UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE ELECTRÓNICA

Doctoral Thesis

Correction of Errors in Time of Flight Cameras

David Jiménez-Cabello

2015

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE ELECTRÓNICA

PHD IN ELECTRONICS: ADVANCED ELECTRONIC SYSTEMS. INTELLIGENT SYSTEMS

Correction of Errors in Time of Flight Cameras

Author

David Jiménez-Cabello

Advisors

Daniel Pizarro-Pérez and Manuel Mazo-Quintas

2015

Doctoral Thesis

A mi familia, y en especial a mis padres, Isaac y Esther:
esta tesis es por y para ellos.

La utopía está en el horizonte. Camino dos pasos, ella se aleja dos pasos
y el horizonte se corre diez pasos más allá.
¿Entonces para que sirve la utopía?
Para eso, sirve para caminar.

—

"Utopia is on the horizon. I walk two steps, she moves two steps away
and the horizon runs ten steps further.
So what good is utopia?
That serves to walk."

—

Eduardo Galeano

Acknowledgements

I would like to thank all the people that have shared these last 4 years with me. They have celebrated all the great moments, but also they have suffered the tough ones. Specially, I would like to thank Natalia, mother of my son Antón and loved partner, for her complete dedication and invaluable help that made this journey extremely easier. I would also like to express my deepest gratitude to my advisors, Daniel Pizarro and Manuel Mazo, for their generous support and the absolute trust that they have showed me since I started this thesis: they represent a big portion of my life. Thanks a lot to my lab-mates: Raquel, Álvaro, Jose (we did it very fun!) and the other members of the Electronics Department: Javier Macías, Sira Palazuelos, Marta Marrón, Cristina Losada, Maria José..., you all teach me a lot of things.

I will also like to mention all those who trusted me: Esther e Isaac (parents), Elena, Teto, Naia and Marcos (sister and co), Leonides-Lorenzo (grandfather), Carlos, Nati, Carlitos, Isma, M. Jose, Dani, Hugo and Marcos (family in law), Marta and Fran (friends in Alcalá), Juanma, Javi, Tato (climbing friends), Mr. Pip, Ernesto, Gabri, Raul, Bull, Jose (climbing daddies), Txus (our perfect host in Clermont), Adrien (academic supervisor in Clermont), Voltereta's families and finally, but not least, my beloved dog Vera (unconditional mate).

To all of you and those I forgot to mention: **Gracias!**

Resumen

En esta tesis se aborda la corrección de errores en cámaras de profundidad basadas en tiempo de vuelo - Time of Flight (ToF). De entre las más recientes tecnologías, las cámaras ToF de modulación continua - Continuous Wave Modulation (CWM) - son una alternativa prometedora para la creación de sensores compactos y rápidos. Sin embargo, existe una gran variedad de errores que afectan notablemente la medida de profundidad, comprometiendo posibles aplicaciones. La corrección de dichos errores propone un reto desafiante. Actualmente, se consideran dos fuentes principales de error: *i*) sistemático y *ii*) no sistemático. Mientras que el primero admite calibración, el segundo depende de la geometría y el movimiento relativo de la escena. Esta tesis propone métodos que abordan *i*) *la distorsión sistemática de profundidad* y dos de las fuentes de error no sistemático más relevantes: *ii.a*) *la interferencia por multicamino* y *ii.b*) *los artefactos de movimiento.*

La distorsión sistemática de profundidad en cámaras ToF surge principalmente debido al uso de señales sinusoidales no perfectas para modular. Como resultado, las medidas de profundidad aparecen distorsionadas, pudiendo ser reducidas con una etapa de calibración. Esta tesis propone un método de calibración basado en mostrar a la cámara un plano en diferentes posiciones y orientaciones. Este método no requiere de patrones de calibración y, por tanto, puede hacer uso de los planos, que de manera natural, aparecen en la escena. El método propuesto encuentra una función que obtiene la corrección de profundidad correspondiente a cada píxel. Esta tesis mejora los métodos existentes en cuanto a precisión, eficiencia e idoneidad.

La interferencia por multicamino - Multipath Interference (MpI) - surge debido a la superposición de la señal reflejada por diferentes caminos con la reflexión directa, produciendo distorsiones que se hacen más notables en superficies convexas. La MpI es la causa de importantes errores en la estimación de profundidad en cámaras CWM ToF. Esta tesis propone un método que elimina la MpI a partir de un solo mapa de profundidad. El enfoque propuesto no requiere más información acerca de la escena que las medidas ToF. El método se fundamenta en un modelo radiométrico de las medidas que se emplea para estimar de manera muy precisa el mapa de profundidad sin distorsión.

Una de las tecnologías líderes para la obtención de profundidad en imagen ToF está basada en Photonic Mixer Device (PMD). Dicha tecnología obtiene la profundidad mediante el muestreado secuencial de la correlación entre la señal de modulación y la señal proveniente de la escena en diferentes desplazamientos de fase. Con movimiento, los píxeles PMD capturan profundidades diferentes en cada etapa de muestreo, produciendo *artefactos de movimiento.* En esta tesis se propone un método para la corrección de dichos artefactos que destaca por su velocidad y sencillez, pudiendo ser incluido fácilmente en el hardware de la cámara. La profundidad de cada píxel se recupera gracias a la consistencia entre las muestras de correlación en el píxel PMD y de la vecindad local. Este método obtiene correcciones precisas, reduciendo los artefactos de movimiento enormemente. Además, como resultado de este método, se obtiene el flujo óptico en los contornos en movimiento a partir de una única captura.

A pesar de ser una alternativa muy prometedora para la obtención de profundidad, las cámaras ToF todavía tienen que resolver problemas desafiantes en relación a la corrección de errores sistemáticos y no sistemáticos. Esta tesis propone métodos eficaces para enfrentarse con estos errores.

Palabras clave: cámara de tiempo de vuelo, errores sistemáticos de profundidad, interferencia por multicamino y artefactos de movimiento.

Abstract

This thesis addresses the correction of errors in Time of Flight (ToF) depth cameras. Among current technologies, Continuous Wave Modulation (CWM) ToF cameras are a promising alternative for creating compact and fast sensors. However, a wide variety of error sources notably affect depth measurements, compromising potential applications. Correction of errors in ToF cameras is a very challenging problem. Two main types of errors are considered: *i*) systematic and *ii*) non-systematic errors. While the former admits calibration, the latter depends on the scene's geometry and motion. This thesis proposes methods to address *i*) systematic depth distortion and two of the most relevant sources of non-systematic errors: *ii.a*) multipath interference and *ii.b*) motion artifacts.

Systematic depth distortion in ToF cameras is mainly produced by the use of non-ideal sinusoidal signals to modulate light. As a result, depth measurements appear distorted but this distortion can be reduced with a calibration step. This thesis proposes a calibration method based on showing the camera a flat surface in different positions and orientations. This method does not require special patterns and thus, natural planes existing in the scene can be used. The proposed method finds a function that gives the corresponding depth correction for every pixel. This thesis improves existing methods in terms of accuracy, efficiency and optimality.

Multipath Interference (MpI) appears due to the overlap of light reflected from different paths with the direct path reflection, producing notable distortions that become more pronounced in convex surfaces. Multipath interference of light is the cause of important errors in CWM ToF depth estimation. This thesis proposes a method that removes MpI from a single depth map. The proposed approach does not require information about the scene, apart from ToF measurements. The method is based on a radiometric model of ToF measurements that is used to estimate very accurately the undistorted depth map.

One of the leading CWM ToF imaging technologies for depth sensing is based on Photonic Mixer Device (PMD), that obtains depth by sequentially sampling the correlation between the modulation signal and the incoming signal from the scene at several phase shifts. With motion, PMD pixels capture different depths at each stage of the pipeline, producing *motion artifacts*. The method proposed in this thesis for correcting motion artifacts is very fast, simple and can be easily included in camera's hardware. Depth of each pixel is recovered by exploiting consistency of each sample of the correlation function given by the PMD pixel and its local neighbours. This method obtains accurate corrections that highly reduce motion artifacts. In addition, as a result of this method, the motion flow of occluding contours is available from a single frame.

Despite being a very promising alternative for depth sensing, ToF cameras need to solve challenging problems regarding the correction of systematic and non-systematic errors. This thesis proposes effective methods to cope with these errors.

Keywords: Time of Flight Camera, Systematic Depth Distortion, Multipath Interference and Motion Artifacts.

Contents

List of Figures

List of Tables

List of Acronyms

BRDF	Bidirectional Reflectance Distribution Function.
CMOS	Complementary Metal Oxide Semiconductor.
CWM	Continuous Wave Modulation.
FPN	Fixed Pattern Noise.
IR	infrared.
LBE	Linear Basis Expansion.
LED	Light Emitting Diode.
LLS	Linear Least-Squares.
LPB	Linear Plane Border.
MpI	Multipath Interference.
PDE	Partial Differential Equation.
PMD	Photonic Mixer Device.
RMS	Root Mean Square.
SLiP	Structure Light Projection.
SPAD	Single Photon Avalanche Diode.
ToF	Time of Flight.
TPS	Thin Plate Spline.

Chapter 1

Introduction

1.1 Motivation

This thesis addresses the correction of errors for one of the most promising technologies in low-cost depth sensing: *Time of Flight (ToF)* cameras. So far, ToF cameras have been used in a wide variety of problems, bringing new frontiers in several fields: *i*) electronics [1, 2], *ii*) computer vision [3, 4], *iii*) robotics [5, 6], *vi*) medical imaging [7, 8], *v*) entertainment [9], *vi*) security [10, 11], *vi*) ray tracing [12, 13]), etc.

A ToF camera consists of a sensor array, similar to that used to capture light in a conventional camera, that provides a measurement of depth for each of its pixels. As a result, the camera produces a range image (i.e. 2.5D image) of the scene. To obtain distances, ToF cameras measure the time-of-flight of a light signal that impacts the scene and returns back to the sensor. Current ToF technologies can be classified into three categories: *i*) *pulsed modulation*, *ii*) *Continuous Wave Modulation (CWM)* and *iii*) *pseudo-noise modulation*. The most established architecture is based on CWM technology. In practice, CWM ToF cameras use modulated *infrared (IR)* light sources, measuring time-of-flights with signal phase shifts. In this thesis we focus on CWM ToF technology which is commonly based on *Photonic Mixer Device (PMD)* pixels.

In 2001, Lange et al. [1] established the foundations of modern ToF cameras. Since then, commercial ToF cameras have progressively reduced their cost while showing improvements in terms of frame rate, size or accuracy. These improved features make recent ToF cameras a suitable solution in many applications and also a promising candidate for small and portable applications, such as mobile phones [14, 15] or medical endoscopes [8, 16]. Compared to multiple camera systems, which need complex processing and textured surfaces, ToF technology provides fast and reliable depth information at the hardware level that is independent of the scene's texture.

Currently, CWM ToF cameras represent one of the leading technologies in range imaging, producing compact and low-cost sensors. However, CWM ToF technology suffers from error sources that still limit its accuracy (further information can be found in [1]). These errors can be classified into *i*) *systematic* and *ii*) *non-systematic* errors. Systematic errors have been discussed widely in the literature and most of them admit calibration (see [17] for a comprehensive survey). Despite most commercial cameras include factory-made hardware compensations for some systematic errors, they result in practice imprecise and require additional calibration processes, involving complex equipment and a highly technical knowledge. Contrary, non-systematic error sources don't admit calibration as they depend on the scene or they have a random behaviour. Correction of non-systematic errors is thus very challenging and some of the solutions proposed in the literature involve several disciplines, such as machine learning, signal processing, graphics, etc.

These aforementioned error sources affect depth measurements significantly and limit the applicability of ToF cameras in practice. Correction of these errors is necessary and decisive to consolidate ToF cameras as a reliable depth sensing technology. A main motivation of this thesis is to propose correction methods for both systematic and non-systematic errors that do not require special equipment or additional hardware. This makes them affordable for a final user. This thesis focuses on CWM ToF cameras, as they represent a very promising technology. However, some of the solutions proposed can be used in any depth camera.

1.2 Problem Statement

This thesis addresses the correction of systematic and non-systematic errors, mainly in CWM ToF cameras. We propose a general calibration method for systematic depth distortion and methods for reducing two main sources of non-systematic errors: multipath interference and motion artifacts, that have a relevant impact on depth's accuracy.

The main cause of *systematic depth distortion* in CWM ToF cameras is due to the fact that the modulation signal is not a perfect sinusoidal, the so-called wiggling error (see [1]). As a consequence, depth distortion can be high in some cases (see Fig. 1.1 for an overview).

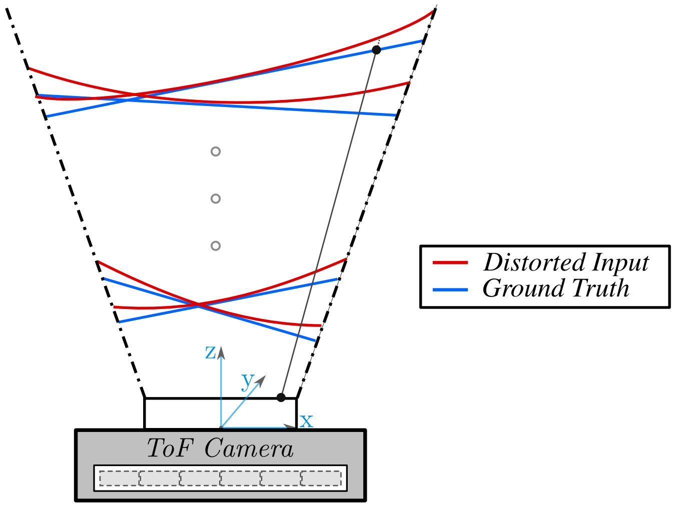


Figure 1.1: Systematic depth distortion phenomenon in a set of planes. We show in colours *i*) the distorted input (red curves) and *ii*) the corresponding ground truth (blue lines).

Most non-systematic errors depend on the scene's geometry or motion, though they are often called random errors. Therefore, they don't admit calibration techniques. Methods reducing non-systematic errors require to analyse the scene seen by the camera and to model how these errors interact with a particular geometry or motion. In this thesis, we propose the modelling and correction of two challenging and relevant non-systematic errors: *i*) *Multipath Interference (MpI)* and *ii*) *motion artifacts.*

Multipath interference appears when reflected light from a given point of the scene overlaps with light reflected from other parts of the scene. This produces a distortion on the depth that becomes more pronounced in the vicinity of corners or holes (see Fig. 1.2 for an illustration).

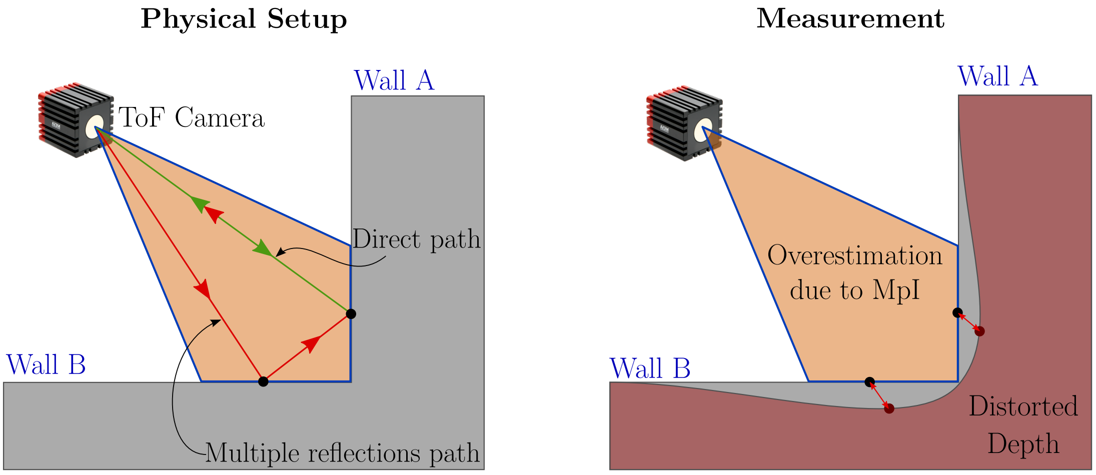


Figure 1.2: Multipath interference phenomenon. On the left, we show the physical setup of a corner-like scene. On the right, we show in red the MpI effect in the scene: the corner has no flat surfaces but curved.

Motion artifacts appear when there is relative motion between the camera and the scene. In that case, during integration time of a frame, a pixel captures light from different depths which generates artifacts. Their magnitude depend on the particular technology used for computing depth. In PMD cameras, depth is obtained by measuring the correlation of the incoming signal in a pipeline of four sequential stages. With motion, the different steps are not coherent with a single depth, producing artifacts (see Fig. 1.3 for an example).

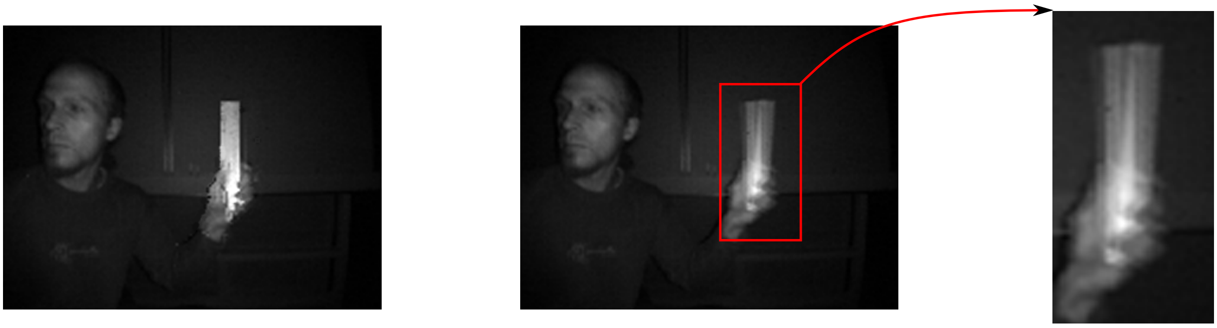


Figure 1.3: Motion artifacts phenomenon. We show a planar object moving in a perpendicular direction in front of the camera. On the left, we show the original object. On the right, the captured amplitude image and a zoomed version of the affected area, respectively.

1.3 Proposed Approach

This section contains an overview of the proposed solutions that address both systematic and non-systematic errors mentioned in Section 1.2. Calibration of systematic depth distortion is presented first, followed by correction methods for non-systematic errors. The ordering is not chronological as can be deduced from the publications derived from the thesis.

1.3.1 Systematic Depth Distortion

Correction of systematic errors in ToF cameras becomes a priority for the acquisition of reliable measurements. In this document, a calibration method is proposed that compensates *systematic depth distortion* for any depth sensor. This thesis proposes a method that doesn't require expensive equipment nor complex settings. Hence, the final user only needs to show the camera a plane in different positions and orientations and to provide a reduced set of anchor points whose ground truth depth is known. A correction function is obtained for every pixel and every depth. Existing works had failed so far to give an efficient and global solution to this problem. The main contribution is to show that finding such correction function is indeed a convex optimisation problem. Convexity is gained by reformulating the constraints on the correction function using differential properties of planes. With this method one can calibrate a depth camera in closed-form using linear least-squares and natural planes existing in most of man-made environments.

1.3.2 Non-Systematic Multipath Interference

This thesis presents a method that automatically corrects the distortion caused by *multipath interference* in depth measurements obtained with CWM ToF cameras. A radiometric model is used to model MpI. Using this model this thesis demonstrates that all the information needed for compensating the influence of MpI on the scene, captured by the camera, is self-contained in the measurements (depth and amplitude of IR signal). An iterative optimisation method is proposed that, based on the measurements contaminated with MpI, gives depth correction for each pixel.

1.3.3 Non-Systematic Motion Artifacts

Finally, this thesis presents a method for correcting *motion artifacts* in PMD-based ToF cameras. The main idea is to use the different stages of the PMD technology to detect and to remove motion artifacts from a single frame. The proposed algorithm is very fast, simple and can be easily included in camera's hardware. This method recovers depth of each pixel by exploiting consistency of each stage of PMD's pipeline and using information from pixels' neighbours. In addition, this method obtains the motion flow of occluding contours in the image from a single frame.

1.4 Contributions of the Thesis

The following contributions are derived from the methods proposed in this thesis:

- A new calibration method for the correction of systematic depth distortion in ToF cameras, which is also applicable to any depth sensor. The proposed method computes a global correction function by simply showing to the camera a plane in different positions and angles covering the desired depth range. The solution is convex and requires linear least-squares.

- A new method for the correction of multipath interference in ToF cameras. This method proposes a radiometric model that explains the multipath phenomenon in ToF imaging. It is used as a generative model of the ToF measurements, allowing one to recover depth deviation caused by multipath distortion.

- A new method for the correction of motion artifacts in a single frame for PMD-based ToF cameras, which also obtains optical flow in occluding contours. This method is simple, fast and can be incorporated in camera's hardware.

- In addition, a new type of primitive specially adapted to range imaging called *Linear Plane Border (LPB)*, which is also applicable for any depth sensor. These are planar stripes of a certain width that are delineated at one side by a linear edge (i.e. depth discontinuity). We extend previous works to robustly detect multiple LPBs in range images from noisy sensors.

The contributions carried out during the thesis generated the following publications:

- D. Jiménez, D. Pizarro, M. Mazo and S. Palazuelos, **"Modelling and Correction of Multipath Interference in Time of Flight Cameras"**, in IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2012. [18]

- D. Jiménez, S. Behnke and D. Pizarro, **"Linear Plane Border. A Primitive for Range Images Combining Depth Edges and Surface Points"**, in Proceedings of 8-th International Conference on Computer Vision Theory and Applications (VISAPP), 2013. [19]

- D. Jiménez, D. Pizarro, M. Mazo and S. Palazuelos, **"Modeling and Correction of Multipath Interference in Time of Flight Cameras"** in Image and Vision Computing Journal (Elsevier), 2013. [20]

- D. Jiménez, D. Pizarro and M. Mazo, **"Single Frame Correction of Motion Artifacts in PMD-based Time of Flight Cameras"** in Image and Vision Computing Journal (Elsevier), 2014. [21]

- D. Jiménez, D. Pizarro, A. Bartoli and M. Mazo, **"CaDDiP: A Closed-Form Solution based on Local Planarity for the Calibration of Depth Distortion in Depth**

Cameras", submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.

During the thesis the author have also participated in the following research-related activities:

- Supervised MSc and BSc projects.
  - "Efficient Implementation of an Algorithm to Correct Multipath Interference in ToF Cameras", 2012-2013.
  - "Optimisation of Time-Consuming Functions using GPU-based implementation", 2013-2014.
  - "Unified Design of a Common Interface for Time of Flight Cameras (QOP)", 2012-2014.
- Informative talks.
  - "Outdoor scene analysis using Time of Flight Cameras", in Proceedings of 3rd Season of Young Researchers of the University of Alcalá, 2011.
  - "2.5D Vision using Time of Flight Cameras", in Season of Conferences and Seminars of the University of Alcalá, 2013.
  - "2.5D Vision using Time of Flight Cameras. Technology and Challenges", in Season of Conferences and Seminars of the University of Alcalá, 2014.

1.5 Structure of the Document

The rest of the document is structured as follows.

- In Chapter 2 we first review the most relevant range imaging techniques in computer vision and we briefly explain PMD-based CWM ToF fundamentals. Finally, we describe previous works regarding the correction of errors in CWM ToF cameras that are addressed in the present thesis.
- Chapter 3 describes the proposed calibration of systematic depth distortion for depth cameras.
- Chapter 4 shows our contribution in the modelling and correction of multipath interference in CWM ToF cameras.
- Chapter 5 describes the proposed correction of motion artifacts in PMD-based ToF cameras with motion flow estimation in a single frame.
- In Chapter 6 we discuss the conclusions of the present work. We focus on both the benefits and the limitations of each proposal and we present future research lines that could be started following the developed work.

- Appendices A and B clarify some details regarding the correction of motion artifacts in Chapter 5 and some related contributions derived during the thesis, respectively.

Every chapter in the document is self-contained and follows the same structure. First, an introduction to the problem and an overview, including a graphical abstract, is presented. Next, we introduce the notation, that can be slightly different on each chapter, and we describe the specific theoretical background. It follows a detailed explanation of the proposed method and finally, results and conclusion are presented.

Chapter 2

State of the Art

2.1 Introduction

This chapter reviews the existing literature on the correction of errors in *Continuous Wave Modulation (CWM) Time of Flight (ToF)* cameras. We focus on three main error sources studied in the thesis: *i*) systematic depth distortion and *ii.a*) multipath interference and *ii.b*) motion artifacts, respectively for systematic and non-systematic errors. In any case, several surveys are referenced in this chapter, should the reader need a deeper knowledge.

This chapter is organised as follows: in Section 2.2 we review the existing alternatives for depth estimation. Section 2.3 presents the fundamentals and different technologies in CWM ToF imaging. In Section 2.4 we review the main systematic error sources and the existing calibration methods. Similarly, in Section 2.5 we review the most significant non-systematic error sources and then we describe existing methods for correction of multipath interference (Section 2.5.1) and the compensation of motion artifacts (Section 2.5.2), respectively.

2.2 Range Sensing Techniques

An important objective in Computer Vision is to achieve similar capacities of living beings to understand the world using vision. Depth perception is one of the most important skills in highly developed animals, such as the human being. Depth computation has been a subject of interest for the research community and several optical and acoustic techniques have been proposed in the literature to compute depth in a particular scenario (e.g. stereo vision or bio-sonar).

According to its working principle, we distinguish in this section three different categories in the extraction of depth information using optical devices: *1*) interferometry, *2*) triangulation and *3*) light's time of flight.

2.2.1 Interferometry

Interferometry is based on the superposition of two monochromatic waves of a fixed frequency f but with different amplitudes (a_1 and a_2) and with different phases (φ_1 and φ_2), [22]. When two light waves are superimposed, it results in a different monochromatic wave with different amplitude a_3 and different phase φ_3 while keeping the same frequency. Based on the well-known phenomenon of interference, the resultant intensity at a particular point depends on whether the waves reinforce or cancel each other. Thus, the amplitude a_3 reaches a maximum for a resulting phase difference of multiples of the light wavelength $\lambda/2$, while it reaches a minimum for phase differences multiple of $\lambda/2 + \lambda/4$.

An interferometer is an optical instrument that splits the light beam into two parts by a beam splitter and travels two different paths. Both reflected beams are made to cross each other and the region of crossing will exhibit an interference fringe. Counting the number of minimum-maximum transitions in the interference, the distance between the optical system and the target surface can be measured with an accuracy of at least λ.

The main disadvantage of classical interferometer is that the unambiguous distance measurements are limited to λ and absolute distance measurements are not possible with this method. To overcome this limitation, different methods have been proposed in the literature using multiple wavelengths or enhanced systems. Despite the requirement of a precise alignment of the optics, the basic application for interferometry is the most accurate technique for small distances, ranging from micrometres to several centimetres.

2.2.2 Triangulation

Triangulation methods obtain depth information by analysing the trigonometric identities of a triangle that is formed between a point in a surface and the lines of sight of an optical system (see [23]). If this optical system consists of two passive receivers, the technique is called *i)* passive triangulation. If it consists of one receiver and one active emitter, the measurement principle is called *ii)* active triangulation.

Passive Triangulation From its similarity to human eyes, passive triangulation is the most common and well-known technique for the extraction of depth cues. Based on the stereopsis of the human eyes, this method computes depth by using the overlapping field of view of a pair of cameras in a stereo configuration. The operating depth range of stereo vision is determined by the separation between cameras (baseline): the higher the required depth range, the higher the required baseline distance. If an specific point can be seen in both images, then it's possible to determine its depth. Thus, the main challenge in stereo vision is to find point correspondences within both images. Based on epipolar geometry and image rectification, the two problem of finding correspondences is commonly reduced to finding the best match along a given scan line (see [24]). Assuming that the baseline b between the two cameras is known and that the measured viewing angles between the baseline and the two projection lines of two optical systems are α and β, the distance between the baseline and the target can be determined by:

$$D = \frac{b}{\dfrac{1}{\tan\alpha} + \dfrac{1}{\tan\beta}}. \tag{2.1}$$

In Fig. 2.1 we show a diagram of the passive triangulation method.

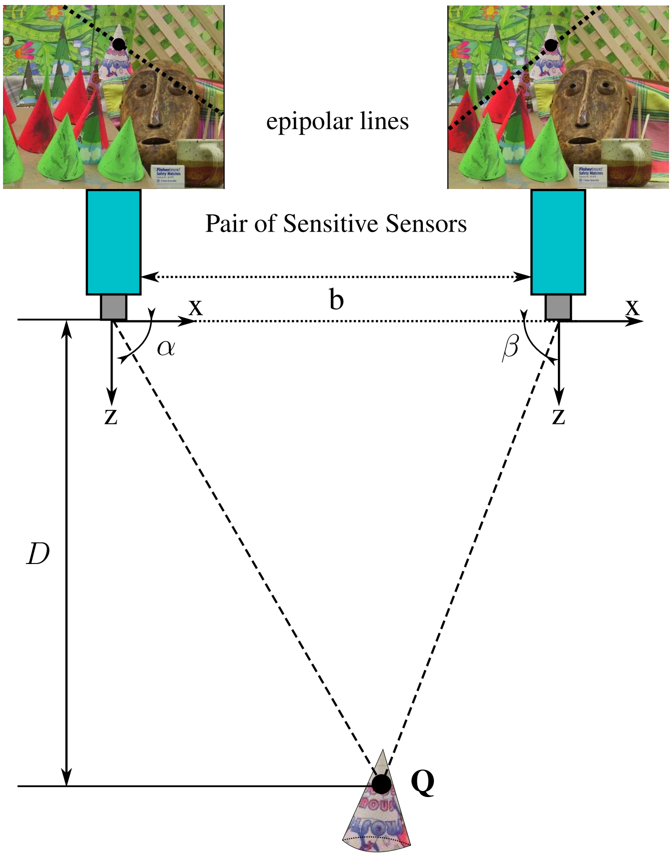


Figure 2.1: Passive triangulation for depth computing. Point **Q** can be recovered looking for the point correspondence in both images.

Passive stereo imaging is computationally expensive. In order to calculate the triangulation, corresponding points or features have to be matched between both images. The offset between the points in the two images is called disparity. From the disparity the distance can be calculated if the intrinsic and extrinsic parameters of both receivers (usually cameras) are known (see [25]). Feature extraction and matching require sufficient intensity and colour variation in the image for robust correlation. This requirement renders stereo vision less effective if the subject lacks these variations, for example, measuring the distance to an uniformly coloured object.

Due to occlusions, some parts of one image may not be visible in the other and hence no depth information can be extracted from the disparity for these parts (shadowing effect). This makes the matching problem difficult as it introduces features with no correspondence.

Active Triangulation In contrast to passive stereo vision, active systems like laser or structured light scanners avoid the problematic task of finding point correspondences by replacing one camera by a light source. By doing so, the triangulation angles are well-defined through the exit angle of the emitted light and the position of signal within the image (see Fig. 2.2). The main problems arise when the emitted light is not reflected properly. This is the case when an object is either too far away or its material properties result in very dark or saturated areas in the image.

The configuration of a simple active triangulation sensor is shown in Fig. 2.2. A light source projects a single point onto the measurement object. The reflected light is measured by the sensor's receiver. With b representing the horizontal distance between an emitter and the optical axis of a receiver (baseline length), f the focal length and p the position where the reflected light hits the image, the distance to the object can be calculated using Eq. (2.2).

$$D = f \cdot \frac{b}{p} \tag{2.2}$$

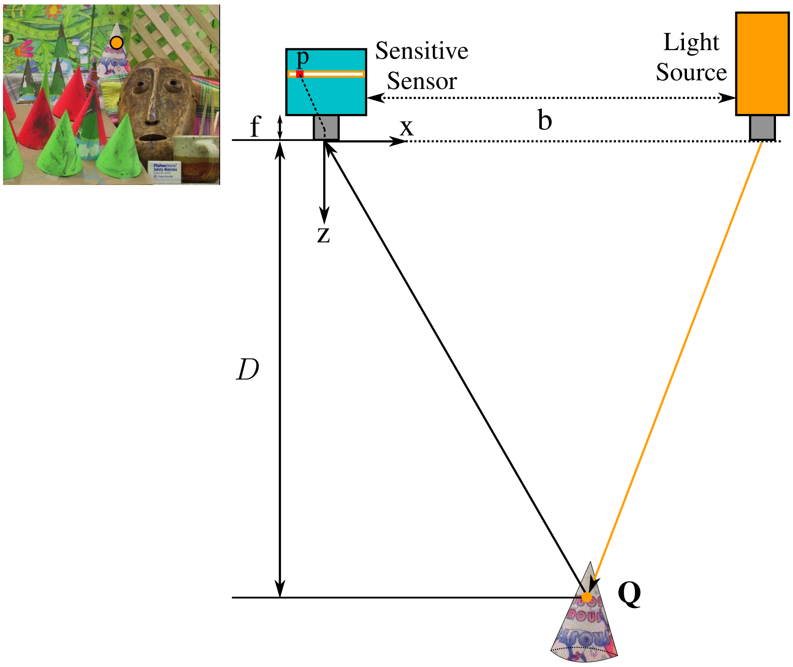


Figure 2.2: Active triangulation for depth computing. Point **Q** can be recovered avoiding point correspondences.

Such a simple sensor configuration restrains the distance measurement capability to a single point. To determine the shape of an object, either the sensor or the object itself must be moved and several measurements have to be taken. More sophisticated triangulation sensors use 2D light patterns and 2D cameras (e.g. the Kinect v1 sensor [26]). They project a light pattern onto the object, which is received by the camera system. These systems directly provide range images, where depth is given for every pixel. Those images are also known as 2.5D images.

2.2.3 Time of Flight

A Time of Flight camera is a sensor that gives, for every pixel, the distance between the camera and the scene in a particular direction. As a result, the camera produces a range image of the scene. The original idea for obtaining depth in ToF cameras consists of measuring the time-of-flight of a signal that travels from a light source, collides in the scene and returns back to a sensing element. Several technologies address depth computation from a different perspectives. We distinguish three categories: *i*) pulsed modulation, *ii*) continuous wave modulation (CWM) and *iii*) pseudo-noise modulation. The most established architecture is based on CWM and

Photonic Mixer Device (PMD) technology that obtains depth from the phase shift between two signals: *i*) square periodic signal used for infrared light modulation and *ii*) light reflected from the scene, that camera pixels convert into an electrical signal (see [1]). In Fig. 2.3 we show a general diagram of the distance recovery using ToF technology.

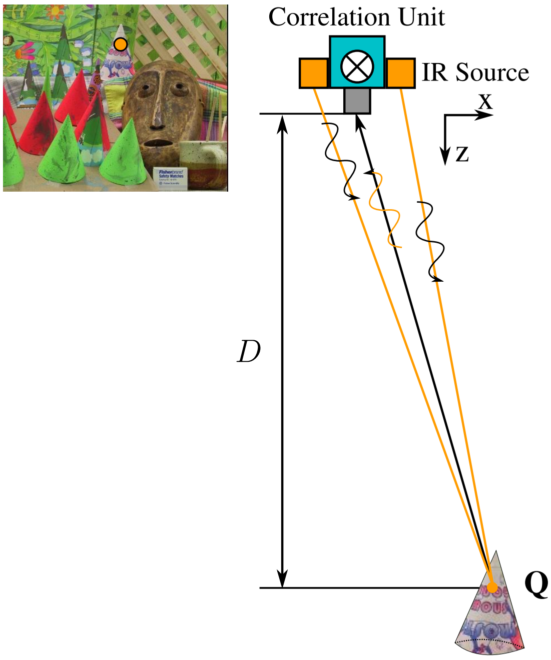


Figure 2.3: Time of flight technology for depth computing. Point **Q** can be recovered measuring the phase shift between the emitted and the received signals.

PMD ToF cameras are based on the so-called *4 phase-shift* system where each pixel of the ToF sensor includes a 2-well *Complementary Metal Oxide Semiconductor (CMOS)* architecture. Each well integrates samples of the correlation function between the incoming reflected *infrared (IR)* signal and the reference signal used for light modulation. The main idea is that if we sufficiently sample the correlation between reference and incoming signals, one can recover depth with invariance to gain and offset variations of the signal. Current PMD ToF cameras sample the correlation function at 4 different phase shifts $\{\varphi_0, \varphi_{\pi/2}, \varphi_\pi, \varphi_{3\pi/2}\}$. From these samples the depth D is obtained as follows:

$$D = \frac{c}{4\pi f_{\text{mod}}} \cdot \arctan\left(\frac{\varphi_{3\pi/2} - \varphi_{\pi/2}}{\varphi_0 - \varphi_\pi}\right), \tag{2.3}$$

where $c = 3 \cdot 10^8 {}^{m}\!/\!_{s}$ is the speed of light and f_{mod} is the modulation frequency.

The maximum unambiguous depth range (D_{max}) in ToF technology is determined by the frequency used for modulation (see Eq. 2.4). For any point further than D_{max}, for instance $D + n\ D_{\text{max}}$, the measured distance would be D. This effect is called *phase wrapping*, and n refers to the number of wrappings.

$$D_{\text{max}} = \frac{c}{2 f_{\text{mod}}} \tag{2.4}$$

2.2.4 Comparison of Depth Imaging Technologies

In Table 2.1 we show a comparison between the most established depth sensing technologies: triangulation and CWM ToF. We highlight in colours the key features of each technology: high performance features are shown in green while low performance features are shown in red. We use yellow to remark some the performance as average.

FEATURES	Passive Triangulation	Active Triangulation	CWM Time of Flight
Image Resolution	High Resolution	High Resolution	Mid Resolution ($< 512 \times 424$)
Frame Rate	< 25 fps	< 25 fps	< 90 fps
Depth Accuracy	Medium	High	Medium
Depth Range	Baseline Dependent	Scalable (Light Dependent)	Scalable
SW complexity	High	Medium	Low
Active Illumination	No	Yes	Yes
Required Texture	Yes	No	No
Shadowing effect	Yes	Yes	No
Correspondence Problem	Yes	No	No
Extrinsic Calibration	Yes	Yes	No
Compactness	Low	Medium	High
Power Consumption	Low	Medium	Medium/High
Material Cost	Low	Medium	Medium
Indoor	Yes	Yes	Yes
Outdoor	No	No	Yes

Table 2.1: Comparison between Triangulation a Time of Flight range sensing technologies.

The key benefit of CWM ToF against triangulation is that the former is a compact cost-effective depth imaging solution that provides *per-pixel* depth measurements in a fast and reliable way at the hardware level. ToF cameras are unaffected by varying environmental illumination, remove the shadowing effect and offers the possibility to work outdoor. This powerful combination makes PMD ToF sensors well-suited for a wide variety of applications, becoming a competitive alternative to other depth sensing technologies (see [27] for an overview).

2.3 Time of Flight Imaging

As described is Section 2.2.3, current ToF technologies can be classified into three categories: *i*) pulsed modulation, *ii*) continuous wave modulation and *iii*) pseudo-noise modulation. In this thesis we focus on the most established architecture: PMD-based CWM technology. In PMD sensors each *smart pixel* samples the correlation between emitted and received light signals. Current PMD cameras compute eight correlation samples per pixel in four sequential stages to obtain depth with invariance to signal amplitude and offset variations. Each pixel integrates charge during hundreds of nanoseconds on each stage, which provides sub-centimetre accuracy at a high frame rate.

2.3.1 Working Principle

Every CWM ToF camera is composed of four basic elements: *i*) an incoherent IR light source, *ii*) a frequency modulator, *iii*) a CCD/CMOS array of correlation pixels and *iv*) a phase shift module. The active light source is modulated using a continuous wave modulator. This light doesn't need to be coherent since no interference is needed for the measurement. Instead, the signal amplitude is modulated using a fixed frequency f_{mod}. Hence, non-specific light sources need to be incorporated. Currently, most of the commercial ToF cameras use different configurations of *Light Emitting Diode (LED)* for the illumination module. In the last few years, researchers have introduced different light sources that provide better properties (e.g. vertical-cavity lasers) or enhanced photo-detectors (e.g. *Single Photon Avalanche Diode (SPAD)* pixel structure), that are still under research.

Therefore, a CWM ToF camera emits a modulated light that collides with the scene, gets reflected and travels back to the sensing elements of the camera, where it is converted into an electrical signal. Through an optical system, the wavefront is imprinted into the sensor which is responsible for the correlation and demodulation step. The emitted light can be modelled theoretically as the following sine wave:

$$e(t) = A_0 \sin\left(2\pi f_{\text{mod}} t\right), \tag{2.5}$$

where f_{mod} is the modulation frequency and A_0 is the emission mean power.

The received signal has a different phase than the emitted one, defined in Eq. (2.5). Thus we can express it as:

$$r(t) = o(t) + A_0 K \sin\left(2\pi f_{\text{mod}} t - \beta\right), \tag{2.6}$$

where $o(t)$ is an offset that models the received optical power from background illumination, K is an attenuation factor due to the propagation and reflection of light (amplitude decay, object reflectivity...) and the optics (lens, filters...) and β is the phase shift that contains depth information. Usually $o(t)$ is considered constant, as background illumination has a significantly lower frequency than the one used by ToF cameras, and thus $o(t) = O_0$.

From the phase shift β, the depth D is obtained as follows:

$$\beta = 2\pi f_{\text{mod}}\ t_d = 2\pi f_{\text{mod}} \frac{2D}{c}, \tag{2.7}$$

where t_d is the time taken by the light to go and to return from the scene.

As the phase shift cannot be measured directly, ToF cameras implements correlation techniques inside each element of the sensor (on-chip correlation). Each *smart pixel* performs the correlation between the received optical echo $r(t)$ and a reference signal which is aligned in phase with the emitted signal $e(t)$. The correlation sample is obtain several times during an integration time T_{int}. Hence, a ToF camera doesn't provide neither a direct measure of the depth nor a measure of the time-of-flight, but a measure of the correlation between the reference and the received signals.

Let $u_m(t)$ be the square periodic signal used as reference for the correlation process. Then, we can define the correlation function ρ_τ between $u_m(t)$ and $r(t)$ and for a specific phase shift τ, as:

$$\rho_\tau = K_{\text{int}} \frac{1}{T} \int_0^T u_m(t+\tau) r(t)\ dt = K_{\text{int}} \left(\frac{A_0}{\pi} \cos\left(\beta + \tau\right) + \frac{O_0}{2} \right) \tag{2.8}$$

where A_0, β and O_0 are the unknown. K_{int} takes into account the total number of signal periods T integrated over the integration time of the sensor T_{int}, which is thousand times bigger than T.

The correlation function contains information about the received signal: the amplitude of the modulation, the intensity of the signal, the accumulated offset and the phase shift. Sampling the correlation function in a few points, the parameters above mentioned are inferred.

Each PMD *smart pixel* contains 2 CMOS wells that accumulate voltage corresponding to samples of the correlation function ρ_τ and $\rho_{\tau+\pi}$. Let U_τ^A and U_τ^B be the voltage accumulated in each well of the CMOS corresponding to ρ_τ and $\rho_{\tau+\pi}$, respectively. Taking the difference, we get:

$$\varphi_\tau = U_\tau^A - U_\tau^B = G \left(K_{\text{int}} \frac{2A_0}{\pi} \cos\left(\beta + \tau\right) \right) + O = A \cos\left(\beta + \tau\right) + O \tag{2.9}$$

where G represents the systematic gain and O is the offset term. There are thus A, β and O as unknowns, whereas only β contains depth information.

In Fig. 2.4 we show a cross-section of a PMD pixel where charges are balanced depending on the reference signal used for modulation $u_m(t)$.

Current PMD devices implement the *4 phase-shift* architecture, which consists on a 4-stage pipeline where φ_τ is computed at $\tau = \{0, \pi/2, \pi, 3\pi/2\}$ in 4 sequential stages. In Fig. 2.5 we show a cross-section of a PMD-based *smart pixel* with the acquisition pipeline.

Provided that offset variations can be eliminated, we can observe from Eq. (2.9) that the following relationships are satisfied: *i)* $\varphi_0 = -\varphi_\pi$ and *ii)* $\varphi_{\pi/2} = -\varphi_{3\pi/2}$. This temporal redundancy between φ_0-φ_π and $\varphi_{\pi/2}$-$\varphi_{3\pi/2}$ is used in the *4 phase-shift* algorithm to cancel the effect of gain and offset variations in the captured signal (see Eq. 2.10a). Usually, the value of φ_τ is accessible from the sensor instead of U_τ^A and U_τ^B.

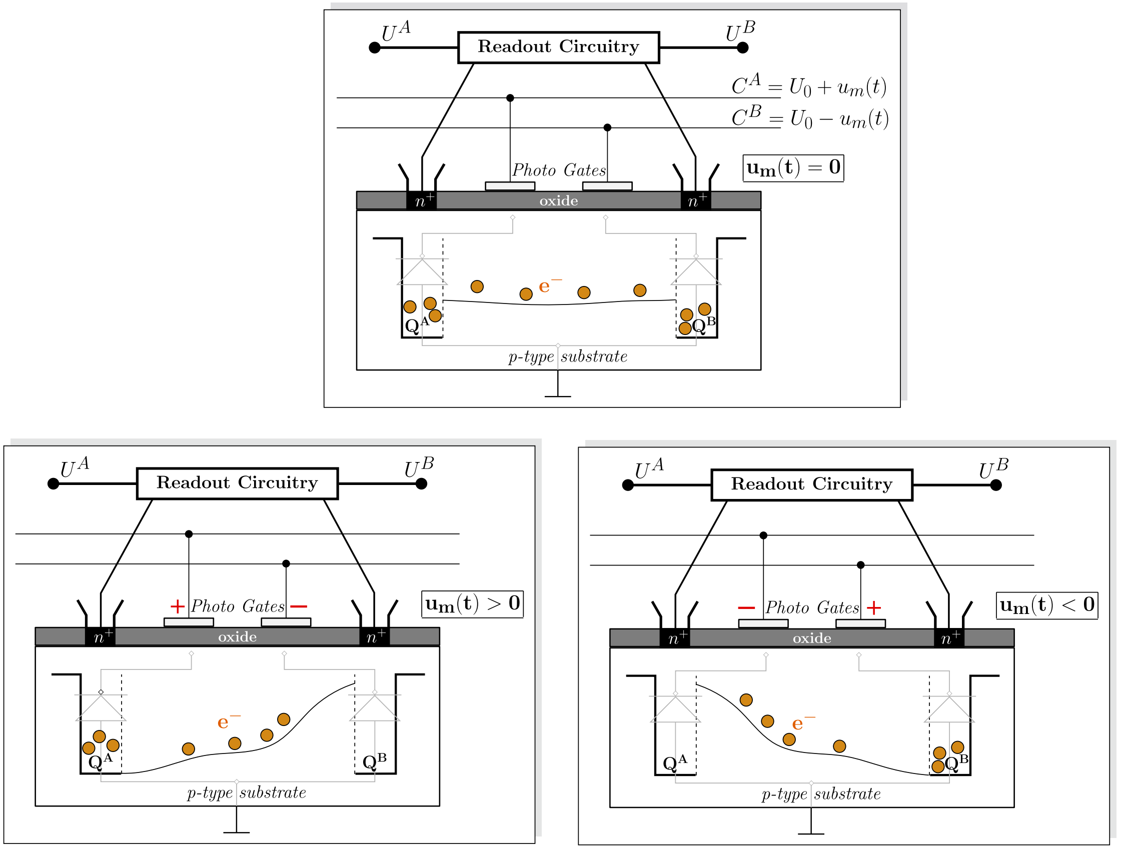


Figure 2.4: Cross-section of a PMD pixel architecture for different values of the reference signal $u_m(t)$. Q^A and Q^B are the accumulated charges in each well.

Depth D, amplitude A and offset O are then obtained:

$$D = \frac{c}{4\pi f_{\text{mod}}} \arctan\left(\frac{\varphi_{3\pi/2} - \varphi_{\pi/2}}{\varphi_0 - \varphi_\pi}\right) \tag{2.10a}$$

$$A = \frac{\sqrt{\left(\varphi_{3\pi/2} - \varphi_{\pi/2}\right)^2 + \left(\varphi_0 - \varphi_\pi\right)^2}}{2} \tag{2.10b}$$

$$O = \frac{1}{8}\left(U_0^A + U_0^B + U_{\pi/2}^A + U_{\pi/2}^B + U_\pi^A + U_\pi^B + U_{3\pi/2}^A + U_{3\pi/2}^B\right). \tag{2.10c}$$

To summarise, in Fig. 2.5 we show the pipeline in the generation of ToF measurements. We first illuminate the scene using an IR illumination unit using CWM technique. The round-trip signal is captured by the PMD sensor where the correlation with a reference signal is performed. Sampling the correlation function at 4 different phase shifts, each sensing element is able to compute depth while removing gain and offset variations.

Depth from Eq. (2.10a) doesn't include several error sources present in ToF cameras and that need to be taken into consideration to obtain accurate depth measurements. We review in the next sections systematic and non-systematic errors in ToF cameras.

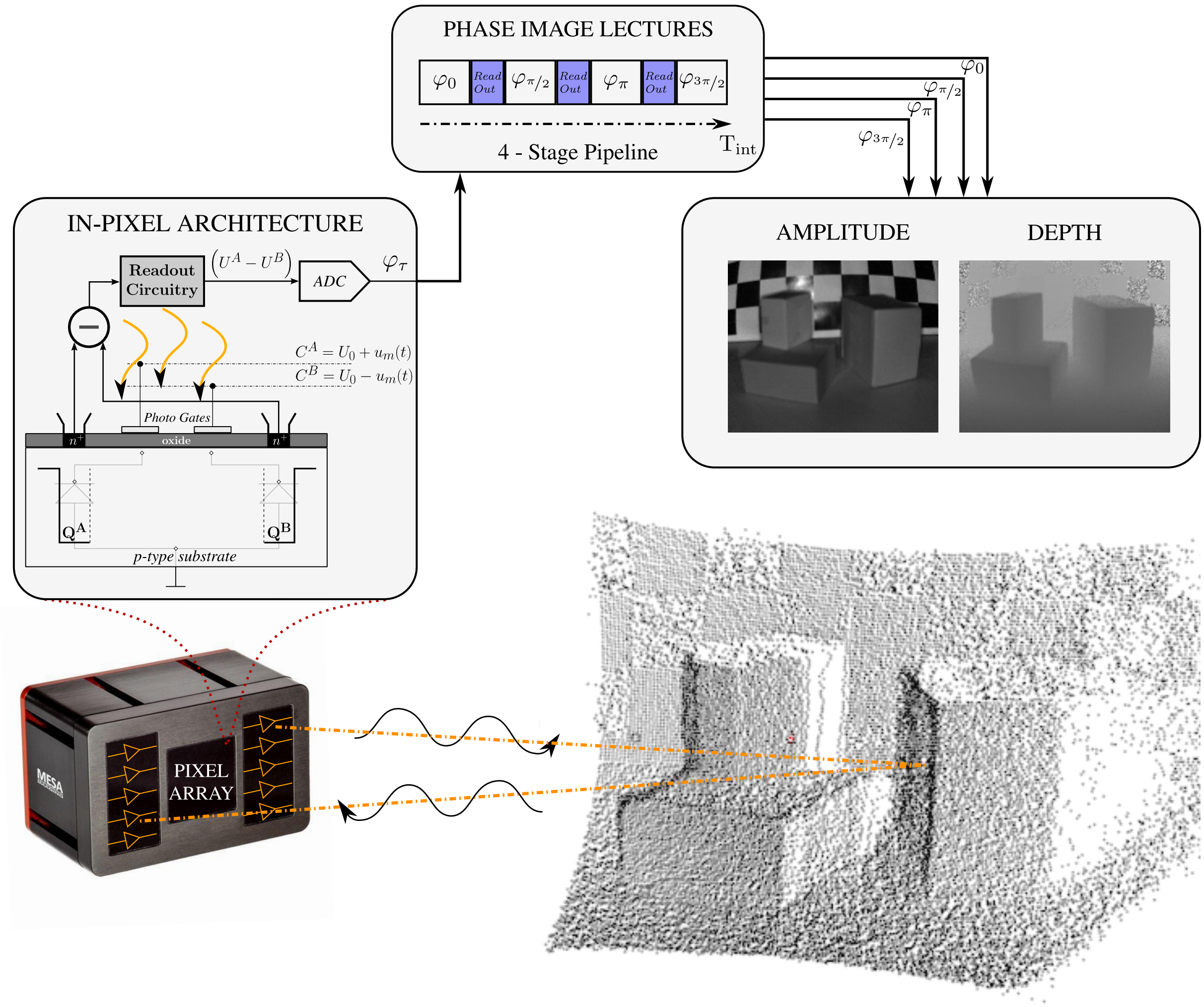


Figure 2.5: General diagram of the ToF imaging process. A cross-section of the pixel architecture is presented, where Q^A and Q^B are the accumulated charges in each of the two well. Notice that the capture of every phase image is followed by a readout gap.

2.4 Systematic Errors in ToF cameras

Systematic errors in depth cameras have a strong impact regarding precision and reliability of depth measurements. Hence, calibration of these errors has been extensively studied in the scientific literature. Usually, ToF systematic errors are originated during the process of light to signal conversion and sensor manufacturing. Each depth sensing technology suffers from different systematic errors. This has motivated calibration methods that are exclusive to each particular technology.

Among systematic errors in ToF cameras we highlight: *i*) wiggling error due to the use of non-ideal sinusoidal signals, *ii*) pixel-related error arising from a *Fixed Pattern Noise (FPN)*, *iii*) amplitude-related errors due to the non-uniformity of the IR illumination and reflectivity variations, *iv*) photon shot noise that influences the number of collected electrons during the integration time and *v*) temperature-related error due to its influence in the semiconductor's properties (see Fig. 2.6 for an overview). Nowadays, most of commercial cameras include hardware compensation for many of these errors. From among all, we focus on the effect of *systematic depth distortion* which is mainly caused by the assumption of idealistic sinusoidal modulation that, in practice, is not perfect: the so-called *wiggling error*.

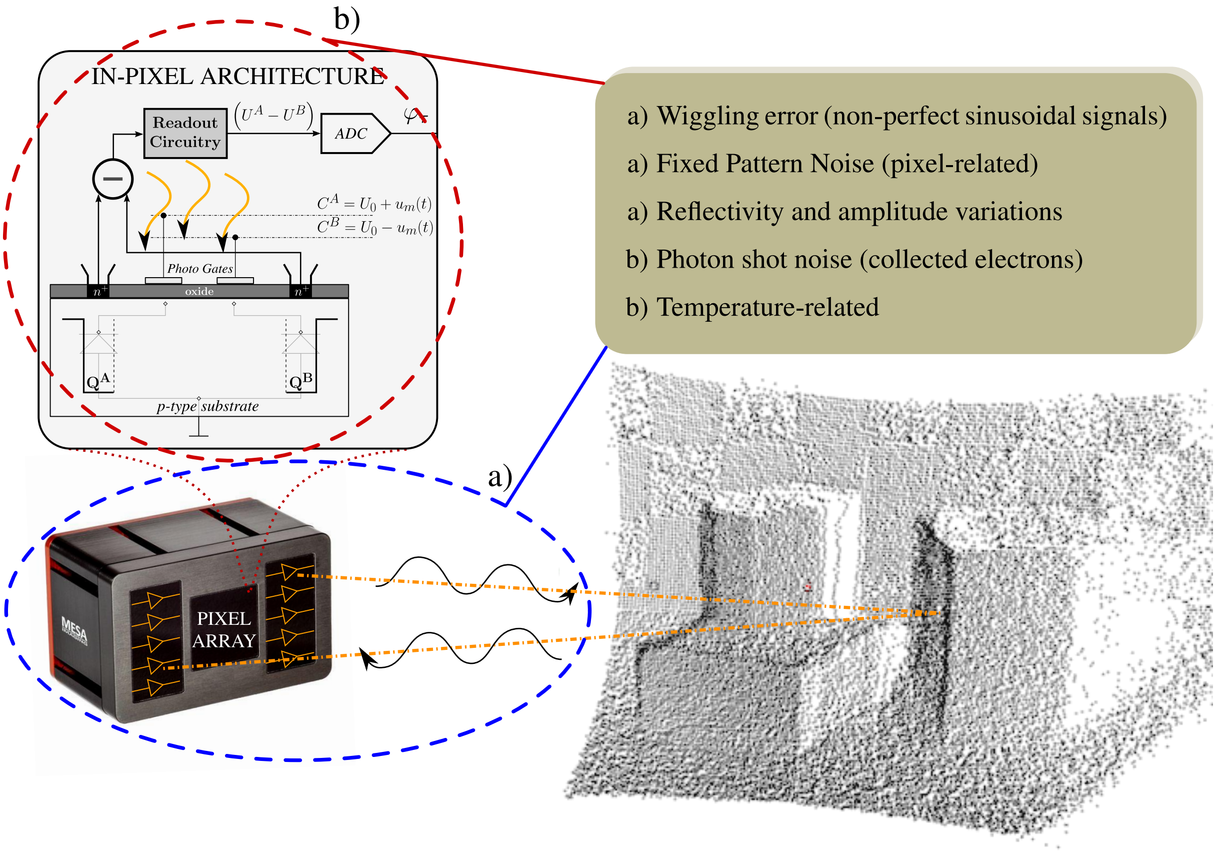


Figure 2.6: Systematic errors in CWM ToF cameras. An overview.

2.4.1 Calibration of Systematic Depth Distortion

Calibration of systematic depth distortion in ToF cameras has been extensively studied in the literature [17]. Some methods require complex hardware to calibrate the camera. For instance, Fuchs et al. [28, 29] propose a method to compensate depth distortion using checker-boards and a robotic arm that holds the camera at its tip. The robotic arm obtains the camera pose at each time instant using internal sensors. Camera pose is then used during calibration. In addition, checker-boards or LED array patterns, that are usual for calibration of intensity cameras, do not lead to very accurate calibration in depth cameras [17], as accurate point correspondences are difficult to obtain in depth sensors. This is mainly due to the low resolution of the sensor, its low contrast and depth artifacts that usually appear around strong intensity edges. Other methods require additional sensors, such as high accuracy track lines ([30,31]) or a colour camera ([32,33]), which makes these methods difficult to apply for a wide range of applications. Kahlmann et al. [30] estimate the distortion function that relates systematic depth distortion with integration times and depth. The distortion function is roughly represented by a Look-Up-Table. Lindner et al. [31] use a B-Spline to accurately model depth distortion and in [32] the same author improves the previous method using two-dimensional B-Splines to describe distance deviations more accurately. Very recently, Hussmann et al. [34] propose a modulation method based on sine waves that describes the effect of the wiggling error and significantly reduces the depth distortion. However, it requires hardware modifications in the camera.

Recently, Belhedi et al. [35] propose to calibrate systematic depth errors without any prior knowledge of the underlying physical model. This method is thus applicable for any depth sensor and reduces the complexity of both the setting and the equipment. Provided that we have multiple views of a plane within the sensor depth range, they propose to find the depth correction function forcing the observations to be planes. An iterative process is proposed that alternates between two optimisation steps: *i*) calculate the planes that best fit the estimated data and *ii*) estimate the correction function from the measured surfaces to planes obtained in *i*). The method iterates until no correction is needed (the error is below a threshold) or a maximum number of iterations is achieved. The authors avoid the use of calibration patterns that do not lead to precise calibration in depth cameras due to sensor's limitations. However, their method need a few correspondences to fix a 3D affine ambiguity. Despite the authors report very accurate results comparing with the state of the art, the main limitation of [35] is the need of solving a non-convex optimisation problem. The authors proposed a greedy solution that does not guarantee one to reach a global minimum and is computationally expensive.

To conclude, most of previous approaches are not applicable to a general depth sensor and some of them need additional sensors for obtaining ground truth data or calibration patterns that are difficult to engineer [28–32]. On the other hand, the method proposed by Belhedi et al. [35] has, as main limitation, that it requires to solve a non-convex optimisation problem whose convergence is not guaranteed. We show in this thesis that this has an impact in the accuracy.

2.5 Non-Systematic Errors in ToF cameras

Several non-systematic errors have been studied in the literature. We highlight: *i*) flying pixels at depth discontinuities, *ii*) depth ambiguity, when depth range involves more than one period of the modulation signal, *iii*) light scattering within the lens and the sensor, *iv*) multipath interference and *v*) motion artifacts (see Fig. 2.7 for an overview). In this thesis, we highlight *multipath interference* and *motion artifacts* as the most challenging problems that have received the attention of the research community.

2.5.1 The Multipath Interference

Multipath Interference (MpI) [36] appears due to multiple reflections of the light: a mixture of incoherent light signals arrives at each pixel from different paths apart from the direct path. This effect causes significant distortions in the estimation of depth, specially in the vicinity of corners and convex areas of the scene (see Fig. 2.7).

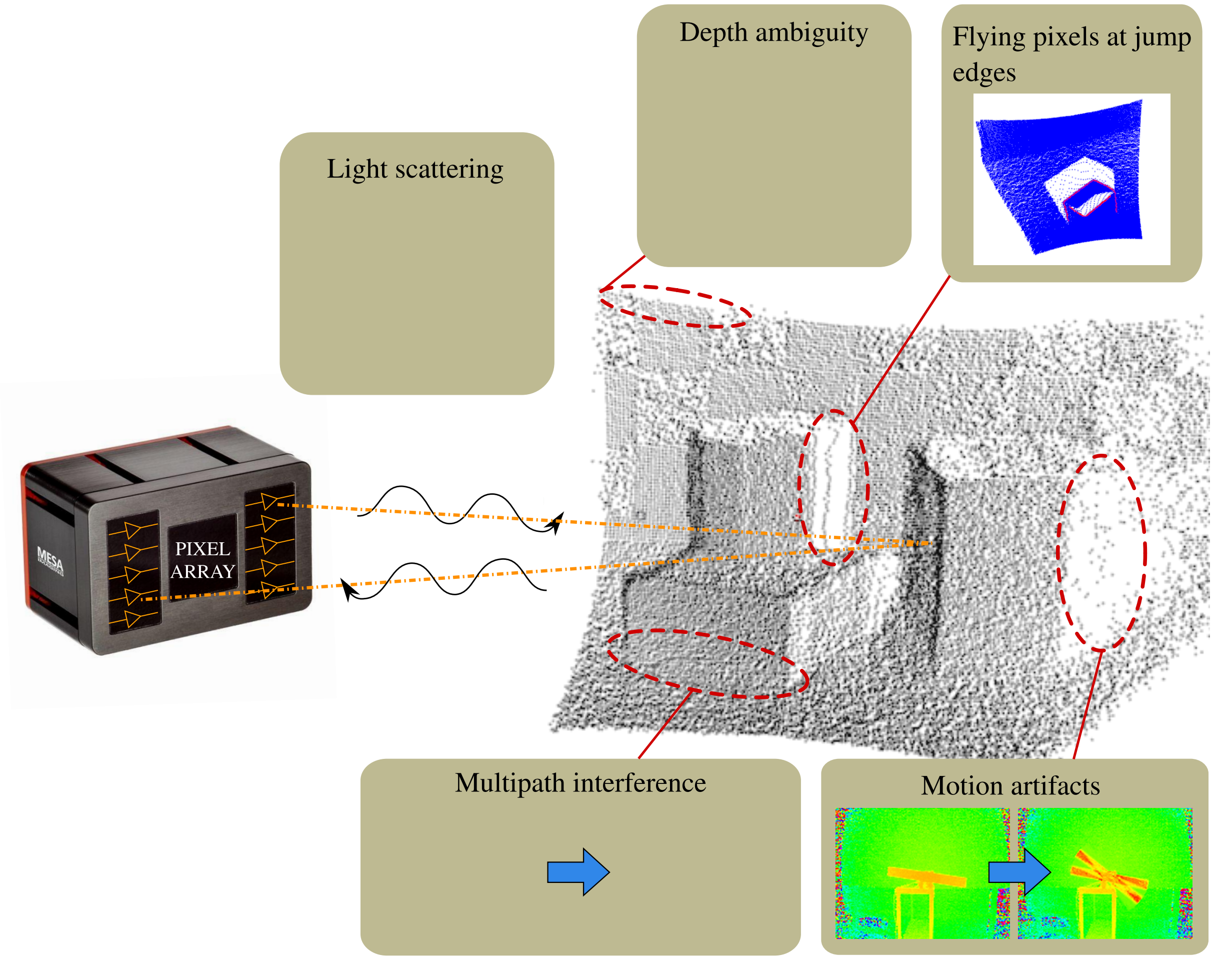


Figure 2.7: Non-systematic errors in CWM ToF cameras. An overview.

In the literature there are several studies which show the impact of MpI in scenes registered by ToF cameras. In [36] Guomundsson et al. show how MpI severely distorts corner-like scenes. In addition, May et al. [37] measure the impact of MpI when building maps using mobile robots equipped with ToF cameras.

Several works in the literature propose hardware modifications in the ToF cameras to remove multipath distortion. Falie and Buzuloiu [38, 39] propose methods based on IR structured light. Dorrington et al. [40] propose a method to correct MpI using light with two different wavelengths. These works are not able to completely remove MpI and they require hardware modifications in the camera, which are not available in commercial sensors.

Fuchs [41] proposes a method to automatically correct MpI from the camera measurements that does not require hardware modifications. The author proposes a model that predicts the multipath distortion given the information captured by a ToF camera (i.e. signal amplitude and depth for each pixel). The main assumption of Fuchs [41] is that, despite the fact that measurements are contaminated with MpI, the signal distortion can be accurately predicted from them. The main drawback of Fuchs work is precisely the model used to predict MpI. It is not demonstrated under which circumstances the predicted MpI is accurate. In fact, Fuchs method fails in general scenes when the surfaces have different albedo. In addition, Fuchs' work requires the adjustment of scene-dependent parameters which is a serious limitation in practice.

There are no hints about how to set these parameters and which is their influence in the solution.

To conclude, most of previous approaches, provide slight reduction of the multipath distortion, while requiring hardware modifications or strong assumptions that limits its applicability.

The following recent works [42–47] appear after publication of our MpI correction method proposed in this thesis. Most of them cite our proposal as a reference of the state of the art and use some of our results.

The work of Fuchs et al. [42] is a generalisation of [41] to a spatially varying, unknown reflection coefficient that requires an iterative solution consisting on multiple passes. The remaining approaches [43–47] are inspired by [40], regarding the idea of using several modulation frequencies to separate the direct path of the signal from the contributions of indirect reflections. The works of Godbaz et al. [43] and Kirmani et al. [44] model a two-bounce multipath, assuming specular surfaces and requiring more than 3 modulation frequencies. Both methods work on a per-pixel basis, using closed form solutions that are expected to run close to real-time. Bhandari et al. [45] reduce MpI providing a general solution based on using several frequencies and holding for any number of bounces. The approach of Freedman et al. [46] can handle limited amount of diffuse bounces and it is limited to scenes where the dominant amount of global illumination is due to only a small number of light paths. The approach presented in [47] show that it's possible to perform the separation of light signals by capturing 3 measurements at a single high temporal frequency. They demonstrated their method by building a hardware prototype based on a low cost CWM ToF sensor. In summary, all these methods [43–47] require several modulation frequencies, that renders impracticable for most commercial ToF sensors. Despite newer ToF cameras, such as the Kinect v2, allow to work with 3 modulation frequencies, the camera's frame ratio is penalised and methods requiring more than 3 frequencies are prohibitive.

2.5.2 Motion Artifacts

High motion dynamics in the scene causes significant distortion in depth estimation, specially in the vicinity of depth discontinuities and points with strong texture changes. Motion artifacts are particular of each technology. This thesis studies their effect in PMD cameras (see Fig. 2.7).

Several studies show the impact of motion artifacts in ToF imaging [48–50]. Some of them propose methods to remove the artifacts. Lottner et al. [48] combine a ToF camera and a colour camera to accurately detect the edges of the scene, removing motion artifacts. Schmidt [49] analyses the origin and effects of motion artifacts in PMD-based ToF cameras and proposes a method to detect pixels affected by motion. Schmidt [49] also proposes a new hardware architecture to reduce motion artifacts that consists of a 4-well CMOS sensor that gets simultaneously 4 correlation samples. In that manner, the total integration time is reduced as it uses a single stage. Schmidt's [49] architecture is nowadays prohibitive and sensor manufacturers mainly produce 2-well sensors. More recently, Lee et al. [51,52] study the physical phenomena that generates motion artifacts and propose some methods to detect them. Furthermore, they propose a simple method to correct the artifacts that require user interaction.

Recent works [50–55] detect and remove automatically motion artifacts by exploiting consistency between phase images in PMD-based architectures. These works can be separated into single frame and multi frame approaches.

Single frame approaches exploit the fact that in existing PMD cameras phase images are sequentially obtained, revealing motion. Hussmann et al. [50] detect motion artifacts from discrepancies between phase images. Motion compensation is proposed only for lateral movements in a conveyor belt. This method requires photometric calibration of the camera, is restricted to a very close range (distances below 100 cm) and is vulnerable to intensity changes by different pixels. Lindner et al. [53] distinguish between lateral and axial motion. In the former, artifacts are compensated by computing dense optical flow between consecutive *intensity phase images*. Intensity phase images are obtained from raw CMOS measurements of each pixel that need to be accessible from the camera (e.g. phase images are computed from the differences of raw images). Optical flow is then used to correct phase images, removing motion artifacts. To correct axial motion, Lindner et al. [53] need two consecutive frames that are individually corrected from lateral motion artifacts. This proposal is time demanding and it has to deal with the inherent problems in optical flow estimation (normalisation of phase images, aperture problem...). Lefloch et al.'s proposal [56] is an improvement of Lindner et al.'s [53] proposal that reduces the number of flow calculations. As it is stated by the authors, optical flow computation is a very time demanding process that requires GPU hardware to be real-time. Very recently Hoegg et al. [55] present a flow like algorithm which uses particular parts of Lindner et al. [53] (flow field) and Lefloch et al. [56] (binarization of the motion area with an experimental threshold). The proposal simultaneously estimates the motion field in motion areas and corrects the artifacts by applying the computed flow field directly to the raw phase values of each individual CMOS channel. As in Lindner et al. [53] the approach does require a phase normalisation step in order to apply the algorithm. The motion direction estimation is based in local consistency and it has a time complexity of $\mathcal{O}(n^2)$ that restricts the local neighbourhood radius search to achieve real time. To overcome this problem, they propose to use a previous frame to initialise the search direction assuming linear motion. A median filter is optionally applied to remove outliers during flow field optimisation. All existing single frame methods have in common that they require to have access to each individual channel of the ToF CMOS sensor.

In multiple frame methods, temporal consistency between frames is exploited to correct motion artifacts. Schmidt [54] detects inconsistent motion events between the different stages of the *4 phase-shift* pipeline using two consecutive frames. The main assumption is that only one motion event can occur within 2 consecutive frames. This assumption is violated easily in many cases considering fast movements in the scene. In practice, this method has some limitations and raises important issues about reading gaps between consecutive frames, that in many cases can be much larger than the integration time.

To conclude, single frame approaches have specific design limitations: while they require to have access to both channels of each individual PMD pixel, manufacturer's trend is just to provide the difference. Moreover, methods considered as single frame propose also to use previous frames to improve efficiency [55] or to correct axial motion artifacts [53]. Contrary, multiple frame approaches rely heavily on camera's readout times or propose hardware modifications, that are not feasible for commercially available ToF cameras.

2.6 Conclusion

In this chapter we show the importance of correcting ToF-related error sources. As it is shown in the previous sections, correction of errors in ToF cameras is still an open problem.

State of the art methods reduce the impact of systematic depth distortion from a calibration perspective. In general, these methods need calibration patterns, that are difficult to engineer, additional sensors to acquire precise ground truth measurements or do not have guarantees of convergence. Thus, these methods show some practical limitations that must be solved for a final user.

Regarding non-systematic errors, the literature counts with several solutions to detect and to correct multipath interference and motion artifacts. Due to the complexity of these phenomena, state of the art methods present several restrictions that limits its applicability. Most of them require hardware modifications, specific equipment or special setups that are prohibitive for most of the users. This thesis presents methods for the correction of the aforementioned errors in CWM ToF cameras. The methods are conceived with the following criteria: *i*) to avoid neither hardware modifications nor additional sensors, *ii*) to reduce software complexity, *iii*) to avoid complex settings and *iv*) to find efficient implementations. The approaches shown in the next chapters improve the state of the art methods in all or most of these aspects.

Chapter 3

A Closed-Form Solution based on Local Planarity for the Calibration of Depth Distortion in Depth Cameras

3.1 Introduction

In this work we propose a calibration method that corrects systematic depth distortion in depth cameras based on showing the camera a plane in different positions and orientations. We call this method CaDDiP: *Calibration of Depth Distortion from Planes*. This approach was first proposed by Belhedi et al. [35], where they propose a solution based on non-convex optimisation. Our main contribution is to show that finding the correction function can be cast of a convex problem. Convexity is gained by reformulating the constraints on the correction function using differential properties of planes, that we refer to as the local planarity constraint. This formulation allows us to calibrate a depth camera with linear least-squares by representing the correction function with a *Linear Basis Expansion (LBE)*[1]. In the proposed method, data is virtually free, as planes are ubiquitous in man-made environments, providing a high amount of information. As opposed to plane based camera calibration [57] or self-calibration [58], we do not need for correspondences. We thus obtain the correction function very efficiently. Our method is very accurate. We tested it with both simulated and real experiments using different depth sensing technologies. Our method improves on [35], especially in the presence of noise.

3.2 Overview

Current depth cameras suffer from multiple error sources that notably degrade their accuracy. Some of these errors are systematic and can be reduced with calibration. Unlike in colour cameras, calibration patterns for depth cameras are difficult to engineer and usually require special equipment. As a consequence, no standard calibration method is currently available. We propose a calibration method based on showing planes to the camera. Our method does not

[1] Such as Thin Plate Spline, the tensor-product B-Spline, finite elements or finite differences.

require a calibration pattern: the planes may be arbitrary, and their model is not required. In addition to the planes this method needs a minimum of 2 anchor points[2] to fix a scale ambiguity. Depth distortion produces curved surfaces as measurements, instead of planes. Our calibration procedure finds the depth correction that forces the observations to be planar. This problem has been studied before using non-convex optimisation. Existing methods are sub-optimal and do not ensure accurate calibration results. We show that this problem can be reformulated as a linear least-squares (and thus convex) problem using differential planarity constraints. Our method improves the state of the art in terms of accuracy, simplicity and computational complexity. We validated our method using simulated and real experiments with three commercial depth cameras (MESA SR4k, PMD CamCube 3 and Microsoft Kinect v1).

Figure 3.1 shows our method's pipeline. First, we capture several plane views (by plane view, we mean any image, or the part of an image, of a planar surface) covering the camera's workspace. Next, we compute a global correction function (φ) based on enforcing differential constraints of planes.

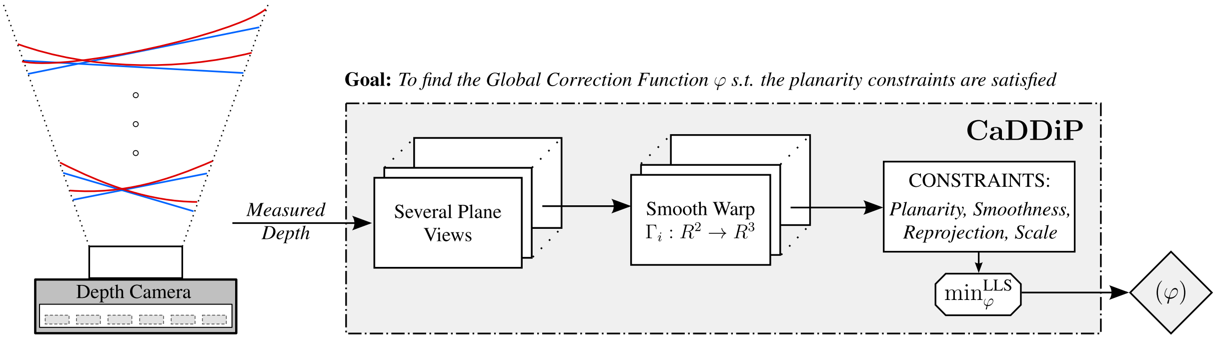

Goal: To find the Global Correction Function φ s.t. the planarity constraints are satisfied
CaDDiP
Measured Depth
Several Plane Views
Smooth Warp $\Gamma_i : R^2 \to R^3$
CONSTRAINTS: Planarity, Smoothness, Reprojection, Scale
$\min_{\varphi}^{\text{LLS}}$
(φ)
Depth Camera

Figure 3.1: Method pipeline for the calibration of systematic depth distortion in depth sensors. We show in colours i) the distorted input (red curves) and ii) the corresponding ground truth (blue lines).

This chapter is structured as follows. Notation is given in Section 3.2.1. A review on previous works regarding *Structure Light Projection (SLiP)* technology and that compliments Section 2.4.1 (previous works on ToF technology), is given in Section 3.3. The problem statement with regards to plane-based calibration and existing approaches are given in Section 3.4. The mathematical modelling of the proposed method is given in Section 3.5. A description of the algorithm and the implementation details are given in Section 3.6. Simulated and real experiments are given in Section 3.7. Finally, conclusions are discussed in Section 3.8.

3.2.1 Notation

Scalar real values are typeset in regular italics lowercase, e.g. d. Domains of functions are typeset in calligraphic uppercase, e.g. $\mathcal{D}$. Vectors are displayed with bold letters. In particular, two-dimensional points are usually written $\boldsymbol{m}$ or $\boldsymbol{p}$ if they belong to the image plane. Points $\in \mathbb{R}^3$ are represented using $\boldsymbol{q}$. Vectors of dimension $\mathbb{R}^l$, $l > 3$, are represented using different

[2] Anchor points are measured points whose depth is known

bold lowercase letters, e.g. $\boldsymbol{v}$. Matrices are typeset in bold uppercase math, e.g. $\mathbf{D}$.[3] Hyperparameters are denoted by the Greek lowercase letters λ, e.g. λ_a. Functions are denoted by different Greek letters, such as φ. Functional use brackets (e.g. $\varepsilon_s[\varphi]$). A bar over a vector represents its homogeneous coordinates, e.g. $\bar{\boldsymbol{p}}$.

3.3 Previous Works on SLiP technology

In Chapter 2 we presented in Section 2.4.1 state of the art methods regarding systematic depth distortion in *Time of Flight (ToF)* cameras. Nevertheless, the proposed approach is applicable to other depth sensor technologies such as SLiP (see Section 2.2.2 for an overview). Resolution and accuracy of these sensors increase every year, which makes them a promising hardware alternative to stereo vision systems. However, there are several error sources that affect depth measurements in SLiP technology. Some of these errors are systematic and can be reduced with calibration. Systematic errors in SLiP cameras are usually generated due to geometric misalignment between the projector and the sensor and the dependence of the pattern density to depth. Unlike in colour cameras, calibration patterns for depth cameras are difficult to engineer and usually require special equipment. As a consequence, no standard calibration method is currently available.

SLiP cameras show the following systematic error sources: *i*) camera calibration errors (focal length, principal point offsets and lens distortion coefficients), *ii*) geometric misalignment between the camera and the projector and *iii*) non constant depth resolution due to the decreasing point density with the squared distance. Depth calibration in SLiP cameras has been addressed in the literature by combining modelling of systematic depth distortion, lens calibration and infrared-RGB image alignment. Khoshelham et al. [59] study error sources in the Kinect v1 sensor and analyse its accuracy, theoretically and experimentally. They also propose calibration methods where lens and depth distortion calibration are treated separately. In [60] the same authors include the effect of depth quantisation in the calibration. Herrera et al. [61] propose a calibration method for depth distortion based on iterative refinement. Initially, the intrinsic parameters of the camera and the plane pose are estimated. The authors reports very accurate results but the optimisation step is very time demanding. Raposo et al. [62] improve [61] by reducing its computational demand. Smisek et al. [26] minimise the residuals of a best fit plane considering both the distortion in the projection and in the depth estimation. More recently, Chow et al. [63] show a bundle adjustment method to fully calibrate Kinect v1 sensors including the estimation of the intrinsic and the extrinsic parameters of both the infrared and the colour cameras and, more importantly, the compensation of the systematic depth distortion.

The aforementioned previous works are technology-specific approaches that are not applicable to both SLiP and ToF sensors. Moreover, they require calibration patterns or additional sensors that are difficult to handle in depth images. This thesis proposes a calibration method which is applicable to any depth sensor and is based on showing planar structures to the depth camera.

[3] Each of the elements of vectors and matrices are represented using sub-indexes (e.g. $\boldsymbol{q} = \begin{pmatrix} q_x & q_y & q_z \end{pmatrix}^\top$).

3.4 Plane-based Calibration: Problem Statement and Existing Approaches

3.4.1 Problem Statement

We define a general depth sensor with a retinal plane $\mathcal{S} \subset \mathbb{R}^2$ and a centre of projection. Depth is measured along optical rays that intersect the retinal plane and originate at the centre of projection. A depth image is defined theoretically as the scalar function $\mathcal{D} : \mathcal{S} \to \mathbb{R}$, where given a point $\boldsymbol{p} \in \mathcal{S}$, $\mathcal{D}(\boldsymbol{p})$ denotes the depth measured in the direction of the optical ray corresponding to $\boldsymbol{p}$. We assume that the intrinsic parameters of the camera, including lens distortion, were calibrated, and that the image coordinates are those of the retinal $\mathcal{S}$ that are undistorted and normalised by the matrix of intrinsic parameters. The 3D coordinates $\boldsymbol{q} = \begin{pmatrix} q_x & q_y & q_z \end{pmatrix}^\top$ of point $\boldsymbol{p} \in \mathcal{S}$ are given by:

$$\boldsymbol{q} = \mathcal{D}(\boldsymbol{p}) \cdot \frac{\bar{\boldsymbol{p}}}{\|\bar{\boldsymbol{p}}\|_2}. \tag{3.1}$$

For a 3D point $\boldsymbol{q}$ we have that $\mathcal{D}(\boldsymbol{p}) = \|\boldsymbol{q}\|_2$ with $\boldsymbol{p} = \Pi(\boldsymbol{q})$, and Π the perspective projection function. In the presence of depth distortion the measured depth-map $\widetilde{\mathcal{D}} \neq \mathcal{D}$ (including additional noise).

We define the calibration function $\varphi \in \mathcal{C}^\infty(\mathbb{R}^3, \mathbb{R}^3)$ as a smooth function that takes any point in 3D and returns its corrected coordinates:

$$\mathcal{D}(\boldsymbol{p}) \cdot \frac{\bar{\boldsymbol{p}}}{\|\bar{\boldsymbol{p}}\|_2^2} = \varphi\left(\widetilde{\mathcal{D}}(\boldsymbol{p}) \cdot \frac{\bar{\boldsymbol{p}}}{\|\bar{\boldsymbol{p}}\|_2^2}\right). \tag{3.2}$$

We highlight that φ has a 3D domain as systematic errors vary over depth.

Our objective is to obtain φ by observing collection of planes and selecting φ as the function that makes the distorted observations planar. In Fig. 3.2 we show a cross-section of the input data-set consisting of multiple planes views.

We assume that we show the sensor n 3D planes that are represented by the function $\Gamma_i \in \mathcal{C}^\infty(\mathcal{S}, \mathbb{R}^3)$ with $i \in \{1, \cdots, n\}$. For each plane we define as $\gamma_i \in \mathcal{C}^\infty(\mathcal{S}, \mathbb{R}^3)$ the corresponding 3D surface recovered by the depth sensor under distortion:

$$\gamma_i = \left\{\widetilde{\mathcal{D}}_i(\boldsymbol{p}) \cdot \bar{\boldsymbol{p}}, \quad \boldsymbol{p} \in \mathcal{S}\right\}, \tag{3.3}$$

where $\widetilde{\mathcal{D}}_i$ corresponds to the depth image of plane Γ_i.

Due to distortion γ_i is not planar but curved. We then have that $\varphi \circ \gamma_i = \Gamma_i$ for $i \in \{1, \ldots, n\}$. We propose to find φ as the solution of the following optimisation problem:

$$\text{Find } \textit{varphi} \text{ such that } \begin{cases} \varphi \circ \gamma_i & \text{is a plane} \\ \Pi \circ \varphi \circ \gamma_i = \Pi \circ \gamma_i \end{cases} \qquad i \in \{1, \cdots, n\}. \tag{3.4}$$

The condition $\Pi \circ \varphi \circ \gamma_i = \Pi \circ \gamma_i$ forces the correction to lie on the optical ray. In this way the corrected points project in the same retinal points than their distorted counterparts.

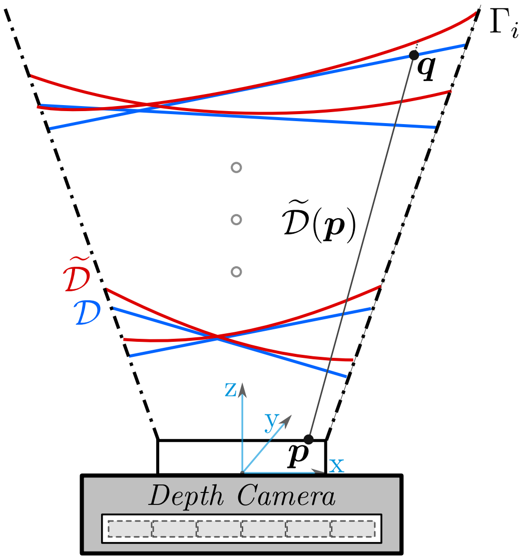


Figure 3.2: Initial setting cross-section example. We show in colours *i*) the distorted input (red curves) and *ii*) the corresponding ground truth (blue lines).

3.4.2 Belhedi et al.'s Method

To solve Eq. (3.4) we define the planarity condition of each corrected plane. A way that is explored in [35] is to include, jointly with φ, new variables for the unknown planes. The resulting problem is:

$$\text{Find } \varphi, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_n \text{ such that } \begin{cases} \boldsymbol{v}_i^\top \cdot \overline{(\varphi \circ \gamma_i)} = 0 \\ \Pi \circ \varphi \circ \gamma_i = \Pi \circ \gamma_i \end{cases} \quad i \in \{1, \cdots, n\}, \tag{3.5}$$

with $\boldsymbol{v}_i = \begin{pmatrix} v_{i,x} & v_{i,y} & v_{i,z} & d_i \end{pmatrix}^\top$ being the point-normal parameters of the i-th plane. As this method assumes a discrete amount of planes in Eq. (3.5), its solution is under-constrained for those points not belonging to the planes and it is non convex. The authors add to Eq. (3.5) the fact that φ is a smooth function as:

$$\min_{\varphi, v_1, \cdots, v_n} \mathrm{TV}_d^k[\varphi] \quad \text{such that} \quad \begin{cases} \boldsymbol{v}_i^\top \overline{(\varphi \circ \gamma_i)} = 0 \\ \Pi \circ \varphi \circ \gamma_i = \Pi \circ \gamma_i \end{cases} \quad i \in \{1, \cdots, n\}, \tag{3.6}$$

where TV_d^k is a *Total Variation* functional that penalises k-th order derivatives:

$$\mathrm{TV}_d^k[\varphi] = \int_{\mathcal{S}} \|D^k[\varphi]\|_d^2 \, \mathrm{d}\mathcal{S}, \tag{3.7}$$

with D^k being the k-th order derivative operator of φ and d the norm type (e.g. $d = 2$ for Euclidean distance, $d = 1$ for the L1 norm or $d = F$ for the Frobenius matrix norm). It is used TV_F^2 that is commonly known as the *Bending Energy*.

However, even with the smoothness constraint Eq. (3.6) is degenerate. For instance, the following:

$$\varphi = k \begin{pmatrix} \Pi \\ 1 \end{pmatrix} \quad \text{with} \quad k \in \mathbb{R}$$
$$\boldsymbol{v}_i = \begin{pmatrix} v_{i,x} & v_{i,y} & v_{i,z} & 0 \end{pmatrix}^\top \quad i \in \{1, \ldots, n\}, \tag{3.8}$$

becomes a trivial solution of Eq. (3.6) by just taking $k \to 0$ in Eq. (3.8). To fix degeneracy, anchor points are added, which finally leads to the cost that represents our calibration problem:

Depth calibration from multiple planes:

$$\min_{\varphi, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_n} \mathrm{TV}_j^k[\varphi] \quad \text{such that} \begin{cases} \boldsymbol{v}_i^\top \cdot \overline{(\varphi \circ \gamma_i)} = 0 & i \in \{1, \cdots, n\} \\ \Pi \circ \varphi \circ \gamma_i - \Pi \circ \gamma_i = 0 & i \in \{1, \cdots, n\} \\ \boldsymbol{q}_j - \varphi(\tilde{\boldsymbol{q}}_j) = 0 & j \in \{1, \ldots, m\}, \end{cases} \tag{3.9}$$

where $\boldsymbol{q}_j$, $i \in 1, \ldots, m$ are ground truth values and $\tilde{\boldsymbol{q}}_j = \widetilde{\mathcal{D}}(\boldsymbol{p}_j) \cdot \frac{\bar{\boldsymbol{p}}_j}{\|\bar{\boldsymbol{p}}_j\|_2^2}$ with $\boldsymbol{p}_j = \Pi(\boldsymbol{q}_j)$ represent the estimation obtained with our correction.

Equation (3.9) is a non-convex problem that requires iterative constrained optimisation. Local optimisation can be also cumbersome due to the growing number of unknowns with the number of planes. Belhedi et al. [35] propose a solution based on iterative alternation, where the planes and φ are obtained in turns. This method is guaranteed to converge and the possible minimum found is not guaranteed to be global impact on accuracy as verified in practice.

3.5 CaDDiP Method

We show that problem (3.9) can be reformulated and it has a convex solution that does not need to explicitly introduce the planes as latent variables. We derive a planarity constraint that involves second derivatives. Before developing the new cost function we give definitions.

Definition 1. ***Affine parametrisation space.*** *We define $\Omega \in \mathbb{R}^2$ as a parametrisation space and an embedding $\gamma : \Omega \to \mathbb{R}^3$. We say that γ has an affine parametrisation space and that Ω is such a space for γ if there exists an affine projection Π_a such that:*

$$\Pi_a(\gamma(\boldsymbol{p})) = \boldsymbol{p} \qquad \forall \boldsymbol{p} \in \Omega, \tag{3.10}$$

with:

$$\Pi_a(\gamma(\boldsymbol{p})) = \mathbf{P}_a \, \overline{\gamma(\boldsymbol{p})} \qquad \text{with } \mathbf{P}_a \in \mathbb{R}^{2\times 4} \tag{3.11}$$

Definition 2. ***Projective parametrisation space.*** *We define Ω a projective parametrisation space of γ if there exists a perspective projection Π_p where:*

$$\Pi_p(\gamma(\boldsymbol{p})) = \boldsymbol{p} \qquad \forall \boldsymbol{p} \in \Omega, \tag{3.12}$$

with:

$$\Pi_p(\gamma(\boldsymbol{p})) \propto \mathbf{P}_p \; \overline{\gamma(\boldsymbol{p})} \qquad \text{with } \mathbf{P}_p \in \mathbb{R}^{3\times 4}. \tag{3.13}$$

As a consequence all affine parametrisation spaces are also projective. For example, given $\boldsymbol{p} \in \Omega$ and the embedding $\gamma(\boldsymbol{p}) = \rho(\boldsymbol{p}) \cdot \bar{\boldsymbol{p}}$ with $\rho : \Omega \to \mathbb{R}$, then Ω is a projective parametrisation space. It is easy to check that Eq. (3.12) holds for γ under canonical perspective projection.

We now present two theorems that allow one to define the differential condition for planarity.

Theorem 1. *For an embedding $\gamma : \Omega \to \mathbb{R}^3$, with Ω an affine parametrisation space, $\gamma(\Omega)$ is a plane if and only if $D^2[\gamma] = \mathbf{0}$*

Proof. Given that Ω is an affine parametrisation space, by definition we can find an affine projection matrix $\mathbf{P}_a = \begin{pmatrix}\mathbf{A} & \mathbf{b}\end{pmatrix} \in \mathbb{R}^{2\times 4}$ such that:

$$\mathbf{A} \cdot \gamma(\boldsymbol{p}) + \mathbf{b} = \boldsymbol{p} \qquad \forall \boldsymbol{p} \in \Omega. \tag{3.14}$$

Now, if γ is a plane there exist a vector $\mathbf{v} \in \mathbb{R}^3$ and an scalar $d \in \mathbb{R}$ such that $\mathbf{v}^\top \gamma = d$. We then have the following linear system:

$$\begin{pmatrix}\mathbf{A} \\ \mathbf{v}^\top\end{pmatrix} \gamma(\boldsymbol{p}) = \begin{pmatrix}\boldsymbol{p} - \mathbf{b} \\ d\end{pmatrix} \qquad \forall \boldsymbol{p} \in \Omega, \tag{3.15}$$

where $\mathbf{v}^\top$ is linearly independent of the two rows of $\mathbf{A}$, since otherwise this would contradict Eq. (3.14). We then have that:

$$\gamma(\boldsymbol{p}) = \begin{pmatrix}\mathbf{A} \\ \mathbf{v}^\top\end{pmatrix}^{-1} \begin{pmatrix}\boldsymbol{p} - \mathbf{b} \\ d\end{pmatrix} \qquad \forall \boldsymbol{p} \in \Omega, \tag{3.16}$$

which means that γ is a linear function and thus $D^2[\gamma] = \mathbf{0}$.

To prove the sufficient condition we first assume that γ is a solution of $D^2[\gamma] = \mathbf{0}$, which is a second order homogeneous linear *Partial Differential Equation (PDE)* that only admits linear functions as solutions (see [64] for comprehensive survey). We can then represent γ as

$$\gamma(\boldsymbol{p}) = \mathbf{M}\boldsymbol{p} + \mathbf{t} \qquad \text{with} \qquad \mathbf{M} \in \mathbb{R}^{3\times 2} \quad \text{and} \quad \mathbf{t} \in \mathbb{R}^{3\times 1} \qquad \forall \boldsymbol{p} \in \Omega. \tag{3.17}$$

As a consequence of Eq. (3.17), γ is either a plane, a point or a line. By including the fact that Ω is an affine parametrisation space we have:

$$\mathbf{A}\ \gamma(\boldsymbol{p}) = \mathbf{A}\mathbf{M}\boldsymbol{p} + \mathbf{A}\mathbf{t} = \boldsymbol{p} \qquad \forall \boldsymbol{p} \in \Omega. \tag{3.18}$$

As $\mathbf{A}$ has full rank, from Eq. (3.18) we have that $\mathbf{A}\mathbf{M} = \mathbf{I}_{2\times 2}$, which makes $\mathbf{M}$ full rank and Eq. (3.17) a plane equation. □

Theorem 2. *For an embedding $\gamma : \Omega \to \mathbb{R}^3$, with Ω a projective parametrisation space $D^2[\gamma] = 0$ is a sufficient but not necessary condition for $\gamma(\Omega)$ to be a plane.*

Proof. For the sufficient condition we have that $D^2[\gamma] = \mathbf{0}$ has linear functions as solutions and similarly to the affine case, if Ω is a perspective parametrisation space γ in fact represents a plane. To prove that it is not a necessary condition we give a counter example. Let $\Omega \in \mathbb{R}^2$ be a projective parametrisation space of the embedding:

$$\gamma(\boldsymbol{p}) = \rho(\boldsymbol{p}) \cdot \bar{\boldsymbol{p}} \qquad \rho(\boldsymbol{p}) = \frac{-d}{\mathbf{v}^\top \bar{\boldsymbol{p}}} \qquad \mathbf{v} \in \mathbb{R}^3,\ \ d \in \mathbb{R}, \tag{3.19}$$

we can verify that Ω is a perspective parametrisation space by verifying that $\Pi(\gamma(\boldsymbol{p})) = \boldsymbol{p}$ for any $\boldsymbol{p} \in \Omega$. Clearly, due to denominators we have that $D^2[\gamma] \neq \mathbf{0}$ in general. We can also verify that γ is a plane:

$$\mathbf{v}^\top \gamma(\boldsymbol{p}) + d = -d \cdot \frac{\mathbf{v}^\top \boldsymbol{p}}{\mathbf{v}^\top \boldsymbol{p}} + d = 0, \tag{3.20}$$

which means that γ lies in the plane with parameters $\begin{pmatrix}\mathbf{v}^\top & d\end{pmatrix}$. □

3.5.1 A Linear Least-Squares Solution to Calibration

We use the definitions above to transform problem (3.9) into one where plane equations are substituted by the planarity constraint $D^2[\gamma] = \mathbf{0}$. From theorems 1 and 2, the planarity condition only applies to affine parametrisation spaces. All surfaces represented by the function γ_i with $i \in \{1, \ldots, n\}$ are defined from the retina which is indeed a projective parametrisation space. We first define $\Omega \in \mathbb{R}^2$ as a subset of the plane $z = 0$. We denote as $\widetilde{\gamma}_i : \Omega \to \mathbb{R}^3$ the re-parametrisation of γ_i in Ω. We express $\widetilde{\gamma}_i$ as follows:

$$\widetilde{\gamma}_i = \begin{pmatrix}\boldsymbol{m} & \widetilde{\gamma}_{i,z}(\boldsymbol{m})\end{pmatrix}^\top \qquad \boldsymbol{m} = \begin{pmatrix}x & y\end{pmatrix}^\top \qquad i \in \{1, \ldots, n\}. \tag{3.21}$$

From Eq. (3.21) we have that:

$$\begin{pmatrix}\mathbf{I}_{2\times2} & \mathbf{0}_{2\times2}\end{pmatrix} \widetilde{\gamma}_i(\boldsymbol{m}) = \boldsymbol{m} \qquad \forall \boldsymbol{m} \in \widetilde{\Omega} \qquad i \in \{1, \ldots, n\}, \tag{3.22}$$

which means that $\widetilde{\Omega}$ is an affine parametrisation space of $\widetilde{\gamma}_i$ with $i \in \{1, \ldots, n\}$.

We redefine the problem shown in Eq. (3.9) using Theorem 1:

$$\min_{\varphi} \mathrm{TV}^2_F[\varphi] \quad \text{such that} \quad \begin{cases} D^2[\varphi \circ \widetilde{\gamma}_i] = 0 & \\ \Pi \circ \varphi \circ \widetilde{\gamma}_i - \Pi \circ \widetilde{\gamma}_i = 0 & i \in \{1, \cdots, n\} \\ \boldsymbol{q}_j - \varphi(\widetilde{\boldsymbol{q}}_j) = 0 & j \in \{1, \ldots, m\}. \end{cases} \tag{3.23}$$

Equation (3.23) is a convex constrained optimisation problem in φ. In practice we convert (3.23) into an unconstrained cost using a Lagrangian relaxation of four functionals:

$$\min_{\varphi} \varepsilon_s[\varphi] + \lambda_d\ \varepsilon_d[\varphi] + \lambda_r\ \varepsilon_r[\varphi] + \lambda_a\ \varepsilon_a[\varphi], \tag{3.24}$$

that impose:

- Smoothness:

$$\varepsilon_s[\varphi] = \mathrm{TV}^2_F[\varphi], \tag{3.25}$$

- Planarity:

$$\varepsilon_d[\varphi] = \sum_{i=1}^{n} \int_{\Omega} \|D^2[\varphi \circ \tilde{\gamma}_i]\|^2_F \, \mathrm{d}\Omega = \sum_{i=1}^{n} \mathrm{TV}^2_F[\varphi \circ \tilde{\gamma}_i], \tag{3.26}$$

- Re-projection:

$$\varepsilon_r[\varphi] = \sum_{i=1}^{n} \int_{\Omega} \|\tilde{\gamma}_i \times (\varphi \circ \tilde{\gamma}_i)\|^2_2 \, \mathrm{d}\Omega, \tag{3.27}$$

- Scale:

$$\varepsilon_a[\varphi] = \sum_{j=1}^{m} \int_{\Omega} \|(\varphi \circ \tilde{\gamma}_i) - \boldsymbol{q}_j\|^2_2 \, \mathrm{d}\Omega. \tag{3.28}$$

If φ is represented with a LBE model the cost (3.24) becomes a linear least-squares problem that can be solved very efficiently. The value of the hyper-parameters λ_d, λ_r and λ_a is chosen empirically.

3.6 CaDDiP: Algorithm and Implementation Details

3.6.1 Function Model

In this thesis we represent the correction function using a Linear Basis Expansion (LBE) parametric representation. These representations are linear in function of the parameters, which allow one to solve many data fitting problems in closed form. Polynomial representations are LBE where the parameters are the polynomial coefficients. Piece-wise polynomials such as the B-Splines or Radial Basis Functions, such as *Thin Plate Spline (TPS)* [65] are very popular. We describe the proposed approach using a general LBE to represent both φ and γ_i.

The parameters of the linear expansion are defined in matrix $\mathbf{L}_{\gamma_i} \in \mathbb{R}^{3\times l}$. The function is then defined using $\mathbf{L}_{\gamma_i}$ and a non-linear lifting function $\boldsymbol{\nu} : \mathbb{R}^2 \to \mathbb{R}^l$:

$$\gamma_i(\boldsymbol{m}_j, \mathbf{L}_{\gamma_i}) = \mathbf{L}^T_{\gamma_i} \boldsymbol{\nu}(\boldsymbol{m}_j) \qquad i \in \{1, \dots, n\} \qquad \text{with} \quad \boldsymbol{m}_j \in \Omega. \tag{3.29}$$

3.6.2 Surface Fitting

The first step in our algorithm is to estimate γ_i for every plane in the scene using a LBE representation. We find the parameters $\mathbf{L}_{\gamma_i}$ as the minimum of the following cost:

$$\min_{\mathbf{L}_{\gamma_i}} \varepsilon_m(\mathbf{L}_{\gamma_i}) + \lambda_{s_1} \varepsilon_s(\mathbf{L}_{\gamma_i}), \tag{3.30}$$

with ε_m penalising the approximation error:

$$\varepsilon_m(\mathbf{L}_{\gamma_i}) = \sum_{\boldsymbol{p}_j \in \Omega} \left\| \tilde{\mathcal{D}}(\boldsymbol{p}_j) \frac{\bar{\boldsymbol{p}}_j}{\|\bar{\boldsymbol{p}}_j\|^2_2} - \gamma_i(\boldsymbol{p}_j, \mathbf{L}_{\gamma_i}) \right\|^2_2 \tag{3.31}$$

and ε_s the bending energy smoother. We can rewrite both terms in Eq. (3.30) as:

$$\begin{aligned} \varepsilon_m(\mathbf{L}_{\gamma_i}) &= \|\mathbf{D}\mathbf{L}_{\gamma_i} - \mathbf{B}\|_F^2 \\ \varepsilon_s(\mathbf{L}_{\gamma_i}) &= \|\mathbf{Z}\mathbf{L}_{\gamma_i}\|_F^2 \end{aligned} \tag{3.32}$$

The compound cost function thus writes as:

$$\min_{\mathbf{L}_{\gamma_i}} \left(\|\mathbf{D}\mathbf{L}_{\gamma_i} - \mathbf{B}\|_F^2 + \lambda_{s_1}^2 \|\mathbf{Z}\mathbf{L}_{\gamma_i}\|_F^2 \right) \tag{3.33}$$

and its *Linear Least-Squares (LLS)* solution is:

$$\mathbf{L}_{\gamma_i}(\mu) = (\mathbf{D}^T\mathbf{D} + \lambda_{s_1}^2 \mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{D}^T\mathbf{B} \tag{3.34}$$

3.6.3 Correction Function Fitting

We demonstrate that, by expressing the global correction function φ using a LBE representation, we can minimise the cost function of Eq. (3.24) with LLS:

$$\begin{aligned} &\min_{\mathbf{L}_\varphi} \left(\varepsilon_d^2 + \lambda_{s_2}\ \varepsilon_s^2 + \lambda_r\ \varepsilon_r^2 + \lambda_a\ \varepsilon_a^2 \right) \\ &\min_{\mathbf{L}_\varphi} \left(\|\mathbf{M}\mathbf{L}_\varphi\|_F^2 + \lambda_{s_2}\|\mathbf{Z}\mathbf{L}_\varphi\|_F^2 + \lambda_r\|\mathbf{C}\mathbf{L}_\varphi\|_F^2 + \lambda_a\|\mathbf{G}\mathbf{L}_\varphi - \mathbf{B}\|_F^2 \right) \end{aligned} \tag{3.35}$$

Finally, the LLS solution for $\mathbf{L}_\varphi$ is:

$$\mathbf{L}_\varphi(\mu) = (\mathbf{M}^T\mathbf{M}\ +\ \lambda_{s_2}^2\mathbf{Z}^T\mathbf{Z}\ +\ \lambda_r^2\mathbf{C}^T\mathbf{C}\ +\ \lambda_a^2\mathbf{G}^T\mathbf{G})^{-1}\left(\mathbf{G}^T\mathbf{B}\right) \tag{3.36}$$

Thus, we convert the calibration of the systematic depth distortion into a convex optimisation problem with guarantee of convergence, providing a LLS solution that can be implemented very efficiently. In practice, we use the TPS parametrisation (see [66]) as a model of LBE. In Algorithm 1 we summarise the implementation details with pseudo-code.

3.7 Results

We validate our algorithm with simulated data and real data acquired from three different commercial depth cameras: *i*) MESA-Imaging SR4k (ToF), *ii*) PMD CamCube 3.0 (ToF) and *iii*) Kinect v1 (SLiP). The data-sets consist of a fixed number of plane views covering the desired depth range. We compared our results with [35] using the implementation provided by the author.

Algorithm 1 CaDDiP. Implementation details.

```
 Computation of the TPS LBE for every plane view:
Require: $n$ plane views, hyper-parameter $(\lambda_{s_1})$
 for every plane view: $i = 1 \cdots n$ do
     Generate the 2D TPS centres
     Compute the TPS LBE $\gamma_i$ using LLS (Eq. 3.34)
 end for
 Solving the Convex Unconstrained optimisation problem using LLS:
Require: $n$ plane views, $m$ ground truth correspondences, hyper-parameters $(\lambda_{s_2},\ \lambda_r,\ \lambda_a)$
 Generate the 3D TPS centres
 Calculate each functional cost: {planarity, smoothness, re-projection, scale}
 Compute $\varphi$ using LLS (Eq. 3.36)
     return $\varphi$
```

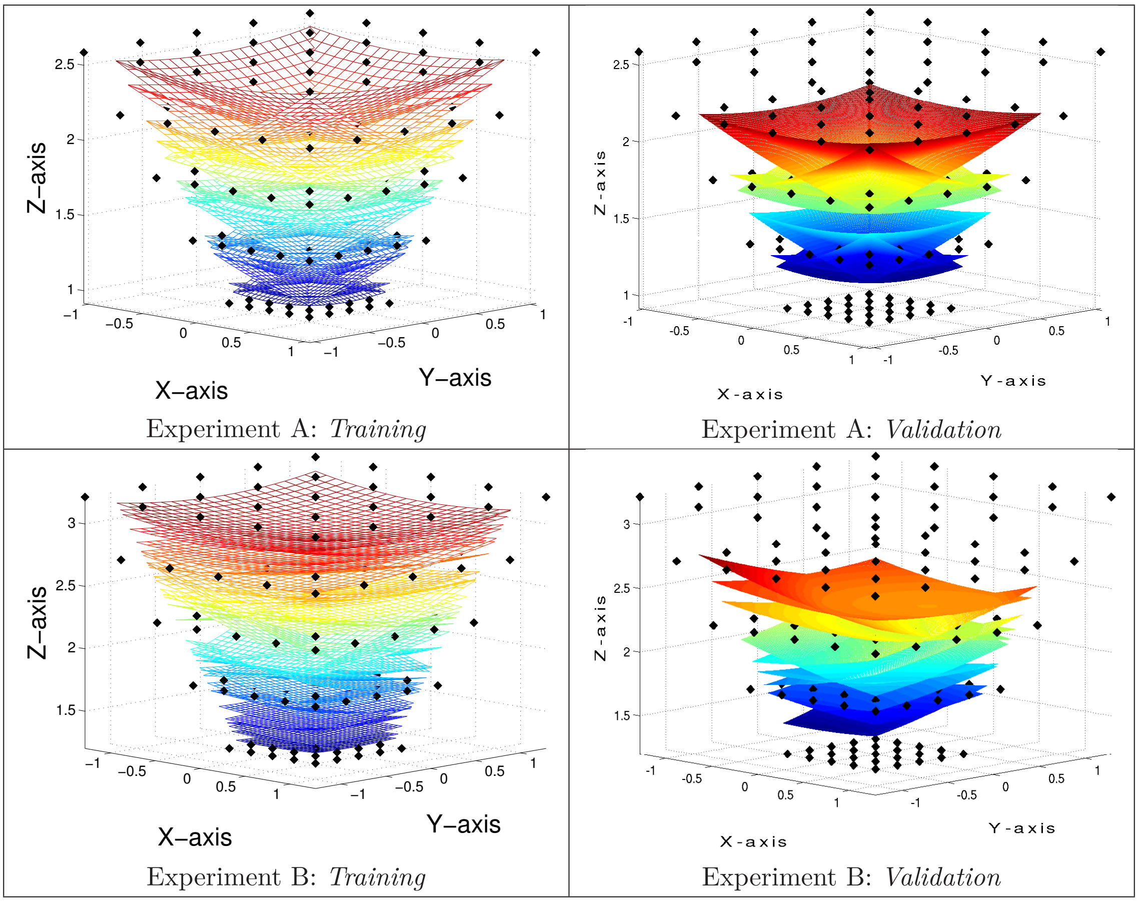
3.7.1 Experimental Setup

For the evaluation of our method we propose a pipeline consisting of two separate stages: *i*) training and *ii*) validation. In the training stage we show the camera a fixed number of plane views covering the whole operating depth range. This can be done easily using a flat surface with no need of special patterns imprinted on it. Given the distortion smoothness assumption we can omit a portion of the input samples in the training stage, highly reducing the computational cost. Next, we check the estimated correction function using a validation set consisting of several plane views into the depth range.

In Fig. 3.3 we show an example where the mesh of a synthetically generated scene in both stages is shown.

As depicted from Eq. (3.24), we need to estimate a set of hyper-parameters. In practice, most of them can be set to a certain fixed value with a minimum of expert knowledge. In fact, only those corresponding to the re-projection and the scale constraints need to be roughly adjusted (λ_r and λ_a). In this work we choose them empirically: we perform a coarse sweep of both terms over a fixed interval looking for the values that maximise performance. The estimation of this set of hyper-parameters must be obtained in a first step of the proposed calibration method for an specific depth camera.

In our experiments we verified that the remaining parameters, related to: *i*) the TPS smoothers (λ_{s_1} and λ_{s_2}), *ii*) the number l of control points (TPS centres) and *iii*) the number m of reference points for the scale constraint, can be fixed for all the experiments such that it doesn't affect significantly to the overall performance of the algorithm. In practice, we fix them to *i*) $\{\lambda_{s_1}, \lambda_{s_2}\} = \{1e^{-6}, 1e^{-3}\}$ in both simulated and real experiments, respectively, *ii*) $l = 5$ centres for each input dimension and *iii*) $m = 8$ reference points distributed among the centres of the planes.



Experiment A: *Training*

Experiment A: *Validation*

Experiment B: *Training*

Experiment B: *Validation*

Figure 3.3: Meshing of the input data-set in the training and validation stages. Given the smoothness assumption, training data-set is sub-sampled in a factor of 10. Black points enclose the depth camera working space.

Let us define the error measurement as a function δ that works over two different sets: *i*) an input data-set and *ii*) the corresponding points in the ground truth. For a particular scenario, given an input set $\mathcal{Q}$ composed of n plane views captured with a depth sensor of resolution $\mathrm{res}_i, \quad i = 1, \cdots, n \quad \text{s.t.} \quad \mathcal{Q} = \{\mathbf{q}_{(1,1)}, \cdots, \mathbf{q}_{(n,\mathrm{res}_n)}\}$, and given a set of ground truth planes $\mathcal{G} = \{\Gamma_1, \cdots, \Gamma_n\}$, we define the error function as:

$$\delta(\mathcal{Q}, \mathcal{G}) = \sqrt{\frac{1}{n \cdot \sum_{i=1}^{n} \mathrm{res}_i} \sum_{i=1}^{n} \sum_{j=1}^{\mathrm{res}_i} |d\left(\mathbf{q}_{ij}, \Gamma_i\right)|^2} \tag{3.37}$$

where $d(\mathbf{x}, \mathbf{y})$ computes the Euclidean distance between an input point $\mathbf{x}$ and its corresponding ground truth value $\mathbf{y}$ in the direction of the plane normal.

Depending on the point to plane distance cases, we define the four error measurements considered in our experiments:

i) *ground truth error before correction*: the distance between the measured set of points distorted by systematic depth distortion $\left(\widetilde{\mathcal{Q}}\right)$ and its corresponding ground truth value $\left(\mathcal{G}^{\mathrm{gth}}\right)$ is considered.

$$\delta_{\mathrm{gth}}^{\mathrm{init}} = \delta\left(\widetilde{\mathcal{Q}}, \mathcal{G}^{\mathrm{gth}}\right) \tag{3.38}$$

ii) *ground truth error after correction*: the distance between the set of points obtained after applying the estimated correction function ($\mathcal{Q}^*$) and its corresponding ground truth value is considered.

$$\delta_{\mathrm{gth}}^{\mathrm{end}} = \delta\left(\mathcal{Q}^*, \mathcal{G}^{\mathrm{gth}}\right) \tag{3.39}$$

iii) *Co-planarity error before correction*: the distance between the measured set of points distorted by systematic depth distortion $\left(\widetilde{\mathcal{Q}}\right)$ and its corresponding point value in the best-fitted plane $\left(\mathcal{G}^{\mathrm{bf}}\right)$ is considered.

$$\delta_{\mathrm{bfp}}^{\mathrm{init}} = \delta\left(\widetilde{\mathcal{Q}}, \mathcal{G}^{\mathrm{bf}}\right) \tag{3.40}$$

iv) *Co-planarity error after correction*: the distance between the set of points obtained after applying a correction function and the best-fitted plane to the points value $\left(\mathcal{G}^{\mathrm{bf}}\right)$ is considered.

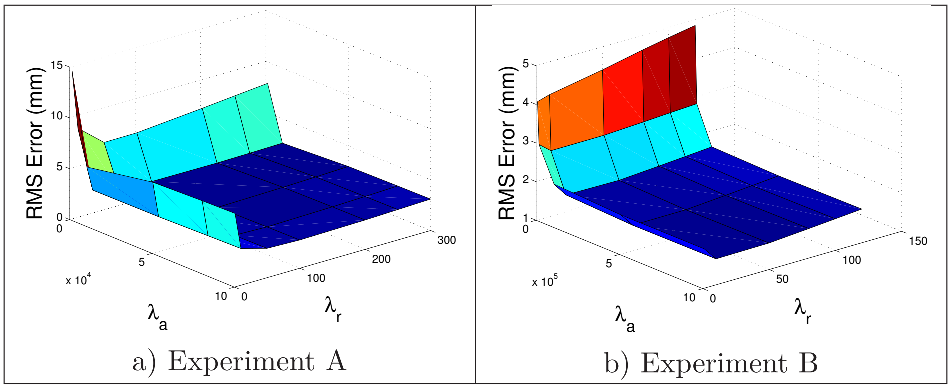
$$\delta_{\mathrm{bfp}}^{\mathrm{end}} = \delta\left(\mathcal{Q}^*, \mathcal{G}^{\mathrm{bf}}\right) \tag{3.41}$$

3.7.2 Results with Synthetic Data

We evaluate our proposal over two different simulated experiments: *i)* A (provided by the authors of [35]) and *ii)* B (proposed implementation), each one consisting of a fixed number of plane views with randomly generated orientations covering the desired calibrated depth range and comprising two different distortion functions for the generation of the measurements. We build on the experiments in [35] where the authors apply a distortion function to every plane view that varies accordingly to the distance and increases from the image centre to the boundaries.

The training set consists of 36 plane views distributed into the predefined depth range, while the validation stage uses 10 arbitrary plane views. In this section we use $\delta_{\mathrm{gth}}^{\mathrm{init}}$ and $\delta_{\mathrm{gth}}^{\mathrm{end}}$ as error measurements.

In Fig. 3.4 we show the *Root Mean Square (RMS)* error for each combination of $\{\lambda_r, \lambda_a\}$ in both synthetically generated experiments. Table 3.1 shows the chosen hyper-parameters and the experiment setup.



a) Experiment A

b) Experiment B

Figure 3.4: Sweep on the hyper-parameters (λ_r, λ_a) for both simulated experiments.

Experiment	Resolution	Range (m)	$\{\lambda_r, \lambda_a\}$
A	204 × 204	$\{0.75$ - $2.8\}$	$\{10,\ 5e^5\}$
B	204 × 204	$\{1.0$ - $3.5\}$	$\{200,\ 1e^4\}$

Table 3.1: Estimated hyper-parameters for both simulated experiments.

In Table 3.2 we show the results comparing with one method from the state of the art under ideal situations: *no noise is added.* The obtained results are as good as [35], which also reports better results compared with [30] and [31].

Experiment	$\delta_{\text{gth}}^{\text{init}}$	[35] : $\delta_{\text{gth}}^{\text{end}}$	CaDDiP : $\delta_{\text{gth}}^{\text{end}}$
A	29.17	2.748	**2.213**
B	18.2	1.152	**1.279**

Table 3.2: Error measurements, when no noise is applied, compared with one method from the state of the art. Bold numbers represent the results after applying the proposed method.

In Fig. 3.5 we show a cross-section of the input data-set for both training and validation stages. Red lines show the synthetically generated depth measurements of several flat surfaces, while the blue lines represent the corresponding ground truth. Green lines are the estimated smooth surfaces using the TPS parametrisation γ_i and cyan lines are the recovered depth after applying the proposed correction function φ.

Depth cameras are noisy sensors. In [35], the authors evaluate the sensor noise (ToF cameras as a particular case) taking 100 measurements of a wall for every pixel, reporting that the standard deviation varies from 5 mm to 18 mm. We validate our results adding Gaussian noise to the synthetically generated depth measurements. In Fig. 3.6 we show the RMS error $\left(\delta_{\text{gth}}^{\text{end}}\right)$ and the confidence interval compared with [35]. We evaluate the results for several additive noise levels after iterating 5 times each experiment.

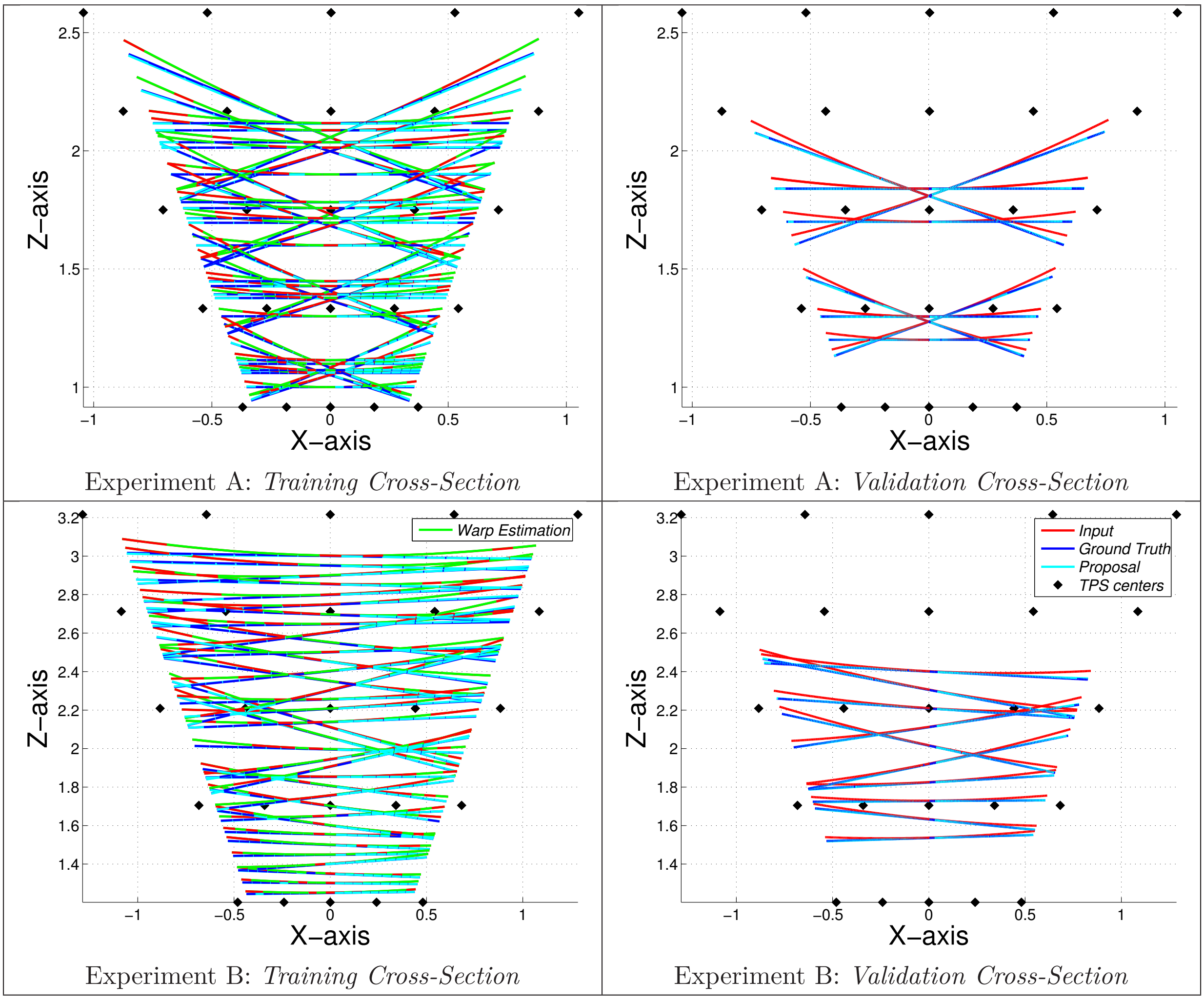
As depicted from Fig. 3.6 our method gets better results in terms of accuracy and dispersion under high noise levels compared with the state of the art, getting similar accuracy when no additive noise is incorporated.

In Figs. 3.7 and 3.8 we show the RMS error colour-map for two arbitrary flat surfaces in the validation stage. We show the results in both experiments A and B under different levels of additive noise.

3.7.3 Results with Real Data

We validate our method using two different ToF cameras: *i*) SR4k from Mesa-Imaging [67] (commercially available device) and *ii*) PMD CamCube 3.0 [68] (research reference design but currently discontinued) and one SLiP camera model: *iii*) Kinect v1. In these experiments we also use 36 plane views into the calibrated depth range for the training set and 10 plane views in the validation stage.

Unlike simulated experiments, we evaluate the accuracy of the proposed method in line with the real experiments in [35] by using a measurement of co-planarity for every point in the scene.



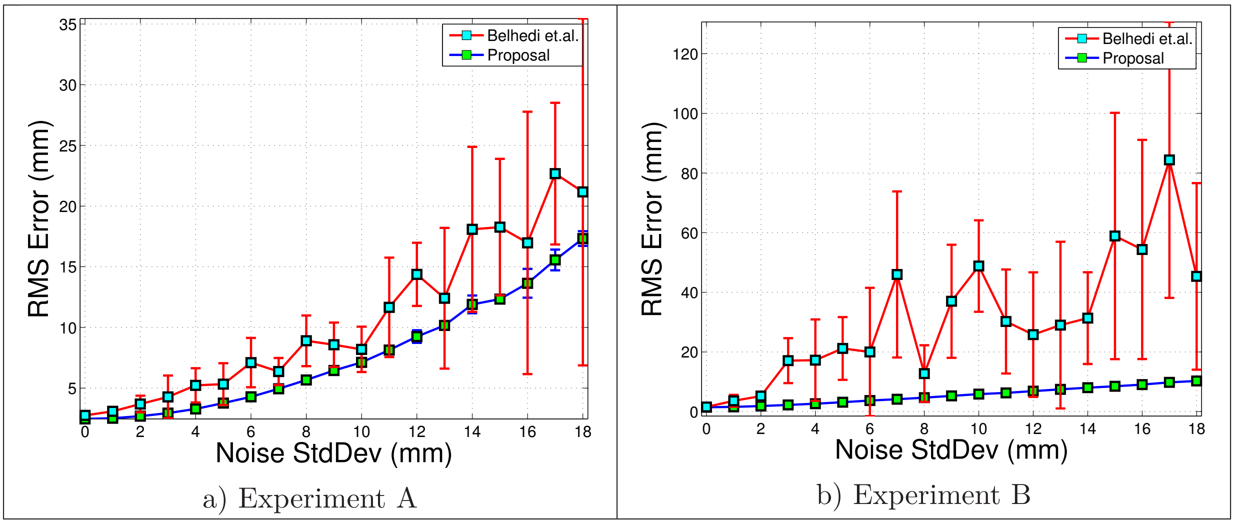
Experiment A: *Training Cross-Section*

Experiment A: *Validation Cross-Section*

Experiment B: *Training Cross-Section*

Experiment B: *Validation Cross-Section*

Figure 3.5: Data cross-section in the training and validation stages. We show in colours: *i*) the input measurement (red lines), *ii*) the ground truth values (blue lines), *iii*) the TPS LBE γ_i (green lines), *iv*) the proposed correction (cyan lines) and *v*) the TPS centres (black dots).



a) Experiment A

b) Experiment B

Figure 3.6: We show the mean and the confidence interval of the RMS error for different values of additive Gaussian noise.

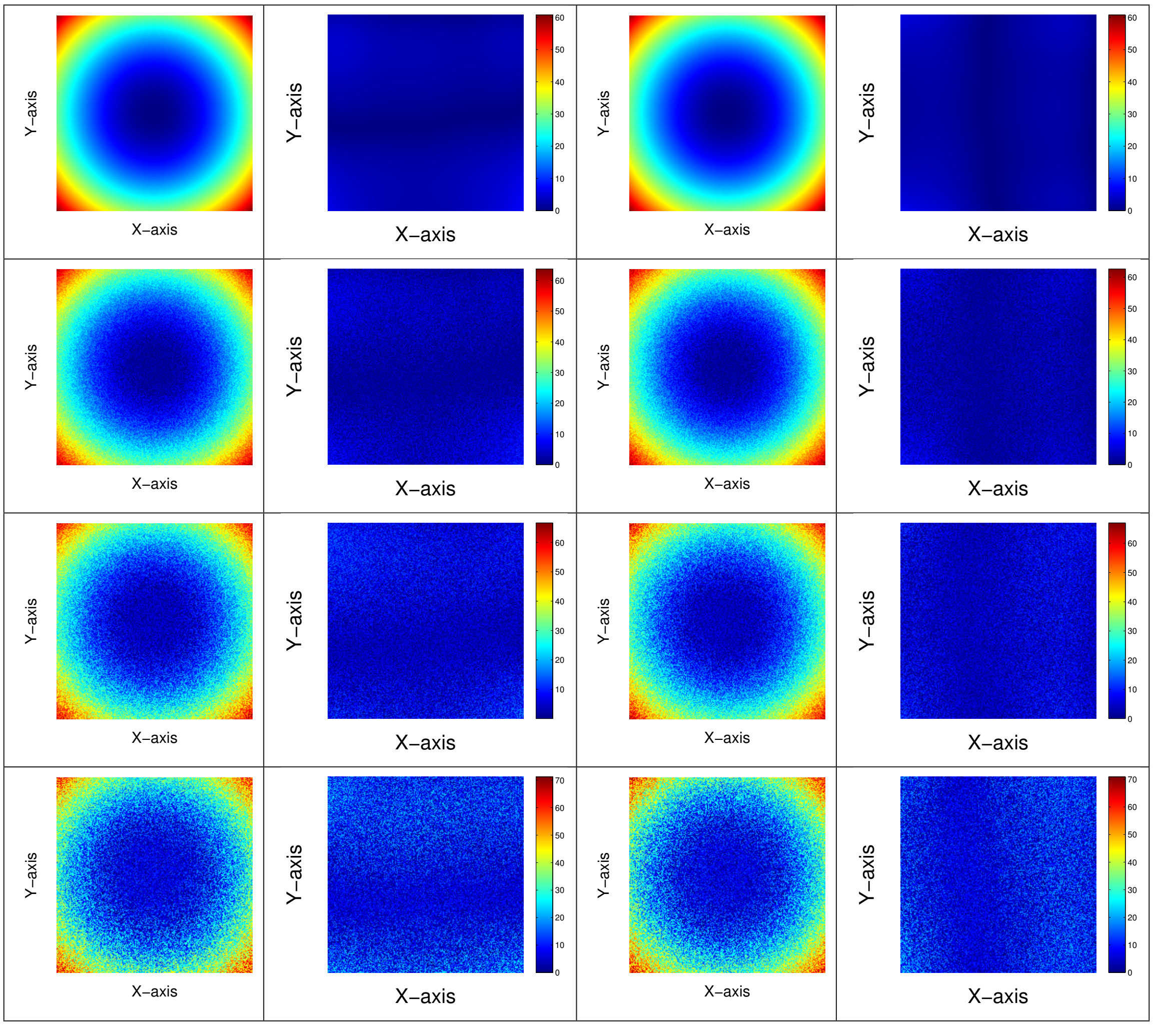


Figure 3.7: Experiment A. We show in columns a colour-map of the RMS error before $\left(\delta_{\text{gth}}^{\text{init}}\text{: odd columns}\right)$ and after $\left(\delta_{\text{gth}}^{\text{end}}\text{: even columns}\right)$ applying the proposed method for two arbitrary planes in the validation stage. In each row, we show the results for different noise levels; $\{0, 4, 8, 14\}$ mm.

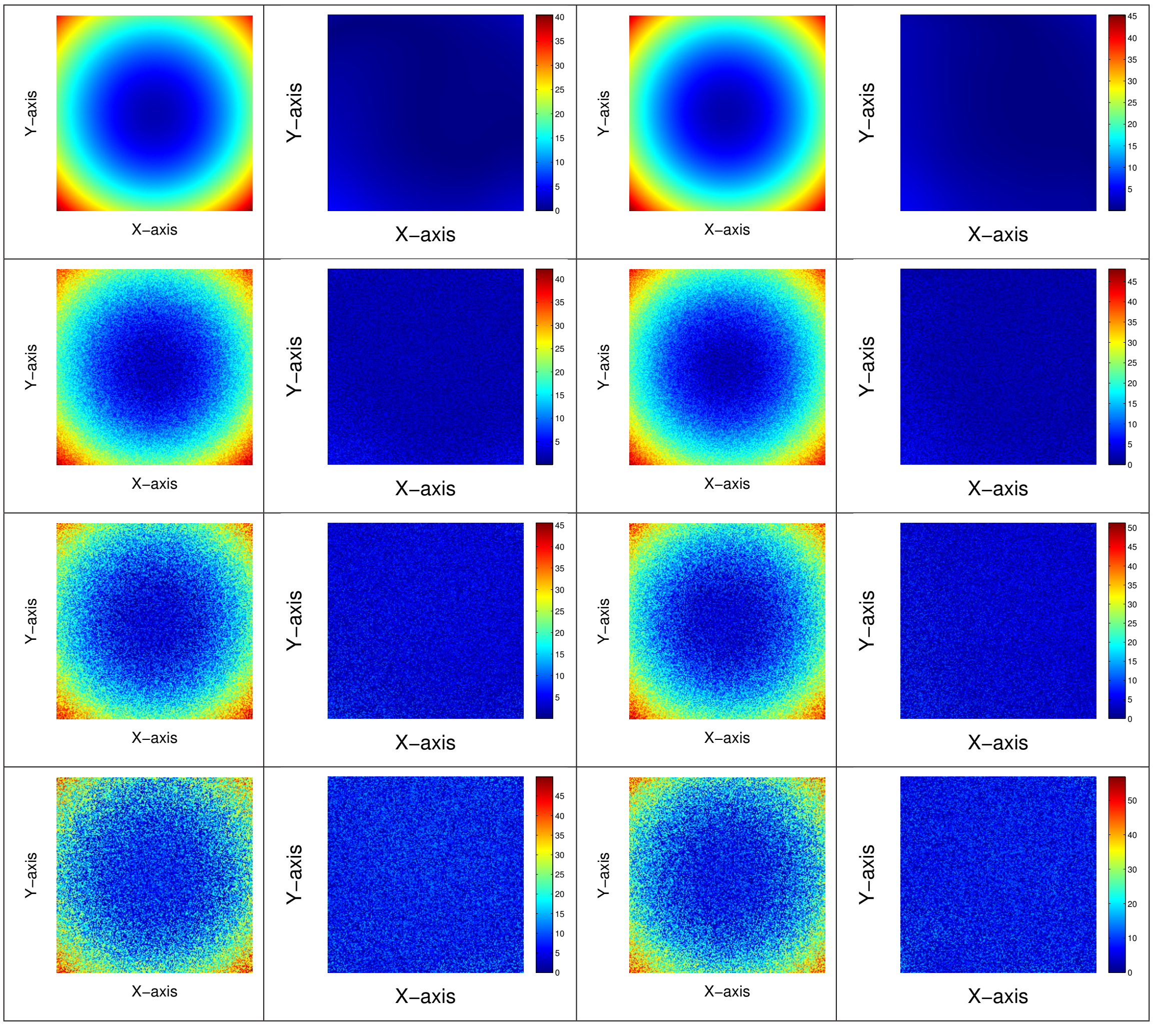


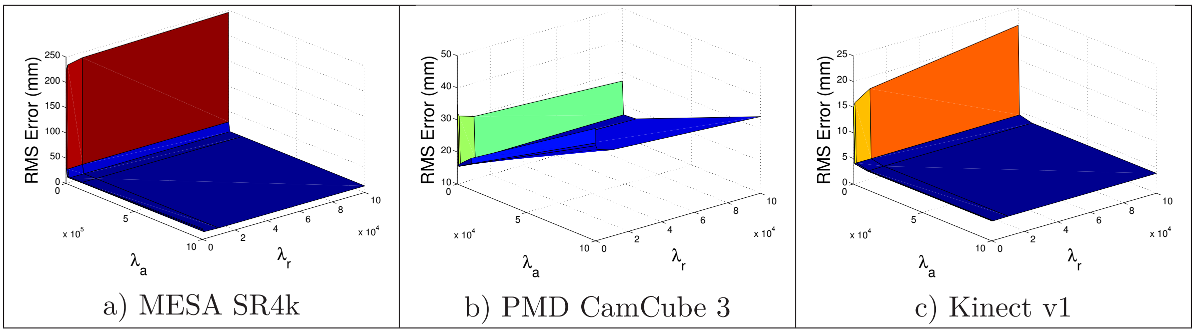
Figure 3.8: Experiment B. We show in columns a colour-map of the RMS error before $\left(\delta_{\text{gth}}^{\text{init}}\text{: odd columns}\right)$ and after $\left(\delta_{\text{gth}}^{\text{end}}\text{: even columns}\right)$ applying the proposed method for two arbitrary planes in the validation stage. In each row, we show the results for different noise levels; $\{0, 4, 8, 14\}$ mm.

We use as reference points in the training stage, points belonging to the best fitted plane to the input data-set, shifted towards the camera a distance corresponding to the maximum deviation error. Hence, we use in this section $\delta_{\text{bfp}}^{\text{init}}$ and $\delta_{\text{bfp}}^{\text{end}}$ as error measurements.

In Table 3.3 we show the setup and the estimated hyper-parameters (see Fig. 3.9) for each camera.

Technology	Camera Model	Resolution	Calibrated Range	$\{\lambda_r, \lambda_a\}$
ToF	MESA SR4k	176 × 144	{0.5 - 3.7} m	$\{1e4, 1e4\}$
ToF	PMD CamCube 3.0	204 × 204	{2.8 - 5.6} m	$\{10, 100\}$
SLiP	Kinect v1	640 × 480	{0.58 - 2.3} m	$\{1e3, 1e4\}$

Table 3.3: Depth cameras features and setup.



a) MESA SR4k b) PMD CamCube 3 c) Kinect v1

Figure 3.9: Sweep on the hyper-parameters (λ_r, λ_a).

In Table 3.4 we show the RMS error compared with the state of the art method showing slightly better results.

Technology	Camera Model	$\delta_{\text{bfp}}^{\text{init}}$	[35] : $\delta_{\text{bfp}}^{\text{end}}$	CaDDiP : $\delta_{\text{bfp}}^{\text{end}}$
ToF	MESA SR4k	27.2	12.209	**11.496**
ToF	PMD CamCube 3.0	34.6	8.705	**8.646**
SLiP	Kinect v1	12.2	4.046	**3.801**

Table 3.4: Comparison with one approach from the state of the art. Bold numbers represent the results after applying the proposed method.

In Figs. 3.10, 3.11 and 3.12 we show the RMS error colour-map for two arbitrary flat surfaces in the validation stage for: *i*) MESA SR4k, *ii*) PMD CamCube 3 and *iii*) Kinect v1.

From simulated experiments it could be expected that the proposed method would outperform the state of the art under real noisy scenarios, on the contrary, we present reduced improvements. This can be explained considering that we don't measure the accuracy using ground truth points but a measurement of co-planarity, avoiding a critical step in [35] procedure: the affinity transformation under noisy environments.

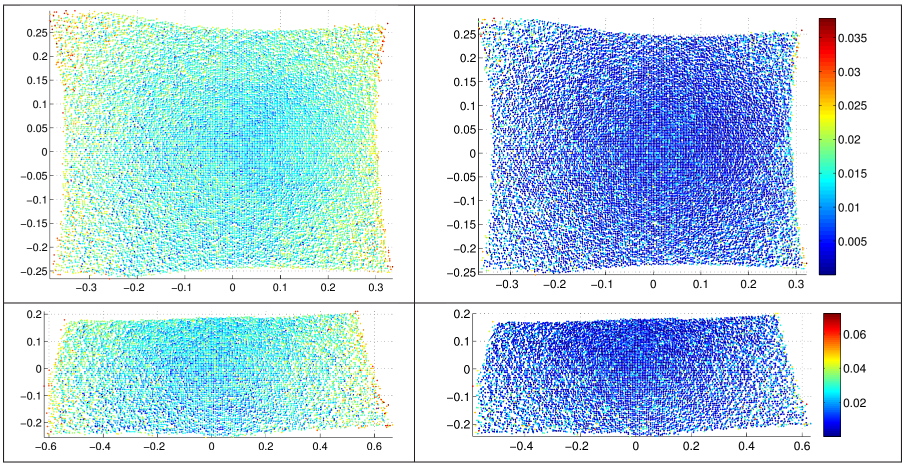


Figure 3.10: We show the RMS error of both the measurement (left) and the proposed correction (right) using the ToF camera model MESA SR4k.

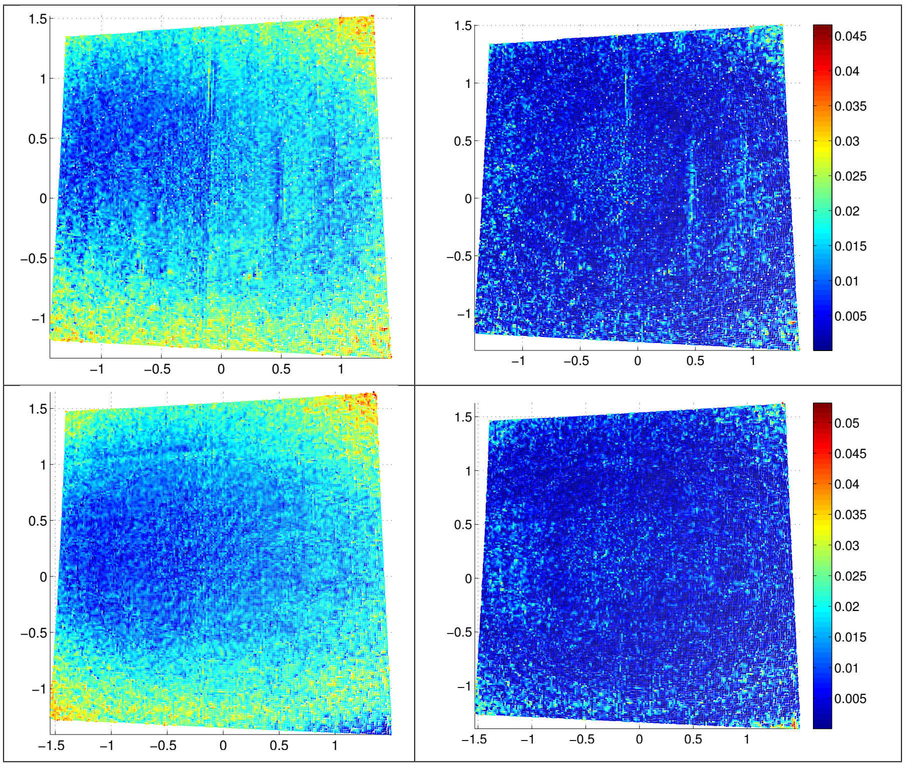


Figure 3.11: We show the RMS error of both the measurement (left) and the proposed correction (right) using the ToF camera model PMD CamCube 3.

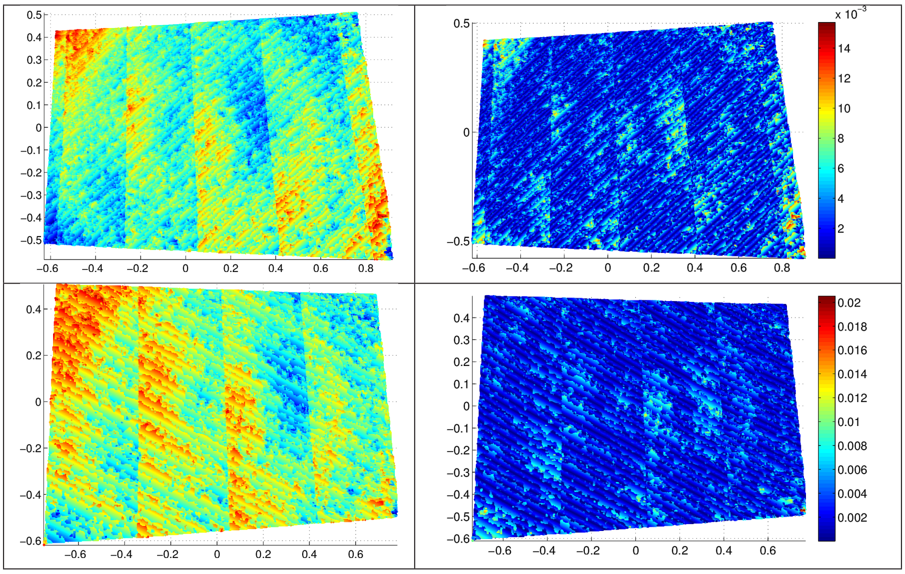


Figure 3.12: We show the RMS error of both the measurement (left) and the proposed correction (right) using the SLiP camera model Kinect v1.

3.8 Conclusions

This work presents a method to correct systematic depth distortion for any depth sensor, one of the most limiting systematic error in depth sensing. We propose a closed-form solution based on local planarity from multiple planes that transforms a non-convex method into an convex constrained optimisation. In practice, we convert it into an unconstrained cost using Lagrangian relaxation of four functional that can be minimised with linear least-squares.

Given the smoothness assumption of the depth distortion we can sub-sample the input dataset highly reducing the computational cost. The global correction function obtained at this point (training stage) is stored for a particular depth sensor in order to be applied afterwards.

We show that we can recover depth from a scene affected with systematic depth distortion with a critical reduction of ground truth reference points (8 points in all the experiments) and with no need of special patterns (only flat surfaces are required). This makes the proposal to be applied easily with no need for expensive additional systems to acquire accurate ground truth for all the pixels. The provided LLS solution makes the proposal to be implemented efficiently with predictable complexity.

We analyse the performance of the proposal under synthetically generated experiments, showing that the method clearly corrects systematic depth distortion under a high level of additive noise outperforming the state of art methods. We validate the proposal in real scenes captured using two depth sensing technologies: *i*) Time of Flight cameras and *ii*) SLiP camera, showing accurate results.

We show in our experiments that only two of the hyper-parameters needs to be roughly adjusted to get proper results. In practice, we empirically choose them.

Chapter 4

Modelling and Correction of Multipath Interference in Time of Flight Cameras

4.1 Introduction

This approach proposes a method that removes multipath distortion from the measurements of a *Time of Flight (ToF)* camera without any prior knowledge about the scene. We also do not require hardware modifications in the camera. Our method optimises a cost function whose global minimum represents the correction needed to obtain the real depth of each pixel, free from multipath distortion. We use a radiometric generative model based on certain properties about the radiometry of the camera and the scene. Unlike the proposal of Fuchs [41], our method is accurate given that the required conditions about the scene are approximately met. The results show that our approach is very accurate, even in complex scenes.

4.2 Overview

Multipath interference of light is the cause of important errors in Time of Flight (ToF) depth estimation. This work proposes an algorithm that removes multipath distortion from a single depth map obtained by a ToF camera. Our approach does not require information about the scene, apart from ToF measurements. The method is based on fitting ToF measurements with a radiometric model. Model inputs are depth values free from *Multipath Interference (MpI)* whereas model outputs consist of synthesised ToF measurements. We propose an iterative optimisation algorithm that obtains model parameters that best reproduce ToF measurements, recovering the depth of the scene without distortion (see Fig. 4.1 for a graphical overview). We show results with both synthetic and real scenes captured by commercial ToF sensors. In all cases, our algorithm accurately corrects the multipath distortion, obtaining depth maps that are very close to ground truth data.

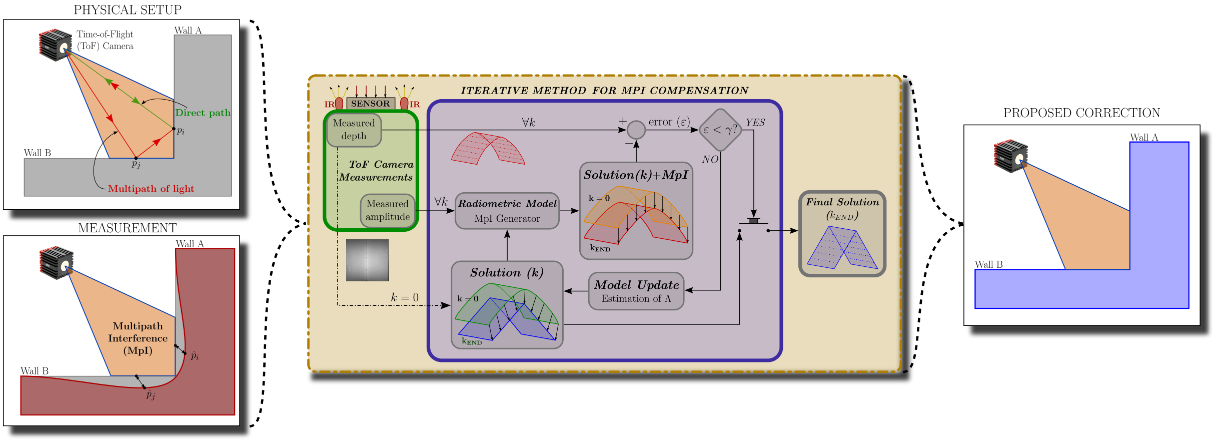


Figure 4.1: Graphical abstract of the proposed method. $\mathbf{\Lambda}$ is the set of fitting parameters to optimise, γ a non-restrictive threshold and k the iteration number.

The chapter is structured as follows. Notation is given in Section 4.2.1. Section 4.3 details the radiometric model used to predict ToF measurements of a known scene. The iterative correction algorithm is given in Section 4.4. In Section 4.5, we show the results for both synthetically generated and real scenes. Finally, conclusions are discussed in Section 4.6.

4.2.1 Notation

Scalar real values are represented in lowercase letters (e.g. a) and complex values, used to define phasors, are denoted by uppercase letters (e.g. S). Bold typography indicates vectors and matrices, using lowercase letters for vectors (e.g. $\mathbf{p}$) and uppercase letters for matrices (e.g. $\mathbf{A}$). The vector norm $\|\mathbf{x}\| = (\sum_i x_i^2)^{1/2}$ is the Euclidean norm. Calligraphic fonts are reserved to define sets or regions:

$$\mathcal{R}(\mathbf{p}, \delta) = \{\mathbf{q} \qquad \text{s.t.} \qquad \|\mathbf{p} - \mathbf{q}\| < \delta\}$$

We use the $\wedge$ modifier in signals, scalar amplitudes or vectors, to represent real measurements that may contain noise and distortions (e.g. $\hat{S}$ is the measured magnitude of S). In the same way, predicted magnitudes using generative models are represented by the superscript $*$. Thus, the term $\hat{S}^*$ would refer to the prediction of the magnitude $\hat{S}$.

4.3 Radiometric Model

In this section we present a generative model capable of rendering realistic ToF measurements, including MpI. We use general properties of light and surfaces to render realistic images as if they were captured in a ToF camera. The radiometric model developed in this section takes the scene geometry as an input (depth and point normals) and gives the depth map and the intensity image measured in a ToF camera as an output. In Section 4.4 we use this generative model to find the scene depth map without multipath distortion. We use a fitting algorithm that obtains the scene depth so that the generative model renders a depth image that is as close as possible to the one measured in the ToF camera. The generative model presented in this section is thus the core of our correction algorithm.

The way light propagates, reflects in the scene, and is then captured back in the camera is a very complex phenomenon. It requires details about how surfaces and camera optics reflect and scatter light. In practice, such information is never available. In this approach we propose a radiometric model based on the following general assumptions (see [69] for more details):

- All the surfaces in the scene are perfect Lambertian reflectors.
- A single isotropic infrared light source illuminates the scene. This light source is located at the projection centre of the camera. We consider that the distance between the light source and any of the pixels is negligible.
- Camera response is linear and narrow band at the infrared wavelength of the light emitted.
- Camera optics corresponds to an ideal pin-hole lens model. Each pixel captures infrared light along an unique line of sight that passes through the optical centre.

Despite these assumptions are approximations, they have been extensively used in other areas of computer vision, such as colour research [70] or Shape-from-Shading methods [71]. With Lambertian surfaces we avoid the need to know the *Bidirectional Reflectance Distribution Function (BRDF)* of each surface. Unlike the Lambertian model, real BRDFs are difficult to model and require calibration of the surface response under different light conditions and orientations. The assumptions about the camera response and optics are also usual in computer vision [23] and photogrammetry [72]. If properly calibrated, including radial and tangential distortion, a ToF camera fits the pin-hole model with high accuracy. In addition, considering the light source as an isotropic light source at exactly the centre of the camera is a common approximation that gives accurate results [41].

To illustrate the radiometric model, Fig. 4.2 shows a scene with two 3-dimensional points $\mathbf{p}_i$ and $\mathbf{p}_j$. These points belong to surfaces with local orientations defined by normal vectors $\mathbf{n}_i$ and $\mathbf{n}_j$ and infinitesimal areas dA_i and dA_j. A ToF camera consisting of two pixels, with indexes i and j, captures the scene. Pixels i and j capture the light reflected from points $\mathbf{p}_i$ and $\mathbf{p}_j$, respectively.

The ToF camera emits modulated infrared light and then computes the phase shift between the signal emitted (e.g. signal emitted in the direction of pixel i) and the reflected signal in the scene (e.g. signal received from $\mathbf{p}_i$). We can write the information taken at pixel i using a complex number (i.e. phasorial signal):

$$\hat{S}_i = \hat{a}_i e^{j\Delta\varphi(2\|\hat{\mathbf{p}}_i\|)}, \tag{4.1}$$

where $\hat{a}_i$ is the signal amplitude and $\Delta\varphi(2\|\hat{\mathbf{p}}_i\|)$ is the signal phase shift. This phase shift is a linear function of the distance between $\hat{\mathbf{p}}_i$ and the sensor. In real scenarios, multipath of light distorts phase and amplitude measurements and hence $\|\hat{\mathbf{p}}_i\| \neq \|\mathbf{p}_i\|$. The time varying cosine representation of phasor (4.1) is:

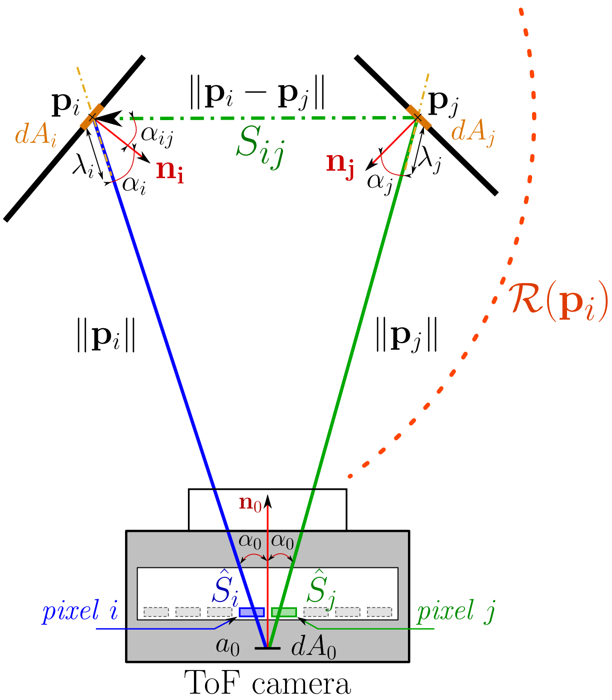


Figure 4.2: ToF camera model of a scene consisting of two points $\mathbf{p}_i$ and $\mathbf{p}_j$, associated with two infinitesimal surfaces with orientations $\mathbf{n}_i$ and $\mathbf{n}_j$, respectively. $\mathcal{R}(\mathbf{p}_i)$ is the neighbourhood region of point $\mathbf{p}_i$; in this case, only $\mathbf{p}_j$.

$$\hat{s}_i(t) = \hat{a}_i cos(wt + \Delta\varphi(2\|\hat{\mathbf{p}}_i\|)), \tag{4.2}$$

where w is the modulation frequency used for the infrared source.

To model the multipath effect, we decompose the signal $\hat{S}_i$ into the direct path signal (S_i) and the signal due to multipath (S_{ij}), which in Fig. 4.2 comes only from point $\mathbf{p}_j$.

$$\hat{S}_i = S_i + S_{ij}. \tag{4.3}$$

In this model the light travels only once from point $\mathbf{p}_j$ to $\mathbf{p}_i$. In general we can model a higher order of light bounces between the two points. However, that would clearly increase the complexity of the model. We study in Section 4.5 the impact in the results of considering only the first bounce of the light.

The two components of $\hat{S}_i$ are detailed below:

- Direct path term (S_i):

  The direct path term of the signal decomposes into the following multiplicative complex factors:

$$S_i = S_0 \; G_{0\to i} \; G_{i\to 0} \tag{4.4}$$

  The term $S_0 = a_{out}$ denotes the light emitted by each pixel. At emission, light is coherent in phase and thus $a_{out} \in \mathbb{R}$.

The complex number $G_{0\to i}$ [1] models the amplitude decrease and phase shift suffered by the light when travelling from the sensor and impacting with point $\mathbf{p}_i$:

$$G_{0\to i} = \frac{\rho_i dA_i \cos(\alpha_i)}{\|\mathbf{p}_i\|^2} e^{j\Delta\varphi(\|\mathbf{p}_i\|)} \tag{4.5}$$

Point $\mathbf{p}_i$ is embedded in a surface with differential area dA_i and orientation α_i relative to the light source. The term $\cos(\alpha_i)$ accounts for the *foreshortening* effect between the patch and the light source. The albedo factor ρ_i gives an idea of the amount of energy absorbed by the surface. In fact, the product $(S_0\ G_{0\to i})$ can be seen as a new point light source placed at $\mathbf{p}_i$.

The term $G_{i\to 0}$ models the gain and phase shift when the light at $\mathbf{p}_i$ comes back to the camera sensor and it is registered on pixel i:

$$G_{i\to 0} = \frac{\rho_0 dA_0 \cos(\alpha_0)}{\|\mathbf{p}_i\|^2} e^{j\Delta\varphi(\|\mathbf{p}_i\|)}, \tag{4.6}$$

where $\rho_0 dA_0 \cos(\alpha_0)$ is the linear response of the camera. We assume that $\alpha_0 = 0$ and dA_0 are replaced by pixel area. Jointly with ρ_0 they are considered unknown camera gain factors.

- Multipath term (S_{ij}):

The energy that hits point $\mathbf{p}_j$, apart from being irradiated towards pixel j (direct path S_j in pixel j), also impacts on pixel i (travelling a distance $d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|$ and being added to the light that reaches pixel i).

The multipath signal is factorised into the following terms:

$$S_{ij} = S_0\ G_{0\to j}\ G_{j\to i}\ G_{i\to 0} \tag{4.7}$$

where $G_{j\to i}$ is the amplitude decrease and phase shift of the light suffers when travelling from $\mathbf{p}_j$ to point $\mathbf{p}_i$:

$$G_{j\to i} = \frac{\rho_i dA_i \cos(\alpha_{ij})}{d_{ij}^2} e^{j\Delta\varphi(d_{ij})} \tag{4.8}$$

and α_{ij} is the angle between the surface at $\mathbf{p}_i$ and the vector $\mathbf{p}_i - \mathbf{p}_j$.

The term $\hat{S}_i$ (measured signal in pixel i) can be rewritten using Eqs. (4.4) and (4.7):

$$\hat{S}_i = S_i + S_{ij} = S_0\ (G_{0\to i} + G_{0\to j}\ G_{j\to i})G_{i\to 0} \tag{4.9}$$

where the terms S_0 and $G_{i\to 0}$ are common to both the direct path term and the multipath term.

[1] Notation: sub-index $x \to y$ means light travelling from x to y, where x (same for y) means point p_x. p_0 is the camera centre of projection

In a real scenario, the light reflects from a dense amount of points and arrives at point $\mathbf{p}_i$. The equations must be integrated over all the geometry of the scene, removing surface differentials. We assume that the scene is piecewise planar between the points measured by the camera pixels. As a consequence, the scene geometry is approximated with planar surface patches. These planar patches are assumed to be small enough to substitute surface differentials for small areas.

The signal received at pixel i can be expressed as:

$$\begin{aligned} \hat{S}_i &= S_i + \sum_{\mathbf{p}_j \in \mathcal{R}(\mathbf{p}_i)} S_{ij} \\ &= S_0 \left(G_{0\to i} + \sum_{\mathbf{p}_j \in \mathcal{R}(\mathbf{p}_i))} G_{0\to j}\, G_{j\to i} \right) G_{i\to 0} \end{aligned} \tag{4.10}$$

where we define $\mathcal{R}(\mathbf{p}_i)$ as the set of point (patches) that are reflecting the light towards point $\mathbf{p}_i$. This set can be found using surface orientations. A surface patch at point $\mathbf{p}_j$ with normal vector $\mathbf{n}_j$ and $\mathbf{n}_i^\top \mathbf{n}_j < 0$ does not reflect light towards point $\mathbf{p}_i$, so it must not be included in $\mathcal{R}(\mathbf{p}_i)$.

4.4 Multipath of Light Correction

In this section we propose a method that takes a single ToF image (amplitude and depth maps) and gives as a result the depth of the scene without multipath distortion. Scene geometry is not known *a priory*. Our method can be applied to any scene where the radiometric model of Section 4.3 is capable of rendering accurate ToF measurements. Section 4.5 shows that this model can be applied even to scenes containing non-Lambertian objects and high differences in albedo.

The correction algorithm can be summarised as follows: If the true depth values for each pixel are found, then the generative model presented in Section 4.3 is able to reproduce camera measurements. Following this idea we propose an optimisation method that finds the scene depth (free from MpI) that makes the generative model render the closest ToF depth map to camera measurements.

Figure 4.3 graphically shows the signals involved during our correction algorithm. We display a corner-like scene (blue curve) seen by a ToF camera. We denote as S_x the phasor corresponding to the direct path of pixel x. The phase of S_x gives the sought-after true depth of pixel x. S_x is not directly visible from the camera measurements due to noise and multipath. We denote as $\hat{S}_x$ (red curve) the measurements obtained in the camera for pixel x. From $\hat{S}_x$ we can obtain a distorted depth for pixel x, as it is clearly seen in the roundish shape of the red curve. We propose a generative model that predicts ToF measurements $\hat{S}_x^*$ (orange curve) given an estimation of the direct path depths S_x^* (green curve). The correction algorithm iteratively finds the values of S_x^* so that $\hat{S}_x^*$ is as close as possible to the measurements $\hat{S}_x$ in terms of phase (we show arrows between both signals). If the model is accurate enough, we expect the phase of S_x^* being also as close as possible to that of S_x, thus obtaining a good estimation of the depth values without MpI.

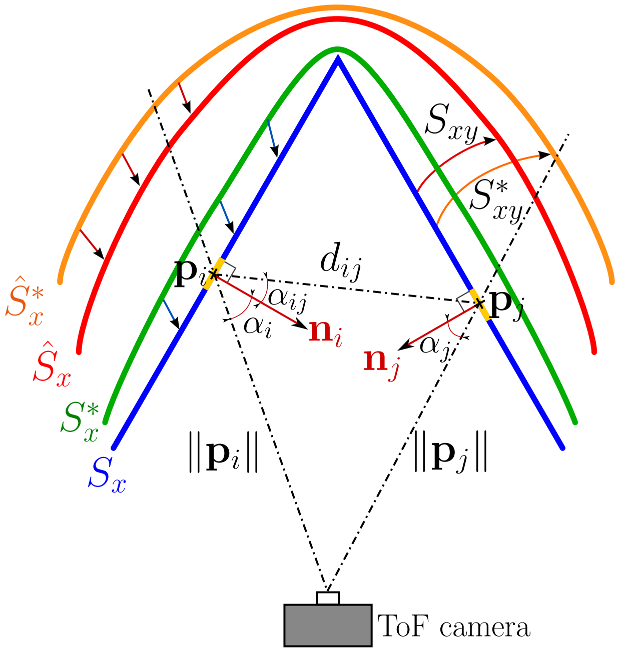


Figure 4.3: Diagram of the multipath interference effect and the correction process in a simplified situation. We show the additive influence of a certain point $\mathbf{p}_j$ in the measured signal coming from $\mathbf{p}_i$ due to the first bounce of the emitted illumination.

4.4.1 Generative Model

We describe the 3D point $\mathbf{p}_i$, measured at pixel i using a scalar depth correction λ_i:

$$\mathbf{p}_i = \hat{\mathbf{p}}_i(1 - \lambda_i), \tag{4.11}$$

where $\hat{\mathbf{p}}_i$ is the noisy measurement of $\mathbf{p}_i$ and λ_i is the correction in depth along the line of sight of pixel i.

We first present the proposed model using the simplified two point scenario shown in Fig. 4.2. Let $\hat{S}_i^*(\lambda_i, \lambda_j)$ defined as the signal in pixel i in function of depth corrections λ_i and λ_j for points $\mathbf{p}_i$ and $\mathbf{p}_j$ respectively:

$$\hat{S}_i^*(\lambda_i, \lambda_j) = S_i^*(\lambda_i) + S_{ij}^*(\lambda_i, \lambda_j), \tag{4.12}$$

where

$$\begin{aligned} S_i^*(\lambda_i) &= S_0\, G_{0\to i}^*\, G_{i\to 0}^* \\ S_{ij}^*(\lambda_i, \lambda_j) &= S_0\, G_{0\to j}^*\, G_{j\to i}^*\, G_{i\to 0}^* \end{aligned} \tag{4.13}$$

Following Eqs. (4.5),(4.6) and (4.8), the gain terms can be expressed as:

$$G^*_{0\to i} = \frac{\rho^*_i dA^*_i \cos(\alpha^*_i)}{\|\hat{\mathbf{p}}_i\|^2(1-\lambda_i)^2} e^{j\Delta\varphi(\|\hat{\mathbf{p}}_i(1-\lambda_i)\|)} \tag{4.14}$$

$$G^*_{i\to 0} = \frac{\rho_0 dA_0 \cos(\alpha_0)}{\|\hat{\mathbf{p}}_i\|^2(1-\lambda_i)^2} e^{j\Delta\varphi(\|\hat{\mathbf{p}}_i(1-\lambda_i)\|)} \tag{4.15}$$

and

$$G^*_{j\to i} = \frac{\rho^*_i dA^*_i \cos(\alpha^*_{ij})}{(d^*_{ij})^2} e^{j\Delta\varphi(d^*_{ij})} \tag{4.16}$$

where $d^*_{ij} = \|\hat{\mathbf{p}}_i(1-\lambda_i) - \hat{\mathbf{p}}_j(1-\lambda_j)\|$.

In this two-point example, given λ_i and λ_j, the proposed model predicts the signals measured in the camera considering a scene with geometry given by $\hat{\mathbf{p}}_i \cdot (1-\lambda_i)$ and $\hat{\mathbf{p}}_j \cdot (1-\lambda_j)$. There are some parameters that must be known to evaluate functions $\hat{S}^*_i$:

- Surface albedo factors ρ^*_i, included in factors $G^*_{i\to j}$. These parameters are scene-dependent. Their estimation is explained below.
- Camera response and gain, included in terms $G^*_{i\to 0}$. In general, unless the camera response is calibrated using light and surface patterns, those parameters are unknown. As it is explained later, the way the generative model is matched to the measurements makes unnecessary to obtain these parameters.

4.4.1.1 Estimation of Albedo Factors

Including surface albedos ρ_i in the model is mandatory as surfaces involved in a real scene have very different albedos (e.g. black and white colours in a checker-board). In most cases, albedo determination from images is a difficult task in computer vision, especially with uncontrolled light scenarios (see [73], [74]).

In this case we propose a method to approximate the scene albedos using ToF measurements and taking into account the radiometric model detailed in Section 4.3. Let ρ_i be the albedo factor for the surface detected at pixel i. This factor can be expressed as the product between a reflectivity term ρ^{rel}_i and a global scale factor ψ:

$$\rho_i = \psi \cdot \rho^{rel}_i, \quad \forall i \tag{4.17}$$

4.4.1.1.1 Reflectivity term ρ^{rel}_i Depth and normals of each pixel are used to decouple foreshortening and amplitude fall-off to obtain a value of ρ^{rel}_i in terms of the amplitude detected at pixel i, namely $\hat{a}_i$:

$$\rho^{rel}_i = \hat{a}_i \frac{\|\hat{\mathbf{p}}_i\|^4}{dA_i \cos(\alpha_i) k_n} \tag{4.18}$$

where ρ_i^{rel} is indeed the amplitude of the signal arriving at pixel i in a common plane parallel to the camera sensor (we use the plane in the coordinate origin). The scalar k_n is chosen so that $\rho_i^{rel} \leq 1$.

4.4.1.1.2 Global scale factor ψ. The global scale factor depends on the camera and light source used. ψ is a camera-dependent parameter and it can be calibrated. In Section 4.5 we describe a calibration algorithm.

Finally, we show in Section 4.5 that despite the fact that albedos are computed using noisy image amplitudes $\hat{a}_i$, they are fair enough to get accurate depth corrections.

4.4.2 Fitting Algorithm

In this section we present an optimisation approach that fits the generative method to ToF measurements, obtaining as a consequence the scene without MpI.

4.4.2.1 Diagram

We show in Fig. 4.4 a flowchart that explains the fitting algorithm; note that we refer to the involved signals using the same colour assignment as used in Fig. 4.3. The inputs are depth and amplitude measurements taken from a single frame capture in the sensor (green box in the diagram). We call *Solution(k)* in the diagram to the estimation of the depths without MpI at iteration k of the algorithm. At $k = 0$, sensor measurements are used as initialisation. Given *Solution(k)*, we simulate the depths with MpI. This is called *Solution(k)+MpI* in the diagram. We use the radiometric model explained before to simulate MpI, using amplitude measurements to estimate albedos. The iterations stop when the differences in depth between the measurements and *Solution(k)+MpI* are below some predefined threshold γ. After the method stops, *Solution(k)* contains the depths for each pixel without multipath errors.

4.4.2.2 Cost Function

To have guarantees of convergence we implement diagram 4.4 using a cost function that is optimised with iterative methods. The cost function involves the depth correction of all pixels in the image:

$$\mathbf{\Lambda} = (\lambda_1, \cdots, \lambda_N), \tag{4.19}$$

grouped in the column vector $\mathbf{\Lambda}$ for an image with N pixels. We propose the following sum of squares cost function:

$$\underbrace{\min}_{\{\mathbf{\Lambda}\}} \left(\sum_{i=1}^{N} \| f_i(\mathbf{\Lambda}) \|^2 \right) \tag{4.20}$$

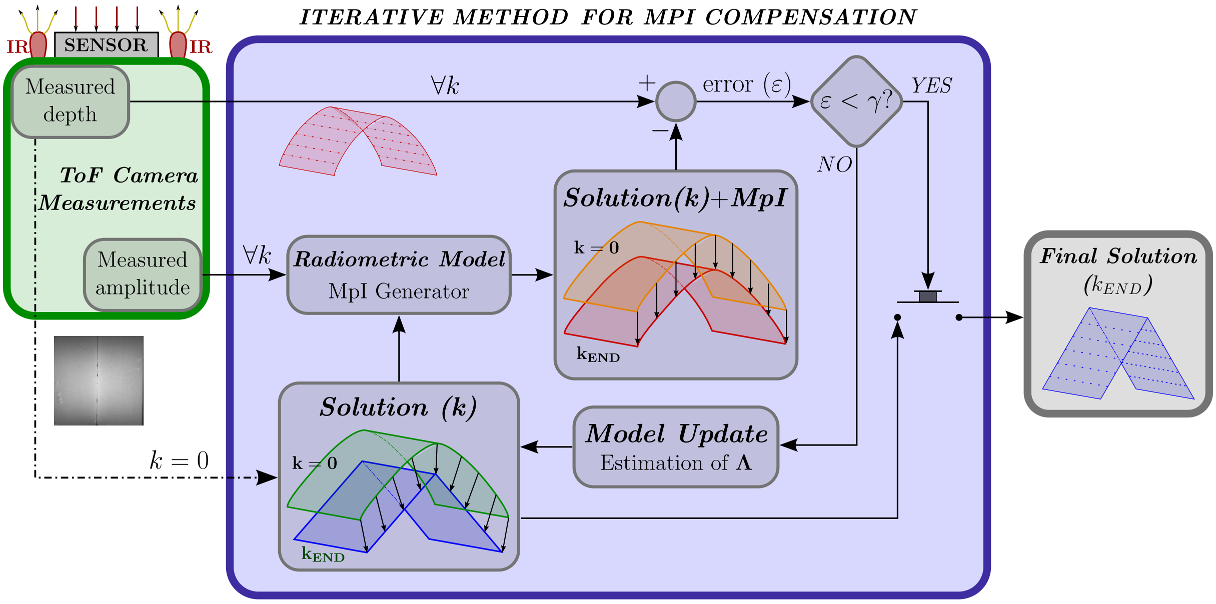


Figure 4.4: General diagram of the iterative algorithm for multipath interference correction. The term k indicates the iteration number and $\mathbf{\Lambda}$ is the unknown vector (depth corrections).

where:

$$\begin{aligned} f_i(\mathbf{\Lambda}) &= \phi\left(\hat{S}_i\right) - \phi\left(\hat{S}^*_i(\mathbf{\Lambda})\right) \\ &= \phi\left(\hat{S}_i\right) - \phi\left(S^*_i(\lambda_i) + \sum_{j\in\mathcal{R}_i} S^*_{ij}(\lambda_i,\lambda_j)\right), \end{aligned} \tag{4.21}$$

and $\phi(S)$ is the phase of the phasor S.

As is displayed in Fig. 4.5 the optimisation approach is forcing every two phasors $\hat{S}_i$ and $\hat{S}^*_i$ to be as parallel as possible.

The cost function (4.20) is non-linear and non-convex in terms of the unknown vector $\mathbf{\Lambda}$. In this study a numerical optimisation algorithm is used to reach a local minimum of the cost. In particular the Levenberg-Marquardt algorithm is proposed, using as initialisation $\mathbf{\Lambda} = \mathbf{0}$.

Although the global minimum of the cost is not ensured, in Section 4.5 we show with experiments that the ToF measurements give a good initialisation so that only a few iterations are needed to reach the sought-after local minimum.

4.5 Results

4.5.1 Implementation Details

In this section we give practical hints and methods to calibrate ψ from checker-board patterns and the algorithm we propose to compute surface normals and areas.

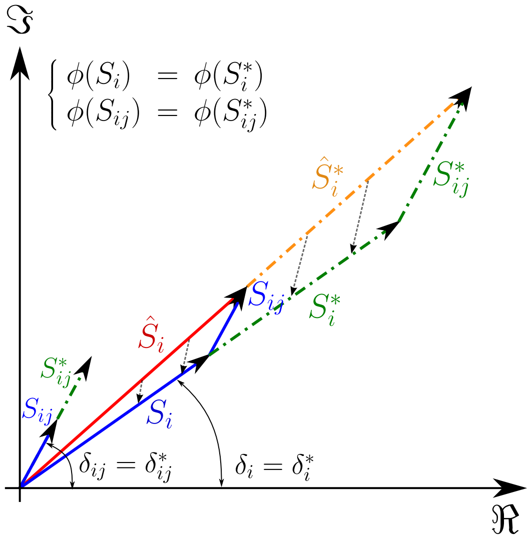

$\Im$
$\begin{cases}\phi(S_i) &= \phi(S_i^*)\\ \phi(S_{ij}) &= \phi(S_{ij}^*)\end{cases}$
$\hat{S}_i^*$
S_{ij}^*
S_{ij}
S_i^*
$\hat{S}_i$
S_{ij}^*
S_{ij}
S_i
$\delta_{ij} = \delta_{ij}^*$
$\delta_i = \delta_i^*$
$\Re$

Figure 4.5: Phasorial diagram with the signals involved in the MpI model. Please note that in this figure we use $\phi(S_x) = \delta_x$ in order to improve its readability.

4.5.1.1 Calibration of Global Scale Factor ψ

The value of ψ depends on camera parameters (e.g. exposition time or camera response), and the light intensity. The global scale factor ψ does not depend on the scene under consideration. It is thus a parameter that we can calibrate independently in a calibration process that we describe with the following steps:

1. We use calibration patterns to obtain a set of 3D points from which we know the ground truth and the multipath distortion at every point. We use two planes with a chessboard texture where we know distances d_{ij} between corner points (see Fig. 4.6). These planes must show multipath effect so that the model can predict ψ. The best way is to create a corner with the two planes.

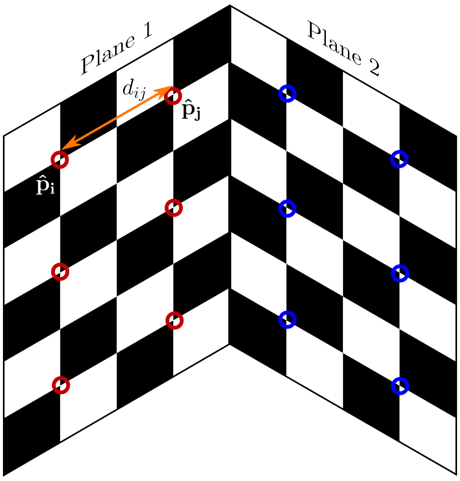


Figure 4.6: Simulation of the experimental setup to calibrate the global scale factor ψ. We establish two planes in a corner like configuration so that multipath interference appears in the scene (any other setting would be possible). Both planes must be imprinted with a chessboard pattern of known distances.

Let $\hat{\mathbf{p}}_i$ be the measurements of the 3D coordinates of the 3D point corresponding to pixel i and $\mathbf{p}_i = \hat{\mathbf{p}}_i(1-\lambda_i)$ the correction of that point. Using the fact that we know distances in the pattern we can write:

$$d_{ij} = \|\hat{\mathbf{p}}_i(1-\lambda_i) - \hat{\mathbf{p}}_j(1-\lambda_j)\| \tag{4.22}$$

We force every pair of points i and j detected from the patterns to meet condition (4.22). We use the following cost function:

$$\underbrace{\min}_{\mathbf{\Lambda}} \left\{ \sum_{i=1}^{M} \sum_{\substack{j=1, \\ j \neq i}}^{M} \left(d_{ij}^2 - \|\hat{\mathbf{p}}_i(1-\lambda_i) - \hat{\mathbf{p}}_j(1-\lambda_j)\|^2 \right)^2 \right\} \tag{4.23}$$

where the vector

$$\mathbf{\Lambda} = (\lambda_i)_{M\times 1}, \quad \forall i \in [1, ..., M]$$

corresponds to the depth corrections for the M detected points in the planes. The minimum of Eq. (4.23) with respect to $\mathbf{\Lambda}$ is obtained easily with non linear refinement algorithm such as Levenberg-Marquardt. We assume that $\mathbf{\Lambda} = \mathbf{0}$ at the first iteration.

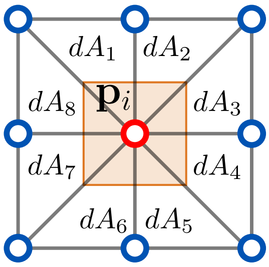
2. Knowing λ_i and $\rho_i^{rel}, \ \forall i \in [1, ..., M]$ values, the global scale factor ψ can be obtained by minimising the following cost function in terms of ψ as the only unknown in the optimisation process:

$$\underbrace{\min}_{\psi} \left(\sum_{i=1}^{N} \|cf_i\|_2^2 \right) \quad / \quad cf_i = \Delta\varphi\left(\hat{S}_i\right) - \Delta\varphi\left(\hat{S}_i^*\right) \tag{4.24}$$

We use also non-linear refinement to obtain the value of ψ. We use $\psi = 1$ as initial value.

4.5.1.2 Computation of Areas

As commented in Section 4.3, each point of the scene integrates a small area that substitutes the differential area terms included in the cost function. We employ as mask the 8 nearest neighbours around the point of interest in the image to get an estimation of the area. If the point is near a discontinuity we remove the neighbours at a distance greater than a predefined threshold (e.g. we use a distance threshold of 80 mm in all experiments). Finally, the area of a point in the scene corresponds to a quarter of the area covered by all 3D triangles formed by the point and its neighbours (see Fig. 4.7).



$$dA_i = \frac{1}{4}\sum_{n=1}^{N} dA_n$$

Figure 4.7: Differential area of a sampling point $\mathbf{p}_i$ with its N neighbours inside the distance threshold (general case: $N = 8$).

4.5.1.3 Computation of Surface Normals

We need surface normals for the radiometric equations. We compute normals from the sampled depths using the method proposed in [75], which is based on fitting planes around each point, using neighbour pixels. The number of neighbours depends on the distance to the point of interest. In other words, the plane is fitted with points covering approximately the same area regardless of the resolution of the device; for example, we fix a distance threshold between the neighbours and the point of interest of 80 mm. In Fig. 4.8 we show the normals obtained for two real examples of corner-like scenes.

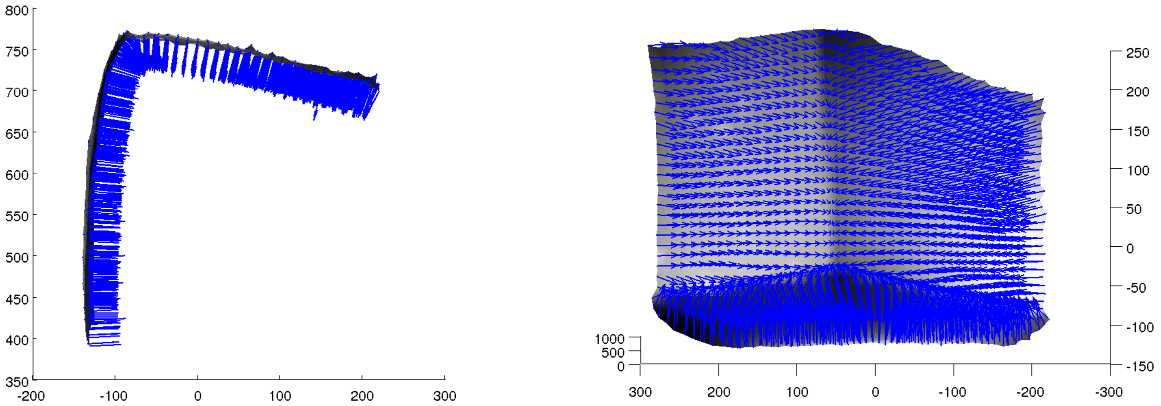


Figure 4.8: Estimation of normal vectors in real scenes (we consider a distance threshold of 80 mm).

4.5.2 Results I: Synthetic Data

4.5.2.1 Noise and Influence of Global Gain for Albedos

Table 4.1 shows the results for two different synthetically generated scenes based on the radiometric model described in Section 4.3. Simulations were performed for ToF images with low resolutions (e.g. 20 × 20 pixels).

In order to measure the amount of correction performed by our method we compare, for each pixel, the 3D positions of the ground truth points against the corresponding estimations obtained from: i) ToF measurements ($\hat{S}_i$) without MpI correction. ii) Predicted value of the direct path term (S_i^*) after the optimisation algorithm reach a minimum.

The first two columns of Table 4.1 show the influence of additive noise in depth measurements, while the last two columns show the influence of the global scale factor ψ used for the albedos. In both cases we show two kinds of plots. In the first row we plot a view of the depth measurements, ground truth and the proposed correction in the X-Z plane; in the second row we show the corresponding *rms* error in function of the two parameters under study.

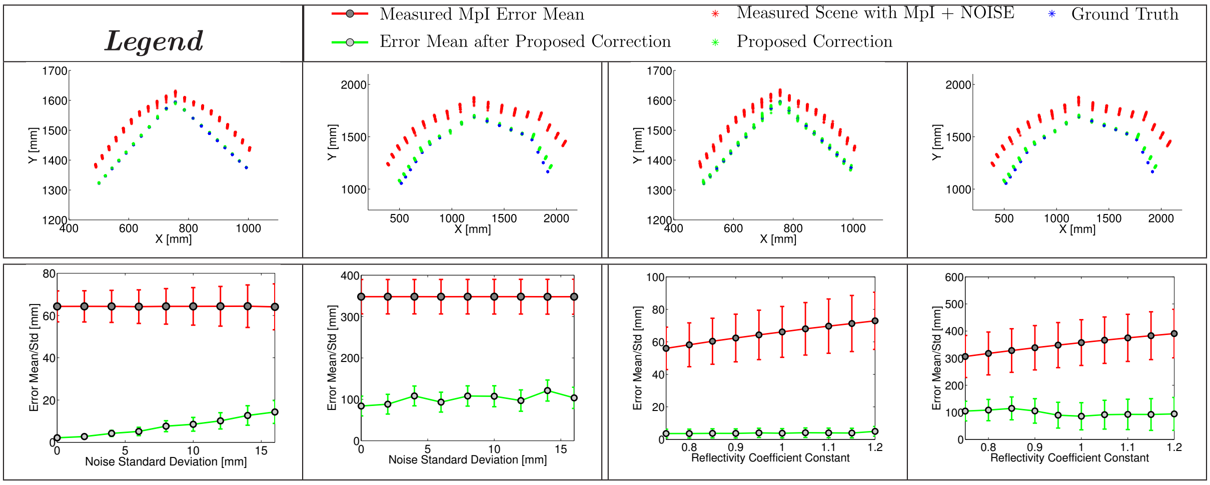


Table 4.1: Synthetic data. The first two columns show the influence of additive noise in depth measurements; we fix the global scale factor to $\psi = 1$. The last two columns show the influence of the global scale factor ψ used for the albedos; we fix noise standard deviation to $\sigma_{NOISE} = 6mm$. In all cases we use 30 random realisations of the same experiment.

There are two main aspects to highlight. First, the stability of the correction with respect to the noise introduced in the depth signal, which remains close to the value of the noise added and greatly corrects the MpI. Second, it must be noted that the performance of the proposal is also highly independent of the reflectivity coefficient range (up to 30% of increment in global energy absorption) introduced in the MpI generation model.

4.5.2.2 Validation of the First Bounce Model

Theoretically, MpI arises from multiple reflections in the scene of $\mathcal{O}$-th order (see the example in Fig. 4.9 for $\mathcal{O} = 2$). Our first bounce assumption ($\mathcal{O} = 1$) is based on the signal decrease due to the foreshortening phenomena and the quadratic inverse dependency to the distance as seen in Eq. (4.5). In this section, we validate our first bounce approach in synthetically generated scenarios by introducing the second bounce term in the generative model that simulates multipath interference (see Eq. (4.25)).

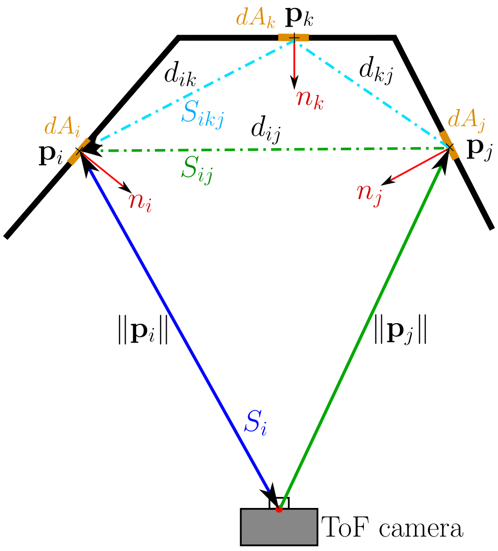


Figure 4.9: Conceptual representation of the signals involved in the multipath interference considering a second bounce model. We show a simplified scenario consisting of 3 points and the specific point-to-point ray tracing.

$$\begin{aligned}
\hat{S}_i^{*(2)} &= S_i^* + \sum_{j\in\mathcal{R}(\mathbf{p}_i)} \left\{ S_{ij}^* + \sum_{k\in\mathcal{R}(\mathbf{p}_j)} S_{ikj}^* \right\} \\
S_{ijk}^*(\lambda_i,\lambda_j,\lambda_k) &= S_0\, G_{0\to j}^*\, G_{j\to k}^*\, G_{k\to i}^*\, G_{i\to 0}^*
\end{aligned} \tag{4.25}$$

In the same way as in the previous section, we present in Table 4.2 the influence of additive noise and the calibrated constant ψ in the proposed correction. The results show that adding the second bounce term does not incorporate significant error in the algorithm and the correction maintains similar levels if we compare with first bounce model. Given that results, we can theoretically validate our generative model.

4.5.3 Results II: Real Scenarios

4.5.3.1 State of the Art Comparison

In the first row of Table 4.3 we show the behaviour of the proposal for a commercial PMD CamCube ToF camera (for more information visit [68]) of 200×200 pixel resolution. The scene consists of a homogeneous colour corner where we can easily appreciate the multipath effect.

The second row of Table 4.3 shows similar results using the MESA-Imaging ToF camera (for more information visit [67]) of 176×144 pixel resolution. We employ a 1:2 re-sampling ratio in the following experiments, so a small error may be introduced due to normal vector computation. Normal vectors have been computed with a 5-neighbourhood kernel and a distance threshold of 500 mm. In this case, the scene consists of a corner with a chessboard pattern.

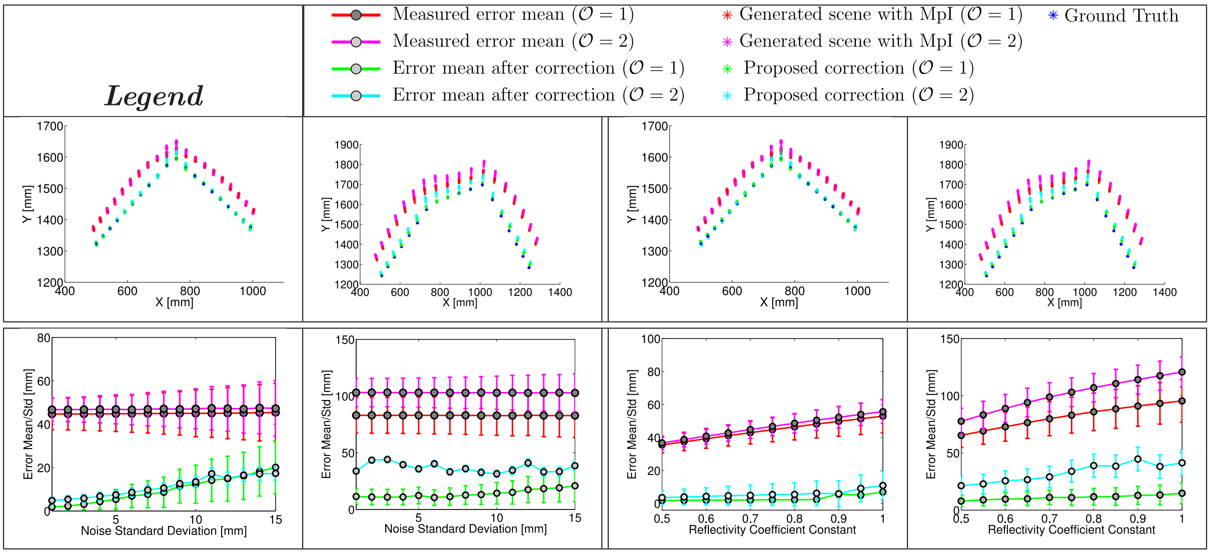


Table 4.2: Synthetic data. The first two columns show the influence of additive noise in depth measurements; we fix the global scale factor to $\psi = 0.75$. The last two columns show the influence of the global scale factor ψ used for the albedos; we fix noise standard deviation to $\sigma_{NOISE} = 6$. All the experiments employ 30 iterations for each sweep iteration.

In Table 4.3 we can see that the amount of multipath present in the scene taken by the MESA-Imaging device is significantly smaller than the one seen in the PMD CamCube experiment. This is due to the fact that the checker-board does not reflect as much light as the uniform surface reflector.

In both scenes of Table 4.3 we compare our approach with the method proposed in [41]. As it is clear in Table 4.3, our proposal is able to correct MpI accurately, clearly surpassing the method proposed in [41]. In the second row of Table 4.3 it can be observed that Fuch's method is not able to correct the scene, due to albedo differences of the checker-board.

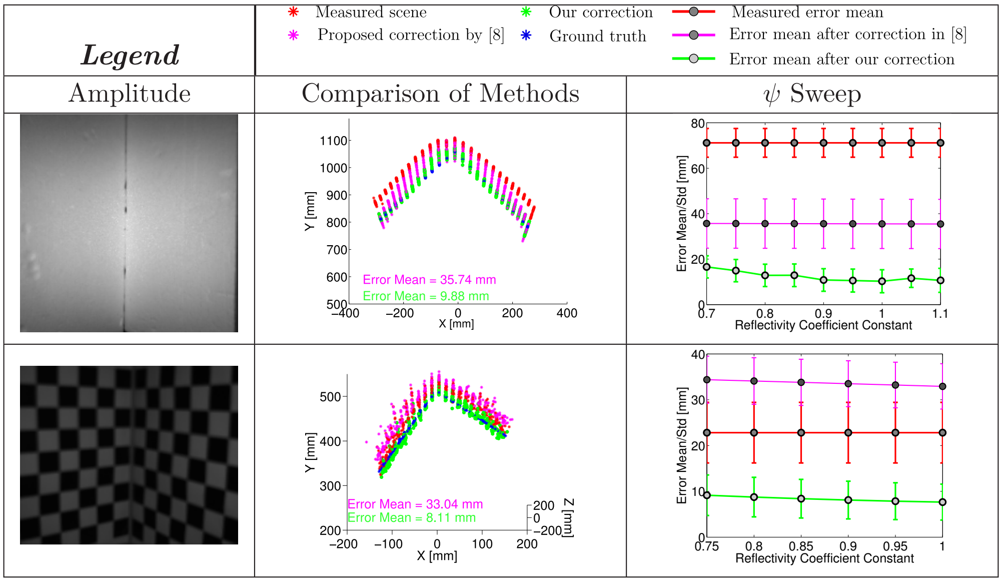


Table 4.3: Comparative results of real corners captured with 2 different ToF cameras. We provide a lateral view of the 3D reconstruction and the average error for each method.

4.5.3.2 Complex Scenarios

In Tables 4.4 and 4.5 we show the reconstructed scenes after applying the proposed method using a commercial time of flight camera (see [67]). In the first column we show the point clouds of the measured (in red) and the corrected scene (in green). On these scenes our baseline is obtained using a range laser sensor, with distance accuracy of $\pm 10mm$, attached to a servo motor. With this device we can reconstruct the scene with minimal presence of MpI as each measurement of the laser is made independently. In columns 3 and 4 we show the initial error from the measurement and the remaining error after our correction, respectively. We represent the value of the error for each point with a different colour following the colour-map of the attached bar (values are given in mm). In the last column we show the error mean and the standard deviation for different values of the general scale factor ψ. Error mean is computed as the mean of the euclidean distance between each measured point and its corresponding point from the baseline.

We can observe that in most cases the correction keeps stable after a value of ψ close to 1 or reduces their slope significantly. Calibrated values close to that constant provides good results (in our case $\psi = 1.45$ for experiments 1-13 and $\psi = 1.12$ for experiments 14-15). The outcomes are validated in different scenarios in various ranges up to the maximum range of the employed laser to elaborate the ground truth (4 m). We provide values of the error mean, Tables 4.6 and 4.7, with two different sub-sampling ratios ($sr = 4$ and $sr = 8$, respectively). The similarity between both remaining errors demonstrate the viability of future efficient implementations that consider homogeneous patches of higher size to speed up the computation of multipath terms. This assumption could reduce the computation complexity of the algorithm and allow real time processing.

4.5.3.3 Large Environment

Finally, in Table 4.8 we validate our algorithm in a complex scene (lab room view). With this purpose, we have developed the correspondence between a high-resolution RGB camera and our ToF device so we can texturise the scene. In this experiment the ground truth is not available. However, it can be observed in the second row of Table 4.8, by simple visual inspection of the texturised mesh, the way our proposal corrects the deformed geometry of the scene captured by the ToF camera.

4.5.3.4 Evaluation of the Proposal with a Scene at Different Distances

In ToF cameras the error usually increases with depth (see [76], [17] for a comprehensive survey of systematic errors in ToF cameras). In order to show if this error affects the proposed algorithm, in Table 4.9 we validate our proposal for the same scene placed at different distances to the camera. We show that correction error keeps stable in all the cases even if the initial error and the amount of multipath increase with the distance to the camera.

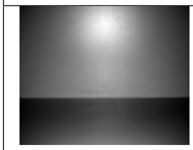
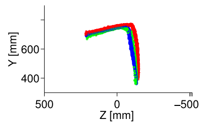
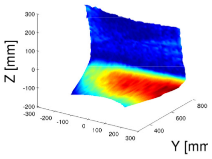
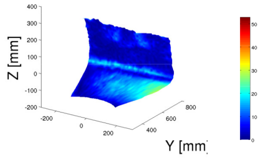
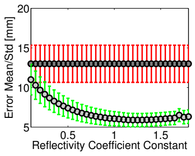
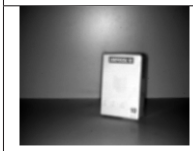
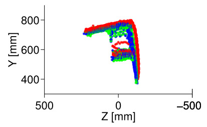
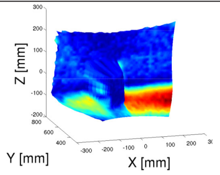
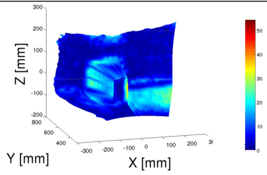
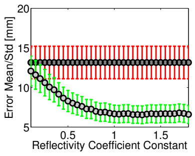
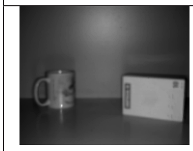
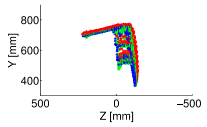
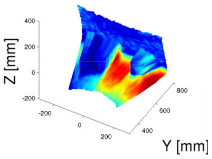
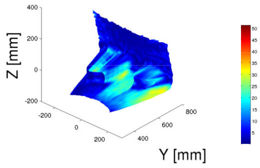
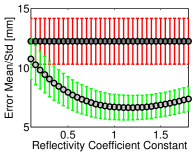
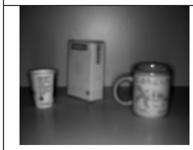
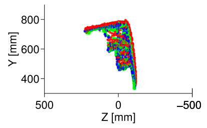
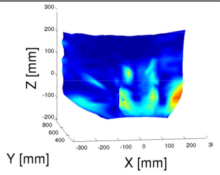
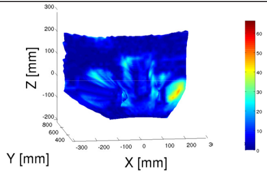
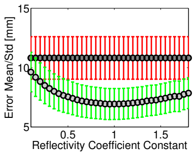
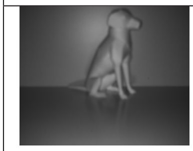
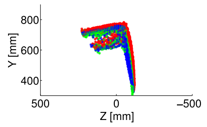
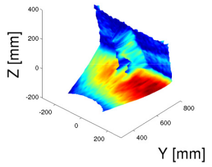
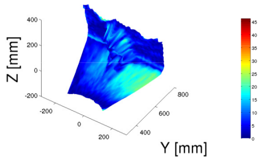
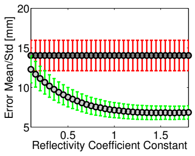
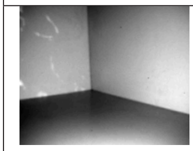
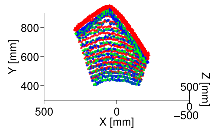
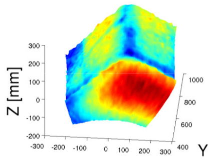
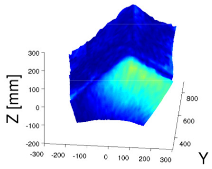
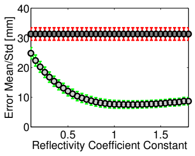
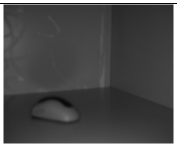
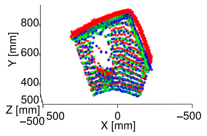
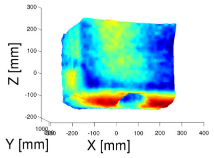
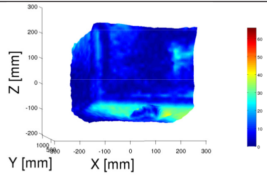
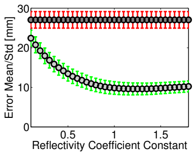
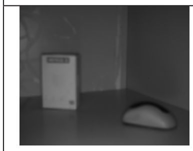
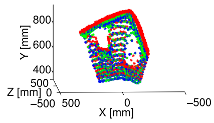
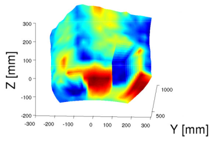
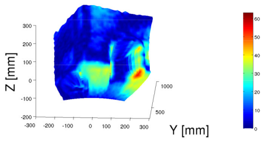
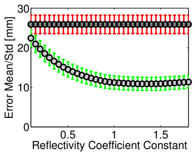
Amplitude	Point Cloud	Measurement Error	Correction Error	ψ Sweep
 1				
 2				
 3				
 4				
 5				
 6				
 7				
 8				

Table 4.4: Real scenes captured with ToF cameras (1-8). We compare point clouds taken with the camera, our MpI correction and the ground truth. We also show 3D meshes with raw and corrected depth values.

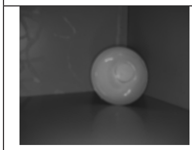
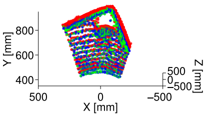
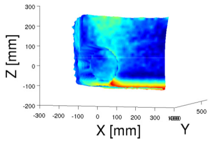
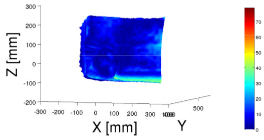
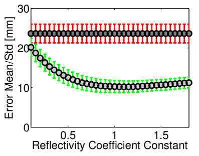
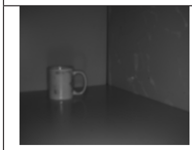
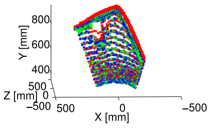
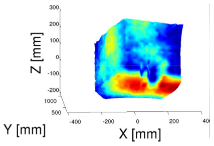
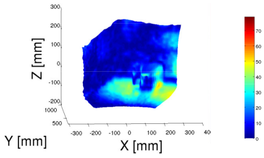
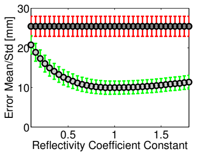
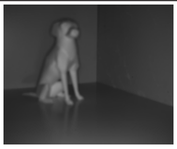
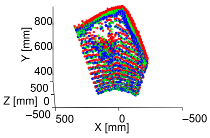
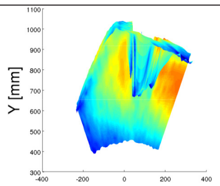
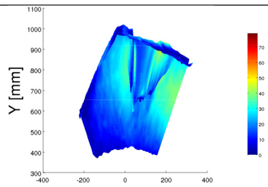
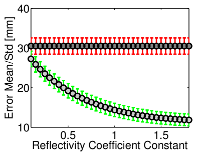
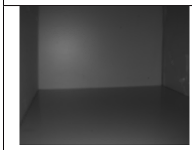
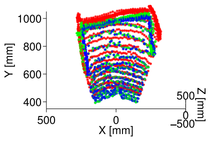
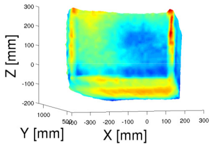
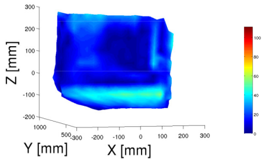
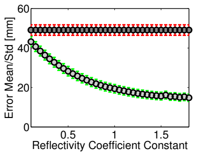
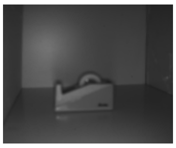
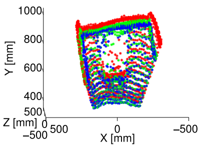
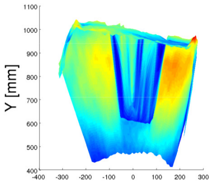
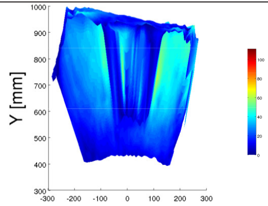
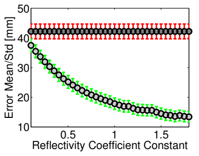
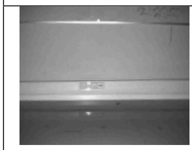
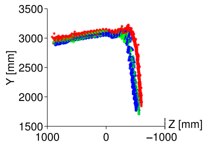
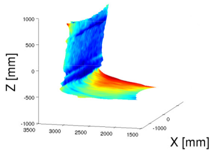
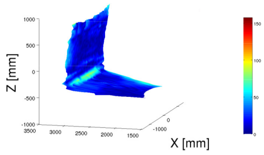
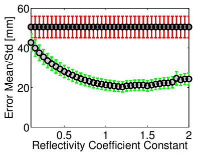
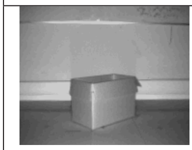
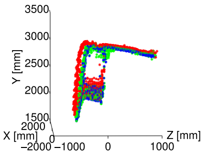
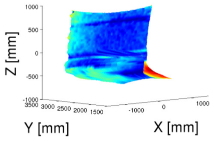
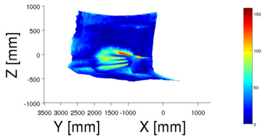
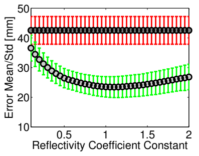
Amplitude	Point Clouds	Measurement Error	Correction Error	ψ Sweep
 9				
 10				
 11				
 12				
 13				
 14				
 15				

Table 4.5: Real scenes captured with ToF cameras (9-15). We compare point clouds taken with the camera, our MpI correction and the ground truth. We also show 3D meshes with raw and corrected depth values.

Computed Error Mean (mm)			
Example	Measurement	Correction$^{sr=4}$	Correction$^{sr=8}$
1	14.2452	6.4415	6.1768
2	14.1558	6.6981	6.2914
3	13.3690	6.3741	6.6369
4	11.8133	7.0452	6.9401
5	15.1268	6.8601	6.9102
6	31.3108	7.4176	7.7069
7	28.0717	8.3186	8.6161
8	26.9024	9.2494	9.9066

Table 4.6: Comparison between the error mean of the raw measurement and the proposed correction with two different sample ratios ($\{sr = 4, sr = 8\}$) in scenes 1-8.

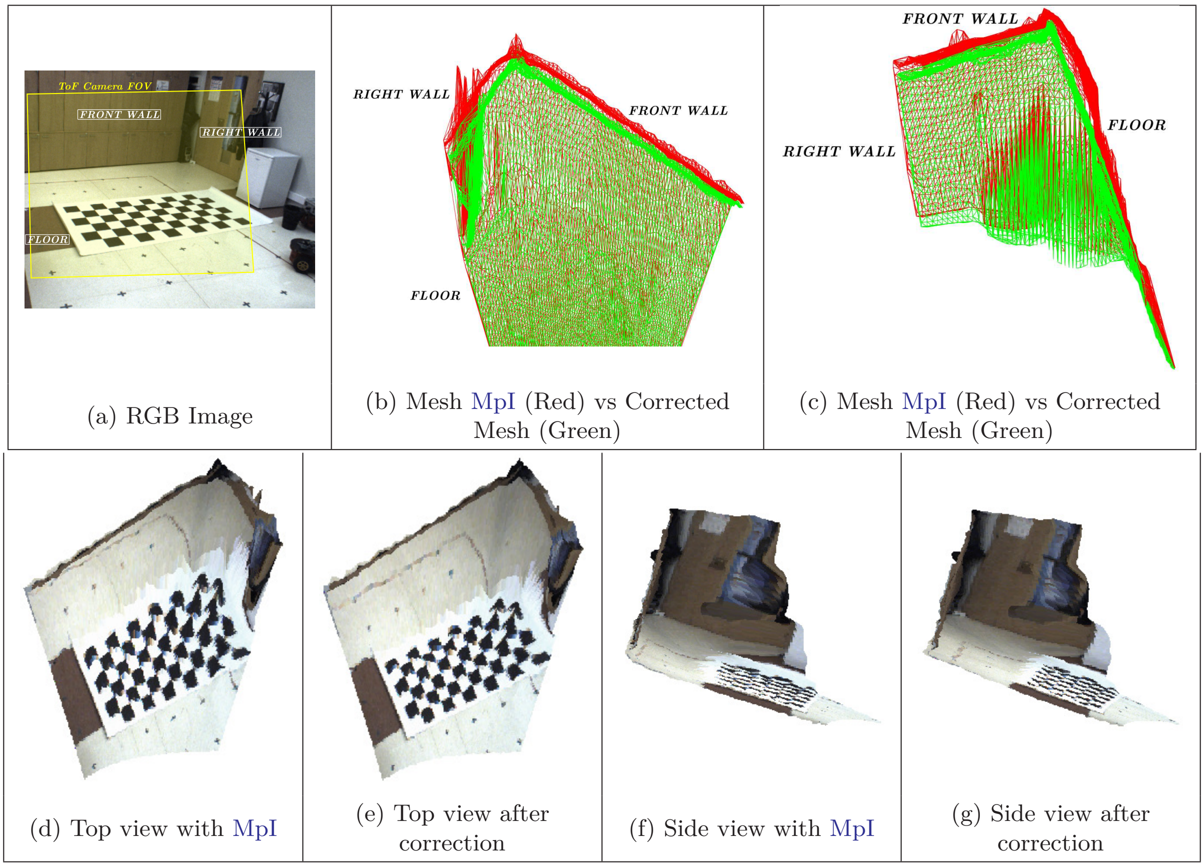
Computed Error Mean (mm)			
Example	Measurement	Correction$^{sr=4}$	Correction$^{sr=8}$
9	24.7013	9.8669	10.2845
10	26.4332	9.9391	10.1186
11	31.7162	12.0318	11.8846
12	49.7808	12.6716	11.8345
13	43.1849	12.5636	11.3301
14	51.7631	18.7637	22.0355
15	43.8015	19.3512	22.4829

Table 4.7: Comparison between the error mean of the raw measurement and the proposed correction with two different sample ratio ($\{sr = 4, sr = 8\}$) in scenes 9-15.

4.6 Conclusion

This work presents a method to correct the distortion due to multipath interference (MpI) in ToF cameras. We propose a radiometric model able to synthesise ToF measurements, including the MpI phenomenon. Using the radiometric model, we propose a method to remove multipath distortion, recovering the original depth of the scene. The solution we propose is based on unconstrained nonlinear optimisation. We obtain the scene depth from which the radiometric model renders the closest ToF depth map to the one actually measured by the camera. We show in the Results section that the method clearly corrects complex scenes captured with different ToF cameras. In all experiments we also show that our approach obtains better results when compared to the state of the art techniques.

At the moment, computational cost is high (taking several minutes for a small scene). However, we have several research lines opened that establish different methods to push the approach to future real time implementations.



(a) RGB Image

(b) Mesh MpI (Red) vs Corrected Mesh (Green)

(c) Mesh MpI (Red) vs Corrected Mesh (Green)

(d) Top view with MpI

(e) Top view after correction

(f) Side view with MpI

(g) Side view after correction

Table 4.8: Complex scenario captured with MESA ToF camera (lab room). We provide different views of the 3D meshing where the geometry distortion caused by MpI becomes more significant.

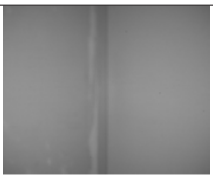
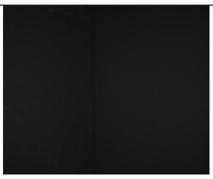
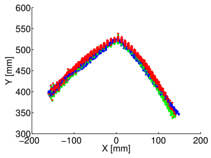
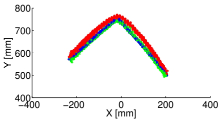
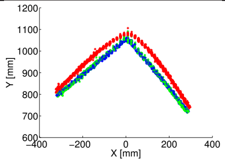
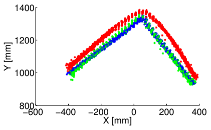
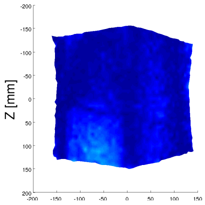
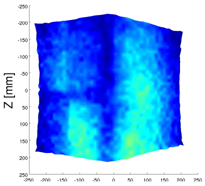
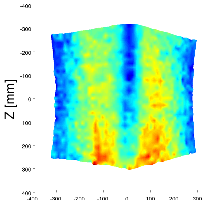
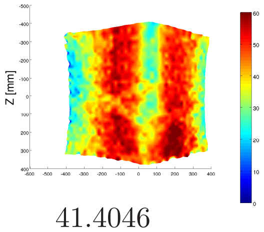
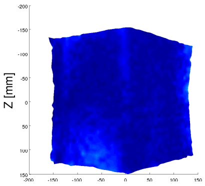
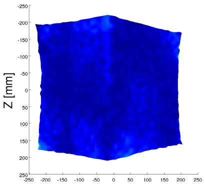
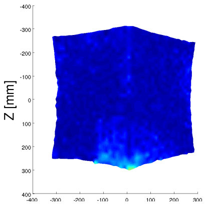
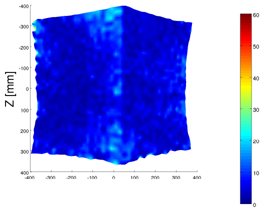
Computed Error Mean of the Same Scene for Several Depths				
Amplitude & Max. Depth $[mm]$	 500	 800	 1100	 1400
Point Cloud				
Meas. Error Mean (mm)	 8.0408	 13.0536	 26.1183	 41.4046
Correction Error Mean (mm)	 3.3268	 3.5322	 3.6077	 5.4879

Table 4.9: Real scene captured with a commercial ToF camera. We compare point clouds taken with the camera, our MpI correction and the ground truth. We also show the error mean before and after the proposed correction.

Chapter 5

Single Frame Correction of Motion Artifacts in PMD-based Time of Flight Cameras

5.1 Introduction

High motion dynamics in the scene causes significant distortion in depth estimation, specially in the vicinity of depth discontinuities and points with strong texture changes. If motion happens during the integration time T_{int} (30-2000 μs in commercial cameras), a pixel integrates incoherent signals, which leads to remarkable artifacts in the depth image. To illustrate the problem, we show in Fig. 5.1 an example of both the effect of motion artifacts and our correction. Left column shows depth and amplitude images of a moving planar object, registered with a commercial *Time of Flight (ToF)* camera, while right column shows the result of our algorithm, where the correct shape is recovered.

This work proposes a method that removes motion artifacts from a single frame captured by a *Photonic Mixer Device (PMD)* ToF camera without altering current hardware. Our method exploits spatial and temporal redundancy present in current PMD pixel pipeline to correct depth measurements caused by the motion. We show in this chapter that as a result of the proposed correction method, a single PMD ToF image can be used to measure motion vectors along occluding contours of the scene in the image. Our method is very accurate, recovering depth in very challenging scenarios under high dynamics. It outperforms the state of the art based on using several frames for correction. In addition, our method runs in real time in a single core CPU processor, does not require high amount of memory and can be integrated in camera's hardware.

5.2 Overview

One of the leading Time of Flight imaging technologies for depth sensing is based on Photonic Mixer Devices (PMD). In PMD sensors each pixel samples the correlation between emitted and

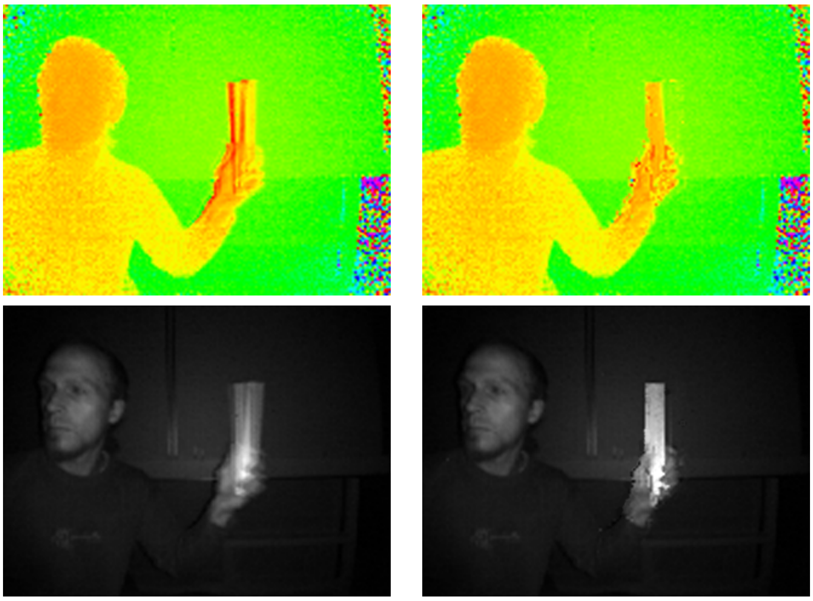


Figure 5.1: Planar object moving in a perpendicular direction in front of the camera. On the left column we show the captured depth and amplitude images, respectively. On the right column we show the proposed correction.

received light signals. Current PMD cameras compute eight correlation samples per pixel in four sequential stages to obtain depth with invariance to signal amplitude and offset variations. With motion, PMD pixels capture different depths at each stage. As a result, correlation samples are not coherent with a single depth, producing artifacts. We propose to detect and remove motion artifacts from a single frame taken by a PMD camera. The algorithm we propose is very fast, simple and can be easily included in camera's hardware. We recover depth of each pixel by exploiting consistency of the correlation samples and local neighbourhood. In addition, our method obtains the motion flow of occluding contours in the image from a single frame. The system has been validated in real scenes using a commercial low-cost PMD camera and high speed dynamics. In all cases our method produces accurate results and it highly reduces motion artifacts.

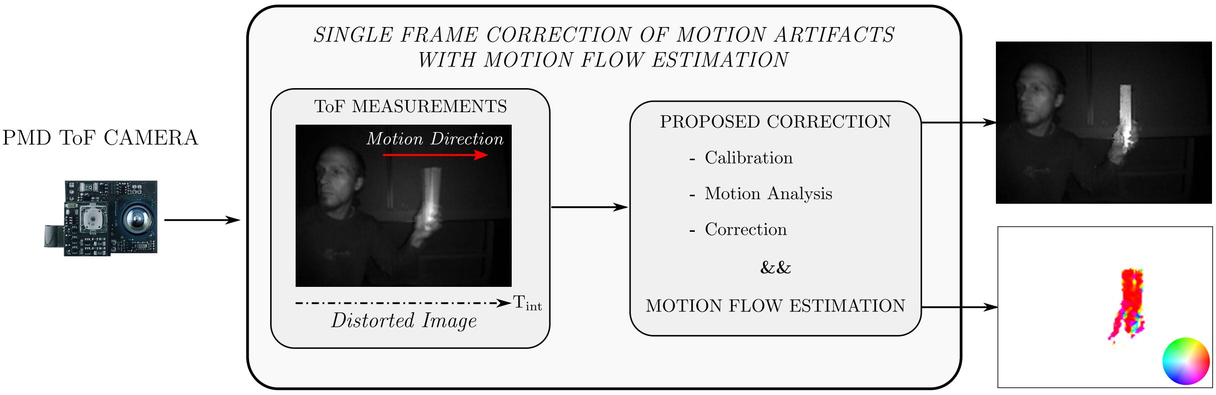


Figure 5.2: Graphical abstract of the proposed method.

This chapter is structured as follows. A description of Time of Flight fundamentals and motion artifacts phenomena is given in Sections 5.3 and 5.4, respectively. Our single frame approach and the motion estimation algorithm are given in Sections 5.5 and Section 5.6, respectively. Quantitative and qualitative results obtained using a commercially available PMD-based ToF camera are shown in Section 5.7. Finally, conclusions are discussed in Section 5.8.

5.3 Preliminary Concepts

ToF technology obtains depth from the phase difference between the emitted and the received signal after colliding with the scene. We define $u_m(t)$ as the T periodic square signal used to modulate light:

$$u_m(t) = \begin{cases} 1, & \textit{if} \quad 0 \leq t - n \cdot T \leq {}^{T}\!/_{2} \\ 0, & \textit{if} \quad {}^{T}\!/_{2} \leq t - n \cdot T \leq T \end{cases} . \qquad n = 0, 1, 2, \ldots \tag{5.1}$$

The received signal $r(t)$, reflected from the scene, is represented with the following continuous wave function:

$$r(t) = a_0 \cdot \sin(\omega t + \beta) + O, \tag{5.2}$$

where a_o is the signal amplitude, $\omega = {}^{2\pi}\!/_{T}$ and $\beta = \omega \cdot T_L$ is the phase shift caused by the time-of-flight T_L. Depth is then computed as $d = \frac{c \cdot T_L}{2}$, with c the speed of light in the medium (i.e. vacuum). The offset O is composed of two additive terms:

$$O = O_{bg} + O_{dc}, \tag{5.3}$$

where O_{bg} is due to background illumination and O_{dc} appears as the DC component of the modulated *infrared (IR)* light source.

PMD technology is based on sampling the correlation function between $r(t)$ and $u_m(t)$ to obtain depth:

$$\rho_\tau = K_{int} \cdot \frac{1}{T} \int_0^T u_m(t + \tau) \cdot r(t)\ dt = K_{int} \cdot \left(\frac{a_0}{\pi} \cdot \cos(\beta + \tau) + \frac{O}{2} \right), \tag{5.4}$$

where a_0, β and O are the unknown. K_{int} takes into account the total number of periods T integrated over the integration time of the sensor T_{int}, which is thousand times bigger than T.

Each PMD *smart pixel* contains 2 CMOS wells that accumulate voltage that corresponds to samples of the correlation function ρ_τ and $\rho_{\tau+\pi}$ (see [77] for further details). We refer to the voltage accumulated on each well as U_τ^A and U_τ^B. Their relationship with ρ_τ can be approximated by the following linear function:

$$U_\tau^A = G^A \cdot \rho_\tau + O^A \qquad U_\tau^B = G^B \cdot \rho_{\tau+\pi} + O^B, \tag{5.5}$$

where G^A and G^B are the conversion gains that translate optical energy into voltage and O^A and O^B are the offsets. In general $G^A \neq G^B$ and $O^A \neq O^B$ due to non-isotropic material properties that appear during sensor manufacturing. Lindner [78] gives a detailed description of these variations called Fixed Pattern Noise (FPN). Despite being considered as noise, it is a systematic error that can be calibrated.

Taking the difference $U_\tau^A - U_\tau^B$ we get:

$$\begin{aligned} \varphi_\tau &= U_\tau^A - U_\tau^B = G^{\text{FPN}} \cdot \left(K_{int} \cdot \frac{2 \cdot a_0}{\pi} \cdot \cos(\beta + \tau) \right) + O^{cal} \\ &= a \cdot \cos(\beta + \tau) + O^{cal} \end{aligned} \tag{5.6}$$

where G^{FPN} is the FPN gain and O^{cal} is the offset term which depends on the background illumination O and can be calibrated:

$$G^{\text{FPN}} = \frac{G^A + G^B}{2} \tag{5.7a}$$

$$O^{cal} = \frac{O}{2} \cdot K_{int} \cdot (G^A - G^B) + (O^A - O^B). \tag{5.7b}$$

There are thus a, β, G^{FPN} and O^{cal} as unknowns, whereas only β contains depth information.

Current PMD devices implement the *4 phase-shift* architecture, which consists on a 4-stage pipeline where φ_τ is computed at $\tau = \{0, \pi/2, \pi, 3\pi/2\}$ in 4 sequential stages. In Fig. 5.3 we show a summarised cross-section of a PMD-based *smart pixel* with the acquisition pipeline (notice that the capture of every phase image is followed by a readout gap). Q^A and Q^B are the accumulated charges in each of the two well.

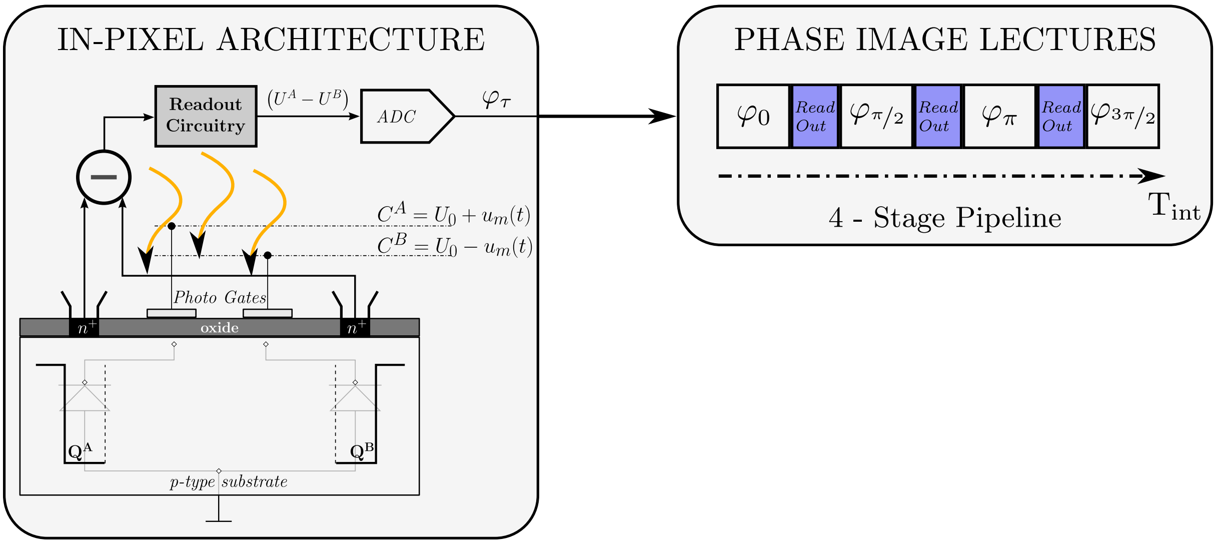


Figure 5.3: In-pixel representation of the sequential 4-stage pipeline in PMD-based *smart pixels*.

Provided that offset variations can be elliminated, we can observe from Eq. (5.6) that the following relationships are satisfied:

$$\begin{aligned} \varphi_0 &= -\varphi_\pi \\ \varphi_{\pi/2} &= -\varphi_{3\pi/2}. \end{aligned} \tag{5.8}$$

This temporal redundancy between $\varphi_0 - \varphi_\pi$ and $\varphi_{\pi/2} - \varphi_{3\pi/2}$ is used in the *4 phase-shift* algorithm to cancel the effect of FPN. Usually the value of φ_τ is accessible from the sensor instead of U_τ^A and U_τ^B.

Depth d and amplitude a are then obtained irrespective of FPN gain and offset variations:

$$d = \frac{c}{2 \cdot \omega} \cdot \arctan\left(\frac{\varphi_{3\pi/2} - \varphi_{\pi/2}}{\varphi_0 - \varphi_\pi}\right) \tag{5.9a}$$

$$a = \frac{\sqrt{\left(\varphi_{3\pi/2} - \varphi_{\pi/2}\right)^2 + (\varphi_0 - \varphi_\pi)^2}}{2}. \tag{5.9b}$$

State of the art methods propose calibration techniques of the terms G^A, G^B, O^A, O^B which can reduce the number of phase images (φ_τ) from the initial 4 samples to 2 or even 1 sample (see [79] and [80], respectively). Calibration of FPN usually requires special lighting conditions (see [79]) or patterns. We show next that only calibration of O^{cal} is needed to get depth from only two phase images. We present a calibration method without patterns or special lighting conditions, just by taking a short sequence of images from a static scene in complete darkness.

5.3.1 Calibration Method

Based on Hussmann et al. [79], if the offset O^{cal} is known, the depth and amplitude values can be obtained with only two samples of φ_τ:

$$d = \frac{c}{2 \cdot \omega} \cdot \arctan\left(\frac{\varphi_{\tau_2} - O^{cal}}{\varphi_{\tau_1} - O^{cal}}\right) \tag{5.10a}$$

$$a = \sqrt{(\varphi_{\tau_1} - O^{cal})^2 + (\varphi_{\tau_2} - O^{cal})^2}, \tag{5.10b}$$

where $\tau_2 = \tau_1 + {}^{\pi}/_{2}$ with $\tau_1 \in \{0, \pi\}$ (note that in Eq. (5.10a) the sign of $\varphi_{\tau_{\{1,2\}}}$ changes accordingly to the sign of the corresponding correlation sample). Equation (5.10a) shows that we can use only two consecutive steps (φ_{τ_1} and $\varphi_{\tau_1 + \frac{\pi}{2}}$) of the *4 phase-shift* pipeline to get depth.

We propose a calibration method that requires a number N_F of views from a static scene (i.e. camera is still with respect to the scene and objects in the scene are not moving). For every frame in the sequence we get 4 raw channel measurements of a single pixel:

$$\left\{\varphi_0^i,\ \ \varphi_{\pi/2}^i,\ \ \varphi_\pi^i,\ \ \varphi_{3\pi/2}^i\right\}_{i=1}^{N_F}. \tag{5.11}$$

We sample O^{cal} from every two channels separated by π:

$$O_i^{cal} = \frac{1}{2}\cdot\left(\varphi_\tau^i + \varphi_{\tau+\pi}^i\right). \tag{5.12}$$

Based on Mufti et al. [81] and the recent work of Hussmann et al. [82], we can assume that, if enough photons are collected, O^{cal} follows a Gaussian distribution $\mathcal{N}\left(\hat{O}^{cal}, \sigma^O\right)$, that we estimate from the samples obtained with Eq. (5.12).

We show in Fig. 5.4 the computed normal distribution of O^{cal} for two random pixels in an image taken by a commercial ToF camera.

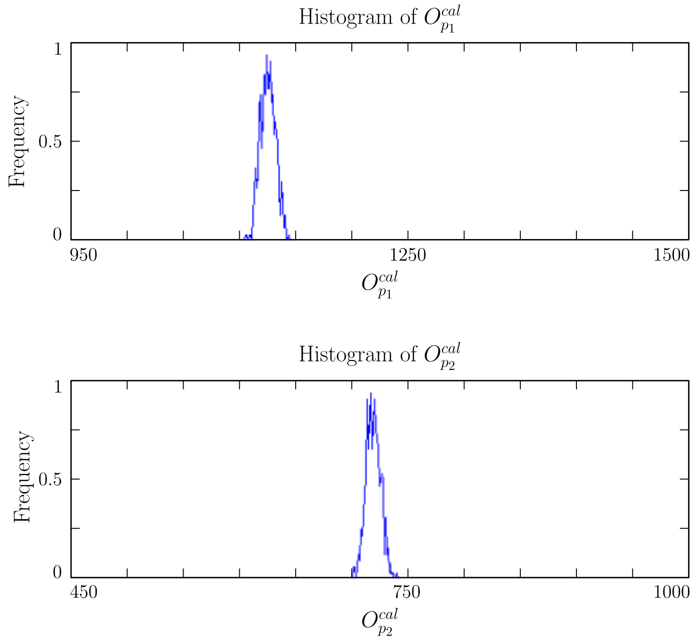


Figure 5.4: Normal distribution of O of two isolated pixels $\{p_1,\ p_2\}$ using 100 samples.

From Eq. (5.7b), O^{cal} is composed of two terms. We assume in practice that $(G^A - G^B) \approx 0$, being the second term $(O^A - O^B)$ the leading factor in O^{cal}. Unlike the first term, that depends on O and K_{int}, the offset term $(O^A - O^B)$ does not vary with depth, integration time, background illumination or light source offset. Calibration of O^{cal} can thus be performed offline and stored. We validate this assumption experimentally in Section 5.7.

5.4 Motion Artifacts

When the scene moves with respect to the camera, each stage of the *4 phase-shift* pipeline captures different depths, producing errors when they are combined to get depth. We exploit the fact that in PMD cameras the 4 stages of the pipeline are run sequentially to detect motion inconsistency at every pixel. As each stage is run at a fraction of T_{int}, we consider in the study that the motion inside each stage is very small.

In Fig. 5.5 we represent the sampled phase images in a pixel that belongs to the left occluding contour of the object. On that pixel we show that while φ_0 and $\varphi_{\pi/2}$ are properly obtained from the foreground, at φ_π there is a sudden change of depth from foreground to background. In last stage ($\varphi_{3\pi/2}$), only background depth is detected.

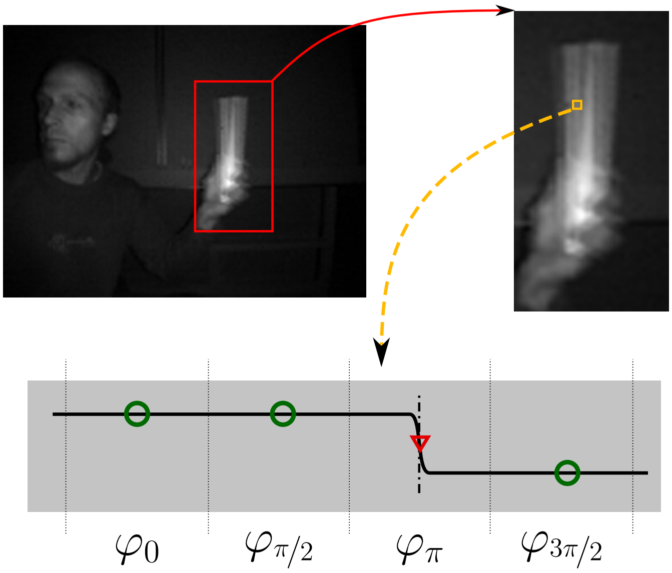


Figure 5.5: Example of the motion inconsistency for a particular pixel.

If we use Eqs. (5.9a) and (5.9b), the resulting depth appears distorted due to the inconsistency of the different samples (distorted image in Fig. 5.1). We propose to detect motion artifacts in a single pixel with the following criterion:

A pixel is said to be affected by motion artifacts if:

$$|(\varphi_0 + \varphi_\pi) - (\varphi_{\pi/2} + \varphi_{3\pi/2})| > \gamma, \tag{5.13}$$

where γ is a fixed threshold obtained by the following calibration process.

5.4.1 Calibration of the Inconsistency Threshold

To establish a threshold γ we model statistically the distribution of:

$$\eta = (\varphi_0 + \varphi_\pi) - (\varphi_{\pi/2} + \varphi_{3\pi/2}) \tag{5.14}$$

in the case of images where the scene is static with respect to the camera. We assume that η follows a Gaussian distribution $\mathcal{N}(\bar{\eta}, \sigma_\eta)$ that can be estimated by sampling. We then propose $\gamma = 3.326 \cdot \sigma_\eta$, that contains the 99.97% of the distribution when no motion artifacts are present. In Section 5.7.1 we show the estimation of the inconsistency threshold for several values of T_{int} in a commercial PMD camera.

5.5 Correction of Motion Artifacts

We propose a method that removes motion artifacts from a single frame (i.e. 4 phase images). Our method is based on two basic assumptions: *i*) after calibration of the offset, depth can reliably be obtained from two consecutive phase images and *ii*) neighbouring pixels are a strong cue to complete information that is lost due to motion artifacts.

We can distinguish 3 main steps in our proposal:

1. *Calibration:* is the responsible for performing the offset calibration (see Section 5.3.1) and determining the inconsistency threshold (see Section 5.4.1) automatically with no user interaction. After these two steps we can detect pixels affected with motion artifacts and we can obtain depth using the pseudo 4 phase-shift algorithm.

2. *Event Detection:* in this step we characterise the event in terms of: *i*) the phase image where the event appears and *ii*) the direction of the event (rising or falling edge). Hence, we have a discrete label for each pixel affected by motion that includes both temporal and depth change information.

3. *Depth Correction:* in this stage we correct the depth value of every pixel affected by motion artifacts. We combine the information available in the phase images with neighbour pixels.

5.5.1 Calibration

As explained in Section 5.3 we remove the influence of Fixed Pattern Noise in order to compute depth measurements using only two of the four phase images provided by the sensor (the so called *pseudo 4 phase-shift* algorithm explained in [79]). In addition, we apply a statistical method to find threshold γ, explained in Section 5.4.1. Both processes are done only once at the beginning. Calibration requires to show the camera a static scene during several seconds. During calibration we avoid regions of the image with strong depth discontinuities, saturation and low reflective surfaces.

5.5.2 Event Detection

When a pixel is affected by motion it receives light from multiple depths during T_{int}. In occluding contours, motion produces abrupt depth transitions, mainly affecting a single phase of the pixel. We detect events by monitoring sudden changes between the 4 phase images labelling them accordingly.

We use a discrete labelling *cel* (coarse event location) that represents the phase image containing the event and the direction (i.e. rising edge or falling edge) of the depth change caused by the motion. We use the following set of 9 labels:

$$cel \in \{-4, -3, -2, -1, 0, +1, +2, +3, +4\}, \tag{5.15}$$

where $|cel|$ represents the phase image affected by a depth transition and its sign characterises the depth gradient (e.g. $cel = +1$ means a rising edge event in φ_0 and $cel = -2$ means a falling edge event in $\varphi_{\pi/2}$). The absence of events is represented by $cel = 0$.

Event labelling is divided into two steps:

1. *Detection of the event location*: in this step we localise the phase image that is distorted due to motion, i.e. $|cel|$.

2. *Detection of the event direction*: we characterise depth transitions with two classes: $\{$*rising, falling*$\}$ events, i.e. sign of *cel*.

5.5.2.1 Detection of the Event Location

Our method for detecting the phase affected by a motion event is based on the following two main assumptions:

- An event is generated by the transition of two depths ($\{\beta_1, \beta_2\}$).

- There is only one depth transition affecting a single phase image during integration time.

From Eq. (5.13), if there is no event in the current pixel ($|cel| = 0$) the following inequalities hold: *i)* $|\varphi_0 + \varphi_\pi| < \gamma$ and *ii)* $|\varphi_{\pi/2} + \varphi_{3\pi/2}| < \gamma$, where γ is a small threshold that has been previously calibrated. Otherwise, we detect the position of the events by checking the following tests in sequential order:

$$\begin{array}{lcclcl} \text{a)} & |cel| = 1 & \Leftrightarrow & |\varphi_0 + \varphi_\pi| > \gamma & \text{AND} & |\varphi_{\pi/2} + \varphi_{3\pi/2}| < \gamma \\ \text{b)} & |cel| = 4 & \Leftrightarrow & |\varphi_0 + \varphi_\pi| < \gamma & \text{AND} & |\varphi_{\pi/2} + \varphi_{3\pi/2}| > \gamma \\ \text{c)} & |cel| = 2 \;\; \textit{or} \;\; |cel| = 3 & \Leftrightarrow & |\varphi_0 + \varphi_\pi| > \gamma & \text{AND} & |\varphi_{\pi/2} + \varphi_{3\pi/2}| > \gamma \end{array} \tag{5.16}$$

To distinguish between events in $|cel| = 2$ or $|cel| = 3$, we first write the equations of all phases from Eq. (5.6), once we have calibrated the offset term O^{cal} (see Section 5.3.1):

$$\begin{aligned} \varphi_0 &= +K_{int} \cdot a' \cdot \cos\beta \\ \varphi_{\pi/2} &= -K_{int} \cdot a' \cdot \sin\beta \\ \varphi_\pi &= -K_{int} \cdot a' \cdot \cos\beta \\ \varphi_{3\pi/2} &= +K_{int} \cdot a' \cdot \sin\beta \end{aligned} \tag{5.17}$$

where $a' = G^{\text{FPN}} \cdot \dfrac{2 \cdot a_0}{\pi}$.

If an event appears in $|cel| = 2$, $\varphi_{\pi/2}$ contains the mixture of two phase shifts $\{\beta_1, \beta_2\}$. Hence, we can express Eqs. (5.17) considering the phase mixture as:

$$\begin{aligned}
\varphi_0 &= +K_{int} \cdot a'_1 \cdot \cos\beta_1 \\
\varphi_{\pi/2} &= -K_1 \cdot a'_1 \cdot \sin\beta_1 - K_2 \cdot a'_2 \cdot \sin\beta_2 \\
\varphi_\pi &= -K_{int} \cdot a'_2 \cdot \cos\beta_2 \\
\varphi_{3\pi/2} &= +K_{int} \cdot a'_2 \cdot \sin\beta_2
\end{aligned} \tag{5.18}$$

where a'_1, a'_2 are the corresponding amplitude values of the two depths involved in the generation of the artifact and $K_{int} = K_1 + K_2$.

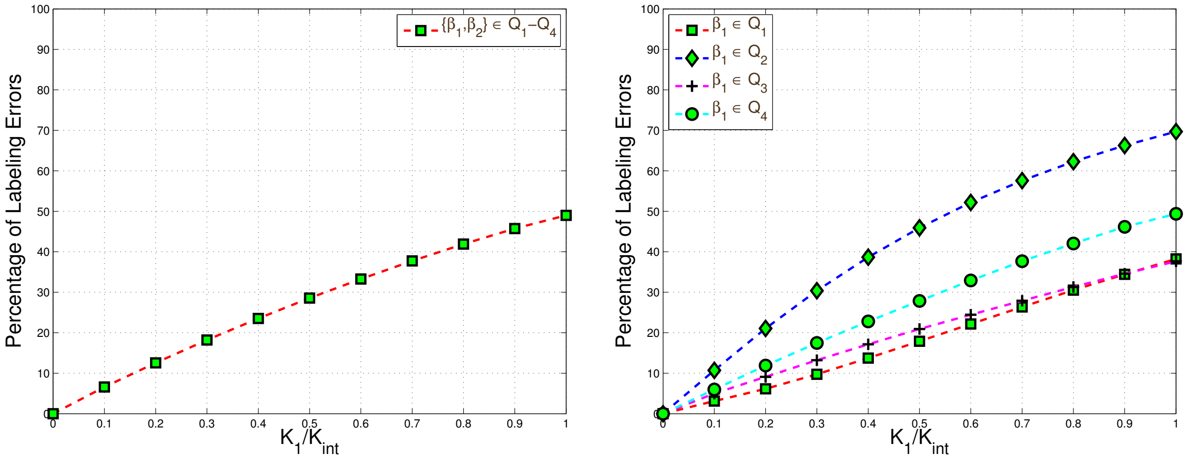
We use Eq. (5.18) to express $\varphi_0 + \varphi_\pi$ and $\varphi_{\pi/2} + \varphi_{3\pi/2}$ as follows:

$$\begin{aligned}
\varphi_0 + \varphi_\pi &= K_{int} \cdot (a'_1 \cdot \cos\beta_1 - a'_2 \cdot \cos\beta_2) \\
\varphi_{\pi/2} + \varphi_{3\pi/2} &= K_1 \cdot (a'_2 \cdot \sin\beta_2 - a'_1 \cdot \sin\beta_1).
\end{aligned} \tag{5.19}$$

We propose to detect events in the second interval ($|cel| = 2$) by checking the following condition:

$$|\varphi_0 + \varphi_\pi| > |\varphi_{\pi/2} + \varphi_{3\pi/2}|. \tag{5.20}$$

Finally, the identification of an event in φ_π ($|cel| = 3$) is defined as the event that does not satisfy the statements of $|cel| = \{1, 4, 2\}$, in that order. As a consequence of following the sequential pipeline described in Eqs. (5.16) to detect the event location, the labelling error is determined by the detection of $|cel| = 2$. To test when inequality (5.20) holds, in Fig. 5.6a we plot the number of incorrectly labelled events depending on the value of K_1 for every possible combination of $\{\beta_1, \beta_2\}$ (sweeping both terms over the $(0,\ 2\pi)$ interval) in the depth range 0-5 m. In Fig. 5.6b we fix β_1 to be in a specific depth quadrant Q_i (see Eq. (5.21)), sweeping β_2 over all possible values. To create Fig. 5.6 we use Eq. (5.18), assuming that amplitudes a'_1 and a'_2 are inversely proportional to the square of the corresponding depth.



(a) $\forall\ \{\beta_1, \beta_2\}$ combination

(b) $\beta_1 \in Q_i,\ i = 1, 2, 3, 4, \quad \beta_2 \in (0, 2\pi)$

Figure 5.6: Labelling error percentage for several combination of $\{\beta_1, \beta_2\}$ and all the values of K_1.

We observe in Fig. 5.6 that the percentage of erroneous event classification increases with the factor K_1. However, when $K_1 \approx K_{int}$, the event is closer to be happening in the transition between $|cel| = 2$ and $|cel| = 3$. Those mistakes are thus not very relevant as there is no practical difference between considering the event to be classified as $|cel| = 2$ or $|cel| = 3$. As shown in Fig. 5.6, if we only consider those classification errors committed when ${}^{K_1}/_{K_{int}} < 0.66$, the detection of events in $|cel| = 2$ remains in average below 35% of error. This corresponds with the results we obtain in real sequences (see Section 5.7), where despite these classification errors, the event detection allows us to effectively detect and compensate motion artifacts. The complete event detection pipeline is resumed in Algorithm 2 (A.1).

5.5.2.2 Detection of the Event Direction

Given that we know the phase image affected by a motion artifact, we identify the direction of the event as follows:

- *Falling Edge*: when there's a transition from foreground to background ($\boldsymbol{\beta_2} > \boldsymbol{\beta_1}$).
- *Rising Edge*: when there's a transition from background to foreground ($\boldsymbol{\beta_2} < \boldsymbol{\beta_1}$).

so that we can make a proper substitution (using local neighbourhood) of the erroneous measurements and correct the artifacts.

We divide the maximum depth range of the camera into its corresponding depth quadrants Q_i s.t.

$$Q_i = i \;\Leftrightarrow\; \beta \in \left\{(i-1)\cdot\frac{\pi}{2}, i\cdot\frac{\pi}{2}\right\}, \quad i \in \{1,2,3,4\}. \tag{5.21}$$

Then, for each pixel affected by motion we can easily obtain the involved depth quadrants (Q^{bg}, Q^{fg}) by clustering the depths of each pixel in a local neighbourhood (we use the 2 depths per pixel using the pseudo four-phase shift algorithm).

The direction of an event is detected by simply checking the sign of the phase image difference shifted by π, according to the involved phase quadrants. The whole labelling method is detailed in A.1 and A.2. In Fig. 5.7 we show the detection of the event direction in several cases where multiple depth quadrants are involved. In the experiments of Fig. 5.7 the background depth varies along the whole range while the foreground varies from motion in Q_1 (first two columns) to motion in Q_2 (last two columns). Note that, due to the weak illumination power, uncertainty is high for depths above 1.5 m approximately (Q_2 and higher), generating undesirable effects.

Finally, in Fig. 5.8 we show the events detected in the example shown in Fig. 5.1, using a colour palette that represents the value of *cel*. We codify every pixel of the object with a colour (i.e. from dark blue to light pink) that represents the stage $\{\varphi_0, \varphi_{\pi/2}, \varphi_\pi, \varphi_{3\pi/2}\}$ in the *4 phase shift* pipeline which is affected by the transition from foreground to background (i.e. blue colours) and from background to foreground (i.e. red colours). Fig. 5.9 illustrates the detection of a rising edge event in $\varphi_{\pi/2}$. We define $\{\varphi^{bg}, \varphi^{fg}\}$ as the phase images associated with background (low charge) and foreground (high charge) depths, respectively.

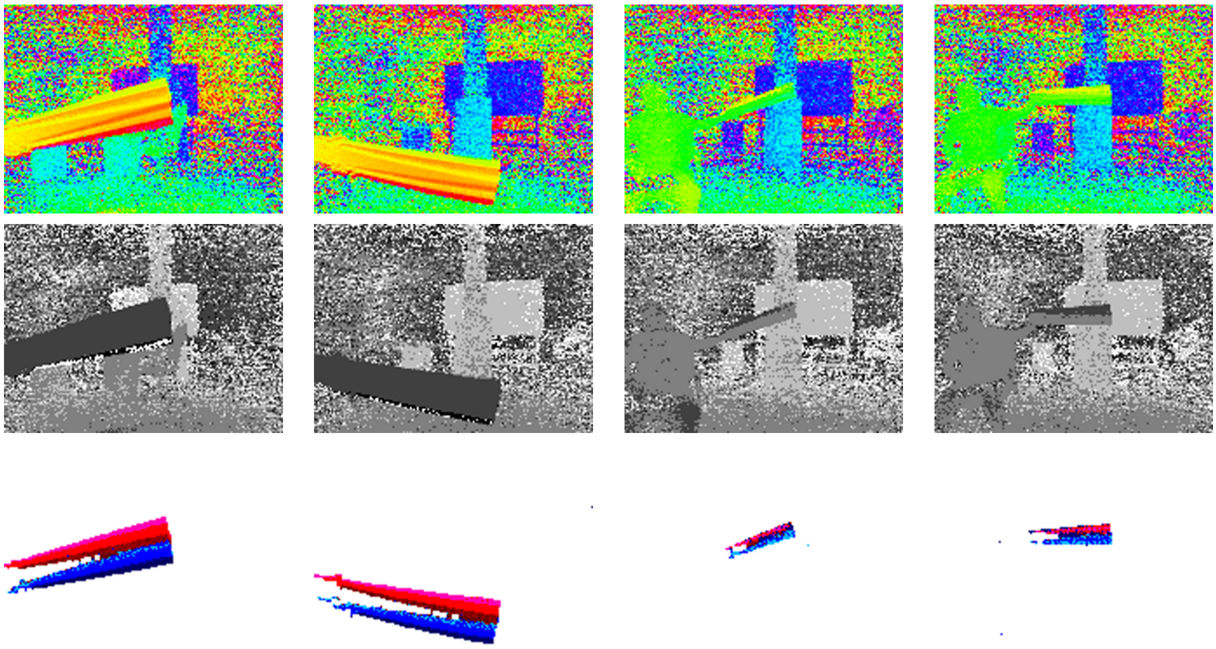


Figure 5.7: Detection of the event direction in a multiple-quadrant scenario. The first row shows depth measurements captured by the ToF camera. The second row shows the corresponding depth quadrants in gray-scale (brightest colours denotes higher depth quadrants) and the last row shows the proposed coarse event labelling (i.e. *cel*).

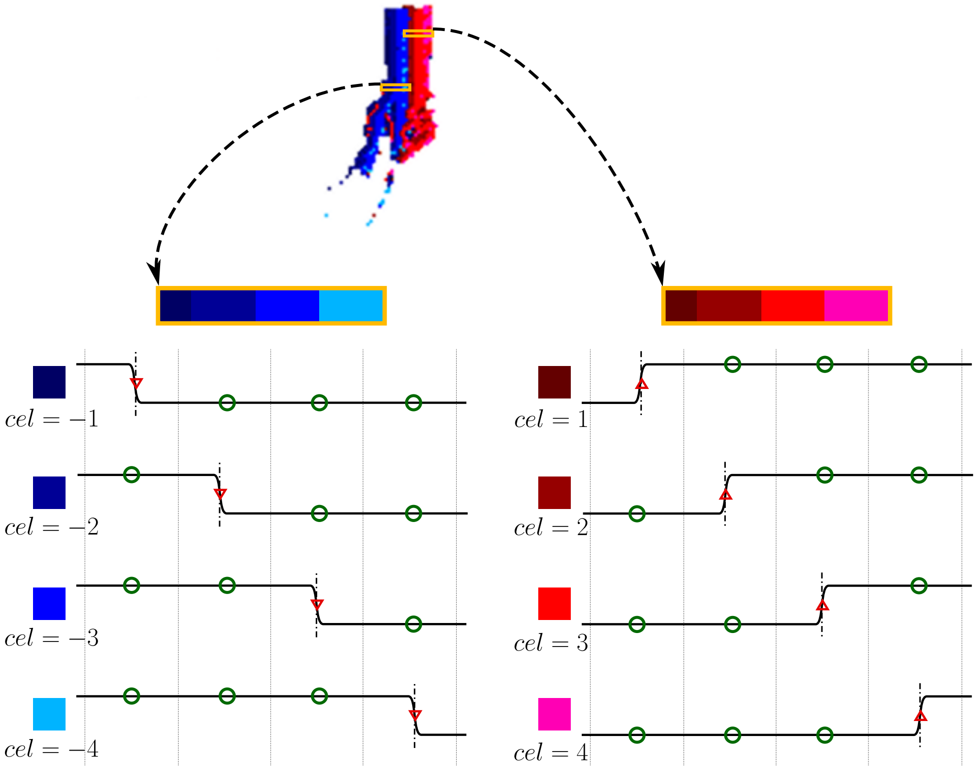


Figure 5.8: Coarse event location in a real scene.

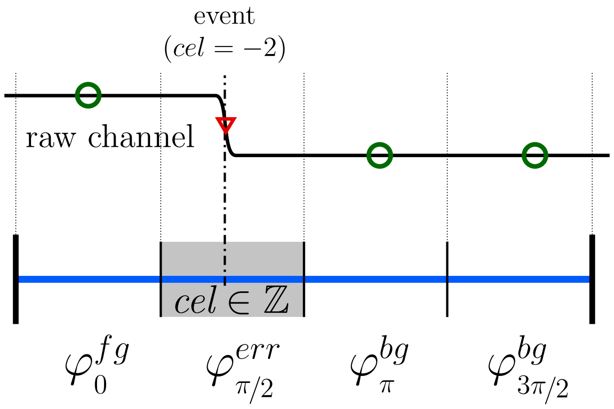


Figure 5.9: Characterisation of a pixel affected by motion: falling edge event in the second phase image ($cel = -2$).

5.5.3 Depth Correction

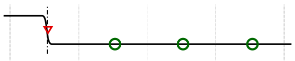
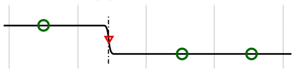
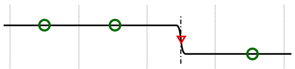
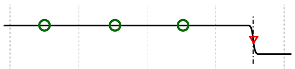
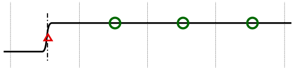
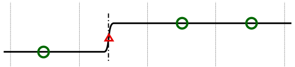
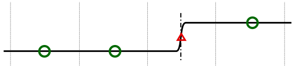
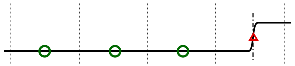
Our depth correction algorithm is based on two basic rules: *i*) we reconstruct the depth detected at the beginning of the integration time and *ii*) we reconstruct depth using two previous consecutive phases to the phase affected by the motion event. If less than two consecutive phase images are available we search in neighbouring pixels to complete data.

According to rule *i*), we obtain local foreground depth on falling edge events (from foreground to background) and the local background depth on rising edge events (from background to foreground).

From rule *ii*), events in φ_π or $\varphi_{3\pi/2}$ allow us to recover depth directly from the phases φ_0 or $\varphi_{\pi/2}$ using Eq. (5.10a). If the event is detected in any of the first two phases, we search in the local neighbourhood to recover depth. We show in Table 5.1 the case studies depending the position of the detected event. We also point out the required phase images needed to recover the corresponding depth and the *cel* labelling we look for in the neighbours to get a replacement. As depicted from Table 5.1, we need to search for neighbours in half of the cases.

The strategy to find neighbours involves two methodologies: *i*) to look for the closest pixels that have the same event type (rising or falling edge) but complementary stable phase images and *ii*) to look for the nearest points that have no motion artifact and compatible stable phase images. In Fig. 5.10 we show an example of our proposal to replace missing data with spatial information of a pixel affected by motion characterised with a falling edge in the second phase image ($\varphi_{\pi/2}$).

Once we have found the most suitable neighbours, we sort them, identifying the median phase value. It's well known in the literature that the median filter is a robust and effective method for outlier removal. In a compromise between robustness and computational efficiency, we replace the corresponding phase images with the mean of the 3 closest candidates around the median. ToF measurements are then recovered using only 2 phase images. In the example shown in Fig. 5.10, where only one phase image is needed for substitution, the replacement would be computed in the following 2-step procedure:

Detected Event	Objective Depth	Required Substitution	Spatial Search Substitution
$cel = -1$ 	*foreground*	φ_0^{fg} *and* $\varphi_{\pi/2}^{fg}$	$cel =$ -3 *or* -4 *or* $0^{(*)}$
$cel = -2$ 	*foreground*	$\varphi_{\pi/2}^{fg}$	$cel =$ -3 *or* -4 *or* $0^{(*)}$
$cel = -3$ 	*foreground*	*(not required)*	*(not required)*
$cel = -4$ 	*foreground*	*(not required)*	*(not required)*
$cel = 1$ 	*background*	φ_0^{bg} *and* $\varphi_{\pi/2}^{bg}$	$cel = 3$ *or* 4 *or* $0^{(*)}$
$cel = 2$ 	*background*	$\varphi_{\pi/2}^{bg}$	$cel = 3$ *or* 4 *or* $0^{(*)}$
$cel = 3$ 	*background*	*(not required)*	*(not required)*
$cel = 4$ 	*background*	*(not required)*	*(not required)*

(*) *No motion artifact is detected but it has compatible stable phase images.*

Table 5.1: Case studies for the event location in depth correction.

1. Find the most suitable neighbours within the neighbour set $\mathcal{R}_i$ as the closest points to the pixel of interest with stable phase images in the required intervals. We search neighbours in a spiral loop until enough pixels are found (i.e. a minimum of $N_n = 7$ is required to increase accuracy) or we exceed a reasonable distance (we fix this internal parameter with a radius distance of 20 pixels). In practice a 20 pixel radius distance can be in fact very conservative. We found that a radius distance between {2.5 and 4.9} pixel is enough in our camera after averaging the radius needed over several frames in a wide variety of dynamic scenarios.

2. Sort the previous data set and identify the median value. Finally, we compute the mean of the 3 closest points to the median:

$$\varphi_{(i,\pi/2)}^{fg} = \frac{1}{3} \sum_{j=\{med_i-1\}}^{med_i+1} \hat{\varphi}_{(j,\pi/2)}^{fg}, \tag{5.22}$$

where $\varphi_{(i,\tau)}^{z}$ refers to the phase image φ_τ at pixel i, identified as background or foreground (i.e. $z = \{bg, fg\}$) and med_i is the median value index for the neighbours data set of pixel i ordered in the previous step ($\hat{\varphi}_{(j,\pi/2)}^{fg}$).

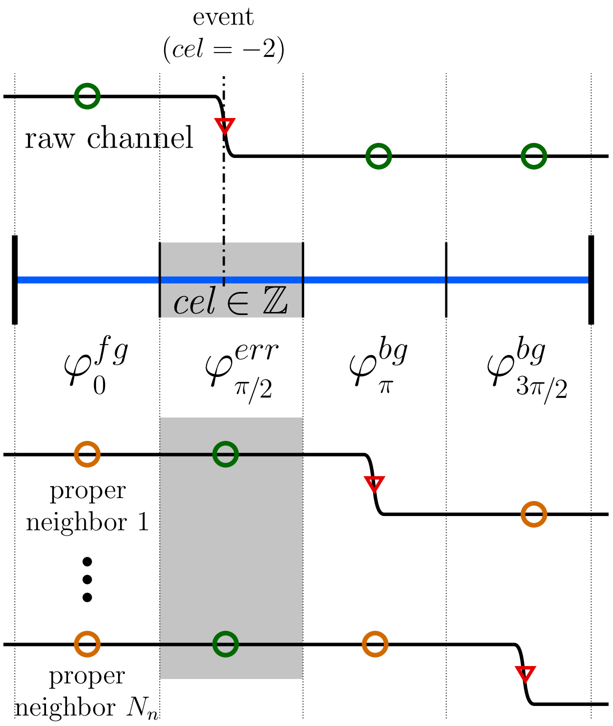


Figure 5.10: In-pixel substitution of a phase image affected by motion ($cel = -2$). We found N_n proper neighbours with reliable data in the affected sample ($\varphi_{\pi/2}^{err}$).

Additional post-processing of depth and amplitude measurements (like outliers removal, bilateral filtering...), as it is done in [53] and [55], could improve the results.

5.6 Motion Estimation

We show that, as a consequence of our depth correction algorithm, we can actually measure motion flow near occluding contours of the scene using a single ToF frame. We first define a method to accurately obtain the instant where a motion event is detected on each pixel, that we call *pel* (precise event location). We then use the gradient of *pel* image to obtain motion flow.

5.6.1 Precise Event Location

Let assume that we find an event in phase τ of current pixel using the proposed coarse event detection. We define φ_τ^{err} as the erroneous value of the phase image in phase τ. We model the event as a change between two depth values (e.g. background and foreground). We define φ_τ^{fg} and φ_τ^{bg} as the foreground and background phase images corresponding to τ. We assume that photon generation in the CMOS photo gates is proportional to exposure time and we define that φ_τ^{err} as a linear combination of φ_τ^{fg} and φ_τ^{bg}. Depending the direction of the event we define:

$$\varphi_\tau^{err} = \begin{cases} \alpha \cdot \varphi_\tau^{fg} + (1-\alpha) \cdot \varphi_\tau^{bg} & , \textit{if} \quad cel < 0 \\ \\ \alpha \cdot \varphi_\tau^{bg} + (1-\alpha) \cdot \varphi_\tau^{fg} & , \textit{if} \quad cel > 0 \end{cases} , \tag{5.23}$$

where α is the fraction of the integration time in phase τ before the event appears (see Fig. 5.11). Using the algorithm from Section 5.5.3, we can obtain φ_τ^{fg} and φ_τ^{bg} from neighbour pixels and from phase images not affected by the event in the pixel. In Table 5.2 we show all possible cases to get φ_τ^{fg} and φ_τ^{bg}.

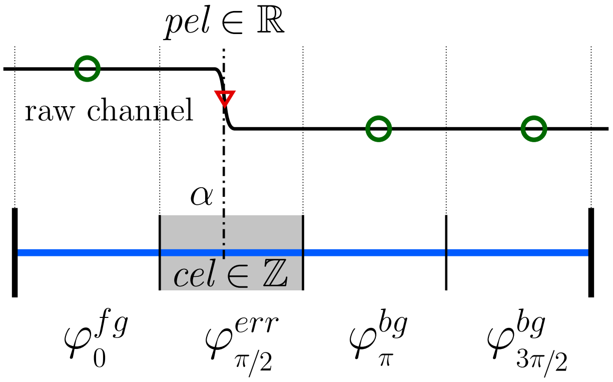


Figure 5.11: Example of the extrapolation of the event location ($cel = -2$).

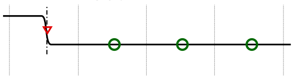
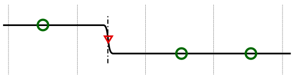
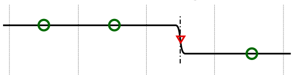
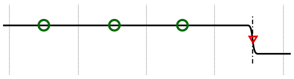
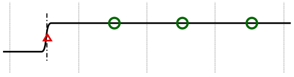
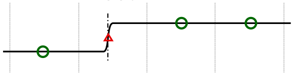
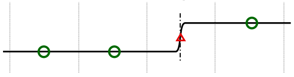
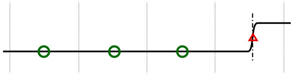
Event	φ_τ^{fg}	φ_τ^{bg}	α
$cel = -1$ 	$\varphi_0^{fg} \rightarrow$ neighbours	$\varphi_0^{bg} = -\varphi_\pi^{bg}$	$\alpha = \frac{\varphi_0^{err} + \varphi_\pi^{bg}}{\varphi_0^{fg} + \varphi_\pi^{bg}}$
$cel = -2$ 	$\varphi_{\pi/2}^{fg} \rightarrow$ neighbours	$\varphi_{\pi/2}^{bg} = -\varphi_{3\pi/2}^{bg}$	$\alpha = \frac{\varphi_{\pi/2}^{err} + \varphi_{3\pi/2}^{bg}}{\varphi_{\pi/2}^{fg} + \varphi_{3\pi/2}^{bg}}$
$cel = -3$ 	$\varphi_\pi^{fg} = -\varphi_0^{fg}$	$\varphi_\pi^{bg} \rightarrow$ neighbours	$\alpha = \frac{\varphi_\pi^{err} - \varphi_\pi^{bg}}{-\varphi_0^{fg} - \varphi_\pi^{bg}}$
$cel = -4$ 	$\varphi_{3\pi/2}^{fg} = -\varphi_{\pi/2}^{fg}$	$\varphi_{3\pi/2}^{bg} \rightarrow$ neighbours	$\alpha = \frac{\varphi_\pi^{err} - \varphi_{3\pi/2}^{bg}}{-\varphi_{\pi/2}^{fg} - \varphi_{3\pi/2}^{bg}}$
$cel = 1$ 	$\varphi_0^{fg} = -\varphi_\pi^{fg}$	$\varphi_0^{bg} \rightarrow$ neighbours	$\alpha = \frac{\varphi_0^{err} + \varphi_\pi^{fg}}{\varphi_0^{bg} + \varphi_\pi^{fg}}$
$cel = 2$ 	$\varphi_{\pi/2}^{fg} = -\varphi_{3\pi/2}^{fg}$	$\varphi_{\pi/2}^{bg} \rightarrow$ neighbours	$\alpha = \frac{\varphi_{\pi/2}^{err} + \varphi_{3\pi/2}^{fg}}{\varphi_{\pi/2}^{bg} - \varphi_{3\pi/2}^{fg}}$
$cel = 3$ 	$\varphi_\pi^{fg} \rightarrow$ neighbours	$\varphi_\pi^{bg} = -\varphi_0^{bg}$	$\alpha = \frac{\varphi_\pi^{err} - \varphi_\pi^{fg}}{-\varphi_0^{bg} - \varphi_\pi^{fg}}$
$cel = 4$ 	$\varphi_{3\pi/2}^{fg} \rightarrow$ neighbours	$\varphi_{3\pi/2}^{bg} = -\varphi_{\pi/2}^{bg}$	$\alpha = \frac{\varphi_{3\pi/2}^{err} - \varphi_{3\pi/2}^{fg}}{-\varphi_{\pi/2}^{bg} - \varphi_{3\pi/2}^{fg}}$

Table 5.2: All possible cases for obtaining α

Using Eq. (5.23) we obtain α:

$$\alpha = \begin{cases} \dfrac{\varphi_\tau^{err} - \varphi_\tau^{bg}}{\varphi_\tau^{fg} - \varphi_\tau^{bg}} & , \textit{if} \quad cel < 0 \\ \\ \dfrac{\varphi_\tau^{err} - \varphi_\tau^{fg}}{\varphi_\tau^{bg} - \varphi_\tau^{fg}} & , \textit{if} \quad cel > 0 \end{cases} . \tag{5.24}$$

We define *pel* as the fraction of T_{int} where the motion event takes place:

$$pel = \frac{(|cel| - 1) + \alpha}{4}, \tag{5.25}$$

where we assume that the integration time of each phase image is $T_{int}/4$.

We show in Fig. 5.12 *pel* computation for the running example used in Fig. 5.1.

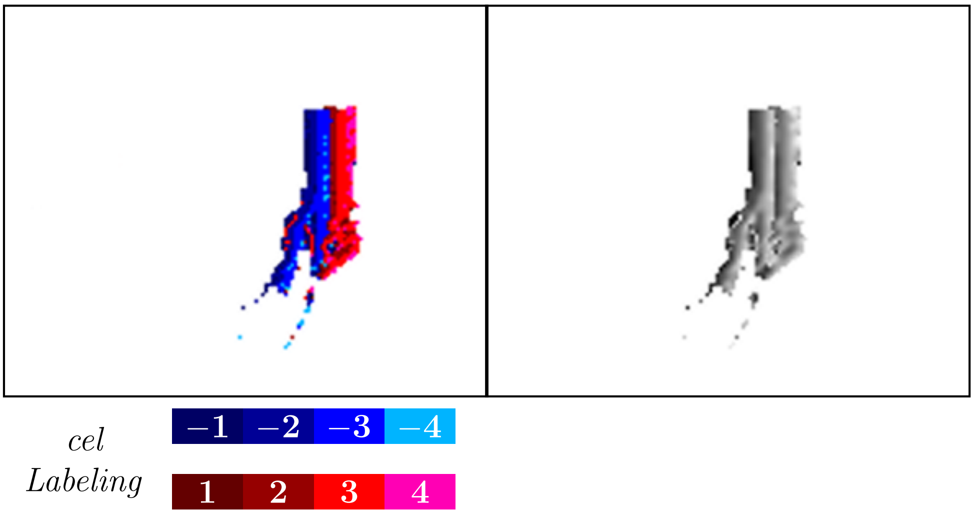


Figure 5.12: From discrete to extrapolated precise event location.

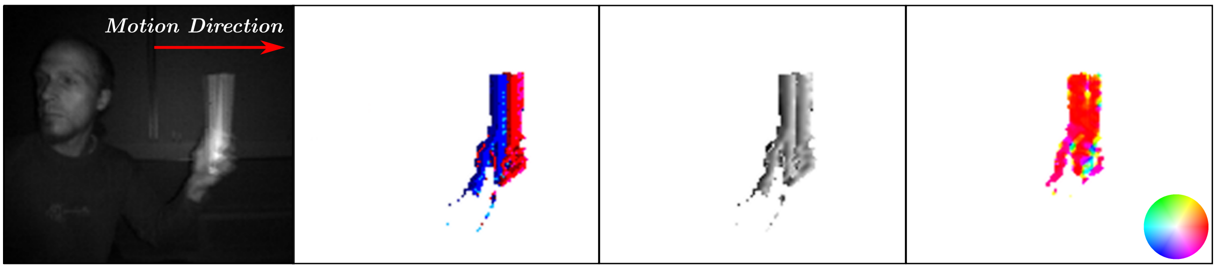
5.6.2 Motion Flow Estimation

During motion, neighbouring pixels near occluding contours are affected by the motion event at different *pel* values. The spatial (i.e. image coordinates) distribution of *pel* values depends on magnitude and direction of motion. We denote as $\mathbf{I}_{pel}$ the image arrangement of *pel* values (see right part of Fig. 5.12). The image gradient of $\nabla\mathbf{I}_{pel}$ gives the distribution of time differences between pixels. The magnitude of $\nabla\mathbf{I}_{pel}$ is proportional to the inverse of motion flow:

$$\|\nabla\mathbf{I}_{pel}\| \propto \frac{1}{v}, \tag{5.26}$$

where v is the speed magnitude. The direction of $\nabla\mathbf{I}_{pel}$ gives the direction of speed in the image. Fig. 5.13 shows $\mathbf{I}_{pel}$ and its gradient for an object following horizontal motion in the image.

Equation (5.26) is only valid when either we detect an event in the pixel ($pel \neq 0$) or the speed of the object produces events in neighbouring pixels that are separated by less than the integration time.



Blurred Object *"cel - Labeling"* *"pel - Computation"* *"pel - Gradient" and Motion Estimation*

Figure 5.13: Motion field estimation using the gradient of the precise event location (*pel*) image.

Axial motion is corrected properly as movements along the viewing direction also generate phase image displacement in occluding contours. However, our motion field estimator doesn't provide information about this kind of movement.

5.7 Results

Our proposal has been implemented into C++ using a commercial PC (Intel(R) Core($^{\text{TM}}$)2 Quad CPU Q9550@2.83 GHz). We must highlight that we don't require any GPU implementation to achieve real time processing. In fact, we believe that our algorithm can be easily implemented in camera's hardware as it only requires basic arithmetic operations and limited memory.

We use a low cost commercial ToF camera (PMDtec CamBoard Nano) for our experiments. This device only has 1 *Light Emitting Diode (LED)* emitter located next to the image sensor (see Fig. 5.14), specially design for short range applications.

The CamBoard Nano has some specific limitations: *i*) non-visible areas when objects are close to the ToF camera due to the baseline between the sensor and the LED emitter (see green ellipses in Fig. 5.14), *ii*) low power emission/SNR (see red ellipses in Fig. 5.14).

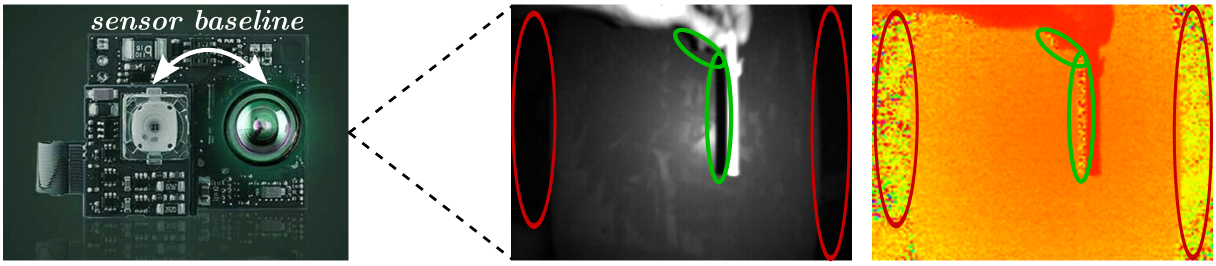


Figure 5.14: PMDtec Camboard Nano pointing at a scene composed of two foreground planes (50 cm and 60 cm, respectively) and a foreground object at 40 cm. We show the errors in amplitude (left) and depth (right) due to the sensor baseline (green ellipses) and the low power emission (red ellipses).

Our method has been validated in real scenarios providing quantitative and qualitative results. In all experiments we use the following challenging scenario: *i*) maximum integration time (2000 μs)[1], *ii*) high motion dynamics, *iii*) objects moving close to the camera and *iv*) low power emission (only one LED is emitting the signal). Depth images have been converted to a colour-map to appreciate subtle depth changes.

5.7.1 Calibration of the Inconsistency Threshold for Different Exposure Times

In Fig. 5.15 we show the values of the inconsistency threshold after applying the on-line calibration proposed in Section 5.4.1 for several values of the integration time. Probabilistic distributions are shown to validate our assumption.

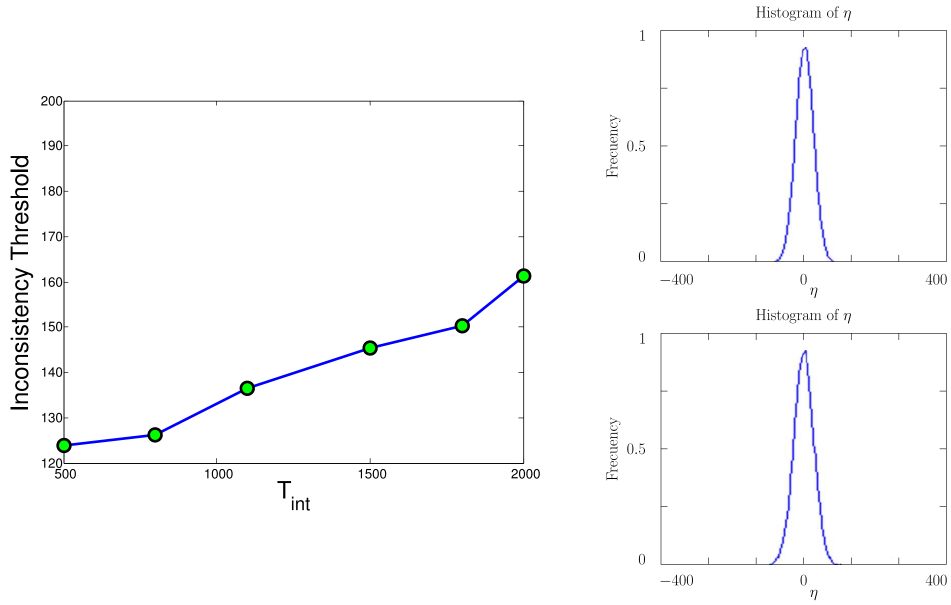


Figure 5.15: Estimation of the inconsistency threshold. On the left side we show the estimated inconsistency threshold (γ) for different integration times (T_{int}). On the right side we show the histogram of η (see Section 5.4.1) for some particular $T_{int} = \{800, 1800\}(\mu s)$.

5.7.2 Quantitative Results

To measure accuracy of our method we use a scene of known geometry moving at a controlled speed. We use a planar target at a fixed distance of the camera that is rotating using a motor. We place the target perpendicularly to the line of sight of the camera and we manually label the region of interest in the image where the target appears when no motion is applied. We compute the mean and the standard deviation of depth measurements in the labelled area. Depth mean and standard deviation ($\mu(d^{\mathrm{GT}})$ and $\sigma(d^{\mathrm{GT}})$, respectively) obtained by hand-labelling the scene are used as ground truth measurements.

In Table 5.3 we show the region we have used to compute the ground truth (notice that no motion should be applied).

[1] The frame-rate of the CamBoard considering the maximum integration time (2000 μs) is 60 fps, approximately.

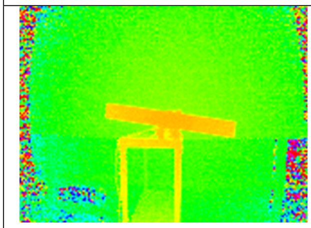
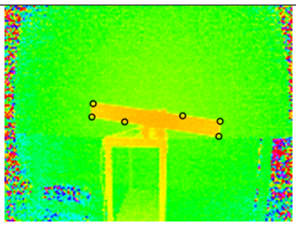
Static Scene	Labelled Scene	Area	Measured Depth (Mean / Std-Dev)
		563 pix	(47.6 *cm* / 0.035)

Table 5.3: Rotating target when no motion is applied used as ground truth.

In Table 5.4 we show quantitative results for several angular velocities of the rotating target, including depth deviation, inliers/outliers deviation and processing time. The associated depth and motion field[2] images are presented in Figs. 5.16 and 5.17. We compare our results with one of the state of the art methods proposed in [54], where multiple frames are considered.

Depth deviation is computed with respect to the mean depth in the ground truth (d^{GT}) such that:

$$\sigma(d^*) = \frac{1}{N_l} \sum_{\forall i \in \mathcal{A}_l} \left(d_i^* - \mu(d^{\mathrm{GT}}) \right) \tag{5.27}$$

where N_l is the number of pixels inside the hand labelled area $\mathcal{A}_l$, d_i is the measured depth at pixel i, d^* is the corrected depth after applying any of the correction algorithms and $\mu(d^{\mathrm{GT}})$ is the depth mean we use as ground truth.

The inlier percentage is computed with respect to the hand-labelled area ($\mathcal{A}_l$) for each scene as

$$\%\mathrm{Inliers} = \frac{\mathcal{A}^*}{\mathcal{A}_l} * 100 \tag{5.28}$$

where

$$\mathcal{A}^* = \sum_{\forall i \in \mathcal{A}_l} \Gamma_i \tag{5.29}$$

$$\Gamma_i = \begin{cases} 1, & \mathit{if} \ \ (\mu(d^{\mathrm{GT}}) - \sigma(d^{\mathrm{GT}})) < d_i^* < (\mu(d^{\mathrm{GT}}) + \sigma(d^{\mathrm{GT}})) \\ 0, & \mathit{others} \end{cases}$$

The percentage of outliers is computed such that:

$$\%\mathrm{Outliers} = \frac{\overline{\mathcal{A}^*}}{\mathcal{A}_l} * 100 \tag{5.30}$$

where

[2] We use HSV colour-map to represent motion direction (best viewed in the lower right corner of motion field images).

$$\overline{\mathcal{A}^*} = \sum_{\forall i \in d^*} \overline{\Gamma_i} \tag{5.31}$$

$$\overline{\Gamma_i} = \begin{cases} 1, & \textit{if} \quad i \in \mathcal{A}_l \quad \textit{AND} \\ & (d_i^* > \mu(d^{\mathrm{GT}}) + \sigma(d^{\mathrm{GT}})) \;\; \textit{OR} \;\; d_i^* < (\mu(d^{\mathrm{GT}}) - \sigma(d^{\mathrm{GT}})) \\ & \\ 1, & \textit{if} \quad i \notin \mathcal{A}_l \quad \textit{AND} \\ & (\mu(d^{\mathrm{GT}}) - \sigma(d^{\mathrm{GT}})) < d_i^* < (\mu(d^{\mathrm{GT}}) + \sigma(d^{\mathrm{GT}})) \\ & \\ 0, & \textit{others} \end{cases}$$

Processing time is shown in the last column.

Rotation Speed (rps)	σ - Depth Deviation (cm)	Inliers (%)	Outliers (%)	Processing Time (ms)
1.5	11.5 / **1.6**	69.2 / **92.1**	32.8 / **16.3**	**13.4**
3	11.9 / **2.4**	57.3 / **91.2**	50.3 / **18.5**	**14.7**
5.9	13.1 / **2.6**	48.3 / **87.9**	54.7 / **19.1**	**15.3**
8.8	14.3 / **3.3**	48.7 / **87.2**	62.4 / **27.2**	**17.1**
10.3	15.2 / **3.5**	45.2 / **87.1**	67.0 / **30.6**	**17.5**
13.2	18.4 / **3.0**	40.0 / **82.8**	69.2 / **34.6**	**18.4**
16.1	15.0 / **3.4**	43.2 / **82.4**	73.7 / **33.2**	**18.9**
17.6	14.9 / **3.2**	47.1 / **80.3**	75.3 / **37.2**	**18.7**
20.5	17.8 / **3.8**	37.2 / **80.0**	79.8 / **35.7**	**18.8**

Table 5.4: Quantitative results for several rotation velocities. In brackets we show the corresponding values for the proposed standard correction by Schmidt et al. [54] and our proposal; i.e. *(Schmidt2011 / Our Proposal)*. Bold numbers represent the values obtained with our proposal.

From left to right, we show in Figs. 5.16 and 5.17 the corresponding results for different angular velocities. From top to bottom, we show *i*) ToF measurements, *ii*) Schmidt et al. [54] standard depth correction, *iii*) our proposed depth correction with the hand labelled area and *iv*) our motion field estimation, respectively.

Figs. 5.16, 5.17 and Table 5.4 show that our method clearly improves with respect to Schmidt et al. [54]. The correction we achieved is remarkable, recovering most of the object boundaries, even at high rotation speed. Motion flow estimation is stable in all cases. Processing time reveals that we achieve more than 55 fps with a low-end computer.

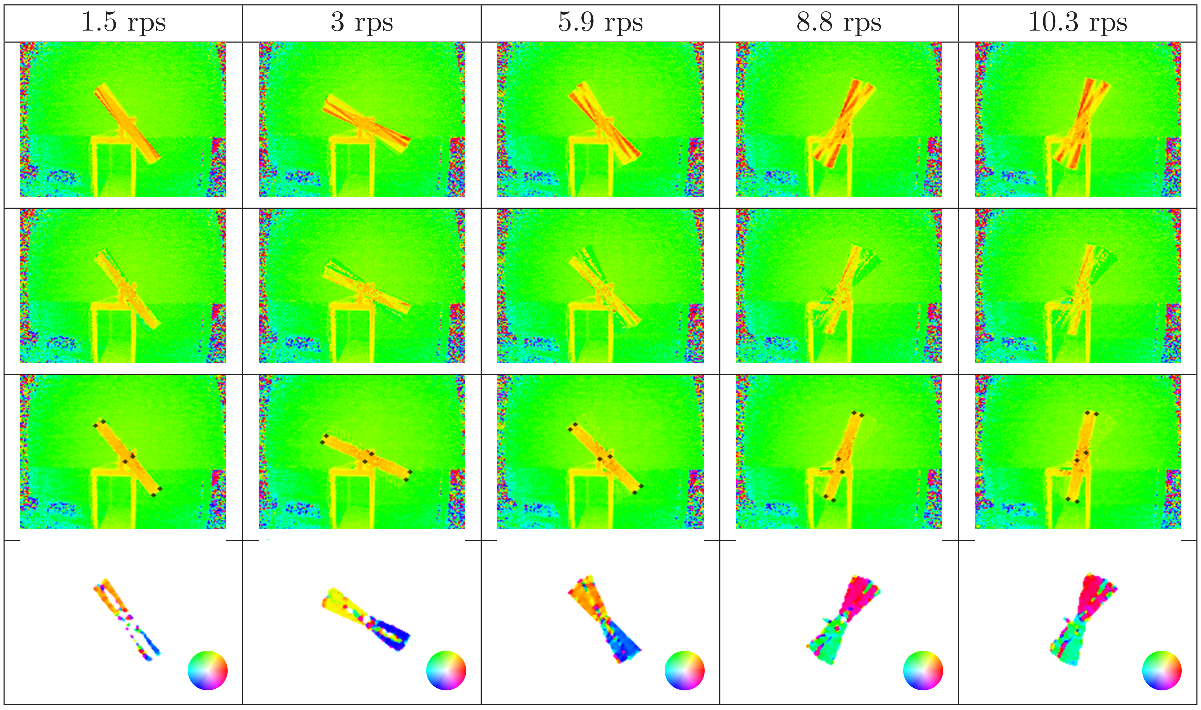


Figure 5.16: Comparative results for different rotation velocities (from 1.5 to 10.3 rps). From top to bottom, we show *i*) ToF measurements, *ii*) Schmidt et al. [54] depth correction, *iii*) our proposed depth correction with the hand labelled area and *iv*) our motion field estimation.

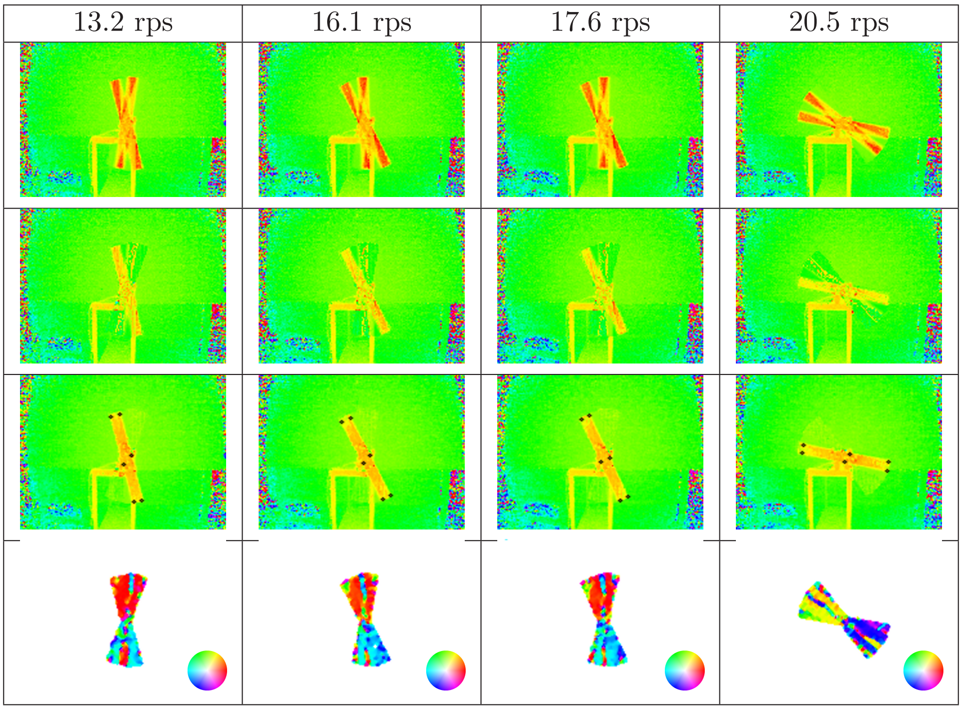


Figure 5.17: Comparative results for different rotation velocities (from 13.2 to 20.5 rps). From top to bottom, we show *i*) ToF measurements, *ii*) Schmidt et al. [54] depth correction, *iii*) our proposed depth correction with the hand labelled area and *iv*) our motion field estimation.

5.7.3 Qualitative Results

In Figs. 5.18 and 5.19 we show qualitative results for several scenes. Each column shows the results for a particular scene under the challenging conditions we defined at the beginning of the section. In rows 1 and 3 we show depth and amplitude measurements and the proposed correction in rows 2 and 4. We show in rows 5 and 6 the coarse[3] and the precise event location images. Finally, we show in the last row the estimated motion flow. We use the colour-map displayed in the lower right corner to codify motion magnitude and direction.

The first three columns of Fig. 5.18 show a stick moving very fast close to the camera in three different movements: vertical, horizontal and diagonal, respectively. The fourth column shows the behaviour of the algorithm using very thin objects. Finally, in the fifth column we consider an object moving towards the camera and rotating at the same time, showing the results when different kinds of motion are applied to the same object.

In the first two columns of Fig. 5.19 we show high motion dynamics where phase images overlap each other. In the next two experiments we show the results when motion artifacts of multiple objects occlude each other. In the last column we show the results when the camera moves.

5.8 Conclusion

This work presents a method to correct motion artifacts in ToF cameras using a single frame with very low processing time (13 ms in a low-end computer without GPU). Motion artifacts in PMD ToF cameras appear when the received optical energy change significantly during the integration time (i.e. multiple depths are considered for a single pixel) and the accumulated charges within a certain *smart pixel* reveal inconsistencies within the 4 stage pipeline. The solution we propose to correct motion artifacts is based on exploiting redundancy of the PMD *4-phase* shift architecture in line with the pseudo 4 phase-shift algorithm. We exploit the fact that the 4 sequential stages must be coherent if the pixel captures the same depth during the integration time. We show that we can recover depth of a pixel affected by motion with minimum calibration and information found in neighbours.

Even though the assumption that phases are linear mixtures is not exact for real ToF cameras, we demonstrate in our experiments that this assumption provides good results in terms of correction and motion estimation. We consider that more precise models could improve the interpolation of the event location and thus the motion field estimation, providing an interesting research line in the future.

We show in the Results section that the method clearly corrects complex scenes captured with a low cost commercial ToF camera. The camera we use has some specific limitations due to its small fill factor. Using better cameras will only improve the results of our proposal. In all experiments we also show that our approach obtains better results when compared to multiple frame state of the art techniques, specially with high dynamics.

The proposal is implemented in C++ without the need of any hardware implementations to achieve real time processing (around 55 fps). Nevertheless, the algorithm could be implemented easily in a GPU if needed.

[3] Coarse events colour-map can be seen in Fig. 5.13.

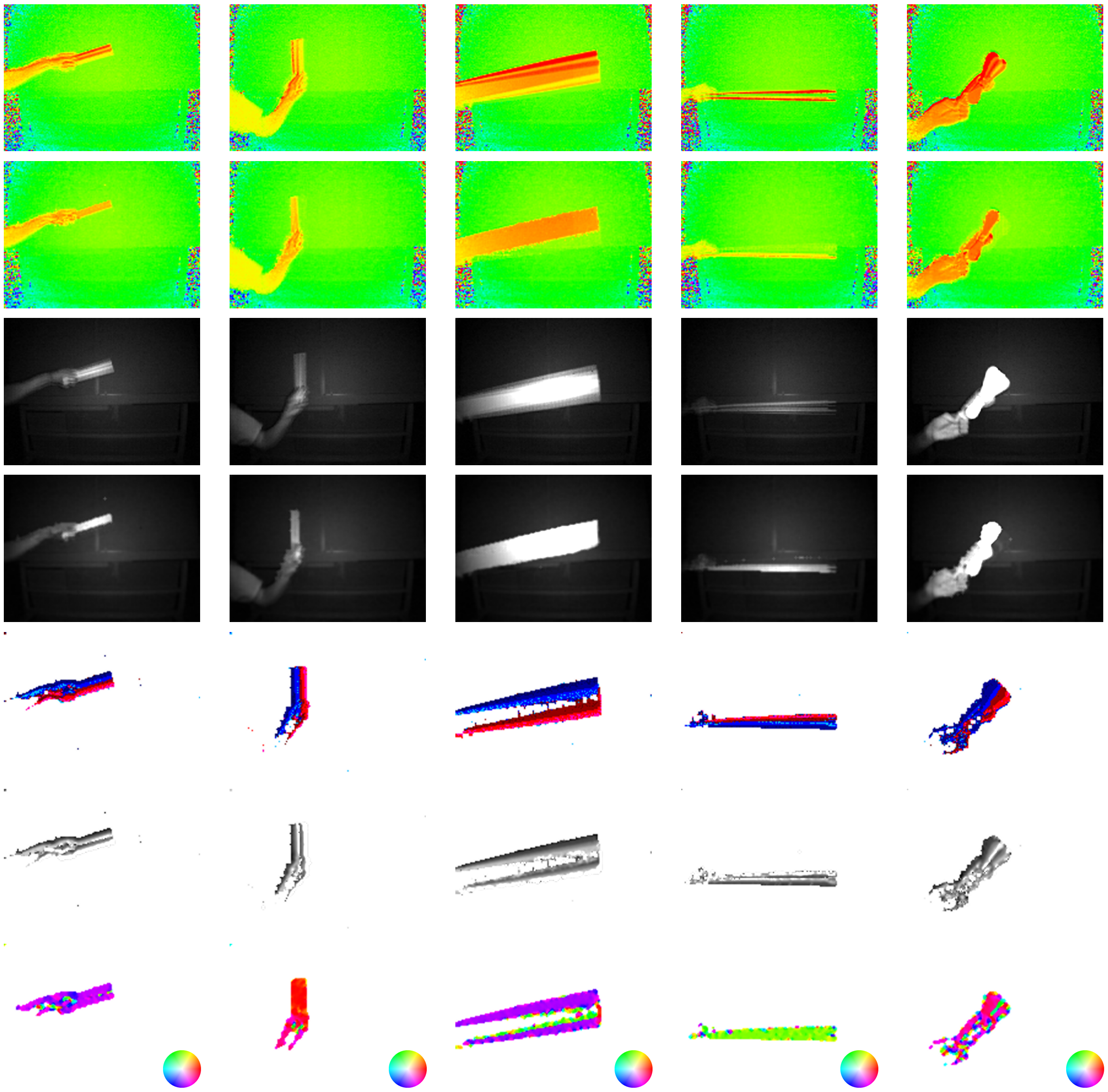


Figure 5.18: Qualitative results 1. Each column shows the results for a particular capture. In rows 1 and 3 we show depth and amplitude measurements and the proposed correction in rows 2 and 4. We show in rows 5 and 6 the coarse and the extrapolated event location. Motion flow is shown in the last row.

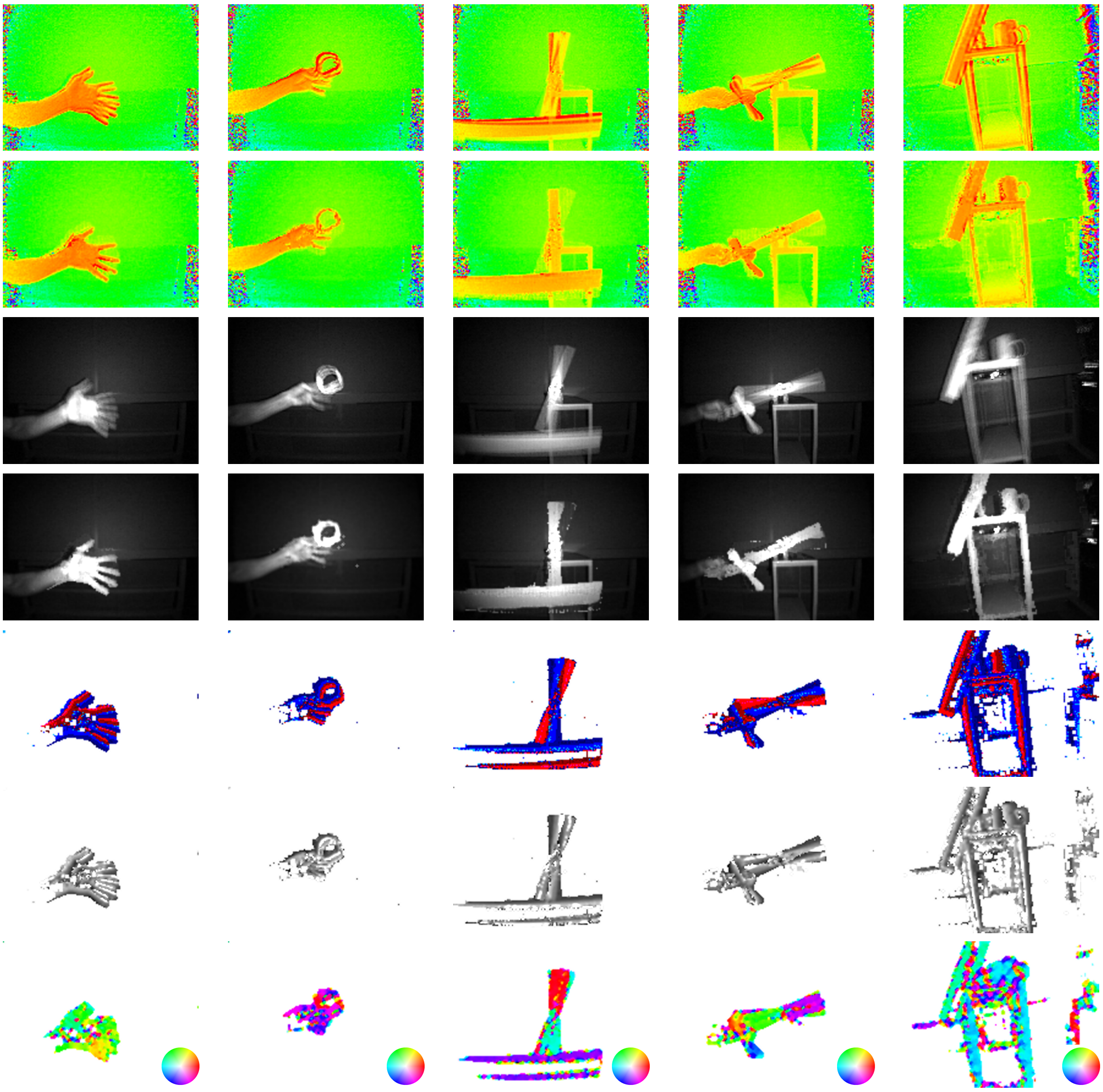


Figure 5.19: Qualitative results 2. Each column shows the results for a particular capture. In rows 1 and 3 we show depth and amplitude measurements and the proposed correction in rows 2 and 4. We show in rows 5 and 6 the coarse and the extrapolated event location. Motion flow is shown in the last row.

Chapter 6

Conclusions

6.1 Introduction

This chapter presents the conclusions derived from the thesis through a summary of the main contributions. Finally, a discussion about limitations and future lines of work is presented.

6.2 Conclusions and Contributions

In this thesis we have justified the importance of the correction of errors in *Continuous Wave Modulation (CWM) Time of Flight (ToF)* cameras for obtaining accurate reliable and repeatable depth measurements. During the thesis we made a clear distinction between *i*) systematic and *ii*) non-systematic errors. Each category of errors requires very different approaches. While systematic allows calibrations, in non-systematic the correction requires to process the data. In particular, this thesis has addressed three main kind of errors: *i*) *systematic depth distortion* and *ii.a*) *Multipath Interference (MpI)* and *ii.b*) *motion artifacts*, respectively for systematic and non-systematic error sources.

Additionally, we also presented in this thesis a new geometric primitive, called *) *Linear Plane Border (LPB)*, which is specially adapted to depth images.

i) In this thesis we have presented a calibration method for correcting *systematic depth distortion*, which is caused mainly due to the use of non-ideal sinusoidal signals for light modulation. We have shown that calibration of depth cameras does not follow the same rules that made geometric calibration of colour cameras a standard procedure. In depth sensors, checker-board patterns are not specially accurate and usually calibration methods use very special equipment. We thus have proposed in this thesis a calibration method that does not require correspondences and that uses planes as the support for calibration. These planes can be natural planes from the scene which makes this method ideal for calibration in man-made environments. Our calibration procedure finds the depth correction that forces the observations to be planar. This problem has been studied before using non-convex optimisation. Existing methods are sub-optimal and do not ensure accurate calibration results. We showed that this problem can be reformulated as a linear least-squares (and thus convex) problem using differential planarity constraints. Our method improves the state of the art in terms of accuracy, simplicity and computational complexity. Our validation includes simulated and real experiments with three commercial depth cameras (MESA SR4k [67], PMD CamCube 3 [68] and Microsoft Kinect v1).

ii.a) In this thesis we proposed an algorithm that removes *multipath distortion* from a single depth map obtained by a ToF camera. Correction of MpI from ToF measurements is a very challenging problem as the amount of distortion depends on the scene's geometry, which is unknown. We have faced this challenge by using a generative model fitting approach. We have proposed a generative model of depth measurements based on simple radiometric properties. Then, we use this generative model to obtain the scene's geometry that, used as an input of our model, reproduce the distorted measurements seen by the camera. We showed results of this idea with both synthetic and real scenes captured by commercial ToF sensors. In all cases, our algorithm accurately corrected the multipath distortion, obtaining depth maps that are very close to ground truth data.

ii.b) *Motion artifacts* in ToF imaging generates strong distortion in moving parts of the scene. We proposed to detect and remove motion artifacts from a single frame taken by a *Photonic Mixer Device (PMD)* ToF camera. Unlike in other depth sensors, PMD pixels divide the integration time in four stages where the correlation function between modulation signal and incoming signal is sampled at different phase shifts. Therefore, the main idea is to detect at which stage of the pipeline motion artifacts are creating the distortion. After this detection we have shown how to reconstruct the depth by exploiting consistency of the correlation samples and local neighbours of the pixel. In addition, our method obtains the motion flow of occluding contours in the image from a single frame. The algorithm we proposed is very fast, simple and can be easily included in camera's hardware. Our correction algorithm has been validated in real scenes using a commercial low-cost PMD camera and high speed dynamics. In all cases, our method produced accurate results and it highly reduced motion artifacts.

***) In this work, we proposed a new primitive specially formulated for depth cameras, that combines properties of planes and lines: LPB. These are planar stripes of a certain width that are delineated at one side by a linear edge (i.e. depth discontinuity). The design of this primitive is motivated by the contours of many man-made objects. We extend the J-Linkage algorithm to robustly detect multiple LPBs in range images from noisy sensors. We validated our method using qualitative and quantitative experiments with real scenes. Results showed that LPBs are able to capture rich information about the objects in the scene compared to looking only for planar structures. It also behaves reasonably well in curved objects.

6.3 Future Work

We present in this section some hints about future research lines that could follow the work presented in this thesis.

i) The proposed calibration procedure for the correction of systematic depth distortion requires some parameter tuning. We showed in our experiments that only two of the hyperparameters needed to be roughly adjusted to get proper results. In practice, we empirically chose them. Future works could handle an automatic selection of these values, possible because of how affordable is to get a big amount of plane views.

ii.a) At the moment, the computational cost of the proposed method for the correction of MpI distortion is high (taking several minutes for a small scene). However, we found some opportunities to speed it up. For example, we can substitute the pixel-based approach by a path-based approach. By replacing individual pixels with super-pixels that share properties, time-demanding routines could be implemented very efficiently. We have also considered to introduce a more realistic radiometric model, however, computational cost would be penalised and there are no hints about great improvements.

ii.b) We proposed a method to correct motion artifacts in current PMD-based ToF cameras. Commercially available ToF cameras do not provide information of each of the CMOS channels of a PMD pixels, but its difference. This limitation introduces some uncertainty to the proposed method. If manufacturers trend suddenly changed and future ToF cameras provide such information, further improvements would be made in order to avoid these errors.

*) Currently, the detection of LPB primitives in depth images is restricted to depth edges. We consider that, extending the proposed method to detect edges due to changes in plane orientations (that are not detected now as jump edges), will increase the number of LPBs that can be obtained from a geometric object. We also believe that LPBs can be extended to curved objects, using curved stripes attached to jump edges.

Bibliography

[1] R. Lange and P. Seitz, “Solid-state time-of-flight range camera,” *Quantum Electronics, IEEE Journal of*, vol. 37, pp. 390 –397, 2001.

[2] A. Payne, A. Daniel, A. Mehta, B. Thompson, C. S. Bamji, D. Snow, H. Oshima, L. Prather, M. Fenton, L. Kordus *et al.*, “A 512× 424 cmos 3d time-of-flight image sensor with multi-frequency photo-demodulation up to 130 mhz and 2 gs/s adc,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), IEEE International*, 2014, pp. 134–135.

[3] E. Kollorz, J. Penne, J. Hornegger, and A. Barke, “Gesture recognition with a time-of-flight camera,” *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 3, pp. 334–343, 2008.

[4] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt, “3d shape scanning with a time-of-flight camera,” in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2010, pp. 1173–1180.

[5] S. May, D. Dröschel, S. Fuchs, D. Holz, and A. Nuchter, “Robust 3D-mapping with time-of-flight cameras,” in *Intelligent Robots and Systems (IROS). IEEE/RSJ International Conference on*, 2009, pp. 1673–1678.

[6] M. Van den Bergh, D. Carton, R. de Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss, “Real-time 3d hand gesture interaction with a robot for understanding directions from humans,” in *RO-MAN, IEEE*, July 2011, pp. 357–362.

[7] S. Soutschek, J. Penne, J. Hornegger, and J. Kornhuber, “3d gesture-based scene navigation in medical imaging applications using time-of-flight cameras,” in *Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE Computer Society Conference on*, 2008, pp. 1–6.

[8] J. Penne, K. Höller, M. Stürmer, T. Schrauder, A. Schneider, R. Engelbrecht, H. Feußner, B. Schmauss, and J. Hornegger, “Time-of-flight 3-d endoscopy,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI).* Springer, 2009, pp. 467–474.

[9] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images,” in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, vol. 2, 2011, p. 7.

[10] S. Hsu, S. Acharya, A. Rafii, and R. New, “Performance of a time-of-flight range camera for intelligent vehicle safety applications,” in *Advanced Microsystems for Automotive Applications*. Springer, 2006, pp. 205–219.

[11] D. Falie and V. Buzuloiu, “Wide range time-of-flight camera for outdoor surveillance,” in *Microwaves, Radar and Remote Sensing Symposium (MRRS)*. IEEE, 2008, pp. 79–82.

[12] A. Velten, T. Willwacher, O. Gupta, A. Veeraraghavan, M. G. Bawendi, and R. Raskar, “Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging,” *Nature Communications*, vol. 3, p. 745, 2012.

[13] D. Wu, A. Velten, M. O’Toole, B. Masia, A. Agrawal, Q. Dai, and R. Raskar, “Decomposing global light transport using time-of-flight imaging,” *International Journal of Computer Vision*, vol. 107, pp. 123–138, 2014.

[14] L. Wang, M. Gong, C. Zhang, R. Yang, C. Zhang, and Y. Yang, “Automatic real-time video matting using time-of-flight camera and multichannel poisson equations,” *International Journal of Computer Cision*, pp. 104–121, 2012.

[15] C. Redondo-Cabrera, R. Lopez-Sastre, S. Maldonado-Bascon, and A.-R. J., “SURFing the point clouds: Selective 3D spatial pyramids for category-level object recognition,” in *Computer Vision and Pattern Recognition (CVPR). IEEE Conference on*, 2012, pp. 3458–3465.

[16] A. Kolb, “Foundations of time-of-flight cameras and their application to surface reconstruction,” in *Workshop on Optical Techniques for 3D Surface Reconstruction in Computer-Assisted Laparoscopic Surgery (MICCAI)*, 2011.

[17] S. Foix, G. Alenya, and C. Torras, “Lock-in time-of-flight (tof) cameras: a survey,” *Sensors Journal, IEEE*, vol. 11, pp. 1917–1926, 2011.

[18] D. Jimenez, D. Pizarro, M. Mazo, and S. Palazuelos, “Modelling and correction of multipath interference in time of flight cameras,” in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2012, pp. 893 – 900.

[19] D. Jimenez, S. Behnke, and D. Pizarro, “Linear plane border. a primitive for range images combining depth edges and surface points,” in *Computer Vision Theory and Applications (VISAPP), 8th International Conference on*, 2013.

[20] D. Jimenez, D. Pizarro, M. Mazo, and S. Palazuelos, “Modeling and correction of multipath interference in time of flight cameras,” *Image and Vision Computing*, vol. 32, pp. 1 – 13, 2014.

[21] D. Jimenez, D. Pizarro, and M. Mazo, “Single frame correction of motion artifacts in pmd-based time of flight cameras,” *Image and Vision Computing*, vol. 32, pp. 1127 – 1143, 2014.

[22] R. Dändliker, Y. Salvadé, and E. Zimmermann, “Distance measurement by multiple-wavelength interferometry,” *Journal of optics*, vol. 29, p. 105, 1998.

[23] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge Univ Press, 2000, vol. 2.

[24] S. T. Barnard and W. B. Thompson, "Disparity analysis of images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pp. 333–340, 1980.

[25] Z. Zhang and O. Faugeras, "3d dynamic scene analysis: a stereo based approach," *UC Berkeley Transportation Library*, 1992.

[26] J. Smisek, M. Jancosek, and T. Pajdla, "3d with kinect," in *Computer Vision (ICCV) Workshops, IEEE International Conference on*, 2011, pp. 1154–1160.

[27] S. Hussmann, M. Gonschior, B. Buttgen, C. Peter, S. Schwope, C. Perwass, J. M., and E. Hallstig, *A Review on Commercial Solid State 3D Cameras for Machine Vision Applications.* Nova Science Publishers Inc., 2013, ch. 11, pp. 303–352.

[28] S. Fuchs and S. May, "Calibration and registration for precise surface reconstruction with tof cameras," in *Dynamic 3D Imaging Workshop in Conjunction with DAGM (Dyn3D)*, vol. 1, 2007.

[29] S. Fuchs and G. Hirzinger, "Extrinsic and depth calibration of tof-cameras," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2008, pp. 1–6.

[30] T. Kahlmann, F. Remondino, and H. Ingensand, "Calibration for increased accuracy of the range imaging camera swissranger," *Image Engineering and Vision Metrology (IEVM)*, vol. 36, pp. 136–141, 2006.

[31] M. Lindner and A. Kolb, "Lateral and depth calibration of pmd-distance sensors," in *Advances in Visual Computing.* Springer, 2006, pp. 524–533.

[32] Lindner, M. and Kolb, A., "Calibration of the intensity-related distance error of the PMD ToF-camera," in *Optics East.* International Society for Optics and Photonics, 2007.

[33] I. Schiller, C. Beder, and R. Koch, "Calibration of a pmd-camera using a planar calibration pattern together with a multi-camera setup," *The international archives of the photogrammetry, remote sensing and spatial information sciences*, vol. 37, pp. 297–302, 2008.

[34] S. Hussmann, F. Knoll, and T. Edeler, "Modulation method including noise model for minimizing the wiggling error of tof cameras," *Instrumentation and Measurement, IEEE Transactions on*, vol. 63, pp. 1127–1136, 2014.

[35] A. Belhedi, A. Bartoli, V. Gay-bellile, S. Bourgeois, P. Sayd, and K. Hamrouni, "Depth correction for depth camera from planarity," in *Proceedings of the British Machine Vision Conference (BMVC).* BMVA Press, 2012, pp. 43.1–43.10.

[36] S. Guomundsson, H. Aanaes, and R. Larsen, "Environmental effects on measurement uncertainties of time-of-flight cameras," in *Signals, Circuits and Systems (ISSCS). International Symposium on*, vol. 1. IEEE, 2007, pp. 1–4.

[37] S. May, S. Fuchs, D. Droeschel, D. Holz, and A. Nüchter, "Robust 3D-mapping with time-of-flight cameras," in *Intelligent Robots and Systems (IROS). IEEE/RSJ International Conference on*, 2009.

[38] D. Falie and V. Buzuloiu, "Distance errors correction for the time-of-flight (ToF) cameras," in *Imaging Systems and Techniques (IST). IEEE International Workshop on*, 2008, pp. 123–126.

[39] Falie, D. and Buzuloiu, V., "Further investigations on ToF cameras distance errors and their corrections," in *Circuits and Systems for Communications (ECCSC). 4th European Conference on.* IEEE, 2008, pp. 197–200.

[40] A. Dorrington, J. Godbaz, M. Cree, A. Payne, and L. Streeter, "Separating true range measurements from multi-path and scattering interference in commercial range cameras," in *IS&T/SPIE Electronic Imaging.* International Society for Optics and Photonics, 2011, pp. 786 404–786 404.

[41] S. Fuchs, "Multipath interference compensation in time-of-flight camera images," *Pattern Recognition (ICPR). International Conference on*, pp. 3583–3586, 2010.

[42] S. Fuchs, M. Suppa, and O. Hellwich, "Compensation for multipath in tof camera measurements supported by photometric calibration and environment integration," in *Computer Vision Systems.* Springer, 2013, pp. 31–41.

[43] J. Godbaz, M. Cree, and A. Dorrington, "Closed-form inverses for the mixed pixel/multipath interference problem in amcw lidar," in *IS&T/SPIE Electronic Imaging.* International Society for Optics and Photonics, 2012, pp. 829 618–829 618.

[44] A. Kirmani, A. Benedetti, and P. Chou, "Spumic: Simultaneous phase unwrapping and multipath interference cancellation in time-of-flight cameras using spectral methods," in *Multimedia and Expo (ICME), IEEE International Conference on*, 2013, pp. 1–6.

[45] A. Bhandari, A. Kadambi, R. Whyte, C. Barsi, M. Feigin, A. Dorrington, and R. Raskar, "Resolving multi-path interference in time-of-flight imaging via modulation frequency diversity and sparse regularization," *Optics Letters*, vol. 39, pp. 1705–1708, 2014.

[46] D. Freedman, E. Krupka, Y. Smolin, I. Leichter, and M. Schmidt, "Sra: Fast removal of general multipath for tof sensors," *arXiv preprint arXiv:1403.5919*, 2014.

[47] M. Gupta, S. Nayar, M. Hullin, and J. Martin, "Phasor imaging: A generalization of correlation-based time-of-flight imaging," 2014.

[48] O. Lottner, A. Sluiter, K. Hartmann, and W. Weihs, "Movement artefacts in range images of time-of-flight cameras," in *Signals, Circuits and Systems (ISSCS). International Symposium on*, vol. 1. IEEE, 2007, pp. 1–4.

[49] M. O. Schmidt, "Spatiotemporal analysis of range imagery," Ph.D. dissertation, Combined Faculties for the Natural Sciences and for Mathematics of the Ruperto-Carola University of Heidelberg, Germany, 2008.

[50] S. Hussmann, A. Hermanski, and T. Edeler, "Real-time motion supression in tof range images," in *Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE, 2010, pp. 697–701.

[51] S. Lee, B. Kang, J. D. Kim, and C. Y. Kim, "Motion blur-free time-of-flight range sensor," in *Proceedings of the SPIE Electronic Imaging*, 2012.

[52] S. Lee, H. Shim, J. D. Kim, and C. Y. Kim, "Tof depth image motion blur detection using 3d blur shape models," in *Proceedings of the SPIE Electronic Imaging*, 2012.

[53] M. Lindner and A. Kolb, "Compensation of motion artifacts for time-of-flight cameras," in *Dynamic 3D Imaging*. Springer, 2009, pp. 16–27.

[54] M. Schmidt and B. Jahne, "Efficient and robust reduction of motion artifacts for 3d time-of-flight cameras," in *3D Imaging (IC3D), IEEE International Conference on*, 2011, pp. 1–8.

[55] T. Hoegg, D. Lefloch, and A. Kolb, "Real-time motion artifact compensation for pmd-tof images," in *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications.* Springer, 2013, pp. 273–288.

[56] D. Lefloch, T. Hoegg, and A. Kolb, "Real-time motion artifacts compensation of tof sensors data on gpu," in *SPIE Defense, Security, and Sensing.* International Society for Optics and Photonics, 2013, pp. 87 380–87 380.

[57] P. F. Sturm and S. J. Maybank, "On plane-based camera calibration: A general algorithm, singularities, applications," in *Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society Conference on*, vol. 1, 1999.

[58] S. Ramalingam, P. Sturm, and S. K. Lodha, "Generic self-calibration of central cameras," *Computer Vision and Image Understanding*, vol. 114, pp. 210–219, 2010.

[59] K. Khoshelham, "Accuracy analysis of kinect depth data," in *ISPRS Journal of Photogrammetry and Remote Sensing. Workshop on Laser Scanning*, vol. 38, 2011.

[60] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, pp. 1437–1454, 2012.

[61] C. Herrera, J. Kannala, J. Heikkilä *et al.*, "Joint depth and color camera calibration with distortion correction," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, pp. 2058–2064, 2012.

[62] C. Raposo, J. P. Barreto, and U. Nunes, "Fast and accurate calibration of a kinect sensor," in *3DTV-Conference, IEEE International Conference on*, 2013, pp. 342–349.

[63] J. C. Chow and D. D. Lichti, "Photogrammetric bundle adjustment with self-calibration of the primesense 3d camera technology: Microsoft kinect," *Access, IEEE*, vol. 1, pp. 465–474, 2013.

[64] E. A. Coddington, *An introduction to ordinary differential equations.* Courier Dover Publications, 2012.

[65] F. L. Bookstein, “Principal warps: Thin-plate splines and the decomposition of deformations,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 567–585, 1989.

[66] A. Bartoli, “Maximizing the predictivity of smooth deformable image warps through cross-validation,” *Journal of Mathematical Imaging and Vision*, vol. 31, pp. 133–145, 2008.

[67] MESA, “MESA imaging AG,” http://www.mesa-imaging.ch/, 2014.

[68] PMD, “PMD Technologies,” http://www.pmdtec.com/, 2014.

[69] D. Forsyth and J. Ponce, *Computer vision: a modern approach.* Prentice Hall Professional Technical Reference, 2002.

[70] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew, “On the removal of shadows from images,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, pp. 59–68, 2006.

[71] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, “Shape-from-shading: a survey,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, pp. 690–706, 1999.

[72] Z. Zhang, “A flexible new technique for camera calibration,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pp. 1330–1334, 2000.

[73] A. Georghiades, P. Belhumeur, and D. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” *Pattern Analysis and Machine Intelligence. IEEE Transactions on*, vol. 23, pp. 643–660, 2001.

[74] S. Biswas, G. Aggarwal, and R. Chellappa, “Robust estimation of albedo for illumination-invariant matching and shape recovery,” *Pattern Analysis and Machine Intelligence. IEEE Transactions on*, vol. 31, no. 5, pp. 884–899, 2009.

[75] K. Klasing, D. Althoff, D. Wollherr, and M. Buss, “Comparison of surface normal estimation methods for range sensing applications,” in *Robotics and Automation (ICRA). IEEE International Conference on*, 2009, pp. 3206–3211.

[76] M. Keller and A. Kolb, “Real-time simulation of time-of-flight sensors,” *Simulation Modelling Practice and Theory*, vol. 17, pp. 967–978, 2009.

[77] S. Hussmann, T. Edeler, and A. Hermanski, “Real-time processing of 3d-tof data in machine vision applications,” *Machine Vision - Applications and Systems*, 2012.

[78] M. Lindner, “Calibration and real-time processing of time-of-flight range data,” Ph.D. dissertation, Siegen University, 2010.

[79] S. Hussmann and T. Edeler, “Pseudo-four-phase-shift algorithm for performance enhancement of 3d-tof vision systems,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, pp. 1175–1181, 2010.

[80] S. Hussmann and A. Hermanski, “One-phase algorithm for continuous wave tof machine vision applications,” *Instrumentation and Measurement. IEEE Transactions on*, vol. 62, pp. 991–998, 2013.

[81] F. Mufti and R. Mahony, “Statistical analysis of signal measurement in time-of-flight cameras,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, pp. 720–731, 2011.

[82] S. Hussmann, F. Knoll, and T. Edeler, “Modulation method including noise model for minimizing the wiggling error of tof cameras,” *Instrumentation and Measurement, IEEE Transactions on*, no. 99, pp. 1–1, 2013.

[83] A. Bartoli, “A random sampling strategy for piecewise planar scene segmentation,” *Computer Vision and Image Understanding*, vol. 105, pp. 42–59, 2007.

[84] B. Steder, R. Rusu, K. Konolige, and W. Burgard, “Point feature extraction on 3D range scans taking into account object boundaries,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 2601–2608.

[85] T. Fiolka, J. Stückler, D. Klein, D. Schulz, and S. Behnke, “Sure: Surface entropy for distinctive 3d features,” *Spatial Cognition VIII*, pp. 74–93, 2012.

[86] Y. Taguchi, O. Tuzel, M.-Y. Liu, and S. Ramalingam, “Voting-based pose estimation for robotic assembly using a 3d sensor,” in *Robotics and Automation (ICRA). IEEE International Conference on*, 2012, pp. 1724–1731.

[87] L. Xu, E. Oja, and P. Kultanen, “A new curve detection method: randomized hough transform (rht),” *Pattern Recognition Letters*, vol. 11, pp. 331–338, 1990.

[88] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 603–619, 2002.

[89] M. Fischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[90] M. Zuliani, C. Kenney, and B. Manjunath, “The multiransac algorithm and its application to detect planar homographies,” in *Image Processing (ICIP). IEEE International Conference on*, vol. 3, 2005, pp. III–153.

[91] R. Toldo and A. Fusiello, “Robust multiple structures estimation with J-Linkage,” *Computer Vision–ECCV 2008*, pp. 537–547, 2008.

[92] W. Zhang and J. Kosecka, “Non-parametric estimation of multiple structures with outliers,” *Dynamical Vision*, pp. 60–74, 2007.

[93] D. Fouhey, D. Scharstein, and A. Briggs, “Multiple plane detection in image pairs using J-Linkage,” in *Pattern Recognition (ICPR), International Conference on*, 2010.

[94] C. Feng, F. Deng, and V. R. Kamat, “Semi-automatic 3D reconstruction of piecewise planar building models from single image,” *Construction Applications of Virtual Reality. The 10th International Conference on*, 2010.

[95] L. Schwarz, D. Mateus, J. Lallemand, and N. Navab, “Tracking planes with time of flight cameras and J-Linkage,” in *Applications of Computer Vision (WACV), IEEE Workshop on*, 2011, pp. 664–671.

[96] R. Toldo and A. Fusiello, “Real-time incremental J-Linkage for robust multiple structures estimation,” in *3D Data Processing, Visualization and Transmission (3DPVT). International Symposium on*, 2010.

Appendix A

Single Frame Correction of Motion Artifacts in PMD-based ToF Cameras

A.1 Single Frame Coarse Event Location

Algorithm 2 How to detect the event location in a single PMD ToF capture.

```
Require: $\varphi_{\{0,\pi/2,\pi,3\pi/2\}}$, $Q^{bg}$, $Q^{fg}$
  for every pixel do
    compute inconsistency value;   $incons = \left|\varphi_0 + \varphi_\pi - \varphi_{\pi/2} - \varphi_{3\pi/2}\right|$
    if $(incons > \gamma)$ then
      if $\left(|\varphi_0 - \varphi_\pi| > \gamma \;\;\&\&\;\; \left|\varphi_{\pi/2} - \varphi_{3\pi/2}\right| < \gamma\right)$ then
        Event at $\varphi_0 \rightarrow\; E = 1$
        Detect Event Direction (see Table A.1 and Table A.2)
      else if $\left(|\varphi_0 - \varphi_\pi| < \gamma \;\;\&\&\;\; \left|\varphi_{\pi/2} - \varphi_{3\pi/2}\right| > \gamma\right)$ then
        Event at $\varphi_{3\pi/2} \rightarrow\; E = 4$
        Detect Event Direction (see Table A.1 and Table A.2)
      else
        if $\left(|\varphi_0 - \varphi_\pi| > \left|\varphi_{\pi/2} - \varphi_{3\pi/2}\right|\right)$ then
          Event at $\varphi_{\pi/2} \rightarrow\; E = 2$
          Detect Event Direction (see Table A.1 and Table A.2)
        else
          Event at $\varphi_\pi \rightarrow\; E = 3$
          Detect Event Direction (see Table A.1 and Table A.2)
        end if
      end if
    end if
  end for
```

A.2 Characterisation of the Event Direction

Condition summary:

$$\text{If } |cel| = 1 \text{ then } \begin{cases} \varphi_0 + \varphi_\pi &= K_1 \cdot (a'_1 \cdot \cos\beta_1 - a'_2 \cdot \cos\beta_2) \\ \varphi_{\pi/2} + \varphi_{3\pi/2} &= 0 \end{cases}$$

$$\text{If } |cel| = 2 \text{ then } \begin{cases} \varphi_0 + \varphi_\pi &= K_{int} \cdot (a'_1 \cdot \cos\beta_1 - a'_2 \cdot \cos\beta_2) \\ \varphi_{\pi/2} + \varphi_{3\pi/2} &= K_1 \cdot (a'_2 \cdot \sin\beta_2 - a'_1 \cdot \sin\beta_1) \end{cases}$$

$$\text{If } |cel| = 3 \text{ then } \begin{cases} \varphi_0 + \varphi_\pi &= K_2 \cdot (a'_1 \cdot \cos\beta_1 - a'_2 \cdot \cos\beta_2) \\ \varphi_{\pi/2} + \varphi_{3\pi/2} &= K_{int} \cdot (a'_2 \cdot \sin\beta_2 - a'_1 \cdot \sin\beta_1) \end{cases}$$

$$\text{If } |cel| = 4 \text{ then } \begin{cases} \varphi_0 + \varphi_\pi &= 0 \\ \varphi_{\pi/2} + \varphi_{3\pi/2} &= K_2 \cdot (a'_2 \cdot \sin\beta_2 - a'_1 \cdot \sin\beta_1) \end{cases}$$

Initially, if we observe Tables A.1 and A.2 it seems that in $(*)^{ij}$, with $i \in \{1,2,3,4\}$ and $j \in \{a,b\}$, cases we don't have enough information to detect the event direction. Nevertheless, as the amplitude decay is inversely proportional to the squared distance ($a' \propto {}^1/_{d^2}$ and thus, $a' \propto {}^1/_{\beta^2}$), we can detect the event direction from the complementary trigonometric function. That is, if we analyse the case $(*)^{1a}$ where there is an event in $|cel| = 4$ with $\beta_1 \in Q_1$ and $\beta_2 \in Q_2$. In this situation, only *cosine* function seems to provide us relevant information about the sign. However, if we look at the equation below:

$$\varphi_{\pi/2} + \varphi_{3\pi/2} = K_2 \cdot (a'_2 \cdot \sin\beta_2 - a'_1 \cdot \sin\beta_1) \tag{A.1}$$

if $\beta_2 \in Q_2$ and $\beta_1 \in Q_1$ ($\beta_2 > \beta_1$) we can identify the event direction using the *"sine"* function relationship given by the sum $\varphi_{\pi/2} + \varphi_{3\pi/2}$:

$$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0 \tag{A.2}$$

The remaining $(*)^{i\{a,b\}}$ cases have been solved in the same way.

Involved Quad-rants	Flank	Condition	$\lvert cel\rvert = 1$	$\lvert cel\rvert = 2$	$\lvert cel\rvert = 3$	$\lvert cel\rvert = 4$
$Q_1 - Q_1$	$\beta_2 > \beta_1$	$\cos\beta_2 < \cos\beta_1$ $\sin\beta_2 > \sin\beta_1$	$\varphi_0 + \varphi_\pi > 0$	$\varphi_0 + \varphi_\pi > 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_0 + \varphi_\pi > 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$
	$\beta_2 < \beta_1$	$\cos\beta_2 > \cos\beta_1$ $\sin\beta_2 < \sin\beta_1$	$\varphi_0 + \varphi_\pi < 0$	$\varphi_0 + \varphi_\pi < 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_0 + \varphi_\pi < 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$
$Q_1 - Q_2$	$\beta_2 > \beta_1$	$\cos\beta_2 < \cos\beta_1$	$\varphi_0 + \varphi_\pi > 0$	$\varphi_0 + \varphi_\pi > 0$	$\varphi_0 + \varphi_\pi > 0$	$(*)^{1a}$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$
	$\beta_2 < \beta_1$	$\cos\beta_2 > \cos\beta_1$	$\varphi_0 + \varphi_\pi < 0$	$\varphi_0 + \varphi_\pi < 0$	$\varphi_0 + \varphi_\pi < 0$	$(*)^{1b}$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$
$Q_1 - Q_3$	$\beta_2 > \beta_1$	$\cos\beta_2 < \cos\beta_1$ $\sin\beta_2 < \sin\beta_1$	$\varphi_0 + \varphi_\pi > 0$	$\varphi_0 + \varphi_\pi > 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_0 + \varphi_\pi > 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$
	$\beta_2 < \beta_1$	$\cos\beta_2 > \cos\beta_1$ $\sin\beta_2 > \sin\beta_1$	$\varphi_0 + \varphi_\pi < 0$	$\varphi_0 + \varphi_\pi < 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_0 + \varphi_\pi < 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$
$Q_1 - Q_4$	$\beta_2 > \beta_1$	$\sin\beta_2 < \sin\beta_1$	$(*)^{2a}$ $\varphi_0 + \varphi_\pi > 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$
	$\beta_2 < \beta_1$	$\sin\beta_2 > \sin\beta_1$	$(*)^{2b}$ $\varphi_0 + \varphi_\pi < 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$
$Q_2 - Q_2$	$\beta_2 > \beta_1$	$\cos\beta_2 < \cos\beta_1$ $\sin\beta_2 < \sin\beta_1$	$\varphi_0 + \varphi_\pi > 0$	$\varphi_0 + \varphi_\pi > 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_0 + \varphi_\pi > 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$
	$\beta_2 < \beta_1$	$\cos\beta_2 > \cos\beta_1$ $\sin\beta_2 > \sin\beta_1$	$\varphi_0 + \varphi_\pi < 0$	$\varphi_0 + \varphi_\pi < 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_0 + \varphi_\pi < 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$

Table A.1: Part 1: Full depth range test for the detection of the direction of the event.

Involved Quad-rants	Flank	Condition	$\lvert cel\rvert = 1$	$\lvert cel\rvert = 2$	$\lvert cel\rvert = 3$	$\lvert cel\rvert = 4$
$Q_2 - Q_3$	$\beta_2 > \beta_1$	$\sin\beta_2 < \sin\beta_1$	$(*)^{3a}\ \varphi_0 + \varphi_\pi < 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$
	$\beta_2 < \beta_1$	$\sin\beta_2 > \sin\beta_1$	$(*)^{3b}\ \varphi_0 + \varphi_\pi > 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$
$Q_2 - Q_4$	$\beta_2 > \beta_1$	$\cos\beta_2 > \cos\beta_1$ $\sin\beta_2 < \sin\beta_1$	$\varphi_0 + \varphi_\pi < 0$	$\varphi_0 + \varphi_\pi < 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_0 + \varphi_\pi < 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$
	$\beta_2 < \beta_1$	$\cos\beta_2 < \cos\beta_1$ $\sin\beta_2 > \sin\beta_1$	$\varphi_0 + \varphi_\pi > 0$	$\varphi_0 + \varphi_\pi > 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_0 + \varphi_\pi > 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$
$Q_3 - Q_3$	$\beta_2 > \beta_1$	$\cos\beta_2 > \cos\beta_1$ $\sin\beta_2 < \sin\beta_1$	$\varphi_0 + \varphi_\pi < 0$	$\varphi_0 + \varphi_\pi < 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_0 + \varphi_\pi < 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$
	$\beta_2 < \beta_1$	$\cos\beta_2 < \cos\beta_1$ $\sin\beta_2 > \sin\beta_1$	$\varphi_0 + \varphi_\pi > 0$	$\varphi_0 + \varphi_\pi > 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_0 + \varphi_\pi > 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$
$Q_3 - Q_4$	$\beta_2 > \beta_1$	$\cos\beta_2 > \cos\beta_1$	$\varphi_0 + \varphi_\pi < 0$	$\varphi_0 + \varphi_\pi < 0$	$\varphi_0 + \varphi_\pi < 0$	$(*)^{4a}\ \varphi_{\pi/2} + \varphi_{3\pi/2} > 0$
	$\beta_2 < \beta_1$	$\cos\beta_2 < \cos\beta_1$	$\varphi_0 + \varphi_\pi > 0$	$\varphi_0 + \varphi_\pi > 0$	$\varphi_0 + \varphi_\pi > 0$	$(*)^{4b}\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$
$Q_4 - Q_4$	$\beta_2 > \beta_1$	$\cos\beta_2 > \cos\beta_1$ $\sin\beta_2 > \sin\beta_1$	$\varphi_0 + \varphi_\pi < 0$	$\varphi_0 + \varphi_\pi < 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_0 + \varphi_\pi < 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} > 0$
	$\beta_2 < \beta_1$	$\cos\beta_2 < \cos\beta_1$ $\sin\beta_2 < \sin\beta_1$	$\varphi_0 + \varphi_\pi > 0$	$\varphi_0 + \varphi_\pi > 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_0 + \varphi_\pi > 0$ $\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$	$\varphi_{\pi/2} + \varphi_{3\pi/2} < 0$

Table A.2: Part 2: Full depth range test for the detection of the direction of the event.

Appendix B

Related Contributions

B.1 Linear Plane Border: A Primitive for Range Images Combining Depth Edges and Surface Points

B.1.1 Introduction

Detecting primitives, like lines and planes, is a popular first step for the interpretation of range images. Real scenes are, however, often cluttered and range measurements are noisy, such that the detection of pure lines and planes is unreliable. In this work, we propose a new primitive that combines properties of planes and lines: *Linear Plane Border (LPB)*. These are planar stripes of a certain width that are delineated at one side by a linear edge (i.e. depth discontinuity). The design of this primitive is motivated by the contours of many man-made objects. We extend the J-Linkage algorithm to robustly detect multiple LPBs in range images from noisy sensors. We validated our method using qualitative and quantitative experiments with real scenes.

B.1.2 Overview

Currently, depth cameras are experiencing an exponential growth, thanks to emerging mass market applications, that, without a doubt, have stimulated the industry to create cheap devices. At the moment, Time of Flight technology [1] and Kinect [26] sensor technology are leading the market. High resolution and high frame rates are becoming available, which make these sensors a good hardware alternative to many complex stereo video systems.

Depth sensors have had a remarkable impact in many computer vision tasks. Disciplines such as object recognition [15], SLAM [5] and human pose estimation [9] have received multiple, in many case outstanding, scientific contributions based on depth sensing.

As in other image sensing technologies, depth images contain rich and complex information. Therefore, a considerable number of methods and algorithms have focused on detecting basic geometric primitives in depth images. These methods are important and necessary as a building block for high level interpretation and recognition algorithms.

Simplifying a scene as a grouping of multiple piecewise planar models is demonstrated to be an efficient and stable representation of most man-made structures [83]. Depth sensors give powerful 3D information that can be used to find planar structures, providing accurate object and environment detection. However, in complex scenes, fitting planar structures can be challenging and time consuming (i.e. a table crowded with small objects). Moreover, depth sensors produce many errors, such as *Multipath Interference (MpI)* in *Time of Flight (ToF)* cameras [20], that can affect full plane representations.

This work defines a new geometric primitive called *Linear Plane Border* (LPB). It is defined as a planar stripe of a given width that can be detected in the vicinity of linear silhouette edges of geometric objects. LPBs give a compact but rich representation to detect regular objects in depth images. The main clue to find them is by looking at depth discontinuities. The benefit of using LPBs instead of full plane representations is demonstrated in this appendix, in particular in complex scenes.

The use of 2D edge information in conjunction with 3D plane information used for the construction of these primitives provides a simplified but robust representation of planar objects in a scene. That such a combination of 2D edge information and 3D surface information is useful, has been demonstrated recently by the design of 3D interest point detectors and descriptors [84, 85] and by the design of pair features for voting-based object detection [86]. Here, we utilise this insight for a primitive-based object representation.

Detecting multiple LPBs in depth images requires a method able to find multiple instances of a geometrical model from noisy data. Many such methods have been proposed in the literature with remarkable success. Among the most commonly used methods are Hough transforms [87], MeanShift [88] and those based on random sampling such as RANSAC [89, 90].

The proposed method finds multiple LPB hypotheses using the very recent J-Linkage [91] algorithm, based on tailored agglomerative clustering, first proposed by Zhang et al. [92]. This method is able to robustly detect multiple models without prior specification of its number and has shown remarkable detection performance. Recently, there exist some approaches that use J-Linkage to obtain piecewise-planar representation of point clouds. Fouhey et al. [93] present a method for the detection and matching of multiple planes in pairs of images. They use J-Linkage to generate multiple local homography hypothesis. Feng et al. [94] use J-Linkage to minimise the user input compared with traditional Single View Reconstruction approaches. Schwarz et al. [95] suggest some adaptation of the original algorithm for the detection and tracking of multiple planes in sequences of ToF depth images.

Taking as starting point the core of the fast J-Linkage implementation [96], we extend in this work the original algorithm to the detection of LPBs in depth images of ToF cameras. The main contribution of this work is to show that LPBs can be effectively detected in depth images using our modified version of J-Linkage. The experimental results show that this approach outperforms the most common case of searching for planar structures in terms of computational time and accuracy. We strongly believe that LPBs can be very useful in object recognition tasks and object pose computation using depth images.

This chapter is organised as follows. In Section B.1.3 a general diagram of the proposal is given and we briefly explain each step of the proposed method. We evaluate our method qualitatively and quantitatively using real scenes. These results are summarised in Section B.1.4. In Section B.1.5 we describe the conclusions.

B.1.3 Detection of Linear Plane Borders

We represent a depth image as a scalar function $\mathcal{D}$, where $\mathcal{D}(\mathbf{p})$ represents the depth measured at pixel position $\mathbf{p} = (u, v)^\top$. Due to optics, depth values are computed along optical rays. We suppose that the camera optics are properly calibrated so that metric 3-dimensional coordinates are available for each image position $\mathbf{p}$. We denote as $Q(\mathbf{p}) = (Q_x, Q_y, Q_z)^\top$ the three-dimensional coordinates of the point $\mathbf{p}$.

Given a depth image $\mathcal{D}$, the main problem to solve is how to find all LPB candidates, grouping together all pixels belonging to the same LPB. We propose a pipeline consisting of three stages, illustrated in Fig. B.1.

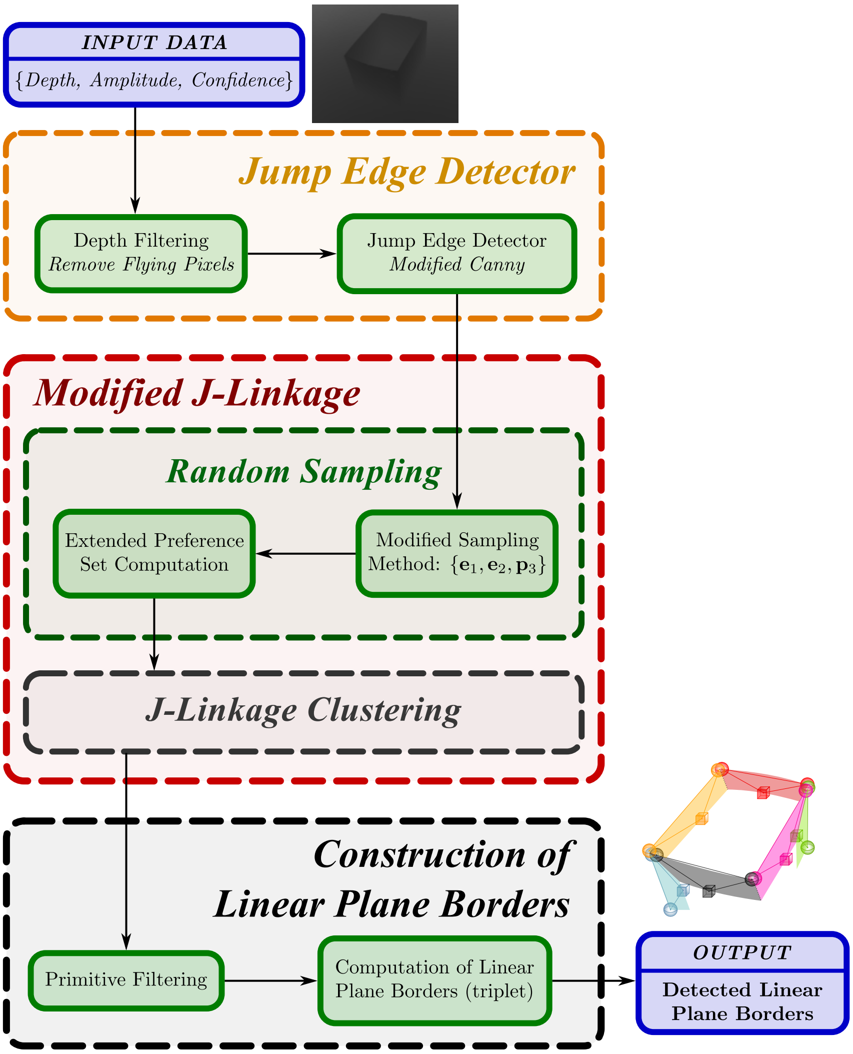

INPUT DATA
{Depth, Amplitude, Confidence}
Jump Edge Detector
Depth Filtering
Remove Flying Pixels
Jump Edge Detector
Modified Canny
Modified J-Linkage
Random Sampling
Extended Preference Set Computation
Modified Sampling Method: $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{p}_3\}$
J-Linkage Clustering
Construction of Linear Plane Borders
Primitive Filtering
Computation of Linear Plane Borders (triplet)
OUTPUT
Detected Linear Plane Borders

Figure B.1: General diagram of the proposed detection of Linear Plane Borders. $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{p}_3\}$ represent the sampled points needed to calculate a LPB hypothesis: two edge points and one point in the correct face of the linear edge.

1. **Jump Edge Detector:** This stage detects depth discontinuities which are the clue of finding edges of geometrical objects. The process is carefully designed to deal with noise and depth artifacts that usually appear around such discontinuities.

2. **Modified J-Linkage:** J-Linkage is an algorithm that groups together points based on the consistency with randomly sampled model hypotheses. In our case, each LPB hypothesis is created by sampling two jump edge points likely belonging to the same plane border and a third point that is far enough from the edge but at a distance limited by the LPB width.

3. **Construction of LPBs:** After running J-Linkage, points in the depth image are grouped in different LPBs hypotheses. This stage refines the LPBs using linear least squares fit producing an accurate estimation of the three points used to parametrise each LPB.

B.1.4 Experiments and Results

This section tests the detection of LPBs in real scenes captured with a commercial ToF camera. We compare LPB detection, as it is proposed in this section against a state-of-art plane fitting using J-Linkage.

In Fig. B.2, we show 7 different scenes composed of planar structures and the result of each step of the algorithm. We also add the result of detecting planes with J-Linkage where input data have been re-sampled in a **1:4 ratio** (one of each two columns and rows), so that the number of input data is comparable in both approaches. In these experiments, the number of J-Linkage random samples are around {5000-7000} for LPB and almost ten times more for planes.

As quantitative results we show the accuracy of LPB detection in Table B.1. Ground-truth data are obtained by manual annotation over the depth images so that the different planes it is composed of are clearly identified. Error is measured as the mean of the Euclidean distance between points projected onto the ground-truth plane and the projection onto the plane obtained in LPBs detection.

Plane Detection Accuracy	
Example	Proposal
1	0.00123253
2	0.00176433
3	0.00192681
4	0.00280092
5	0.00795743
6	0.00384587
7	0.00461953

Table B.1: Plane detection accuracy using LPB.

It is clearly shown that LPB detection is very accurate, capturing very useful geometric information. It is also faster than detecting only planes as it is shown in Table B.2.

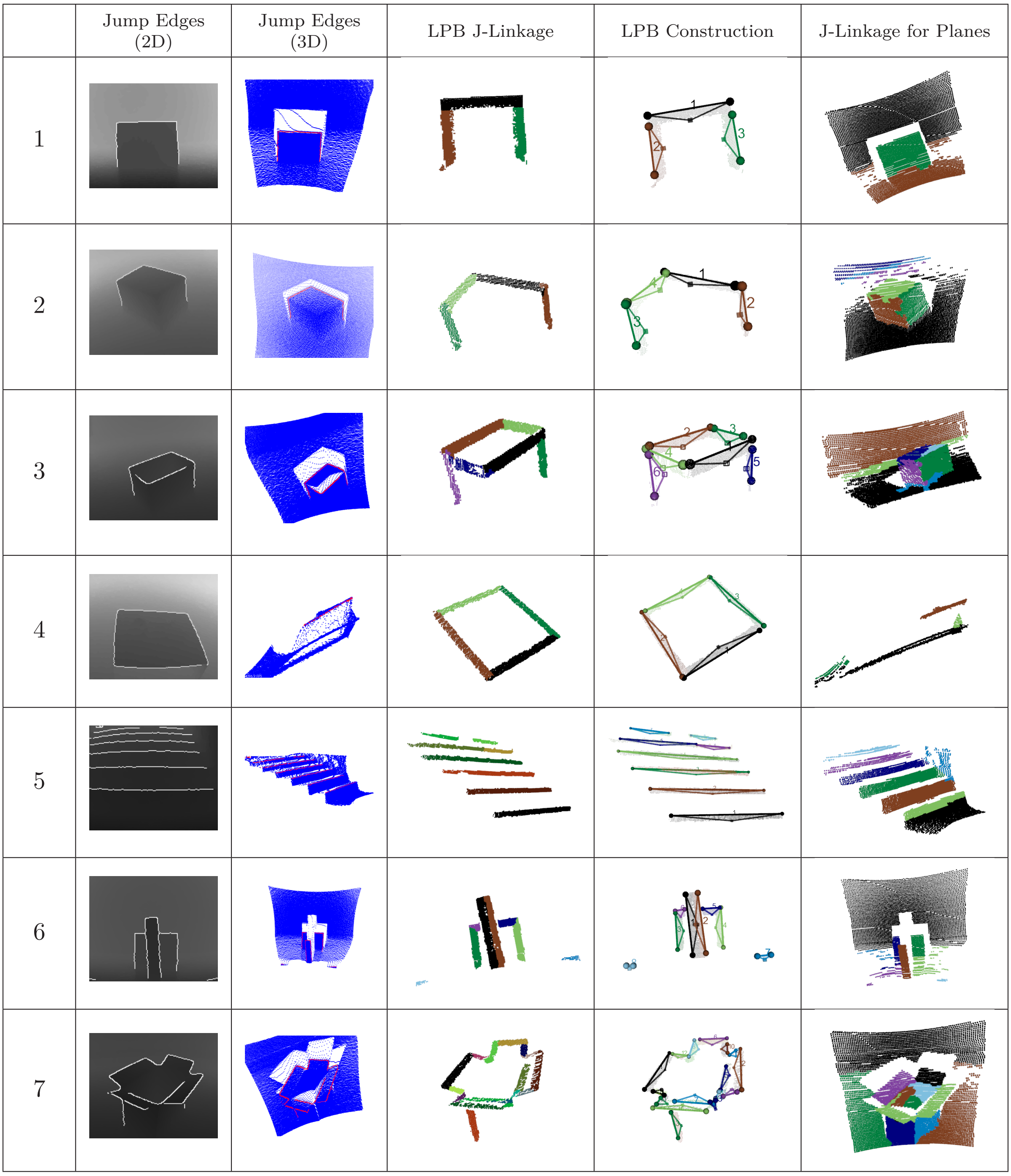


Figure B.2: Comparison between the detection of planes using the original J-Linkage method and the main stages of the LPB detection algorithm.

Achieved Speed-Up							
Example	1	2	3	4	5	6	7
Random Sampling	x124.5	x103.5	x101.2	x78.3	x48.2	x85.6	x74.5
Clustering	x8.8	x23.1	x6.1	x4.7	x1.8	x9.1	x5.4

Table B.2: Achieved speed-up for the main consuming processes.

In a qualitative manner we show in Fig. B.3 the results with curved objects (non-planar objects can also be approximated as an arrangement of LPBs). While these scenes are not piece-wise planar, LPB detection is giving useful and stable information.

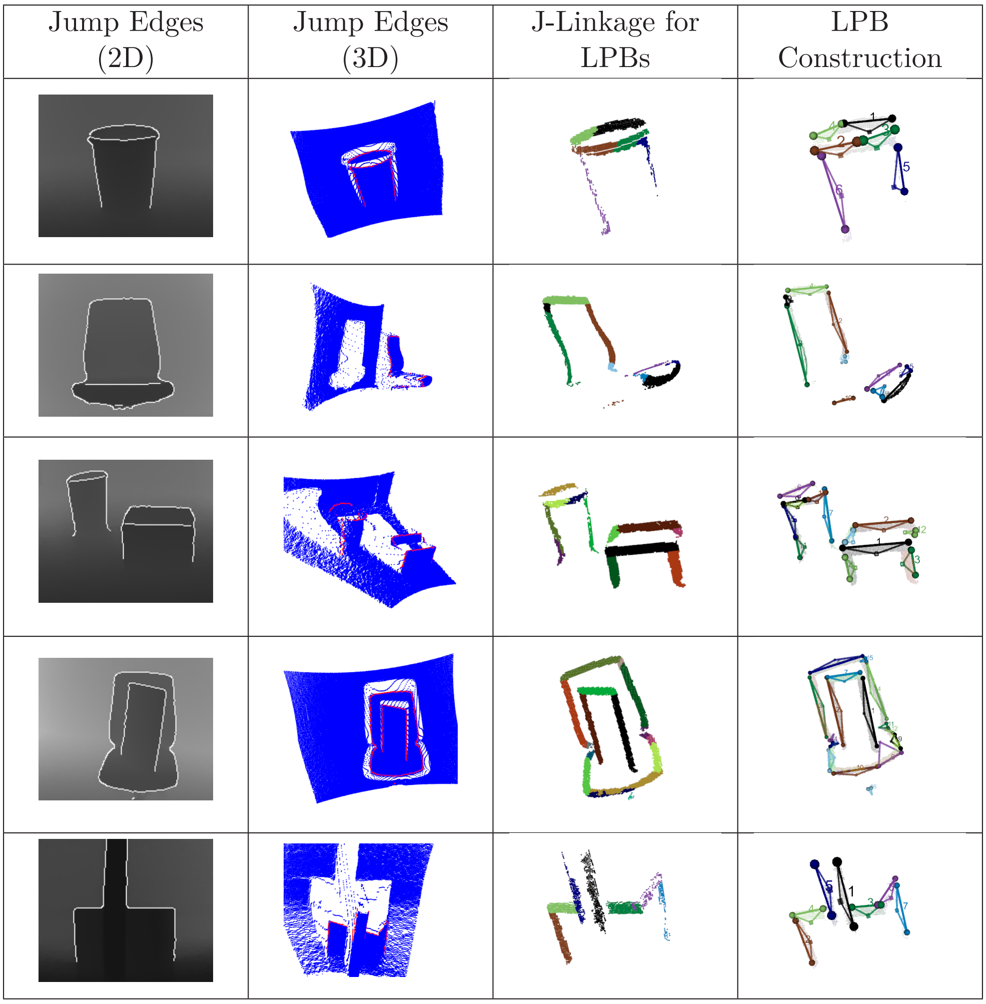


Figure B.3: Qualitative results for curved objects in more complex scenarios.

B.1.5 Conclusions

This work shows that LPBs can be efficiently detected in depth images, taken with a commercial depth sensor of real man-made scenes composed of planar structures.

The proposed semi-deterministic sampling method for LPBs allows to perform J-Linkage with ten times less hypothesis than general plane fitting methods. This reduction has a critical impact on the algorithm's speed.

As it can be seen in the results, LPBs are able to capture rich information about the objects in the scene compared to looking only for planar structures. It also behaves reasonably well in curved objects.