the tables perpendicular to each other).

Both camera and positioner tables are controlled by a standard PC. This allows a full automation of the experiments (besides the changing of targets and cameras on the carriers). The motor driven tables can be moved stepwise or continuous. The position accuracy is at least 1mm. We used both stepwise and continuous mode to acquire motion sequences. The stepwise mode allows to acquire several images of one position and therefore to do a statistical analysis of the temporal noise in the camera-signal and to separate it from the fixed pattern noise. Images may be denoised by averaging over a number of snapshots. Moreover, motion sequences taken in step mode do not suffer from PMD-specific motion-artifacts and motion blur. Sequences taken in continuous mode are therefore more realistic because they show these artifacts. By noting down the basic distance between target and camera, which we defined as the shortest distance between a point on the camera casing and

a point on the target, we have ground truth information about the relative position of the target to the camera.

We used three different targets shown in figure 5.3. Two of them, *whiteboard* and *checkerboard* target, have a plane surface and were used mostly for calibration. The whiteboard target consists of 8 photo-cards of a specified reflectivity of 84%. The checkerboard target is made of patches of photo-cards of 4 different reflectivities: 84%, 50%, 25% and 12.5%. For a more detailed description of the hardware used in the context of calibration we refer to [Rap07].

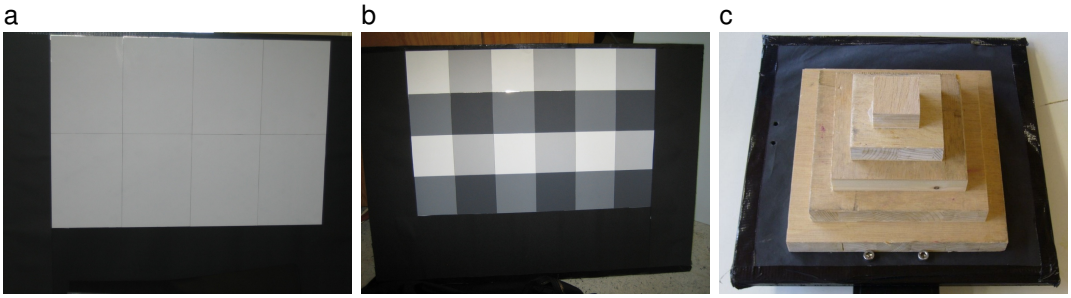


Figure 5.3: Targets used for the experiments: a *whiteboard target* of high-reflectivity b *checkerboard target* and c *pyramid target*

The third target is a pyramidal one. It consists of five wooden quadratic plates, put centered above each other. The bottom plate has a width of 20cm, the top one has 4cm, in between the width is decreasing with 4cm per plate. The depth of each plate is 2cm. The pyramid target was used for motion estimation.

We acquired sequences with three different TOF-camera models. An overview on their technical specifications is given in figure 5.4. All three models are based on the principles of optoelectronic modulation-based time-of-flight measurement, that we presented in section 2.2, and to which we refer as *PMD-technique*. While the acronym PMD (photonic mixer device) relates to a specific realization of this technique protected by patent (held by *PMDTec*hnologies GmbH), we still use it for all similar realization as there is no other common acronym for this technique. The *PMD[vision]®19k* and *O3D* both use sensors from *PMDTec*, while the *SR3000* is a development of *CSEM/MESA*. Little is known about the details of the SR3000 sensor, the main difference seems to be that the *SR3000* a single tap sensor, *i.e.* it has only one storage site (*i.e.* capacity) per pixel, while the sensors of *PMDTec* have 2 storage sites per pixel. For different reasons in the context of this thesis mainly the *PMD19k* data was used. While the *PMD19k* is the oldest model it has a reasonable

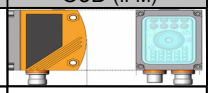
	PMD19k (PMDTec)	SR3000 (MESA)	O3D (IFM)
			
Resolution [Pixels]	160 x 120	176 x 144	64 x 50
Pixel Dimensions [µm]	40 x 40	40 x 40	100 x 100
Focal Length [mm]	12.0	8.0	8.6
Light-Source	2 LED Arrays	1 LED Array	1x LED Array, Laser
λ [nm]	870	850	850
v [MHz]	20, variabel	20, variabel	20
Frame rate [FPS]	1-15	10-20	1-50
Connection	FireWire, Ethernet	USB 2	Ethernet
Dimensions LxWxH[mm]	220x210x55	60x50x65	55x45x85

Figure 5.4: Overview on the technical specifications of the used TOF-camera models

resolution compared to the *O3D*. Furthermore, we needed to get access to the raw (or correlation) data of the cameras, and not only to the depth, amplitude and offset data, that have been calculated by the driver software of the camera. While there is a good documentation on how to access this data for the *PMD19k* and the *O3D*, nothing similar exists for the *SR3000*. While we could get access to the raw-data for the *SR3000*, we found that a depth map calculated from this data shows major systematic errors, which could not be corrected without further knowledge about the internal details of the camera. Because *MESA* would not give us any information about how to process the data, we worked preferentially with the *PMD19k*.

5.1.1 Power Budget

Before we discuss the testbench measurements of the PMD-signal, we need to introduce the relevant radiometric terms and state the power budget for the PMD-camera system. We will give a concise description of results and refer for details on this topic to [Sch03, chap. 2.1.1] and [Lan00, chap. 4.1].

We are interested in the incident radiant energy $Q(r)$ on a PMD-pixel within the exposure time T, in dependence of the observed object *obj* or rather object surface patch at a distance r of the camera. The active illumination i (the *transmitting optic*, *e.g.* a LED-array) is modeled as a light source emitting a radiant flux Φ_i in a solid angle Ω_i. The light is reflected from an object *obj* with a surface that is assumed

to be Lambertian. Specular reflection is neglected. The reflected light is seen by a sensor pixel p of size A_p. The PMD-camera's receiving optic that projects the reflected light of the surface patch on the pixel is characterized by the corresponding solid angle Ω_p of the pixel.

Given the above model assumptions, the following equations hold true and relate illumination i, surface patch obj and sensor pixel p to each other:

$$E_i(r) = \frac{\Phi_i}{\Omega_i\, r^2}\, \cos(\alpha_i)\, \exp(-k_a\, r) \tag{5.1}$$

$$E_{obj}(r) = E_i(r) + E_b$$

$$\Phi(r) = E_{obj}(r)\, A_{vir}(r)\, \eta(r)\, \varrho = E_{obj}(r)\, \frac{\Omega_p\, r^2}{\cos(\alpha_r)}\, \eta(r)\, \varrho$$

$$E_p(r) = \Phi(r)\, \frac{\cos(\alpha_r)}{\pi\, r^2} = \frac{\Omega_p\, \varrho\, \eta(r)}{\pi}\, (E_i(r) + E_b) \tag{5.2}$$

$$Q(r) = E_p(r)\, T\, A_p \tag{5.3}$$

where the used physical quantities are

$E_i(r)$	irradiance of active illumination on the scene/object at distance r $[Wm^{-2}]$
Φ_i	radiant flux of illumination / sending optic $[W]$
Ω_i	solid angle of emitted radiant flux, characterizing the sending optic$[sr]$
α_i	illumination angle: angle of the object's surface normal against the direction of illumination $[rad]$
k_a	absorption coefficient (Lambert–Beer law) $[1/m]$
$E_{obj}(r)$	overall irradiance on the object, including background irradiance E_b $[Wm^{-2}]$
$A_{vir}(r)$	area of a virtual sensor pixel (*i.e.* its projection/image on the object surface)
$\Phi(r)$	reflected radiant flux of a virtual pixel $[W]$
α_r	angle of reflection w.r.t. the sensor pixel $[rad]$
ϱ	diffuse reflectance of the object surface, *i.e.* $1 - \textit{absorption coefficient} - \textit{specular reflectance}$
$\eta(r)$	fraction of exposed pixel-area (< 1, if a surface patch is smaller than a virtual pixel; corresponds to small/distant objects or object edges)
Ω_p	solid angle of pixel, characterizing the receiving optic $[sr]$
E_p	irradiance seen by a sensor pixel $[Wm^{-2}]$

We learn from equation (5.2) that the deposit radiant energy $Q(r)$ does not depend on the distance from object surface to camera but only on the distance from object to light source by $E_i(r)$ (5.1). Because camera and illumination distance are approximated to be equal, Q still depends on the camera distance. Naïvely one might

assume that $Q(r)$ decreases not with r^2 but $(2r)^2$. This is not the case because the solid angle Ω_p seen by a pixel is independent of r. Only if the observed surface patch is smaller than the virtual pixel size $A_{vir}(r)$, which is determined by Ω_p, the irradiance decreases additionally with $\eta(r)$. Actually, this is quite obvious if we think of a scene with ambient (homogeneous) illumination (*e.g.* a landscape on a cloudy day): we will not perceive an object moving away as becoming darker until it is very small and begins to vanish (except for dust or foggy air, corresponding to a high k_a).

Furthermore we note that Q is independent of the angle of reflection α_r and only depends on the illumination angle α_i, by $\cos(\alpha_i)$. If the illuminations angle is approximately zero (*i.e.* the object surface is rather perpendicular to the straight line illumination–object), then small changes in the illumination angle (due to *e.g.* a translatory motion) will change Q only marginal because then $\cos(\alpha_i) \approx 1 - \alpha^2 \approx 1$.

So the radiant power or energy of the active modulated illumination, that can be used to determine the distance r, decreases with $1/r^2$, while the irradiance E_b of the background illumination (*e.g.* from sunlight) stays constant. This implies rather high signal dynamics. Taking into account the dependence of the noise in the range-signal on the background illumination, as discussed in section 2.2.2.2, we find here the most serious problems regarding the applicability of PMD-cameras.

5.2 Depth and Amplitude Analysis

The TLS motion estimation technique presented in chapter 4 depends on range as well as amplitude data of the PMD-camera. Because the flow estimate is essentially based on derivatives of the input data, not so much the absolute value of range information is of major importance but rather the linearity of the range signal in depth (whereas there is also a dependence on the absolute range value in equations (4.14) and (4.34)). Therefore we need to do a calibration of the camera to come to a range measurement as linear in depth as possible. Rapp [Rap07] discussed some of the most important errors of PMD-cameras. We will present results only that are new compared to those in Rapp [Rap07].

5.2.1 Fixed Pattern Noise

First thing we did was to correct for the (temporal) constant depth (or phase) error in each pixel, known as fixed pattern noise. We described the approach in section 2.2.2.1 and used equation (2.26) to calculate a depth image $\boldsymbol{E}$ of the fixed pattern noise of our *PMD19k* model. Figure 5.5a and b shows the noise image and the corresponding histogram of the errors in depth. The histogram is centered at zero and has a

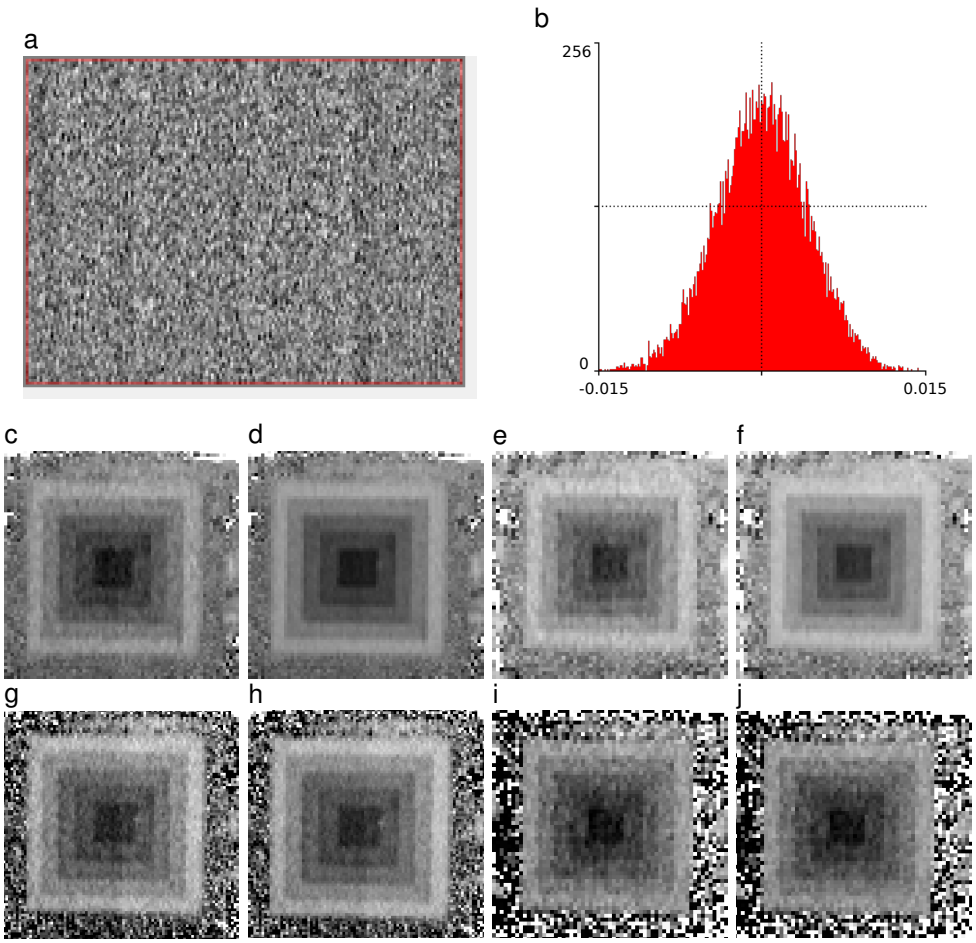


Figure 5.5: Correction of fixed pattern noise: a noise image, b histogram of noise, c original and d corrected image of pyramid target at a distance of 86cm, e & f the same at 134cm. g-j same distances but for input images with a higher level of temporal noise.

standard deviation of approx. 0.425cm. Thus approx. 95% of the pixel errors lie within a range of $[-0.85, 0.85]$ cm, assuming normal distribution of the errors.

The images 5.5c-f show how the subtraction of $\boldsymbol{E}$ from two depth maps taken at different distances (c:86cm and e:134cm) of the pyramid target, improves the depth accuracy (image d and f). The input images c and e were temporal denoised by averaging over a set of 100 images taken with an exposure time of 5ms. The images 5.5g-j show the same as c-f but the input images g and i are single shots (not averaged) taken with 5ms exposure time. The images shown are cropped and scaled (in size and color range) versions of the original images to reveal the changes in detail.

While the fixed pattern noise correction improves the data in all cases its clear to see that for bad illumination conditions (or too short exposure times) as for g and even more for i, the temporal noise dominates and the improvements due to correction for fixed pattern noise are marginal.

5.2.2 Range Calibration

The sequences for range calibration were taken is *step mode*. Step mode means that we acquired at a specific distance [target–camera] a subsequence of a specific number of frames (here 100) of the steady scene and then moved the target (and/or the camera) for a specified distance, to take another subsequence. From each subsequence a *mean image* was calculated by taking the arithmetic mean in each pixel. Furthermore the variance in each pixel was calculated giving a *variance image*.

For range calibration we took several sequences at different exposure times in a depth range from 0.26m–6.22m, at 150 positions with a step-size of 40mm. The different exposure times were necessary to have for all distances images that are neither in underexposure nor overexposure, and to keep the noise level for the calibration rather low and constant. We used the mean images as input to the calibration.

Figure 5.6 shows the error in range measurement calculated from the ground truth data and the whiteboard sequences. We tracked eight positions on the target over the measured depth range using a pinhole camera model (4.11), with a focal length of 12mm, as specified for the *PMD19k*. The positions are marked as magenta points in figure 5.7. We interpolated the depth values at the tracking positions (these are not necessarily on the pixel grid) using cubic interpolation. The ground truth distance, denoted as *range* in the following plots, is the *radial distance* calculated

from the pinhole model using the known position of the targets. The difference between tracked range values and ground truth distance is the range error shown in figure 5.6 as crosses. The range error offset of approx. 0.62m is that large, because the offset-calibration of the camera was erroneous (possibly due to a firmware upgrade).

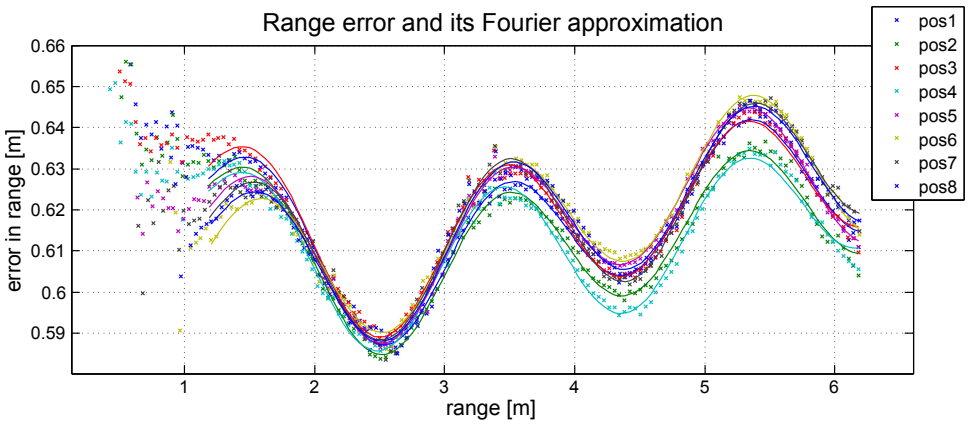


Figure 5.6: Range error (crosses x) and approximation (line —) of range error by 3 modes of a Fourier series for 8 positions on the whiteboard target

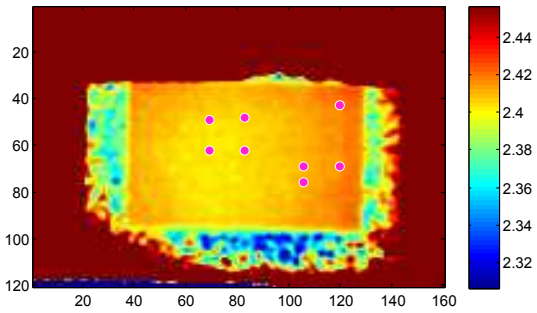


Figure 5.7: Range image with the tracked positions (magenta dots) on the whiteboard

Based on the range errors we fitted a Fourier series equation (2.21) with only the first, second and fourth harmonic. The first harmonic has a wavelength of 7.5m, the unambiguity range of the camera. The approximation, the continuous lines in figure 5.6, fits the error quite well. We used the approximative inverse function of the range error (2.23) to correct for the periodic error in the depth measurements. Figure 5.8a shows the remaining error, if the calibration is applied on the same data set from which the calibration coefficients were calculated. We notice that the *approximation errors*, indicated by the circles (o) all near to zero, are marginal; the approximation error is the difference between the error in the Fourier fit (indicated

by crosses (x)) and the error in the corrected (or *calibrated*) range measurement. Therefore the remaining error of the calibration (line —) is approximately equal to the difference between the Fourier fit and the original errors.

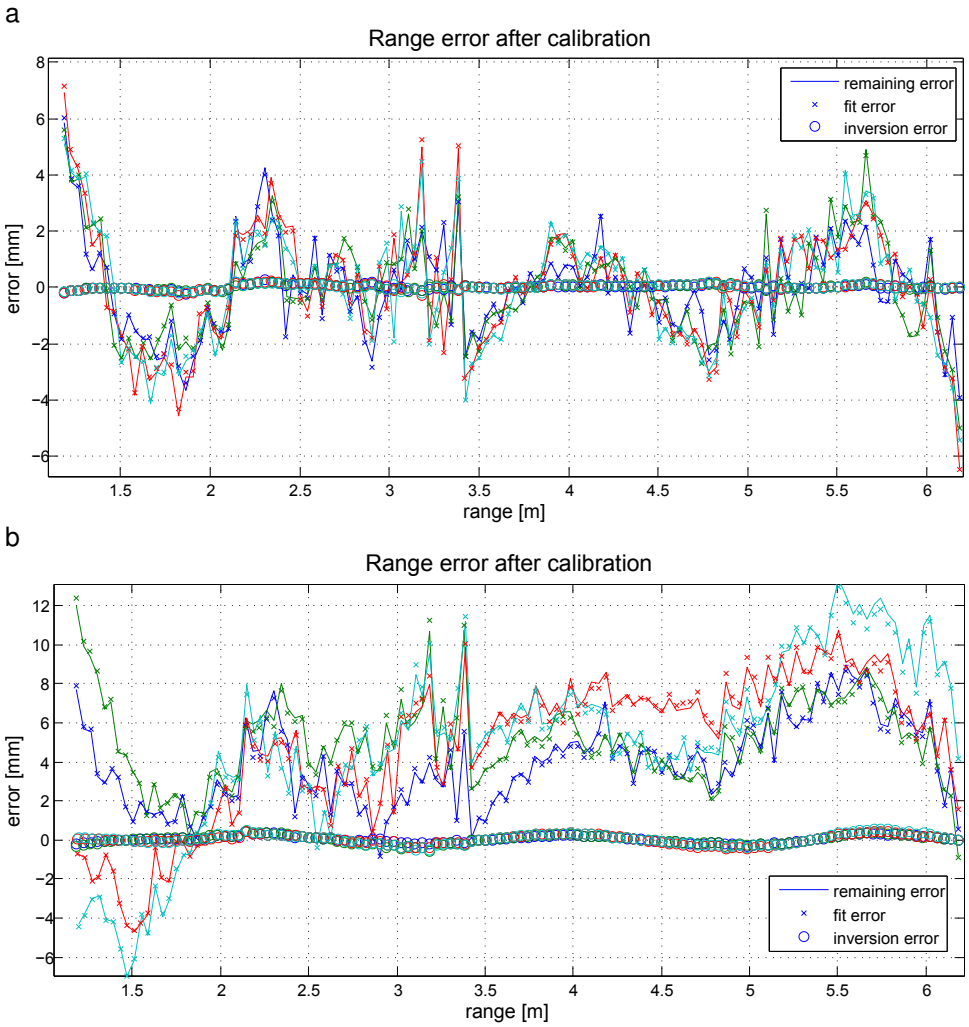


Figure 5.8: Remaining errors for 4 different tracked target positions after a applying the calibration coefficients on the same position they were calculated from and b using a single set of coefficients for the positions

Figure 5.8b shows the errors of the calibration, where we used a single set of Fourier coefficients for the different tracked target positions. The errors were corrected for the position 1,3,5,7, two in the center of the target and two more peripherally, while the calibration coefficient set was calculated from position 2. The errors are clearly

increased but still small against the original errors: the standard deviation σ for the set of 4 position is 3.3mm at a mean error of 4.7mm, while σ of the uncalibrated errors is 15.1mm. The errors introduced by the approximative inversion are rather small again.

Actually, a calibration based on data from tracked points on the target, is not optimal: we intermix the errors of various pixels and can not separate the influence from a change in depth from that of a change in irradiance completely (even by compiling sequences of different exposure time). A more advanced experimental setup would allow to change the depth information at all pixels over the whole unambiguity range. Thus the need for tracking individual target positions would cease to exist and the calibration could be done per pixel as well as the correction of the range errors. For each pixel an individual set of calibration coefficients could be calculated. As we need only 7 real numbers (1 offset and 3 complex Fourier coefficients) per pixel, the memory requirements would be rather low even for an embedded system.

5.2.3 Interdependence of Amplitude and Range

For amplitude analysis and its interdependence to the range measurement we used the checkerboard target. Similar to above we tracked 8 positions on the target. But this time we used a single sequence at an exposure time of 12ms. Figure 5.9 shows a range and an amplitude image of the checkerboard target and the tracked positions on it (the magenta dots).

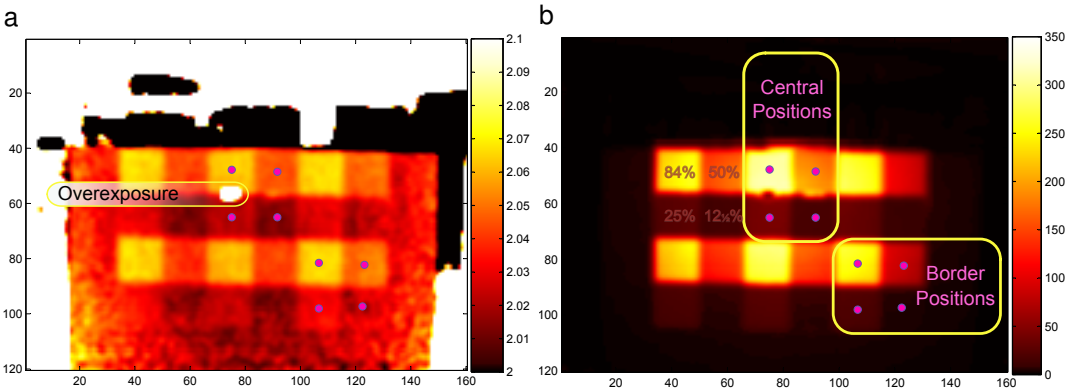


Figure 5.9: Tracked positions (magenta dots) on the checkerboard: a range image, b amplitude image

The most notable difference between the range image 5.9a of the checkerboard and the whiteboard target figure 5.7 are the range variations of several cm within the smooth plane of the checkerboard target. Another interesting, while not intended feature, is the bright white spot between 84% and 25% patch. Its corresponds to a (partial) overexposure due to a defect in the checkerboard target: there is a small gap between the reflectivity patches and the underlying metal plate reflects the light in contrast to the patches not (approximately) Lambertian but specular. While the irradiance is not that strong that the PMD-pixel capacities are all saturated (otherwise the amplitude was zero), the partial saturation leads to a lower amplitude and a higher range measurement compared to the 84% patch above. Right to the bright spot there is a similar defect in the target (between 50% and 12.5% patch), but the gap is smaller and the specular reflection does not cause overexposure. Here both range and amplitude measurement are increased relative to the neighborhood.

Looking at the range-error plot figure 5.10 we easily identify the correspondence between reflectivity differences and differences in the range measurements, while the ground truth distance has not changed. A lower reflectivity coincides with a smaller range measurement. Between the 84% and 12.5% reflectivity patch there is a difference of about 5cm. The reflectivity dependent range difference (RDRD) is quite constant over the acquired depth range. However, near the camera and far away from it, there are obvious variations in the RDRD, which we want to investigate a little further.

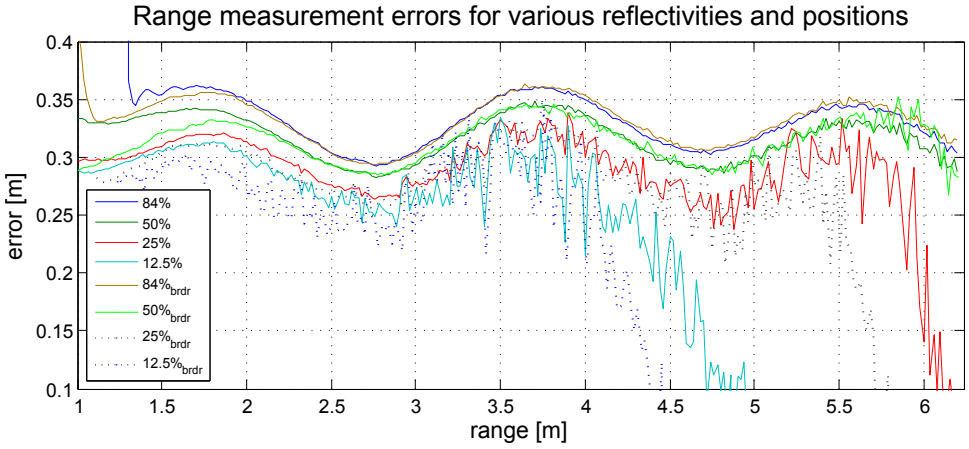


Figure 5.10: Range measurement errors for various reflectivities and positions

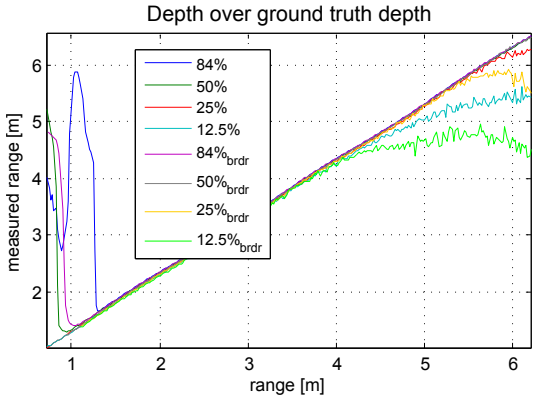


Figure 5.11: Measured range over ground truth range for the tracked positions

---

For large distances the statistical errors in the range measurement for the low reflectivity patches increase due to underexposure particularly strong, with an evident bias for too small distance values. While we are not sure about the reason for the bias, which is even more clear to see in the range plot 5.11, we know that the increase in statistical error is due to the noise dependence of the PMD-signal, that we described by equations (2.29) and (2.30) and illustrated in figure 2.6.

The variations in the RDRD near to the camera can be exemplified best for the 50% center and border position (light and dark green lines in figure 5.10). We find the positions with increasing ground truth range to converge to a common measurement range value (at approx. 2.4m) to keep a constant RDRD to the 84% positions, which behave similar.

Thus we not only have RDRD errors but also an explicit dependence of the range on the position (central or border). The reason for this are near field errors of the camera, because its illumination system is neither punctual nor of rotational symmetry. Generally speaking the assumptions we made for finding equation (5.3) are only valid approximatively in the far field of an extended light source.

The *PMD19k* has two LED-arrays for illumination, that are aligned along the horizontal axis (see the small picture in figure 5.4). Therefore in the near field of the camera the illumination can not be approximated as a punctual light source: while a border position pixel is irradiated by a mixture of light that traveled (essentially) two different distances, a central pixel sees light of only one distance. The demodulation is ideally a linear process, therefore the resulting distance is the arithmetic mean of

the, simplified spoken, "phase information" of the generated photo electrons. Because the irradiance E_i (5.1) depends on the distance from the light source, there will be always more electrons in the mix corresponding to photons that took a shorter path. Therefore the border position pixels are biased for too short distances in the near field of the camera. This is exactly what we see in figure 5.10, most pronounced for the 50% border position that has a too small range (compared to the central positions) up to at least 2.2m.

Another reason for the RDRD variations is overexposure. While for example the dark green line (50% center position) is at 1m not any longer in total overexposure, still some of the correlation samples may belong to (nearly) saturated capacities that corrupt the linearity of the demodulation, as we have explained in section 2.2.2.1 on page 27. Therefore the dark green line at approx. 1m or the dark blue line at 1.4m show too high range measurements.

This *partial overexposure* can be identified even better in the amplitude signal $A(r)$ shown in figure 5.12. While total overexposure corresponds to zero amplitude, partial overexposure shows varying amplitude. We marked the region of partial overexposure by reddish ellipses in figure 5.12. Looking at the log-log plot 5.12b, we see that the amplitude for the high reflectivity patches, reaches a maximum after which it decreases first slowly (small negative slope) to decrease faster for higher distances.

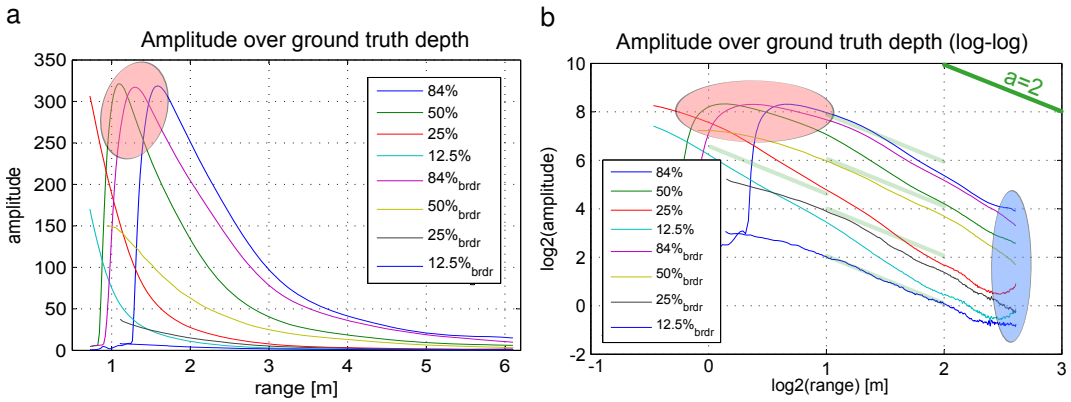


Figure 5.12: Amplitude signal A of the *PMD19k*: a amplitude over radial distance, b log2-log2 plot of a. The reddish ellipses indicate the range of partial overexposure.

The RDRD (including near field errors and partial overexposure) are of the same magnitude as the periodic systematic distance dependent errors, that we corrected

in the previous section. Being constant within only a limited distance range not too near to or too far from the camera, it is an open question if and how they may be corrected.

The bluish ellipse marks a region where we can observe an unusual increase of the amplitude. However, this is only due to a weakness in the experimental setup and an imperfect tracking of the patches: the tracked positions run into false patches; and the manufacturing defects in the target (shown in figure 5.9) induce due to the specular reflection in the surrounding pixels an increased irradiance. Hence we can ignore the measurements in the ellipse as corrupted.

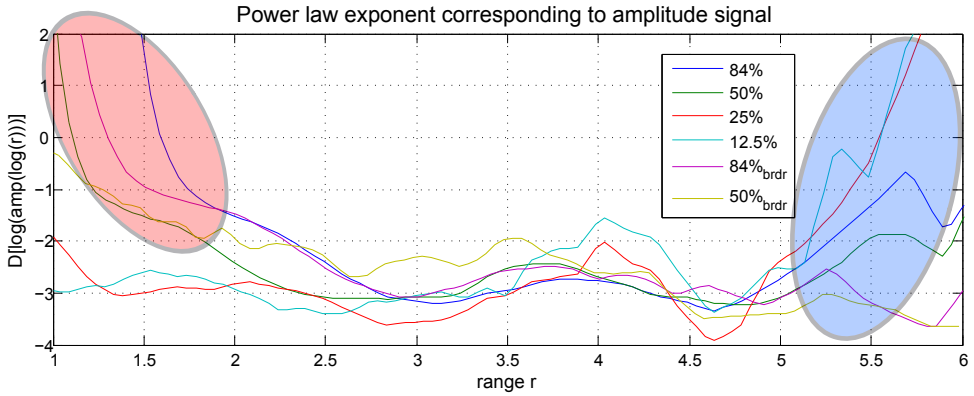


Figure 5.13: The scaling exponent for the amplitude modeled locally by a power law

In figure 5.13 we plotted the *scaling exponent* a of the power law model (4.32) that we would like to use for motion estimation. We calculated the exponent as the first derivative of the log-log data, by resampling the (binomially smoothed) logarithmic amplitude-range data (of the log-log plot 5.12b) using spline-interpolation at evenly spaced sampling position in the logarithmic domain. The derivative may then be calculated as forward difference of the resampled amplitude measurements multiplied by the sampling frequency. This is necessary to compensate for the increased noise in the measurements at farther distance and to adapt for the scaling behavior w.r.t. the power law.

Unfortunately the exponent is not constant, not even beyond the overexposure range in the far field of the camera. Neither a is approx. 2 as $Q(r)$ (5.3) would suggest. The pale green diagonals in figure 5.12b indicate the resulting scaling exponent, if we assume the measured amplitude $A(r)$ to be proportional to $Q(r)$. We suppose that the deviation is partially due to the demodulation contrast (see section 2.2.1.3)

which is not independent of the deposit radiant energy $Q(r)$, as discussed in [Lan00, chap. 5.2.4]. Therefore $A(r)$ is not proportional to $Q(r)$ (see equation (2.18)). Furthermore, if one excludes the range of (partial and total) overexposure one might identify a kind of periodic modulation of the scaling exponent, similar to the periodic range error; but this is very speculative. A final evaluation seems within the limitations of the current experimental setup not possible.

For similar reflectivities and not too bad under- or overexposure the scaling exponents are similar too (compare the blue (84% central), green (50% central) and magenta (84% border) line within a range from 2.7–5.2m in figure 5.13). However, the uncertainty interval is rather high (standard deviation is approx. 0.2). Therefore its questionable if a distance dependent scaling exponent $a(r)$ (calculated from the data for a specific reflectivity or averaged over a reflectivity range) can improve the results of motion estimation essentially. So far, we used for motion estimation only constant scaling exponents.

Chapter 6

Applications

6.1 Still Image Processing

We applied the new *two state smoothing*, that we described in section 3.3, on various real world scenes and test patterns. A final evaluation is however outstanding. Anyhow we want to present some intermediate results because they look promising.

In figure 6.1 you can see a (partial) GUI-snapshot of the image processing software *Heurisko* we used to implement the two state smoothing. It shows different data sets of a scene with a medium level of noise acquired by the *PMD19k* at 20ms exposure time. The upper left image is the range map of the scene which is rich in fine structures. There are errors in the distance data that are much higher in magnitude than the regular noise, *i.e.* outliers (some of them are marked by the red boxes with round edges). The observed distance range is from about 2.0–5.5m.

The image on the right shows an average over 10 frames, that we used as a kind of "ground truth". However, while noise is strongly reduced the outliers are still present. They occur typically either due to under- or overexposure. For example the wooden pillar in the very front of the scene, reflects the light of the PMD partially specular, what leads to overexposure and therefore wrong range estimates. In the lower right of figure 6.1 you find the weight image we used for two state channel smoothing. It is the square root of the amplitude image. We found the square root (or alternatively the logarithmized amplitude) to yield better results than weighting directly with the amplitude.

The lower left of figure 6.1 shows the result of channel smoothing using 50 channels and a binomial 3×3 filter mask. Actually, the results look similar using 25 channels;

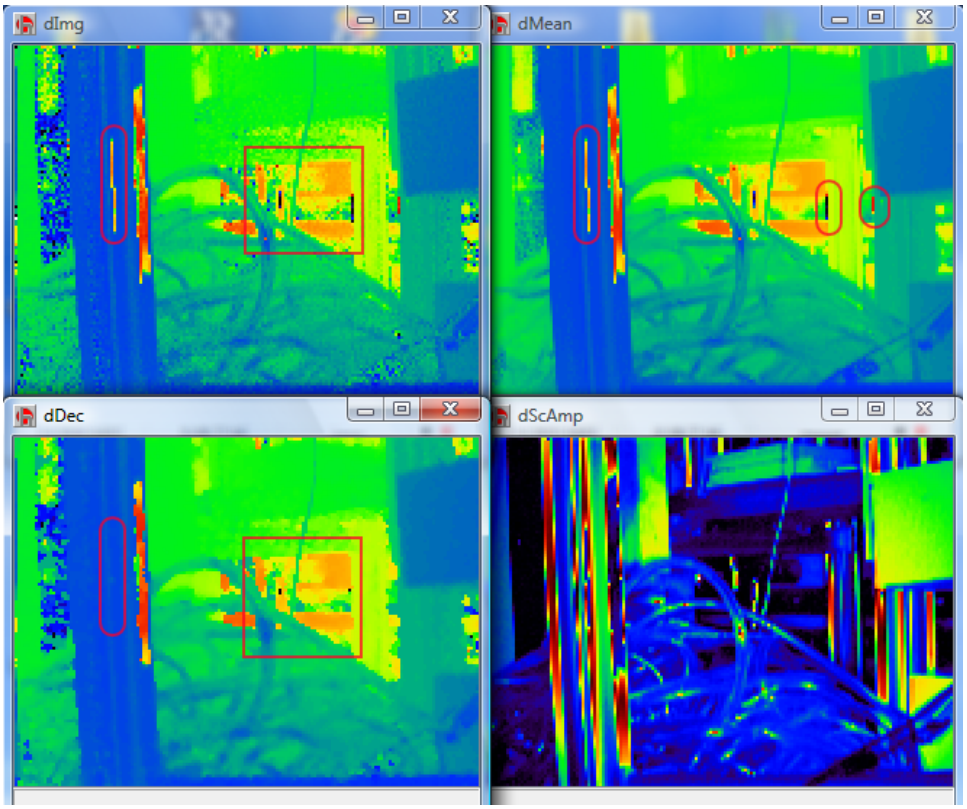


Figure 6.1: Two-State-Channel smoothing applied: original range image (dImg), arithmetic average over 10 frames (dMean), two state smoothed range image (dDec) and the used weighting image (dScAmp).

---

however the very details of the image are partially blurred. We notice that a large part of the outliers has been removed, the image is denoised and the fine details of the image still exist. The algorithm could however not (completely) remove the big faulty area on the wooden pillar. It is just to big to be treated as an outlier using a filter mask size of 3×3.

Simple binomial smoothing depicted in figure 6.2a is, besides the blurring of the object features, also quite problematic, because it also blurs the outliers and therefore introduces new errors in their neighborhood. Conventional B-spline channel smoothing, shown in figure 6.2b, removes less outliers because it is missing the additional confidence information from the amplitude data.

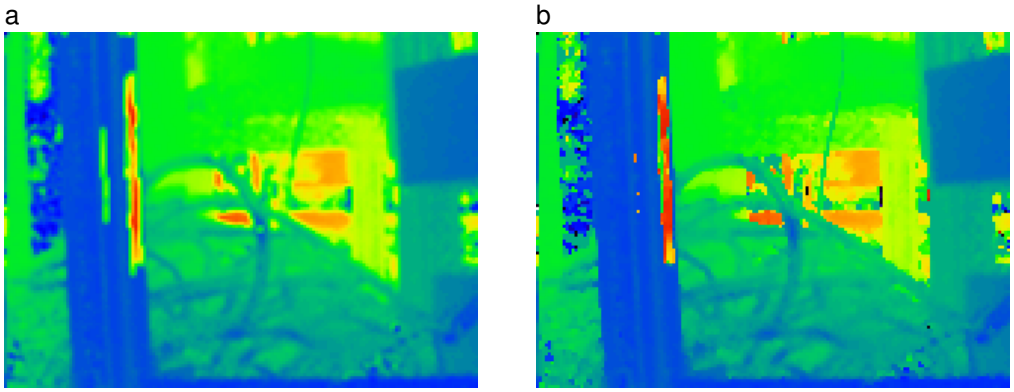


Figure 6.2: a Simple binomial smoothing introduces additional errors around outliers and blurs image features/details. b Conventional B-spline channel smoothing removes less outliers.

In figure 6.3 we depicted a detail of the scene (marked by the central red box in figure 6.1) denoised with different methods. Particularly notice the thin line coming from the top (a cable): Channel smoothing conserves all the details while binomial smoothing destroys them. The averaged range image b and the channel smoothing result d are very similar, but in d only 3 outlier pixels are left, while b is rather identical to a w.r.t. outliers.

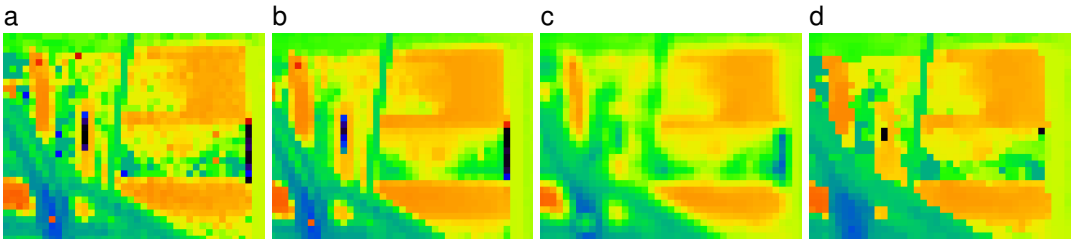


Figure 6.3: Details of the scene: a original with noise and outliers, b temporal average over 10 frames, binomial smoothing (c) and channel smoothing with 50 channels (d) applied to a

Using the mean image as ground truth and the amplitude data as a confidence measure we calculate a quality measure Q:

$$\begin{aligned} q_{ch} &= \text{clip}\left(\log \frac{\max((r-m)^2, \text{2.0e-6})}{(s_{ch}-m)^2}, -3, 3\right) \ , \\ Q_{ch} &= \langle q_{ch} \rangle \ , \end{aligned} \tag{6.1}$$

where r, m and s are the pixel values of the original range image $\boldsymbol{R}$, the mean image $\boldsymbol{M}$ and the channel smoothed image $\boldsymbol{S}$. $\langle q \rangle$ is the arithmetic mean over the all pixels that have an amplitude of more than 0.3. And $\mathrm{clip}(x, a, b)$ limits the summands to values in the range $[a, b]$. s_{ch} depends on the number of channels ch used for smoothing and therefore Q depends on ch too.

Figure 6.4 shows how the quality measure Q grows with increasing channel number to find a constant level at approx. 60 channels. For larger filter mask sizes the quality measure looks similar. However, if one uses the same scene but bearing a higher noise level (because it was acquired with a shorter exposure time), the quality finds a maximum somewhere around 55 channels and then slowly decreases. The quality for conventional channel smoothing is always below two state channel smoothing. The quality measure for binomial smoothing is equal to Q_1 and less than 0.02.

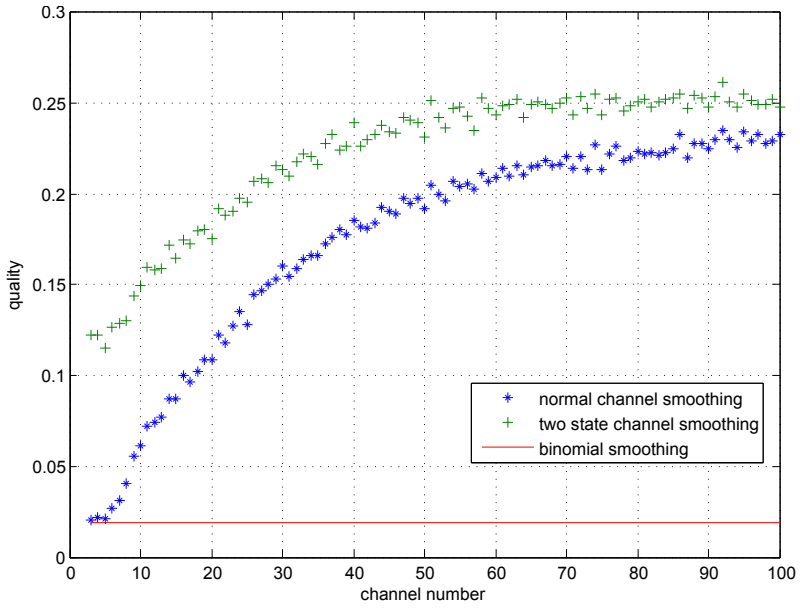


Figure 6.4: Quality measure used to find optimal number of channels.

However, Q does only partially capture the quality of the denoising, because the "ground truth" we have is not a real ground truth. Moreover, Q is somewhat heuristic and the simple average (6.1) can not really measure how good the details of the image have been conserved. Therefore the above analysis is preliminary.

6.2 Synthetic Test Sequences

In the following we will present results on synthetic image sequences, which correspond to our model assumption of a single, translatory motion. The simulated optical imaging is that of a pinhole camera. The range measurement is a perfect radial distance (except for noise we add to the data) to the simulated object surface. The amplitude information follows exactly a power law. We use the sequences for a *proof of concept* of our method. We illuminate its advantages and show what the requirements for its successful application are. We discuss possible extensions and limits.

Tabular Result Scheme We discuss the method at examples that try to catch the typical problematic situation in context of 3D motion estimation. All examples are given in a fixed tabular scheme identical to that of figure 6.5. The content is as follows:

a, b two amplitude image frames of the sequence with frame index 3 and 5, to give an idea of the motion. a contains a smaller subframe showing the range-image (that reveals not that much information).

c,d,e show the errors in the velocity components U,V,W of the estimated motion field. U,V,W are the velocity components in the direction of the Cartesian coordinate axes X,Y,Z. The error is given relative to the single ground truth velocity components. The value range covered is shown in a small colormap on the left of the images.

f, g are the confidence and type measure of the corresponding pixels. The value range is from zero to one.

Parameters the table describes the noise levels of the synthetic data and the parameters of the motion estimation algorithm:

- nl_r and nl_g are the noise level of the range and amplitude data. nl_r is the standard deviation σ_r of the added normal noise in cm. nl_g is σ_a relative to the contrast of the amplitude-signal.

- ps is the pre-smoothing level. The pre-smoothing for range data is a normalized averaging with a binomial applicability of (2ps+1)×(2ps+1) pixels. For the amplitude image standard binomial smoothing with the same mask size was applied. No smoothing in time was done (this typically decreases the quality of the estimate).
- ws is the size of the binomial integration window (ws×ws) of the structure tensor.
- τ is the threshold of the confidence measure (4.30).
- τ_2 is the minimum value that the second smallest eigenvalue must have such that the estimate is assumed to be a full flow. It corresponds to a threshold on the magnitude of the type measure (4.31).
- β is the weighting factor for the amplitude structure tensor equation (4.35), *i.e.* the squared, global weighting factor for the rows of the data matrix of the extended BCCE (4.33).
- w_σ and w_μ are the weights of the gradient based weighting (4.22) (only given if applied)

Error analysis The table shows the density d of the estimated motion field and statistics (mean, standard deviation σ, minimum and maximum) to the errors of the estimate. The density d is defined as the number of pixels for which a full flow could be estimated and which have a confidence of at least 0.5 divided by the number of pixels for which a ground truth flow exists, *i.e.* non moving regions are excluded from the statistic. With the estimated flow vector $\hat{\boldsymbol{f}} = \left[\hat{U}, \hat{V}, \hat{W}\right]^{\mathrm{T}}$ and ground truth flow $\boldsymbol{f} = [U, V, W]^{\mathrm{T}}$ the given error types are

- relative magnitude error: $\left|\hat{\boldsymbol{f}} - \boldsymbol{f}\right| / |\boldsymbol{f}|$
- angular error: $\arccos \frac{\hat{\boldsymbol{f}}_h \cdot \boldsymbol{f}_h}{|\hat{\boldsymbol{f}}_h||\boldsymbol{f}_h|}$, where $\hat{\boldsymbol{f}}_h, \boldsymbol{f}_h$ are the homogeneous flow vectors, *i.e.* the vectors extended for the temporal dimension, $\boldsymbol{f}_h = [U, V, W, 1]^{\mathrm{T}}$ (because the "change" in time is by definition always 1)
- directional error: $\arccos \frac{\hat{\boldsymbol{f}} \cdot \boldsymbol{f}}{|\hat{\boldsymbol{f}}||\boldsymbol{f}_h|}$
- absolute magnitude error: $\left|\hat{\boldsymbol{f}} - \boldsymbol{f}\right|$

- bias of estimate: $(\boldsymbol{e} \cdot \boldsymbol{f})/|\boldsymbol{f}|$, where $\boldsymbol{e} = \hat{\boldsymbol{f}} - \boldsymbol{f}$. Thus this is the projection of the error vector on the ground truth flow vector. It indicates a bias (positive or negative) of the error in direction of the true flow.

Description to the right of the error analysis table there is a short description and discussion of the most relevant features of the example.

Algorithms and Performance Issues All results presented in the following sections use derivative filters optimized for orientation estimation along edges (*i.e.* optimized Sobel filters, see also section 3.1.4 on page 41 and [Sch00]). This is reasonable because motion estimation is basically spatiotemporal orientation estimation, as we have shown before. For the spatial derivatives we use 5×5 filters (*i.e.* no smoothing in time is applied), while we use for temporal derivatives a $3 \times 3 \times 3$ filter, which smoothes within the spatial domain.

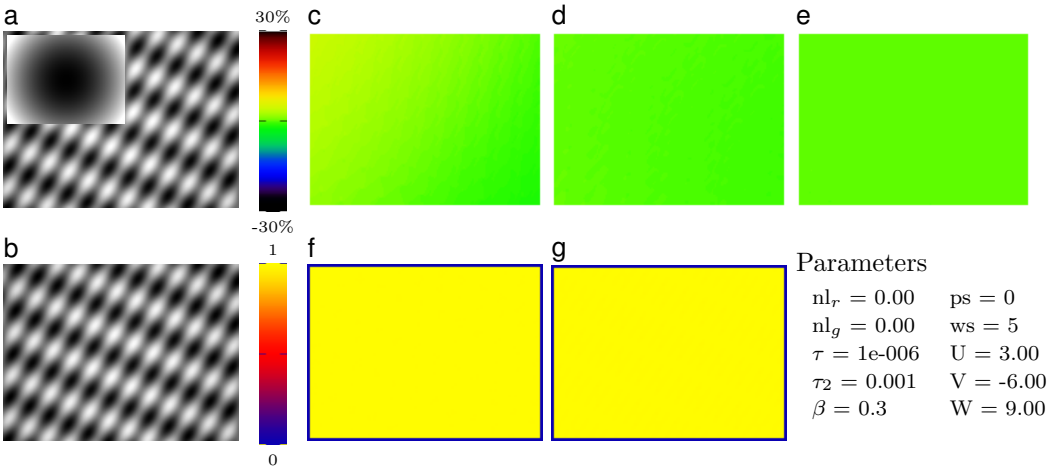
The results were calculated using local TLS-motion-estimation as described in section 4.1.6. The local flow estimates were not regularized (see section 4.1.6.2). While we also used the subspace regularization scheme developed by Spies and Garbe [SG02], we found the improvements in quality compared to the increase in computational costs disappointing. The regularization of a flow field, with a full-flow-density of about 10%, takes about 4–5 times longer than the calculation of the basic flow field. Because regularization is implemented by iterative algorithms and may converge slowly (or not at all) the needed time also depends on the maximum number of iterations allowed (we used 500). Moreover, we found the regularization results to be very sensitive to parametrization and the structure of the processed sequences. This is also the reason, why we do not discuss the results for the estimated plane and line flow. These are only of interest if used within a regularization schema. We achieved reasonable densities in many cases without regularization by choosing the thresholds τ and τ_2 appropriately.

We achieved about 20 flow-field-frames/sec using a single threaded implementation on a 2.4GHz *Intel Core 2* processor. The frames had a size of 160×120. The implementation is not optimized for avoiding unnecessary operations but rather to be as flexible as possible (for large parts of the implementation the high-level image-processing language and library *Heurisko* was used). This is why the author expects a possible increase in frame rate of about a factor 3, only by avoiding unnecessary operations that are trivial to detect. A elaborate analysis of the mathematical structure of the calculations, might reveal further optimization possibilities (the author

thinks that a multigrid (and/or multiscale) implementation could boost computational performance for about one magnitude at least). Most of the used algorithms are standard implementations (like Jacobi-rotations for eigenvalue analysis). Using more efficient, but equivalent algorithms and a multithreadable/parallelized implementation (for multi-core machines) should increase the frame rate significantly.

6.2.1 Motion of a plane

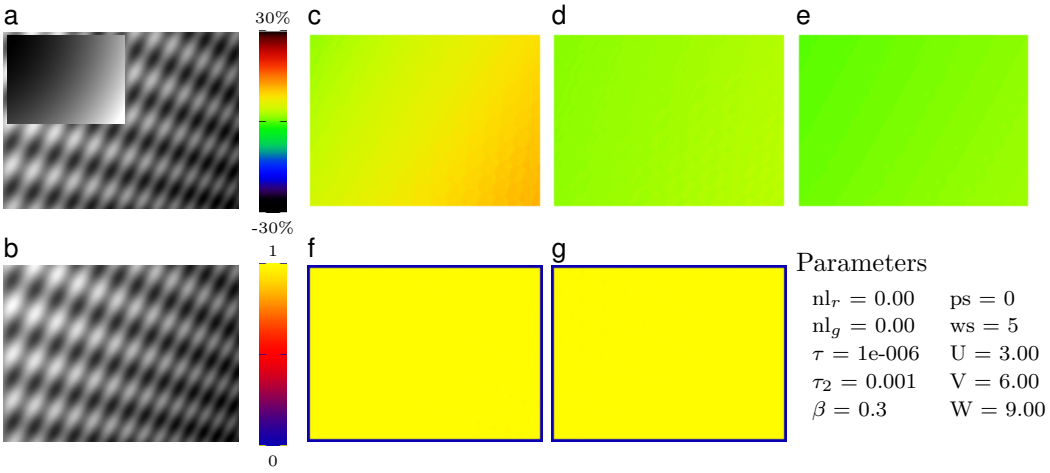
We start on the most basic kind of motion, a (linear) translatory motion of a plane. It is the basis of our motion model, because we approximate the observed surface in motion as conjoined small planar surface patches. Therefore the motion field which our algorithm estimates should be optimal compared to scenes where the surface is curved or has discontinuities; the results are a kind of upper limit in motion field quality for the proposed method.



d=94.3 %	mean	sigma	min	max
relative [%]	0.20	0.04	0.00	0.34
dir'nal [°]	0.21	0.13	0.00	0.62
angular [°]	0.21	0.13	0.00	0.61
magnitude	0.05	0.02	0.01	0.12
bias	-0.01	0.00	-0.02	0.01

This is the most ideal constellation, fitting perfectly to our model assumption. The sampling theorem is completely fulfilled, both in space and time. The only noise is due to (single-precision) floating point calculation roundoff errors. The density of the motion field is maximal (only the border pixels decrease the density). Maximum errors are less than 1% in magnitude and 1° in direction and there is no bias. The plane has a paraboloid-like shape in the range image, because of the radial distance measurement.

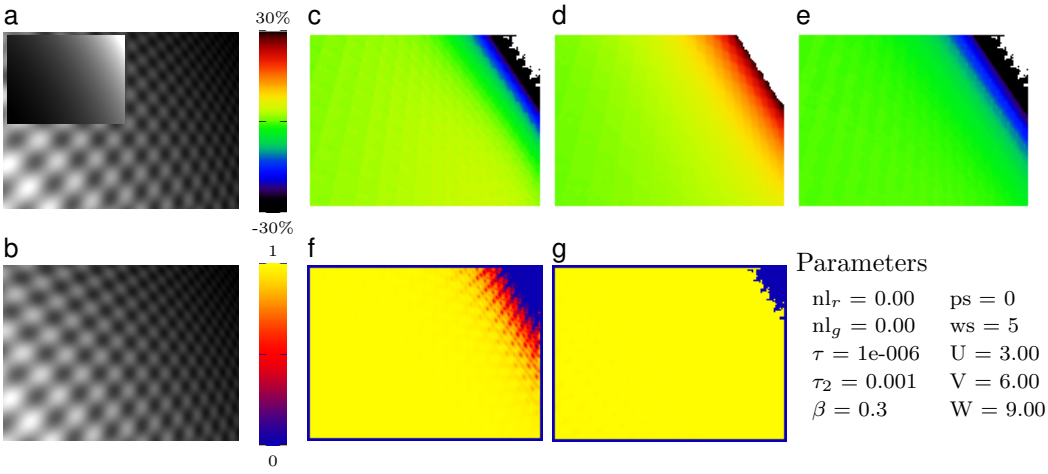
Figure 6.5: Plane perpendicular to optical axis at distance Z=3m. Two superimposed planar wave patterns of different orientation yield a plaid like texture.



d=94.3 %	mean	sigma	min	max
relative [%]	1.71	0.63	0.42	3.15
dir'nal [°]	0.83	0.22	0.39	1.33
angular [°]	0.83	0.22	0.39	1.33
magnitude	0.25	0.08	0.09	0.43
bias	-0.06	0.00	-0.06	-0.02

The plane covers a distance range from 2.6m–3.9m. The approximative character of the derivative filters introduces minor errors for the farther regions were the texture gets smaller, *i.e.* the curvature increases. There is a minor bias toward too small velocities.

Figure 6.6: Plane at same distance but tilted with surface normal having zenith θ and azimuth ϕ of both 30°



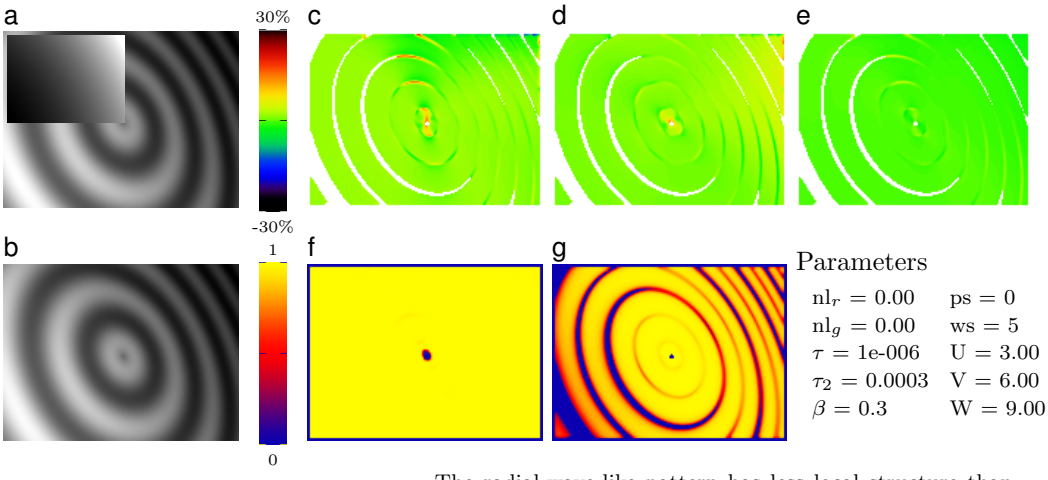
Parameters

$\mathrm{nl}_r = 0.00$	ps = 0
$\mathrm{nl}_g = 0.00$	ws = 5
τ = 1e-006	U = 3.00
$\tau_2 = 0.001$	V = 6.00
$\beta = 0.3$	W = 9.00

d=89.0 %	mean	sigma	min	max
relative [%]	0.82	1.30	0.00	13.72
dir'nal [°]	2.01	2.55	0.20	35.07
angular [°]	2.01	2.55	0.20	34.91
magnitude	0.40	0.50	0.04	6.47
bias	-0.36	0.46	-5.41	-0.02

The flow estimate begins to break down for very far distances (the range is 2m–7m). For far distances the spatial sampling theorem is hardly fulfilled anymore. The filter derivatives introduce large errors and the flow estimate is biased for to small estimates. The density decreases. Notice that increasing the pattern size would improve the results (not shown). The confidence measure excludes successfully the worst region and indicates correctly were the measurements are less trustworthy.

Figure 6.7: The plane is tilted heavy with a surface normal $\theta = 60°, \phi = -30°$



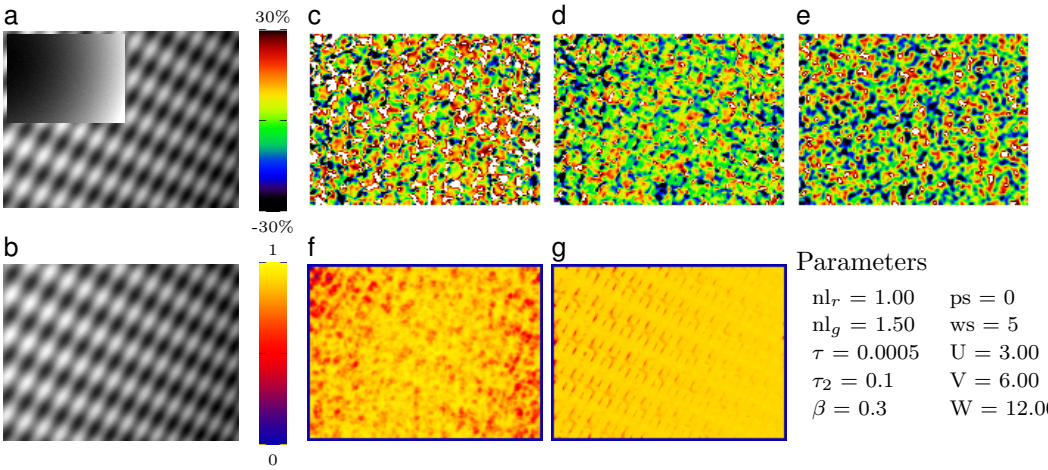
Parameters

$\mathrm{nl}_r = 0.00$	ps = 0
$\mathrm{nl}_g = 0.00$	ws = 5
τ = 1e-006	U = 3.00
$\tau_2 = 0.0003$	V = 6.00
$\beta = 0.3$	W = 9.00

d=86.7 %	mean	sigma	min	max
relative [%]	0.27	0.28	0.00	5.69
dir'nal [°]	0.57	0.35	0.03	4.65
angular [°]	0.57	0.35	0.03	4.64
magnitude	0.12	0.07	0.02	1.06
bias	-0.09	0.07	-0.77	0.34

The radial-wave-like pattern has less local structure than the plaid pattern. Around the contour lines of small (brightness) gradient magnitude, i.e. the crests and troughs of the radial wave pattern, there is an aperture problem, only line flow can be determined. Moreover, the motion estimate using this pattern is more sensitive to noise compared to results for the plaid pattern (not shown).

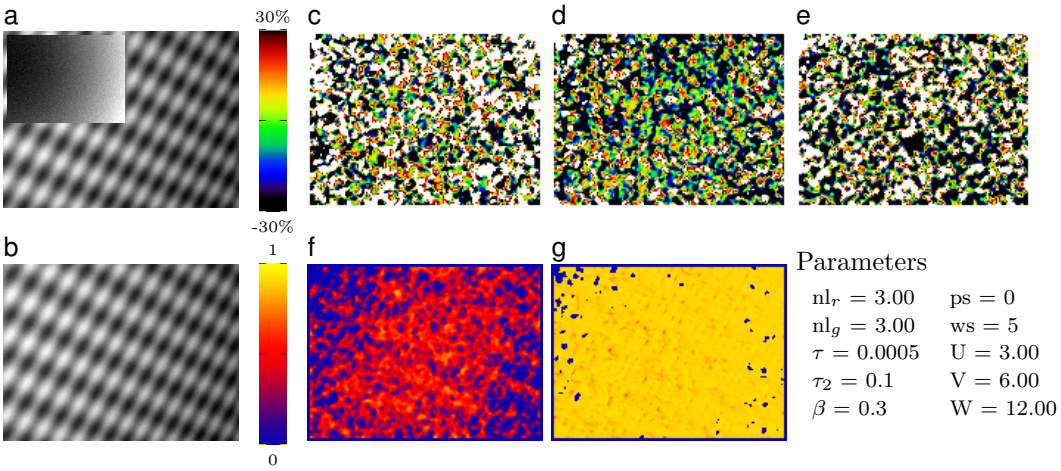
Figure 6.8: Plane with a radial-wave-like pattern tilted for $\theta = 45°, \phi = -30°$



d=93.7 %	mean	sigma	min	max
relative [%]	9.04	6.86	0.00	54.26
dir'nal [°]	4.74	3.34	0.04	36.33
angular [°]	4.75	3.33	0.07	36.29
magnitude	1.79	1.06	0.05	13.00
bias	0.02	1.55	-5.97	5.63

Adding a weak noise to both data channels deteriorates the motion estimate. However, the density is still about maximum. The noise in range of σ_r = 1cm is about the noise of a PMD-sensor under good illumination conditions. Notice that increasing the velocity would improve the relative errors of the estimate. The error decreases approx. inverse proportional to the velocity, until the temporal sampling theorem is violated (then the motion estimate fails).

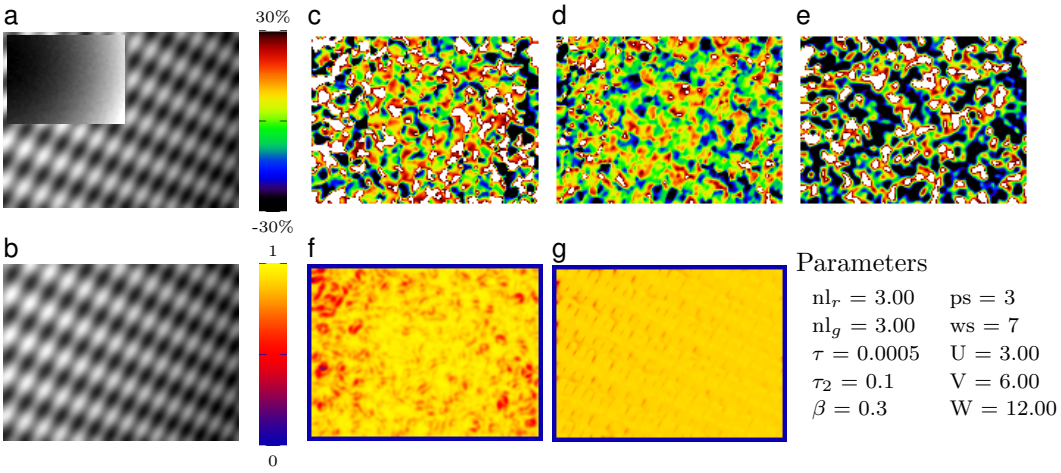
Figure 6.9: Plane ($\theta = 25°, \phi = 15°, r_c = 3.5$m) with weak normal i.i.d. noise added to both amplitude and range data



d=18.7 %	mean	sigma	min	max
relative [%]	23.00	16.97	0.02	105.60
dir'nal [°]	11.31	10.03	0.12	94.22
angular [°]	11.34	9.99	0.31	93.68
magnitude	4.25	2.65	0.25	16.61
bias	0.13	4.04	-14.47	14.26

A high noise level decreases the density of the motion field clearly. The remaining velocity field is roughly of correct direction and magnitude. The confidence measure values indicate that the estimates are not too trustworthy. The type measure correctly indicates a full flow estimate. The estimate can be improved by pre-smoothing the data, as the next figure demonstrates.

Figure 6.10: Same plane as before but noise is increased about a factor 3



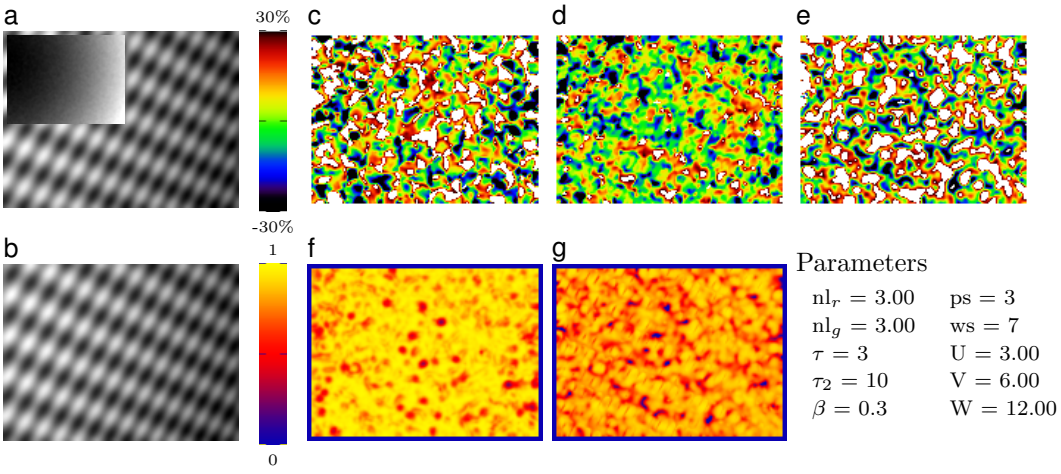
Parameters

$\mathrm{nl}_r = 3.00$	ps = 3
$\mathrm{nl}_g = 3.00$	ws = 7
$\tau = 0.0005$	U = 3.00
$\tau_2 = 0.1$	V = 6.00
$\beta = 0.3$	W = 12.00

d=91.1 %	mean	sigma	min	max
relative [%]	16.71	12.38	0.00	89.15
dir'nal [°]	7.69	6.53	0.03	78.70
angular [°]	7.71	6.52	0.08	78.32
magnitude	3.00	1.98	0.10	14.78
bias	-0.32	2.98	-14.06	13.33

By pre-smoothing the data by normalized averaging with a 7×7 binomial applicability and using a larger integration windows size (of the same size) we recovered nearly maximum density. Both magnitude and directional errors are improved. While maximum errors are still high the width of the error distribution was improved.

Figure 6.11: Same plane and noise level as before but pre-smoothed



Parameters

$\mathrm{nl}_r = 3.00$	ps = 3
$\mathrm{nl}_g = 3.00$	ws = 7
$\tau = 3$	U = 3.00
$\tau_2 = 10$	V = 6.00
$\beta = 0.3$	W = 12.00

d=90.4 %	mean	sigma	min	max
relative [%]	16.85	13.97	0.00	129.36
dir'nal [°]	5.68	3.67	0.00	32.69
angular [°]	5.71	3.66	0.11	32.72
magnitude	2.85	1.98	0.03	21.30
bias	1.23	2.52	-5.88	15.31

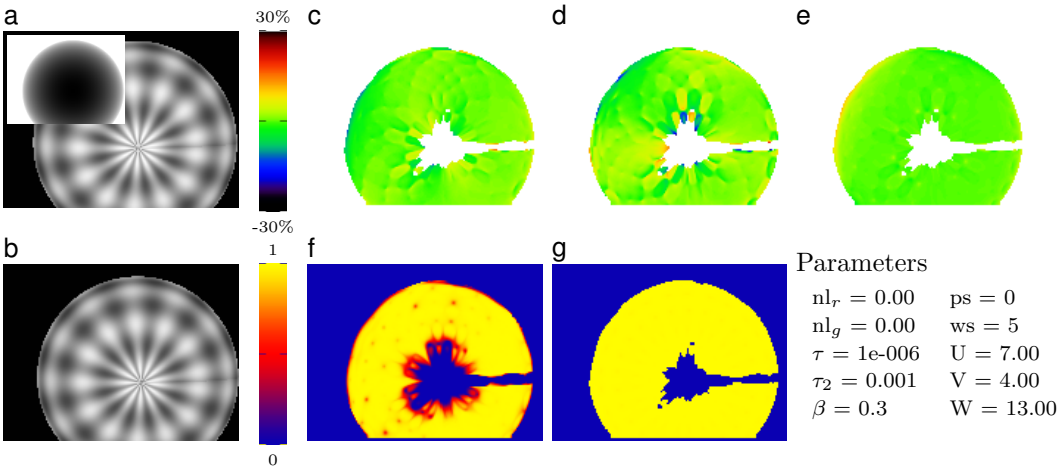
Here the same pre-smoothing and integration window size was used, but the structure tensor was equilibrated. While the error in magnitude has not changed significantly the directional errors were improved clearly (particularly σ is much smaller). However, the equilibration is strangely more biased then the not equilibrated estimate. Because of the equilibration the thresholds (τ, τ_2 for confidence and type measure) need to be adapted.

Figure 6.12: Flow estimation using an equilibrated structure tensor

The presented results for a plane at pure translatory motion are in good agreement with our expectations. The method is somewhat sensitive to noise. For weak noise the results are still good. For medium level noise the results are acceptable if a pre-smoothing filter is applied that is about the same size as the integration window (here 7×7). Most important for a satisfactory result is that the sampling theorem is not violated. We also saw that if there is not enough structure in the neighborhood a full flow can not be estimated and the density of the motion field decreases.

6.2.2 Motion of a sphere

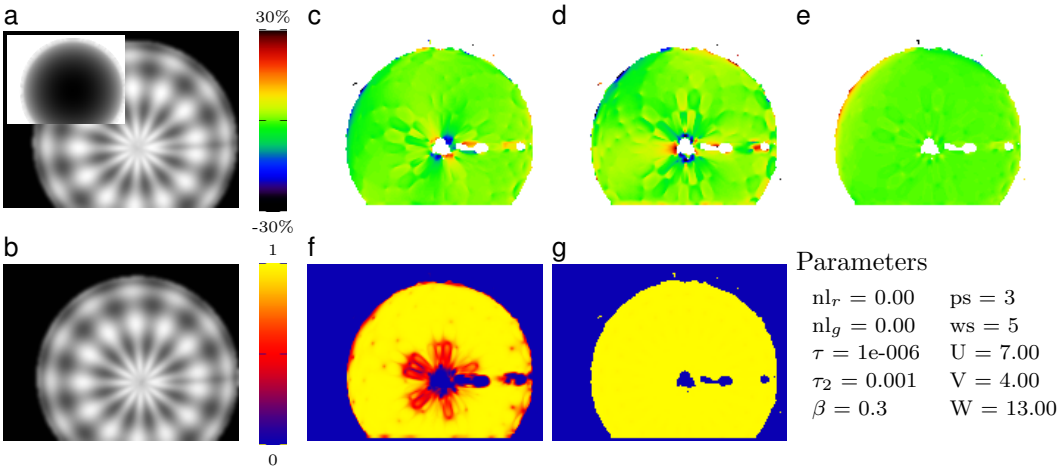
We now present results for a sphere in motion. This introduces additional problems compared to the plane motion, because a sphere has a curved surface and is of limited extension. There are motion boundaries present that are not explicitly modeled. The texture on the surface is of varying curvature and has discontinuities that are in conflict with the spatial sampling theorem, which is rather problematic for the employed derivative filters.



d=70.9 %	mean	sigma	min	max
relative [%]	0.36	0.46	0.00	4.69
dir'nal [°]	0.47	0.39	0.00	3.94
angular [°]	0.47	0.38	0.02	3.93
magnitude	0.14	0.12	0.00	1.21
bias	-0.01	0.15	-0.53	1.20

The errors are comparable to those of the noise-free plane of similar texture. But the density of the motion field is decreased because in the center and on the central-right side of the sphere the discontinuities in the texture violate the spatiotemporal sampling theorem. Also at the edge of the sphere no motion estimation is possible due to the motion boundary. The size of missing border is about the size of the integration window.

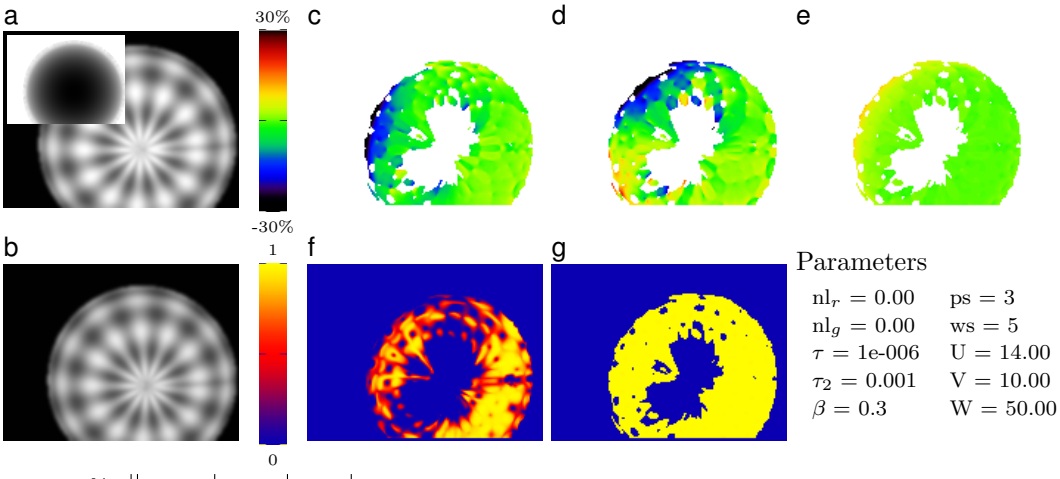
Figure 6.13: A sphere with a radius of 50cm and at a distance of 2.2m having a sinusoidal plaid texture. Motion boundaries and the violation of sampling theorem decrease the density of the motion field.



d=77.8 %	mean	sigma	min	max
relative [%]	0.41	0.58	0.00	5.58
dir'nal [°]	0.52	0.48	0.00	5.36
angular [°]	0.52	0.48	0.00	5.36
magnitude	0.16	0.15	0.01	1.70
bias	-0.00	0.17	-1.05	1.69

The pre-smoothing removes the very fine structures and discontinuities. Therefore the sampling theorem is no longer violated and the density of the motion field is increased.

Figure 6.14: Same sphere as above but with pre-smoothing of range and intensity data.



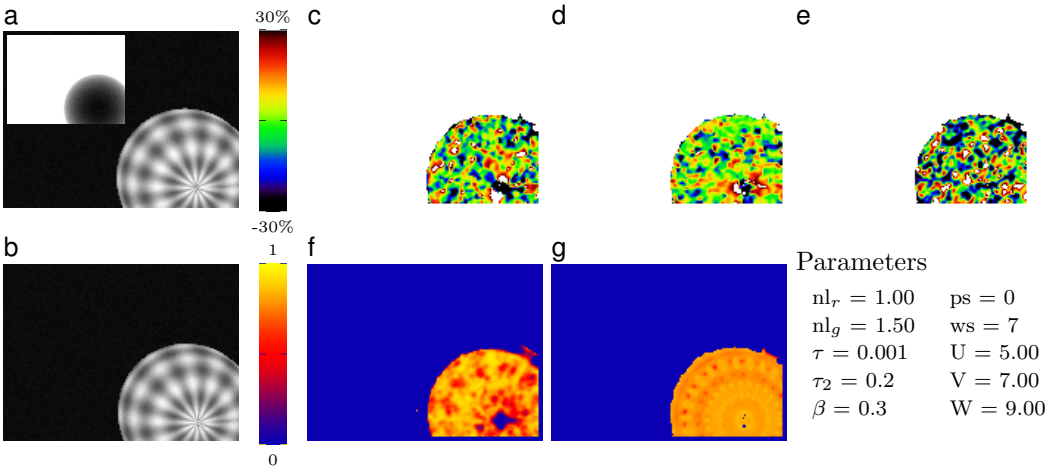
Parameters

$\mathrm{nl}_r = 0.00$	ps = 3
$\mathrm{nl}_g = 0.00$	ws = 5
τ = 1e-006	U = 14.00
$\tau_2 = 0.001$	V = 10.00
$\beta = 0.3$	W = 50.00

d=32.5 %	mean	sigma	min	max
relative [%]	0.49	0.55	0.00	3.00
dir'nal [°]	0.72	0.73	0.00	3.48
angular [°]	0.72	0.73	0.00	3.48
magnitude	0.73	0.72	0.04	3.56
bias	0.33	0.68	-0.48	3.04

Increasing the velocity leads around fine structured regions to a violation of the sampling theorem in time. The motion field density decreases, but the estimated velocity vectors are still very good, *i.e.* the relative and directional errors show no significant difference to the results of the sphere with lower speed.

Figure 6.15: Same sphere but with increased velocity of the sphere. The motion estimate breaks down around the fine textured structures



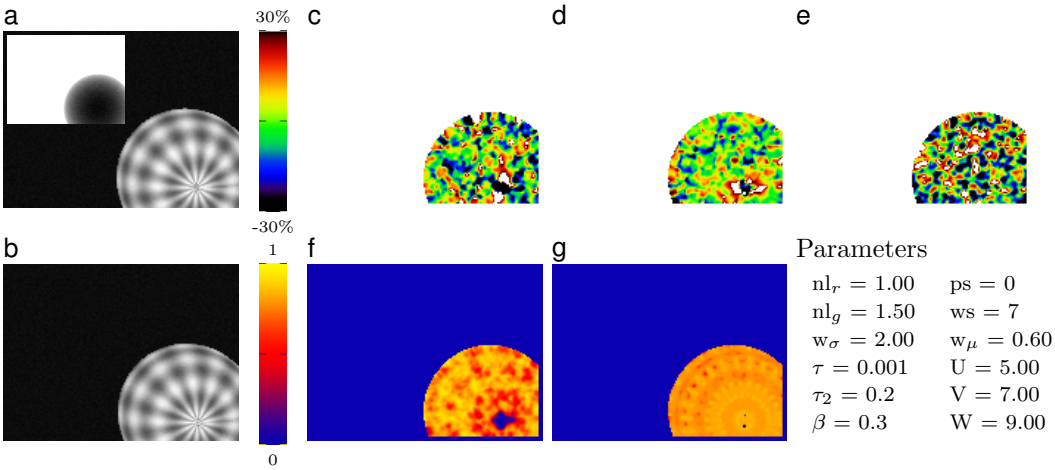
Parameters

$\mathrm{nl}_r = 1.00$	ps = 0
$\mathrm{nl}_g = 1.50$	ws = 7
$\tau = 0.001$	U = 5.00
$\tau_2 = 0.2$	V = 7.00
$\beta = 0.3$	W = 9.00

d=73.8 %	mean	sigma	min	max
relative [%]	8.46	6.58	0.01	43.64
dir'nal [°]	5.78	4.08	0.08	33.68
angular [°]	5.79	4.06	0.15	33.57
magnitude	1.75	1.01	0.05	8.09
bias	-0.15	1.34	-5.50	6.37

The integration window size was increased to 7×7, to compensate for the increased noise. The errors in the motion estimate are of the same magnitude as those for the plane subject to noise of the same (weak) noise level. Only the density is smaller. No pre-smoothing was applied. Thus we find in the center of the sphere an unestimated region.

Figure 6.16: The same sphere at a farther distance subject to weak noise, with no pre-smoothing applied



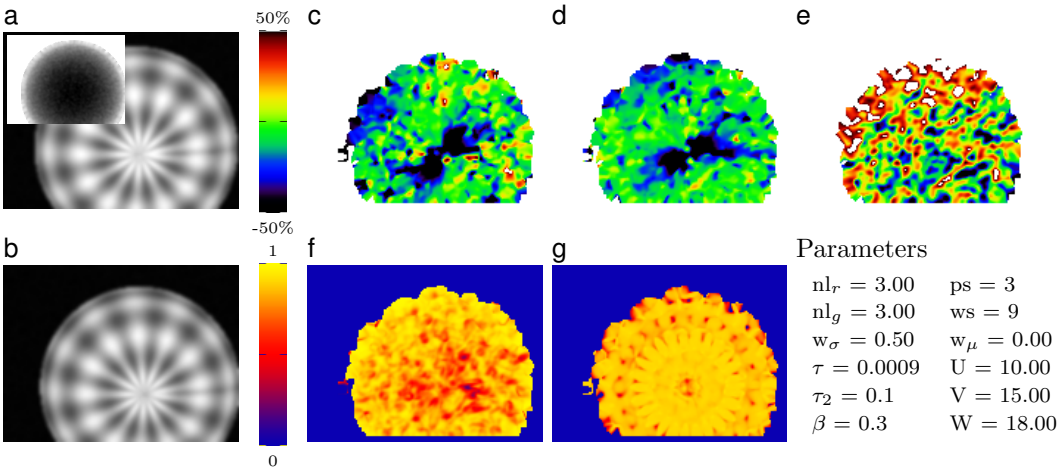
Parameters

$\mathrm{nl}_r = 1.00$	ps = 0
$\mathrm{nl}_g = 1.50$	ws = 7
$\mathrm{w}_\sigma = 2.00$	$\mathrm{w}_\mu = 0.60$
$\tau = 0.001$	U = 5.00
$\tau_2 = 0.2$	V = 7.00
$\beta = 0.3$	W = 9.00

d=81.3 %	mean	sigma	min	max
relative [%]	9.06	6.77	0.00	53.97
dir'nal [°]	6.16	3.90	0.09	36.56
angular [°]	6.17	3.88	0.09	36.43
magnitude	1.85	0.99	0.08	7.77
bias	-0.09	1.35	-5.53	4.89

Here we demonstrate how the proposed gradient based weighting of the data terms integrated by the structure tensor can increase the density of the estimate without reducing the quality of the estimate. The errors are comparable to the prior example but the density is increased.

Figure 6.17: Identical to the prior example but with gradient based weighting applied. The density of the estimated motion field is increased



Parameters

$\mathrm{nl}_r = 3.00$	ps = 3
$\mathrm{nl}_g = 3.00$	ws = 9
$\mathrm{w}_\sigma = 0.50$	$\mathrm{w}_\mu = 0.00$
$\tau = 0.0009$	U = 10.00
$\tau_2 = 0.1$	V = 15.00
$\beta = 0.3$	W = 18.00

d=81.0 %	mean	sigma	min	max
relative [%]	13.50	10.64	0.00	115.92
dir'nal [°]	9.53	6.97	0.07	53.38
angular [°]	9.53	6.97	0.09	53.32
magnitude	5.78	3.80	0.18	38.67
bias	3.12	4.28	-10.59	33.47

Here we increased the integration window size to 9×9 and applied pre-smoothing. Gradient based weighting improves the density here too, at a small loss of quality in the estimate (2% in relative magnitude, insignificant in angular error) compared to the binomial weighted structure tensor (not shown).

Figure 6.18: Sphere corrupted by noise of same level as used in figures 6.10 and 6.12. Pre-smoothing and gradient based weighting yields an acceptable quality at a high density of the motion field.

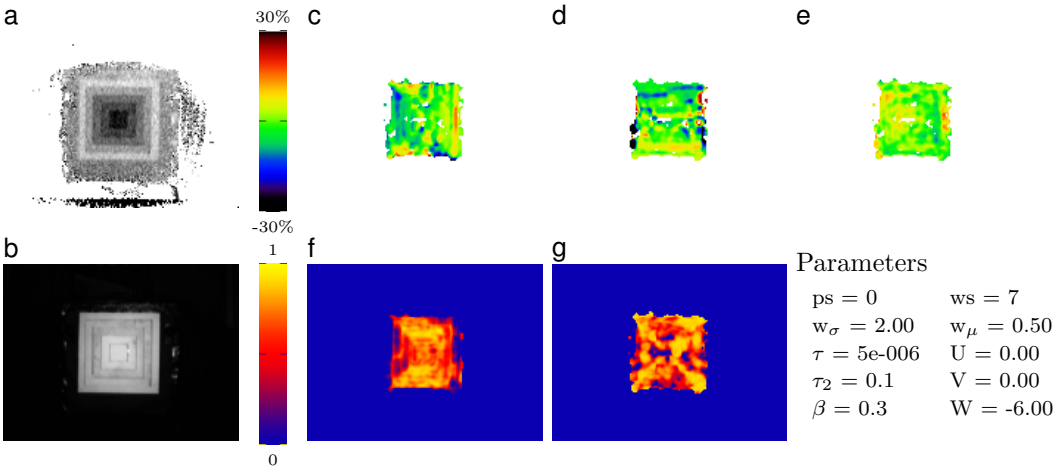
The estimated motion fields for the sphere we presented are of similar quality as those for the plane motion. Motion boundaries do not increase the errors in the velocity field but are excluded by the confidence measure and therefore reduce the density of the motion field. Gradient based weighting can increase the density of the motion field, at an only small loss of quality.

Its important to notice that the ratio of density to error of the motion field is not fixed but depends on the chosen parameters. Particularly the thresholds τ and τ_2 on confidence and type measure regulate if an estimate is a full flow or not. Typically increasing τ increases the density at a cost of quality. And the right choice of the threshold depends on the noise level of the signal, which may vary over the image. While weighting the input data during pre-smoothing according to a confidence measure (as discussed in section 3.1.5) is a first step, an automatic adaption to the noise level would be desirable. However, the author is not sure how this could be done best. Some type of analysis of the kind of a Wiener-filter may be appropriate, but a detailed analysis of this topic is outstanding.

6.3 Real World Sequences

After the proof of concept by simulated data, we demonstrate the performance of the motion estimation algorithm on real world sequences. We used sequences of the pyramid target moving in horizontal direction and parallel to the optical axis. The input data are *mean images* (from step mode acquisition described in section 5.2.2) that have a lower noise level and show no motion artifacts compared to live sequences. The results are presented in the same tabular scheme as the synthetic sequences. The meaning of the error colormaps has changed slightly. While in the previous figures it has been the percental error w.r.t. the single ground-truth velocity components U, V and W, its now the percental error w.r.t. the ground-truth velocity magnitude, *i.e.* all three error images have the same scale. The definition of the density d has changed slightly too (because we have no ground truth velocity field of the hole scene, but only the ground truth velocity of the target). It is defined by the number of pixels with a full flow estimate and a confidence of at least 0.5 divided by the number of pixels that have an amplitude higher than a specific threshold (which is 20 for all pyramid sequences).

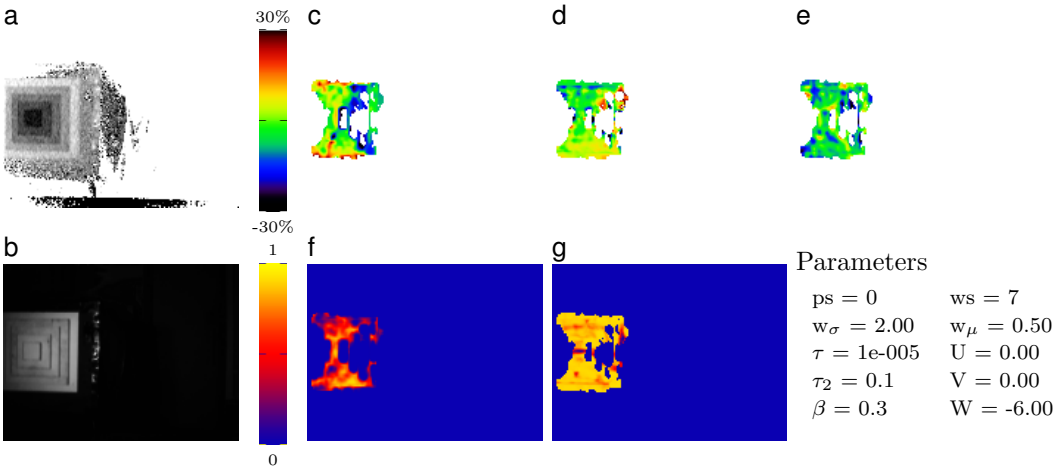
All motion fields were calculated using a fixed scaling exponent $a = 3$ (see equation (4.32)). This is a sane compromise for the distance range from 1.8–4.8m, present in the used sequences, w.r.t. the varying and not fully understood dependence of a on the distance as depicted in figure 5.13 and discussed in section 5.2.3.



d=45.9 %	mean	sigma	min	max
relative [%]	2.92	2.18	0.00	11.30
dir'nal [°]	4.15	2.32	0.12	19.02
angular [°]	4.11	2.28	0.24	18.79
magnitude	0.49	0.24	0.03	2.10
bias	-0.01	0.22	-0.60	0.67

The pyramid target performs a pure translation in depth. The errors are in the magnitude of the spatial resolution of the sensor. This is not trivial, as the range sequence appears divergent. The estimate is not biased for this frame, but generally there is a bias for uncalibrated camera data that corresponds to the periodic phase errors show in figure 5.6.

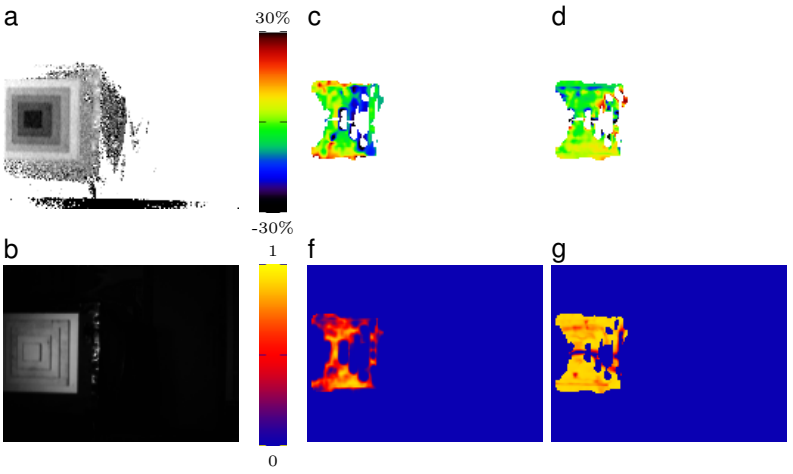
Figure 6.19: Translation of the pyramid target in Z-direction at central position. The density and quality of the motion field are good.



d=18.8 %	mean	sigma	min	max
relative [%]	6.64	4.21	0.03	26.67
dir'nal [°]	3.88	2.10	0.04	13.80
angular [°]	3.91	2.02	0.52	13.62
magnitude	0.62	0.25	0.07	1.68
bias	0.36	0.28	-0.33	1.58

The target is no longer in a central position. The confidence of the left and right triangle of the pyramid is about zero. While in the central position (of the prior example) the triangles have a reduced confidence too, its much more articulate in the border positions. The estimate errors are only increased in magnitude but not directional.

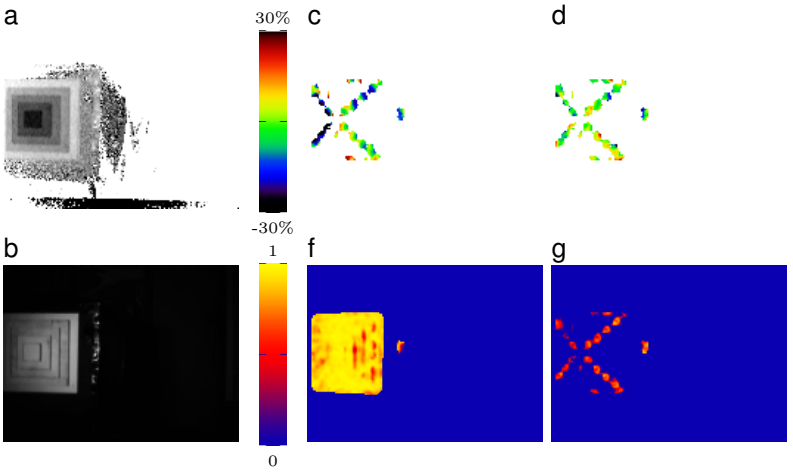
Figure 6.20: Translation of the pyramid target in Z-direction at left position. The density of the estimate is clearly reduced.



d=19.8 %	mean	sigma	min	max
relative [%]	6.32	4.15	0.01	24.86
dir'nal [°]	3.98	2.71	0.02	33.32
angular [°]	4.00	2.65	0.30	32.99
magnitude	0.61	0.32	0.04	4.10
bias	0.35	0.25	-0.25	1.47

The fixed pattern noise was removed by calibration. While the range and amplitude data looks much better, the motion estimate is not significantly improved. Sigma of the angular error got even worse. Only the magnitude error shows a minor improvement.

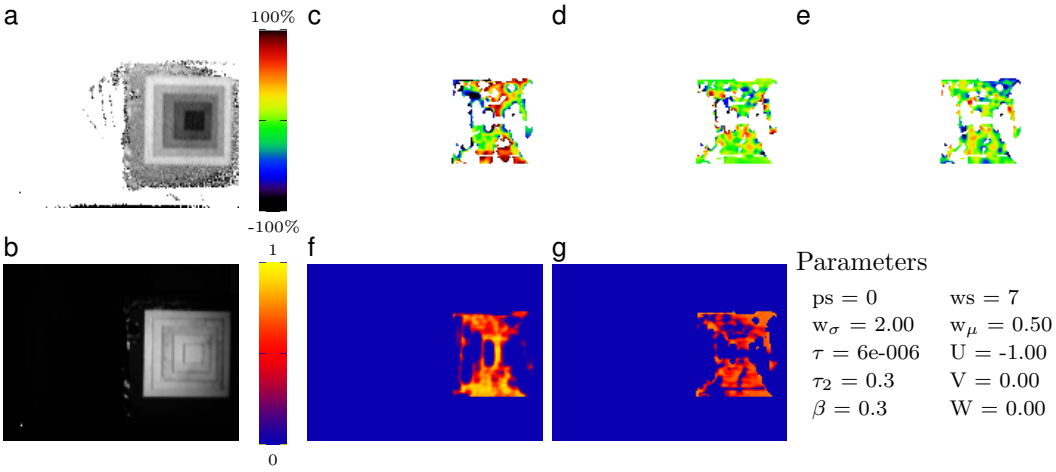
Figure 6.21: Same as before but fixed pattern noise was removed. Interestingly the estimate is not improved.



d=18.1 %	mean	sigma	min	max
relative [%]	9.96	5.90	0.16	37.29
dir'nal [°]	7.75	5.20	0.15	26.48
angular [°]	7.73	5.09	0.61	26.17
magnitude	1.10	0.56	0.19	3.00
bias	0.51	0.38	-0.59	2.12

Here we demonstrate that a motion estimation fails in determining a full flow, if it does not take advantage of the amplitude information using the extended BCCE (4.33). While the confidence in the estimate is increased, only line or plane flows can be estimated (not shown). The errors are clearly increased.

Figure 6.22: No amplitude information for the motion estimate was used ($\beta = 0$). The aperture problem allows only an estimate at the diagonal edges of the pyramid.



d=20.3 %	mean	sigma	min	max
relative [%]	47.65	29.47	0.08	181.15
dir'nal [°]	41.90	41.83	0.41	175.36
angular [°]	28.75	16.80	1.11	92.92
magnitude	0.70	0.37	0.04	2.18
bias	0.49	0.53	-1.77	2.11

A horizontal translation of the pyramid target is rather problematic for the motion estimation. Besides missing information in the left and right triangle, there is a clear bias in the estimate of the upper and lower triangle. The bias contributes to about 70% of the magnitude error. The angular error is large. Pre-smoothing the data will not help to solve the problem. Dependent on the chosen threshold it will either decrease the density or increase the errors.

Figure 6.23: Horizontal translation of the pyramid target. There is a strong bias in the motion estimate.

The motion estimates for the pyramid sequences 6.19 to 6.23 show rather heterogeneous results. While pure motions in Z-direction could be estimated well, the results for the estimates in X-direction are rather bad. The problem is not due to noise in the data as this is really low. Interestingly removing the fixed pattern noise increased the standard deviation in the angular errors! The author thinks that the pyramid target is a quite difficult object for motion estimation. While it looks rather optimal, as it has structure both in range and amplitude information, a second look reveals that the texture information is of a kind that is suboptimal for our algorithm. The texture in the amplitude is due to *shadows* and varying angles of reflectance. This kind of texture is not modeled as we assume *spatially* but not *temporally* varying reflectivity on the object surface, *i.e.* the reflectivity texture on the surface is constant in time. Moreover the step edges in the range data of the pyramid are hard to handle properly by the derivative filters. At each edge there is a discontinuity and on each step we find an aperture problem. This might be addressed by pre-smoothing, but it seems that the integration window size is to small to come to an unique estimate. Possibly a multiscale approach could help, but we are not sure about this. An open question in this context is weather the simple pinhole camera model introduces significant errors. Therefore a geometric calibration of the camera is desirable.

Figure 6.22 showed the benefit of taking both information channels of the PMD-signal into account. If only the range information is used the errors in magnitude at similar density increased about 50% in magnitude and 100% in direction.

We remarked with figure 6.19, that the motion estimates are generally biased if the camera is not properly calibrated in range, with a similar method as exemplified in section 5.2.2. Figure 6.24 shows the motion estimation results for an uncalibrated sequence of the checkerboard target performing a motion along the optical axis. Besides the decreased density due to the low reflectivity patches (pixels of low am-

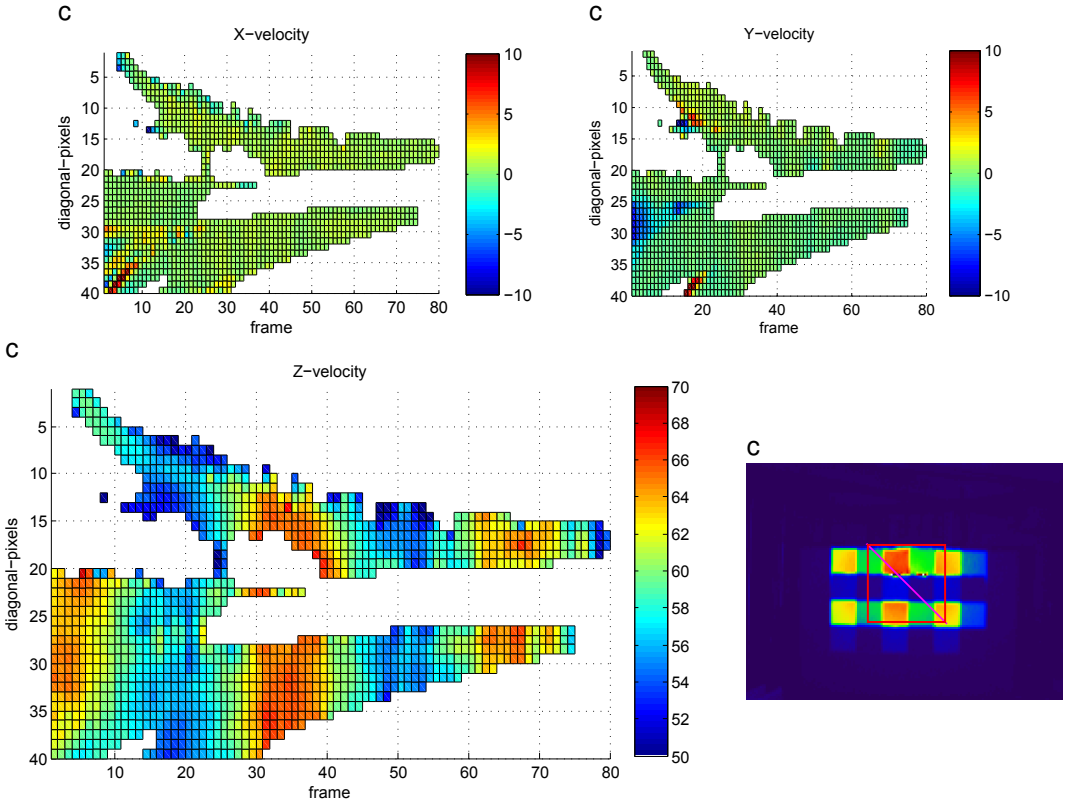
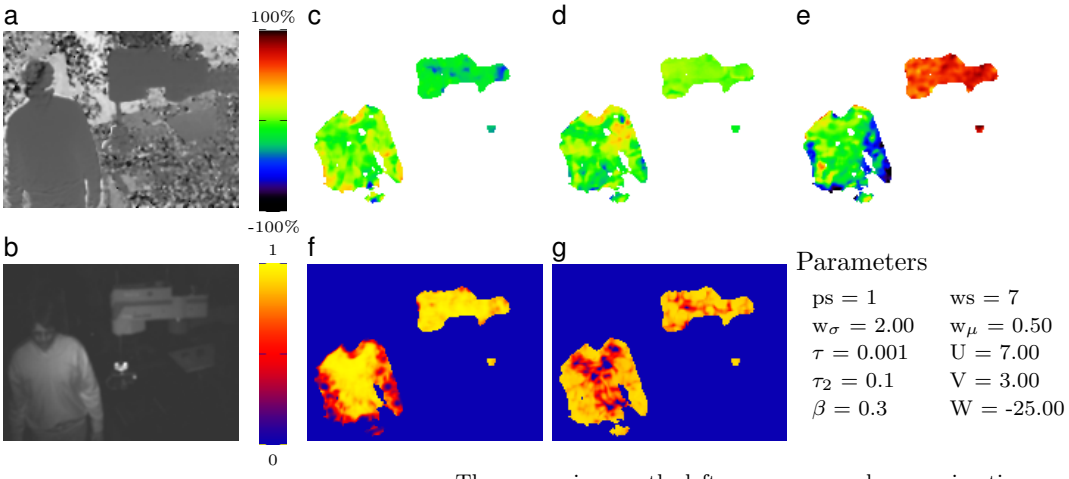


Figure 6.24: Checkerboard target translation of 60mm/frame in Z-direction from 1.2m–6.0m: a, b and c velocity components U, V and W [mm/frame] at the pixel positions along the (magenta) diagonal, shown in the amplitude image d (at 2.2m). The variation of W with the distance corresponds to the periodic phase error of the camera.

plitude are excluded from the motion field), we see the bias of the motion estimate that varies with the range in correspondence to the measurements 5.6 and 5.10.

The results for the pyramid target are not satisfactory. Therefore, we have a look at a real world sequence in a more natural setup which is depicted in figure 6.25. While there is no ground truth motion data for these kind of sequences we can at least check if the results are consistent with the apparent motion. We used the average



d=53.3 %	mean	sigma	min	max
relative [%]	35.26	25.73	0.05	99.14
dir'nal [°]	13.38	9.07	0.20	52.26
angular [°]	13.56	9.35	0.33	78.63
magnitude	10.88	6.28	0.22	26.54
bias	-3.73	9.58	-21.57	22.08

The errors given on the left are a very rough approximation. We took the approximate mean velocity of the person and used it as ground truth for the sequence. Because the robot moves much slower than the person, it is of red color in the W-error-map (e). We find the motion field to be consistent with the apparent motion. Moreover, it has a high density (except for the missing head and the motion boundaries).

Figure 6.25: Real world sequence with a person moving toward the camera and a robot that moves in the same direction but much slower.

velocity of the moving person calculated from the motion field as "ground truth". Because there are at least two major motions in the sequence the mean errors are not too meaningful. More interesting are the standard deviations as they should be an upper limit of the true standard deviation because we falsely treat two motions as one w.r.t. the error statistic. Nevertheless sigma is relatively small which indicates that our estimate is consistent w.r.t. the apparent major motions. The similar color in the velocity components of each of the two moving objects indicates this too.

6.4 Summary

We have shown that the proposed local motion estimation method yields correct motion fields for synthetic range sequences that show translatory motion. Weak to medium level i.i.d. noise in the range and amplitude data can be compensated by an appropriate pre-smoothing if necessary at all. The method is numerically stable, *i.e.* (small) errors in the input are not magnified in the output. Therefore the quality of the motion field might be reduced due to noise, but only relative to the level of noise introduced.

The method however is sensitive to a correct spatiotemporal sampling of the input data. Motions large compared to the spatial frequency of the range or amplitude data (*i.e.* its range-structure or reflectivity-texture) can not be handled directly, as they violate the temporal sampling theorem. An extension of the method to a multiscale implementation similar to the coarse-to-fine strategy developed by Black and Anandan [BA96] would be necessary to overcome this problem. More advanced techniques including a regularization of the motion field as introduced by Bruhn, Weickert, and Schnoerr [BWS05] seem applicable too. Very fine (spatial) textures are problematic w.r.t. the employed derivative filters, independent of the magnitude of the motion, and need to be addressed by pre-smoothing the data.

The results for the real world sequences are heterogeneous. Most likely due to the discussed problematic nature of the pyramid target w.r.t. our model assumptions, we could not yield satisfactory results for a translation in horizontal direction, while the results for a motion in depth are in very good agreement with the true motion field. Analysis of a motion sequence in a natural setup showed the consistency of the estimated motion field with the apparent motion. The density of the locally determined motion field is pleasantly high.

The aperture problem was successfully addressed by using both range *and* amplitude signal of the PMD-sensor. Combining both data channels yields for most cases a higher quality and density of the determined motion field.

Chapter 7

Conclusion and Outlook

In the previous chapters we have presented and discussed methods from a large number of research fields, with the final goal to extend the possibilities of image processing w.r.t. the analysis of dynamic processes captured in image sequences. In particular we want to estimate correct, physical, three dimensional motion fields. The image sequences of interest are acquired with range cameras based on the TOF distance measuring principle realized in PMD-technology.

In the following we give a summary of the major topics we worked up and how well we met our goal. The author states his personal evaluation of PMD-technology w.r.t. motion estimation and highlights important topics that should be addressed regarding the sensor technology as well as algorithmic aspects of motion estimation.

7.1 Summary

To optimally exploit the extended information content that a PMD-camera features, we need to understand where this information originates from. Therefore we analyzed the basic measuring principle and showed possible sources of errors that can be avoided, if specific details about the technical realization are know; particularly the blindfold application of the common formula (2.12), which is actually valid only for sinusoidal modulation (and even harmonics of higher order), can lead to articulate systematic errors if the exact modulation of the emitted (infrared) light is more rectangular than sinusoidal. We presented formula (2.14) to calculate the correct range and amplitude for a rectangular modulated optical signal. Actually, the producers of the cameras should be aware of the problem and correct or better avoid them. However, for current camera models this is not the case yet.

We have shown a new way to correct for one of the most prominent systematic errors by means of a calibration based on Fourier approximation. The correction of the error, after calibration has been done, can be calculated easily and from very few data. Therefore an implementation of the correction algorithm directly on the camera should be unproblematic.

We also discussed several further errors of the sensor and derived, based on the statistical errors, an uncertainty measure for the acquired range data, which can be used in further image processing steps. Thus, we might increase the accuracy of results or at least come to a better estimate about the magnitude of the possible errors, *i.e.* a confidence measure. We successfully used the uncertainty measure within a novel extension to B-spline channel smoothing, that we named *two state smoothing*.

We introduced the necessary theoretical fundament for motion estimation from image sequences. We use the *range flow constraint equation* in a novel formulation that embeds the used pinhole camera model analytically and reduces the number of necessary filter operations. To improve motion estimation results w.r.t. the unavoidable aperture problem, we extended the *brightness change constraint equation* for the PMD-camera, assuming the irradiance to follow a (inverse) power law. Thus we can take both BCCE and RFCE into account and use both in a combined structure tensor approach, that is computational efficient.

On synthetic and real world data we applied our method successfully, achieving rather dense motion fields without the need for regularization. However, the method has still weaknesses that need to be addressed: the algorithm fails to estimate a satisfactory motion field under specific conditions for which, within the limits of the method, an estimate should be possible.

Within the project Lynkeus, funded by the *Federal Ministry of Education and Research* (BMBF), we implemented an interface to our algorithm, that allows it to be used within a complex runtime environment. Figure 7.1 shows the range flow estimation module within a simple configuration of the Lynkeus runtime environment (RTE)*.

Furthermore, we developed a method for motion artifact detection and utilized it in conjunction with two state channel smoothing for the successful realization of

*the Lynkeus RTE is developed by the collaboration partner *Elektrobit*

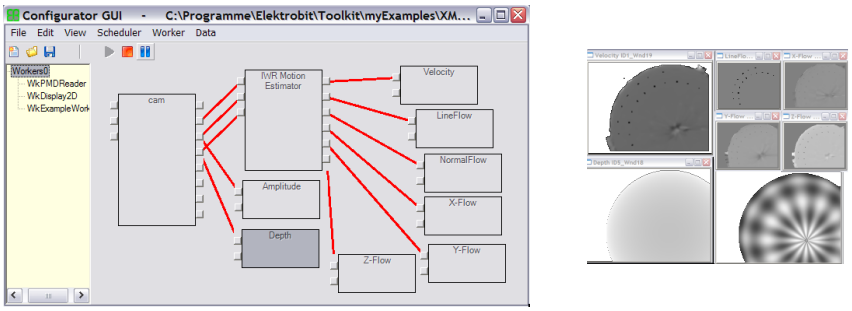


Figure 7.1: A prototype of the range flow module developed for the LYNKEUS runtime environment.

a demonstrator within the project SMARTVISION. Figure 7.2 shows the GUI to our software and the associated experimental setup (from the collaboration partner *Schmersal*), which demonstrates the application of PMD technology in the field of safety engineering.

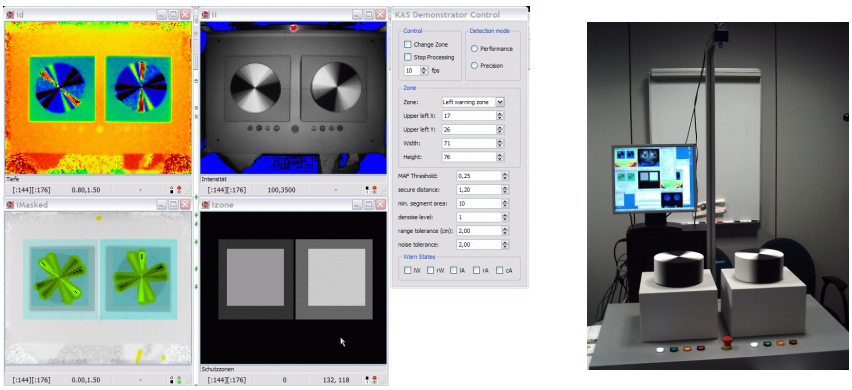


Figure 7.2: GUI and experimental setup, demonstrating the application of PMD technology in the field of safety engineering.

7.2 Evaluation and Outlook

First of all the author would like to state that he thinks the PMD technology to be a promising new method to acquire quickly and with rather small effort range maps of the close surrounding. The ease of use and installation may be one of the major advantages of this technology.

However, the method still has several limitations that hopefully will be addressed in near future. The most important from the author's point of view are:

framerate the current framerates of maximal 20 frames/sec are rather problematic for high accuracy motion estimates that our method aims at.

dynamic range due to the dependence of pixel irradiance on object distance and of range measurement accuracy on irradiance and the limited capacitance of the pixels' storage sites (see section 2.2.2.2), the dynamic range of a PMD-sensor is critical w.r.t. its applicability in real world tasks. Techniques like multiple exposure and adaptive illumination should be exploited to overcome shortcomings of current sensor models w.r.t. this topic.

systematic errors a better calibration of the camera is possible and should be done on producers side. While some problems like the interdependence of range and amplitude/reflectivity are not well understood yet, there are still several errors that can be corrected easily.

motion artifacts the problem of motion artifacts should be addressed technically by taking the necessary correlation samples at the same time. While prototypic PMD's of this kind were realized as 4-tap lock-in pixels by Lange [Lan00], their realization seems to contradict other optimization criteria of the CMOS-process, like for instance the fill factor. However, for the sake of a simple and robust analysis of dynamic image data, the problem should be addressed nevertheless.

objective the optical elements used for some camera models seem to be partially suboptimal. Scattered light can influence the range measurements very badly.

temperature the range measurement is not stable over time, but depends clearly on the temperature of the sensor (however this problem is more serious for the SR3000, than for O3D and *PMD19k*).

documentation a better technical documentation on specific features of the camera would be of great benefit, particularly regarding demodulation contrast and modulation of the light source.

With respect to our own algorithms there is room for various extension and improvements:

- So far, the thresholds necessary for the eigenvalue analysis of the structure tensor (and hence for motion estimation) need to be tuned manually. While the same thresholds can be used for various kinds of sequences, still the need to adapt for local and global noise levels exists.

  A detailed analysis of the novel constraint equations w.r.t. error propagation and particularly equilibration is standing out. An automatic adaption of the thresholds based on this analysis seems possible, more particularly as we consider the uncertainty measure that can be derived from the PMD-signal. An additional local analysis in the kind of a Wiener filter might be helpful for noise estimation; also a local analysis of the variation of the eigenvalues in the structure tensor might be appropriate, but computational costly.

- We used the subspace regularization scheme presented in [SG02], but found the improvements in quality compared to the increase in computational costs disappointing. Meanwhile there exist more advanced concepts of global methods incorporating local estimates. An extension of our method towards the concepts and techniques presented in [BWS05; Pap+06; Bru+06] seems attractive.

  This would allow the estimation of dense flow fields in the presence of locally extended aperture problems and in situations where the temporal sampling theorem is partially violated. At the moment our method is limited to rather "well behaving" image data, *i.e.* data that complies to the sampling theorem spatially and temporally. With current limitations of the cameras in frame rate and resolution and the given (rather high) noise level such sequences can only be acquired for a limited number of real world applications.

- An additional analysis of the method on more realistic simulated test data is advised. Originally we wanted to test the algorithms also on sequences created with the TOF-simulator developed by Keller et al. [Kel+07]. Unfortunately the simulator was not yet capable of simulating textures on surfaces, which however are essential to our method.

- Finally a more thorough study of the interdependence of range and amplitude/reflectivity measurement is of interest for an exact motion estimate. A mostly uncertain property in this context is the amplitude/irradiance dependent demodulation contrast of the various sensors. It could be essential for an optimal modeling of the irradiance used in the extended BCCE.

Part III

Appendices

Acronyms and Notation

Acronyms and Abbreviations

APS	Active Pixel Sensor
CCD	Charge Coupled Device
CMOS	Complementary Metal–Oxide–Semiconductor
CTE	Charge Tranfer Efficiency
DFT	Discrete Fourier Transform
DRNU	Dark Response NonUniformity
FFT	Fast Fourier Transform
FT	Fourier Transform
i.i.d.	Independent and Identically Distributed
MTF	Modulation Transfer Function
NLS	Nonlinear Least Squares
OLS	Ordinary Least Squares
pdf	Probability Density Function
PMD	Photonic Mixer Device
PRNU	Photo Response NonUniformity
PSF	Point Spread Function
PTLS	Partial Total Least Squares
SNR	Signal-to-Noise Ratio

SVD	Singular Value Decomposition
TLS	Total Least Squares
TOF	Time Of Flight
w.l.o.g.	Without Loss Of Generality

General notation

$\langle \boldsymbol{a}, \boldsymbol{b} \rangle$	Standard dot product between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$
$\langle v \rangle$	Expectation value or average w.r.t. to random variable or set v
$\langle\langle \boldsymbol{A}, \boldsymbol{B} \rangle\rangle$	Dot product between matrices $\boldsymbol{A}$ and $\boldsymbol{B}$
$\hat{v}$	Estimate of a (random) variable v in a statistical sense
$\imath$	Imaginary unit $\sqrt{-1}$
$\boldsymbol{M}$	$m \times n$ matrix
$\operatorname{diag} \boldsymbol{a}$	A diagonal matrix with vector $\boldsymbol{a}$ on its diagonal.
$\mathfrak{I}$	Identity matrix
$\operatorname{tr} \boldsymbol{A}$	The trace of matrix $\boldsymbol{A}$.
$\mathbb{1}$	Unit matrix of ones
$\mathcal{B}$	Binomial convolution operator or averaging operator
$\mathcal{O}, \mathcal{P}, \mathcal{Q}, \ldots$	Caligraphic letters indicate a representation-independent operator
$\boldsymbol{v}$	Column vector
$\bar{\boldsymbol{m}}$	Statistical mean (either arithmetic mean or population mean) over a set of measurements $\boldsymbol{m}$
$\hat{\boldsymbol{v}}$	Normalized or unit vector
(v_i)	Vector $\boldsymbol{v}$ with components v_i.
$\boldsymbol{a_n}$	The n^{th} vector of a sequence of vectors
$\boldsymbol{v}^{\mathrm{T}}, \boldsymbol{M}^{\mathrm{T}}$	Transposed (column) vector (*i.e.* row vector) or matrix

Greek Symbols

φ, θ	Phase of a periodic signal
σ	Standard deviation of a normal distribution

Latin Symbols

$\mathbb{C}$	Complex numbers
$\mathbb{N}$	Natural numbers
$\mathbb{R}$	Real numbers
$\mathbb{R}^n$	n-dimensional vector space over $\mathbb{R}$

Bibliography

[Alb07] Martin Albrecht. "Untersuchung von Photogate-PMD-Sensoren hinsichtlich qualifizierender Charakterisierungsparameter und -methoden". PhD thesis. Siegen, Germany: Department of Electrical Engineering and Computer Science, 2007.

[BA96] M. J. Black and P. Anandan. "The robust estimation of multiple motions: parametric and piecewise-smooth flow fields". In: *Computer Vision and Image Understanding* 63 (1996). Pp. 75–104.

[Bar00] E. Barth. "The Minors of the Structure Tensor". In: *DAGM*. Kiel, Germany 2000. Pp. 221–228.

[Bar+03] E. Barth et al. "Spatio-temporal Motion Estimation for Trancparency and Occlusions". In: *In Proceedings of IEEE International Conference on Image Processing*. 2003.

[Big06] Josef Bigun. *Vision with direction*. Berlin: Springer Verlag, 2006. URL: http://www2.hh.se/staff/josef/.

[Bla+98] M. J. Black et al. "Robust anisotropic diffusion". In: *IEEE Transactions on Image Processing* 7.3 (Mar. 1998). Pp. 412–432.

[BR96] Michael J. Black and Anand Rangarajan. "On the Unification of Line Processes, Outlier Rejection and Robust Statistics with Applications in Early Vision". In: *International Journal of Computer Vision* 19.1 (July 1996). Pp. 57–92.

[Bru+06] A. Bruhn et al. "A Multigrid Platform for Real-Time Motion Computation with Discontinuity-Preserving Variational Methods". In: *International Journal of Computer Vision* 70.3 (2006). Pp. 257–277. DOI: 10.1007/s11263-006-6616-7.

[BWS05] A. Bruhn, J. Weickert, and C. Schnoerr. "Lucas/Kanade Meets Horn/Schunk: Combining Local and Global Optic Flow Methods". In: *International Journal of Computer Vision* 61.3 (2005). Pp. 211–231.

[DD02] Frédo Durand and Julie Dorsey. "Fast bilateral filtering for the display of high-dynamic-range images." In: *SIGGRAPH*. Ed. by Tom Appolloni. ACM, 2002. Pp. 257–266. ISBN: 1-58113-521-1. URL: http://dblp.uni-trier.de/db/conf/siggraph/siggraph2002.html#DurandD02.

[Far03] Gunnar Farnebäck. "Two-Frame Motion Estimation Based on Polynomial Expansion". In: *Proceedings of the 13th Scandinavian Conference on Image Analysis*. LNCS 2749. Gothenburg, Sweden 2003. Pp. 363–370.

[FFS06] Michael Felsberg, Per-Erik Forssén, and Hanno Scharr. "Channel Smoothing: Efficient Robust Smoothing of Low-Level Signal Features." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 28.2 (Aug. 23, 2006). Pp. 209–222. URL: http://dblp.uni-trier.de/db/journals/pami/pami28.html#FelsbergFS06.

[FGW02] Per-Erik Forssén, G.H. Granlund, and J. Wiklund. *Channel Representation of Colour Images*. Technical Report LiTH-ISY-R-2418. Dept. of Electrical Eng., Linköping Univ., 2002.

[For04] Per-Erik Forssén. "Low and Medium Level Vision using Channel Representations". Dissertation No. 858, ISBN 91-7373-876-X. PhD thesis. SE-581 83 Linköping, Sweden: Linköping University, Sweden, 2004.

[For98] Bength Fornberg. "Calculation of Weights in Finite Difference Formulas". In: *SIAM Review* 40.3 (1998). Pp. 685–691.

[Fos93] E. R. Fossum. "Active pixel sensors: are CCDs dinosaurs?" In: *Charge-Coupled Devices and Solid State, Optical Sensors III*. Ed. by M. M. Blouke. Vol. 1900. Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference. 1993. Pp. 2–14.

[Fow+98] B. Fowler et al. "A Method for Estimating Quantum Efficiency for CMOS Image Sensors". In: *Proc. SPIE* 3301 (1998). Pp. 178–185.

[FPA99] C. Fermueller, R. Pless, and J. Aloimonos. "Statistical Biases in Optic Flow". In: *CVPR'99*. Fort Collins, Colorado 1999.

[Fra07] Mario Frank. "Investigation of a 3D Camera". MA thesis. Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg, 2007.

[FSF02] Michael Felsberg, Hanno Scharr, and Per-Erik Forssén. *The B-Spline Channel Representation: Channel Algebra and Channel Based Diffusion Filtering*. Tech. Report LiTH-ISY-R-2461. Dept. of Electrical Eng., Linköping Univ., 2002.

[GHO99] G. H. Golub, P. C. Hansen, and D. P. O'Leary. "Tikhonov Regularization and Total Least Squares". In: *SIAM Journal on Matrix Analysis and Applications* 21.1 (1999). Pp. 185–194.

[GK95] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Dordrecht, The Netherlands: Kluwer Academic, 1995.

[GL80] G. H. Golub and C. F. van Loan. "An Analysis of the Total Least Squares Problem". In: *SIAM Journal on Numerical Analysis* 17.6 (Dec. 1980). Pp. 883–893.

[GL96] G. H. Golub and C. F. van Loan. *Matrix Computations*. 3rd ed. Baltimore and London: The Johns Hopkins University Press, 1996.

[Gov06] V.M. Govindu. "Revisiting the Brightness Constraint: Probabilistic Formulation and Algorithms". In: *ECCV*. 2006. III: 177–188.

[Grö03] Hermann Gröning. "Radiometrische Kalibrierung und Charakterisierung von CCD- uund CMOS Bild-Sensoren und Monokulares 3D-Tracking in Echtzeit". PhD thesis. University of Heidelberg, 2003. URL: http://www.ub.uni-heidelberg.de/archiv/3589.

[Had02] J. Hadamard. "Sur les problèmes aux dérivées partielles et leur signification physique". In: *Princeton University Bulletin* (1902). Pp. 49–52.

[Hei01] Horst G. Heinol. "Untersuchung und Entwicklung von modulationslaufzeitbasierten 3D-Sichtsystemen". German. PhD thesis. Siegen, Germany: Department of Electrical Engineering and Computer Science, 2001. P. 157.

[HF01] H. Haußecker and D. J. Fleet. "Computing Optical Flow with Physical Models of Brightness Variation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.6 (June 2001). Pp. 661–673.

[Hil82] E. C. Hildreth. "The Integration of Motion Information along Contours". In: *IEEE Workshop on Computer Vision, Representation and Control*. 1982. Pp. 83–91.

[Hor86] B. K. P. Horn. *Robot Vision*. Cambridge, MA: MIT Press, 1986.

[Hor87] B. K. P. Horn. "Motion fields are hardly ever ambiguous". In: *Int.J.of Computer Vision* 1 (1987). Pp. 259–274.

[HS99] Horst Haußecker and Hagen Spies. “Motion”. In: *Handbook of Computer Vision and Applications*. Ed. by Bernd Jähne, Peter Geißler, and Horst Haußecker. Vol. 2: Signal Processing and Pattern Recognition. Academic Press, 1999. Chap. 13.

[Häu+99] G. Häusler et al. “Three-Dimensional Sensors - Potentials and Limitations”. In: *Handbook of Computer Vision and Applications*. 1. Academic Press, 1999. Pp. 485–506.

[Hub81] P. J. Huber. *Robust Statistics*. New York: John Wiley and Sons, 1981.

[JDD03] Thouis R. Jones, Frédo Durand, and Mathieu Desbrun. “Non-iterative, feature-preserving mesh smoothing.” In: *ACM Trans. Graph.* 22.3 (Feb. 9, 2003). Pp. 943–949. URL: http://dblp.uni-trier.de/db/journals/tog/tog22.html#JonesDD03.

[JGH99] Bernd Jähne, Peter Geißler, and Horst Haußecker. *Handbook of Computer Vision and Applications*. San Diego: Academic Press, 1999.

[JH00] Bernd Jähne and Horst Haußecker. *Computer Vision and Applications: A Guide for Students and Practitioners*. Academic Press, 2000.

[Jäh02] B. Jähne. *Digital Image Processing*. 5th ed. Berlin, Germany: Springer, 2002.

[Jäh04] Bernd Jähne. *Practical Handbook on Image Processing for Scientific and Technical Applications*. 2nd ed. Boca Rota London New York Washington, D.C.: CRC Press, 2004.

[JHG99] B. Jähne, H. Haußecker, and P. Geißler. “Neighborhood Operators”. In: *Handbook of Computer Vision and Applications*. 5 2. Academic Press, 1999. Pp. 93–124.

[JSK99] B. Jähne, H. Scharr, and S. Körkel. “Principles of Filter Design”. In: *Handbook of Computer Vision and Applications*. Ed. by B. Jähne, H. Haußecker, and P. Geißler. Vol. 2. Academic Press, 1999. Pp. 125–151.

[Jus01] Detlef Justen. “Untersuchung eines neuartigen 2D- gestützten 3D-PMD-Bildverarbeitungssystems”. German. PhD thesis. Siegen, Germany: Department of Electrical Engineering and Computer Science, 2001.

[Kel+07] M. Keller et al. “A Simulation Framework for Time-Of-Flight Sensors”. In: *International Symposium on Signals, Circuits and Systems (ISSCS)*. Vol. 1. Iasi, Romania: IEEE CAS Society, 2007. Pp. 125–128.

[Kla05] M. Klar. “Design of an endoscopic 3-D Particle-Tracking Velocimetry system and its application in flow measurements within a gravel layer”. PhD thesis. University of Heidelberg, 2005. URL: http://archiv.ub.uni-heidelberg.de/volltextserver/volltexte/2005/5961/pdf/klar_PHD2005.pdf.

[KRI06] T. Kahlmann, F. Remondino, and H. Ingensand. “Calibration for increased accuracy of the range imaging camera SwissRanger”. In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXVI.5 (2006). Pp. 136–141.

[Köt03] U Köthe. “Edge and junction detection with an improved structure tensor”. In: *Proc. of 25th DAGM Symposium*. Ed. by B. Michaelis and G. Krell. Vol. 2781. Lecture Notes in Computer Science. DAGM. Magdeburg 2003. Pp. 25–32. URL: http://kogs-www.informatik.uni-hamburg.de/~koethe/papers/structureTensor.pdf.

[KW93] H. Knutsson and C. F. Westin. “Normalized and Differential Convolution: Methods for Interpolation and Filtering of Incomplete and Uncertain Data”. In: *CVPR*. New York City 1993. Pp. 515–516.

[Lan00] Robert Lange. “Time-of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology”. English. PhD thesis. Siegen, Germany: Department of Electrical Engineering and Computer Science, 2000.

[LK06] M. Lindner and A. Kolb. “Lateral and Depth Calibration of PMD-Distance Sensors”. In: *International Symposium on Visual Computing (ISVC06)*. Vol. 2. Lake Tahoe, Nevada: Springer, 2006. Pp. 524–533. ISBN: 978-3-540-48626-8.

[LP02] Prof. Dr. Wolfgang von der Linden and DI Alexander Prüll. *Wahrscheinlichkeitstheorie, Statistik und Datenanalyse*. Course-Script, Institute of Theoretical and Computational Physics, TU Graz, 2002. URL: http://itp.tugraz.at/LV/wvl/Statistik/A_WS_pdf.pdf.

[LS01] R. Lange and P. Seitz. “Solid-state time-of-flight range camera”. In: *Quantum Electronics, IEEE Journal of* 37.3 (2001). Pp. 390–397.

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

[MF81] L. Mortara and A. Fowler. “Evaluations of charge-coupled device (CCD) performance for astronomical use”. In: *Proc. SPIE* 290 (1981). Pp. 28–33.

[Müh04] Matthias Mühlich. “Estimation in Projective Spaces and Applications in Computer Vision”. PhD thesis. Johann Wolfgang Goethe Universität in Frankfurt am Main, 2004.

[MM01] Matthias Mühlich and Rudolf Mester. “Subspace Methods and Equilibration in Computer Vision”. In: *Proceedings of Scandinavian Conference on Image Analysis SCIA 2001 Bergen*. 2001.

[MSB01] C. Mota, I. Stuke, and E. Barth. “Analytic solutions for multiple motions”. In: *Proc. of International Conference on Image Processing*. Vol. 2. 2001. Pp. 917–920.

[MV06] Kurt Meyberg and Peter Vachenauer. *Differentialgleichungen, Funktionentheorie, Fourier-Analysis, Variationsrechnung*. Vol. 2. Höhere Mathematik. Berlin ; Heidelberg: Springer, 2006. XIII, 457 S. ISBN: 3-540-41851-2, 978-3-540-41851-1.

[NGK94] Klas Nordberg, Gösta H. Granlund, and Hans Knutsson. “Representation and Learning of Invariance”. In: *ICIP (2)*. 1994. Pp. 585–589. URL: http://dblp.uni-trier.de/db/conf/icip/icip1994-2.html#NordbergGK94.

[Pap+06] Nils Papenberg et al. “Highly Accurate Optic Flow Computation with Theoretically Justified Warping.” In: *International Journal of Computer Vision* 67.2 (2006). Pp. 141–158.

[Pla06] Matthias Plaue. *Analysis of the PMD Imaging System*. Tech. rep. Interdisciplinary Center for Scientific Computing (IWR), Univ. of Heidelberg, 2006.

[PM90] P. Perona and J. Malik. “Scale Space and Edge Detection using Anisotropic Diffusion”. In: *PAMI* 12 (July 1990). Pp. 629–639.

[Rap07] Holger Rapp. “Experimental and Theoretical Investigation of Correlating TOF-Camera Systems”. MA thesis. Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg, 2007.

[RL87] Peter J. Rousseeuw and Annik M. Leroy. *Robust Regression and Outlier Detection*. Wiley & Sons, New York, 1987.

[Sch00] H. Scharr. “Optimale Operatoren in der Digitalen Bildverarbeitung”. PhD thesis. Heidelberg, Germany: University of Heidelberg, 2000.

[Sch03] Bernd Schneider. “Der Photomischdetektor zur schnellen 3D-Vermessung für Sicherheitssysteme und zur Informationsübertragung im Automobil”. PhD thesis. Siegen, Germany: Department of Electrical Engineering and Computer Science, 2003.

[Sch+98] R. Schwarte et al. “Novel 3D-vision systems based on layout optimized PMD-structures”. German. In: *Technisches Messen* 65.7-8 (1998). Pp. 264–271. ISSN: 0171-8096.

[SG02] H. Spies and C. S. Garbe. “Dense Parameter Fields from Total Least Squares”. In: *Pattern Recognition*. Ed. by L. Van Gool. Vol. LNCS 2449. Lecture Notes in Computer Science. Zurich, CH: Springer-Verlag, 2002. Pp. 379–386. URL: http://books.google.com/books?id=0xcL1dIafSUC.

[SJB00] H. Spies, B. Jähne, and J. L. Barron. “Regularised Range Flow”. In: *ECCV*. Ed. by D. Vernon. Vol. 2. Lecture Notes in Computer Science 1843. Dublin, Ireland: Springer, 2000. Pp. 785–799.

[Spi01] H. Spies. “Analysing Dynamic Processes in Range Data Sequences”. PhD thesis. Heidelberg, Germany: University of Heidelberg, 2001.

[Spi+99] Hagen Spies et al. “Differential Range Flow Estimation”. In: *DAGM-Symposium*. 1999. Pp. 309–316.

[Ste99] Charles V. Stewart. “Robust Parameter Estimation in Computer Vision”. In: *Society for Industrial and Applied Mathematics, SIAM* 41.3 (1999). Pp. 513–537.

[Stu+03] I. Stuke et al. “Estimation of multiple motions: regularization and performance evaluation”. In: *Image and Video Communication and Processing, Proceedings of SPIE*. Vol. 5022. 2003. Pp. 75–86.

[Tel+06] Alexandru Telea et al. “A Variational Approach to Joint Denoising, Edge Detection and Motion Estimation.” In: *DAGM-Symposium*. 2006. Pp. 525–535.

[TM98] Carlo Tomasi and Roberto Manduchi. “Bilateral Filtering for Gray and Color Images.” In: *ICCV*. 1998. Pp. 839–846. URL: http://dblp.uni-trier.de/db/conf/iccv/iccv1998.html#TomasiM98.

[TRK01] Yanghai Tsin, Visvanathan Ramesh, and Takeo Kanade. “Statistical Calibration of CCD Imaging Process”. In: *In IEEE International Conference on Computer Vision*. 2001.

[Tsc02] D. Tschumperle. “PDE’s based regularization of multivalued images and applications”. PhD thesis. Université de Nice-Sophia, 2002.

[Tsc06] David Tschumperlé. “Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDE’s”. In: *Int. J. Comput. Vision* 68.1 (2006). Pp. 65–82. ISSN: 0920-5691. DOI: http://dx.doi.org/10.1007/s11263-006-5631-z.

[VHV91] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. http://www.netlib.org/vanhuffel/. Philadelphia: Society for Industrial and Applied Mathematics, 1991.

[Wag03] C. Wagner. “Informationstheoretische Grenzen optischer 3D-Sensoren”. PhD thesis. Universität Erlangen-Nürnberg, 2003.

[Wes94] C. F. Westin. “A Tensor Framework for Multidimensional Signal Processing”. PhD thesis. Linköping, Sweden: Linköping University, 1994.

[Xia+06] J. Xiao et al. “Bilateral Filtering-Based Optical Flow Estimation with Occlusion Detection”. In: *ECCV06*. 2006. I: 211–224.

[Xu99] Zhanping Xu. *Investigation of 3D-Imaging Systems Based on Modulated Light and Optical RF-Interferometry (ORFI)*. English. Vol. 14. ZESS Forschungsberichte. Aachen, Germany: Shaker Verlag, 1999. ISBN: 3-8265-6736-6. URL: http://www.shaker.de/Online-Gesamtkatalog/Booklist.idc?Reihe=102.

[Yam+93] F. Yamamoto et al. “Three-dimensional PTV based on binary cross-correlation method”. In: *JSME International Journal, Series B* 36.2 (1993). Pp. 279–284.