TECHNISCHE UNIVERSITÄT MÜNCHEN

Chair for Computer-Aided Medical Procedures & Augmented Reality

Constructive interference for Multi-view Time-of-Flight acquisition

Víctor Antonio Castañeda Zeman

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Darius Burschka

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Nassir Navab
2. Prof. Dr. Peter Sturm,
   INRIA Grenoble / Frankreich

Die Dissertation wurde am 22.12.2011 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 17.03.2012 angenommen.

Acknowledgments

It is a pleasure to thank those who made this thesis possible. I want to thank the German Academic Exchange Service (DAAD) and the Chilean National Commission for Science and Technology (CONICYT), whose funding through the program "Becas Chile" and confidence have done possible to carry out my project in Germany.

A person who motivated me to get interested in this area of study is my uncleÂ Alfonso Calvo, without his guidance and ideas; I would not have thought this issue of development.

For his support, confidence and enthusiasm, I would like to thank my Prof. Dr. Nassir Navab; and for her patience and help, Dr. Diana Mateus. I would like to show my gratitude to my colleagues at the Technische Universität München and CAMP chair, for their help and support, in every time I requested. Also, I want to thank them for the many good advices, ideas, productive discussions and joint research work.

Two friends, who were always willing to help me and they have made available their support in a number of ways, are Anabel and Asad. Thank you for all.

I cannot finish without mentioning my whole family in Chile, for their constant interest in my work.

Finally, I would like to especially thank my family, my wife Carolina and my children Esteban and Cristóbal. Without their company, support and affection, I could not fulfill my dream of studying a PhD. in Germany.

Abstract

An increasing number of computer vision applications, including medical systems, are using range imaging which require real-time and accurate 3D data acquisition. Time-of-Flight cameras allow us to acquire 3D data in real-time and at high frame-rate, however, the provided depth accuracy is in order of centimeters.

This thesis describes a novel method for acquiring depth images using multi-view Time of Flight (ToF) cameras, which uses a constructive interference between the emitted signals to enhance the depth accuracy. This work proposes to combine the measurements of the multi-view cameras at the acquisition level, in opposite to approaches that filter, calibrate or do 3D reconstructions posterior to the image acquisition. This thesis presents an example using a pair of ToF cameras in stereo-set-up defining a three-stages procedure, in which the infrared lighting of the scene is actively modified: first, the two cameras emit an infrared signal one after the other (stages 1 and 2), and then, simultaneously (stage 3). The third stage is where the constructive interference between the two cameras is produced. These redundant measurements are optimized to obtain the enhanced depth information. Based on a simulation of the ToF cameras, a quantitative evaluation of the proposed method is provided. The performance of this novel 3D acquisition method is evaluated for different objects and configurations. Results on real images using different optimization framework are also presented. The acquisition is optimized first for each camera and then within the complete 3D reconstruction process. Both simulation and real images prove that the proposed Stereo ToF camera acquisition produces more accurate depth measurements. We then extend this concept and formulation from two ToF cameras to a multi-view ToF cameras set up.

This thesis also presents two applications of Monocular Simultaneous Localization and Mapping (MonoSLAM). The first application is using MonoSLAM in endoscopic images obtaining a rough idea of the 3D scene observed by the endoscope. The proposed method applies a pre-processing step in order to enhance the texture information in the endoscopic images which normally have poor texture. The second application involves a RGB-D sensor as ToF cameras combined with a High-resolution cameras or Kinect. An extension of the traditional Monocular Simultaneous Localization and Mapping (MonoSLAM) for 3D reconstruction of the scene is also presented. The traditional Monocular SLAM (MonoSLAM) is extended in order to take advantage of the new depth data provided by the RGB-D sensors. The algorithm produces a sparse-map. The quantitative analysis shows that our proposed method improves the recovered trajectory of the camera in comparison to using only the traditional MonoSLAM (RGB). To simulate the RGB-D sensor, we use a ToF camera and an RGB camera in stereo set-up; and to define the ground-truth, a commercial optic tracking system is used. Note that this proposed method would function with any high-frame rate 3D acquisition modality including the Multi-view ToF system introduced in this thesis.

Finally, a discussion about the feasibility of each proposed method and how they can be used in real-world applications is provided. Details of the implementation and results of the multi-view ToF acquisition process are then presented, demonstrating that acquisition and processing of the proposed constructive interferences result in considerable improvement of 3D acquisition and reconstruction accuracy.

Keywords: 3D-Reconstruction, ToF Camera, Stereo-ToF, Multi-View ToF-Camera, SLAM, Endoscopic images

Zusammenfassung

Die Zahl an Computer-Vision-Anwendungen, die sowohl in der Medizin als auch in anderen Gebieten eine genaue 3D-Datengewinnung in Echtzeit benötigen, nimmt stetig zu. Time-of-Flight (ToF) Kameras ermöglichen eine solche 3D-Datengewinnung in Echtzeit und mit hohen Bildraten, aber die Messungsgenauigkeit in der Tiefe liegt nur in der Größenordnung von Zentimetern.

Diese Arbeit behandelt eine neuartige Methode zur Erfassung von Tiefenbildern mittels Multi-View ToF-Kameras, die auf konstruktiver Interferenz zwischen den ausgesendeten Signalen zur Verbesserung der Tiefengenauigkeit beruht. Es wird ein Verfahren vorgeschlagen, bei dem die Messungen der Multi-View-Kameras bereits bei der Aufnahme zusammengeführt werden, im Gegensatz zu anderen Ansätzen, bei denen erst nach der Aufnahme der Tiefendaten gefiltert, kalibriert oder in 3D rekonstruiert wird. Beispielhaft wird in dieser Arbeit eine Stereo-Anordnung von zwei ToF-Kameras vorgestellt mit einer drei-stufigen Prozedur zur aktiven Änderung der Infrarotausleuchtung der Szene: Zuerst senden beide Kameras nacheinander ein Infrarotsignal aus (Stufen 1 und 2) und dann zeitgleich (Stufe 3). In der dritten Stufe wird die konstruktive Interferenz zwischen den beiden Kameras erzeugt. Die redundante Information wird optimiert, um korrigierte Tiefendaten zu erhalten. Die Qualität der vorgestellten neuartigen 3D-Messmethode wird in einer quantitativen Analyse auf Basis einer ToF-Kamera-Simulation evaluiert. Beim Test werden verschiedene Objekte und Konfigurationen zugrunde gelegt. Ergebnisse, die auf realen Aufnahmen beruhen und verschiedenartige Optimierungsverfahren verwenden, werden ebenfalls besprochen. Die Optimierung erfolgt zunächst getrennt für jede Kamera und dann integriert im gesamten 3D-Rekonstruktionsprozess. Sowohl die Simulation als auch die realen Aufnahmen belegen, dass das vorgeschlagene Stereo-ToF-Aufnahmeverfahren eine höhere Tiefengenauigkeit bietet. Schließlich wird in dieser Arbeit der Ansatz und dessen Formulierung von zwei ToF-Kameras auf eine Multi-View-Anordnung erweitert. In dieser Doktorarbeit werden weiterhin zwei Anwendungen von Monocular Simultaneous Localization and Mapping (monoSLAM, deutsch: simultane Lokalisierung und Kartenerstellung mit einer Kamera) vorgestellt. Die erste Anwendung benutzt monoSLAM in endoskopischen Bilddaten, um eine grobe Vorstellung von der dreidimensionalen Szene zu erhalten, die vom Endoskop gesehen wird. Die Methode wendet einen Vorberechnungsschritt auf die Endoskopiebilder an, um die typischerweise schwache Texturinformation zu verstärken. Die zweite vorgestellte Anwendung benutzt einen RGB-D-Sensor, der aus einer ToF-Kamera kombiniert mit einer hochauflösenden RGB-Kamera besteht, oder aus einer Kinect. Es wird auch eine Erweiterung des traditionellen monoSLAM-Ansatzes zur 3D-Rekonstruktion der Szene vorgestellt. Hierbei wird die durch den RGB-D-Sensor neu hinzugekommene 3D-Information gewinnbringend genutzt. Ausgabe des Algorithmus ist eine Sparse-Map. Eine quantitative Analyse zeigt, dass die vorgestellte Methode die berechnete Trajektorie der Kamera im Vergleich zum traditionellen monoSLAM auf RGB-Daten verbessert. Zur Simulation der RGB-D-Daten kommt eine ToF-Kamera in Stereo-Anordnung mit einer RGB-Kamera zum Einsatz. Die Grundwahrheit wird mittels eines kommerziellen optischen Tracking-Systems erzeugt. Der Ansatz funktioniert mit jeder Art von schneller 3D-Bildmodalität, einschließlich dem zuvor in dieser Arbeit beschriebenen Multi-View-ToF-System.

Die Arbeit wird mit einer Diskussion der Eignung jeder der vorgestellten Methoden abgerundet, wobei darauf eingegangen wird, wie die Methoden in realen Anwendungen zum Einsatz kommen können. Es werden Implementierungsdetails vorgestellt und Ergebnisse der Aufnahmen mit dem Multi-View-ToF-System, die belegen, dass die Verarbeitung der konstruktiven Interferenz zu einer deutlichen Verbesserung der Genauigkeit bei der 3D-Datenerfassung und Rekonstruktion beiträgt.

Stichworte: 3D-Rekonstruktion, ToF Kamera, Stereo-ToF, Multi-View ToF-Kameras, SLAM, Endoskopiebilder

Contents

List of Figures

List of Tables

List of Algorithms

Part I.

Introduction and Theory

1. Introduction

Nowadays, the technology of three-dimensional (3D) sensing is present in different areas of our life. It can be found in the cinema, video gaming and technological applications, *i.e.*, robots and industrial machines. Nevertheless, this technology is still in a developing state and it is necessary to continue working in this direction, because it can be of great help to medical treatments, improvement of productive process and other diverse applications. At the beginning, the researchers tried to obtain the 3D scene from a monocular camera or a stereo camera (pair of cameras). Both provide plane images without the 3D information of the scene. They normally use motion of the camera and matching between points in the images, to estimate the 3D information. For this reason, at the beginning, this work was focused to obtain a 3D reconstruction from two-dimensional endoscopic images. This challenge involves the matching of image points while the endoscope is moving. The matching task in endoscopic images is more difficult compared to a normal scene (*e.g.*office, room), due to the reduced texture observed in the endoscopic sequence. Normally, the stomach and esophagus tissues have a pseudo-continuous color with some vessels producing texture. To make a 3D reconstruction from a monocular camera, usually is employed an algorithm named Monocular Simultaneous Localization and Mapping (MonoSLAM). This algorithm makes a scene reconstruction from a moving camera and a localization of the camera in this reconstructed scene, simultaneously. This thesis applied Monocular Simultaneous Localization and Mapping (MonoSLAM) to endoscopic sequence and shows a rough three-dimensional (3D) reconstruction of stomach and esophagus from a two-dimensional (2D) endoscopic sequence, which is explained in Chapter 4.

But now, the availability of a real-time 3D sensor, *i.e.*, a sensor which can provide the 3D information directly at a fast frame-rate, was of a great impact in the computer vision and robotics community. Actually, real-time 3D imaging has been included in several applications, allowing an observation of the 3D shape, of objects and scenes. This 3D information has helped to solve many computer vision problems simpler than using only a color image. A 3D sensor at a fast frame-rate helps the incorporation of the depth information in a lot of real-time applications. Specifically, the integration of real-time 3D data helps us, not only to create a 3D reconstruction of the scene in real-time, but also to recognize objects from their shape at fast frame-rate. With the availability of a fast-frame 3D sensor, we realize that instead of trying to obtain 3D information from a 2-D sensor, we can use a 3-D sensor, directly in the endoscope. Subsequently, we reviewed literature and we found a group, who were developing a 3-D endoscope using Time-of-Flight camera [92]. Then, the focus of the thesis changed to the creation of an algorithm to make a 3D reconstruction from a Time-of-Flight camera joint with a high-resolution camera (simulation of Time-of-Flight endoscope). For that, it was necessary to study the state of the art of the Time-of-Flight camera, in order to learn its working principles and its applications in computer graphics. This state of the art is presented in Chapter 3. Afterwards, an extension of the traditional MonoSLAM to be used

with the new generation of RGB-D (Color 2D image plus depth) sensor (*i.e.*, ToF endoscope), was developed. The SLAM using RGB-D sensor was made in general and has many applications in robotics and computer vision. Furthermore, this work gives a 3D reconstruction of the observed scene and the camera trajectory. The extension of MonoSLAM including its results are explained in detail in Chapter 5.

However, the 3D information quality of real-time devices are worse than the non-real-time scanners (*e.g.*, laser scanner), *i.e.*, the depth accuracy is lower compared to the precision of a laser scanner. This quality is important, because it reduces the use of the sensor in some applications, which requires a high precision. After using the ToF cameras and knowing its working principles, a very low depth accuracy was observed, and for endoscopy, is desired a high depth accuracy. Then, the focus of the thesis changed to improve this depth accuracy of the ToF 3D sensor. Therefore, a novel acquisition process for a Multi-view Time-of-Flight (multi-ToF) system was developed. This proposed method defines a stage-process to capture redundant depth measurements, which are optimized allowing an improvement of the depth accuracy. Depending on the number of cameras used in the Multi-view system, it defines the necessary number of stages. An example of two ToF cameras (Stereo-ToF) in stereo set-up has been utilized to prove the feasibility of the proposed method. Three stages are defined, where the illumination unit is manipulated in order to produce a constructive interference between the ToF cameras and finally, to improve the accuracy of each camera depth measurement. This interference is achieved using the same modulation frequency in all the ToF cameras. The explanation of the proposed method, results and limitations are shown in Chapter 6.

But before going into details of the methods developed throughout this thesis, we will first focus on a motivational introduction, contributions and theory introduction. Following, in section 1.1 and 1.2 are shown the motivation and the contributions of this thesis respectively. An introduction to basic optics, an introduction to multi-view geometry and an introduction to 3D imaging technologies are shown in section 2.1, 2.2 and 2.3 of Chapter 2.

1.1. Motivation

Range imaging has been used for a long time in robotics, computer vision and industrial application; specially the well-known laser scanner, due to its accurate 3D data. Laser scanner generates a 3D surface with a very high depth accuracy, however, this device is slow and could not be included in some applications, which requires a real-time 3D acquisition. Nowadays, real-time range imaging has been developed and included in many computer vision and robotic applications, due to its depth information available in a fast frame-rate. This acquisition speed allows its incorporation into some applications, for example, in real-time 3D reconstructions, simultaneous localization and mapping, dynamic scene analysis, touch-free interfaces, gaming control, 3D modeling, etc.

Besides, the computer vision community has reported an improvement of robustness and accuracy in complex tasks, thanks to the available depth data. For example, using only 2D-color images, the segmentation task is hard and it needs a very costly computation. Instead the foreground can be simply recognized from the background using depth data, due to the fact that objects are normally in front of the background. Another example is the tracking of objects. Normally, tracking where only is used color images, it will depend on the texture of the object, instead, the depth informa-

tion allows the tracking of the non-textured objects using its object shape. Sadly, using only depth information, it is impossible to recognize two objects with a similar shape but different texture. Therefore, if the color image and depth image are combined, it is possible to track using, not only the color texture, but also the shape of the object. This combination helps to obtain better results in more complex scenes. Moreover, gesture recognition and touch-free interfaces can be attempted using depth data, helping to easily observe and recognize the motion of the hands, arms, face and whole/part of the body. Again, if the color and depth data are mixed, a more robust and accurate gesture recognition can be achieved. Additionally, Simultaneous Localization and Mapping has been applied to 3D data from laser scanner. Normally, this laser scanner is used to capture the 3D data of big scenes, *e.g*.tunnels, mines, buildings, etc. This scanning has to be made with a slow motion even sometimes static, to reduce the motion artifacts due to its slow scanning. The use of real-time range imaging allows a fast scanning in mobile vehicles (*i.e*.robots), without motion artifacts and can be used joint with real-time 3D reconstruction algorithms to achieve 3D modeling of the observed scene.

Nevertheless, the real-time 3D sensing devices are less accurate comparing to laser scanners. Time-of-Flight cameras provide the observed surfaces in a high frame-rate and has been used in different applications. Unfortunately, the ToF camera has a poor depth accuracy and sometimes cannot be used in some applications. For this reason, a precise real-time range imaging device is very interesting for the computer vision and robotic community.

1.2. Contributions

In the course of this thesis, several contributions have been made to the computer vision community.

- **Time-of-Flight camera Theory.** A new point of view for the explanation of the ToF camera working principles is shown, including a mathematical formulation and its solution. The formulation is presented from a new perspective, obtaining a more general and a better understanding of the working principles equations. This explanation is based on Lange's PhD thesis [64]. This work is shown in Chapter 3 including a summary of ToF applications in computer vision which is an updated state-of-the-art based on the article of Kolb *et al*. [62].

- **Monocular Simultaneous Localization and Mapping (MonoSLAM) in endoscopic images.** The MonoSLAM algorithm aims to reconstruct the 3D scene observed by a monocular camera, while this camera is localized in the reconstructed scene simultaneously. This algorithm has been applied to endoscopic sequences, to reconstruct part of the patient stomach and esophagus. However, for the reconstruction of the esophagus surface from endoscopic videos, several challenges need to be addressed such as less-textured images. The proposed algorithm, pre-process the endoscopic images in order to enhance its texture, de-interlacing the images and highlighting the edges in the images. Chapter 4 presents the proposed algorithm and the addressed challenges in endoscopic video.

- **Simultaneous Localization and Mapping (SLAM) using RGB-D sensor.** This work describes an extension for the Monocular Simultaneous Localization and Mapping (MonoSLAM)

[20] method. This extension relies on the images provided by RGB-D sensor as a combination of a high resolution and a Time of Flight (HR-ToF) sensor or Kinect. The proposed method incorporates the depth data into the MonoSLAM modeling, showing an improvement in the accuracy of the recovered camera trajectory and a reduced map of uncertainty. Chapter 5 introduces the developed extension of the traditional MonoSLAM. It is also proposed and discussed the possible solutions to improve the robustness of the presented method.

- **Multi-view Time-of-Flight acquisition.** This research proposes a novel acquisition process for Multi-view ToF camera system, which makes redundant measurements and uses a constructive interference in order to, not only reduce the noise in the ToF camera depth map, but also to enhance the surface details. Chapter 6 presents a detailed explanation of this acquisition process using as an example, a pair of ToF cameras, and its extension to a multi-view ToF. Furthermore, this chapter presents a quantitative evaluation of the results based on a ToF camera simulation. Besides, this Chapter includes the limitations and the theoretical proof of the proposed method. Additionally, it is included the hardware modification necessary to make the cameras to work together; and the electronic circuit which was built, to make the real experiments.

- **Measurement of attenuation in the ToF camera.** Additionally, this work has experimentally measured the infrared light attenuation in a ToF camera in order to prove the existing models of attenuation which are exhibited in Chapter C). This attenuation has been used in the ToF simulator utilized to quantify the improvement of the proposed Multi-view ToF acquisition process.

2. Theory

THis chapter introduces basic ideas of optic physics, multi-view geometry and 3D imaging technologies. This theory introduction is used in the explanation of the algorithms and methods presented in this thesis. This explanation is an important introduction to a better understanding of these proposed ideas.

2.1. Basic Optics

The behavior and properties of light can only be explained using the wave-particle duality theory of contemporary physics, which says that light behaves as a particle and a wave at the same time. Light has wave-like properties such as diffraction and interference; besides, it has particle-like properties as reflection and refraction. Light is an electromagnetic radiation that can be visible or invisible to the human eye, and it is responsible for the sense of sight. Light is composed of the elementary particle named *photon*. Primary properties of light are intensity, propagation direction, frequency or wavelength spectrum, and polarization [29]. While its speed is constant in a material, for example, in vacuum is $3 \cdot 10^8$ meters per second which is similar to air speed of light. These concepts and properties are used in the explanation of the ToF camera theory, due to the use of infrared light.

2.1.1. Reflection

Reflection explains the effect of light when it touches a surface of an object or scene, and it can be divided into two types: specular reflection and diffuse reflection. Specular reflection describes a group of surfaces which reflects light in a simple direction (for example a mirror); and it follows the Fresnel equations [29]. The diffuse reflection describes a group of surfaces which are opaque and non-limpid materials such as a rock; and it can be modeled in different mathematical formulas; nevertheless, this thesis is focused in the Lambertian model. Most of the materials exhibit a combination of specular and diffuse reflection. Furthermore, not all light is necessarily reflected, some of the light can be absorbed by the material or propagated through the material or surface. The relation between the incident power and reflected power is named *reflectivity* which depends on the material characteristics. That means, reflective and refractive index of the material respect to the light wavelength.

Specular Reflection

Specular reflection describes the reflection on flat and continuous surfaces, *e.g.*a mirror. This reflection follows the law of reflection which says that the reflected ray is determined by the angle

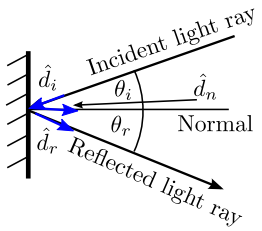


Figure 2.1.: Specular reflection example. An incident light ray with angle θ_i and direction $\hat{d}_i$ is reflected with a reflection angle θ_r and direction $\hat{d}_r$ in a surface with a normal direction $\hat{d}_n$.

between the incident ray and the surface normal at the incident point. The surface normal is a perpendicular vector to the tangent plane to the surface. In the Figure 2.1 can be observed, that the reflected light has the same angle and in the same plane as the incident ray with respect to the surface normal θ. In vectorial formulas we have that the direction of the reflected ray can be calculated as:

$$\hat{d}_r = 2\left(\hat{d}_n \cdot \hat{d}_i\right)\hat{d}_n - \hat{d}_i \tag{2.1}$$

where $\hat{d}_r$ is a unit vector of the reflected ray, $\hat{d}_i$ is a unit vector of the incident ray, $\hat{d}_n$ is a unit vector of the surface normal and $\hat{d}_n \cdot \hat{d}_i$ is the dot product between incident ray and surface normal. The quantity of reflected light compared to incident light depends on the material reflectivity and, in some cases, also relies upon the angle of incidence, hinging on the refraction index of the propagation medium and material.

Diffuse Reflection

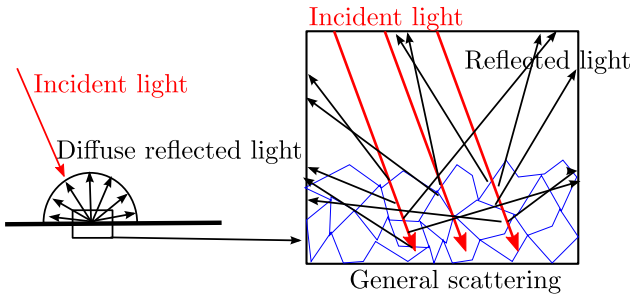


Figure 2.2.: Lambertian diffuse reflection. The incident light suffer scattering in the different molecular layers, producing reflection in any directions.

Diffuse reflection describes the effect of one incident ray, which strikes on a material and it is reflected in several angles; and not only in one direction, as the case of specular reflection. All these reflection angles generate a feeling of opaque material. The ideal diffuse reflection, reflects the same quantity of light in all directions in the same hemisphere of the reflecting surface. This model

is named *Lambertian reflectance*. This diffuse effect is produced by the irregularity of the reflecting surface and scattering of light which passes through the first and second molecular layers, being reflected again and over again; producing a reflection in random directions; and therefore a diffuse reflection. Figure 2.2 shows an example of diffuse reflection. One of the most used model for diffuse reflection is the *Lambertian* model, due to its simplicity and follows the equation 2.2.

$$I_r = \left(\hat{d}_n \cdot \hat{d}_i\right) I_i \tag{2.2}$$

where I_r is the reflected intensity, I_i is the incident intensity, d_n is the direction of the normal and d_i is the incident direction of incident light.

2.1.2. Refraction

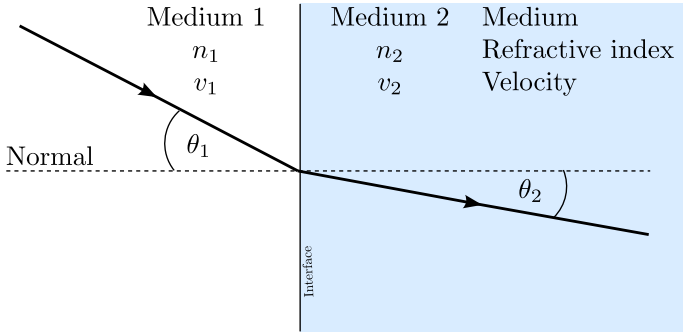


Figure 2.3.: Refraction of light between two mediums (1 and 2) with velocities v_1 and v_2 and refractive indexes n_1 and n_2, respectively.

The change of light direction due to a switch of its speed, is named refraction, and normally is produced by a variation of medium, while the wave is traveling. For example, when the light goes through a window, it is refracted when it interchanges from air to window, because of its different speeds. It also happens when the light moves from air to water. The refraction is described by Snell's law, which says that, if light changes from medium 1 to medium 2, the incident angle θ_1 is related to refraction angle θ_2 as follows:

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{v_1}{v_2} = \frac{n_1}{n_2} \tag{2.3}$$

where v_1 and v_2 are speeds of light in medium 1 and 2 respectively, and n_1 and n_2 are refractive indexes. In the case of light, normally in a camera, the refraction does not play an important role. However, if we want to use a camera under water, then the refraction will influence a lot in the acquired images. An example of the refraction effect in a wave that change from medium 1 to 2 is shown in Figure 2.3.

2.1.3. Scattering

Inter-reflection between molecules and irregular surfaces is named *scattering* and it is one of the causes of diffuse reflection (Figure 2.4 as example). Scattering is produced when the light goes into

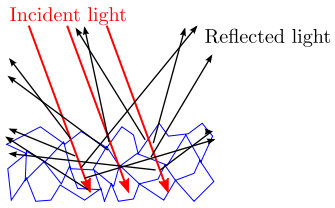


Figure 2.4.: General scattering of light in an opaque surface.

an object, and relying on the spaces between the molecules, the light is inter-reflected. A part of the light can be reflected and the other part can cross through the object. A typical example of scattering is when we illuminate a finger, very near to the light source, where we can observe that the emitted light can cross to the other side of the finger producing the effect of illumination of all the finger and not only as a point of light. This effect is produced only in objects where light can cross and cannot be completely absorbed. For example, if a foot is illuminated, the scattered light cannot cross due to big wide of the foot. Another example of scattering is when the light is crossing smoke, we can see that in the other side, the light is observed in the smoke in a bigger area compared to the source light point.

2.1.4. Absorption

Part of the light which hits an object is absorbed by the surface. The absorbed flux of light is normally transformed to other form of energy, such as heat. Normally, the absorption of light when is traveling on the air is known as light *attenuation*. The absorption coefficient on the object is also known as *reflectivity*, which tell us how much of the incident light is absorbed by the object and how much it is reflected. Besides, the absorption is partially due to the scattering in the object, "losing" part of the incident light.

2.1.5. Polarization

Polarization is a property of certain waves, *e.g.*light, which describes the orientation of the wave oscillation. By convention, the polarization of light is defined by the orientation of electric field of the electromagnetic wave in the space in one oscillation period. The light polarization in free-space is usually perpendicular to the direction of motion, which is also named linear polarization. It is possible to generate light with circular or elliptical polarization, which means a rotation of the electric field when it is generated. Besides, the change of amplitude in each electrical and magnetic field independently creates a circle or ellipse, producing the circular or elliptical polarization.

2.1.6. Diffraction

Diffraction of light corresponds to the apparent curving of light around small obstacles and the dispersion of light, which passes through small openings. This effect produces that light invades the dark portion of the matter when crosses small holes, and that light propagation blends when travel near to small obstacles. The diffraction can be observed when we make a photograph directly

Figure 2.5.: Diffraction produced by sunlight passing through the diaphragm of the camera

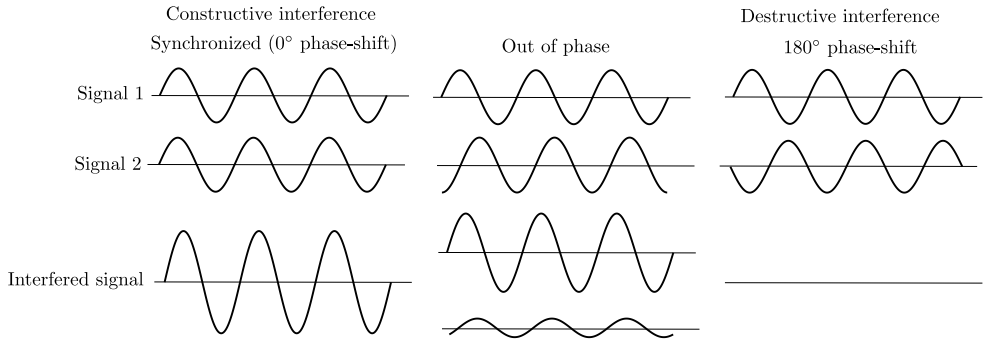


Figure 2.6.: Possible interferences in general waves. Left: Constructive interference ($0°$ phase-shift). Middle: Out of phase, it can occur additive interference or the signal can be partially destructed. Right: Destructive interference, $180°$ phase-shift, signal completely destructed.

to a light source as the sun or a laser pointer. The Figure 2.5 shows an example of a sun picture produced by the diffraction of the sunlight passing through the camera diaphragm.

2.1.7. Interference

Interference is the effect of waves superposing in the same space. In the case of light, the superposing occurs in absence of non-linear effects. Interference can be explained with the superposing of waves with different frequencies and/or phases. As example, Figure 2.6 shows the effect of two sinusoidal waves with the same frequency, but with a different phase-shift. The case of a phase-shift of $0°$ degrees, the crests of the original waves are aligned, then the outcome sinusoidal is the sum of the original signals, raising the amplitude. This can also happen with two signals with small phase-shift. This case is named *constructive interference*. On the other hand, if the waves are out-of-phase and a crest of an original wave and a dip of the other original wave are aligned, then the outcome sinusoidal wave has an attenuated amplitude. The case of $180°$ degrees produces a total destruction of the original waves. This case is named *destructive interference*. The constructive interference is the basis of the proposed method "Multi-view ToF acquisition", although this interference is produced in the modulated wave and not in the light wave itself.

2.2. Multi-view Geometry

In this section a basic multi-view geometry introduction, utilized in this thesis, is exhibited. The multi-view geometry is a known field by the computer vision community and it gives us tools for manipulation and modeling of multi-view systems, such as camera calibration and 3D transformations. This section is focused in the understanding of camera calibration, 3D transformations and triangulation which were used in the developing of this doctoral work.

2.2.1. Camera Calibration

A camera is normally modeled as a *pin-hole camera* joint to a distortion model for the lens. This pin-hole camera model assumes that the camera is a simple camera, without lens and with a single small aperture. The camera model has its own coordinate system in its center point. The camera can be described using the intrinsic parameters, which defines two focal lengths and a principal point. Where *Focal length* describes how strongly the light converges (focuses) to the sensor and its defined for each image coordinate, *i.e.*, X and Y. The *principal point* represents the alignment of the sensor center and the camera model. These parameters define a transformation from the image plane coordinate system (U,V) to camera coordinate system (X,Y,Z). An image position (u,v) can define a ray which crosses the image position (u,v) to the infinity. This ray represents the captured light by the sensor pixel, thus the 3D point lies in this ray. The variable related to the axis Z is defined by the depth of the object captured in the image. Therefore, in a normal camera, it is not possible to determine the depth of the object, unless we use a multi-view system to intersect the observed object on different views and in order to estimate the depth via triangulation. This camera model is illustrated in Figure 2.7, showing the focal length, the principal point and a ray of an image point (u,v). The intrinsic parameters are mathematically represented as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.4}$$

where $K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$ is the matrix of intrinsic parameters, with f_x the focal length in the axis X, f_y in the axis Y, (c_x,c_y) is the principal point in the image plane (intersection between the axis Z and the image plane). These intrinsic parameters represent the mathematical model of the camera. These parameters depend on the sensor size, and distance between lens and sensor (named *focus*).

The lens distortion is produced by the deviation of light, from a rectilinear trajectory due to the curvature of the lens glass. In this thesis, the lens distortion is modeled as a radial and tangential distortion. Such model says that the image is deformed depending on the distance from the center of the image, which is typically the behavior in photographic lenses. This distortion is mathematically modeled as shown in equations 2.5 and 2.6. Figure 2.8 shows graphically an example of a radial and tangential distortion used in computer vision.

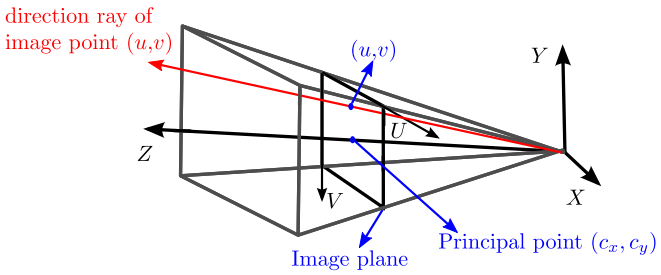


Figure 2.7.: The common camera model. Camera coordinate system (X,Y,Z) defined by the focal length f_x in the axis X, f_y in the axis Y and the principal point (c_x,c_y) in the image plane.

$$x'' = x' \cdot (1 + k_1 r^2 + k_2 r^4) + 2p_1 x' y' + p_2(r^2 + 2x'^2) \tag{2.5}$$

$$y'' = y' \cdot (1 + k_1 r^2 + k_2 r^4) + p_1(r^2 + 2y'^2) + 2p_2 x' y' \tag{2.6}$$

where $x' = \frac{x}{z}$, $y' = \frac{y}{z}$, $r = \sqrt{x'^2 + y'^2}$, k_1, k_2 are radial distortion coefficients, and p_1, p_2 are tangential distortion coefficients.

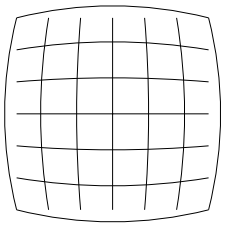


Figure 2.8.: Radial distortion example. The lines are originally straight and after the distortion, the lines are bent.

These parameters are estimated through standard camera calibration, known by the computer vision community. For further details about camera model, please read the book of Multi-view Geometry [5].

2.2.2. 3D Transformation

In this subsection, a typical euclidean 3D transformation is reviewed. A transformation describes how to move the representation of a 3D point from a coordinate system to another. For that, the 3D transformation is separated in two complementary transformations: *translation* and *rotation*; which together can move any 3D point described by a coordinate system to another. A three-dimensional coordinate system has six degrees of freedom (6 DOF), in other words, any coordinate system in the 3D space can be represented by a translation (3 DOF) and a rotation (3 DOF) of an original fixed coordinate system. In this thesis, such transformation is used to modify a 3D point representation from a camera coordinate system to another camera.

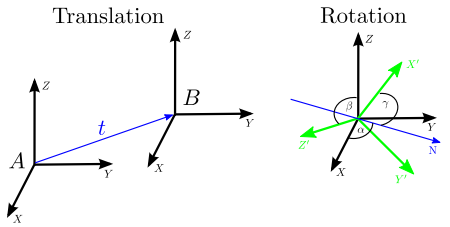


Figure 2.9.: Translation from A to B and rotation in 3 degree of freedom of a coordinate system

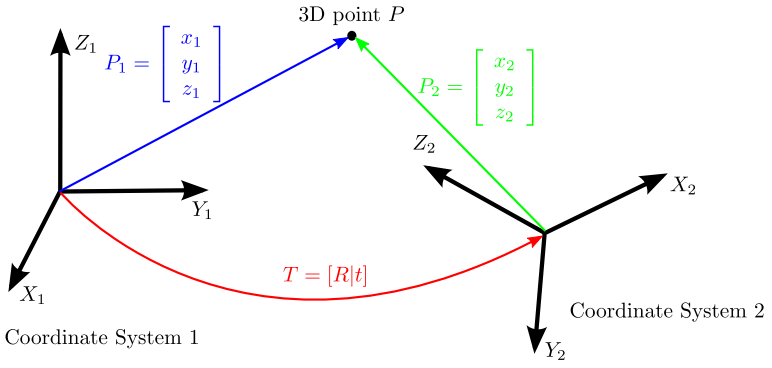


Figure 2.10.: Transformation of point P from coordinate system 1 to coordinate system 2.

- **Translation:** It describes the fact of moving a coordinate system from a point A to a point B in a 3D space. It is represented by three values in a vector $t = (x, y, z)^\intercal$. See the graphical representation in Figure 2.9.

- **Rotation:** It describes the rotation of the axis in the three possible directions. The graphical representation can be observed in Figure 2.9. This rotation can be described by the *Euler angles* (α, β, γ) or other representations, *e.g.quaternions* (used in Simultaneous Localization and Mapping - Chapters 4 and 5). This rotation can be mathematically represented by a matrix as shown in Equation 2.7.

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.7}$$

where the matrix values $r_{ii}, i \in 1, \cdots, 3$ depends on the rotation angles (*i.e.*Euler angles). The rotation matrices are orthogonal matrices with derminant 1 ($R^\intercal R = I$ and $\det(R) = 1$). In general, the euclidean transformation of a 3D point $P_1 = [x_1, y_1, z_1]^\intercal$ from a coordinate system 1 to a coordinate system 2 can be expressed in a matrix representation as follows (Equation 2.8):

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = R \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + t = [R|t] \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = T \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} \tag{2.8}$$

where $P_2 = [x_2, y_2, z_2]^\intercal$ is the 3D point P described by the coordinate system 2, T transformation matrix, R rotation matrix and t translation vector. The last equation was formulated using homogeneus coordinates. The transformation produced by equation 2.8 can be observed in the Figure 2.10.

Transformations are used in this thesis in order to change the representation of the 3D point from one camera coordinate system to another camera. For further details, please read the book of Multi-view Geometry [5].

2.2.3. Triangulation

Triangulation is a technique which calculates the depth of an object observed from images of two cameras with a known transformation between them. In other words, the cameras have a distance between them which allows us to estimate the depth thanks to the observed displacement of the object in the images. Practically, this displacement is estimated throughout the matching of the same object points in the two image planes, and the distance between the cameras is calculated using the transformation between camera coordinate systems. The matrix F is named *Fundamental matrix*, which relates corresponding points in stereo images. Let a 3D point P which has its correspondent projected image points p and p' in two images and satisfy $p'Fp = 0$, where F is the fundamental matrix between cameras. This constraint tells that the two rays are coplanar and therefore they will intersect in one 3D point P. This procedure is named triangulation and is illustrated in Figure 2.11.

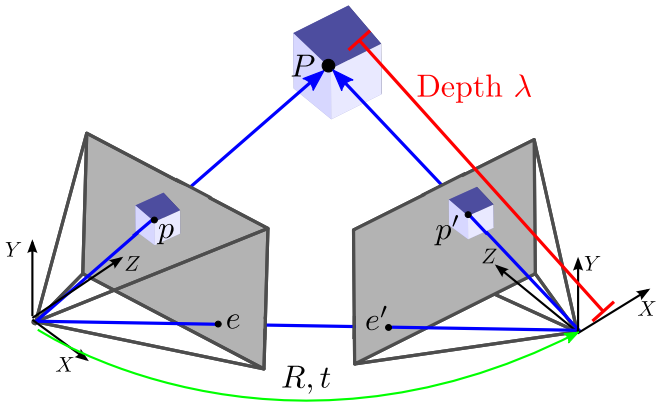


Figure 2.11.: Triangulation principle. The transformation R,t is known. The point P is observed by two cameras and they are projected to the image planes in p and p' image position respectively.

In order to estimate the distance between the object and cameras, we have to find the 3D point P

that satisfies next Equation:

$$\hat{p} = P_1 \hat{P} \qquad\qquad \hat{p}' = P_2 \hat{P} \tag{2.9}$$

where P_1 and P_2 are projection matrices of camera 1 and 2, composed of its intrinsic parameters and transformation between cameras. We can estimate the value of P through the optimized 3D point $\hat{P}$ and the optimized image points $\hat{p}$ and $\hat{p}'$, where p and p' cannot be coplanar, but it is assumed that they have to intersect in a single point (the same 3D point projected to the image planes). In this case $\hat{p}$ and $\hat{p}'$ have to be similar to measurements p and p'. Minimizing these constrained equations, the value of the 3D point P can be estimated, and therefore the distance from the 3D point to cameras.

For further details on triangulation, please read the book of Multi-view Geometry [5].

2.3. 3D Imaging

Three-dimensional imaging represents a group of devices which can generate 3D information from the observed scene. Until now, there are different types of 3D imaging, including stereo vision, shape-from-shading, shape-from-focus, algorithms based on sillhouttes, photometric stereoptics, laser scanner, Time-of-Flight camera, structured light camera and others. This thesis is focused in four types of 3D imaging: laser scanner, stereo vision, Time-of-Flight camera and structured light camera (*i.e.*Kinect). 3D scanning can be active or passive depending on the necessity of its light emission. Active sensors use the help of the active light emission to measure depth distances. On the other hand, passive sensors only use normal light of the observed scene to estimate depth data.

In general, there are two methods to measure the depth between a sensor and a scene. These methods can be used with different technologies, *i.e.*, laser, ultrasound. The methods are shown and detailed in the following paragraphs:

- **Time-of-Flight:** The first method is an **active** sensing which measures the *Time-of-Flight* taken by an emitted signal to travel from the sensor to the scene and return. It is possible to use different wavelengths of light, such as, visible light, ultra-violet, infra-red, etc, or different type of wave, *i.e.*ultrasound. Furthermore, this method can be used for a single point or a matrix of point (1-D or 2-D). Then, the distance between the object and the sensor is related to the measured time and medium speed (air, water, etc.).

- **Triangulation:** Given two sensors or two emitters with a known distance (baseline) between them, it is possible to calculate the depth using triangulation. This means that the rays of the same sampled point in the two sensors or emitted signals intersects in one single point. The known 3D transformation between the sensors and the correspondences among the images are used to determine the depth via triangulation. These devices can be **active** or **passive**.

In general, Time-of-Flight scanners are noisier than triangulation scanners due to the electronic difficulty to measure very low times, as the case of Time-of-Flight scanner which has to measure time in order of nano-seconds. Therefore, the accuracy in triangulation systems is better than time-of-flight systems. On the other hand, triangulation systems need to calculate the triangulation in every point and it depends on the texture or pattern projected. This triangulation requires a complicated calculation for every point, making a slower system compared to Time-of-Flight systems.

Furthermore, triangulation systems have limited range of work which depends on the visibility and quality of the matching, necessary for the triangulation.

In the following subsections, an overview of different 3D imaging technologies are described.

2.3.1. Stereo Vision

Stereo vision is a passive scanning system which is based on the triangulation of two calibrated optical cameras. These two cameras observe the same target object. Detecting correspondences between the camera images is possible to estimate the depth information via the triangulation. Having a good quality of correspondences between the images, it is possible to get a good depth estimation. One computer vision challenge is to find these correspondences between the images in a robust and exact manner. However, the correspondences depend on the object texture, being very difficult to be robust to rotation and changing of scale. Also, the resolution of the depth map is dependent on the number of found correspondences which is also related to the observed object texture. Normally is not possible to estimate the depth in the whole image pixels and therefore the depth map is usually not complete (*i.e.*image areas without depth information).

2.3.2. Laser Scanner (LIDAR)

Light Detection And Ranging (LIDAR) is an active scanner which can work with ultraviolet, visible or near-infrared light. The emitted light allows to create a 3D image of the objects using a laser beamer and an optical device. In general, the laser scanner takes the advantage of some basic attributes of laser light: light travels in straight line; the speed of light is known; and the laser typically produces only one wavelength, making easier its detection. There are two types of laser scanner: time-of-flight based and triangulation based. The first type of laser scanner has a laser beamer which emits a laser pulse to measure the time of flight used by light, to go from the sensor to the target object and then coming back. The second type of laser scanner has two laser beamers and an intersection between the emitted light on the target object is detected. This detection is triangulated using the known position of the emitters and the necessary angle of the beamers to produce the intersection. Sometimes, this second beamer is replaced by an optical camera in order to observe the position of the laser point in the object. The laser scanner usually measures only one object point per measurement and it can be moved in a plane, to create a 2D scan. This 2D scan is a line of depth in a certain plane. If we acquire a lot of 2D scan of different parallel lines consequentially, then we can generate a 3D scan. Due to the motion of the laser beamer in the 3D scan, the scanning takes a time which is usually bigger than the necessary for a real-time acquisition. Furthermore, if the scanned subject moves, its 3D scan can exhibit motion artifacts.

2.3.3. Time-of-Flight Camera

The Time-of-flight camera is an active sensor which emits an infrared light signal into the scene to measure the time that the light takes in going and coming back to the sensor. These devices use a special camera sensor which allows to acquire the required samples to calculate the depth in every pixel point. The ToF camera has a low resolution, but can work in a high frame-rate. Also, the ToF camera can measure the depth of the scene thanks to the optical physics of the reflected light, and

thus there is no geometric calculation. Its accuracy depends on the distance and it is in order of centimeters. For further details, read the detailed explanation of ToF camera working principles in Chapter 3.

2.3.4. Structured Light Camera (Kinect)

The structured light is an active 3D scanner which has a camera and a projector, with a known transformation between them (calibrated). The projector projects into the scene a known pattern, allowing the camera to recognize each pattern point and therefore to estimate the depth of each recognized point via triangulation. This pattern is typically composed of lines or ellipses which has to be known and calibrated in advance, to be used for the recognition of pattern points and calculation of depth. The size of the ellipses and the separation between the lines is totally dependent on the distance which is wanted to measure, and thus it limits the distance range of work. However, it is possible to project different patterns with distinct sizes, allowing to measure different distance ranges. An example of the projection of different patterns is Kinect that projects three different patterns of ellipses to measure three different distance ranges. Notice that the different patterns estimate the distances with distinct accuracies. Sometimes, the change of pattern used to calculate the depth, can produce a jumping of depth accuracy depending on the transition of the utilized pattern. Furthermore, Kinect has a high frame-rate and it can be used in real-time applications.

2.3.5. Comparison between 3D imaging technologies

Every 3D imaging technology has both advantages and disadvantages. In general, the devices which use triangulation and time-of-flight basis have some differences independently of the technology used. These differences are summarized in the next Table 2.1:

Device	**Speed**	**Accuracy**	**Distance Range**	**Example**
Triangulation based	Slow	High	Short	Kinect, Stereo vision, Laser scanner
Time-of-Flight based	Fast	Low	Large	ToF camera, Laser scanner

Table 2.1.: Differences between technologies which uses triangulation or time-of-flight measurement basis

In general, triangulation systems are slower compared to time-of-flight systems, nevertheless, triangulation systems are more accurate in the distance measurement. Moreover, triangulation systems cannot measure the occluded and non-projected points observed by the optical sensor instead of Time-of-Flight systems which generate a complete depth map. Furthermore, ToF systems cannot measure edges of the object correctly due to the multi-path of the reflected light or absence of light (non-reflection) captured by the sensor.

In the following, four typical devices mostly used in computer vision community as 3D imaging device are compared, showing their advantages and disadvantages in comparison to the other technologies. The main differences between these four devices are summarized in the Table 2.2.

Device	Frame rate	Resolution	Depth accuracy	Type	Price
Laser Scanner (LIDAR)	0.5 - 2 fps	Very High	mm	Active	$> 20.000 €$
Stereo Vision	1 - 10 fps	High	mm to m (it depends)	Passive	$\approx 500 €$ - $5.000 €$
Time-of-Flight camera	20 - 50 fps	Low ($\approx 200 \times 200$)	cm	Active	≈ 6.000 €
Structured light camera (Kinect)	10 - 30 fps	Medium ($\approx 640 \times 480$)	between mm and cm	Active	$\approx 150 €$

Table 2.2.: Summary of differences between four typical 3D imaging devices. Some example devices have been used as reference.

Laser scanners are very accurate devices, however, their frame-rate and price are the most important impediment for their massive utilization. They normally have a low frame-rate and high price. Usually, laser scanners are used to capture 3D data as ground truth or training data-set. Stereo vision is a very developed device which has a lower accuracy compared to laser scanners, having a texture-dependence, *i.e.*, the depth calculation depends on the matching between the correspondences in the images. This produces a depth map, which depends on the texture of the observed scene and sometimes it is not possible to create a complete 3D surface exhibiting area without depth information.

Time-of-flight camera and Kinect give an answer for real-time 3D data acquisition. Both devices have real-time 3D data acquisition, allowing to use them in real-time applications. Time-of-Flight cameras have a similar frame rate as Kinect, being in general, slightly faster than Kinect. Kinect has higher depth accuracy compared to ToF cameras, nonetheless, ToF cameras have bigger distance range of work. Kinect projects three different pattern sizes into the scene in order to measure three different ranges of distance. This produces a small "jumping" of depth accuracy, depending on the pattern size which was used to estimate the depth. On the other hand, ToF cameras cannot recover correctly edges and corners, instead that Kinect can produce a good estimation. Furthermore, Kinect sensor cannot measure the depth in the points not illuminated with the projector, but observed by the sensor, producing a black band or shadow without depth estimation. Another difference is the image resolution, ToF cameras, until now, have low resolution (order of 200×200 pixels) instead of Kinect, which produces medium resolution depth images (640×480 pixels). Additionally, ToF cameras can work in multiple-view systems changing its modulation frequency and Kinect is still not known. In terms of cost, Kinect is a cheaper device in comparison to a ToF camera.

Finally, both devices have advantages and disadvantages, and therefore the adequate device selection for an application depends on not only the restrictions of the application (*e.g.*resolution, depth accuracy), but also the budget of the project.

Part II.

State of the Art

3. State of the Art of Time-of-Flight Camera

THis Chapter presents the state of the art of the Time-of-Flight (ToF) camera. It incorporates a ToF camera theory description, which explains, not only the working principles, but also its electronic requirements and its behavior on real sensors. Furthermore, a description of the errors and limitations of the ToF cameras are shown. Finally, a list of the different applications in computer graphics is exhibited, which is classified by the distinct challenges in the computer vision community.

3.1. Introduction

Time of flight is the technique to recover the distance between a sensor, and an object or scene with the help of the time-of-flight required by the wave, going to and coming back from the scene. This idea was used in several distance sensors (*e.g.*ultrasound, laser, radar, etc.). The general idea is to measure the time-of-flight, which is directly related to the distance between the sensor and the object due to the assumed constant speed. Generally for this purpose, electromagnetic wave, light or ultrasound are used because they have a constant velocity in the same medium. The measured time by the sensor, is used to calculate the real distance, due to its constant speed which shows the traveled distance in a time unit. Of course, the type of source used determines the precision of the measured distance and the electronic requirements. This chapter is focused, not only on the ToF camera theory and description, but also on the ToF camera applications in the computer vision community.

3.2. Time-of-Flight Camera Theory

The Time-of-Fight (ToF) camera is a CCD/CMOS based device that captures reflected infrared light which allows to measure the time-of-flight; and therefore the distance in each image pixel. This camera gives a distance for every image pixel, creating a depth image which represents the observed surface. This distance is calculated thanks to the time-of-flight obtained from the measured phase-shift of a sinusoidal signal, which is an intensity modulated infrared light. This phase-shift is the time-of-flight taken from the infrared light while going to the object, and coming back to the sensor.

3.2.1. Types of Time-of-Flight cameras

The time-of-flight can be measured using a pulse wave (PW), a continuous wave (CW) or a combination of both. The difference is shown in the Figure 3.1, the pulse wave is based on sending only one pulse and then measuring the delay time between the sent and received signal which is

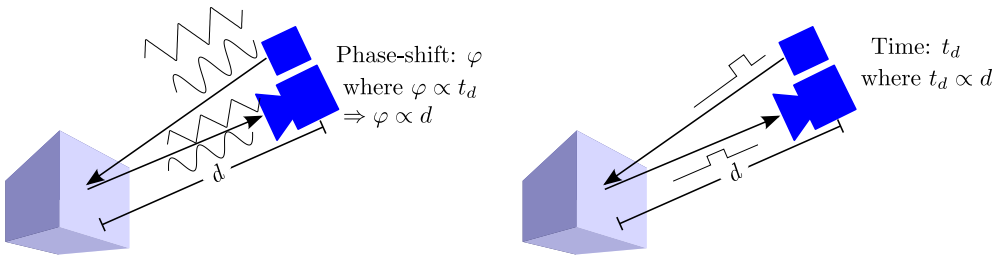


Figure 3.1.: Example of Time-of-flight working. Left: Continuous Wave based ToF. Right: Pulse Wave based ToF.

necessary to flight from the camera to the scene and from the scene to the camera. This measured delay time t_d is proportional to the distance d between the sensor and the object. In the case of a continuous wave, a periodic signal is emitted (as sinusoidal, triangular and so on) and measured the phase-shift between the emitted and received signals in the sensor, which is proportional to the time t_d and therefore to the distance d.

Pulse Wave

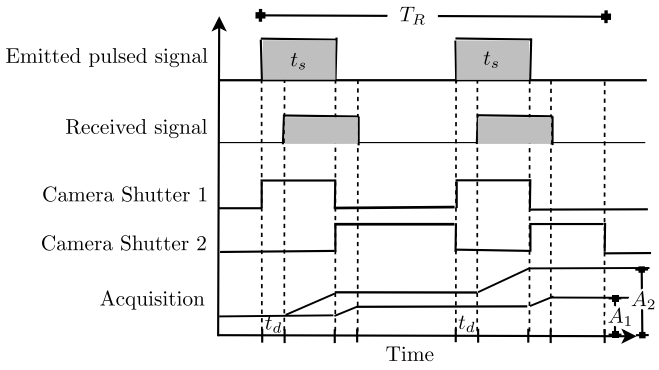


Figure 3.2.: Pulse wave time-of-flight camera working diagram. [80]

The Time-of-Flight cameras based on pulse wave (PW) work sending a pulse in order to measure the time used to travel from the camera to the scene and then coming back. The Figure 3.2 shows the main idea of measuring the time-of-flight using a pulse wave. We use two measurements one after the other synchronized with the emitted signal. Let us have a pulsed signal with time width t_s and two synchronized samples with the emitted pulse signal. The first shutter captures the light coming to the sensor in the same interval of the emitted pulse and the second shutter captures the light received in the consequent window time t_s. Intuitively, the first shutter measures the quantity of light received which is coming back to the sensor, and delayed by the time t_d. The second shutter measures how much light is remaining after the pulse width t_s. In other words, the second light acquisition measures a value proportional to the delay time t_d. Additionally, the sum of acquired

light one and two is proportional to the pulse width t_s because is the complete emitted pulse. In equations, we have:

$$\left.\begin{array}{rcl} A_2 & \propto & t_d \\ A_1 + A_2 & \propto & t_s \end{array}\right\} \Rightarrow \frac{t_d}{t_s} = \frac{A_2}{A_1 + A_2}$$

$$\Rightarrow t_d = t_s \cdot \frac{A_2}{A_1 + A_2} \qquad \Rightarrow d = \frac{1}{2} \cdot c_{light} \cdot t_s \cdot \frac{A_2}{A_1 + A_2}$$

where c_{light} is the speed of light. Normally, to measure distance in order of meters, the necessary time t_s is in order of nanoseconds (ns). Unfortunately, the quantity of light which can be acquired in this window of time is low; and therefore it is necessary to make several captures in a *repetition time* or *integration time* so it is possible to increase the *Signal-to-Noise Ratio* (SNR). Signal-to-Noise ratio is a ratio to measure how good is the signal compared to noise. When a SNR is high, it means that the signal is bigger than the noise; and in consequence the noise disturb in a smaller quantity to the captured signal. Instead, when a SNR is low, the amount of captured signal is comparable to noise This type of time-of-flight measurement is used not only with infrared light, but also with laser and visible light, due to its simplicity.

Continuous Wave

The Time-of-Flight cameras based on a continuous wave (CW), as opposite to pulse wave, uses a continuous wave signal instead of a pulse signal only. This continuous wave needs demodulation properties in order to measure the time, taken for the signal which goes to the scene and comes back to the sensor. In the majority of the implemented ToF cameras, the preferred signal is a *sinusoidal* wave, due to the facility of its demodulation using a reasonable quantity of samples. However, the work of Linder *et al.*[70] studies the feasibility of the use of triangular signal, instead of the sinusoidal, which electronically can be generated easier than a sinusoidal wave. In Figure 3.3 are shown two different periodic signals, which can be used in a ToF camera.

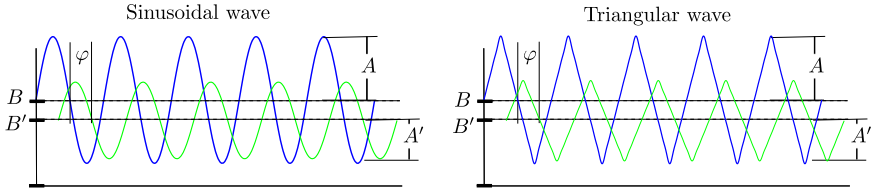


Figure 3.3.: Continuous wave time-of-flight camera example.

In the following subsections, the working principles of ToF cameras based on a sinusoidal continuous wave will be explained in detail, due to its use in the majority of time-of-flight cameras in the actual market.

Combined Wave

The time-of-flight cameras based on a combined wave mix the advantage of each modality: continuous and pulse wave. This type of camera is only theoretical, due to its difficult implementation.

An example of the waves used in this case can be observed in the Figure 3.4.

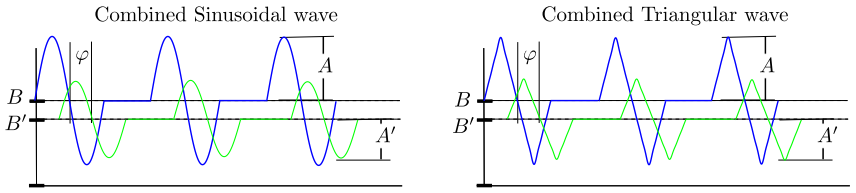


Figure 3.4.: Combined wave time-of-flight camera example.

3.2.2. Demodulation of the Sinusoidal Signal

The ToF camera emits a sinusoidal intensity modulated signal and its sensor captures the reflected signal. This reflected signal is used to calculate the *phase shift* between the emitted and received signals as it can be observed in Figure 3.5. This phase shift is proportional to the time-of-flight and in consequence is directly related to the distance between the sensor and scene. The formula which relates the phase shift φ and the distance d is as follows:

$$2 \cdot d = \frac{c_{light}}{2\pi f} \cdot \varphi \Rightarrow d = \frac{c_{light}}{4\pi f} \cdot \varphi \tag{3.1}$$

where f is the modulated frequency of the sinusoidal signal (normally between 20 and 30 MHz), c_{light} is the speed of light ($3 \times 10^8 \frac{m}{s}$). One full cycle of the modulated signal defines a non-ambiguous range, where the distance can be measured. Out of this range, the measured depth is the difference between the real distance and a multiple of the maximum non-ambiguous distance. For example, in a PMD CamCube 2.0[1] working at a modulation frequency of 20 MHz, the non-ambiguous range goes until 7.5 meters.

In order to estimate the received signal, the ToF camera takes four samples. These measurements are made in four suitable time points of the original signal corresponding to $t = i\frac{\pi}{2\omega}$ where $i = \{0, 1, 2, 3\}$ and ω is the angular frequency. These four samples give four values named $C(\tau_i)$, $i = \{0, 1, 2, 3\}$. With these four samples, the ToF camera can completely reconstruct the received sinusoidal signal. The demonstration is shown as follows:

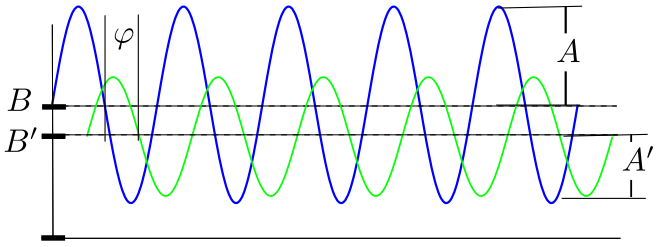


Figure 3.5.: Sinusoidal continuous wave.

[1]Camera from PMD Technologies `http://www.pmdtec.com`

Given the emitted signal $g(t)$ and the received signal $S(t)$:

$$\text{Emitted signal: } g(t) = A \cdot \cos(\omega \cdot t) + B \tag{3.2}$$

$$\text{Received signal: } S(t) = A' \cdot \cos(\omega \cdot t + \varphi) + B' \tag{3.3}$$

where φ is the phase-shift between the emitted and received signals, A is the amplitude and B is the offset or bias of the emitted signal; A' is the amplitude and B' is the offset or bias of the received signal; and finally ω is the angular frequency of the sinusoidal wave.

The ToF camera measures four samples at $\omega \cdot t = i\frac{\pi}{2}$ where $i = \{0, 1, 2, 3\}$ which gives the following measurements:

$$\text{Time } i = 0 \wedge \omega \cdot t = 0 \Rightarrow \quad C(\tau_0) = A' \cdot \cos(0 + \varphi) + B' = A' \cdot \cos(\varphi) + B' \tag{3.4}$$

$$\text{Time } i = 1 \wedge \omega \cdot t = \tfrac{\pi}{2} \Rightarrow \quad C(\tau_1) = A' \cdot \cos(\frac{\pi}{2} + \varphi) + B' \tag{3.5}$$

$$\text{Time } i = 2 \wedge \omega \cdot t = \pi \Rightarrow \quad C(\tau_2) = A' \cdot \cos(\pi + \varphi) + B' \tag{3.6}$$

$$\text{Time } i = 3 \wedge \omega \cdot t = \tfrac{3\pi}{2} \Rightarrow \quad C(\tau_3) = A' \cdot \cos\left(\frac{3\pi}{2} + \varphi\right) + B' \tag{3.7}$$

Simplifying and replacing the cosine rule $\cos(\alpha+\beta) = \cos(\alpha)\cdot\cos(\beta) - \sin(\alpha)\cdot\sin(\beta)$, it is obtained:

$$\begin{aligned}
C(\tau_0) &= A' \cdot \cos(0 + \varphi) + B' = A' \cdot \cos(\varphi) + B' \\
C(\tau_1) &= A' \cdot \cos(\frac{\pi}{2} + \varphi) + B' = -A' \cdot \sin(\varphi) + B' \\
C(\tau_2) &= A' \cdot \cos(\pi + \varphi) + B' = -A' \cdot \cos(\varphi) + B' \\
C(\tau_3) &= A' \cdot \cos(\frac{3\pi}{2} + \varphi) + B' = A' \cdot \sin(\varphi) + B'
\end{aligned}$$

Then, four theoretical formulas are acquired, representing the measured data by the ToF camera. Therefore, the phase shift (φ), amplitude (A') and offset (B') of the received signal can be calculated using the samples $C(\tau_i)$ ($i = 0, \cdots, 3$) as follows:

- Offset B' (Gray-scale):

$$\begin{aligned}
C(\tau_0) + C(\tau_1) + C(\tau_2) + C(\tau_3) =& A' \cdot \cos(\varphi) + B' - A' \cdot \sin(\varphi) + B' - A' \cdot \cos(\varphi) + B' + A' \cdot \sin(\varphi) + B' \\
\Rightarrow B' =& \frac{C(\tau_0) + C(\tau_1) + C(\tau_2) + C(\tau_3)}{4}
\end{aligned}$$

- Amplitude A':

$$\begin{aligned}
\sqrt{\left(C(\tau_3) - C(\tau_1)\right)^2 + \left(C(\tau_0) - C(\tau_2)\right)^2} &= \sqrt{\left(2A' \cdot \sin(\varphi)\right)^2 + \left(2A' \cdot \cos(\varphi)\right)^2} \\
\Rightarrow A' &= \frac{\sqrt{\left(C(\tau_3) - C(\tau_1)\right)^2 + \left(C(\tau_0) - C(\tau_2)\right)^2}}{2}
\end{aligned}$$

- Phase Shift φ:

$$\begin{aligned} C(\tau_3) - C(\tau_1) &= A' \cdot \sin(\varphi) + B' - (-A' \cdot \sin(\varphi) + B') \\ &= 2A' \cdot \sin(\varphi) \\ C(\tau_0) - C(\tau_2) &= A' \cdot \cos(\varphi) + B' - (-A' \cdot \cos(\varphi) + B') \\ &= 2A' \cdot \cos(\varphi) \\ \frac{C(\tau_3) - C(\tau_1)}{C(\tau_0) - C(\tau_2)} &= \frac{2A' \cdot \sin(\varphi)}{2A' \cdot \cos(\varphi)} \\ &\Rightarrow \varphi = \arctan\left(\frac{C(\tau_3) - C(\tau_1)}{C(\tau_0) - C(\tau_2)}\right) \end{aligned}$$

Finally, the received signal can be reconstructed with the following three equations using only four samples:

$$\text{Offset:} \qquad B' = \frac{C(\tau_0) + C(\tau_1) + C(\tau_2) + C(\tau_3)}{4} \tag{3.8}$$

$$\text{Amplitude:} \qquad A' = \frac{\sqrt{(C(\tau_3) - C(\tau_1))^2 + (C(\tau_0) - C(\tau_2))^2}}{2} \tag{3.9}$$

$$\text{Phase Shift:} \qquad \varphi = \arctan\left(\frac{C(\tau_3) - C(\tau_1)}{C(\tau_0) - C(\tau_2)}\right) \tag{3.10}$$

Another and more general point of view is the demodulation of the phase between the emitted and received signals using its *cross-correlation*. The cross-correlation technique gives a comparison between two signals, and recovers the relation of amplitude and phase between the emitted and received signals. Usually, the emitted signal has to be known. The general *cross-correlation* function $C(\tau)$ is as follows:

$$C(\tau) = S(t) \otimes g(t) = \lim_{T \to \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} S(t) \cdot g(t + \tau) \cdot dt \tag{3.11}$$

where $\otimes$ is the convolution in the time between two waves, $S(t)$ received signal and $g(t)$ reference signal, τ is the sampling time.

Resolving this convolution for our signal $g(t)$ and $S(t)$, we get the next equation:

$$C(\tau) = \frac{A' \cdot A}{2} \cos(\omega\tau + \varphi) + B' \cdot B \tag{3.12}$$

When we replace in this equation different values of $\omega\tau_i = i\frac{\pi}{2}$ with $i = \{0, 1, 2, 3\}$, we obtain the following equations:

$$\begin{aligned} C\left(\tau_0 = 0\right) &= \frac{A' \cdot A}{2} \cos(\varphi) + B' \cdot B \\ C\left(\tau_1 = \frac{\pi}{2}\right) &= -\frac{A' \cdot A}{2} \sin(\varphi) + B' \cdot B \\ C\left(\tau_2 = \pi\right) &= -\frac{A' \cdot A}{2} \cos(\varphi) + B' \cdot B \\ C\left(\tau_3 = \frac{3\pi}{2}\right) &= \frac{A' \cdot A}{2} \sin(\varphi) + B' \cdot B \end{aligned}$$

Simplifying and resolving, the values of the phase φ, amplitude A' and offset B' can be recovered as follows:

$$\text{Offset:} \qquad B' = \frac{C(\tau_0) + C(\tau_1) + C(\tau_2) + C(\tau_3)}{4B} \tag{3.13}$$

$$\text{Amplitude:} \qquad A' = \frac{\sqrt{(C(\tau_3) - C(\tau_1))^2 + (C(\tau_0) - C(\tau_2))^2}}{2A} \tag{3.14}$$

$$\text{Phase Shift:} \qquad \varphi = \arctan\left(\frac{C(\tau_3) - C(\tau_1)}{C(\tau_0) - C(\tau_2)}\right) \tag{3.15}$$

3.2.3. Sensor Sampling

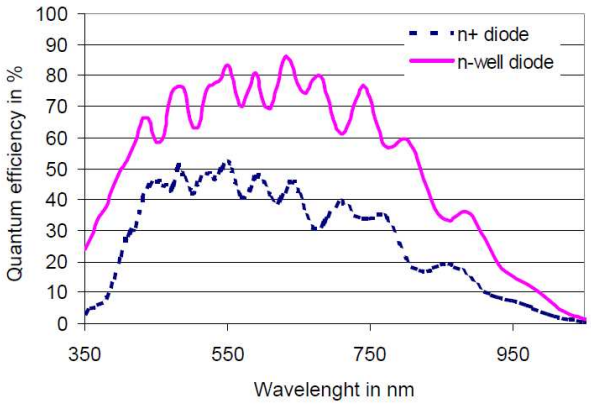


Figure 3.6.: Measured quantum efficiency curves realized in a $0.5\mu m$ CMOS process. [64]

The image sensor used in the ToF camera is a CMOS/CCD[2][3] sensor, which combines the idea of CCD internal storage and fast CMOS acquisition. This sensor has a response to the infrared light and the typical sensor has a spectrum response as shown in the Figure 3.6. This spectrum allows to select a wavelength for the light, in order to acquire enough light for the ToF calculation. Besides, we can observe that the sensor can acquire light in the Infra-Red $750nm - 1400nm$ (IR) spectrum, allowing to make an active camera which is invisible for the human eye. In practice, the real sensor cannot take a single measurement in one point of time t, due to the quantum structure of the light, in other words, the sensor has to accumulate photons to obtain a usable measurement. For this reason, a window of time Δt is needed for capturing a sample with the ToF camera. In this case, the measurement function $S(t)$ in the time-domain is:

$$S(t) = \int_{t-\frac{\Delta t}{2}}^{t+\frac{\Delta t}{2}} \left(A' \cos(\omega t + \varphi) + B'\right) \cdot dt \tag{3.16}$$

[2]CMOS: Complementary Metal Oxide Semiconductor
[3]CCD: Charge-coupled Device

This integral function represents the light captured by the sensor between the time $t - \frac{\Delta t}{2}$ and $t + \frac{\Delta t}{2}$; and the solution of equation 3.16 gives the next measurement function:

$$S(t) = \frac{2A'}{\omega} \sin\left(\omega \frac{\Delta t}{2}\right) \cos\left(\omega t + \varphi\right) + B' \cdot \Delta t \tag{3.17}$$

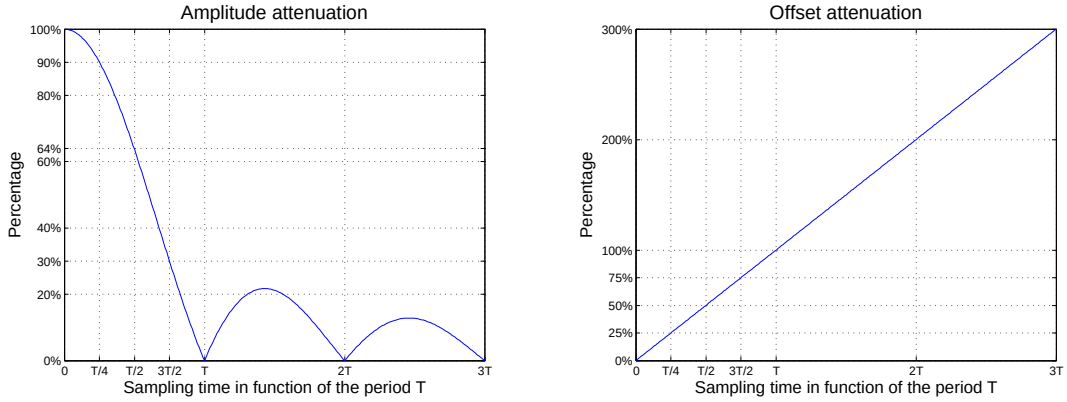
Recalculating the values of the offset B', the phase shift φ and the amplitude A', the obtained values are:

$$\text{Offset:} \qquad B' = \frac{A_0 + A_1 + A_2 + A_3}{4 \cdot \Delta t} \tag{3.18}$$

$$\text{Amplitude:} \qquad A' = \frac{\omega}{2 \sin\left(\omega \frac{\Delta t}{2}\right)} \cdot \frac{\sqrt{\left(A_3 - A_1\right)^2 + \left(A_0 - A_2\right)^2}}{2} \tag{3.19}$$

$$\text{Phase Shift:} \qquad \varphi = \arctan\left(\frac{A_3 - A_1}{A_0 - A_2}\right) \tag{3.20}$$

where $A_i = S(t_i)$ and $t_i = \frac{i\pi}{2\omega}$, with $i = 0, 1, 2, 3$.



(a) (b)

Figure 3.7.: Amplitude (a) and offset (b) attenuation due to the window sampling

These equations show that the measured phase-shift is independent of the size of the window sampling Δt, however, the measured amplitude and offset have an influence from Δt. In the Figure 3.7 the influence of the sampling time Δt in the recovered values of amplitude and offset is presented, which shows different attenuation depending on the sampling time. Normally, the ToF cameras manufacturers select half of the period for sampling, merely because the amplitude is only reduced to 64% and the offset is reduced to 50%, but also by reason of the sampling time that is big enough to make easier the electronic implementation.

In addition, the window sampling used in ToF cameras is in order of nano-seconds, meaning that in this time the quantity of light captured (photons) do not allow us to generate a read-out with a good Signal-to-Noise Ratio (SNR) since the sensor noise is similar to the acquired light. Therefore, the ToF camera usually samples many times in an *integration time*. In Figure 3.8 an example of acquisition in the integration time can be observed, in this case N samples are used.

For further details, you can check the PhD thesis of Robert Lange [64].

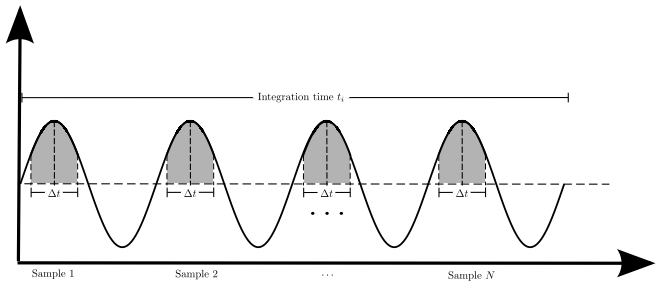


Figure 3.8.: Normal sampling of a ToF camera in the *integration time*.

3.3. Time-of-Flight Camera

The Time-of-Flight cameras suffer from different kind of errors and distortions which are produced by the ToF sensor noise and involved physics. This section introduces a summary of ToF camera errors and distortions, including its causes and possible solutions.

3.3.1. ToF camera parts

In general, a ToF camera can be manufactured using three essential parts: a sensor unit, a process unit and an illumination unit. The sensor unit is in charge of the acquisition process of the sinusoidal signal, this is, capturing the infrared light, reading-out samples and lock-in system (synchronization). Also, this unit includes the used optics in the ToF camera (*e.g.*lens). The process unit not only calculates phase-shift, amplitude and offset of the received signal in every pixel, but it also makes the communication with the external computer, sending data and receiving function commands (*e.g.*setting parameters). Finally, the intensity modulated infrared light is generated by the illumination unit, using Light Emitting Diode (LED) due to its low cost and power efficiency.

3.3.2. Sensor Behavior

There are possible sensors to build a ToF camera. Each sensor has both, advantages and disadvantages. The sensors can be classified based on the sensor type and the lock-in system. The criterion of sensor type refers to the electronic technology used to acquire the sample images and the lock-in system criterion makes reference to the gating of the acquisition and to the storing of every sample pixel in the chip.

Sensor Type

There are two sensor types, which can be used in a ToF camera: Charge-Coupled Device (CCD) and Complementary Metal-Oxide-Semiconductor Active Pixel Sensor (CMOS-APS). These sensors are often used in normal cameras; not only in gray-scale cameras, but also in color cameras. These sensors have some design differences, which have complementary design ideas. The distinctions between them are shown in the following list:

- The CCD sensor has a high power consumption and power dissipation, instead of CMOS-APS sensor which has a low power consumption and power dissipation, being more power efficient than the CCD sensor.
- On the other hand, the CCD sensor has low noise in the capture of photons, being a very noiseless sensor. Instead, CMOS-APS sensor is strongly influenced by the noise, nevertheless, this noise can be electronically reduced using a pre-measured fixed noise pattern.
- The CMOS-APS can select which pixel wants to read, instead of CCD can not do that. The CMOS-APS sensor can read each pixel independently, allowing to read a region of interest without the necessity of reading all the pixels, in opposite to CCD which always has to read and reset all the pixels together.
- As opposed to CMOS-APS, which can directly read the voltage of the pixel storage, CCD needs to move the charges from the pixel storage to a special conditioned voltage reading storage.
- The CMOS-APS has lower production cost compared to CCD sensor.

For the reasons exposed before, the developers of ToF camera usually use a combination between these sensor types. This new sensor used in ToF cameras is named *Photo Mixing Device* (PMD). Basically, the developers obtain in the same chip sensor (PMD), the benefits of every sensor type: low noise plus directed transportation and storage into defined storages due to CCD architecture; and random pixel access, allowing a selection of a region of interest (ROI) due to CMOS-APS. They can achieve this sensor making it, using the architecture of CMOS-APS circuitry fabricated with CCD technology. Furthermore, there is an improvement on the exposed sensor, as used by PMD Technologies, named Range Imaging (RIM) sensor, which uses in every pixel a special distribution of the semi-conductor in order to measure the phase-shift more accurately.

Lock-in System

The lock-in system is the process used to acquire ToF camera samples. The lock-in system allows the demodulation of the received signal and therefore to estimate the phase-shift. There are basically three types of lock-in systems: 1-tap lock-in, 4-tap lock-in and multi-tap lock-in. The different lock-in systems get different quantities of data in each capturing shot, which is explained in section 3.2. That section explains how to make samples, in order to recover the phase shift, amplitude and offset of the received signal using only four sampling points. These different lock-in systems define different ways to take these samples, and are explained in the next list:

- **1-tap Lock-in**: The 1-tap system allows to capture only one sample by cycle of the sensor clock. This is usually used in a grayscale camera, capturing only one data per clock cycle using one pixel storage. In the case of ToF camera, the camera can capture only one sample point in each clock cycle. That means that the camera can acquire only one information per clock cycle, increasing the time of acquisition when it is necessary to capture these ToF camera four samples. For example, if there is a sampling frequency of 80 MHz, $C(\tau_0)$ can be obtained in the first cycle and then it is necessary to wait three cycles more to sample again $C(\tau_0)$ (to accumulate and improve the Signal-to-Noise Ratio) or read-out $C(\tau_0)$ with a low SNR and

change to sample $C(\tau_1)$. Having only one storage, it expands the acquisition time of every sample, by four clock cycles to make measurements with a good SNR.

- **4-tap Lock-in**: This system allows to capture four samples in independent storages in every pixel sensor per clock cycle. In other words, this system allows to get four samples at the same time which allows to have overlapping between the samples. Furthermore, the system permits to accumulate each sample $C(\tau_i)$ with $i \in 0, \ldots, 3$ in different charges storage. Each pixel has its own readout-line[4] and its own shutter line[5], making easier the simultaneous acquisition, but making more difficult the sampling synchronization.

- **Multi-tap Lock-in**: The multi-tap system is also composed of a four pixel sensor, which allows capturing each sample separately. Every pixel sensor is connected to the same read-out line and the same shutter, but each pixel storage captures with a defined shift. This type of sensor is named 4-phase CCD sensor, which allows capturing the samples and store them in an internal storage. With this type of sensor is possible to capture with a very high frequency and with an easy synchronized way, allowing us to capture the samples, in order to demodulate the received sinusoidal signal.

3.3.3. ToF camera error

The Time-of-Flight camera has different errors and noises influencing in its depth accuracy. These errors are principally produced by sensor noise and light noise. The sensor noise is produced by the thermal noise and electronic noise (extra electrons); and the light noise is produced by the photon shot noise and background noise (extra photons).

Sensor Noise

The sensor noise is produced by the electronic involved in the device, and it can be reduced with a proper management and design of the device. This noise is produced by the following:

- **Electronic shot noise**: The electronic shot noise is related to the collision between charges going to the output voltage.

- **Thermal noise**: This is produced by the vibration of the electrons in different temperatures of the material. For this noise is important to know the temperature range where the ToF camera works and where the dissipater can work properly.

- **Reset noise**: the reset noise is generated by the movement of charges to reset the sensor pixels and to get a new data.

- **Flicker noise**: The flicker noise is produced by the noise in the continuous voltage which is used to supply the sensor. We can never get a perfect continuous voltage, but today this noise is almost reduced using voltage regulator.

[4]readout-line is the electronic line to transport charges from the pixel or voltage storage to the reader device
[5]Shutter line is the line to activate the capturing in every pixel storage

- **Quantization noise**: This noise is made by the conversion of photon to electric charges in the semi-conductor sensor. This quantization depends on the depth penetration of the photon in the semi-conductor to produce the charge.

It is important to mention, that these noises can be almost totally reduced not only using electronics, but also using filtering and/or fixed pattern noise.

Light Noise

Light noise is produced not only by the quantity of noise, which is shutting the sensor, but also by the background noise. The first noise depends of the light quantity reflected to the sensor, which changes, depending on the reflectivity coefficient of the material and the shape (normals) of the object. As we read in the section 2.1, the angle between the normal of the surface and the direction of the light is related to the reflected light. There are two reflection types, specular and diffuse reflections. The first type corresponds to the reflection in a surface with perfect reflectivity, reflecting the light in the same angle with respect to the surface normal. The second type corresponds to the reflection in an opaque surface with scattering which can be estimated as a reflected light in all directions. This diffuse reflection is usually modeled, using the Lambertian Law. The majority of the materials have specular and diffuse reflections, being both complementary, depending on the observer position. The second noise is related to the light noise generated by the environment and other equipment emitting infrared light near to the scene, *e.g.*sun light, remote control, infrared equipment and so on. This noise produces an interference in the received signal, generating a noise, which can be observed in the sinusoidal signal as a high-frequency noise creating an error in the distance measurement. The background noise can be reduced using an adaptive close-loop in the ToF camera in order to learn this noise in the time to compensate it. This feature is named *Suppression of Background Intensity (SBI)* and some commercial ToF cameras have this feature available in the camera setting.

3.3.4. Motion Blur

Motion blur is produced because the samples $C(\tau_i)$ are taken in different times and are taken repetitively in the integration time. This produces, that the phase shift calculation has an introduced artifact by the motion of the object, since the samples have been acquired while the object is moving. This artifact generally appears as a blurring in the object edges and in the noncontinuous places in the scene. This blurring can be reduced decreasing the *integration time* and reducing the delay between the samples. Some ToF cameras have this option and can be activated in order to reduce the motion blur. The work of Hussmann [49] proposed a suppression method of the motion artifacts in the ToF camera in real-time. Another work compensates the lateral and axial motion artifacts by tracking the object on the level of phase images. The tracking is used to align the object, and in consequence, to correct the depth image computation, reducing the artifacts and enhancing the depth reliability [69].

3.3.5. Systematic Error or Wiggling

A perfect and theoretical sinusoidal signal cannot be generated electronically producing the systematic error. This error can be learned from calibration and be reduced. This error depends on the generated signal and the position of the object, however, it can be reduced calibrating depending on the distance and position in the image plane of the ToF camera. There are some approaches which reduce this error and manipulate the data to improve the accuracy of the measurement, it is shown in the subsection 3.3.6.

3.3.6. Calibration

Here, they are shown different ways to calibrate a ToF camera in order to calculate the 3D point w.r.t.the camera origin and, to improve the depth accuracy of the camera. A normal ToF camera can be calibrated using a standard camera calibration technique, *i.e.*, intrinsic and distortion parameters (pin-hole model). The simplest way is using a normal checkerboard which is moved in front of the ToF camera changing distances from the camera. Thanks to the property of the reflectivity coefficient, the white color has higher amplitude of the received signal than the black color, and therefore it is possible to identify the corners of the checkerboard squares. These corners are the basis of the standard calibration techniques. However, the ToF camera has a low resolution and, in consequence this calibration totally depends on the corner detection. In the case of a ToF camera resolution range of 160x120 pixels and 204x204 pixels, the corners can be detected in a reasonable accuracy which allows to get a good calibration [67]. In the case of lower resolution, it is necessary to use an optimization approach in order to get a good calibration [11, 72]. Other works use different types of checkerboard, for example, point reflector with different depth and shapes; and so on [36, 26, 25].

In addition, the ToF cameras can be calibrated in order to improve its depth accuracy, which is named *depth calibration*. There are some works in this direction and they usually create a function relating an image position and an uncalibrated depth in order to get the new corrected depth. This function can be a constant offset [68, 99, 52], linear function [68], polynomial function [70], B-Spline [70] or just look-up table [52].

Furthermore, some works attempt to improve the depth accuracy of the ToF camera adding the intensity and amplitude values to the calibration [68, 98]. The work of Linder *et al*. [68] uses the intensity in the function of depth correction to incorporate the information of the reflected light for the correction function. The work of Radmer *et al*. [98] uses the amplitude value for the calculation of the corrected depth, using Lambertian reflectance properties. The ToF cameras can also be influenced by the thermal condition and the best integration time for different environmental conditions which can be calibrated [108].

Another important topic in the calibration step is how to obtain the ground truth. There are basically two different directions, the first is getting the ground truth using a robot or pan-tilt, where it is possible to obtain the 3D position and the orientation of the camera [33, 32]. The second is getting the ground truth data using a visual-based estimation, using checkerboard or line tracking [67, 52]. This visual-based estimation can be made, using a checkerboard observed by one or more high-resolution cameras and a ToF camera. With a multi-camera view, it is possible to estimate the 3D position of each checkerboard corner and it can be optimized using different points

of view [68, 102]. In addition, when the ground truth has to be obtained, it is necessary to define how many data is necessary for the calibration. In this way, it is necessary to define, not only the quantity of images obtained in one determined distance, but also the step and range of distance for the calibration. The range and step have to be defined, depending on the range of work and the accuracy of the ToF camera application. Furthermore, the size of the ground truth will define the time for the calibration which also depends on the used algorithm. The work of Schiller *et al*. [102] shows an optimization method using a checkerboard as ground truth and they calibrate the rigid transformation between the ToF and a number of high-resolution cameras. This calibration gives a polynomial or b-spline function and it can also be calibrated adding the intensity values. There is an open-source toolbox and it can be downloaded from www.mip.informatik.uni-kiel.de.

Besides, there is a new technique to denoise the depth image from a ToF camera, using an adaptive total variation based approach of first and second order, which takes into account geometric properties, *i.e.*, edges and slopes [65]. Another way to improve the ToF camera accuracy is based on the modeling of the multi-path interference of the ToF camera and the scene structure in order to correct the depth measurements [31].

3.4. Time-of-Flight Camera and Computer Vision

Recently, the time-of-flight camera has been introduced in large number of applications in computer vision and robotics. This camera provides three-dimensional information which can be used in different applications: 3D reconstruction, segmentation, augmented reality, motion estimation, object recognition and so on. This camera can be used alone or can be combined with one or more ToF and/or normal color cameras.

3.4.1. ToF Camera Simulation

There are some works which simulate the ToF camera modeling the acquisition process, demodulation, noise (error), motion artifacts and flying pixels. These simulations allow us not only to understand and model the principles working of the ToF camera, but also to evaluate the ToF sensor with different parameters. From this point of view, the simulations are very important due to allow us to generate synthetics depth and amplitude images, so it is possible to test and evaluate new algorithms using this synthetic data as ground truth. The work of Peters *et al*. [95] shows a basic model of a ToF camera and its simulation is implemented in the mathematical software MATLAB[6] and therefore cannot achieve real-time frame rate, nevertheless, it was the first ToF camera simulator. The works of Keller *et al*. [54, 53] incorporate motion artifacts, sensor error and scattering in the object surface which are implemented using parallel programming (OpenGL/C++) which works in real-time frame-rate. Work [53] modeled the scattering using a Lambertian model for the reflectance of the light and they also integrate the captured light by the ToF camera as a multi-ray reflection in a certain area of the object instead of using a single ray.

[6]MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language, developed by MathWorks.

3.4.2. Range Image Processing

The ToF camera range image has some known error, it can be reduced using different algorithms applied to the depth image. This range image can be manipulated to eliminate the outliers and to reduce the depth error of the ToF camera depth data as described before (*e.g.*flying pixel, motion artifacts). The processing of range images can be divided in the following categories:

- **Outliers filtering:** A basic filtering can be done using the confidence map provided by the amplitude images, allowing us to detect depth information with a low probability of being a correct depth. One way for outliers removal is to filter with an amplitude value between 20% and 80% which corresponds to low confidence and saturated data respectively. However, the amplitude image is related to the received light on the sensor which decreases when a distant object increases and when the pixel is closer to the image boundaries.

- **Flying pixels removal:** Flying pixels are an artifact produced in object discontinuities due to a calculated average of signals with different depths. This artifact can be eliminated using a bilateral filter applied to the range data [47]. It is also possible to remove them using edge-directed re-sampling combined with an upscaling of the depth image [71].

- **Depth data improvement:** The ToF camera also provides an intensity image (offset of the received signal), and therefore, the intensity data can be used to improve the depth data or vice-versa [90]. In addition, the depth data can be improved, using approaches from boundary preservation of subdivision surfaces, and a bimodal infra-patch similarity with optional color information [47].

3.4.3. Color image fusion

In this part, there are several works which use a combination of one or more ToF cameras with one or more high-resolution color cameras. Many researchers combine one ToF camera and one high-resolution RGB camera [66, 46, 50, 71, 111], or many high-resolution RGB cameras [38], in order to enhance the low resolution of the ToF camera using the high-resolution color data. This combination allows us to register each 3D point from the ToF camera to one pixel in the high-resolution camera image plane, using a rigid transformation (extrinsic and intrinsic parameters) and it can be used to color the surface provided by the ToF camera. There is also a commercial and compact camera which combines a normal camera with a ToF camera from 3DV Systems[7] as a webcam and could be used for gaming consoles [121]. In some approaches, it is utilized a simple mapping of ToF camera 3D point onto the image plane, obtaining a single color value per 3D point [66, 46, 50]. More sophisticated mapping algorithms can be used, as the work of Linder *et al*. [71], where they relate a portion of color image to the projected ToF camera pixel using texture mapping techniques, and they also detect the occlusion artifacts produced by the binocular set-up.

The fusion between a high-resolution color camera and a ToF camera can be utilized, to increase the low-resolution of the ToF camera, obtaining the named "super-resolution" range map. Super-resolution methods generally are based on the fact that the discontinuities of the depth data are shown as changes of color or brightness in a color image. For example, the discontinuities can be

[7] 3DV Systems is an Israeli developer of ToF camera.

fitted in a Markov Random Field (MRF) which allows us to obtain a high-resolution range map based on the MRF net [23]. Another work uses an upscaling of the range map resolution to the high-resolution, using a bilateral filtering and sub-pixel smoothing [122]. A super-resolution can be also created from a slightly shifted point of view and an optimization of these different points of view of the scene [104, 105].

There are also some cameras, which integrate the 3D ToF sensor with the color sensor, using the same single lens. One advantage of this kind of camera is that the fusion of the range map with the color image is direct because the same 3D point is acquired by the depth and color sensor. However, the hardware and optics have to be more sophisticated to achieve that. There are some examples of this kind of camera. One is an early 3DV Vizcam which is a 2D/3D camera aimed to TV production. This camera uses a pulsed wave modulation for calculating the depth data [121]. Another 2D/3D camera is a 2-chip sensor which uses a beam-splitter for alignment and an auto-registered 2D/3D acquisition [74].

Futhermore, there are some works of the combination of a stereo camera and a ToF camera. In [63], a combination of a stereo system and ToF camera is introduced to complement the advantages of both sensors. ToF-stereo combination can also be used in order to speed-up the stereo algorithm and it can help to manipulate the texture-less portions of the color image [37]. Another work [10] uses the fusion of ToF-stereo, to estimate small 3D planar patch with its normal. Also [128] uses the combination ToF stereo to get a higher-resolution range data combining stereo and ToF depth data using a Markov Random Field (MRF) which was also extended to a Dynamic MRF using the temporal information [127].

Another completely different new approach uses a shading constraint which models the reflected active light of the ToF camera in order to optimize the 3D surface, based on the measured intensity, amplitude and depth [14]. In practice, this approach assumes a reflectance model, *i.e.*Lambertian model. The reflectance model can be imposed by using a probabilistic model of the image formation to find a maximum posteriori probability to estimate the real surface. This approach produces a notable improvement in the depth accuracy of the ToF camera. Also, this algorithm allows us to estimate the reflectivity of the surface, both globally in the surface and locally when the reflectivity changes across the surface.

There are some works which use a multi-view system employing more than one ToF camera, where the interference begins to be a new problem. The works of Kim *et al*. [55, 56] and Guan *et al*. [35] use different modulation frequencies in the ToF cameras so they can separate the signal avoiding some interference between them. Nevertheless, they do not make any discussion about the management of the signal to guarantee a non-interfered measurement.

Meanwhile, some manufacturers have developed a new active illumination in order to separate the sources using binary codes for example [79].

3.4.4. Geometric Extraction

In this subsection we show the geometric extraction algorithm using time-of-flight camera, including not only mapping, but also camera motion estimation. Time-of-flight camera provides directly the 3D surface observed in a static scene and dynamic scene. Moving the camera and registering these surfaces in a common coordinate system can be used to reconstruct the 3D scene [46]. In the

case of obtaining a high-quality 3D reconstruction, it is necessary to combine the low resolution ToF camera with a high-resolution image-based 3D scene reconstruction, *e.g.*, by utilizing Structure from Motion (SFM) [60, 8]. In the SFM algorithm a no-metric scale reconstruction is obtained, and it can be solved, using the metric information of the ToF camera [111]. This type of work allows us to reconstruct a high-resolution 3D scene at interactive rates, *e.g.*, 3D map building and navigation [96]. When the color and depth can be obtained simultaneously, a depth compensated warping can be utilized for a free point of view rendering [58]. Besides, free view point TV or 3D-TV can be done utilizing ToF cameras, because of its real-time 3D surface acquisition and enables a 3D object recognition. This type of system can be done using a camera or multi-view system combined with a ToF camera and can give an estimated depth of the scene. The depth can be up-scaled and fused creating a super-resolution image with colored depth map like in [122, 128], or including the temporal information like in [127]. These 3D reconstruction sequences can be shown to the viewer, employing auto-stereoscopic display to give glass-less 3D impression or using a display with polarization glasses. The ToF camera was also utilized in the medical domain, helping to intra-operatively register the patient and the pre-operation 3D data which introduces a new registration method taking in account, the ToF camera noisy data matching between meshes [76]. In addition, Schuon *et al*. [18] developed a 3D scanning technique using a moving ToF camera which obtains the 3D model of the object.

3.4.5. Dynamic Scene Analysis

The dynamic scene analysis is a big challenge in computer vision. However, the ToF camera gives a powerful tool to analyze the scene, thanks to its real-time observed 3D object. A ToF/multi-view system can be used to create a 3D reconstruction, for analyzing the scene and also it can be achieved using a combination of a computer-driven pan-tilt unit and a scanning of the scene, in a controlled manner. This scanning is utilized to create a 3D panorama of the environment through the stitching, both, depth and color images into a cylindrical or spherical panorama. Then, a dynamic scene can be captured on-line (*e.g.*a person moving in front of the camera) using adaptive object tracking, and/or using tracking with a fish-eye camera [9, 59]. The information obtained using this type of algorithms can be used in depth-based keying, shadow computation, object segmentation and general mixed reality systems. Furthermore, this information can be also used to detect and track pipeline features, such as junction, bends and obstacles, which extract the features fitting cylinders and cones to depth images acquired inside the pipe [115]. Another application of this type of analysis was made to be used for parking assistance which uses a RANSAC[8] algorithm in order to fit planes in the 3D data for the recognition of ramps and curbs [34].

Another interesting application is related to head detection in order to create a system which deploys the airbag of the car as a function of the head position [30]. Another application includes the recognition of different types of seat-occupant, such as child, child seat, adult, etc. For example, in order to deploy the airbag when the face is too close. For this purpose, an algorithm was developed, that tracks the human body based on Reeb graphs extracted from the range data [22] and 3D skeleton [21].

[8] RANSAC is an abbreviation for "RANdom SAmple Consensus". It is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers.

In the medical area, there are many applications, regarding compensation of the respiratory motion or gating the capture in different modalities of medical imaging. In this area, a ToF camera was also used to measure the respiratory motion, allowing a compensation of the motion in different medical applications with a precision of $0.1mm$, which is competitive to other images approaches [93, 101]. In addition, this respiratory motion detection can be used to monitor respiration during sleep and to recognize any sleep disorder such as sleep apnea [27]. Another medical application is the re-positioning of a patient to a previously examination using a ToF camera. The ToF camera can be used to register rigidly 3D-3D surfaces, in order to align the patient body and the previous position, with a resultant error of about 2.8 mm for translation and 0.28 degrees for rotation of the human body [100]. Another technique for analyzing scene was presented in [114] which uses a ToF camera to create an articulated scene model with the help of the foreground extraction (static objects) and background (moving object) easily provided by a 3D camera, including an analysis of the interaction between the moving objects. Furthermore, ToF cameras allow us to make real-time depth keying in the dynamic 3D scene in order to segment the foreground object, which can be inserted in an artificial rendered 2D background.

3.4.6. Segmentation

Segmentation is a hard procedure, where a computer has to distinguish between the foreground and background. Nevertheless, the segmentation task using a ToF camera reduce the difficulty due to the use of the depth in order to segment. For example, when a person is moving in a scene, the person changes his/her depth w.r.t.the background, allowing us to find easier and faster the object which we want to segment. One way to segment is to make a bilateral filtering of the object boundary based on 2D/3D images [17].

3.4.7. Multi-camera view

The ToF camera can also be utilized to improve multi-camera view systems. As a ToF camera provides a depth map observed by the camera, a multi-view system can be calibrated with the ToF camera in order to use its 3D data for the improvement of the 3D reconstruction, distinguishing occlusion and concave surfaces. One example was developed by Guan *et al*. [35], where a combination of multiple ToF cameras and a multi-view system was made to reconstruct the scene integrating a Shape-from-Silhouettes reconstruction with ToF range data. They use the color cameras and a group of ToF cameras working with slightly different modulation frequency together to extract the silhouette and thus, the 3D object volume using a probability model for the occupancy in the time. Besides, another multi-ToF system focuses on fusing depth images to build 3D reconstructions relying on occupancy probability grids which was presented by Kim *et al*. [56].

3.4.8. User Interaction

One of the first applications of a ToF camera was an interactive screen which tracks the hand to have a touch-free interaction [89]. Also, a similar work was made by Soutchek *et al*. [107] but applied for touch-free navigation of 3D medical visualization. Another interaction system was made by Haker *et al*. [40], which consists of a touch-free mouse. This mouse uses the nose as the pointer which

can be used as keyboard, Dasher control, etc. The tracker uses a special set of features related to the geometry and multidimensional signals, which are utilized to classify bounding boxes and therefore, to determine robustly the position of the nose in the image. The classifier has to be trained with a set of labeled sample images and it is used with a very simple classifier, allowing the extension to another objects. Furthermore, its robustness can be increased using also the intensity image. A similar approach was developed by Witzner [119] which detects faces using the intensity and depth images and segments heads utilizing active contours. One extension of the touch-free mouse was made in [41] which detects the direction of the pointing of the hand. This type of interaction can be used to discriminate if the user is pointing to the system or if he is interacting with other people. One application is to use this pointing to control a slide-show presentation, which depends on the pointing direction on the screen and on the motion of the hand, which gives different instructions to the presentation system, *e.g.*, next/previous slide, laser point, etc. An important application of the user interaction was used in the operating system [94, 107], which allows the physician to use hand gesture and hand movements to control robustly a mark-less 3D medical visualization system and/or a medical imaging system in real-time.

3.4.9. Gesture Recognition and Tracking

Holte *et al*. [44] use the range data to recognize gesture using the motion, which is detected employing band-pass filtered difference of consecutive range images. This group made an extension of the system to recognize full-body gesture including spherical harmonics [45]. In addition, the ToF cameras were used to fit articulated human models, not only full-body articulated [129], but also partially articulated [61, 51]. In [129], some features in the body from range data are tracked and an articulated human model is fitted to estimate the pose of the body. Furthermore, a gait analysis was implemented in [51] which fits a foot-leg-torso articulated model, and estimates the pose using pose-cut algorithm. The pose-cut algorithm minimizes the cost function, based on a Conditional Random Field (CRF) which combines all information of the image (edges, background and foreground), previous pose and shape of the person in a probabilistic Bayesian framework. In [38], a system using a ToF camera and RGB cameras allows tracking people in order to analyze the visible actions. This work is based on a refined shape-from-silhouette algorithm which uses a binary segmentation from RGB and range data. Additionally, another tracking algorithm was made in [42] which uses only a ToF camera, observing the scene at an oblique angle and segments 3D data into non-background clusters. The system, also merges and removes individual clusters in the tracking, due to the occlusion between the objects in the scene. Besides, a range flow estimation was applied to ToF depth data, which can help in the interpretation of complex gesture, camera motion, or object motion[103].

Part III.

Simultaneous Localization and Mapping (SLAM)

4. Simultaneous Localization and Mapping (SLAM) in Endoscopic images

THis chapter explains a proposed Simultaneous Localization and Mapping (SLAM) applied to endoscopic videos. This SLAM is based on the work of Davison [20], but we modified the feature tracking, in order to have better results in endoscopic sequences. This chapter is composed of an introduction, related work, proposed method and results. The introduction explains the motivation of applying SLAM to endoscopic sequences, and its general challenges and difficulties in this type of sequences. The related work shows not only the monocular SLAM but also the works of SLAM applied to medical images, giving a general overview of the state of art of SLAM in medical imaging.

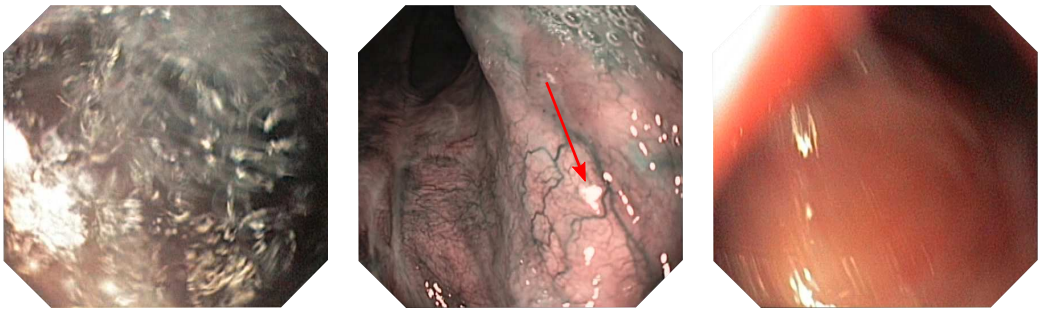
4.1. Introduction

The esophageal cancer is a highly lateral disease affecting a significant percentage of the population in the United States and in the Western World [28, 106]. Recent technological developments provide powerful tools, such as a white light endoscopy, narrow-band endoscopic imaging (NBI) and auto-fluorescence imaging (AFI), which allow an advanced visualization of the malignant tissue. Current gold standard protocol for screening and surveillance of the esophageal cancer, known as Gastrointestinal (GI) endoscopy, consists of visual examination of the esophagus surface under endoscopic guidance and acquisition of biopsies from endoscopically visible lesions. The first time screening is followed by surveillance endoscopies at regular intervals in order to track the evolution of the malignant tissue. This protocol requires identification and re-targeting of the examined regions in the surveillance endoscopy for potential biopsy acquisition. However, currently there is no method available for accurate localization of these regions or re-targeting of previous biopsy sites, under endoscopic guidance.

In this work, we explore the possibility of creating a patient-specific visualization of the esophagus from endoscopic images by performing a 3D reconstruction of its surface. Such a 3D visualization of the esophageal surface can help the endoscopist as a road-map and provide a significant support for re-targeting the biopsy sites during the surveillance procedure.

Reconstruction of the esophagus surface from endoscopic videos involves several challenges: the esophagus tissue presents big deformations, which may greatly modify the appearance of the tissue; moreover, several factors affect the quality of the images, e.g. liquid inside the esophagus generates bubbles (Figure 4.1-a) or specular highlights (see Figure 4.1-b), and a fast motion of the camera leads to blurred images (Figure 4.1-c).

In this work, we investigate the feasibility of using Monocular Simultaneous Localization and Mapping (MonoSLAM) [20] for 3D reconstruction of the esophagus surface, in presence of the



(a) (b) (c)

Figure 4.1.: Challenges of endoscopic video images. Liquid inside the esophagus creates bubbles (a) and specular highlights (b). Also, the camera motion leads to blurred images (c).

above-mentioned challenges. We discuss the extensions required for its application in monocular endoscopic videos and demonstrate that MonoSLAM is a promising framework for 3D visualization of esophageal tissue.

Our approach consists of the following steps: First, in-vivo images are pre-processed in order to remove blurred frames and to enhance edges. Second, Scale Invariant Features (SIFT) [75] are detected and tracked in consecutive frames, using template matching [20, 91] and the Efficient Second-order Minimization (ESM) tracking algorithm [13]. As the video advances, new features are detected and tracked. Then, the MonoSLAM [20] framework, is employed for both, to create a 3D map of the tracked features, and to localize the endoscope (6 DoFs) within this map, simultaneously. Finally, we approximate the observed esophagus surface by linking the features in 3D with a Delaunay triangulation. We also map the endoscopic image tissue texture (from the endoscopic image) to the approximated 3D surface, to facilitate the visualization.

We illustrate our approach on a video sequence acquired with an NBI endoscope. We are able to reconstruct segments of the surface up to for to 35 frames. This upper bound depends on the results of feature tracking and thus, indirectly, on the quality of the video. The fast camera motions or the large number of blurred images prevent us from reconstructing larger segments, due to the lack of successfully tracked features after a period of time. We also analyze the improvement, achieved by integrating the ESM tracker into the MonoSLAM framework.

4.2. Related Work

The reconstruction of 3D surfaces from videos is a well-established research area in the computer vision community [20, 5]. In the medical field, there has been an increasing interest in 3D surface reconstruction from endoscopic image sequences [85, 86, 91, 97, 109, 120, 125]. Stoyanov *et al*. [109] present a method for depth recovery from stereo laparoscopic images of deformable soft-tissue reconstruction. Mourgues *et al*. [86] propose a correlation-based stereo method for surface reconstruction and organ modeling from stereo endoscopic images. Quartucci *et al*. [97] apply a shape-from-shading technique to estimate the surface shape by recovering the depth information from the surface illumination. Methods in [86, 97, 109] provide a dense reconstruction of the

observed surface under well-behaved imaging conditions. However, they are computationally expensive and provide only a reconstruction from two contiguous or stereo images. In practice, in order to localize the endoscope with respect to the observed surface, a longer-term reconstruction is desirable. Such a reconstruction would require the 3D registration of the individually estimated surfaces; however, this procedure is difficult and can easily fail under the challenging imaging conditions of the endoscopic scene.

Introducing a-priori knowledge of the scene can alleviate the computational cost of dense reconstructions. For instance, Zhou *et al*. [125] reconstruct the 3D structure using a Circular Generalized Cylinder (CGC) model by representing and parameterizing the esophagus as a tube formed by a series of 3D circles. A second cost reducing strategy, is the use of features as opposed to dense models. The same group has developed a 3D imaging by synthesizing stereoscopic views from a monocular endoscope [126].

Recently, a number of feature-based techniques have been applied for 3D reconstruction from endoscopic videos [85, 117, 120]. Feature-based matching approaches are able to handle wide-angle viewpoint changes and small overlapping regions. In consequence, they are suitable for long-term reconstructions from conventional endoscopic examinations. An example of feature-based methods is the work of Wang *et al*. [117], who track Scale Invariant Features (SIFT) [75] in the endoscopic sequence, and use Adaptive Scale Kernel Consensus (ASKC) for robust motion estimation. Likewise, Wu *et al*. [120] track SIFT features but use an iterative factorization method for structure estimation. Similar to [117, 120], we use SIFT to detect features. This choice is driven by the properties of SIFT, which allows us to identify distinctive candidate features for tracking at different scales (in scale space). As opposed to [117, 120], we do not use SIFT matching to find correspondences from one video frame to the next. This is to avoid the potentially large number of false correspondences (outliers), that SIFT matching may find due to the homogeneity of the esophagus surface. Instead, we use two methods for tracking the detected features: the ESM tracker and a NCC-based template tracking. We show that the use of ESM improves the results of the reconstruction, when compared to those of the template matching alone. Furthermore, we rely on the incertitude of the current map estimate to constraint the search space.

The estimation of the surface is also different from the approaches presented in [117, 120]. We model the problem as an instance of SLAM (Simultaneous Localization and Mapping). The SLAM framework allows us to consistently gather the information collected up to the current frame, and incrementally reconstruct a map (3D surface model), online, *i.e*.as the endoscope moves. Simultaneously, the position of the camera inside the reconstructed model is recovered. Mountney *et al*. [85, 84] presents a very similar work to ours, who have used SLAM for building a 3D map of the scene for minimally invasive endoscopic surgery with a stereoscopic endoscope. As opposed to [85, 84], we explore the possibility of reconstructing segments of the surface from a monocular endoscope, which is the standard tool used in current GI-endoscopy protocol.

In summary, the method proposed in this work is a feature-based approach using the SLAM framework, which provides estimations of the surface and the camera position online. Specifically, we use the MonoSLAM [20] algorithm designed for reconstruction of conventional indoor spaces. Extensions adapting MonoSLAM to the challenges of in-vivo images are proposed and discussed.

4.3. Proposed Method

The goal of this work is to build a 3D visualization of the esophagus surface in order to assist endoscopist in the localization and re-targeting tasks. The proposed method aims at reconstructing segments of the esophagus surface from a monocular endoscopic image sequence. By formulating the problem within the SLAM framework, we are able to build a 3D map of the observed scene, while simultaneously localizing the camera in the reconstructed 3D map.

Our solution is based on the MonoSLAM [20] algorithm, which was originally designed to reconstruct the map of rigid scenes, mainly of indoor environments. The algorithm starts by detecting features (interest points) in the images, and tracking them along the video. A first rough reconstruction of the map/surface is obtained using the internal parameters of the camera and some initial estimate of the depth of the features. Then, the map is built incrementally and refined as new images are acquired. The incremental procedure alternates between the estimation of the camera position and the 3D construction/update of the features that constitute the map. The alternation is a solution to the ill-posed nature of simultaneous estimation of both camera position and the features. In this way, given an estimated camera location, the position of the features in 3D can be updated and new features added. Conversely, knowing the position of the 3D features and their projection on the current image, an update of the camera position can be estimated.

The result at each frame is a map consisting of a collection of 3D points, as well as the current camera position. In order to reduce the effect of noisy measurements, the update is done through a Kalman Filter, as explained in the Reconstruction section.

Naturally, the quality of the reconstructed map depends on the results of tracking the feature points in the images. In order to improve the results of the template-matching method in [20], we combine the MonoSLAM algorithm with the ESM tracker [13]. The procedure is described in the Feature Tracking section.

Additionally, endoscopic image sequences present several practical issues that make the tracking task more difficult. First, the scene (here, the esophagus surface) may be subject to deformations; this breaks the rigidity assumption of MonoSLAM. Second, the images can be blurred due to the presence of water, or as a consequence of fast camera motion. Finally, the detection and tracking of features in in-vivo endoscopic images is more difficult than in an indoor environment, since the latter does not contain the corner-like features (high frequency variation of the image), which can be reliably tracked in video sequences; instead, endoscopic images of soft-tissue mainly consists of homogeneous regions and lack of the discriminative features. To overcome the difficulties of endoscopic videos, we preprocess the images. In particular, we remove blurred images and enhance the edges in order to improve the tracking results.

Summarily, the proposed method begins with the image-preprocessing step. Following this, SIFT features are detected in every frame, and tracked using either ESM or NCC template matching. Simultaneously, the results of detection and tracking are used within the MonoSLAM [20] framework in order to recover the 3D position of the features as the camera moves. A Kalman filter is used to update the 3D position of the features and the camera. Finally, a surface approximation is obtained with a 3D approximate Delaunay triangulation of the 3D features. In the following sections we describe the different stages of the method in details.

4.3.1. Image preprocessing

Endoscopic images present several challenges. In the preprocessing step we discard blurred images, and we enhance the ridge structures to improve the tracking results.

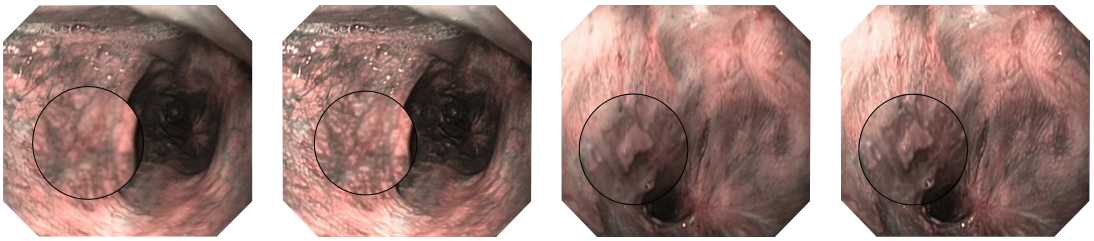
To remove the blurred images, the standard deviation of the colors in the image is calculated. Blurred images tend to have a very small variation of colors; thus, the standard deviation σ_C serves as a measure of blurriness. If the value of σ_C is below a threshold (τ_B), then the image is classified as blurred and skipped.

The enhancement of the edge structures consists of accentuating the high frequency information of the image. High frequency information is easily obtained by comparing the images, before and after applying a Gaussian filter. The procedure allows us to highlight the vessels observed in the stomach and esophagus.

The image operations are described in Equation 4.1. The enhanced image I is obtained from the sum of the original image I_O with the edge image I_E. The edge image is in turn obtained by subtracting the low frequency components of I_O, i.e. subtracting a low-pass filtered image I_L. To compute the blurred image I_L, the original image is subsequently convolved with Gaussian kernels at different scales (in our experiments $S = 9$).

$$\begin{aligned} I &= I_O + I_E = I_O + (I_O - I_L) \\ I &= I_O + (I_O - (G_S \otimes (G \cdots \otimes (G_5 \otimes (G_3 \otimes I_O))))) \end{aligned} \tag{4.1}$$

The result of enhancement is illustrated in Figure 4.2. Vessels in Figure 4.2-a are highlighted after the procedure (Figure 4.2-b). Similarly the structure indicated by the arrow in Figure 4.2-c is easier to detect after the enhancement.



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

Figure 4.2.: Enhancing the edge structures in the images to improve the tracking results. Edge structures are more distinct in the enhanced images (b and d) with respect to the original (a and c).

4.3.2. Feature Detection

After preprocessing the images, the MonoSLAM algorithm [20] is used to reconstruct the model of the observed surface and estimate the camera position. In order to achieve this goal, MonoSLAM incrementally gathers and filters the information coming from *image observations* or *measurements*.

Since a single camera is used, the measurements consist of the position of meaningful features in each of the images.

Features correspond to distinctive local patches in the images. The distinctiveness should permit to easily localize them in a new incoming image. Their position will later provide the information to infer the relative motion of the scene, from one frame to the next. There are different ways to extract such distinctive patches from the images; we employ the SIFT detector described in [75]. In every frame, we select the best 20 features adjusting the peak threshold (τ_{SIFT}). To add a new feature, we also verify that the new detected and tracked patches do not overlap. Finally features detected in specular highlights are removed by identifying if the patch, which is describing the feature, contains a large amount of saturated pixels.

4.3.3. Feature Tracking

After detection, we track the features in the video as the camera is moving. We use two methods for tracking. The first is a template-matching algorithm based on the normalized cross-correlation (NCC) [20, 91] similarity measure. The second is the Efficient Second-order approximation Method (ESM) [13], which is based on a non-linear optimization. Both of these methods allow us to find the position of the features in the new image.

Each feature is described with a template J consisting of the patch around the detected feature. The NCC template tracking consists in searching a similar patch by sliding a window over a new incoming image I. As the window is slid within a search region, the intensity values between the template and the window are compared using the NCC similarity measure. A vector $d = (d_i, d_j)^\intercal$ defines the displacement of the window. The displacement d with the highest NCC score determines the new position of the patch. Given the template J, the new image I, and a displacement in the second image d, the NCC is computed as:

$$NCC(d_i, d_j) = \frac{1}{n^2} \sum_{i,j}^{n^2} \frac{\big(J(i,j) - \bar{J}\big)\big(I(i+d_i, j+d_j) - \bar{I}\big)}{\sigma_J \sigma_I} \tag{4.2}$$

where $J(i,j)$ denotes the intensity values at location (i,j) of the initial template J, and $I(i + d_i, j + d_j)$ stands for intensity values at $(i + d_i, j + d_j)$ of the displaced window. $\bar{J}$ and $\bar{I}$ are the mean intensity values of the template J, and of the current window in I, respectively. Similarly, σ_J and σ_I represent the standard deviation of their intensities. Finally, n denotes both the width and height of the (squared) patch. The optimal displacement d provides the position estimate of the tracked feature (patch) in the current frame.

The second method used for tracking is ESM [13]. ESM is a real-time algorithm based on the non-linear optimization of the sum-of-squared-difference between a given template J and the current image I:

$$SSD(d_i, d_j) = \frac{1}{n^2} \sum_{i,j}^{n^2} \big(\big(J(i,j) - \bar{J}\big) - \big(I(i+d_i, j+d_j) - \bar{I}\big)\big)^2 \tag{4.3}$$

The optimization in ESM is based on an efficient approximation of the SSD (Refer to [13] for an in depth explanation). We have chosen ESM over other tracking methods because of its efficiency

and high convergence rate. One of its major advantages is handling changes of the template due to the 3D motion of planar patches describing the features.

In every new image, we track the features from previous frames using ESM. Unfortunately, given the mentioned difficulties of endoscopic images, it is not always possible to maintain the track of features for long sequences of frames. In order to guarantee that the result ESM tracking is valid, we retain the estimated position of the feature only if it is coherent with the SLAM prediction (see section Reconstruction), i.e. if it falls within the uncertainty region of the feature's predicted location. In case this test fails, the NCC template tracking is used. Finally, if NCC also fails the test, no measurement for that feature is provided and the new position is only updated with the Kalman Filter prediction. This strategy allows us to maintain a reasonable amount of features tracked at each frame.

4.3.4. Reconstruction

The reconstruction is build by triangulating 3D points that lie on the surface of interest. To estimate and update the 3D points from the image observations (in this case tracked features) we use the MonoSLAM [20] algorithm. This solution is based on an Extended Kalman Filter (EKF) to update the feature and the camera positions with every new image in the video. In this section, we recall the main steps of the MonoSLAM algorithm.

The goal of SLAM is to estimate a map of an observed scene, as well as the trajectory of the camera within the map. In the feature-based case, the map is composed of a series of 3D features (also known as landmarks). For the endoscopic application, the problem can be restated simultaneously, estimating the 3D position of some key points on the surface of the esophagus while localizing the endoscope. At the end of the acquisition, the map and the camera trajectory can be visualized in a common global coordinate system.

In EKF-SLAM, the position and orientation of both, the features and the camera, are represented using state vectors and updated with dynamical systems. The camera state vector x is composed of a vector containing the 3D position r, the orientation q, and the linear and angular velocities v and ω of the camera, as shown in Equation 4.4.

$$x = \begin{pmatrix} r & q & v & \omega \end{pmatrix}^{\mathsf{T}}, \tag{4.4}$$

Likewise, each 3D feature $i \in \{1, \cdots, N_k\}$ (with N_k the current number of features) is assigned a state vector that consists of the 3D position.

The camera motion and the feature locations are updated using two dynamic models. These models allow us to predict the camera and feature positions at each time step k.

$$x_k = f(x_{k-1}) + w_k \tag{4.5}$$

The *camera dynamics* model (Equation 4.5) is composed of a non-linear function f depending on the previous camera location x_{k-1}, plus a zero-mean Gaussian noise w_k with covariance Q_k. It is assumed that the camera has constant linear and angular acceleration between times $k-1$ and k, and that accelerations are random processes, following a zero-mean Gaussian distribution.

A second dynamic system is used to model the *measurement* or *observation* process. Our measurements consist of the tracked 2D features in the images. Therefore, the observation model corresponds to the pinhole projection plus the distortion. In this case, a function h is used to model the projection of the 3D feature y_i on to the image; h is a function of the location of the feature $y_{i,k}$ and the camera state x_k, and contains the pre-computed internal and distortion parameters of the endoscope. The resultant model of the observation process is given as:

$$z_{i,k} = h(x_k, y_{i,k}) + v_k, \tag{4.6}$$

where $z_{i,k}$ denotes the 2D position of the feature in the image at time k. As before, a zero-mean additive Gaussian noise v_k is considered, this time with covariance R_k.

In terms of the dynamic systems, the goal of SLAM is to provide an estimate for the full state vector $\hat{X}$, composed of the camera position $\hat{x}_k$ and the collection of features $Y = \begin{bmatrix} \hat{y}_{k1} & \hat{y}_{k2} & \cdots & \hat{y}_{kN} \end{bmatrix}^\intercal$. We use the notation $\hat{}$, to refer to the estimated values. Since a Kalman-based solution is used in [20], a covariance matrix P is associated to the state vector $\hat{X}$ (see Equation 5.22). P provides an uncertainty measure of the current estimates of the camera P_{xx} and map P_{YY} , as well as the correlation between them (P_{xY} and P_{Yx}).

$$\hat{X} = \begin{pmatrix} \hat{x}_k \\ \hat{Y}_k \end{pmatrix}, P = \begin{bmatrix} P_{xx} & P_{xY} \\ P_{Yx} & P_{YY} \end{bmatrix}. \tag{4.7}$$

The algorithm to compute the estimates for $\hat{x}_k$ and y_{ki} proposed in [20] consists of two recursive steps. The first step is a prediction or *time-update* and the second, a correction or *measurement-update*.

The time-update estimates the new camera position and its covariance when going from time $k-1$ to k. The update is computed using Equations 4.8 and 4.9. The position and velocities are computed with an impulse of acceleration. The covariance P_{xx} at time $k-1$ is updated according the rate of change of the current camera estimate; this is computed from ∇f, the Jacobian of f evaluated at $\hat{x}_{k-1}$. Additionally, the covariance update accumulates the covariance Q_k from the disturbance error in Equation 4.5.

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1}) \tag{4.8}$$

$$P_{xx,k|k-1} = \nabla f \cdot P_{xx,k-1} \cdot \nabla f^\intercal + Q_k, \tag{4.9}$$

The *observation-update* deals with the estimation of both, the camera and 3D feature states, given the observed image features $Z = \begin{bmatrix} z_{k1} & z_{k2} & \cdots & z_{kN} \end{bmatrix}^\intercal$ at time k. Z is actually the result of the tracking procedure described in the previous section. The computations of the estimates are done, following the Extended Kalman Filtering. The map and the camera state are updated from their previous state (Equation 4.10) using the optimal Kalman gain W_k and the error e between the measures and the predicted positions of the 2D features ($e = \left[Z_k - h(\hat{x}_{k|k-1}, \hat{y}_{k-1})\right]$). The prediction uses the results of the time-update, noted as $k|k-1$ in the subscript of the camera state $\hat{x}_{k|k-1}$. The full covariance P is updated with Equations 4.11 to 4.13, where ∇h is the Jacobian of h evaluated at $\hat{x}_{k|k-1}$ and $\hat{Y}_{k-1}$.

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{Y}_k \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{Y}_{k-1} \end{bmatrix} + W_k \left[Z_k - h(\hat{x}_{k|k-1}, \hat{y}_{k-1}) \right] \tag{4.10}$$

$$P_{k|k} = P_{k|k-1} - W_k S_k W_k^{\intercal} \tag{4.11}$$

$$S_k = \nabla h P_{k|k-1} \nabla h^{\intercal} + R_k \tag{4.12}$$

$$W_k = P_{k|k-1} \nabla h^{\intercal} S_k^{-1} \tag{4.13}$$

The two updates (time and measurement) are each one repeated when a new image arrives. In this way, the 3D map and the camera trajectory are built increasingly.

As mentioned above, $z_{i,k}$ are the results of the tracking stage. Thus, we consider that a feature is lost, if it falls beyond the uncertainty ellipse around their predicted location $\hat{z}_{i,k}$. Furthermore, the current map and camera position and orientation estimates provide a mean to predict the change of appearance of each template, due to perspective distortion. Thus, to improve the tracking results, the template patch is transformed (warped) according to the estimated 3D location of the current feature $y_{i,k}$, and the camera current state x_k using the observation model h, i.e. $\hat{z}_{i,k} = h(x_k, y_{i,k})$. At the end, the warp permits to correct the projective transformation induced by different viewpoint angles after the camera motion [20].

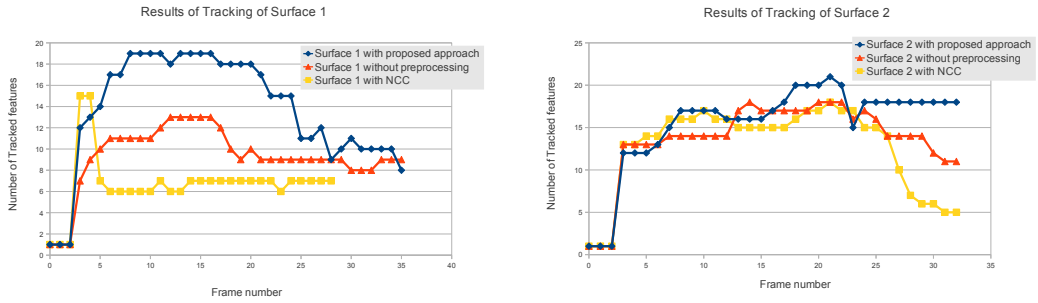
To initialize the SLAM algorithm an initial estimate of the depth is required. First, we manually select a feature to fix the scale of the scene (the reconstruction will be up to scale). Then, all the features are initialized with uniform uncertainty shaped as a 3D Gaussian.

The summary of the MonoSLAM method presented in this section is provided for completeness. Readers should refer to the original publication [20] for further details. The result of applying the overall method is a series of 3D points modeling the surface of the esophagus, as well as the trajectory traveled by the camera. These estimates are updated online, as the endoscopic video acquisition advances. When the number of tracked features becomes insufficient, the current reconstruction is stopped. A new segment of the reconstructed surface starts when a reliable number of features (10) is detected and tracked again.

4.4. Experimental Results

The patient data for the experiments was acquired during an endoscopic examination, using Olympus Exera II endoscope with NBI image enhancement. NBI provides an enhancement of the contrast in the mucosal pattern and submucosal vasculature [124, 19]. As a consequence, the visualization of the structure in the tissue is improved, allowing for more robust feature detection and tracking.

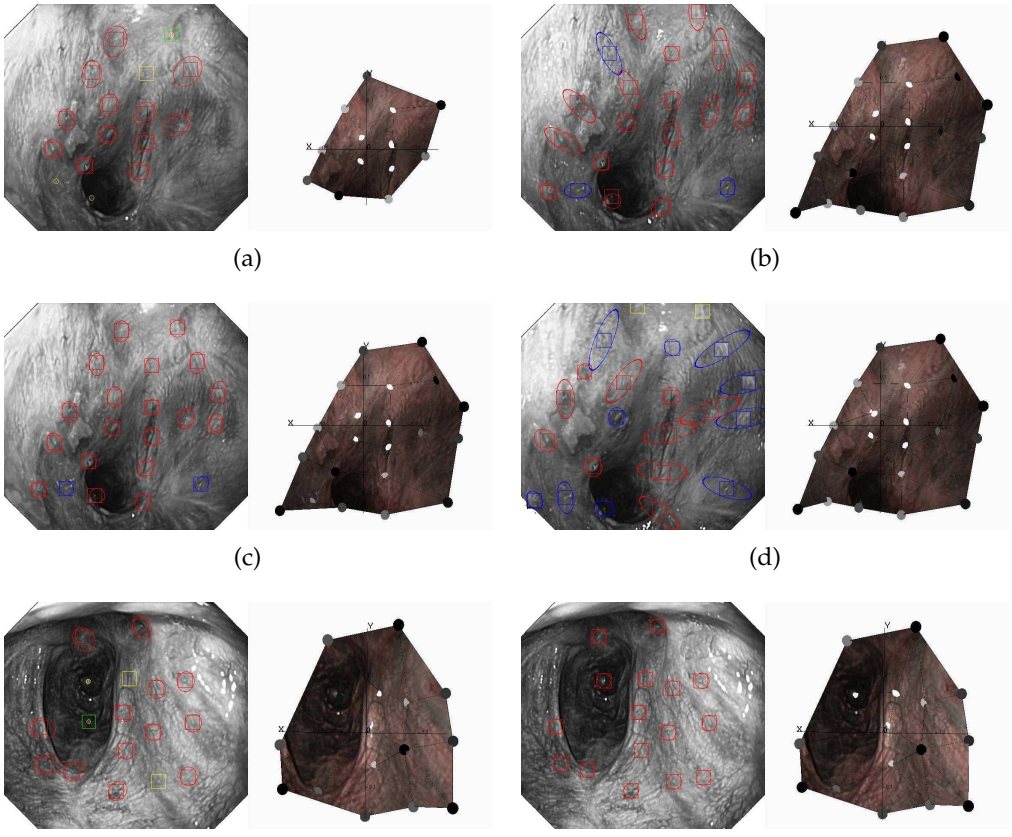
Our proposed approach was applied to NBI patient data. Three 3 sequences of 10, 20, and 30 consecutive frames respectively were successfully reconstructed. Some of the tracking results are shown in Figure 4.4. The red boxes show the tracked features, blue boxes illustrate the predicted position of the features in the image when tracking fails. The position of the features and the trajectory of the camera are shown in Figure 4.4. The surface then has been approximated with a mesh and the texture of the tissue has been mapped onto the approximated surface. Results of the evolution of the algorithm for two of the reconstructed segments are shown in Figure 4.4.



(a) (b)

Figure 4.3.: Results of Tracking of Surface 1 (a) and Surface 2 (b).

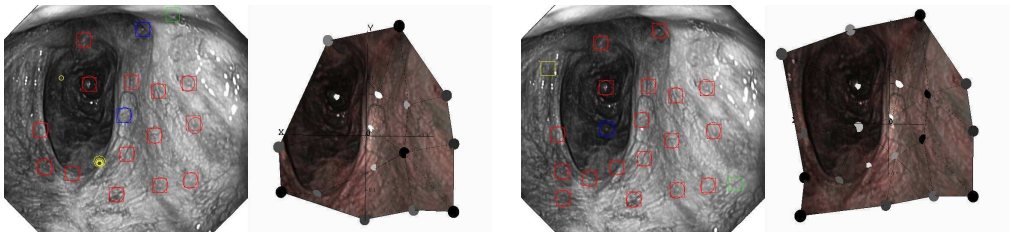
Finally, we demonstrate in Figure 4.3 the improved performance of the algorithm, when using the combined tracking strategy (the ESM tracker and the NCC template matching). We also show the results when using the original images as compared to the preprocessed images. Briefly, these two modifications allow us to track a larger number of features, for larger periods of time.



(a) (b)

(c) (d)

(e) (f)



(g) (h)

Figure 4.4.: (a), (b), (c) and (d) are the sequence of images in the estimation of the Surface 1. (e), (f), (g) and (h) are the sequence of images in the estimation of the Surface 2. The color of the circle represents the depth of the feature, being white the furthest feature and black the nearest feature.

5. Simultaneous Localization and Mapping (SLAM) combining Time-of-Flight (ToF) and High-Resolution cameras

THis Chapter introduces the proposed extension of Monocular Simultaneous Localization and Mapping (SLAM) [20] in order to be used with the new camera generation RGB-D, *i.e.*, a SLAM algorithm which utilize a sensor that provides a color information (RGB) and a depth (D). This extension incorporates the depth information into the internal models of the SLAM algorithm. This proposed extension is evaluated using a combination of a ToF and a High-Resolution cameras. The results show an improvement in the recovered camera trajectory compared to using the normal MonoSLAM. Following, an Introduction, a related work, a problem statement, a proposed method and an experimental validation are presented.

5.1. Introduction

The simultaneous localization and mapping (SLAM) problem consists of finding the position of an object (e.g. robot, camera, etc.) in a map, while simultaneously building the map as the object is moving [24, 7]. Although SLAM has been widely studied in robotics and computer vision, it remains challenging due to the ill-posed nature of the problem, especially when an online performance is desired. A large variety of SLAM approaches have been proposed, that differ both in the type of sensors used (e.g onboard laser scanners [116, 88] or cameras [20, 83, 57, 87]) and in the actual algorithms used to estimate the map and sensor positions [110]. Given their availability and flexibility, the use of cameras has become very popular. One reference algorithm that solves SLAM from image data is the Monocular SLAM (MonoSLAM) proposed by Davison *et al*. [20]. In the MonoSLAM approach, the map is composed of 3D features that are estimated (up to scale) from a calibrated camera and image correspondences in two consecutive views. The estimation of the map and the camera position are alternated. The problem is formulated in terms of two dynamic systems that model the motion of the camera, the measurement (imaging) process and the noise. To find the current state of the systems an Extended Kalman filter is used. Several extensions to the original method have been proposed, for example, that employs stereo cameras [81, 85] to recover a more precise estimation of the 3D position of the features. In this work, we investigate an extension to the MonoSLAM algorithm for a combined High-Resolution Time of Flight (HR-ToF) camera. As opposed to the above stereo approaches, the use of direct depth measures provided by the ToF camera enable our method to work under non-textured surfaces and poor lighting conditions.

In the last few years, there have been increasing advancements in the development of Time of Flight (ToF) cameras. ToF devices have faster frame rates ($\sim$40 fps) than laser scanners ($\sim$2 fps) [116,

a) HR-ToF sensor b) High-resolution imagec) ToF depth imaged) ToF amplitude imagee) ToF offset image

Figure 5.1.: High-resolution ToF camera and the four images simultaneously captured per frame.

88] and are therefore an interesting option to estimate 3D maps from a moving sensor. Recently, methods have been proposed that use ToF technology to estimate the pose of the camera in order to create 3D maps [12, 78, 112, 82]. For the most part these algorithms work off-line or use only the ToF camera information. We are instead interested in an online SLAM approach that takes advantage of the ToF high frame rates. Unfortunately, current ToF devices have low-resolution and precise feature tracking for online SLAM solutions such as that in [20] cannot be achieved.

To cope with the low resolution, new cameras that capture both color intensities and depth information per pixel [1, 2, 3, 4] are currently under development, though no yet commercially available. The combined sensor is often simulated using a ToF camera and a standard high-resolution RGB camera in stereo set-up [78, 43]. Calibration of such set-up suffices to provide RGB-D images. In order to achieve real-time performance while maintaining high quality results, we propose to extend the MonoSLAM algorithm to use such an integrated HR-ToF sensor, composed of a ToF and a regular high-resolution (HR) camera. After the registration step we are able to incorporate the real-time depth information of the ToF camera in to the MonoSLAM framework, while reliably tracking the features in the high resolution camera. The result of the proposed HR-ToF SLAM are a 3D sparse map in metric coordinates (no longer up to-scale) and the trajectory of the camera.

The experiments in section 5.5 show the performance of the proposed method applied to the HR-ToF in comparison to the MonoSLAM approach applied to high-resolution images (HR-SLAM), and to our method using only the low-resolution offset images of the ToF camera. In particular, we measure the camera localization error w.r.t. the ground truth position of the camera obtained with an optical motion capture system. As expected, the availability of direct depth measures improves the localization precision and accuracy, while the high-resolution image guarantees a long-term tracking.

In the section 5.2 we recall the functioning principles of the ToF cameras and briefly describe the registration procedure to simulate the combined HR-ToF sensor. The problem statement and method are explained in sections 5.3 and 5.4 respectively.

5.2. Combined HR-ToF Sensor

A Time of Flight (ToF) camera emits an intensity modulated sinusoidal signal of infrared (IR) light and uses a CCD/CMOS sensor to detect the reflected light. According to the time-of-flight principle, the phase between emitted and received sinusoidal signals is proportional to the distance between the light source and the reflecting surface. Thus, by measuring the phase, amplitude and the offset of the received signal is possible to calculate a depth value for each pixel. Given that the signal is periodic, the modulation frequency limits the maximum distance which the camera can measure to half period of the modulated signal (e.g. from 50cm to 7.5m) [16, 62].

The ToF camera provides at least two images: the depth (Fig.5.1-c) and the amplitude (Fig.5.1-d). The *depth image* gives the measured distance between the reflecting surface and each pixel in the sensor. The *amplitude image* captures at every pixel the amount of IR light that was reflected back to the camera. The amplitude serves as a quality measure of the depth, as poor quality measures arise from low-amplitudes. In addition, the ToF camera may also provide an *offset image* (Fig.5.1-e), which is usually used as gray-scale image. The amplitude and the offset images can be used to calibrate the intrinsic and external parameters of the camera, as well as, the distortion parameters. The calibration procedure is equivalent to the used for high-resolution cameras, *e.g.*moving a known calibration pattern (chessboard) in front of the camera and optimizing the intrinsic and distortion parameters over a video sequence.

In order to simulate the behavior of a combined sensor capable of measuring reflected light and depth, a high resolution and a ToF cameras are mounted in a rigid stereo set-up, as illustrated in Figure 5.1-a. The relation between the two views is then found by performing a standard stereo calibration with the amplitude or offset image, moving a checkerboard in front of the cameras. To give a depth estimate to each pixel in the high resolution image, we use an inverse weighted distance interpolation. This method gives a depth value $\lambda(p)$ to pixel with coordinates p based on a weighted average of M available depth values λ_i at neighboring positions p_i ($1 \leq i \leq M$) :

$$\lambda(p) = \sum_{i=0}^{M} \frac{w_i(p)}{\sum_{j=0}^{M} w_j(p)} \lambda_i. \tag{5.1}$$

where weights w_i are computed as $w_i(p) = \frac{1}{\text{dist}(p,p_i)^d}$ (we use $d = 1$). The number of neighbors M is determined by the number of depth values being projected to a fix window size around p, 30×30 in our experiments. This interpolation allows us to handle the uneven distribution of the depth values when projected to the high resolution image.

5.3. Problem Statement: Feature-based SLAM

Our goal is to solve the SLAM problem using the combined HR-ToF sensor described above. We follow the standard formulation of feature-based SLAM based on dynamic systems modeling the motion of the sensor and the measurement process. In [20], image measurements are obtained by establishing feature correspondences in the images over time. Depth measurements are computed using the camera geometry (calibration) and its estimated motion. We introduce an extension, where the depth estimates are directly obtained from the combined HR-ToF sensor. The extension includes the depth in the state vector, and modifies the measurement model and the observation update (see details in section 5.4). In the following, we state the SLAM problem formally and we explain the extension to depth measures.

SLAM is the problem of localizing an object (here a camera) in a map that is being simultaneously built. In the dynamic system formulation of SLAM [24, 7, 20], the position of both the sensor (camera) x and the map Y are modeled to be state vectors that evolve over time. The map is considered to be a collection of N features, whose 3D positions at time k are denoted $y_{i,k}$ and grouped in a vector:

$$Y_k = \begin{pmatrix} y_{1,k} & y_{2,k} & \dots & y_{N,k} \end{pmatrix}^\top. \tag{5.2}$$

The state vector of the camera x_k in the k-th frame is composed of the camera 3D position r^w, orientation (in quaternions) q^{wc}, velocity v^w and angular velocity ω^c, that is:

$$x_k = \begin{pmatrix} r^w & q^{wc} & v^w & \omega^c \end{pmatrix}^\top, \tag{5.3}$$

where superscripts indicate the world (w) and camera (c) coordinate systems.

The evolution of the two state vectors, the camera position x_k and the map Y_k, is modeled in terms of two dynamic systems. First, the *camera motion model*, that determines the state vector of the camera x_k at instant k by means of a function f. Second, the *measurement model*, that models the measurement process by means of a function h that relates the actual feature positions Y_k to their measurements $Z_k = \begin{pmatrix} z_{1,k} & z_{2,k} & ... & z_{N,k} \end{pmatrix}^\top$. The two dynamic systems take the form:

$$x_k = f(x_{k-1}, u_k) + w_k, \tag{5.4}$$

$$Z_k = h(x_k, Y) + v_k. \tag{5.5}$$

Eq. 5.4 predicts the camera position at time k as a function of its previous state x_{k-1}, an optional input u_k and a motion disturbance w_k modeled with a zero-mean Gaussian distribution with covariance Q_k. In Eq. 5.5, $h(x_k, Y)$ models the measurement process, and v_k is the measurement disturbance modeled again with a zero-mean Gaussian distribution with covariance R_k. Notice that in the case of the combined HR-ToF sensor, h needs to take in account the depth measures, as explained in section 5.4.

Given the above formulation, the SLAM problem becomes by finding the estimates of the full state vector, composed of the camera and the map state vectors $[\hat{x}_k \ \hat{Y}_k]^\top$ where $\hat{}$ is used to denote variable estimates.

5.4. Proposed Method: HR-ToF SLAM

One common solution to the problem is to estimate the state vector by means of an Extended Kalman Filter (EKF) [118]. Kalman based algorithms compute the estimate of the state vector describing the camera $\hat{x}_k$ and the feature positions $\hat{Y}_k$ in two recursive steps: a prediction (time-update) and a correction (measurement-update). As an auxiliary outcome of the Kalman filtering, a covariance matrix P is obtained representing the uncertainty of each estimation:

$$P = \begin{pmatrix} P_{xx} & P_{xY} \\ P_{Yx} & P_{YY} \end{pmatrix}. \tag{5.6}$$

In sections 5.4.1 and 5.4.2 we describe the updates, the instantiation of the motion-model and the extension introduced to the measurement model, in order to consider the combined HR-ToF images.

5.4.1. Time Update

As described in [20], the *time-update* upgrades the camera position at time k given conditions at time $k-1$. More precisely, the updated state of the camera $\hat{x}_{k|k-1}$ is first predicted based on its

motion during previous frames. The covariance matrix corresponding to the camera position P_{xx} is updated accordingly. The time update is resumed in the following equations:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k), \tag{5.7}$$

$$P_{xx,k|k-1} = \nabla f \cdot P_{xx,k-1|k-1} \cdot \nabla f^\top + Q_k, \tag{5.8}$$

where ∇f is the Jacobian of function f evaluated at time $k-1$. The explicit dynamic model of the camera motion is $\hat{x}_{k|k-1} = f(\hat{x}_{k-1}, u_k) =$

$$\begin{pmatrix} r^w_{k|k-1} \\ q^{wc}_{k|k-1} \\ v^w_{k|k-1} \\ \omega^c_{k|k-1} \end{pmatrix} = \begin{pmatrix} r^w_{k-1} + (v^w_{k-1} + \Delta v^w) \cdot \Delta t \\ q^{wc}_{k-1} \times q((\omega^c_{k-1} + \Delta\omega^c) \cdot \Delta t) \\ v^w_{k-1} + \Delta v^w \\ \omega^c_{k-1} + \Delta\omega^c \end{pmatrix}, \tag{5.9}$$

where $q((\omega^c_{k-1} + \Delta\omega^c) \cdot \Delta t)$ is the angle change due to angular velocity in the quaternion representation. The velocities change per time step Δt is modeled as:

$$\Delta v^w = a^w \cdot \Delta t, \tag{5.10}$$

$$\Delta\omega^c = \alpha^c \cdot \Delta t, \tag{5.11}$$

where the acceleration a^w and the angular acceleration α^c are modeled as processes of Gaussian distribution and zero mean (refer to [20] for details).

5.4.2. Observation Update

The observation-update describes the new position of both the camera $\hat{x}_{k|k}$ and the map $\hat{Y}_k$ given newly observed feature positions Z_k in the combined HR-ToF images at time k, and the updated $\hat{x}_{k|k-1}$. The positions are upgraded according to the error in the prediction $Z_k - h(\hat{x}_{k|k-1}, \hat{Y}_{k-1})$ and the optimal Kalman gain matrix W_k:

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{Y}_k \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{Y}_{k-1} \end{bmatrix} + W_k \left[Z_k - h(\hat{x}_{k|k-1}, \hat{Y}_{k-1}) \right], \tag{5.12}$$

$$P_{k|k} = P_{k|k-1} - W_k \cdot S_k \cdot W_k^\top, \tag{5.13}$$

where S_k is the innovation or residual covariance matrix representing the uncertainty of the prediction at time k (see section 5.4.2 for computation details). Notice that here, Z_k is the collection of measurements obtained with the combined HR-ToF sensor, *i.e.*$z_{i,k} = (u_i \ v_i \ \lambda_i)^\top$, where λ_i is the depth value at image coordinates u_i, v_i, and $1 \leq i \leq N$.

In order to relate the 3D features Y_k to the image (2D) and the depth measurements contained in Z_k, the pinhole camera model with an invertible distortion model is used. The position of the features in the image plane is computed by projecting the expected positions $h(\hat{x}_{k|k-1}, \hat{Y}_{k-1})$ of the 3D features Y_{k-1} given the updated camera position $\hat{x}_{k|k-1}$ and using the intrinsic and distortion parameters. Let h^c_i be the result of the prediction $h(\hat{x}_{k|k-1}, \hat{y}_{i,k-1})$ in the camera coordinate system for a feature i, with $1 \leq i \leq N$, then the coordinates of the projection of $\hat{y}_{i,k-1}$ are:

$$u_u = u_0 - K_u \cdot \frac{h^c_{i_x}}{h^c_{i_z}}, \qquad v_u = v_0 - K_v \cdot \frac{h^c_{i_y}}{h^c_{i_z}}, \tag{5.14}$$

where (u_u, v_u) is the image coordinates of the features without distortion, (u_0, v_0) are the coordinates of the principal point, and (K_u, K_v) are the focal lengths in each direction. An invertible distortion model [20] is used, so that the distorted (real) image positions (u_d, v_d) are found with the expressions:

$$u_d = \frac{u_u - u_0}{\sqrt{1 + 2 \cdot k_1 r^2}} + u_0, \qquad v_d = \frac{v_u - v_0}{\sqrt{1 + 2 \cdot k_1 r^2}} + v_0, \tag{5.15}$$

where $r = \sqrt{(u_u - u_0)^2 + (v_u - v_0)^2}$ is the radial distance from the center of the image, using undistorted coordinates.

Once the coordinates of the projections of each feature are computed, the error between the expected and the measured positions $Z_k - h(\hat{x}_{k|k-1}, \hat{Y}_{k-1})$ are used to correct the Kalman filter prediction applying Eq. 5.12 to update the camera position $\hat{x}_k$ and map $\hat{Y}_k$.

The difference to the state vectors and measurements of [20] is summarized in the following table:

MonoSLAM	HR-ToF SLAM
$z_{i,k} = (u_i \ v_i)^\top$	$z_{i,k} = (u_i \ v_i \ \lambda_i)^\top$
$h_i = (u_{d_i} \ v_{d_i})^\top$	$h_i = (u_{d_i} \ v_{d_i} \ \lambda_i)^\top$

where $\lambda_i = \sqrt{{h^c_{i_x}}^2 + {h^c_{i_y}}^2 + {h^c_{i_z}}^2}$ is the depth of the image position (u_i, v_i) associated to feature i.

Innovation Formula

The innovation formula is used to update (correct) the Kalman filter model, that is to calculate the Kalman Gain W_k and the innovation S_k in Eq. 5.12 and Eq. 5.13. The correction is computed from the difference between the measured and predicted positions, the Jacobians of the projections ∇h and the measurement noise R_k. Recall from Sect. 5.4.2 that the innovation covariance matrix S_k represents the uncertainty of the predictions ($x_{k|k-1}$ and the set of h_i) at time k and depends on the uncertainty of both the camera x_k and map Y_k (contained in P) as well as the measurement noise R_k. The updates to the EKF are computed as follows:

$$W_k = P_{k|k-1} \cdot \nabla h^\top \cdot S_k^{-1}, \tag{5.16}$$

$$S_k = \nabla h \cdot P_{k|k-1} \cdot \nabla h^\top + R_k. \tag{5.17}$$

Given the block form of P in Eq. 5.6, the 3×3 sub-matrices of S_k associated to feature i, *i.e.*$S_{k,i}$ can be obtained using:

$$S_{k,i} = \frac{\delta h_i}{\delta x} P_{xx} \frac{\delta h_i}{\delta x}^\top + \frac{\delta h_i}{\delta x} P_{xy_i} \frac{\delta h_i}{\delta y_i}^\top + \frac{\delta h_i}{\delta y_i} P_{y_i x} \frac{\delta h_i}{\delta x}^\top + \frac{\delta h_i}{\delta y_i} P_{y_i y_i} \frac{\delta h_i}{\delta y_i}^\top + R_{k,i}. \tag{5.18}$$

To calculate $R_{k,i}$ we model the noise associated to the measurement of each feature. The model takes into account that several sources of noise affect the HR and ToF cameras. First, we model the noise of the HR image position R_{uv} with a linear radial function to take in account the radial distortion, such that the noise increases when the measurement is further away from the center of the image. The image position noise model is then:

$$R_{uv} = \sigma_{uv}^2 \cdot \left(1 + \frac{r}{\max r}\right), \tag{5.19}$$

where σ_{uv} determines the minimum level of noise (here assigned to the image center) and the factor $\frac{r}{\max r}$ controls the increase in the noise factor as the coordinates u and v go away from the center.

Second, the noise of the depth measurements is modeled as a linear function with respect to the depth itself, as far away the depth measures are usually noisier. Similar to 5.19 the model also takes into account the radial error. However, this time the linear model reflects the concentration of the infrared light in the middle of the image that causes noisier measurements in the borders of the image. Additionally, we use the amplitude image provided by the ToF camera as final indicator of the measurement noise, as it is known that measurements computed from low amplitudes (when the amount of reflected light is low) are noisier. In practice, we employ a function $\sigma_\lambda = \sigma_{min} + (1 - A_{uv})$, where σ_{min} ensures a minimum level of noise and A_{uv} is the corresponding amplitude value. Thus, the depth noise model is resumed in the expression:

$$R_\lambda = \sigma_\lambda^2 \left(1 + \cdot \frac{\lambda}{\max \lambda} \cdot \frac{r}{\max r}\right). \tag{5.20}$$

Finally, we assume independence between the noise of the depth information and the image position. The resultant noise matrix R has the following form:

$$R_{k,i} = \begin{pmatrix} R_{uv} & 0 & 0 \\ 0 & R_{uv} & 0 \\ 0 & 0 & R_\lambda \end{pmatrix} \tag{5.21}$$

Using the proposed noise model, the innovation matrix S_k can be computed. In particular, the 3×3 block elements of the Jacobian in Eq. 5.18 takes the form:

$$\frac{\delta h_i}{\delta y_i} = \begin{pmatrix} \frac{\delta u_u}{\delta y_{i_x}} & \frac{\delta u_u}{\delta y_{i_y}} & \frac{\delta u_u}{\delta y_{i_z}} \\ \frac{\delta v_u}{\delta y_{i_x}} & \frac{\delta v_u}{\delta y_{i_y}} & \frac{\delta v_u}{\delta y_{i_z}} \\ \frac{\delta \lambda}{\delta y_{i_x}} & \frac{\delta \lambda}{\delta y_{i_y}} & \frac{\delta \lambda}{\delta y_{i_z}} \end{pmatrix}. \tag{5.22}$$

The first two rows in the expression above are the same used in [20], however the third row contains the derivatives of the new measurement value λ, namely:

$$\frac{\delta \lambda}{\delta y_{i_x}} = \frac{y_{i_x}}{||y_i||}, \qquad \frac{\delta \lambda}{\delta y_{i_y}} = \frac{y_{i_y}}{||y_i||}, \qquad \frac{\delta \lambda}{\delta y_{i_z}} = \frac{y_{i_z}}{||y_i||}$$

where, $||y_i|| = \sqrt{y_{i_x}^2 + y_{i_y}^2 + y_{i_z}^2}$. Using this noise model and computing W_k and S_k according to Eqs. 5.16 and 5.17, the EKF is updated and the system is ready for iteration $k + 1$.

5.4.3. Feature extraction and depth initialization

In standard monocular approaches to SLAM, the initialization of the features depth is difficult, as it is not possible to measure the real distance from a feature to the camera. The usual way to initialize the features is to fix the depth of a reduced set of features, and to find a depth estimate for any other feature taking into account the motion of the camera. In the particular case of MonoSLAM [20], the depth of a new feature is initialized as a probabilistic uniform distribution on a finite ray passing through both the projection of the feature in the image plane and the origin of the camera coordinate system. This probability distribution is updated with each new frame using a particle filter that weights each discrete value of depth according to the intersection of the current and previous backprojected rays, under the estimated motion of the camera. In the case of PTAM [57], an initial translational motion is used to recover the depth of the initial features using stereo disparities.

In our case the initialization becomes easier, as the probabilistic estimation is simply replaced with the depth measurement provided by the ToF device in the location where image features are detected. We use SIFT [75] to detect the features in the high-resolution image and use the first position of the camera as the origin of the world coordinate system.

5.5. Experimental Validation

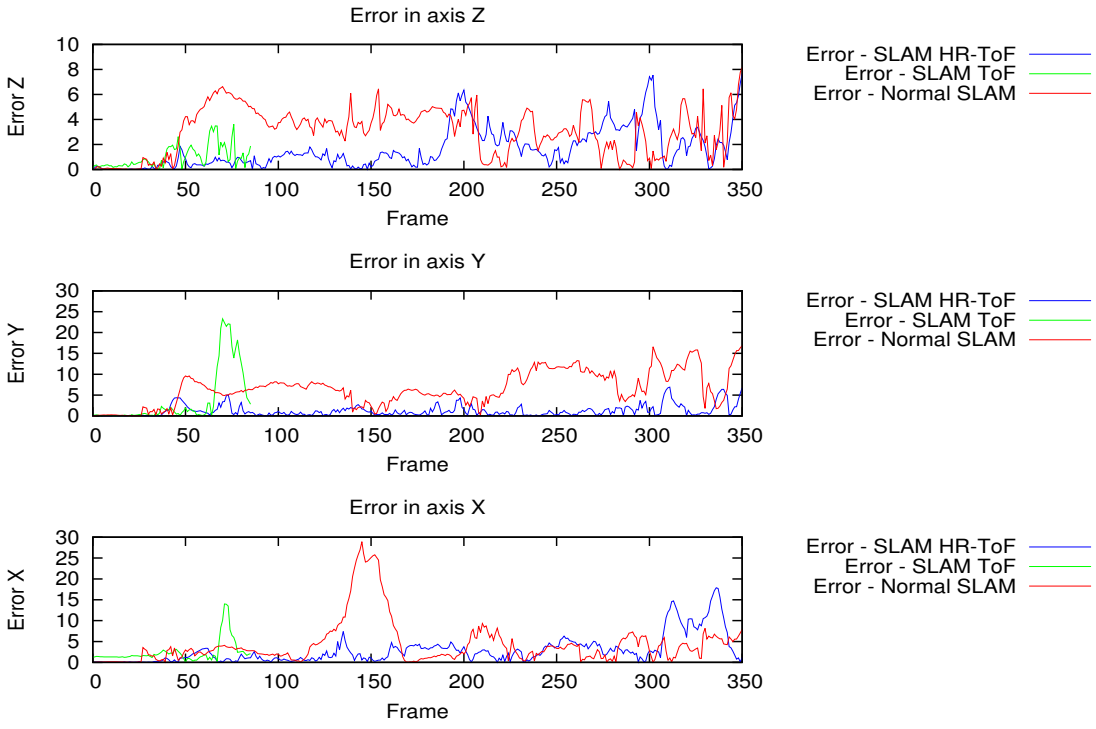


Figure 5.2.: 3D error (X_{axis},Y_{axis} and Z_{axis}) in cms for each camera pose of the trajectory of the HR SLAM (red line), HR-ToF SLAM (blue line) and ToF SLAM (green line) versus the ground truth.

We recorded a video sequence of the camera moving in an office environment, where the distance of the camera to the objects ranges from 1 to 3 meters. We used a Point-Grey Flea2 HR camera (640×480) and a PMD CamCube2 (204×204) ToF camera in a stereo setup as shown in Figure 5.1-a. The cameras are calibrated and synchronized.

In order to validate the performance of our approach we measure the error of the camera position. To capture the ground truth trajectory for the evaluation we place infrared markers on the camera and track them (using a commercial Tracking system). The markers are registered to the camera coordinate system with an additional calibration step that uses a checkerboard equipped with infrared markers. The resultant ground truth trajectory provides the position and orientation of the camera at each frame.

	HR-ToF (cms)	ToF (cms)	HR (cms)
X_{axis}	2.966 ± 3.312	2.426 ± 2.723	4.212 ± 5.187
Y_{axis}	1.632 ± 2.509	3.824 ± 6.628	6.657 ± 4.090
Z_{axis}	2.015 ± 2.190	1.018 ± 0.913	3.138 ± 1.895
Total	2.204 ± 2.670	2.450 ± 3.422	4.669 ± 3.724

Table 5.1.: Mean and standard deviation of the camera localization error using the proposed approaches HR-ToF SLAM and ToF SLAM, and the monoSLAM algorithm applied individually to the high resolution (HR) camera.

We measure the error of the estimated camera trajectories with respect to their ground truth. Three trajectories are compared. The first is obtained with the proposed HR-ToF SLAM method. The second is the result of our method when using only the ToF images (the HR image is replaced by the low-resolution offset image), here named ToF SLAM. Finally, the third trajectory is the estimated pose of the camera obtained with the MonoSLAM algorithm on the high resolution images (HR-SLAM). As shown in Fig. 5.2, the proposed HR-ToF SLAM method (blue line) follows very closely the ground truth (error near to zero). ToF SLAM (green line) rapidly looses track due to the difficulty of reliably tracking features in the low-resolution images and to infrared specular reflections. Finally, HR-SLAM (red line) is able to keep the track but leads to a less accurate trajectory. Error peaks observed in the HR-ToF SLAM and HR-SLAM are explained by fast camera movements. The overall mean and standard deviation errors (using the magnitude of the error in the three axis) over the sequence with 380 frames are reported in Table 5.1. The camera trajectory using the proposed extension with the combined HR-ToF camera has the lowest average error ($\sim$2.2 cms). Despite the low resolution of the offset images, our method applied to the ToF images along ($\sim$2.4 cms) still has a lower average error compared to the MonoSLAM with the high-resolution camera ($\sim$4.6 cms). The 3D error in the graph is calculated at each frame in the camera coordinate system, in order to observe the error in the depth (Z_{axis}), and in the horizontal (X_{axis}) and vertical (Y_{axis}) axes.

To start building the maps we initialize the depth of the detected features in the first frame using the values provided by the ToF data (*c.f.*section 5.4.3). This is done for the initialization of the map in the three compared methods in order to avoid the need of knowing an object dimensions for the HR-SLAM. We manually select four features in the first image to create a common world coordinate system. The chosen features are required by the HR SLAM, as it needs a reasonable number of features for initialization. Although not required in our approach, we also provide the same four features to HR-ToF and to ToF SLAM for the sake of evaluation. Finally, we choose the world coordinate system to be the first frame position of the camera, this allows us to relate the camera coordinate system to the ground truth.

Snapshots in Fig. 5.3 shows the difference between the uncertainty of the feature position estimates. The proposed HR-ToF method has less uncertainty than MonoSLAM applied to the high-resolution video. The difference is explained by the use of the ToF depth measurements.

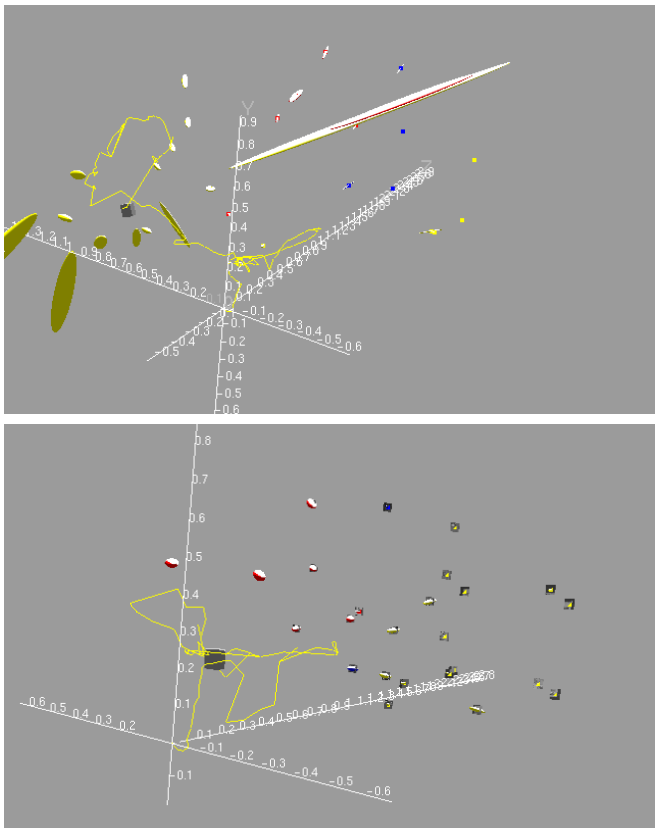


Figure 5.3.: Snapshot of SLAM results for frame 380 of the sequence, based only on the HR camera (top) and on the combined HR-ToF sensor (bottom). The uncertainty in the depth data is displayed as ellipsoids. Lower uncertainties are achieved with the HR-ToF SLAM method.

Part IV.

Multi-view Time-of-Flight

6. Stereo Time-of-Flight - An Example of Multi-view ToF

THis chapter presents the Multi-view ToF system, using the case of stereo ToF as an example. An introduction of the multi-view ToF system, is given, including the most important related works. Also are presented, an explanation of the proposed method and its depth optimization for a stereo set-up. Additionally, a quantitative evaluation from simulated ToF images and the results of a pair of real ToF cameras are exhibited. Furthermore, a radiometric analysis and the limitations of the system are shown. Besides, this Chapter incorporates the explanation of how to extend the proposed method from the two cameras to a multi-view ToF system. Finally, the electronic circuit for the stereo ToF system and its specifications, including its schematic and the list of commands, for the communication between the computer and the electronic circuit, are presented.

6.1. Introduction

Time of Flight (ToF) cameras are active range sensors that provide depth images at a high frame-rates, as it was explained before. They are equipped with an infrared (IR) light source that illuminates the scene, and a CMOS/CCD sensor that captures the reflected infrared light. The depth is measured based on the time of flight principle, *i.e.*it is proportional to the time spent by the IR signal to reach the scene and come back. Depth measurements are obtained for each pixel of the sensor, and together produce a depth image. Fast acquisition of depth images is of great use in a wide range of applications, *e.g.*in robotics, human machine interaction and scene modeling [62]. Unfortunately, the available ToF cameras have a low resolution and are affected by different measuring errors [72].

These include noise caused by the sensor; the systematic *wiggling* error due to the difficulty of generating sinusoidal signals; a non-linear depth offsets dependent on reflectivity and integration-time; and the flying pixels generated by the superposition of signals at depth inhomogeneities (edges). As a result, the uncertainty of the ToF depth measurement is important (in order of centimeters).

Several approaches have been proposed that target the improvement of the depth measurements, including different ways to calibrate the ToF camera [72, 11, 108, 32, 102], combining ToF cameras with a single or a stereo RGB cameras [48, 37, 127, 10, 59], or using a sequence of depth images to improve the resolution [18, 123]. Also exists some methods that combine the depth images of several ToF cameras to create 3D reconstructions [56]. Another work recover better accuracy by using a reflection model and optimizing the surface to the model and measurement [15].

In this work, we focus on a different approach to improve the acquisition of depth images using a multi-view system of ToF cameras, instead of the state of the art (Chapter 3) which focus in

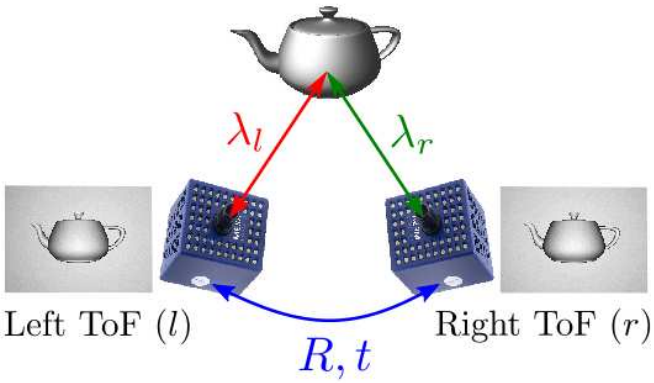


Figure 6.1.: Stereo ToF: two calibrated ToF cameras acquire measurements under different IR lighting conditions. The measurements are optimized to recover more accurate depth images.

calibration and/or 3D reconstruction. Our method relies on a calibrated multi-ToF configuration and on an active control of the infrared lights. To explain the main idea, we use the case of two cameras as illustrated in Figure 6.1. We devise an acquisition, where we alternatively turn on and off the lighting of the cameras, and acquire measurements in each lighting state. Then, we propose to optimize the depth images in each camera based both on the measurements gathered during three stages and the geometry of the stereo setup.

We provide, a quantitative evaluation based on a simulation of the ToF cameras [53] under different levels of noise and for varying geometric configurations of the cameras, as well as experiments with real images, both showing the improvement on the accuracy of the depth measurements. Furthermore, the limitations and a radiometric analysis are presented, as well as the real implementation of the system.

6.2. Theory

Consider a calibrated stereo setup (as example) such as the one in Figure 6.1 which use exactly the same modulation frequency and the same IR light wavelength. We propose a stereo ToF acquisition, where a series of measurements are taken with the two cameras while the infrared lighting of the scene is actively changed. Our goal is to find the optimal depth image in each camera, based on these measurements and on the known geometry of the stereo setup. The three lighting stages (shown in Figure 6.2) are:

Stage 1: Only the emitter of the left camera is active and *both* cameras capture the reflected light. Each camera provides three images: depth, amplitude and offset.

Stage 2: Only the emitter of the right camera is active and *both* cameras capture the reflected light (similar to stage 1 but changing the emitter).

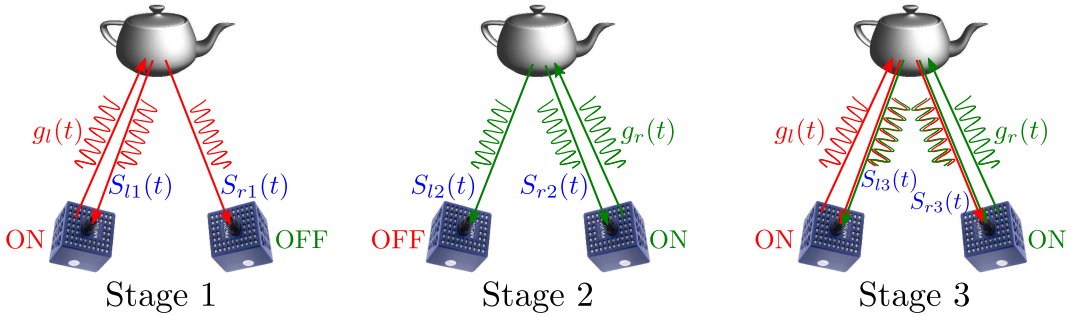


Figure 6.2.: The three stages Stereo ToF acquisition.

Stage 3: The two lights emit simultaneously an IR signal with the exact same modulating frequency[1] and *both* cameras capture the reflected light. The amount of light received in each sensor is equivalent to the superposition (interference) of the received signals when each IR light is independently active.

We assume that the scene is static during the three stages, that the cameras work with the same IR wavelength and that their modulating frequency is the same. Additionally, the stereo configuration should be setup so that enough light is reflected into the left and right cameras in order to make valid measurements.

We now formally describe how to recover the parameters of the received signals in the three described stages. Consider the sinusoidal signals g_l and g_r used to modulate the emitted IR light of the left and right ToF cameras respectively. We denote with ω the common modulation frequency of the two emitted signals, and with ϕ_{lr} the phase shift between them. Then,

$$g_l(t) = A_l \cdot \cos(\omega \cdot t) + B_l, \tag{6.1}$$

$$g_r(t) = A_r \cdot \cos(\omega \cdot t + \phi_{lr}) + B_r. \tag{6.2}$$

After reflection on the scene, signals S_l and S_r are received in the left and right cameras. As we detail next, these signals have a different form in the three stages. In each case, we aim at recovering the *amplitudes* $A'_{l,r}$ and $A''_{l,r}$, the *offsets* $B'_{l,r}$ and $B''_{l,r}$, and the phase-shifts; where a single $'$ indicates the reflected signal is captured with the same camera emitting the light, and a double $''$ indicates the receiving camera is different from the emitting one. As before, the parameters are obtained by sampling the convolution of the received ($S_{l,r}$) and the reference ($g_{l,r}$) signals.

6.2.1. Stage 1

Let only the light of the left camera be active and emit signal g_l (Equation 6.1), while both cameras capture the reflected light on the scene. The received signals in the left and right ToF sensors,

[1] This is necessary since small differences in frequency lead to destructive interference. One way to ensure that both cameras have *exactly* the same modulation frequency is to interconnect their clock and start signals.

denoted S_{l1} and S_{r1}, have the following form:

$$S_{l1}(t) = A'_l \cdot \cos(\omega \cdot t + \varphi_l) + B'_l \tag{6.3}$$

$$S_{r1}(t) = A''_r \cdot \cos(\omega \cdot t + \frac{\varphi_l + \varphi_r}{2} + \phi_{lr}) + B''_r. \tag{6.4}$$

We seek to recover the parameters of the two signals, *i.e.*the amplitudes (A', A''), offsets (B', B''), and phases (φ_l, $\frac{\varphi_l+\varphi_r}{2}$). Notice that in Equation 6.4, the phase shift $\frac{\varphi_l+\varphi_r}{2}$ is related to the distance traveled by the signal from the left camera to the reflecting surface, and then from the surface back to the right camera. The total phase of S_{r1}, $\frac{\varphi_l+\varphi_r}{2} + \phi_{lr}$, additionally considers the phase shift ϕ_{lr} between the emitted signals $g_l(t)$ and $g_r(t)$.

Similar to the monocular case, we use samples of the correlation between the received and reference signals in each ToF camera, which results in the following expressions:

$$\begin{aligned} C_{l1}(\tau) &= g_l(t) \otimes S_{l1}(t) \\ &= \frac{A'_l A_l}{2} \cdot \cos(\omega \cdot \tau + \varphi_l) + B_l B'_l \end{aligned} \tag{6.5}$$

$$\begin{aligned} C_{r1}(\tau), &= g_r(t) \otimes S_{r1}(t) \\ &= \frac{A''_r A_l}{2} \cdot \cos(\omega \cdot \tau + \frac{\varphi_l + \varphi_r}{2} + \phi_{lr}) + B_l B''_r. \end{aligned} \tag{6.6}$$

Using samples of $C_{l1}(\tau)$ and $C_{r1}(\tau)$ at times $\tau_0 = 0$, $\tau_1 = \frac{\pi}{2\omega}$, $\tau_2 = \frac{3\pi}{2\omega}$, $\tau_3 = \frac{\pi}{\omega}$, and Equations. 3.9 to 3.10, we recover the parameters of S_{l1} and S_{r1} per pixel and in each camera:

Left camera: we calculate the amplitude A'_l, offset B'_l and phase φ_l from the samples of $C_{l1}(\tau)$. Using $\lambda_l = \frac{c}{2\omega} \cdot \varphi_l$ (Equation 3.1) we obtain a first depth estimate per pixel.

Right camera: from $C_{r1}(\tau)$'s samples we compute the phase $\xi_1 = \frac{\varphi_l+\varphi_r}{2} + \phi_{lr}$ and the values of A''_r and B''_r.

6.2.2. Stage 2

We invert the role of the cameras w.r.t.to Stage 1. Now, only the right camera emits a signal $g_r(t)$. To recover the parameters of the received signals $S_{l2}(t)$ and $S_{r2}(t)$:

$$\begin{aligned} S_{l2}(t) &= A''_l \cdot \cos(\omega \cdot t + \frac{\varphi_l + \varphi_r}{2} - \phi_{lr}) + B''_l, \\ S_{r2}(t) &= A'_r \cdot \cos(\omega \cdot t + \varphi_r) + B'_r, \end{aligned}$$

we sample the correlations $C_{l2}(\tau)$ and $C_{r2}(\tau)$:

$$\begin{aligned} C_{l2}(\tau) &= g_l(t) \otimes S_{l2}(t), \\ &= \frac{A''_l A_r}{2} \cdot \cos(\omega \cdot \tau + \frac{\varphi_l + \varphi_r}{2} - \phi_{lr}) + B_r B''_l. \end{aligned} \tag{6.7}$$

$$\begin{aligned} C_{r2}(\tau) &= g_r(t) \otimes S_{r2}(t), \\ &= \frac{A'_r A_r}{2} \cdot \cos(\omega \cdot \tau + \varphi_r) + B_r B'_r. \end{aligned} \tag{6.8}$$

With these relations we compute:

Left camera: the values of $\xi_2 = \frac{\varphi_l+\varphi_r}{2} - \phi_{lr}$, A''_l, and B''_l based on $C_{r2}(\tau)$.

Right camera: the values of A'_r, φ_r and B'_r using $C_{l2}(\tau)$. From φ_r a first depth estimate $\lambda_r = \frac{c}{2\omega} \cdot \varphi_r$ is computed.

Until here, we can optimize the surface captured by the cameras, using 4 measurement and 2 stages. This is named optimization using only two stages.

6.2.3. Stage 3

In the third stage, the lights of the left and right cameras emit simultaneously signals $g_l(t)$ and $g_r(t)$, and both cameras capture the total amount of reflected light. The received signals in the left ($S_{l3}(t)$) and right ($S_{r3}(t)$) cameras are of the form:

$$\begin{aligned} S_{l3}(t) =& A'_l \cdot \cos(\omega \cdot t + \varphi_l) + B'_l + \\ & A''_l \cdot \cos(\omega \cdot t + \frac{\varphi_l + \varphi_r}{2} - \phi_{lr}) + B''_l, \\ S_{r3}(t) =& A'_r \cdot \cos(\omega \cdot t + \varphi_r) + B'_r + \\ & A''_r \cdot \cos(\omega \cdot t + \frac{\varphi_l + \varphi_r}{2} + \phi_{lr}) + B''_r. \end{aligned}$$

Convolving the received signals with the reference signals in each camera leads to:

$$\begin{aligned} C_{l3}(\tau) =& g_l(t) \otimes S_{l3}(t) = \frac{A'_l A_l}{2} \cdot \cos(\omega \cdot \tau + \varphi_l) + B_l B'_l + \\ & \frac{A''_l A_r}{2} \cdot \cos(\omega \cdot \tau + \frac{\varphi_l + \varphi_r}{2} - \phi_{lr}) + B_r B''_l, \\ C_{r3}(\tau) =& g_l(t) \otimes S_{r3}(t) = \frac{A'_r A_r}{2} \cdot \cos(\omega \cdot \tau + \varphi_r) + B_r B'_r + \\ & \frac{A''_r A_l}{2} \cdot \cos(\omega \cdot \tau + \frac{\varphi_l + \varphi_r}{2} + \phi_{lr}) + B_l B''_r. \end{aligned}$$

In stage 3, there is no closed form solution to find the values of φ_l, φ_r and ϕ_{lr}. Instead we use directly the samples of $C_{l3}(\tau)$ and $C_{r3}(\tau)$ as explained next.

6.3. Depth optimization

To optimize the depth images, we define a per-pixel cost function that simultaneously considers the measurements acquired during the three stages as well as the geometry of the stereo setup. In the following we describe the cost for optimizing the depth estimate of a pixel $\hat{\lambda}_l$ in the left camera (the process is analogous for the right image).

First we consider the cost in each camera. In the left image, the depth estimate $\hat{\lambda}_l$ at pixel $\mathbf{x}_l$, is expected to lie close to the measurement λ_l obtained in stage 1. The cost penalizing this error is defined as:

$$E_l = [\hat{\lambda}_l - \lambda_l]^2.$$

A similar error is calculated in the right image, where the measured depth $\tilde{\lambda}_r$ should agree with the current depth estimate after a geometric transformation T_l^r that converts it to a valid depth in the right image $T_l^r(\hat{\lambda}_l)$ (See Figure 6.3). The measurement $\tilde{\lambda}_r$ is taken at location $\hat{\mathbf{x}}_r$ in the right image, where $\hat{\mathbf{x}}_r$ is the projection of the 3D point $\hat{\Lambda}_l$ obtained by backprojecting $\hat{\lambda}_l$. Thus, the cost in the right image is:

$$E_r = [T_l^r(\hat{\lambda}_l) - \tilde{\lambda}_r]^2$$

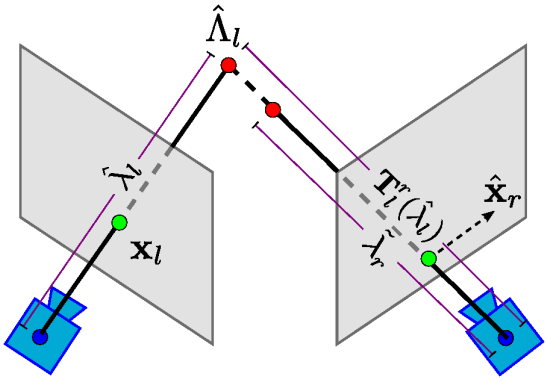


Figure 6.3.: The optimization relies on the stereo geometry.

For the path traveled by the IR light from one camera to the other, we are considering a phase sum $\zeta_1 = \xi_1 + \xi_2 = \varphi_l + \varphi_r$ from phases ξ_1 and ξ_2 in stages 1 and 2. We define an additional cost E_{lr} which penalizes the difference between the measured depth $\frac{2\omega}{c}\zeta_1$ and its equivalent estimate, *i.e.*:

$$E_{lr} = [\hat{\lambda}_l + T_l^r(\hat{\lambda}_l) - \tfrac{2\omega}{c}\zeta_1]^2$$

Finally, we consider the following relations among the measurements, which hold in the absence of noise:

$$C_{l3}(\tau) = C_{l1}(\tau) + C_{l2}(\tau), \tag{6.9}$$

$$C_{r3}(\tau) = C_{r1}(\tau) + C_{r2}(\tau). \tag{6.10}$$

We use the current depth estimate $\hat{\lambda}_l$ to compute estimates of the measurements $\hat{C}_{l1}(\tau)$, $\hat{C}_{l2}(\tau)$, $\hat{C}_{r1}(\tau)$ and $\hat{C}_{r2}(\tau)$. This is done by replacing φ_l and φ_r in Equations 6.5-6.8, by $\hat{\varphi}_l = \frac{2\omega}{c}\hat{\lambda}_l$ and by $\hat{\varphi}_r = \frac{2\omega}{c}T_l^r(\hat{\lambda}_l)$, for the 4 values of τ. The phase difference $\phi_{l,r}$ is a fixed value calibrated in advance or 0 if the cameras are synchronized. Once these estimates are computed, we compare them to the measurements $C_{l3}(\tau)$ and $C_{r3}(\tau)$ according to Equations 6.9-6.10:

$$E_C = \sum_{\tau} \left[C_{l3}(\tau) - \hat{C}_{l1}(\tau) - \hat{C}_{l2}(\tau)\right]^2 + \sum_{\tau} \left[C_{r3}(\tau) - \hat{C}_{r1}(\tau) - \hat{C}_{r2}(\tau)\right]^2, \tag{6.11}$$

where $\tau \in \{0, \frac{\pi}{2\omega}, \frac{3\pi}{2\omega}, \frac{\pi}{\omega}\}$. In summary, the optimal depth $\hat{\lambda}_l^*$ is found by minimizing the cost function:

$$\mathcal{J} = A_l' E_l + A_r' E_r + \rho_1 E_{lr} + \rho_2 E_C \tag{6.12}$$

where the first two terms have been weighted by a confidence value, obtained from the amplitudes of the received signals, A_l' and A_r''. Similarly $\rho_1 = \frac{A_l'' + A_r''}{2}$. Finally, the last two terms are multiplied by a constant weight ρ_2.

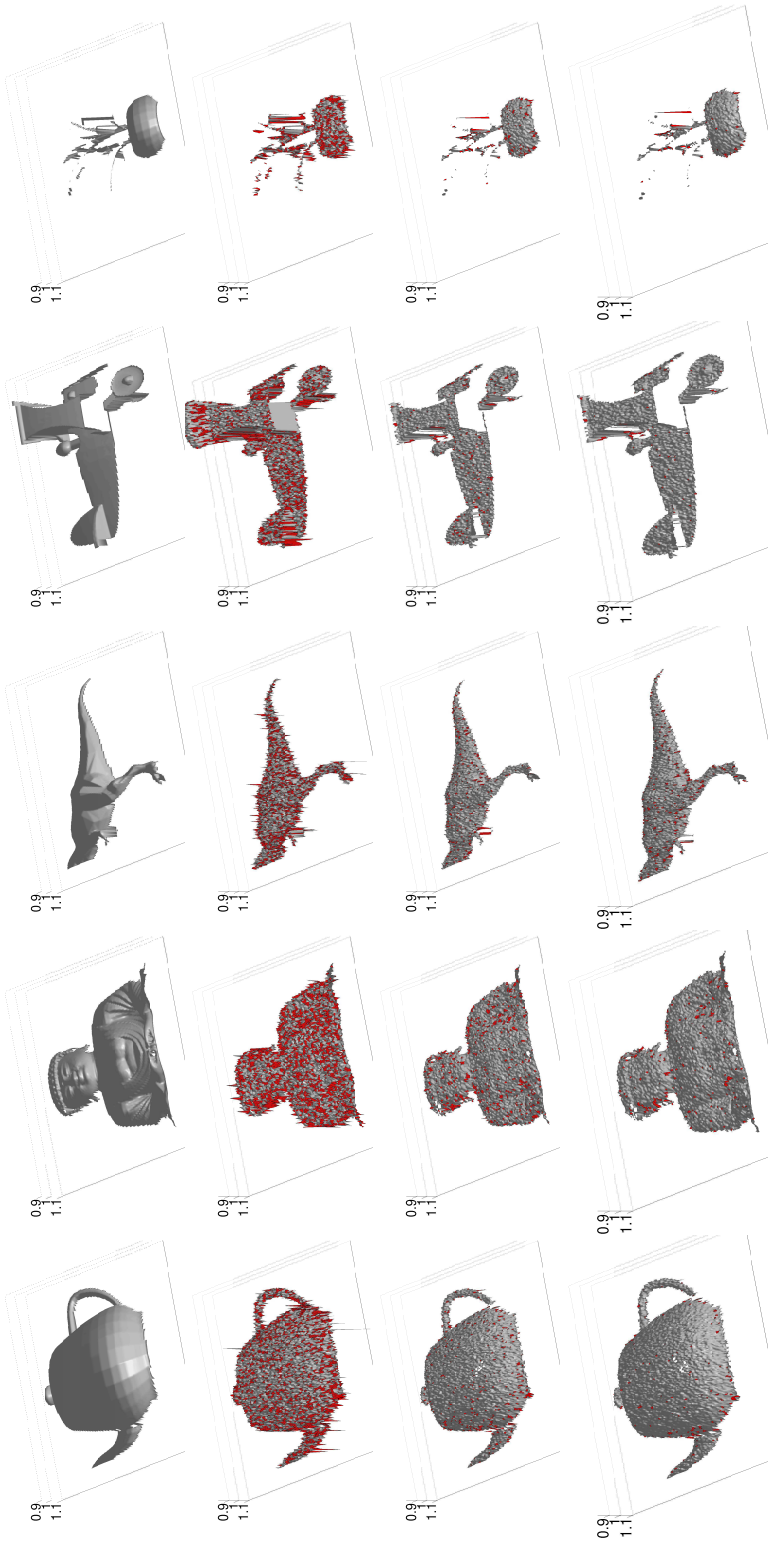


Figure 6.4.: Comparison of the depth images recovered with a single ToF camera and with the proposed stereo ToF approach (Only images from the left camera are shown). (top) Ground truth images. (2nd row) Depth images obtained with a single ToF camera and a level of noise of 0.05%. (3rd row) Depth images obtained with the proposed stereo ToF using only 2 stages. (bottom) Depth images with the 3 stages of the stereo ToF. Red points on the surface show errors greater than 0.3cm.

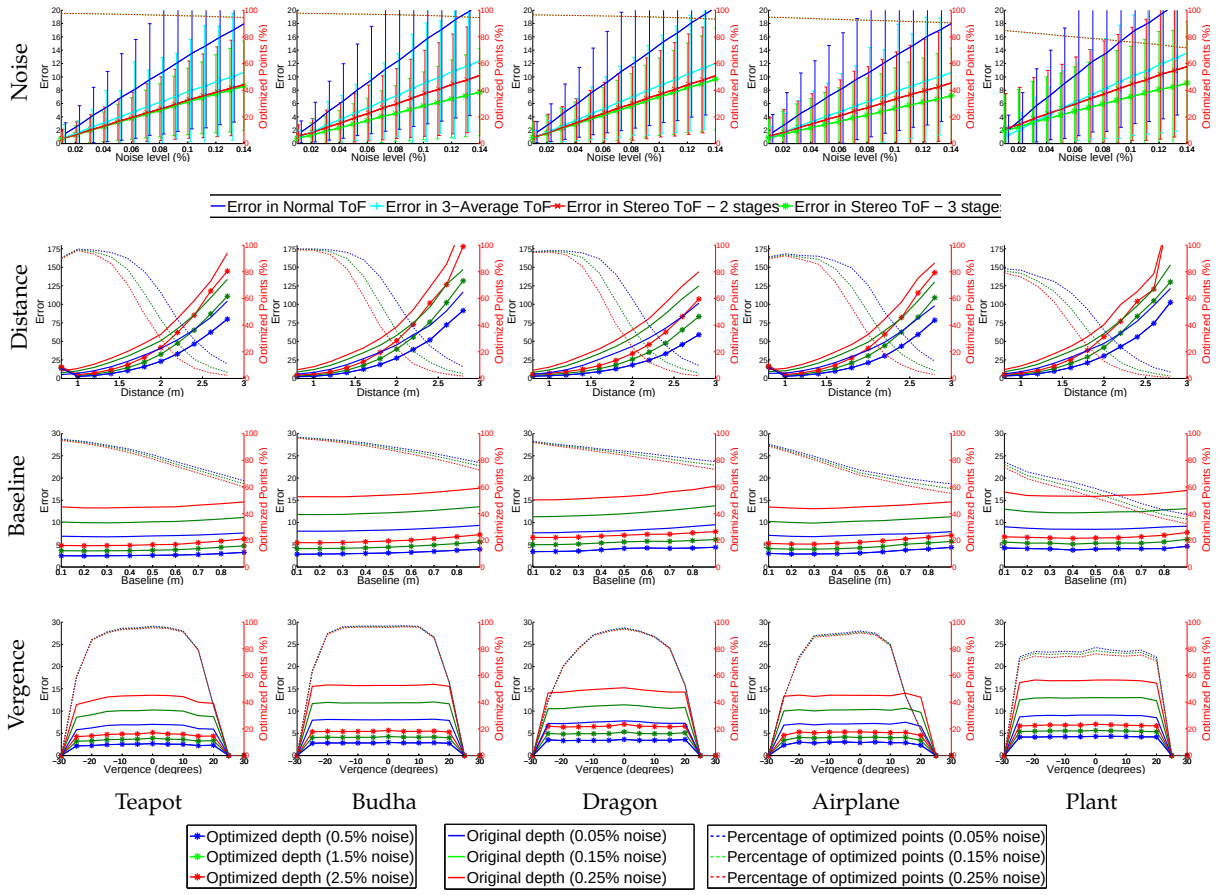


Figure 6.5.: Evaluation of the depth error (in mm) and percentage of optimized points for ToF camera and monocular ToF camera w.r.t.the ground truth against changes in the noise level, distance to the object, baseline and vergence for different objects.

The minimization is performed individually for every pixel in the image and we optimize the left and right depth maps separately. The optimization is solved using gradient descent. Initial values are obtained from the measurements in stage 1 and 2, λ_l and λ_r. Because each pixel is optimized individually, it is easy to parallelize the computations increasing the frame rate of the three-stage acquisition.

Handling occlusions and outliers. We test the visibility of every pixel in both cameras and skip occluded pixels from the optimization. To detect occluded pixels, all depths from one camera are converted to 3D points and projected to the second camera. If several points project to the same pixel in the second camera, only the foremost point (the closest to the camera) is considered valid, all points behind are marked as occluded (See Figure 6.6). Also, only pixels in the field of view of one of the two cameras are optimized. In the case of depth measurements with big errors, initial estimates for the depths will be far from their optimal value. Therefore we use the divergence of the optimization in a given pixel as an indicator of an outlier measurement.

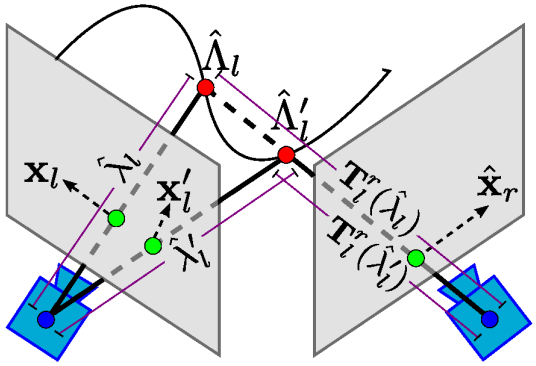


Figure 6.6.: Example of occlusion handling.

6.4. Experimental Validation

In the following we provide a quantitative evaluation based on a simulation of the stereo ToF system (Section 6.4.1), and qualitative results with real depth images (Section 6.4.2).

6.4.1. Experiments with simulated ToF images

In order to quantitatively validate the proposed approach, we simulated a pair of ToF cameras relying on the work of Keller *et al.* [53]. The simulation uses a point light-source and a Lambertian reflection model with a non-linear attenuation of the signal w.r.t.the depth. The depth noise affects directly each measurement $C_i \in \{C_{l1}, C_{r1}, C_{l2}, C_{r2}, C_{l3}, C_{r3}\}$ and is modeled as $\tilde{C}_i = \alpha\gamma + (1+\beta)C_i$, where γ is a zero-mean Gaussian noise and $\alpha = 1$ and $\beta = 0.00035$ as suggested in [53]. We assume the radial distortion and systematic depth errors have been corrected in advance. Finally, we consider a rigid stereo setup with a phase shift $\phi_{lr} = 0$ and set $\rho = 100$.

Using the stereo-ToF simulator, we generate amplitude, offset and depth images of different 3D models including a teapot, a Budha, a dragon, an airplane and a plant. For each object we evaluate

the accuracy of the recovered depths for increasing the levels of noise. We further analyze the performance of the approach under dissimilar configurations of the stereo setup (changing the baseline and vergence[2]) and for different depths of the object. For each configuration, we consider three different levels of noise and perform 10 experiments per level. We report a reduction of the depth error for all the configuration using the ToF stereo. To obtain the error, we calculate the mean over all the optimized pixels and for the 10 experiments. Finally, we also compute the percentage of optimized pixels w.r.t.the total number of foreground pixels. This percentage is important for the analysis of the results as the number of foreground pixels depends on the size of the object and its distance to the camera. Example depth images in Figure 6.4, show the improvement of the optimized depth surfaces (using 2 and 3 stages) w.r.t.the original ToF depth images, where pixels with large errors are depicted in red. The noise is reduced (as evidenced by a fewer red pixels) using 3 stages rather than 2. Also all configuration exhibit a reduction of noise compared to the 3-images average. For 3 stages one can also observe an improved behavior on flat or smooth surfaces due to both cameras receiving higher amounts of light, allowing to recover better scene details. We summarize the quantitative results for the different configurations and noise levels in Figure 6.5. Graphs are explained in details bellow.

Noise level: In this experiment we fix the stereo configuration to a baseline of 10cms and a vergence of $0°$. The object is located at 1m from the camera. The depth error is analyzed for different noise levels, from 0.01% to 0.14% of the maximum grayscale variation that the sensor can measure, here 2^{16} (16 bits per pixel)[3]. As shown in the graphs, the mean error and standard deviations using the ToF stereo are significantly reduced w.r.t.the originally noisy monocular depth images and 3-images average. The percentage of optimized pixels decreases for higher levels of noise, mainly due to the noisier initial values handed to the optimization (outliers). The third stage not only increases the accuracy and number of optimized pixels, but also improves the results in curvature discontinuities, see for instance Fig 6.4-Budha.

Distance to target: In this experiment the distance between the observed object and the camera is changed from 0.8m to 3m. Standard values for the baseline (10cms) and the vergence ($0°$) are used. The experiment shows that as the distance to the observed object increases, the percentage of optimized points decreases. This is natural as the noise also tends to increase with the distance, generating worse initial values for the optimization. The depth error increases with the distance but the percentage of correction w.r.t.the original noisy image is similar for the different values of noise and distance. Bellow 80 cms, there is a drop in the percentage of optimized points, because there is a smaller number of common pixels in the two cameras (the object lies very close to the camera and the vergence is $0°$). For the last object representing a plant the percentage of optimized pixels is lower due to the significant amount of depth discontinuities that generate large noise values in the measured images.

Baseline: At a distance of 1m from the object, the baseline of the stereo setup is varied (from 10 to 90cms) and the vergence is automatically adjusted such that the principal rays of the cameras

[2]The vergence is the deviation angle of the principal ray of each camera from a line perpendicular to the baseline passing through the camera center. Negative values indicate cameras look towards the interior of the setup.

[3]Remember the noise is applied to the source images C_i, thus the corresponding error in depth depends on the amount of received light. In particular, due to the attenuation, for the same level of noise in C_i, the noise in the depth increases exponentially w.r.t.distance of the camera to the object.

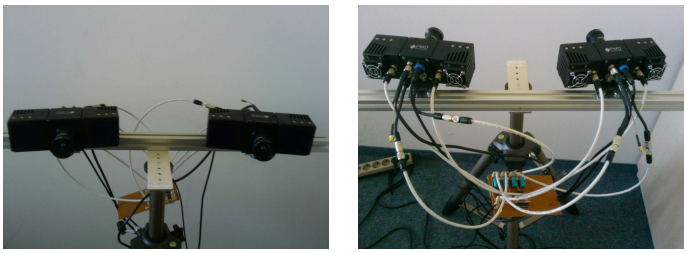


Figure 6.7.: Real set-up of Stereo Time-of-Flight (including the electronic circuit).

point to the center of the observed object. For the tested objects, the improvement of the stereo ToF is only slightly affected by changes in the baseline. However, the quantity of optimized pixels decreases due to the cameras having less common pixels for larger baselines.

Vergence: Using a baseline of 10cms and locating the object at 1m from the setup, we vary the vergence of the two cameras. The improvement in the optimized pixels remains constant for vergences around $0°$. However, the percentage of optimized pixels depends on the number of pixels visible simultaneously in the two views. In the case of very low or high vergences, the cameras have very few or no common pixels, resulting in small percentage of the optimized pixels. The behavior of the stereo-ToF according to the vergence also depends on the observed object, because high curvature surfaces may generate occlusions which also have an effect on the number of common pixels visible in the two views.

6.4.2. Experiments with real images

We performed experiments with a real ToF stereo setup (using 2 and 3 stages) for different scenes. The real set-up is shown in Figure 6.7. We show a selection of the results in Figures 6.8 and 6.9. For the planar surface, details of the board are better observed in the two optimized images. The transversal view shows a reduction of the noise with the stereo ToF w.r.t.a single depth image (the plane looks flatter). Notice, that using 3 stages improves the details of the plug on the wall. For the experiments with the shelves and books one can observe enhancements in the borders and frontal faces. In the kitchen scenes, the noise of the cups and the tablet are reduced leading to smoother surfaces. The optimized part of the teapot is smoother using 3 stages rather than 2. In the case of the chair, the noise in the chair surface is reduced, but the multi-path error provoked by the floor was not improved. For the body phantom, the surface is also smoother. Additionally, in all the cases, the stereo setup allows detecting and eliminating the pixels which are occluded or inconsistent between the two views (shown in gray). In general, the optimization using 3 stages recovers more details, further reduces the noise and results in more pixels being optimized pixels than when using only 2 stages. One advantage of the third stage is the increased amount of emitted light which reduces the uncertainty of the measurements and thus has a higher signal-to-noise ratio. To avoid sensor saturation it is important to adjust the integration time of the camera according to the distance to the scene.

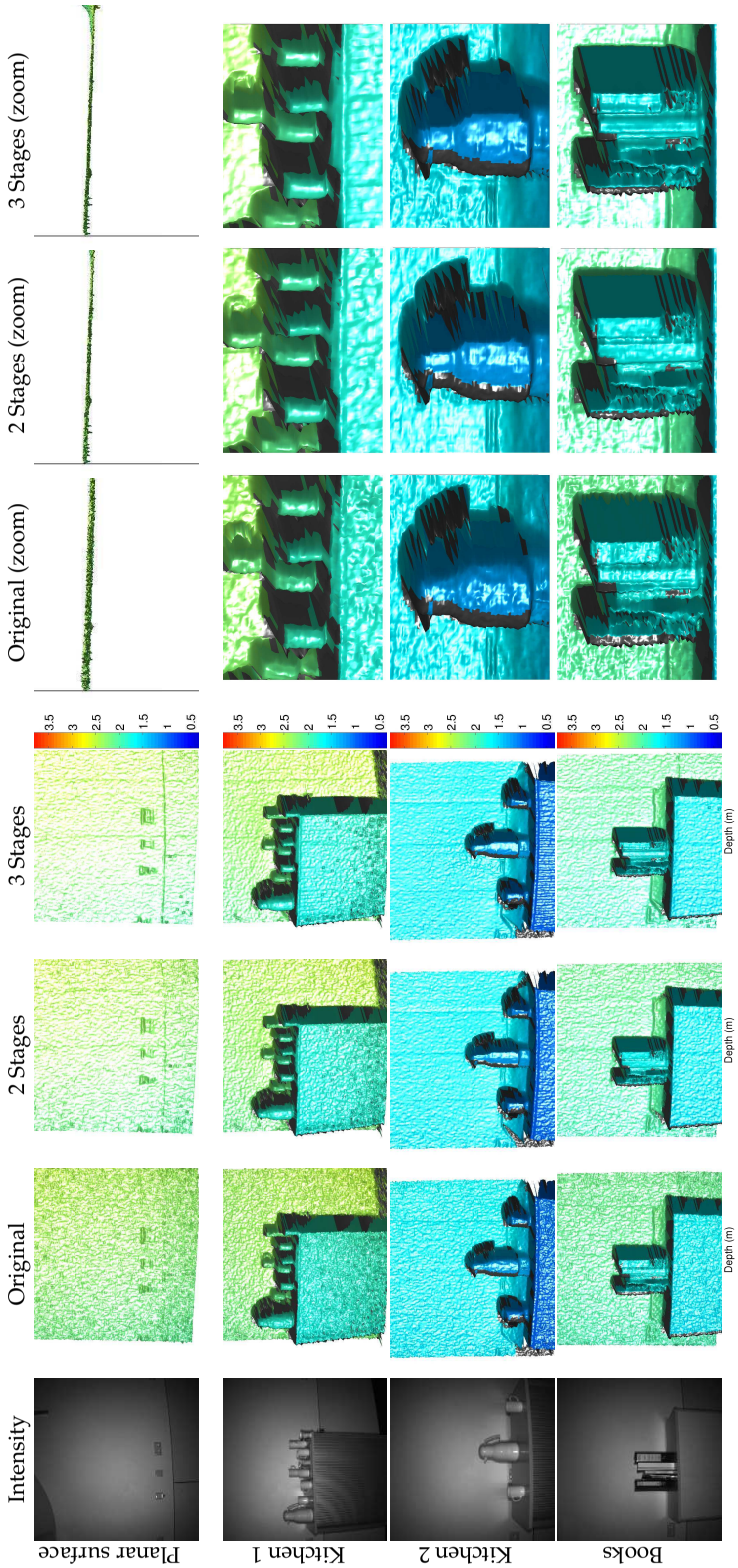


Figure 6.8.: Comparison of real depth images acquired with a monocular and the proposed ToF stereo approach. Only left images shown.

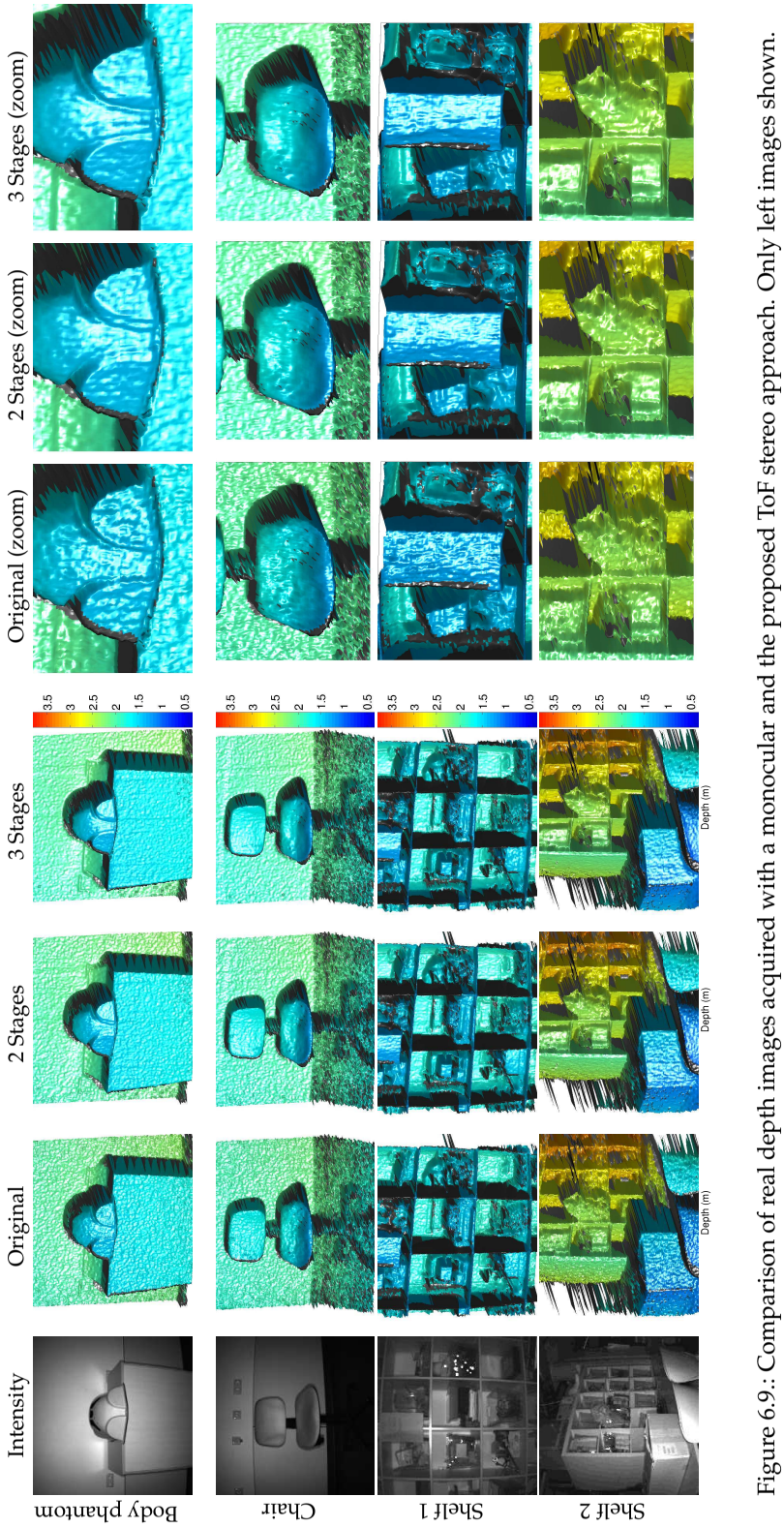


Figure 6.9.: Comparison of real depth images acquired with a monocular and the proposed ToF stereo approach. Only left images shown.

6.5. Limitations

The proposed algorithm have some limitations regarding to the reachable improvement using the stereo ToF. The presented method cannot enhance the inherent error of the Time-of-Flight camera physics, *i.e.*, multi-path artifact or high uncertainty by a low signal-to-noise ratio. Furthermore, when the initial depth is far from the optimal depth, the method tends to diverge. These diverged pixels are marked as outlier measurements. In addition, the 3D points which are not observed by both cameras cannot be optimized due to the occlusion. These occluded points are labeled as occluded points. The labeled pixels as occluded point or outlier measurement can be eliminated from the final results or filled with the original ToF camera depth (as we show our results in gray scale).

Another limitation is related to the acquisition time which is extended to three shots, *i.e.*, three times longer than only one ToF shot. Therefore, the scanned object has to be static in the acquisition process, so it does not produce motion artifacts.

Finally, the last limitation is regarding to the interference[4]. To ensure a constructive interference in the third stage is necessary to calculate the maximum phase-shift between two general sinusoidal signals, *i.e.*, the superposition of the signals does not produce a smaller amplitude besides the original ones.

Let's assume that we have two sinusoidal signals at the same frequency:

$$g_1(t) = A_1 \cos(\omega t) \tag{6.13}$$

$$g_2(t) = A_2 \cos(\omega t + \phi) \tag{6.14}$$

The superposition of the signals (interference) is $g_1(t) + g_2(t)$. Now, our goal is to calculate the maximum ϕ allowed to generate a constructive interference. For that, we have to calculate the amplitude of $g_{12} = g_1(t) + g_2(t)$, which cannot be smaller than the biggest amplitude between $g_1(t)$ and $g_2(t)$, $max(A_1, A_2)$.

In order to find the maximum ϕ, it is necessary to determine the amplitude of the function $g_{12}(t)$. To do so, we differentiate the function $g_{12}(t)$ and find $(\omega t)_{max}$ which satisfies $\frac{\delta g_{12}(t)}{\delta \omega t} = 0$. The differentiation of $g_{12}(t)$ is:

$$\frac{\delta g_{12}(t)}{\delta \omega t} = -A_1 \sin(\omega t) - A_2 \sin(\omega t + \phi) \tag{6.15}$$

Solving for $\frac{\delta g_{12}(t)}{\delta \omega t} = 0$, we obtain $(\omega t)_{max} = \arctan\left(\frac{-A_2 \sin(\phi)}{A_2 \cos(\phi) + A_1}\right)$.

Then, to calculate the amplitude of the function $g_{12}(t)$, we have to replace ωt by $(\omega t)_{max}$:

$$g_{12}((\omega t)_{max}) = A_1 \cos\left(\arctan\left(\frac{-A_2 \sin(\phi)}{A_2 \cos(\phi) + A_1}\right)\right) + \tag{6.16}$$

$$A_2 \cos\left(\arctan\left(\frac{-A_2 \sin(\phi)}{A_2 \cos(\phi) + A_1}\right) + \phi\right) \tag{6.17}$$

The value $g_{12}((\omega t)_{max})$ represents the maximum or minimum value of the function $g_{12}(t)$ and thus, the amplitude of $g_{12}(t)$.

[4]It refers to the interference produced between the sinusoidal signals and not between the light beams

The amplitude of $g_{12}(t)$ has to be at least $max(A_1, A_2)$. Depending on the case, the $max(A_1, A_2)$ can be A_1 or A_2. Then, solving ϕ for $g_{12}((\omega t)_{max}) = A_1$ and $g_{12}((\omega t)_{max}) = A_2$, we have:

$$\phi_{max} = \pi - \arccos\left(\frac{A_2}{2A_1}\right) \text{ if } A_1 > A_2 \tag{6.18}$$

$$\phi_{max} = \pi - \arccos\left(\frac{A_1}{2A_2}\right) \text{ if } A_2 > A_1 \tag{6.19}$$

For example, if $A_1 = 1$ and $A_2 = 1$, we obtain $\phi_{max} = \frac{2\pi}{3}$. Another example, if $A_1 = 1$ and $A_2 = \frac{\sqrt{3}}{2}$, then $\phi_{max} = \frac{5\pi}{6}$.

The limit case of $A_1 >> A_2$ or $A_2 >> A_1$, then the ϕ_{max} tends to $\frac{\pi}{2}$. Therefore, we can say that, in general, the phase-shift between the signals cannot exceed $\frac{\pi}{2}$ to assure a constructive interference in any case. This constraint ensures that the interfered signal has a larger amplitude than the maximum amplitude of the original sinusoidals $max(A_1, A_2)$. Figure 6.10 shows the behavior of the maximum delay to guarantee a constructive interference for an object to a certain distance. As an example, it was plotted an object distance of two, four and six meters from the stereo system. This plot assumes that $A_1 \propto \frac{1}{\lambda_l^2}$ and $A_2 \propto \frac{1}{\lambda_r^2}$ (because the amplitude is attenuated as Equation 6.21 and more details in Appendix B). The minimum phase-shift is plotted for every object distance (black line). In conclusion, a value of $\phi_{max} = \frac{\pi}{2}$ ensures a constructive interference for any object distance.

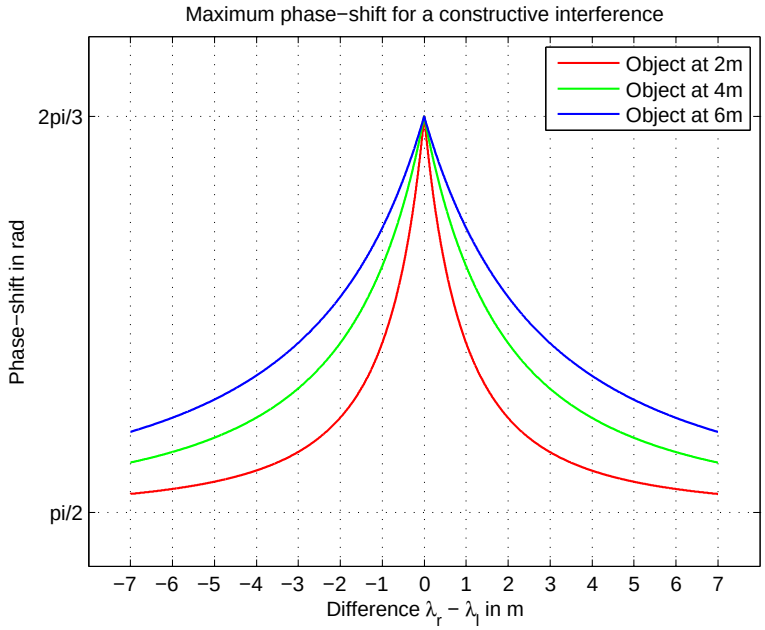


Figure 6.10.: Maximum phase-shift allowed to obtain a constructive interference between the emitted signals. Plot lines have been made with different object distance.

Using the value of $\phi_{max} = \frac{\pi}{2}$, we know that any delay between the cameras should not exceed $\frac{\pi}{2}$ and it guarantees a constructive interference. Now, it is necessary to see which constraints

generates in the case of stereo ToF. In this case, the received data, for example in the left camera, is $\frac{A_l' A_l}{2} \cdot \cos(\omega \cdot \tau + \varphi_l) + B_l B_l' + \frac{A_l'' A_r}{2} \cdot \cos(\omega \cdot \tau + \frac{\varphi_l + \varphi_r}{2} - \phi_{lr}) + B_r B_l''$. The maximum phase-shift between $\frac{A_l' A_l}{2} \cdot \cos(\omega \cdot \tau + \varphi_l) + B_l B_l'$ and $\frac{A_l'' A_r}{2} \cdot \cos(\omega \cdot \tau + \frac{\varphi_l + \varphi_r}{2} - \phi_{lr}) + B_r B_l''$ has to be $\frac{\pi}{2}$. Therefore, the difference of phase-shift $\frac{\varphi_l + \varphi_r}{2} - \phi_{lr} - \varphi_l$ cannot surpass $\frac{\pi}{2}$, *i.e.*, $\frac{\varphi_l + \varphi_r}{2} - \phi_{lr} - \varphi_l < \frac{\pi}{2} \rightarrow \frac{\varphi_r - \varphi_l}{2} - \phi_{lr} < \frac{\pi}{2}$.

Assuming a known delay ϕ_{lr} between the cameras, the maximum difference of measured distances $\lambda_r - \lambda_l$ between the cameras has to be $\frac{c_{light}}{2\pi f}\left(\frac{\pi}{2} - \phi_{lr}\right)$ to assure a constructive interference. For example, if $\phi_{lr} = 0$ (cameras are synchronized), the maximum difference $\lambda_r - \lambda_l$ cannot exceed 3.75m. Practically, this difference is difficult to achieve it and then, the improvement of the stereo ToF can be attempted in the most of the cases when $\phi_{lr} = 0$. Furthermore, if the object distance changes, the improvement can be obtained in the most of the pixels, due to the maximal difference of 3.75m. However, if we include the baseline as the delay ϕ_{lr} between the cameras, the stereo ToF improvement can be achieved using a maximum baseline of 3.35m (for a full range of measurement of $0 - 7$ meters and vergence $0°$). Larger baselines create a destructive interference for far objects. Figure 6.11 exhibits the maximum field of view, that allows a constructive interference for different object distances and baselines. This plot shows the field of view for different object distances and the baseline was included as the delay ϕ_{lr} between the cameras (it is assumed that the cable between the cameras produces a delay between the emitted signals). This plot assumes that $A_1 \propto \frac{1}{\lambda_l^2}$ and $A_2 \propto \frac{1}{\lambda_r^2}$ (due to the attenuation of the amplitude of the signal traveling in the air). The maximum baseline of 3.35m was found to assure a constructive interference, for the maximum object distance of 7m and for the maximum ToF camera's field of view of $40°$ (for the specifications of a PMD CamCube 2.0).

6.6. Radiometric Analysis

A radiometric analysis of the working principles of the proposed multi-ToF system, is presented, specifically, the case of stereo ToF. In this proof, we use a point light source, a ray light and a Lambertian reflection model on the surface.

Given two cameras, C_l and C_r, in stereo set-up with a baseline b meters and a vergence α degrees, we want to analyze the received signal in both cameras when only the camera left emits IR light (it represents the stage 1 in the proposed method in Section 6.2). We assumed a point on the surface S has an unitary normal $\hat{N}_S$. The analogous procedure can be applied to the cases of stage 2 and 3. An illustration of the set-up and the observed surface is shown in Figure 6.12. We illustrated three different surface points S_1, S_2, S_3 to analyze the behavior of the received signal for these distinct cases.

First of all, we assume a Lambertian reflection model of the surface, which follows the next equation:

$$I_r = C_R(\hat{w}_i \cdot \hat{N}) I_i \tag{6.20}$$

where C_R is the reflectivity, $\hat{w}_i$ is the unitary direction vector of the incident light ray, $\hat{N}$ is the unitary direction vector of the surface normal and I_i is the incident light intensity. The reflected light by the Lambertian model is the same in all directions.

We assume a quadratic light attenuation in the air as follows:

$$I_a = \frac{I}{\lambda^2}, \tag{6.21}$$

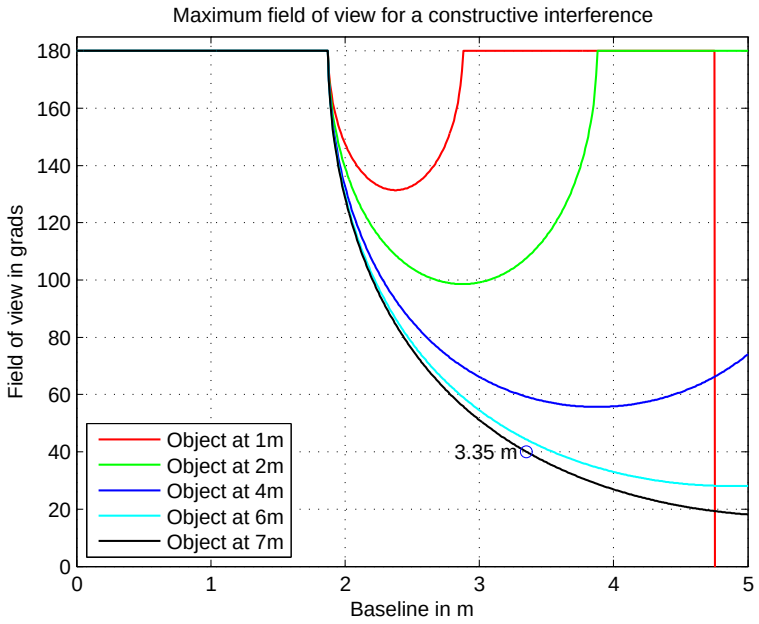


Figure 6.11.: Maximum field of view to have a constructive interference between the emitted signals. It is assumed the baseline as the delay ϕ_{lr} between the cameras.

where I_a is the attenuated intensity light, I is the original intensity light and λ is the traveled distance.

Using this two basic equations, we analyze the quantity of light which each camera receives while camera left C_l is emitting a signal. Assuming a point ray light and Lambertian reflection model on the surface, the incident light in the point S from camera in C_l is as follows:

$$I_{Si} = \frac{C_R(\hat{w}_i \cdot \hat{N})I_e}{\lambda_{l_S}^2} \tag{6.22}$$

where λ_{l_S} is the distance between the camera C_l and the surface point S, I_e is the emitted light, C_R is the reflectivity of the surface, $\hat{N}$ is the normal direction and $\hat{w}_i$ is the incident light direction.

The equation 7.4 gives the quantity of reflected light in the surface point S. Lambertian reflection model says that the same quantity of light is reflected in all directions. If we compute how much light is acquiring each sensor, we obtain the next equations (including the attenuation):

$$\text{Camera left: } I_{C_l} = \frac{C_R(\hat{w}_i \cdot \hat{N})I_e}{\lambda_{l_S}^4} \tag{6.23}$$

$$\text{Camera right: } I_{C_r} = \frac{C_R(\hat{w}_i \cdot \hat{N})I_e}{\lambda_{l_S}^2 \lambda_{r_S}^2} \tag{6.24}$$

where λ_{r_S} is the distance between the camera C_r and the surface point S.

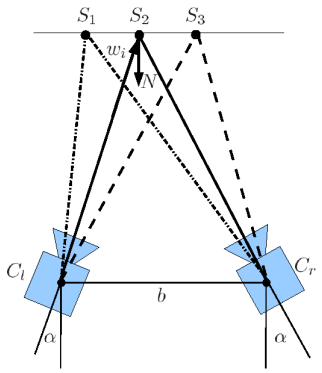


Figure 6.12.: Set-up of two cameras (C_l and C_r) with a baseline b, a vergence α are observing three points on a surface S (S_1, S_2, S_3) with its surface normal N.

We can observe that the amount of received light in the cameras depends, not only, on the reflectivity C_R of the surface and dot product between the incident direction $\hat{w}_i$ and surface normal $\hat{N}$, but also on the distance between the surface point S and the cameras. We assume that the reflectivity on the surface and the incident light direction are constant for the following calculations. For instance, if we compare the difference between the reception of each camera, we can see, that the only difference is the attenuation which is related to the traveled distance from the surface point to the camera. Analyzing the Equations 6.23-6.24, we can observe that if the surface point S is nearer to the camera C_r than the camera C_l, then, the camera C_r will receive a bigger amount of light than the camera C_l. On the other hand, if the surface point S is nearer to the camera C_l, then the amount of the received light in the camera C_l is bigger than in camera C_r.

To understand better the behavior of this formula, we will analyze three different points in a planar surface S_1, S_2, S_3. The first point S_1 shows that the received signal in the camera C_l is bigger than the second camera C_r, due to the point S_1 is nearer to camera C_l than to camera C_r. On the other hand, in the case of the surface point S_3, the camera C_r receives more light than the camera C_l, because the camera C_r is closer to the surface point than then camera C_l. Finally, in the case of the surface point S_2, both cameras receive the same quantity of light (both cameras are at the same distance to the surface point).

In conclusion, the received light in each camera is totally dependent on the distance of each camera to the surface point S, *i.e.*, the observed surface point S determines the quantity of light received in each sensor. Furthermore, the light received in each camera depends on the distances and do not depends directly on the baseline and vergence. Actually, the baseline b and vergence α modifies the common area observed by the cameras, which are the pixels that the system can optimize (See Figure 6.13). Therefore, the vergence and baseline have to be selected optimizing the observed scene and maintaining a baseline for a constructive interference, in order to obtain better results.

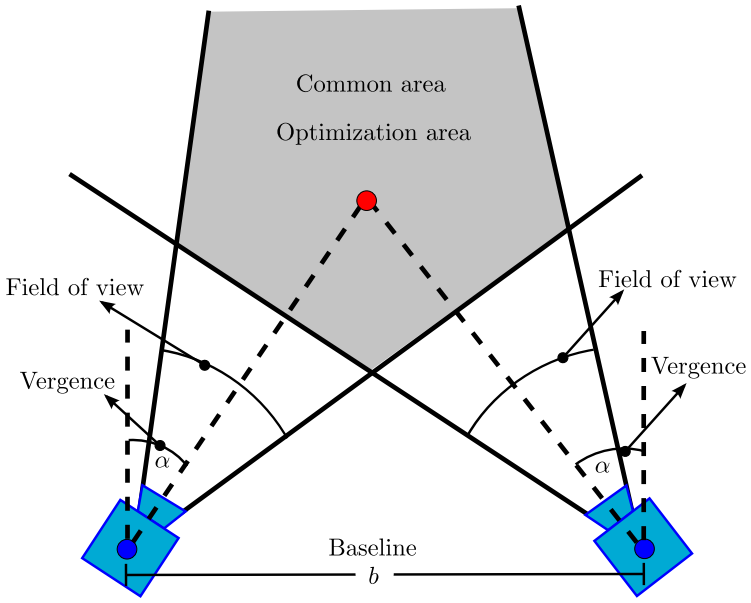


Figure 6.13.: Set-up of two cameras (C_l, C_r) with baseline b, vergence α observing three points in a surface S (S_1, S_2, S_3) with a normal N.

6.7. Extension to Multi-view ToF

This section explains how it is possible to extend the stereo ToF idea to a multi-view ToF system. First of all, having a Multi-view ToF system allows us to create a collection of pairs, which can be used as different stereo systems. Unfortunately, to use a collection of stereo ToF, is necessary to employ them at different modulation frequencies. On the other hand, if we use the cameras at the same modulation frequency, a new set of stages can be determined in order to manipulate the multi-ToF. We will make the example of three ToF cameras working at the same modulation frequency. In this case, we have to define a new stage procedure, for example for three cameras, it is possible to use 4 stages:

We assumed that the infrared light of every camera is turned on and off. All cameras are cap-

	Camera 1	Camera 2	Camera 3
Stage 1	**ON**	OFF	OFF
Stage 2	OFF	**ON**	OFF
Stage 3	OFF	OFF	**ON**
Stage 4	**ON**	**ON**	**ON**

Table 6.1.: Infrared manipulation for three cameras. In stage 1 to 3, one camera is activated in every stage, and in stage 4, all the cameras are emitting signal. All the cameras are always capturing data.

turing data in every stage. Besides, it is feasible to create a different combination of infrared light source activation. For example, we can activate every one separately and then combination of two cameras. These measurements allow to have more data to optimize, however, it is a longer procedure.

In general, the easiest way is to activate each camera separately in each stage, and then, in the final stage, all the infrared light sources are activated, obtaining a constructive interference between the signals. Then, for any case, the stages required are the number of cameras plus one stage where all the sources light are activated and thus the interference is produced. Unfortunately, when all the infrared light sources are activated, the ToF camera sensor can be saturated, and therefore, it is not possible to make any optimization. So, the number of cameras that it is possible to use are limited by the range of distance and it depends on the saturation of the sensor.

In conclusion, if we have N cameras in the multi-view ToF, the number of suggested stages is $N + 1$, generating $N(N + 1)$ measurements.

6.8. Real Implementation

In the experiments, it was used two PMD CamCube 2.0 cameras, which are setup equally, *i.e.*, at the same modulation frequency and integration time. The real set-up using these cameras is shown in Figure 6.7. In the beginning, it was tested by capturing images while the left camera emits and the right camera captures. The right camera did not capture any usable data, it was only noise. Thus, we realize that the cameras have to use **exactly** the same modulation frequency, or the camera may not recover the correct data (destructive interference). That means, the reference signal of the receiving camera has to be generated with exactly the same modulation frequency as the emitted infrared light. Unfortunately, the synchronization between the cameras requires a hardware manipulation, where the clock and start signal have to be shared. This modification is very difficult, but it ensures that both cameras work at exactly the same modulation frequency and have the same start signal (acquisition synchronized).

However, the stereo Time-of-Flight theory gets 6 measurements in these 3 stages. Then, we designed a 6-stages procedure where these 6 measurements may be obtained. The camera used in the experiments is composed of a camera unit (Figure 6.14(a)) and two illumination units (Figure 6.14(b)), which are shown in Figure 6.14. The camera and the illumination unit are connected through a special cable, which is specified by the manufacturer (pin-out shown in Figure 6.14(c)). Thanks to this cable and the camera modularity, it is feasible to connect the illumination units from the left camera to the right camera and vice-versa. This flexibility allows us to capture the 6 measurements of stereo ToF in 6 stages, connecting the camera unit to the illumination unit accordingly. These six measurements are used to optimize/calculate the depth maps in every camera, as shown in Section 6.3. One disadvantage of using these interconnections is the introduction of a new offset in the measured depth (due to the extra cable) which has to be estimated and compensated in the measurements.

(a) Camera Unit (b) Illumination Unit (c) Connector pin-out

Figure 6.14.: Camera and illumination units. Pin-out of the cable between

Stage	Left camera				Right camera			
	IU 1	IU 2	MOD SIG	CAPTURE	IU 1	IU 2	MOD SIG	CAPTURE
Stage 1	**ON**	**ON**	**ON**	**ON**	OFF	OFF	OFF	**ON**
Stage 2	OFF	OFF	OFF	**ON**	**ON**	**ON**	**ON**	**ON**
Stage 3	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**

Table 6.2.: Table of the 3 stages used in stereo ToF. It assumes that the left and right cameras have exactly the same modulation frequency. (IU: Illumination Unit, MOD SIG: Modulation signal connected to the illumination unit in state **ON** and CAPTURE: Camera capturing the scene when is in state **ON**)

6.8.1. From 3-stages to 6-stages

The experiments were made using the 6-stages procedure, which acquire the same 6 measurement made in a 3-stage procedure by the stereo ToF. The 6-stage procedure ensures that the modulation frequency of the illumination unit is exactly the same of the receiving camera unit. The constraint of the same modulation frequency is the most important issue in the stereo Time-of-Flight theory.

The stereo ToF theory is designed to work with measurement obtained in 3 stages as follows:

- **Stage 1:** The illumination units of camera left are ON and both cameras capture the scene.
- **Stage 2:** The illumination units of camera right are ON and both cameras captures the scene.
- **Stage 3:** The illumination units of both cameras are ON and both cameras captures the scene.

The next table 6.2 summarize the state of every camera and illumination unit per stage:

In other words, the stage 1 works with the idea that the left camera captures the scene as a normal ToF camera and the camera right captures the reflection of the scene in his direction. The stage 2 is the same idea as stage 1 but using the right camera. Stage 3 is working with all the illumination

units turned on and both cameras capturing. This 3 stages produces 6 measurement which can be measured separately defining 6 stages. These 6 stages are divided in order to obtain the six measurements of this 3-stage system allowing us to reproduce the stages 1,2 and 3. These 6 stages are defined as follows:

- **Stage 1:** We connect the illumination units of the left camera to the left camera as a normal ToF camera and captures the data. It simulates the stage 1 in the left camera.
- **Stage 2:** We connect the illumination units of the right camera to the left camera, with a crossed illumination unit and the right camera is measuring. It simulates the stage 2 in the right camera.
- **Stage 3:** We connect the illumination units of the right camera to the right camera as a normal ToF camera and captures the data. It simulates the stage 2.
- **Stage 4:** We connect the illumination units of the left camera to the right camera with a crossed illumination unit and the left camera left is measuring. It simulates the stage 1 in the left camera.
- **Stage 5:** We connect the illumination units of the left and right cameras to the left camera, and both cameras are capturing. It simulates the stage 3 in the left camera.
- **Stage 6:** We connect the illumination units of the left and right cameras to the right camera, and both cameras are capturing. It simulates the stage 3 in the right camera.

The stages 1-4 can be done using the extension cable (Cable 1-2, Figure 6.15(a)), because only 2 illumination units are connected to the camera unit and it can support it (the camera unit support up to 2 illumination units). The length of these cables has to be long enough to connect the illumination units of one camera to another (our case 50-60 cms). The case of stage 5-6 is different, because each camera has to connect 4 illumination units and is not possible to use only two extension cables. For this reason, the cables 3-4 (Figure 6.15(b)) are built to allow us to provide the same modulation signal to four illumination units. That means, each output connector of the camera unit could send the modulation signal to two illumination units, *i.e.*, using a splitter. However, each output connector of the camera unit can provide the enough supply power to only one illumination unit. Therefore, the cable 3-4 was designed to get the power supply from each own camera, but the modulation signal only from one camera (master). To achieve this modulation signal sharing, it is necessary to disconnect the modulation signal of the slave camera and interconnect the grounds to leave the circuit with the same ground reference.

In summary, the cable 3-4 shares the modulation signal of the camera master to one of the illumination units of camera master and one of the camera slave, gets the power supply of the illumination units from its own camera. Finally, the cable interconnects the ground reference between the cameras. Using cable 3 and 4, it is possible to emit in the four illumination units at the same modulation frequency of the master camera.

The table 6.3 shows the state of each device in each stage. The columns named IU corresponds to the illumination unit of the cameras, which is in use or not (ON or OFF). MOD SIG means modulation signal and tells if this signal is used in the IU or not (ON or OFF). Finally, the column CAPTURE indicates if the camera is capturing the scene or not (ON or OFF). Combining this features, we can

3 Stages	6 Stages	Camera Left				Camera Right			
		IU 1	IU 2	MOD SIG	CAPTURE	IU 1	IU 2	MOD SIG	CAPTURE
Stage 1	Stage 1	**ON**	**ON**	**ON**	**ON**	OFF	OFF	OFF	OFF
	Stage 2	OFF	OFF	**ON**	**ON**	**ON**	**ON**	OFF	OFF
Stage 2	Stage 3	OFF	OFF	OFF	OFF	**ON**	**ON**	**ON**	**ON**
	Stage 4	**ON**	**ON**	OFF	OFF	OFF	OFF	**ON**	**ON**
Stage 3	Stage 5	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	OFF	OFF
	Stage 6	**ON**	**ON**	OFF	OFF	**ON**	**ON**	**ON**	**ON**

Table 6.3.: Table of the 6-stage used in Stereo ToF related with the 3-stage system. (IU: Illumination Unit, MOD SIG: Modulation signal connected to the illumination unit in state **ON** and CAPTURE: Camera capturing the scene when is in state **ON**)

show that the different configurations can obtain the desired measurement when using the 3-stage procedure.

These six stages allows us to capture the 6 measurements used in stereo ToF, without the necessity of the hardware integration and modification. These measurements are equivalent to the values that can be acquired using the original 3-stage system.

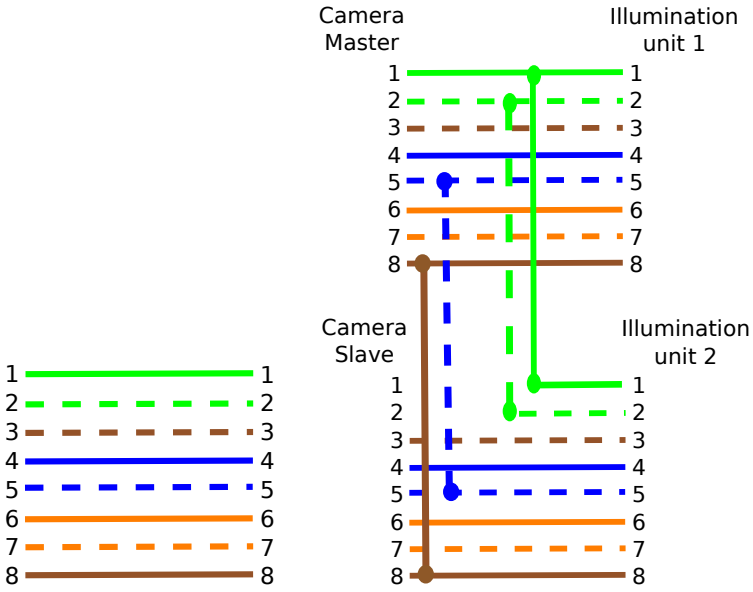
6.8.2. Manual switch - cables

Thanks to the fact that the PMD CamCube 2.0 is composed of one camera unit and two illumination units, it is possible to obtain the required six measurements to test and prove the stereo ToF theory using real ToF images. The camera unit has two output connectors in order to connect up to two illumination units, which each one has an input connector. The cable between the camera and the illumination unit has the next pin-out:

- **2 wires:** Modulation signal sent as a differential signal (LED+, LED−).
- **2 wires:** Power Supply (+12 V).
- **2 wires:** Ground reference.
- **2 wires:** It is not used until now (NC: No connection).

This cable has 4 wires to supply the illumination unit (2 wires with the supply voltage and 2 wires with the ground) and 2 wires for the modulation signal which is sent using a differential voltage. We built 4 simple cables, which connect point to point these 6 wires, in order to supply the infrared light source and send the modulation signal as in a normal ToF camera. This connection diagram is shown in Figure 6.15(a).

Notice that the stages 1-4 only requires a simple extension cable which connects a camera unit to the illumination units. For stages 1-4, we built 2 cables to connect the two illumination units.



(a) Cable 1 and 2 (b) Cable 3 and 4

Figure 6.15.: (a) Cable 1 and 2 are an extension of the original cable. The connection is one to one. (b) Cable 3 and 4 are an interconnection between the two cameras and shares only one modulation signal in the four illumination units. Pin 1 and 2 are the modulation signal which is shared to the camera slave (camera slave modulation signal is disconnected). Pin 5 and 8 are the ground which are shared between the cameras to have the same reference.

For stages 5 and 6, two cables were designed and built to connect the 4 illumination units of the two cameras at the same modulation frequency, and to provide the enough supply power to the each illumination unit. The connection diagram is shown in figure 6.15(b).

The next list shows the cables which were built for the experiments:

- **Cable 1 and 2:** Length of 50 cms and are used as an extension of the normal cable, which allows us to connect the infrared light source with the camera.

- **Cable 3 and 4:** Length of 60 cms and it takes the supply of each camera to each infrared light source. It gets the modulation signal from only one connector (one camera) and it distributes it to four illumination units.

These two cables ensure that all the illumination units use the same modulation signal and a similar offset distance (if the building is made taking care of the cable length). The introduced offset depth has to be estimated and then treated for the next data usage.

6.8.3. Automatic switch - electronic circuit

In the beginning, connection switches between the cameras and the illumination units were made manually (Section 6.8.2). This manual switching required a static scene for at least 1 minute. Therefore, an electronic circuit was designed in order to change the interconnections between the camera units and illumination units automatically. The electronic circuit was designed including the following parts:

- **Control unit:** The control unit is the responsible for the communication with the computer and the activation and state of the switches chip. The unit is composed of a microchip with a serial connection and digital outputs.
- **Switching unit:** It is the responsible of the electronic switch as the control unit wants. This unit is composed of 8 multiplexers or 8 switches Single Pole Double Throw (SPDT), two per illumination unit (LED$+$, LED$-$).
- **Supply and communication unit:** This unit is in charge to supply the multiplexers and the microchip. Also, it translates from USB to serial communication (USB to RS232-TTL).

For the control unit was selected a microchip ATMEGA8, due to its simplicity to program and internal clock (it is not necessary to connect an external clock, and make the circuit design simpler). Furthermore, the program speed is sufficient to make the communication to the computer and to manipulate the multiplexers. The switching unit is composed of 8 analogous multiplexers or switches, and it was implemented using 4 relays of Dual SPDT Switch RAL3W-K. These relays allows us to connect the two modulation signals to an illumination unit and then, we can select which modulation signal is used. The complete circuit uses the supply voltage provided by the USB connection of the computer (maximum of 500mA).

Control Unit

The control unit is composed of a microchip ATMEGA8 and was included a led array as a user output. Also, a connector for the USB-RS232 adapter was incorporated to make the connection to the computer and to obtain the power supply of $+5$ volts from the USB standard. Besides, a power led was added, which indicates when the circuit is supplied. One capacitor was included in parallel to the power supply, to filter any noise in the supply voltage. The circuit also includes a power led, which indicates when the circuit is connected to the supply voltage (USB). Furthermore, there are six leds used to show the stage of the system and code of errors.

The microchip was set up to work at a clock frequency of 8 Mhz (the maximum of internal clock) and serial connection of 38400 baud per second. Then, to connect to the control chip, it is necessary to set up the speed of the serial port to 38400 baud per second. The algorithm used in the control chip is as follows:

The control chip checks if some message is incoming and then, it is processed to determine the requested command. After the execution of the requested command, the chip sends an acknowledge to the computer in order to inform if the chip executed the command successfully (*i.e.*, the computer will know the state of the chip).

The commands of execution are detailed as follows:

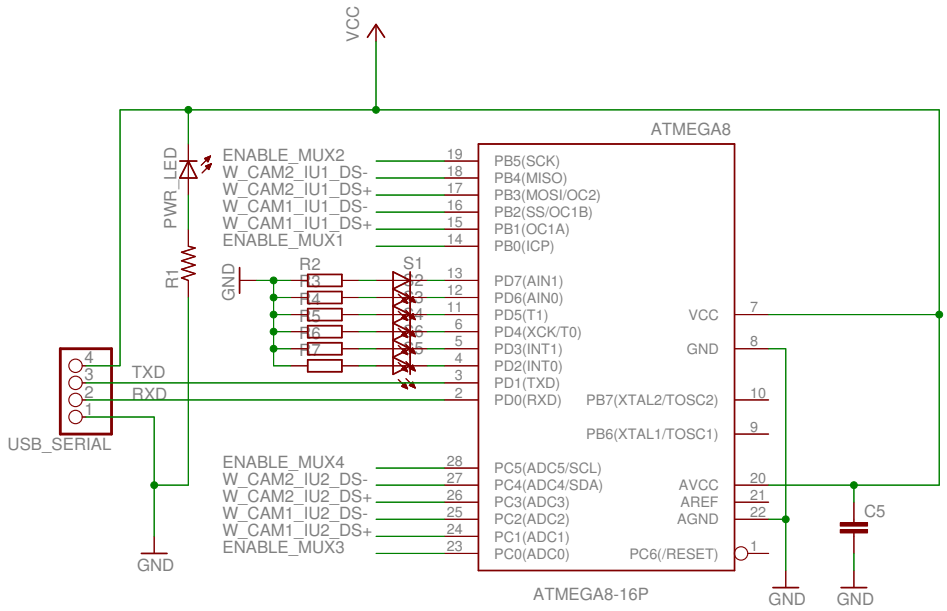


Figure 6.16.: Control circuit composed of a Microchip ATMEGA8, ADC connected to the multiplexers and serial connection to the USB-RS232-TTL connector.

- "SNCM$!": The circuit will be set the illumination units to the stage number N and camera M.
- "$H!": The circuit will be stopped.
- "$CALIB!": The circuit will connect each infrared source with its camera (Infrared source 1 and 2 to camera 1 and 2 respectively). This is set to aquire synchronized images from both ToF cameras, as the case of the stereo calibration of the system.
- "Tt$!": The space time between switching is set to t milliseconds, where t is an interger value.

The messages of acknowledge are explained in the following:

- "$EC!": Error of camera number, values allowed: 0 and 1 (in the case of two cameras).
- "$ES!": Error of stage number, values allowed: 0, 1 and 2 (three stages).
- "$EM!": Incorrect command. The command cannot be recognized by the control chip.
- "$STOP!": The circuit was stopped.
- "$CALIB!": The circuit was set-up in calibration mode.
- "$TIME!": The space time between switching of the ciruit was set-up.
- "OKSNM!": The circuit executed the instruction. Stage number N and camera M.

Algorithm 1 Control code: checking and understanding messages; and manipulating the multiplexers

```
1. Creation of codes to the multiplexers
2. Initialization of ADC ports (0x00) // Everything deactivated
while true do
  if No pending sending message and received message then
    3. Process the message
    if The message is correct then
      4. Activate the correspondent illumination unit
      5. Turn on the leds indicating the stage
      6. Add a pending acknowledge message (OK)
    else
      7. Deactivate all illumination unit
      8. Turn on leds indicating the error
      9. Add a pending error message (Error)
    end if
  end if
  if There is a pending sending message then
    10. Send the pending acknowledge/error message
  end if
end while
```

Switching Unit

This switching unit is composed of four relays RAL3W-K. Every RAL3W-K contains two multiplexers or switches, which are used for every LED+ and LED− of the camera. Each multiplexer is connected to the LED+/− of camera left and right, and the output to the illumination unit. The multiplexer can be managed to connect the output to camera left LED+/− or camera right LED+/−. Thus, each illumination unit can be connected to LED+/− of camera left or camera right independently, depending on the code given to the multiplexers.Due to the use of a relay for the switching, a half H-bridge is necessary for the relays control. In this case, we use a LD293D, which is composed by two half H-bridge, allowing us to control two relays. Then, two LD293D are necessary to control the 4 relays of the circuit. The interconnection can be see in Figure 6.17.

In Figure 6.17 are shown the groups to manipulate the illumination unit of the cameras. The first group manages the left illumination unit of both cameras, and it is used to select which modulation signal is sent to the illumination unit. Then, the first multiplexer connect LED+ and LED− from camera 1 and 2 as input and the left illumination unit of the camera left as output. The second multiplexer do the same, but in the illumination unit of camera right as output. The second group is analogous, but it manipulates the right illumination unit of both cameras. Every multiplexer has its power supply connected to the $+5$ volts from the USB. Also, every multiplexer has the enable pin connected to the control chip. The control chip can enable and disable each multiplexer.

(a)

(b)

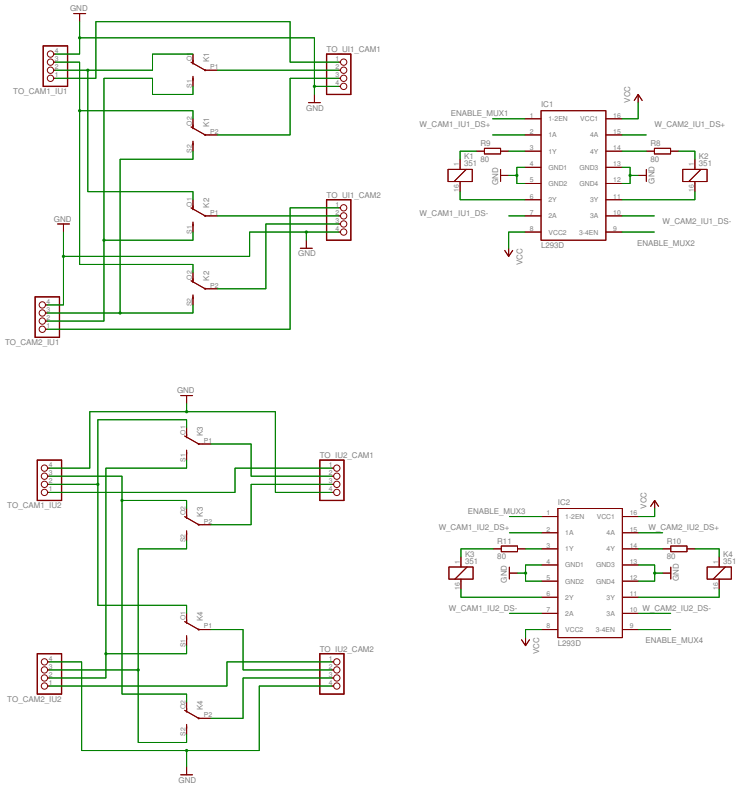


Figure 6.17.: Switching unit composed of four relays and two half H-bridge chip, divided in two groups. First group (a) manages the left illumination unit of both cameras. Second group (b) manipulates the right illumination unit of both cameras.

Supply and communication unit

This unit is in charge to supply the circuit and communicate to the computer. The USB port provides $+5$ volts and a maximum current of 500mA. The required energy can be provided by the USB port. For that, an adapter from RS232-TTL to USB is used to translate the serial information from USB to serial RS232. With that, the computer can connect through the USB port to the serial port RS-232-TTL of the ATMEGA8. Also, this adapter gives the electric energy to the circuit from the USB port of $+5$ volts. The multiplexer and control circuit are isolated from the illumination unit supply. The ground of circuit and the illumination unit are connected to make a general reference.

7. Stereo Time-of-Flight Simulator

THis Chapter introduces the implementation details of the Stereo ToF simulator, which was used to evaluate the Multi-view Time-of-Flight system (Chapter 6). The Stereo ToF simulator is based on the work of Keller *et al*. [53], which uses a Lambertian reflection model. The implemented simulator generates a 3D object, using OpenGL and from this 3D model the four ToF source images $C(\tau_i)$ where $i = 1, 2, 3, 4$ are calculated. The simulation includes attenuation of light in the air, the object reflectance (Lambertian), and noise in the sensor modeled as a zero-mean Gaussian noise. These source images are utilized to calculate the respective depth, amplitude and offset images, using the equations of demodulation, which were introduced in Chapter 3.

7.1. Introduction

The Time-of-Flight camera simulator attempt to emulate the real ToF camera behavior, even its noise and errors. There are principally two works where the researchers created a model for ToF cameras, specifically a model of Photonic Mixing Device (PMD) sensor [95, 53]. Differences between these works are mostly related to their performance and their inclusion of error and noise models. The first simulator was implemented in MATLAB, and thus, it is not suitable for real-time simulations. The measurement of the phase shift of the received signal is the basis of their simulated ToF sensor, which depends on the distance to the object. In this work, they simulate the response of every sensor pixel, as a single point response of the 3D scene. Also, they have the possibility of changing the position and orientation of the illumination unit and sensor separately, which allows simulating ToF camera with bistatic and multibistatic (multi-illumination unit systems) configurations. The case of the second simulator is different, it was implemented in OpenGL, which allows running in real-time. This simulator models the most of the errors and noise, which can be observed in a real ToF camera. It includes noise, flying pixels, motion artifacts and wiggling error. The source of wiggling error is produced due to two effects: the emitter of infrared light is an area and it is not the point source light; and the sensor pixel receives signal from an object area, *i.e.*many rays come back to the sensor pixel. To emulate this error, two accumulations of light are utilized: the integration of every ray emitted from the area of the illumination unit and hits each single object point; and the accumulation of every ray reflected from the object area and hits each pixel camera. To emulate the flying pixels, they generate a higher resolution depth image and then they take an average of a neighborhood of the pixel, in order to simulate the multi-path averaging. They also add noise to the source images to simulate the noise observed in the real sensor. Our simulator uses two single ToF camera simulators based on the second work, which are in a stereo set-up. In the simulation is included a delay between the cameras. Our simulation uses only a single ray simulation, a Lambertian reflectance model of the objects and a Gaussian zero-mean noise in the source images. Every noise image is calculated independently from each other ensuring different

noise pattern in every source image. The level of noise, reflectivity of the object and intensity of the emitted signal can be changed in the simulation program as parameters.

7.2. Simulation Theory

In this section an overview of the Stereo ToF theory is reviewed. Assuming that we have a 3D model of the object and two ToF cameras in stereo set-up with a known transformation between them.

Given the emitted signal $g(t)$ and the received signal $S(t)$:

$$\text{Emitted signal:} \quad g(t) = A \cdot \cos(\omega \cdot t) + B \tag{7.1}$$

$$\text{Received signal:} \quad S(t) = A' \cdot \cos(\omega \cdot t + \varphi) + B' \tag{7.2}$$

where φ is the phase-shift between the emitted and received signals, A is the amplitude and B is the offset or bias of the emitted signal; A' is the amplitude and B' is the offset or bias of the received signal; and finally ω is the angular frequency of the modulated sinusoidal wave. Usually, the frequency f is given in MHz which can be converted to angular frequency with the equation $\omega = 2\pi f$.

The emitted signal is assumed as point source light and it is located at the same position and orientation of the sensor. First, the infrared light is emitted from the illumination unit into the air to the object. Then, the object reflects part of the incident light and finally the reflected light comes back to the sensor through the air. When the light is traveling through the air, the light signal is attenuated. Therefore, the emitted signal is attenuated from the illumination unit until the object. Later, a quantity of incident light is reflected on the object surface. This quantity of reflected light depends on reflectivity of the object surface. After, the reflected light flies back to the sensor and it is also attenuated. The attenuation depends not only on the distance between the illumination unit and object, but also on the distance between the object and sensor. For the case of a single ToF camera the distance between the illumination unit-object and object-sensor is the same distance d. Therefore, if we assume an attenuation as $\frac{1}{d^2}$ and a Lambertian reflection model, the reflected light I_R can be calculated as:

$$I_R = \frac{C_R(\hat{w}_i \cdot \hat{N})I_e}{d^2} \tag{7.3}$$

where $\hat{w}_i$ is the incident vector, $\hat{N}$ is the surface normal, C_R is the reflectivity of object material, $I_e = A \cdot \cos(\omega \cdot t + \varphi) + B$ is the emitted signal and $\varphi = \frac{4\pi f d}{c_{light}}$ is the phase-shift of the received signal.

Then, the light received by the sensor I_R can be estimated as the attenuation of the reflected light:

$$I_R = \frac{C_R(\hat{w}_i \cdot \hat{N})I_e}{d^4} \tag{7.4}$$

The light acquired by the sensor I_R is a simulated value of the captured light by a real sensor. The acquired light has to be calculated in every pixel of the sensor. With the value of I_R, the source images $C(\tau_i)$ with $i = 0, 1, 2, 3$ are computed. For that, it is necessary to calculate I_e in the different sampling points $\omega \cdot t = 0, \frac{\pi}{2}, \frac{3\pi}{2}, 2\pi$. Also, the value of I_e incorporates the phase-shift φ of the object obtained from the distance d between the sensor and object.

The values of $C(\tau_i)$ are:

$$C(\tau_i) = \frac{C_R(\hat{w}_i \cdot \hat{N})(A \cdot \cos(\frac{(i+1)\pi}{2} + \frac{4\pi f d}{c_{light}}) + B)}{d^4} + \nu \tag{7.5}$$

where $i = 0, 1, 2, 3$ and ν is a noise introduced to the measurement.

Finally, having the values of $C(\tau_i)$ where $i = 0, 1, 2, 3$, it is possible to compute the offset, amplitude and phase shift of the received signal as follows:

$$\text{Offset:} \qquad B' = \frac{A_0 + A_1 + A_2 + A_3}{4} \tag{7.6}$$

$$\text{Amplitude:} \qquad A' = \frac{\sqrt{(A_3 - A_1)^2 + (A_0 - A_2)^2}}{2} \tag{7.7}$$

$$\text{Phase Shift:} \qquad \varphi = \arctan\left(\frac{A_3 - A_1}{A_0 - A_2}\right) \tag{7.8}$$

To create the simulated images is necessary to compute the offset, amplitude and phase shift for every pixel of the observer, using the 3D model of the object in the software.

The result of this simulation is a depth image, an amplitude image and an offset image of a ToF camera. To simulate the stereo ToF, it is necessary to make some modification of the models. First of all, a delay between the cameras and two camera positions, have to be introduced. The stereo ToF system needs two ToF camera simulators, and they are located in two different positions. Besides, the calculation of received signal in every sensor has to change properly depending on the simulated stage. The simulated stages are as follows:

	left IR light	right IR light
Stage 1	ON	OFF
Stage 2	OFF	ON
Stage 3	ON	ON

Table 7.1.: Summary of illumination unit manipulation for stereo ToF. All cameras are always capturing.

Given an emitted signal in every camera as follows:

$$\text{Camera left: } g_l(t) = A_l \cdot \cos(\omega \cdot t) + B_l \tag{7.9}$$

$$\text{Camera right: } g_r(t) = A_r \cdot \cos(\omega \cdot t + \varphi) + B_r \tag{7.10}$$

where both cameras are working at the same modulation frequency ω. We assume a distance d_l between an object point and the camera left. In the same way, we assume a distance d_r between an object point and the camera right. The states captured light in every stage per camera is shown in the following items.

- **Stage 1:** The illumination unit of camera left is turned on and both cameras capture. The emitted signal is g_l, then $I_e = A_l \cdot \cos(\omega \cdot t + \varphi) + B_l$, where the value of φ depends on the camera where is computed.
  - **Camera left:** The received signal is as a normal ToF: $I_R = \frac{C_R(\hat{w}_i \cdot \hat{N}) I_e}{d_l^4}$. Notice that the attenuation is influenced only by d_l. Here I_e is computed using $\varphi = \frac{4\pi f d_l}{c_{light}}$.

  - **Camera right:** The received signal is the reflection of the light from the object point to the right camera. Then, the received signal is: $I_R = \frac{C_R(\hat{w}_i \cdot \hat{N})I_e}{d_l^2 d_r^2}$. Notice now that the attenuation is dependent on both distances. This dependency is produced because the first attenuation is made from camera left to the object, then reflected and finally the reflected signal is again attenuated from the object to the camera right. Here I_e is computed using $\varphi = \frac{2\pi f(d_l+d_r)}{c_{light}}$.

- **Stage 2:** The illumination unit of camera right is turned on and both cameras capture. The emitted signal is g_r.
  - **Camera left:** The received signal is the reflection of the light from the object point to the left camera. Then, the received signal is: $I_R = \frac{C_R(\hat{w}_i \cdot \hat{N})I_e}{d_l^2 d_r^2}$. Notice now that the attenuation is dependent on both distances as in stage 1 in camera left. Here I_e is computed using $\varphi = \frac{2\pi f(d_l+d_r)}{c_{light}}$.
  - **Camera right:** Similar to stage 1, the received signal is as a normal ToF: $I_R = \frac{C_R(\hat{w}_i \cdot \hat{N})I_e}{d_r^4}$. Notice that the attenuation is influenced only by d_r (now the camera right is turned on). Here I_e is computed using $\varphi = \frac{4\pi f d_r}{c_{light}}$.

- Stage 3: The illumination unit of both cameras are turned on and both cameras capture. The emitted signals are g_l and g_r.
  - **Camera left:** The received signal is the reflection of the light from the object point to the left camera. Then, the received signal is: $I_R = \frac{C_R(\hat{w}_i \cdot \hat{N})I_{el}}{d_l^4} + \frac{C_R(\hat{w}_i \cdot \hat{N})I_{er}}{d_l^2 d_r^2}$. Note that the received signals are the addition of the stage 1 and 2. Here I_el is computed using $\varphi = \frac{4\pi f d_l}{c_{light}}$ and I_er is computed using $\varphi = \frac{2\pi f(d_l+d_r)}{c_{light}}$.
  - **Camera right:** The received signal is the reflection of the light from the object point to the right camera. Then, the received signal is: $I_R = \frac{C_R(\hat{w}_i \cdot \hat{N})I_{el}}{d_l^2 d_r^2} + \frac{C_R(\hat{w}_i \cdot \hat{N})I_{er}}{d_r^4}$. Notice that the received signals are the addition of the stage 1 and 2. Here I_el is computed using $\varphi = \frac{2\pi f(d_l+d_r)}{c_{light}}$ and I_er is computed using $\varphi = \frac{4\pi f d_r}{c_{light}}$.

Having the considerations mentioned before, the stereo ToF camera can be simulated. The computation of the light captured by the sensor is dependent on the distance, between the object - camera left and object - camera right. The simulation has to calculate this distance and then it can emulate the light received by the sensor.

7.3. Software implementation

The Simulator was implemented in OpenGL through OpenGL Shading Language (GLSL) which allows a rendering of the source images at a fast frame-rate. From this source images is possible to calculate the depth, amplitude and offset images using Equations 7.6-7.8. The implemented GLSL program computes the source images $C(\tau_i)$ using the distance from the cameras to the object point and the normals of the surface. The distances are used to computed the attenuation and phase shift. The surface normals are used to compute the quantity of reflected light through the Lambertian model. We utilized two Frame Buffer Object (FBO), one to render the source images and the other to save the ground truth. For each camera, it is assumed that the sensor is located in the same

position of its light source. In the case of stereo ToF, we have two cameras, and therefore, we have two illumination units and two view points (cameras). The location of the cameras are given as parameters, *i.e.*baseline and vergence. Every source image $C(\tau_i)$ for every camera is stored in an image class to later compute the offset, amplitude and phase-shift. Besides, a Gaussian zero-mean noise is added to every source image, it is possible to simulate the noise in the ToF cameras. The source images are computed sequentially, *i.e.*, one camera source image is computed one after the other. For that, the program compute every source image by changing the eye view position and storing every source image in distinguishable image objects. A shading program was implemented to compute the source images, depending on the eye position, illumination unit activated and other parameters which can be transfered from the main code. The simulation of the stages of the stereo ToF is achieved changing the light condition (as in table 7.1) and the view point per acquisition in the main code. Two OpenGL source light are created representing the illumination unit of every ToF camera and they can be controlled from the main code. The illumination units can be activated or deactivated separately. The activation and deactivation of the light source is made from the main program and this command is communicated to the GLSL program. The activation of the light sources allows the simulation of the three stages including the third where the interference between the cameras occurs.

The simulation software is composed of two main codes, the first one controls the simulation and the display, written in C++; and the second is the GLSL code which implements the simulation of the source images and rendering of the 3D object in the OpenGL environment. The value of the modulation frequency, material reflectivity, emitted amplitude and emitted offset are transfered from the C++ code to the GLSL program through uniform variables.

The vertex shading program calculates the incident light vector, using the light source position and object vertex, surface normal and observer vector with respect to the object vertex. Three vectors are computed in the vertex program to represent the variables of the stereo-ToF simulator and then they are communicated from the vertex program to the fragment software. The vertex and fragment code check in every computation if the source light are turned on/off of each illumination unit. This checking helps to manipulate the illumination units from the C++ code in order to simulate the stages utilized for the Multi-view ToF system.

The main code is implemented in C++ and it is responsible of the rendering of the 3D object, the manipulation of the light sources, changing the view points, addition of the noise in the source images, calculation of the amplitude, offset and depth images from the simulated source images and the optimization of the depth images. This main program allocate memory not only in the Shading (FBO) for the GLSL programs, but also in the C++ memory for the storage of the images. Also, the main code checks the states of the program, *i.e.*, if the user wants to exit, start or stop the simulation. The stages are represented by the variable i and each camera Cam is processed separately. Furthermore, this main program run the GLSL programs, vertex and fragment programs. The main code is summarized in the algorithm 2.

The vertex program calculates the vectors involved in the calculation the source images and it automatically computes the interpolation necessary to transfer the information to the fragment code. The vertex code is shown in the algorithm 3. This code shows that the state of turned on/off of a total number of illumination units NUM-LIGHTS (in the stereo ToF are two) are checked. In the code, w_i represents the vector between the light source and the object point, $\hat{N}$ is the normalized

Algorithm 2 Main code: rendering of object, manipulation of light sources, addition of noise and optimization of stereo ToF

1. Creation of Buffers (FBO)
2. Creation of image classes per camera (4 source images, amplitude, offset and depth images)
3. Creation of the two OpenGL source light
4. $i \leftarrow 0$

while true **do**
  5. Rendering of 3D object
  for Every camera Cam **do**
    if i is 0 **then**
      Stage 1
      6. Activate illumination unit 1
      7. Deactivate illumination unit 2
    end if
    if i is 1 **then**
      Stage 2
      8. Deactivate illumination unit 1
      9. Activate illumination unit 2
    end if
    if i is 2 **then**
      Stage 3
      10. Activate illumination unit 1
      11. Activate illumination unit 2
    end if
    12. Locate eye view point into camera Cam position
    13. Source images $\leftarrow$ Run GLSL program (Vertex and Fragment code)
    14. Source images $\leftarrow$ Source images $+$ Noise
    15. Amplitude image $\leftarrow$ Computation of amplitude image from source images
    16. Offset image $\leftarrow$ Computation of offset image from source images
    17. Depth image $\leftarrow$ Computation of depth image from source images
    18. Display source images
    19. Display amplitude, offset and depth images
    20. $i \leftarrow i + 1$
    if i is 3 **then**
      21. $i \leftarrow 0$
    end if
  end for
  22. Optimize depth image of camera 1
  23. Optimize depth image of camera 2
  24. Display optimized images
  25. Check pressed button (exit, stop, start)
end while
26. Save output and print messages

normal vector of the surface and w_r is the object vertex position w.r.t.the eye position.

Algorithm 3 Vertex code: computation of surface normal, object-illumination unit and object-eye vectors

for $i = 1 \rightarrow$ Number of illumination units NUM-LIGHTS **do**
  if Illumination unit i is enabled **then**
    1. Compute vertex-eye vector w_r
    2. Compute vertex-illumination unit i vector w_i
    3. Compute vertex normal $\hat{N}$
  end if
end for
ftransform(); // Interpolate every vertex and transfer to fragment code (every pixel position)

The fragment code computes the received intensity in the image plane of the camera, and for that, we use the vector calculated in the vertex code, which is also interpolated to every pixel position by OpenGL. Here the final intensity value of the source image is calculated using the modulation frequency, material reflectivity; and the dot product between the normal and incident light vector. The fragment code is summarized in algorithm 4. In the algorithm, the `for` of MAX-LIGHTS is checking, if the source light is activated or not and it is made in each source image computation. d_1 and d_2 are calculated to simulate the distances from the source light to the object and from the object to the point of view (camera). Also, the delay ϕ_{lr} between the cameras is added in the simulation.

Algorithm 4 Fragment code: computation of surface normal, object-illumination unit and object-eye vectors

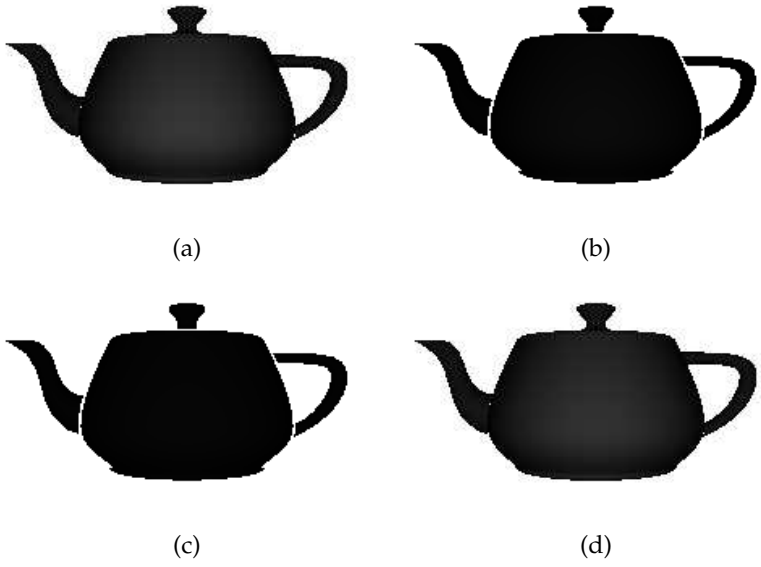
$C_R \leftarrow$ reflectivity value from C++ program
$\phi_{lr} \leftarrow$ delay value from C++ program
$sample \leftarrow \omega t = i\frac{\pi}{2}$ with $i = 1, 2, 3, 4$ from C++ program
$I \leftarrow$ intensity value from C++ (factor to reduce or raise the emitted signal)
for $i = 1 \rightarrow$ Number of illumination units MAX-LIGHTS **do**
  if Illumination unit i is enabled **then**
    1. Compute Lambertian term (Dot product between vertex-illumination unit i vector and normal vector) $LambTerm \leftarrow C_R(\hat{w}_i \cdot \hat{N})I$
    2. Compute distance vertex-illumination unit i $d_1 \leftarrow |w_i|$
    3. Compute distance vertex-eye $\rightarrow d_2 \leftarrow |w_r|$
    4. Compute the emitted signal $I_e \leftarrow 0.5(cos(sample + 2\pi f \frac{(d_1+d_2)}{c_{light}} + \phi_{lr}) + 1.0)$
    5. Compute received signal in the pixel (including attenuation) $I_R \leftarrow \frac{LambTerm \cdot I_e}{d_1^2 d_2^2}$
    6. Store pixel value `gl_FragData[0]` $\leftarrow I_R$ (in FBO 0)
    7. Store distance d_2 as ground truth `gl_FragData[1]` $\leftarrow d_2$ (in FBO 1). Normalized between 0 and 1 using range 0-MaxDistance meters ($20MHz \rightarrow 7.5$m).
  end if
end for

The noise of the source images is introduced in the C++ code after the calculation of the simulated source images by GLSL program. Every source image is stored in a buffer of 16-bits, simulating the

same data storage of the typical ToF camera. Therefore, the source images are stored using values between 0 and 2^{16}. The noise is simulated as a percentage of this 2^{16} possible values, *i.e.*, a noise level of $p\%$ is $\frac{p*2^{16}}{100}$. Finally, the range of noise is $\pm\frac{p*2^{16}}{200}$ from the ground truth.

7.4. Results

The results of ToF camera simulator are shown in Figure 7.1. These source images are simulated without noise. Furthermore, depending on the sample, the images are with different illumination levels. As it is expected, the first source image is more illuminated than the others (different position in the sinusoidal signal). Using these four sources images, the phase-shift, the amplitude and the offset can be computed, obtaining the images shown in Figure 7.2.



(a) (b)

(c) (d)

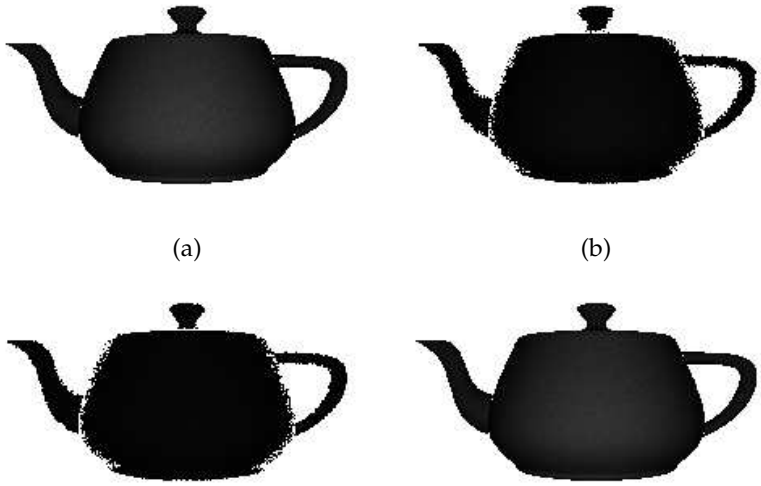
Figure 7.1.: Simulated source images in absence of noise. Object 1 meter of distance. (a) First sample ($wt = \frac{\pi}{2}$), (b) Second sample ($wt = \pi$), (c) Third sample ($wt = \frac{3\pi}{2}$), (d) Fourth sample ($wt = 2\pi$).

The source images with a noise are shown in Figure 7.3. The noise in the source images can be observed as small changes of grayscale levels. The source images, including the noise, are used to calculate the phase-shift, the amplitude and the offset. These images also exhibits noise transfered from the source images. The phase-shift, the amplitude and offset images can be seen in Figure 7.4.

The stereo ToF images are shown in Figure 7.5, where can be observed, not only the different position and orientation of each ToF camera, but also the distinct noise introduced in the source images.

(a) (b) (c)

Figure 7.2.: Amplitude (a), offset (b) and depth (c) images computed from the simulated source images without noise.



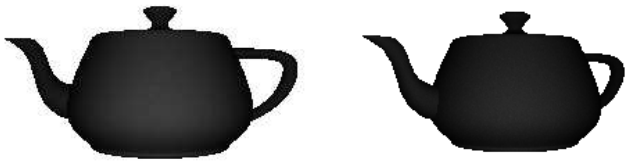
(a) (b)

(c) (d)

Figure 7.3.: Simulated source images with a level of noise of 5%. Object is located 1 meter from the camera. (a) First sample ($wt = \frac{\pi}{2}$), (b) Second sample ($wt = \pi$), (c) Third sample ($wt = \frac{3\pi}{2}$), (d) Fourth sample ($wt = 2\pi$).

(a) (b) (c)

Figure 7.4.: Amplitude (a), offset (b) and depth (c) images computed from the simulated source images with a level of noise of 5%.



(a) (b)

Figure 7.5.: The offset images from left camera (a) and right camera (b) computed from the simulated source images with a level of noise of 5% and baseline of 40 cms. Every camera has its own pattern of noise.

Part V.

Conclusion and Discussion

8. Conclusion and Discussions

First of all, we presented a detailed state of the art of Time-of-Flight camera (Chapter 3), which includes its working principles and applications in computer graphics. The working principles were presented with a new formulation of the basic equations, including an explanation of its electronic implementation. Furthermore, the errors and limitations were described, showing the current algorithm to improve and use them in real applications. Also, the applications of the ToF cameras in computer graphics, developed until now, were enlisted.

In Chapter 4, we have investigated the feasibility of MonoSLAM for 3D surface reconstruction from monocular NBI sequences. The results demonstrate that MonoSLAM is a promising tool for surface reconstruction in NBI sequences. However, this study shows that there are some challenges for further improvement, which need to be addressed before MonoSLAM can be applied on endoscopic images in-vivo and in-situ. The tissue deformation which is present in the esophagus, can greatly modify the appearance of the tissue, causing a failure in the feature tracking. Moreover, due to the specular reflections and blurring caused by the esophageal surface, or as a result of the fast camera motion, the image quality can be very poor. We have proposed simple methods to address some of these challenges within the MonoSLAM framework. More robust approaches for detecting and tracking of the landmark features could also be considered. For example geometric verifications such those proposed in [6] could be added. Similarly, more sophisticated methods could be applied for the removal of specular lights [97] and blurred images or regions [73]. Finally, a-priori knowledge may also be used to enhance the accuracy of the reconstruction. For instance, in [86] organ models are learned in order to constraint the depth estimates from stereo.

Additionally, we have presented a new online SLAM approach which uses a combined sensor gathering images from a high-resolution camera registered to a ToF device (Chapter 5). The combined HR-ToF sensor includes valuable depth information while allowing for high-precision tracking. We address the SLAM problem by extending the measurement model and the innovation formulas of the MonoSLAM algorithm. The results show that these extensions improve the map uncertainty and the localization error of the camera. Future work directions include detecting 3D features and using 3D tracking (scene-flow) methods to take full advantage of the depth image. Furthermore, can be added the registration using a combined stereo and depth calibration [39] method that models the depth error with B-spline or polynomial functions [68]. This is expected to reduce the noise of the depth measurements. Finally, it would be interesting to use the proposed model with an Unscented Kalman filter (UKF) [113] instead of Extended Kalman filter, since the UKF converges better when the measurements have a non-Gaussian distributed noise as is the case of the ToF depth measurements.

Finally, we have proposed a novel multi-ToF depth acquisition method that exploits the physical properties of ToF devices and integrates measurements from the N cameras at a low-level (Chapter 6). The $(N+1)$-stage acquisition method permits obtaining redundant measurements that are

used together with the geometry of the multi-view setup to optimize the depth values per pixel. The optimization considers $N(N+1)$ measurements acquired under $N+1$ different infrared lighting conditions from N points of view. We have used the example of two cameras to explain and demonstrate the feasibility of the proposed method. All the calculated formulas are based on the case of stereo-ToF. Results on simulated and real data show that the proposed method produces more accurate depth images for reasonable stereo configurations. In addition, a theoretical proof and the limitations of the system was presented, showing that the multi-view system, not only reduces the noise of the measurement, but also detects the outliers measurements and occluded points. In addition, the stereo-ToF simulator was explained in detail, including the program algorithm and implementation instructions (Chapter 7). We focused on keeping high acquisition rates, and thus proposed an optimization method that works pixel-wise, which enables real-time implementations. Nevertheless, regularization terms could be incorporated to enforce surface smoothness and photometric models could be considered to relate the normals and reflection properties of the surface with the measured values by the ToF camera [15]. Since the result provides N optimized depth images, the proposed methodology can be combined with complementary methods for depth calibration [72] and/or as an improved input to 3D reconstruction algorithms that combine several ToF images [56, 77]. Finally, the approach was extended from two to N cameras (or lighting units) by increasing the number of stages to $N+1$ and not all the combination of stages are always required.

Appendix

A. Phaseshift calculation details

Being the original and received signal:

$$\text{Original signal:} \quad S(t) = A \cdot \cos(\omega \cdot t) + B \tag{A.1}$$

$$\text{Received signal:} \quad R(t) = A' \cdot \cos(\omega \cdot t + \varphi) + B' \tag{A.2}$$

The ToF camera measures at $\varphi_{org} = \omega \cdot t = i\frac{\pi}{2}$.

$$\text{Time } i = 0 \wedge \varphi_{org} = 0 \quad C(\tau_0) = A' \cdot \cos(0 + \varphi) + B' = A' \cdot \cos(\varphi) + B' \tag{A.3}$$

$$\text{Time } i = 1 \wedge \varphi_{org} = \tfrac{\pi}{2} \quad C(\tau_1) = A' \cdot \cos(\frac{\pi}{2} + \varphi) + B' = A' \cdot cos(\frac{\pi}{2}) \tag{A.4}$$

$$\text{Time } i = 2 \wedge \varphi_{org} = \pi \quad C(\tau_2) = A' \cdot \cos(\pi + \varphi) + B' \tag{A.5}$$

$$\text{Time } i = 3 \wedge \varphi_{org} = \tfrac{3\pi}{2} \quad C(\tau_3) = A' \cdot \cos\left(\frac{3\pi}{2} + \varphi\right) + B' \tag{A.6}$$

Simplifying and replacing the cosine rule $\cos(\alpha + \beta) = \cos(\alpha) \cdot \cos(\beta) - \sin(\alpha) \cdot \sin(\beta)$, we obtain:

$$\begin{aligned} C(\tau_0) &= A' \cdot \cos(0 + \varphi) + B' \\ &= A' \cdot \cos(\varphi) + B' \end{aligned}$$

$$\begin{aligned} C(\tau_1) &= A' \cdot \cos(\frac{\pi}{2} + \varphi) + B' \\ &= A' \cdot \cos\left(\frac{\pi}{2}\right)\cos(\varphi) - \sin\left(\frac{\pi}{2}\right)\sin(\varphi) + B' \\ &= A' \cdot \cancelto{0}{\cos\left(\frac{\pi}{2}\right)}\cos(\varphi) - \cancelto{1}{\sin\left(\frac{\pi}{2}\right)}\sin(\varphi) + B' \\ &= -A' \cdot \sin(\varphi) + B' \end{aligned}$$

$$\begin{aligned} C(\tau_2) &= A' \cdot \cos(\pi + \varphi) + B' \\ &= A' \cdot \cos(\pi)\cos(\varphi) - \sin(\pi)\sin(\varphi) + B' \\ &= A' \cdot \cancelto{-1}{\cos(\pi)}\cos(\varphi) - \cancelto{0}{\sin(\pi)}\sin(\varphi) + B' \\ &= -A' \cdot \cos(\varphi) + B' \end{aligned}$$

$$\begin{aligned} C(\tau_3) &= A' \cdot \cos(\frac{3\pi}{2} + \varphi) + B' \\ &= A' \cdot \cos\left(\frac{3\pi}{2}\right)\cos(\varphi) - \sin\left(\frac{3\pi}{2}\right)\sin(\varphi) + B' \\ &= A' \cdot \cancel{\cos\left(\frac{3\pi}{2}\right)}^{0}\cos(\varphi) - \cancel{\sin\left(\frac{3\pi}{2}\right)}^{-1}\sin(\varphi) + B' \\ &= A' \cdot \sin(\varphi) + B' \end{aligned}$$

Then, we get four formulas for the values which are measured with the ToF camera. The phase shift (φ), amplitude (A') and offset (B') of the received signal can be calculated as follows:

- Offset B' (Gray-scale):

$$\begin{aligned} C(\tau_0) + C(\tau_1) + C(\tau_2) + C(\tau_3) =& A' \cdot \cos(\varphi) + B' - A' \cdot \sin(\varphi) + B' - A' \cdot \cos(\varphi) + B' + A' \cdot \sin(\varphi) + B' \\ \Rightarrow C(\tau_0) + C(\tau_1) + C(\tau_2) + C(\tau_3) =& \cancel{A' \cdot \cos(\varphi) - A' \cdot \cos(\varphi)}^{0} + \cancel{A' \cdot \sin(\varphi) - A' \cdot \sin(\varphi)}^{0} + 4B' \\ \Rightarrow C(\tau_0) + C(\tau_1) + C(\tau_2) + C(\tau_3) =& 4B' \\ \Rightarrow B' =& \frac{C(\tau_0) + C(\tau_1) + C(\tau_2) + C(\tau_3)}{4} \end{aligned}$$

- Phase Shift φ:

$$\begin{aligned} C(\tau_3) - C(\tau_1) &= A' \cdot \sin(\varphi) + B' - (-A' \cdot \sin(\varphi) + B') \\ &= A' \cdot \sin(\varphi) + A' \cdot \sin(\varphi) + \cancel{B' - B'}^{0} \\ &= 2A' \cdot \sin(\varphi) \end{aligned}$$

$$\begin{aligned} C(\tau_0) - C(\tau_2) &= A' \cdot \cos(\varphi) + B' - (-A' \cdot \cos(\varphi) + B') \\ &= A' \cdot \cos(\varphi) + A' \cdot \cos(\varphi) + \cancel{B' - B'}^{0} \\ &= 2A' \cdot \cos(\varphi) \end{aligned}$$

$$\begin{aligned}
\frac{C(\tau_3) - C(\tau_1)}{C(\tau_0) - C(\tau_2)} &= \frac{2A' \cdot \sin(\varphi)}{2A' \cdot \cos(\varphi)} \\
&= \frac{\cancel{2A'}^{1}}{\cancel{2A'}} \cdot \frac{\sin(\varphi)}{\cos(\varphi)} \\
&= \tan(\varphi) \\
&\Rightarrow \varphi = \arctan\left(\frac{C(\tau_3) - C(\tau_1)}{C(\tau_0) - C(\tau_2)}\right)
\end{aligned}$$

- Amplitude A':

$$\begin{aligned}
\sqrt{(C(\tau_3) - C(\tau_1))^2 + (C(\tau_0) - C(\tau_2))^2} &= \sqrt{(2A' \cdot \sin(\varphi))^2 + (2A' \cdot \cos(\varphi))^2} \\
&= \sqrt{4A'^2 \cdot \sin(\varphi)^2 + 4A'^2 \cdot \cos(\varphi)^2} \\
&= \sqrt{4A'^2 \cdot (\cancel{\sin(\varphi)^2 + \cos(\varphi)^2})^{1}} \\
&= \sqrt{4A'^2} = 2A' \\
\Rightarrow A' &= \frac{\sqrt{(C(\tau_3) - C(\tau_1))^2 + (C(\tau_0) - C(\tau_2))^2}}{2}
\end{aligned}$$

Finally, the received can be reconstructed with the following three equations using only four measurements of the received signal:

$$\text{Offset:} \qquad B' = \frac{C(\tau_0) + C(\tau_1) + C(\tau_2) + C(\tau_3)}{4} \tag{A.7}$$

$$\text{Amplitude:} \qquad A' = \frac{\sqrt{(C(\tau_3) - C(\tau_1))^2 + (C(\tau_0) - C(\tau_2))^2}}{2} \tag{A.8}$$

$$\text{Phase Shift:} \qquad \varphi = \arctan\left(\frac{C(\tau_3) - C(\tau_1)}{C(\tau_0) - C(\tau_2)}\right) \tag{A.9}$$

Another and more general point of view is to demodulate the phase between the emitted and received signal using the *cross-correlation*. The cross-correlation technique gives a comparison between two signal and we can recover the relation of amplitude and phase between the emitted and received signal. The *cross-correlation* function is as follows:

$$C(\tau) = R(t) * S(t) = \lim_{T \to \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} R(t) \cdot S(t + \tau) \cdot dt \tag{A.10}$$

where $*$ is the convolution in the time between two waves, $s(t)$ received signal and $g(t)$ reference signal.

Resolving this convolution for our signal $S(t)$ and $R(t)$, we get the next function:

$$C(\tau) = \frac{A' \cdot A}{2} \cos(\omega\tau + \varphi) + B' \cdot B \tag{A.11}$$

When we replace in this equation different values of $\omega\tau_i = i\frac{\pi}{2}$ with $i = \{0, 1, 2, 3\}$, we obtain the next equations:

$$
\begin{aligned}
C\left(\tau_0 = 0\right) &= \frac{A' \cdot A}{2}\cos(\varphi) + B' \cdot B \\
C\left(\tau_1 = \frac{\pi}{2}\right) &= -\frac{A' \cdot A}{2}\sin(\varphi) + B' \cdot B \\
C\left(\tau_2 = \pi\right) &= -\frac{A' \cdot A}{2}\cos(\varphi) + B' \cdot B \\
C\left(\tau_0 = \frac{3\pi}{2}\right) &= \frac{A' \cdot A}{2}\sin(\varphi) + B' \cdot B
\end{aligned}
$$

In the same way of our last calculations, we can deduce the values of the phase φ and amplitude A'.

Amplitude:
$$A' = \frac{\sqrt{\left(C(\tau_3) - C(\tau_1)\right)^2 + \left(C(\tau_0) - C(\tau_2)\right)^2}}{2A} \tag{A.12}$$

Offset:
$$B' = \frac{C(\tau_0) + C(\tau_1) + C(\tau_2) + C(\tau_3)}{4B} \tag{A.13}$$

Phase Shift:
$$\varphi = \arctan\left(\frac{C(\tau_3) - C(\tau_1)}{C(\tau_0) - C(\tau_2)}\right) \tag{A.14}$$

B. Details of calculation of the limitations

In the limitations part (Section 6.5), the maximum phase-shift was determined as:

$$\phi_{max} = \pi - \arccos\left(\frac{A_2}{2A_1}\right) \text{ if } A_1 > A_2 \tag{B.1}$$

$$\phi_{max} = \pi - \arccos\left(\frac{A_1}{2A_2}\right) \text{ if } A_2 > A_1 \tag{B.2}$$

To plot this ϕ_{max} (Figure 6.10), some assumptions were made:

- The surface has Lambertian reflection
- The infrared light has an inverse of quadratic attenuation

Then, the fraction $\frac{A_2}{2A_1}$ will be (using equations of Section 6.6):

$$\frac{A_2}{2A_1} = \frac{\frac{C_R(\hat{w}_i \cdot \hat{N})I_e}{\lambda_1^2 \lambda_2^2}}{2\frac{C_R(\hat{w}_i \cdot \hat{N})I_e}{\lambda_1^4}} \tag{B.3}$$

$$\frac{A_2}{2A_1} = \frac{\frac{1}{\lambda_1^2 \lambda_2^2}}{2\frac{1}{\lambda_1^4}} \tag{B.4}$$

$$\frac{A_2}{2A_1} = \frac{\frac{1}{\lambda_2^2}}{2\frac{1}{\lambda_1^2}} \tag{B.5}$$

$$\frac{A_2}{2A_1} = \frac{\lambda_1^2}{2\lambda_2^2} \tag{B.6}$$

C. Measurement of the ToF camera attenuation

To measure the attenuation of the Time-of-Flight camera in the air, we acquire ToF images at different distances from a white wall. The measurements were performed statically in each distance position and where it was captured 100 images. We show in Figure C.1 the graph of the distance versus the amplitude for an average value in a matrix in the center of the depth image of 10×10. This matrix is selected to reduce the influence of the noise. The original attenuation values were fitted to an inverse of a polynomial function, obtaining $\frac{1}{2.41+6.76\cdot\lambda^2+0.70\cdot\lambda^4}$.

We can conclude that the attenuation of the ToF signal is dependent on $\frac{1}{\lambda^4}$, $\frac{1}{\lambda^2}$ and a constant; and compared to the normal model of attenuation $\frac{1}{\lambda^4}$, this new model includes the variables of $\frac{1}{\lambda^2}$ and a constant with some coefficients. This attenuation model was used in the ToF simulator, having a simulation more realistic to the real ToF camera images, instead using the typical attenuation model.

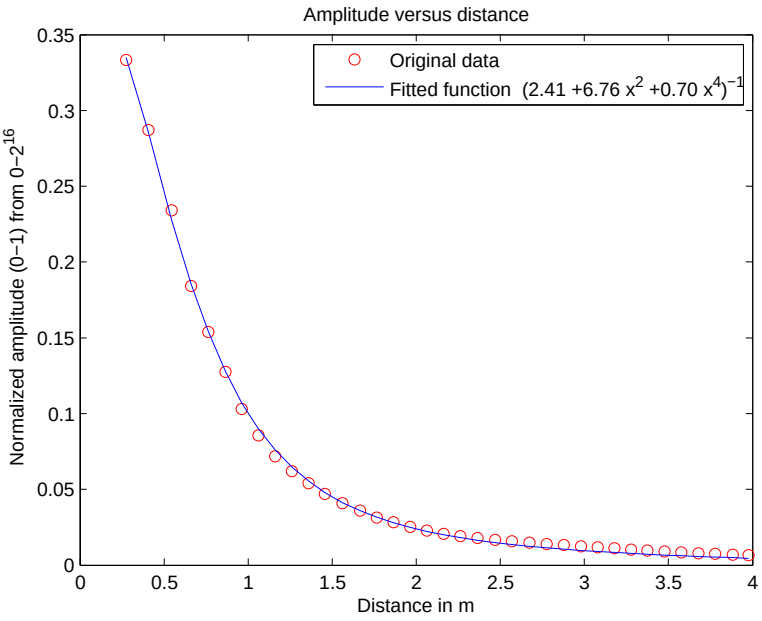


Figure C.1.: Real attenuation (red circles) and its fitted function (blue line).

D. ToF commander PCB print

The designed PCB is a 2-layer PCB with superficial components: resistance, led, multiplexer; and non-superficial components: microcontroller Atmega 8 and pin head connector (4 for camera unit, 4 for illumination unit and 1 for USB connection).

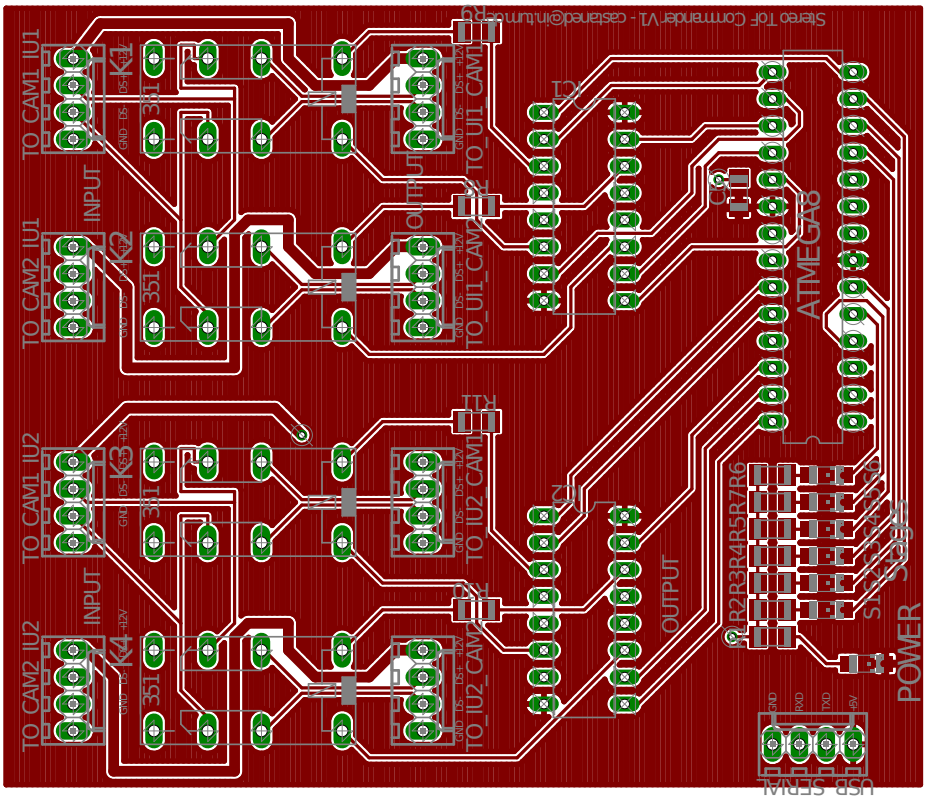


Figure D.1.: PCB print of the front part of the circuit

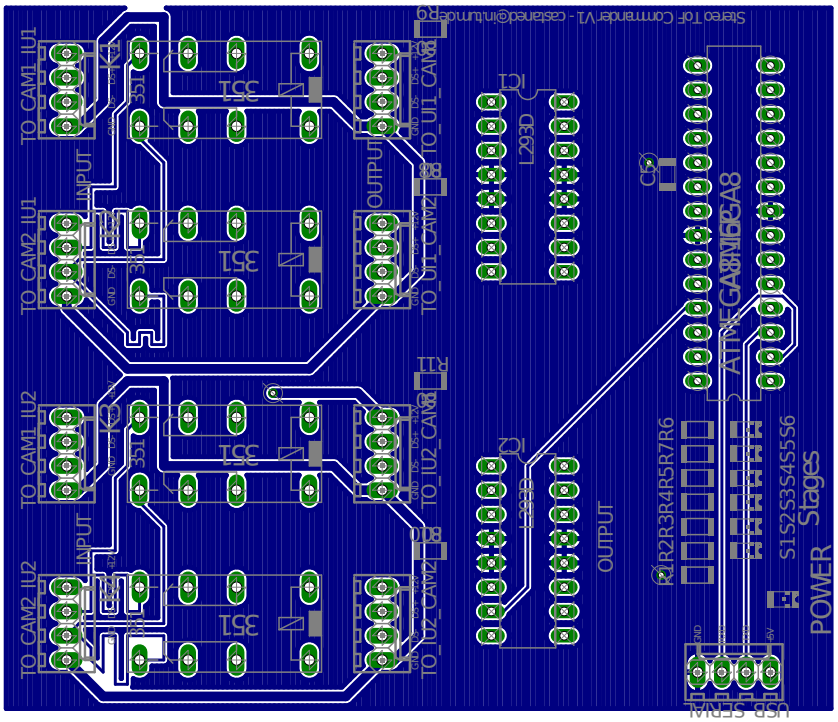


Figure D.2.: PCB print of the back part of the circuit

E. List of (Co-)Authored Publications

Related to this Thesis:

- "Reconstructing the Esophagus Surface from Endoscopic Image Sequences", V. Castañeda, S. Atasoy, D. Mateus, N. Navab, A. Meining, 5th Russian-Bavarian Conference on Bio-Medical Engineering, Munich, Germany, July 1-4 2009.
- "SLAM combining ToF and High-Resolution cameras", V. Castañeda, D. Mateus, N. Navab, IEEE Workshop on Motion and Video Computing, Winter Vision Meetings, Hawaii, January 2011.
- "Stereo Time-of-Flight", V. Castañeda, D. Mateus, N. Navab, IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, November 2011.
- "Method of Enhanced Depth Image Acquisition", Authors: V. Castañeda, D. Mateus, N. Navab, International patent PCT.

Other publications:

- "Manifold Learning for ToF-based Human Body Tracking and Activity Recognition", L. Schwarz, D. Mateus, V. Castañeda, N. Navab, British Machine Vision Conference (BMVC), Aberystwyth, United Kingdom, August 2010.
- "Skin Lesions Classification with Optical Spectroscopy", A. Safi, V. Castañeda, T. Lasser, N. Navab, 5th International Workshop on Medical Imaging and Augmented Reality, September 2010, Beijing.
- "Manifold Learning for Dimensionality Reduction and Clustering of Skin Spectroscopy Data", A. Safi, V. Castañeda, T. Lasser, D. Mateus, N. Navab, Proceedings of SPIE (2011).
- "Computer-Aided Diagnosis of Pigmented Skin Dermoscopic Images", A. Safi, M. Baust, O. Pauly, V. Castañeda, T. Lasser, D. Mateus, N. Navab, R. Hein, M. Ziai, MICCAI Workshop on Medical Content-based Retrieval for Clinical Decision Support, Toronto, Canada, September 2011.

Bibliography

[1] http://pro.jvc.com/prof/attributes/features.jsp?model_id=MDL101309.

[2] http://www.primesense.com/.

[3] http://www.xbox.com/en-US/hardware/k/kinectforxbox360/.

[4] http://panasonic-electric-works.net/D-IMager/index.html.

[5] *Multiple View Geometry in Computer Vision Second Edition*, chapter 3D Reconstruction of Cameras and Structure, pages 262–278. Cambridge University Press, 2004.

[6] S. Atasoy, B. Glocker, S. Giannarou, D. Mateus, A. Meining, GZ. Yang, and N. Navab. Probabilistic region matching in narrow-band endoscopy for targeted optical biopsy. In *Int. Conf. on Medical Image Computing and Computer Assisted Interventions (MICCAI)*, pages 499–506, 2009.

[7] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II State of the Art. *IEEE Robotics & Automation Magazine*, 13(3):108–117, August 2006.

[8] B. Bartczak, K Koeser, F. Woelk, and R. Koch. Extraction of 3d freeform surfaces as visual landmarks for real-time tracking. *J. of Real Time Image Processing*, 2(2-3):81–101, November 2007.

[9] B. Bartczak, I. Schiller, C. Beder, and R. Koch. Integration of a time-of-flight camera into a mixed reality system for handling dynamic scenes, moving viewpoints and occlusions in real-time. In *International Symposium on 3D Data Processing, Visualization and Transmission*, Atlanta, GA, USA, June 2008.

[10] C. Beder, B. Bartczak, and R. Koch. A combined approach for estimating patchlets from pmd depth images and stereo intensity images. In *Annual Sym. of the German Assoc. for Pattern Recognition (DAGM)*, pages 11–20, 2007.

[11] C. Beder and R. Koch. Calibration of focal length and 3d pose based on the reflectance and depth image of a planar object. *Int. J. of Intelligent Systems Technologies and Applications (IJISTA)*, 5(3/4):285–294, 2008.

[12] C. Beder, I. Schiller, and R. Koch. Real-time estimation of the camera path from a sequence of intrinsically calibrated pmd depth images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVII:45–50, 2008.

[13] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 943–948, 2004.

[14] M. Böhme, M. Haker, T. Martinetz, and E. Barth. Shading constraint improves accuracy of time-of-flight measurements. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, pages 1–6, 2008.

[15] M. Böhme, M. Haker, T. Martinetz, and E. Barth. Shading constraint improves accuracy of time-of-flight measurements. *Computer Vision and Image Understanding (CVIU)*, 114(12):1329–1335, 2010.

[16] B. Büttgen, T. Oggier, and M. Lehmann. CCD/CMOS lock-in pixel for range imaging: challenges, limitations and state-of-the-art. In *1st range imaging research day*, pages 21–32, 2005.

[17] R. Crabb, C. Tracey, A. Puranik, and J. Davis. Real-time foreground segmentation via range and color imaging. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, 2008.

[18] Y. Cui, S. Schuon, C. Derek, S. Thrun, and C. Theobalt. 3d shape scanning with a time-of-flight camera. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, (USA)*, 2010.

[19] W.L. Curvers, R. Singh, LM Song, H.C. Wolfsen, K. Ragunath, K. Wang, M.B. Wallace, P. Fockens, and J. Bergman. Endoscopic tri-modal imaging for detection of early neoplasia in barrett's esophagus: a multi-centre feasibility study using high-resolution endoscopy, autofluorescence imaging and narrow band imaging incorporated in one endoscopy system. *Int. J. of Gastroentorology and Hepatology, Gut*, 57(2):167, 2008.

[20] A. J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052–1067, June 2007.

[21] P.R.R. Devarakota, M. Castillo-Franco, R. Ginhoux, B. Mirbach, S. Kater, and B. Ottersten. 3-d-skeleton-based head detection and tracking using range images. *IEEE Transactions on Vehicular Technology*, 58(8):4064–4077, 2009.

[22] P.R.R. Devarakota, M. Castillo-Franco, R. Ginhoux, B. Mirbach, and B. Ottersten. Application of the reeb graph technique to vehicle occupant's head detection in low-resolution range images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, (USA)*, pages 1–8. IEEE, 2007.

[23] J. Diebel and S. Thrun. An application of markov random fields to range sensing. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, Cambridge, MA, 2005. MIT Press.

[24] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I The Essential Algorithms. *IEEE Robotics & Automation Magazine*, 13(2):99–110, June 2006.

[25] D. Falie. 3d image correction for time of flight (tof) cameras. In *Int. Conf. of Optical Instrument and Technology*, 2008.

[26] D. Falie and V. Buzuloiu. Distance errors correction for the time of flight (tof) cameras. In *European Conf. on Circuits and Systems for Communications*, 2008.

[27] D. Falie, M. Ichim, and L. David. Respiratory motion visualization and the sleep apnea diagnosis with the time of flight (tof) camera. *Int. Conf. on Visualisation, Imaging and Simulation (VIS)*, 2008.

[28] R.C. Fitzgerald. Review article: Barrett's esophagus and associated adenocarcinoma uk perspective. *Aliment Pharmacol Ther 20, UK*, 8:45–49, 2004.

[29] Grant R. Fowles. *Introduction to Modern Optics*. Dover Publications, 2 edition, June 1989.

[30] M. Fritzsche, M. Oberländer, T. Schwarz, B. Woltermann, B. Mirbach, and H. Riedel. Vehicle occupancy monitoring with optical range-sensors. In *Proc. IEEE Intelligent Vehicles Symp.*, 2001.

[31] S. Fuchs. Multipath interference compensation in time-of-flight camera images. In *Int. Conf. on Pattern Recognition (ICPR)*, pages 3583–3586, Washington, DC, USA, 2010. IEEE Computer Society.

[32] S. Fuchs and G. Hirzinger. Extrinsic and depth calibration of tof-cameras. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, 2008.

[33] S. Fuchs and S. May. Calibration and registration for precise surface reconstruction with time-of-flight cameras. *Int. J. on Intell. Systems Techn. and App., Issue on Dynamic 3D Imaging 5*, 2008.

[34] O. Gallo, R. Manduchi, and A. Rafii. curb and ramp detection for safe parking using the canesta tof camera. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, 2008.

[35] Li Guan, J.S. Franco, and M. Pollefeys. 3d object reconstruction with heterogeneous sensor data. In *International Symposium on 3D Data Processing, Visualization and Transmission*, 2008.

[36] S.A. Gudmundsson, H. Aanaes, and R. Larsen. Environmental effects on measurement uncertainties of time-of-flight cameras. In *IEEE Sym. on Signals Circuits and Systems (ISSCS), session on Alg. for 3D ToF-cameras*, 2007.

[37] S.A. Gudmundsson, H. Aanaes, and R. Larsen. Fusion of stereo vision and time-of-flight imaging for improved 3d estimation. *Int. J. of Intelligent Systems Technologies and Applications (IJISTA)*, 5(3/4):425–433, 2008.

[38] S.A. Gudmundsson, R. Larsen, H. Aanaes, M. Pardás, and J.R. Casas. TOF imaging in smart room environments towards improved people tracking. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, jun 2008.

[39] U. Hahne and M. Alexa. Combining time-of-flight depth and stereo images without accurate extrinsic calibration. *Int. J. of Intelligent Systems Technologies and Applications (IJISTA)*, 5(3/4):325–333, 2008.

[40] M. Haker, M. Böhme, T. Martinetz, and E. Barth. Geometric invariants for facial feature tracking with 3d tof cameras. In *IEEE Sym. on Signals Circuits & Systems (ISSCS), session on Alg. for 3D ToF cameras*, pages 109–112, 2007.

[41] M. Haker, M. Böhme, T. Martinetz, and E. Barth. Deictic gestures with a time-of-flight camera. In *Gesture in Embodied Communication and Human-Computer Interaction - International Gesture Workshop GW*, 2009.

[42] D. Hansen, M. Hansen, M. Kirschmeyer, R. Larsen, and D. Silvestre. Cluster tracking with time-of-flight cameras. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, 2008.

[43] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *RGB-D: Advanced Reasoning with Depth Cameras Workshop in conjunction with RSS, Zaragoza, Spain*, 2010.

[44] M. Holte and T. Moeslund. View invariant gesture recognition using the csem swissranger sr-2 camera. *Int. J. on Intell. Systems Techn. and App.*, 3/4:295–303, 2008.

[45] M. Holte, T. Moeslund, and P. Fihl. Fusion of range and intensity information for view invariant gesture recognition. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, 2008.

[46] B. Huhle, P. Jenke, and W. Strasser. On-the-fly scene acquisition with a handy multi-sensor system. *Int. J. of Intelligent Systems Technologies and Applications (IJISTA)*, 5(3/4):255–263, 2008.

[47] B. Huhle, T. Schairer, P. Jenke, and W. Strasser. Robust non-local denoising of colored depth data. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, pages 1–7, 2008.

[48] B. Huhle, T. Schairer, P. Jenke, and W. Strasser. Fusion of range and color images for denoising and resolution enhancement with a non-local filter. *Computer Vision and Image Understanding (CVIU)*, 114(12):1336–1345, 2010.

[49] S. Hussmann, A. Hermanski, and T. Edeler. Real-time motion artifact suppression in tof camera systems. *IEEE Trans. Instrumentation and Measurement*, 60(5):1682–1690, 2011.

[50] P. Jenke, B. Huhle, and W. Straber. Self-localization in scanned 3dtv sets. In *3DTV07*, pages 1–4, 2007.

[51] R. Jensen, R. Paulsen, and R. Larsen. Analyzing gait using a time-of-flight camera. volume 5575 of *Lecture Notes in Computer Science*, pages 21–30. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-02230-2-3.

[52] T. Kahlmann, F. Remondino, and S. Guillaume. Range imaging technology: new developments and applications for people identification and tracking. volume 6491, 2007.

[53] M. Keller and A. Kolb. Real-time simulation of time-of-flight sensors. *J. of Simulation Modelling Practice and Theory*, 17(5):967–978, 2009.

[54] M. Keller, A. Kolb, and V. Peters. A simulation-framework for time-of-flight sensors. In *IEEE Sym. on Signals Circuits and Systems (ISSCS), session on Alg. for 3D ToF-cameras*, 2007.

[55] Y.M. Kim, D. Chan, C. Theobalt, and S. Thrun. Design and calibration of a multi-view tof sensor fusion system. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, pages 1–7, 2008.

[56] Y.M. Kim, C. Theobalt, J. Diebel, J. Kosecka, B. Micusik, and S. Thrun. Multi-view image and tof sensor fusion for dense 3d reconstruction. In *IEEE Workshop on 3-D Digital Imaging and Modeling (3DIM), co-hosted with ICCV 2009*, 2009.

[57] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *IEEE International Symposium on Mixed and Augmented Reality*, Nara, Japan, 2007.

[58] R. Koch and J. Evers-Senne. *View Synthesis and Rendering Methods*. 2005.

[59] R. Koch, I. Schiller, B. Bartczak, F. Kellner, and K Koeser. Mixin3d: 3d mixed reality with tof-camera. In *Dynamic 3D Imaging DAGM 2009 Workshop, Dyn3D, LNCS 5742*, pages 126–141, Jena, Germany, September 2009.

[60] K Koeser, B. Bartczak, and R. Koch. Robust gpu-assisted camera tracking using free-form surface models. *J. of Real Time Image Processing*, 2(2-3):133–147, November 2007.

[61] P. Kohli, J. Rihan, M. Bray, and P. Torr. Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *Int. J. of Computer Vision (IJCV)*, 79:285–298, 2008. 10.1007/s11263-007-0120-6.

[62] A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-flight sensors in computer graphics. In *Proc. Eurographics (State-of-the-Art Report)*, 2009.

[63] K.D. Kuhnert and M. Stommel. Fusion of stereo-camera and pmd-camera data for real-time suited precise 3d environment reconstruction. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4780–4785, 2006.

[64] R. Lange. *3D time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology*. PhD thesis, University of Siegen, 2000.

[65] F. Lenzen, H. Schäfer, and C. Garbe. Denoising time-of-flight data with adaptive total variation. In *International Symposium on Visual Computing*. Springer, 2011. in press.

[66] A. G. Linarth, J. Penne, B. Liu, O. Jesorsky, and R. Kompe. Fast fusion of range and video sensor data. *J. of Advanced Microsystems for Automotive Applications*, (2):119–134, 2007.

[67] M. Lindner and A. Kolb. Lateral and depth calibration of PMD-distance sensors. In *J. of Advances in Visual Computing*, pages II: 524–533, 2006.

[68] M. Lindner and A. Kolb. Calibration of the intensity-related distance error of the PMD TOF-Camera. In *Intelligent Robots and Computer Vision XXV, The International Society for Optical Engineering (SPIE)*, volume 6764, pages 6764–35, 2007.

[69] M. Lindner and A. Kolb. Compensation of motion artifacts for time-of-flight cameras. In Andreas Kolb and Reinhard Koch, editors, *Dynamic 3D Imaging*, volume 5742 of *Lecture Notes in Computer Science*, pages 16–27. Springer Berlin / Heidelberg, 2009.

[70] M. Lindner, A. Kolb, and T. Ringbeck. New Insights into the Calibration of TOF Sensors. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, pages 1–5, 2008. DOI 10.1109/CVPRW.2008.4563172.

[71] M. Lindner, M. Lambers, and A. Kolb. Data Fusion and Edge-Enhanced Distance Refinement for 2D RGB and 3D Range Images. *Int. J. on Intell. Systems and Techn. and App. (IJISTA), Issue on Dynamic 3D Imaging*, 5(1):344 – 354, 2008.

[72] M. Lindner, I. Schiller, A. Kolb, and R. Koch. Time-of-flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding (CVIU)*, 114(12):1318–1328, 2010.

[73] R. Liu, Z. Li, and J. Jia. Image partial blur detection and classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, (USA)*.

[74] O. Lottner, K. Hartmann, W. Weihs, and O. Loffeld. Image registration and calibration aspects for a new 2d / 3d camera. In *EOS Conference on Frontiers in Electronic Imaging*, pages 80–81. European Optical Society, 2007.

[75] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. of Computer Vision (IJCV)*, 60:91–110, 2004.

[76] L. Maier-Hein, M. Schmidt, A.M. Franz, T.R. dos Santos, A. Seitel, B. Jähne, J. M. Fitzpatrick, and H-P. Meinzer. Accounting for anisotropic noise in fine registration of time-of-flight range data with high-resolution surface data. In *Int. Conf. on Medical Image Computing and Computer Assisted Interventions (MICCAI)*, pages 251–258, 2010.

[77] S. May, D. Droeschel, S. Fuchs, D. Holz, and A. Nüchter. Robust 3d-mapping with time-of-flight cameras. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1673–1678. IEEE, 2009.

[78] S. May, D. Droeschel, D. Holz, S. Fuchs, E. Malis, A. Nüchter, and J. Hertzberg. Three-dimensional mapping with time-of-flight cameras. *Journal of Field Robotics, Special Issue on Three-Dimensional Mapping, Part 2*, 26(11-12):934–965, December 2009.

[79] M Mechat and B. Büttgen. Realization of multi-3d-tof camera environments based on coded-binary sequences modulation. In *Proceedings of the Conference on Optical 3-D Measurement Techniques*, 2007.

[80] A. Medina, F. Gayá, and F. del Pozo. Compact laser radar and three-dimensional camera. *J. of Optical Society of America A (JOSA A)*, 23(4):800–805, Apr 2006.

[81] C. Mei, G. Sibley, M. Cummins, P.M. Newman, and I.D. Reid. A constant time efficient stereo SLAM system. In *British Machine Vision Conference (BMVC)*, 2009.

[82] E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, and E. Pagello. Range-only SLAM with a mobile robot and a wireless sensor networks. In *Int. Conf. on Robotics and Automation (ICRA)*, pages 1699–1705, 2009.

[83] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Conference on Artificial Intelligence (AAAI)*, pages 593–598, 2002.

[84] P. Mountney, S. Giannarou, D. Elson, and GZ. Yang. Optical biopsy mapping for minimally invasive cancer screening. In *Int. Conf. on Medical Image Computing and Computer Assisted Interventions (MICCAI)*, pages 483–490, 2009.

[85] P. Mountney, D. Stoyanov, A. J. Davison, and GZ. Yang. Simultaneous stereoscope localization and soft-tissue mapping for minimal invasive surgery. In *Int. Conf. on Medical Image Computing and Computer Assisted Interventions (MICCAI)*, pages 347–354, 2006.

[86] F. Mourgues, F. Devernay, and E. Coste-Maniére. 3D reconstruction of the operating field for image overlay in 3d-endoscopic surgery. In *IEEE and ACM International Symposium on Augmented Reality (ISAR)*, page 191, USA, 2001.

[87] R.A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera davison. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, (USA)*, 2010.

[88] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM - 3D mapping outdoor environments: Research articles. *J. Field Robot*, 24(8-9):699–722, 2007.

[89] T. Oggier, B. Büttgen, F. Lustenberger, G. Becker, B. Rüegg, and A. Hodac. Swissranger sr3000 and first experiences based on miniaturized 3d-tof cameras. In *First Range Imaging Research Day at ETH Zurich*, 2005.

[90] S. Oprisescu, D. Falie, M. Ciuc, and V. Buzuloiu. Measurements with tof cameras and their necessary corrections. In *Proceedings of the IEEE International Symposium on Signals, Circuits and Systems (ISSCS)*, 2007.

[91] D. Pangercic, R. Bogdan, and M. Beetz. 3D-based monocular SLAM for mobile agents navigating in indoor environments. In *13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Germany, 2008.

[92] J. Penne, C. Schaller, R. Engelbrecht, L. Maier-Hein, B Schmauss, H-P. Meinzer, and J. Hornegger. Laparoscopic Quantitative 3D Endoscopy for Image Guided Surgery. In Hans-Peter Meinzer, Thomas Martin Deserno, Heinz Handels, and Thomas Tolxdorff, editors, *Bildverarbeitung für die Medizin*, pages 16–20, Berlin, 2010.

[93] J. Penne, C. Schaller, J. Hornegger, and T. Kuwert. Robust real-time 3d respiratory motion detection using time-of-flight cameras. *Computer Assisted Radiology and Surgery 3, 5*, pages 427–431, 2008.

[94] J. Penne, S. Soutschek, L. Fedorowicz, and J. Hornegger. Robust real-time 3d time-of-flight based gesture navigation. In *IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2008.

[95] V. Peters, O. Loffeld, K. Hartmann, and S. Knedlik. Modeling and bistatic simulation of a high resolution 3d pmd-camera. In *Proc. Congress on Mod- elling and Simulation (EUROSIM)*, 2007.

[96] A. Prusak, O. Melnychuk, I. Schiller, H. Roth, and R. Koch. Pose estimation and map building with a pmd-camera for robot navigation. *International Journal of Intelligent Systems Technologies and Applications*, 5(3-4):355–364, 2008.

[97] C.H. Quartucci and C.L. Tozzi. Towards 3d reconstruction of endoscope images using shape from shading. In *Computer Graphics and Image Processing*, pages 90–96. Brazil, 2000.

[98] J. Radmer, P. M. Fusté, and J. Krüger. Incident light related distance error study and calibration of the PMD-range imaging camera. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, 2008.

[99] H. Rapp. Experimental and theoretical investigation of correlating tof-camera systems. Master's thesis, 2007.

[100] C. Schaller, Adelta, J. Penne, and J. Hornegger. Time-of-flight sensor for patient positioning. In *Medical Imaging: Visualization, Image-Guided Procedures, and Modeling, The International Society for Optical Engineering (SPIE)*, volume 7258, 2009.

[101] C. Schaller, J. Penne, and J. Hornegger. Time-of-flight sensor for respiratory motion gating. *J. of Medical Physics*, 35(7):3090–3093, 2008.

[102] I. Schiller, C. Beder, and R. Koch. Calibration of a pmd camera using a planar calibration object together with a multi-camera setup. *The Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXI. ISPRS Congress*, 2008.

[103] M. Schmidt, M. Jehle, and B. Jähne. Range flow estimation based on photonic mixing device data. *Int. J. of Intelligent Systems Technologies and Applications (IJISTA)*, 5(3/4):380–392, 2008.

[104] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. High-quality scanning using time-of-flight depth superresolution. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, pages 1–7, 2008.

[105] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. Lidarboost: Depth superresolution for tof 3d shape scanning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Florida, (USA)*, pages 343–350, 2009.

[106] P. Sharma, A. Bansal, S. Mathur, S. Wani, R. Cherian, D. McGregor, A. Higbee, S. Hall, and A. Weston. The utility of a novel narrow band imaging endoscopy system in patients with barrett's esophagus. *Gastrointestinal Endoscopy*, 64:167–175, 2006.

[107] S. Soutschek, J. Penne, and J. Hornegger. 3d gesture-based scene navigation in medical imaging applications using time-of-flight cameras. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, 2008.

[108] O. Steiger, J. Felder, and S. Weiss. Calibration of time-of-flight range imaging cameras. pages 1968–1971, 2008.

[109] D. Stoyanov, A. Darzi, and G. Yang. Dense 3d depth recovery for soft tissue deformation during robotically assisted laparoscopic surgery. *Int. Conf. on Medical Image Computing and Computer Assisted Interventions (MICCAI)*, pages 41–48, 2004.

[110] H. Strasdat, J.M.M. Montiel, and A. J. Davison. Real-time monocular SLAM: Why filter? In *Int. Conf. on Robotics and Automation (ICRA)*, pages 2657–2664, 2010.

[111] B. Streckel, B. Bartczak, R. Koch, and A. Kolb. Supporting structure from motion with a 3d-range-camera. In *Scandinavian Conf. Image Analysis (SCIA)*, pages 233–242, 2007.

[112] B. Streckel, B. Bartczak, R. Koch, and A. Kolb. Supporting structure from motion with a 3D-range-camera. In *Scandinavian Conf. Image Analysis (SCIA)*, LNCS, pages 233–242. Springer, 2007.

[113] N. Sunderhauf, S. Lange, and P. Protzel. Using the unscented kalman filter in mono-slam with inverse depth parametrization for autonomous airship control. In *IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR)*, pages 1–6, Sept. 2007.

[114] A. Swadzba, N. Beuter, S. Wachsmuth, and F. Kummert. Dynamic 3d scene analysis for acquiring articulated scene models. In *Int. Conf. on Robotics and Automation (ICRA)*, Anchorage, AK, USA, 2010. IEEE, IEEE.

[115] J. Thielemann, G. Breivik, and A. Berge. Pipeline landmark detection for autonomous robot navigation using time-of-flight imagery. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, 2008.

[116] R. Triebel and W. Burgard. Improving simultaneous localization and mapping in 3D using global constraints. In *Conference on Artificial Intelligence (AAAI)*, 2005.

[117] H. Wang, D. Mirota, M. Ishii, and G.D. Hager. Robust motion estimation and structure recovery from endoscopic image sequences with an adaptive scale kernel consensus estimator. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, (USA)*. IEEE, 2008.

[118] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[119] D. Witzner, R. Larsen, and F. Lauze. Improving face detection with tof cameras. In *IEEE Sym. on Signals Circuits & Systems (ISSCS), session on Alg. for 3D ToF cameras*, pages 225–228, 2007.

[120] C.H. Wu, Y.N. Sun, Y.C. Chen, and C.C. Chang. Endoscopic feature tracking and scale-invariant estimation of soft-tissue structures. *IEICE transactions on information and systems*, 91(2):351–360, 2008.

[121] G. Yahav, G. J. Iddan, and D. Mandelboum. 3d imaging camera for gaming application. In *Consumer Electronics, 2007. ICCE 2007. Digest of Technical Papers. International Conference on*, pages 1–2, 2007.

[122] Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-depth super resolution for range images. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, (USA)*, 0:1–8, 2007.

[123] Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-depth super resolution for range images. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2007.

[124] T. Yoshida, H. Inoue, S. Usui, H. Satodate, N. Fukami, and S. Kudo. Narrow-band imaging system with magnifying endoscopy for superficial esophageal lesions* 1. *Journal on Gastrointestinal endoscopy*, 59(2):288–295, 2004.

[125] J. Zhou, A. Das, F. Li, and B. X. Li. Circular generalized cylinder fitting for 3D reconstruction in endoscopic imaging based on MRF. In *IEEE Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW), Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 1–8, 2008.

[126] J. Zhou, Q. Zhang, B. Li, and A. Das. Synthesis of stereoscopic views from monocular endoscopic videos. In *IEEE Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW), Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 55–62. IEEE, 2010.

[127] J. Zhu, L. Wang, J. Gao, and R. Yang. Spatial-temporal fusion for high accuracy depth maps using dynamic mrfs. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 32:899–909, 2010.

[128] J. Zhu, L. Wang, R. Yang, and J. Davis. Fusion of time-of-flight depth and stereo for high accuracy depth maps. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, pages 1–8, 2008.

[129] Y. Zhu, B. Dariush, and K. Fujimura. Controlled human pose estimation from depth image streams. In *Proceeding of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Workshop on ToF Camera based Computer Vision (TOF-CV)*, 2008.