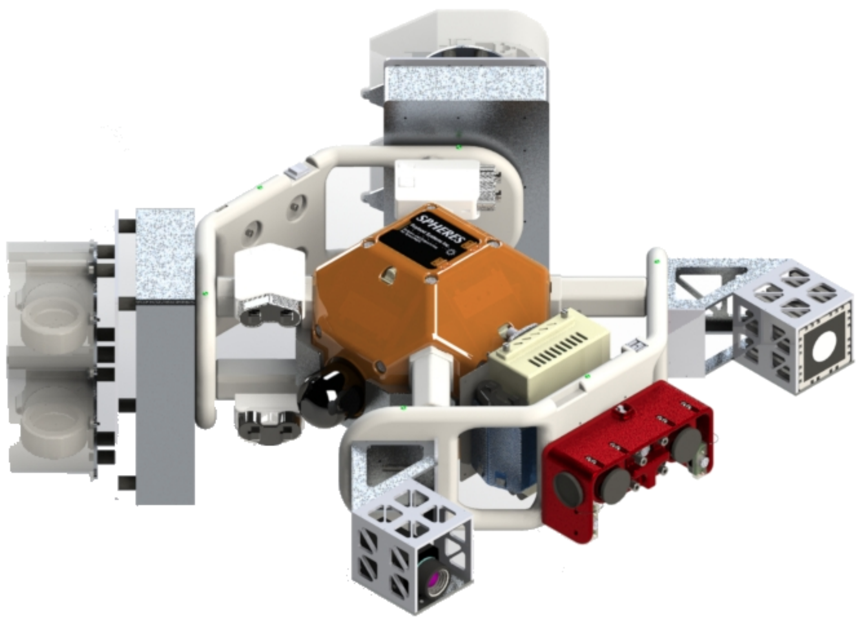
Calibration and Fusion of Stereoscopic and Time-of-Flight Cameras for Zero Gravity Targets Inspection

by

Gabriel P. Urbain



Submitted to the Department of Electronics, Optronics and Signal Processing, ISAE
in partial fulfillment of the
requirements for a double degree of

Master of Science in Aerospace Engineering at Supaero, ISAE, France

and

Master of Science in Electrical Engineering at Faculté Polytechnique, UMONS, Belgium

October 2014

Calibration and Fusion of Stereoscopic and Time-of-Flight Cameras for Zero Gravity Targets Inspection

by

Gabriel P. Urbain

Submitted to the Department of Electronics, Optronics and Signal Processing (DEOS)
in partial fulfillment of the requirements for the degree of

Master of Science in Aerospace Engineering

at the

Ecole Nationale Supérieure de l'Aéronautique et de l'Espace (ISAE-SUPAERO)

October 2014

Author . . . .
Department of Electronics, Optronics and Signal Processing (DEOS)
October 23, 2014

Certified by . . . .
Alvar Saenz-Otero
Principal Research Scientist, Space Systems Laboratory, MIT
Thesis Supervisor

Certified by . . . .
Daniel Alazard
Professor, Department of Mathematics, Computer Science and Control (DMIA), ISAE Supaero
Thesis Supervisor

Calibration and Fusion of Stereoscopic and Time-of-Flight Cameras for Zero Gravity Targets Inspection

by

Gabriel P. Urbain

Submitted to the Department of Electronics, Optronics and Signal Processing (DEOS)
on October 23, 2014, in partial fulfillment of the
requirements for the degree of
Master of Science in Aerospace Engineering

Abstract

In many areas of robotics, vision is becoming more and more common in applications such as localization, automatic map construction, autonomous navigation, path following, inspection, monitoring or risky situation detection. With the increasing performance of embedded computers, the development of faster algorithms and the apparition of new types of devices in the last few years, multi-sensor data fusion is considered as an opportunity to take better advantage of different sensors features to stretch the limits. This project aims at implementing a multi-sensor data fusion algorithm involving two stereoscopic cameras and a Time-of-Flight camera (ToF) in nano-satellites called SPHERES.

This document is the result of a five-month internship at the MIT SSL, USA as part of the final project of a double Master degree in Aerospace Engineering at ISAE, France, and in Electrical Engineering at UMONS, Belgium. The first chapter introduces the goal and the context of the project. The second chapter is dedicated to the theoretical aspect and aims at summarizing the required mathematical background. A third chapter analyzes concretely the implementation and finally, the results of two different experiment sets will be detailed in the fourth chapter before concluding.

Thesis Supervisor: Alvar Saenz-Otero
Title: Principal Research Scientist, Space Systems Laboratory, MIT

Thesis Supervisor: Daniel Alazard
Title: Professor, Department of Mathematics, Computer Science and Control (DMIA), ISAE Supaero

Acknowledgments

Je remercie toutes les personnes remarquables qui m'ont aidé tout au long de ce stage, à commencer par le Dr. Alvar Saenz-Otero, qui m'a accueilli dans son service et a toujours sacrifié de son temps précieux afin de m'orienter, tous les membres du SSL pour leur aide et leurs conseils, le professeur Daniel Alazard pour sa participation dans le démarrage du projet, Marina G. March qui a entrepris cette expérience à mes côtés ainsi que tous mes professeurs à Toulouse et à Mons, et plus particulièrement Stéphanie Lizy-Destrez et Thierry Dutoit qui m'ont aidé personnellement dans tous mes projets.

J'aimerais aussi exprimer toute ma gratitude envers David Steinberg, Alice Malter, Jean-François Toubeau et mon père pour leur relecture de ce document mais aussi GDF Suez, l'Université de Mons et la Fondation Fernand Lazard pour l'apport financier qui m'a permis de mener à bien ce stage aux Etats-Unis.

Enfin, un merci tout particulier va à mes parents, ma soeur et mes amis de Mons et Toulouse qui, à travers leur soutien, leur énergie et leur sourire, ont transformé mes six années universitaires en une expérience inoubliable.

Contents

List of Figures

Glossary

API Application Programming Interface.

CMG Control Moment Gyroscope.

DoF Degree of Freedom.

EVA Extra Vehicular Activity.

FoV Field-of-View.

HEOMD NASA Human Exploration and Operations Mission Directorate.

INSPECT Integrated Navigation Sensor Platform for EVA Control and Testing.

ISAE Instititut Superieur de l'Aeronautique et de l'Espace.

ISS International Space Station.

MIT Massachusetts Institute of Technology.

ORF Optical Range Finder.

PCL Point Cloud Library.

RGA Reduced Gravity Aircraft.

SDK Software Development Kit.

SLAM Simultaneous Localization and Mapping.

SPHERES Synchronized Position Hold, Engage, Reorient Experimental Satellites.

SSL Space System Laboratory.

ToF Time-of-Flight.

TRL Technology Readiness Level.

VERTIGO Visual Estimation for Relative Tracking and Inspection of Generic Objects.

Chapter 1

Introduction

This document is the result of five-month internship at the Massachusetts Institute of Technology Space System Laboratory (MIT SSL) as part of the final project of a double Master degree in Aerospace Engineering, Space Systems and Telecommunications major, at Institut Supérieur de l'Aéronautique et de l'Espace (ISAE), formation SUPAERO and Electrical Engineering, Telecommunications and Multimedia major at Faculté Polytechnique of the University of Mons (UMONS). The first chapter introduces the goal and the context of the project to give the reader a global overview of the state-of-art, the SSL experience and facilities as well as the new equipment this project focuses on. The second chapter is dedicated to the theoretical aspect and aims at summarizing the required mathematical background and development describing fully and unambiguously every processes. A third chapter analyzes concretely the implementation of the same mechanism from a programmer's point of view. Finally, the results of two different experiment sets will be detailed in the fourth chapter before giving conclusion and future perspectives.

1.1 Context

In many areas of robotics, vision is becoming more and more common in applications such as localization, automatic map construction, autonomous navigation, path following, inspection, monitoring or risky situation detection [2]. However, given the features of the cheap sensors currently on the market in the fields of robotics or unmanned vehicles, computer vision is a challenging domain for space navigation. With the increasing performance of embedded computers, the development of faster algorithms and the apparition of new types of devices in the last few years, multi-sensor data fusion is considered as an opportunity to take better advantage of different sensors features to stretch the limits. This project aims at implementing a multi-sensor data fusion algorithm on the space SPHERES testbed, as part of the INSPECT project. In this perspective, the two stereoscopic cameras of VERTIGO and a new Time-of-Flight (ToF) camera, also called Optical Range Finder (ORF), have been fixed to the Halo hardware of the SPHERES nano-satellites. In the future, a thermographic camera will be added to the assembly to increase the data diversity and lead to better results in very dark environments such as spacecrafts in the shadow.

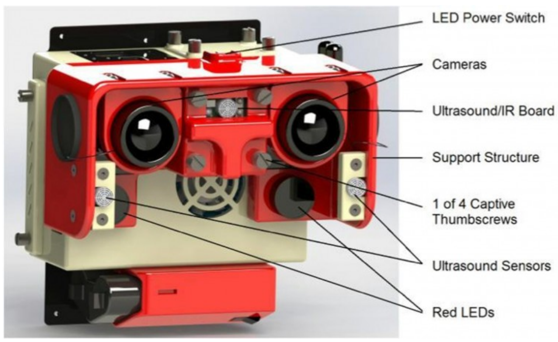
1.1.1 SPHERES

SPHERES is a facility for demonstrating advanced satellite formation flight, docking, and autonomy algorithms aboard the International Space Station (ISS). Created in 1999, SPHERES was one of the first educational programs that launched student-designed hardware to the ISS in 2006. The ISS contains three SPHERES nano-satellites with fully functional propulsion, guidance, communications, and power systems which enable the nano-satellites to maneuver, communicate with each other and with a laptop control station, and to identify their relative positions.

Figure 1-1: One of the three SPHERES nano-satellites and a brief description of its interfaces.

1.1.2 VERTIGO

VERTIGO is an extension of the SPHERES satellite that develops computer vision navigation and mapping algorithms. Launched in 2012, the VERTIGO goggles add to the SPHERES satellites a set of stereoscopic cameras and a 1.2 GHz Linux computer that sense the depth of different features on the target object in much the same way as human eyes. Software algorithms running on the compact single-board computer are able to create a detailed three-dimensional map of the unknown, uncooperative and possibly spinning target and estimate its dynamics. The inspector satellite can use this knowledge to plan trajectories around the target achieving autonomous vision-based relative spacecraft navigation.



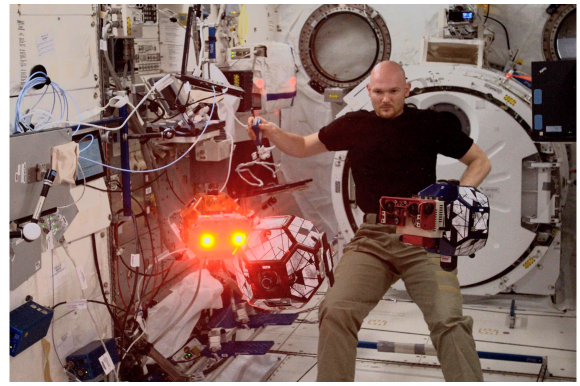


Figure 1-2: *Left*: the VERTIGO stack and goggles with a few explanations. *Right*; a test session in the ISS with two SPHERES equipped with the VERTIGO hardware.

1.1.3 Halo

Halo is a ring-shaped structure that is fastened around a SPHERES satellite and electrically connected to the VERTIGO computer. The structure is made out of several pieces of 3D-printed plastic and provides 22W of electrical power, 2 USB ports and one 1 Gb/s data connection to each one of the 6 expansion ports on the outer face of the ring. Using standardized interface and connector, a wide variety of robotic peripherals can be connected to the satellite, increasing the capability and flexibility of SPHERES, and allowing for a wider range of experiments to be conducted. A Time-of-Flight Camera (ToF) also called Optical Range Finder (ORF), a thermocamera and Control Moment Gyroscopes have been tested as part of the INSPECT program to augment the capabilities of an inspector satellite.

Figure 1-3: The Halo structure plugged around a SPHERES nano-satellite.

1.1.4 INSPECT

The NASA Human Exploration and Operations Mission Directorate (HEOMD) is invested in researching technologies that could reduce risks associated with astronaut spacewalks. One of these is an autonomous system capable of inspecting the exterior of space hardware, a common reason for sending an astronaut outside of the ISS. The INSPECT system, an Integrated Navigation Sensor Platform for Extravehicular Control and Testing, has been developed as a first step in the progression towards developing a system capable of operating outside of the ISS and serving the HEOMD's needs. The sensors on INSPECT were selected to fulfill the mission-level requirements to reduce risk by testing capability inside of the ISS prior to moving into the vacuum of space.

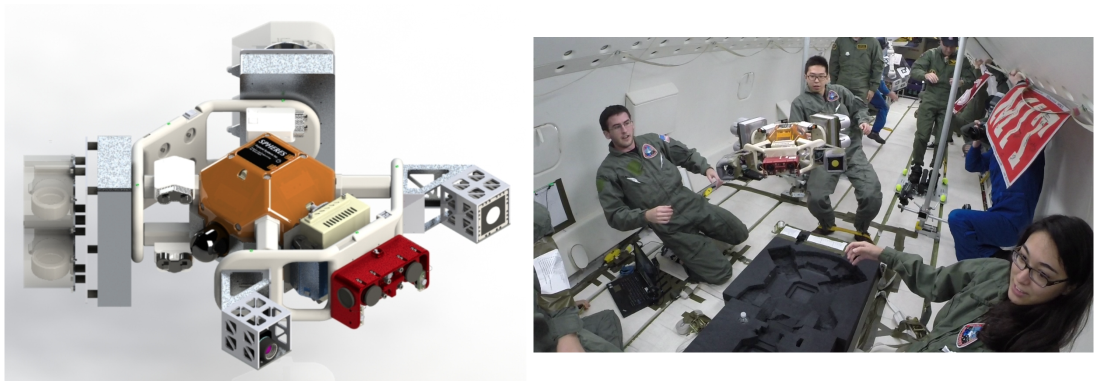


Figure 1-4: *Left*: a CAD model of the Halo proto-flight equipped with the four cameras and CMG's of the INSPECT project. *Right*: a RGA parabolic flight test session in July 2014 running the acquisition software developed during this internship.

1.1.5 Testing Environments

SPHERES takes advantage of three different environments to test new control and sensing algorithms as well as new pieces of hardware. In the *ground laboratory*, three SPHERES robots can move freely on an air-cushion table around three degrees of freedom (DoF). Besides, *parabolic flights* provided by NASA allow short test sessions in six DoF. Finally, in *the ISS*, groups of SPHERES tests are run by a crew member in test sessions which occur approximately once every three months.

1.2 Objectives

The goal of this work is to create a fusion algorithm taking advantage of VERTIGO stereo cameras and the ORF camera to provide a 3D cloud with a better accuracy and completeness than the one provided by each sensors separately and to demonstrate the feasibility of this algorithm on the ground and during a Reduced Gravity Aircraft (RGA) parabolic flights campaign.

1.2.1 Sensors

The sensors used in this project are part of a selection performed in previous projects in order to answer precise performance and Technology Readiness Levels (TRL). Their features are described in table 1.1.

	Stereo cameras	**ORF**	**Thermocam**
Brand	IDS-Imaging	MESA-Imaging	FLIR
Model	2x uEye LE 1225-M-HQ	SwissRange 4000	A5
Frequency Domain	$717nm$ dominant (visible)	$850nm$ (far IR)	$7.5 - 13\mu m$ (near IR)
Resolution	752x480 pixels	176x144 pixels	80x64 pixels
Pixel size	$6\mu m$	$40\mu m$	$50\mu m$
FPS	10 FPS (typical) to 87 FPS (max)	10 to 30 FPS (typical)	60 FPS (typical)
Range	N/A	0.1m to 7m	N/A
Horizontal FoV	35 degrees	69 degrees	44 degrees
Vertical FoV	35 degrees	55 degrees	36 degrees
Focal length		$10mm$ (typical)	$5mm$ (fixed)
Output Data	10 bits monochrome	14 bits depth, 16 bits visual 16 bits confidence	8 bits monochrome

Table 1.1: Brief summary of the sensors characteristics.

1.2.2 Global Architecture

This fusion algorithm and more broadly the machine vision of the SPHERES nano-satellites is part of a SLAM algorithm aiming at localizing the robots as well as mapping and understanding the geometry and the motion parameters of objects around the satellite (figure 1-5). Besides, in visual navigation, the outputs of this process are also the inputs of a motion control algorithm which can only be fully tested in zero-gravity environments. Therefore, in this topic, the motion of objects (defined by space dynamics) and the performance associated should be taken into account during the analysis of the results. Moreover, the luminous environment the satellite will evolve in is also very specific, especially during EVA, and particular attention on adaptivity to various conditions will be attached during test sessions.

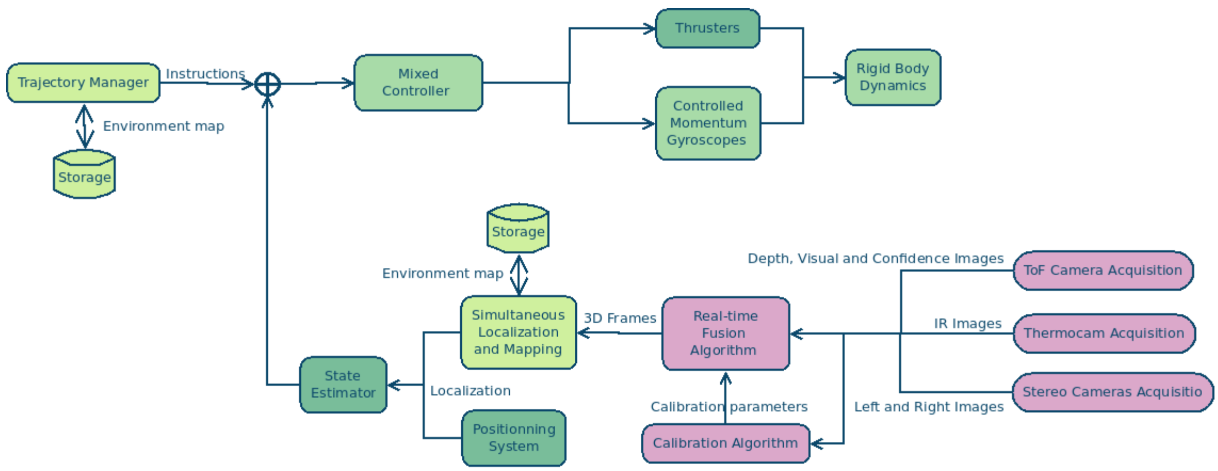


Figure 1-5: The algorithms developed in this thesis (in rose), situated in the control process of the INSPECT project. Sensor fusion 3D frames outputs are directly used in a SLAM process which to provide localization estimation but also a useful map for the trajectory manager.

Chapter 2

Theoretical Approach

In this chapter, we will try to give all the theoretical tools to understand the algorithm developments carried out in this project. The first section aims at reminding the reader of background models and theories but we will assume he masters his basics of mechanics of the rigid body, optics, image processing, numerical analysis and computer science. Section two focuses on the calibration algorithm. It reviews solutions found in the literature to calibrate separately stereoscopic cameras and a ToF camera but also describes the implementation and adaptation of a new algorithm proposed in [6] to calibrate the whole ToF and stereo cameras system while taking advantage of each sensors characteristics. Finally, the last section analyzes the core implementation of the fusion algorithm tested in this project, try to set out arguments to the choices that have been made and describes each parts in details.

2.1 Camera Models

Before going into further details into the algorithms breakdown, it may be necessary to clarify the models employed throughout this document. Indeed, to simplify the sensor fusion analysis and implementation, we will have to make numerous simplifications and hypothesis about the camera models impacting the result discussion in chapter 4.

2.1.1 Mathematical Notations

Several coordinate systems are used in this work leading themselves to different transformations between each others. To give a better overview, this paragraph summarizes the mathematical notations employed in this document.

Coordinates Systems:

- $\mathbf{T}$: Coordinate system of the ToF) camera (or ORF). In the 3D system, the origin is situated in the pinhole, the Z axis points forward, Y axis points down and X points right. In the 2D coordinate system, the origin is situated in the top left corner of the image plane, U points to the right and V points down.
- $\mathbf{R}$: 3D coordinate system of the right camera of the stereo rig. In the 3D system, the origin is situated in the pinhole, the Z axis points forward, Y axis points down and X points right. In the 2D coordinate system, the origin is situated in the top left corner of the image plane, U points to the right and V points down.
- $\mathbf{L}$: 3D coordinate system of the left camera of the stereo rig. In the 3D system, the origin is situated in the pinhole, the Z axis points forward, Y axis points down and X points right. In the 2D coordinate system, the origin is situated in the top left corner of the image plane, U points to the right and V points down.

Transformations Matrices: To represent both affine (translation and rotation) and projective transformations from one coordinate system to another, we use 4x4 transformation matrices. For instance, in the case of an affine transformation of a point P from R to L coordinate system, we write:

$$\begin{pmatrix} x_L \\ y_L \\ z_L \\ 1 \end{pmatrix} = M_{LR} * \begin{pmatrix} x_R \\ y_R \\ z_R \\ 1 \end{pmatrix} \tag{2.1}$$

Where:

$$M_{LR} = \begin{pmatrix} r_{XX} & r_{XY} & r_{XZ} & t_X \\ r_{YX} & r_{YY} & r_{YZ} & t_Y \\ r_{ZX} & r_{ZY} & r_{ZZ} & t_Z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.2}$$

Which allows to multiply matrices directly between each other:

$$M_{TR} = M_{TL} * M_{LR} \tag{2.3}$$

2.1.2 Pinhole Camera Model

According to [35], a common model of camera is composed of a lens represented by a single pinhole O in the *focal plane* $\mathcal{F}$ and a sensor matrix in the *image plane* $\mathcal{I}$ at a distance f from the focal plane. As represented on figure 2-1, O, also called optical center, is the origin of the world 3D coordinate system $OXYZ$ where Z is perpendicular to the focal plane and directed in the opposite direction of the image plane and X and Y are included in the plane. The pixels on the image plane are localized with a 2D coordinates system $O'U'V'$ where the origin is situated in the lower-right corner of the sensor matrix. The Z axis intersects $\mathcal{I}$ in a point $c' = (c'_U, c'_V)$ called the *principal point*.

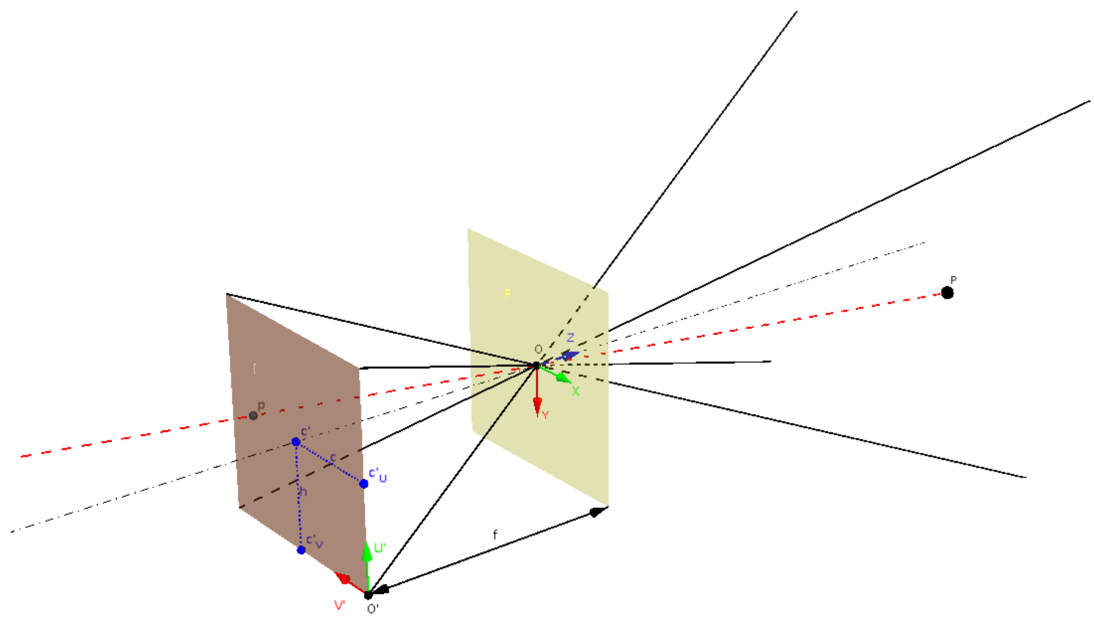


Figure 2-1: Geometry of the pinhole model for a single camera.

To facilitate the representation, we can consider a *virtual image plane* at a distance f on the positive Z axis, which does not change anything to the problem but helps to recreate directly a *projected image* with the same orientation than the real object. We now have a 2D coordinate system $O"UV$ (figure 2-2).

In this optimal model, the *intrinsic* geometry of the camera is completely represented by the parameters f, c_U and c_V measured in pixels. However, to make it more realistic, we can introduce extra parameters such as:

- The lens enlargement k, whose value is different along U and V axis and generally represented in the model through $f_U = k_U * f$ and $f_V = k_V * f$.
- The skew s_{UV}, assessing the non-orthogonality between rows and columns of the sensor photosensitive cells.

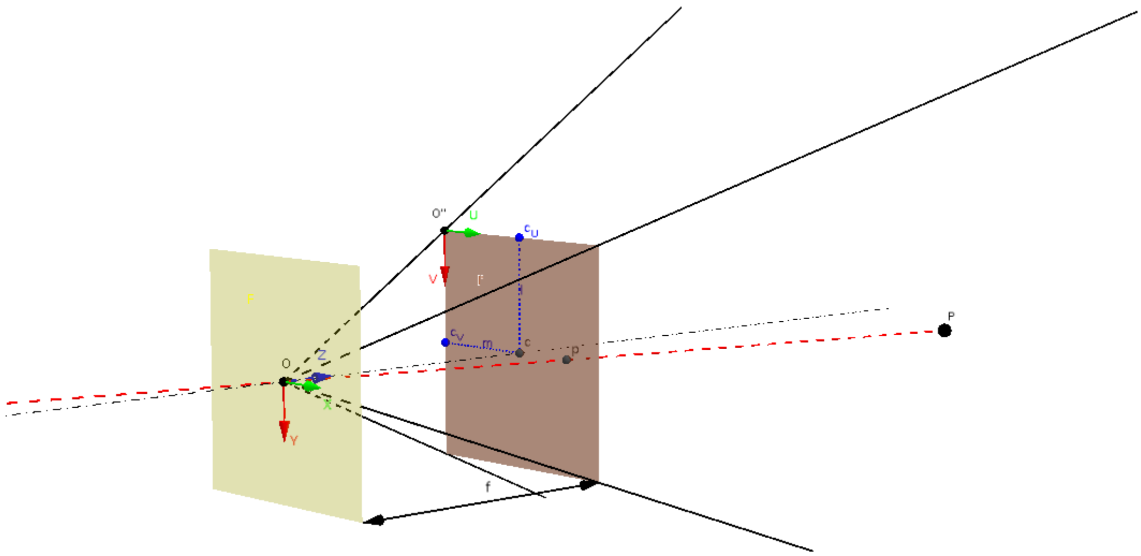


Figure 2-2: In this simplified representation, the point P is projected in the virtual plane.

Those five parameters f_U, f_V, c_U, c_V, s_{UV} constitute what we call the *intrinsic matrix* of the camera:

$$K = \begin{pmatrix} f_U & s_{UV} & c_U \\ 0 & f_V & c_V \\ 0 & 0 & 1 \end{pmatrix} \tag{2.4}$$

Therefore, we can write the affine transformation linking a *world point*, represented by its *homogeneous coordinates* in the camera 3D coordinate system and a *projected point* represented by its *homogeneous coordinates* in the image 2D coordinate system, considering a scaling factor s:

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{2.5}$$

Finally, the model can be refine to take the optical distortions into account. As proposed by Brown [4], the distortions may be divided into radial and tangential and can be represented by a second degree polynomial which maps the undistorted and distorted images together thanks to six parameters. This will be detailed in section 2.2.

One should also notice that the parameters discussed here define the mathematical model of the camera but performance of a camera can be determined as well. For instance, the *Field-Of-View* (FoV) of the camera and the pixel size ϵ_p of the photosensitive cells (sometimes given by the pixel density, in pixels/meters) are two characteristics to measure camera performance.

2.1.3 Stereoscopic Cameras Model

In the literature, the stereo camera are commonly represented as the assembly of two pinhole models whose focal points are separated by a distance called *baseline*. As in figure 2-3, we thus have now two 3D coordinate systems $L = O_L X_L Y_L Z_L$ and $R = O_R X_R Y_R Z_R$.

Projection

We can still use the model developed in the precedent paragraph but the rotation and the translation between L and R must be taken into account when a point is represented in 3D. An *extrincic matrix* is then defined to perform that transformation:

$$M = \begin{pmatrix} r_{X,X'} & r_{X,Y'} & r_{X,Z'} & t_X \\ r_{Y,X'} & r_{Y,Y'} & r_{Y,Z'} & t_Y \\ r_{Z,X'} & r_{Z,Y'} & r_{Z,Z'} & t_Z \end{pmatrix} \tag{2.6}$$

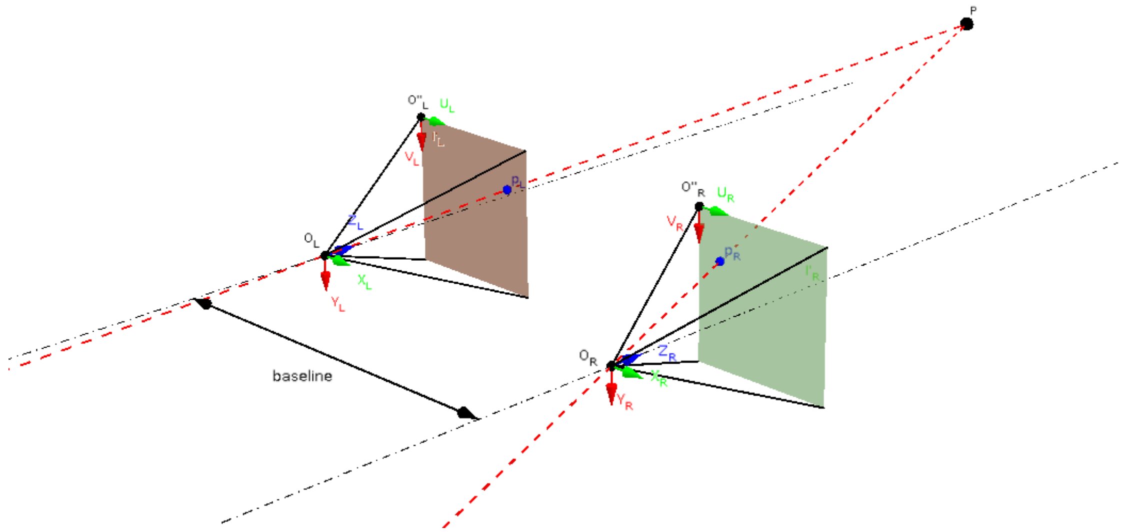


Figure 2-3: Model of an assembly of stereoscopic cameras.

Which gives:

$$s\begin{pmatrix}u\\v\\1\end{pmatrix} = K * M * \begin{pmatrix}x\\y\\z\\1\end{pmatrix} = P * \begin{pmatrix}x\\y\\z\\1\end{pmatrix} \tag{2.7}$$

Where P is also called the *projection matrix*.

When we write the equations for both left and right cameras, the *extrinsic matrix* can refer to a transformation in relation to a third coordinate system, either to L or R. In the last case, one of the two M matrix is useless. For instance, if we consider the world coordinate system as L, we now write:

$$
\begin{cases}
s\begin{pmatrix} u_L \\ v_L \\ 1 \end{pmatrix} = K_L * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \\
s\begin{pmatrix} u_R \\ v_R \\ 1 \end{pmatrix} = K_R * M_R * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}
\end{cases}
\tag{2.8}
$$

As for the camera model, those equations can be used to compute directly (u_L, v_L) and (u_R, v_R) from the 3D coordinates of a point with the knowledge of *intrinsic* and *extrinsic* matrices for both cameras. This process is known as ***projection***.

Triangulation

Intuitively, if inverting the pinhole equation was not directly useful with a mono camera because there was one DoF remaining, we can invert the stereo cameras model to find the 3D coordinates given the projected points in L and R. However, this process, known as ***triangulation***, requires the triangulated points to respect the *epipolar constraint* in order to give coherent results [13], i.e. (u_L, v_L) and (u_R, v_R) must be defined to ensure that the two *epipolar rays* from L and R cross in one point in the real world as represented in figure 2-3.

Practically, in the 3D reconstruction from stereo sensors problem, the points $p_L = (u_L, v_L)$ and $p_R = (u_R, v_R)$ in the focal images are found using image processing techniques. The precision of this method, the accuracy of the physical sensors, the precision of the calibration matrices, the numerical errors,... Everything makes this constraint difficult to respect. We can therefore consider two ways to overcome this issue:

Simplify the problem : In the first option, we suppose the cameras to be perfectly aligned in Y and Z coordinates, the *baseline* is measured on the X axis. Thus, *epipolar lines* are totally included in XZ planes for each points and as soon as those one are visible in left and right images, they will cross for sure. This leads to simplified projection matrices:

$$P_L = \begin{pmatrix} f & 0 & c_U & 0 \\ 0 & f & c_V & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.9}$$

$$P_R = \begin{pmatrix} f & 0 & c_U & t_{LR} \\ 0 & f & c_V & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.10}$$

Minimize errors : The other solution is to keep an elaborate model where left and right cameras can be misaligned but try to minimize the sum of euclidean errors when we are triangulating several points. Various algorithm concerning the subject have been analyzed in [13] or [14].

The two kinds of *triangulation* and *projection* methods have been implemented in the stereo cameras software but this thesis mostly focus on the first one for it is easier to implement and give sufficient results with VERTIGO [27].

2.1.4 Optical Range Finder Model

A Time-Of-Flight camera (ToF), also called Optical Range Finder (ORF) in this document, is a class of LIDAR that can measure an entire 3D scene in real-time (more than 25 FPS) [11], [25]. As shown in figure 2-4, modulated light bursts illuminating the scene are produced by an IR emitter on the camera, the light is scattered by objects in the scene and a fast CMOS sensor synchronized with the emitter samples the received pulse and retrieves its phase. For each pixel, a distance camera-object can be compute as:

$$d = \frac{c}{2} . \frac{\Delta \phi}{2 \pi f} \tag{2.11}$$

Where:

- $\Delta\phi$ is the phase shift between the emitted and received light
- c is the light speed: $299792458 m/s$.
- f is the IR modulation frequency which has been set to $15 MHz$ in our experiments.

Which also means that the camera maximal range D is approximately equal to $10m$.

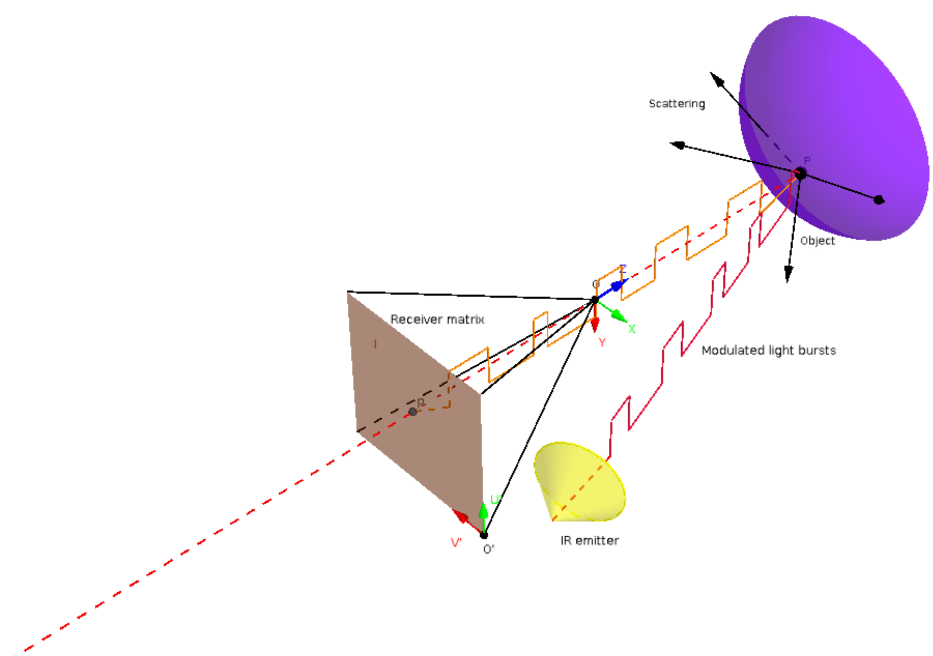


Figure 2-4: Principle of a time-of-flight camera. The pinhole model is still applicable but the measurement of the phase allows to create a new image: the *depth map* D_T.

In addition to the *depth map* D_T, two other images can be extracted from the sensor. In figure 2-5 provided by the Data Sheet of the camera, we are using [24], A is a measure of the modulated signal amplitude and helps to compute a *confidence map* C_T (the larger A, the better confidence on the measure); B is a measure of the mean signal amplitude and gives a *visual image* V_T very similar to the one we can have with traditional black and white cameras.

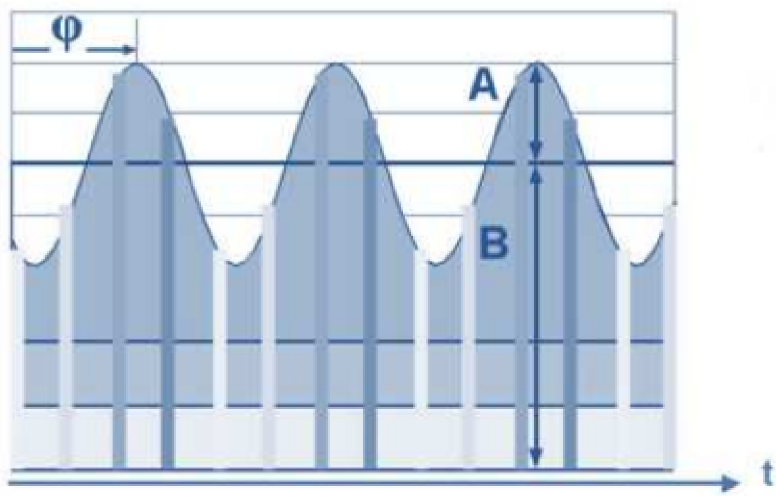


Figure 2-5: The received modulated IR signal is sampled and three images can be computed from the parameters A, B and $\Delta\phi$: *depth map* D_T, *confidence map* C_T and *visual image* V_T *- Mesa Imaging SR4k Data Sheet [24].*

2.2 Calibration Algorithm

2.2.1 Literature Overview

The calibration is a process that aims at finding the most accurate *instrinsic* and *extrinsic* matrices of the pinhole model described in sections 2.1.2 and 2.1.3 without any prior knowledge of the camera geometry. A precise camera calibration constitutes an important problem since it determines the accuracy of the results. Thus, numerous distinct methods have been developed to guarantee the best possible calibration. According to [37], those can be divided into three categories:

- **3D object-based calibration**: The known geometry of a 3D object as well as its projection are used to compute the best matrices that verify the pinhole equation (2.5). It however requires the presence of an accurate 3D calibration target for each calibration.
- **Self-calibration**: No object is used but the rigidity of a static environment induces enough constraints to estimate the calibration parameters. If this is very flexible, it is still not always reliable as a lot of initial parameters has to be estimated.
- **2D pattern-based calibration**: This third category is a compromise of flexibility (it requires only a 2D pattern like a checkerboard) and reliability (the solution always converges).

ORF calibration

Calibrating the ORF can be seen as a single camera calibration as it involves only the estimation of the *intrinsic matrix*. Different methods has been proposed in [22] or [21]. Due to its success, the Brown method presented in [37] and [15] will be considered. Apart from the *intrinsic* calibration, we shall however notice that [6] and [31] explain that we shall correct the systematic depth measurement error and this can be realized with a polynomial correction functional approach.

Stereoscopic cameras calibration

This category is relatively old, which is an advantage since we can find numerous efficient ways to realize it in the literature. For example, in the VERTIGO project, the Brown method [4] has been tested through the *OpenCV* libraries [3] leading to satisfying results [32].

ORF-stereo system calibration

This last category aims at finding *extrinsic matrices* between the ORF camera and the assembly of stereo cameras. Since the problem is quite recent, different algorithms have been proposed these last few years in [6], [38], [31]. In this paper, we will focus on the algorithm presented in [6] because the process does not only takes profit of the ORF *visual image*, whose space resolution is very low, but also of the *depth map* to increase the accuracy.

2.2.2 Optical Range Finder Calibration

As the ORF can be seen as a simple camera, the goal of the process is to find the matrix K in the pinhole equation:

$$s \begin{pmatrix} u_T \\ v_T \\ 1 \end{pmatrix} = K * \begin{pmatrix} x_T \\ y_T \\ z_T \\ 1 \end{pmatrix} \tag{2.12}$$

Let's take a set of world points P_T^i $(i = 1...m)$ and their projection in the image plane p_T^i. For each i, this equation ca be rewritten:

$$p_T^i = \begin{pmatrix} u \\ v \end{pmatrix}_T^i = \mathcal{K}(f_U, f_V, c_U, c_V, s_{UV}, P_T^i) \tag{2.13}$$

If we consider now the distortion model proposed in [4], we can match initial coordinates (u_{init}, v_{init}) in the *image plan* with *undistorted* coordinates (u_{corr}, v_{corr}) in the same plan through the equation:

$$p^i_{corr} = \begin{pmatrix} u \\ v \end{pmatrix}^i_{corr} = \mathcal{D}(u^i_{init}, v^i_{init}, k_1, k_2, k_3, k_4, k_5) \tag{2.14}$$

Where:

$$\begin{cases} u_{corr} = u * (1 + k_1 r + k_2 r^4 + k_3 r^6) + 2k_4 uv_+ k_5(r^2 + 2u^2) \\ v_{corr} = v * (1 + k_1 r + k_2 r^4 + k_3 r^6) + k_4(r^2 + 2v^2) + 2k_5 uv \\ r = \sqrt{u + v} \\ u = \dfrac{u_{init}}{z} \\ v = \dfrac{v_{init}}{z} \end{cases} \tag{2.15}$$

After simplifying notations, equations (2.13) and (2.14) give:

$$p^i_T = \begin{pmatrix} u \\ v \end{pmatrix}^i_T = \mathcal{F}(f_U, f_V, c_U, c_V, s_{UV}, k_1, k_2, k_3, k_4, k_5, P^i_T) \tag{2.16}$$

Or more simply:

$$p^i_T = \mathcal{F}(\theta^i_T) \tag{2.17}$$

If we define $\theta^i_T = (f_U, f_V, c_U, c_V, s_{UV}, k_1, k_2, k_3, k_4, k_5, P^i_T)$ as the unknown vector.

Practically, all those P^i_T correspond to the points on a plane pattern (the corner of a checkerboard). Therefore, to find the vector θ^i_T, we first detect the checkerboard corners projections p^i_T in the *visual image* V_T, then we create an initial guess $\hat{\theta}^i_T$ for each point and finally we solve the constrained iterative least square problem:

$$\theta_T^i = \texttt{arg}\ \ \mathsf{min}\left\{ \sum_{i=1}^{m} \|\mathcal{F}(\hat{\theta}_T^i) - p_T^i\|^2 \right\} \tag{2.18}$$

Where the problem's constraints are:

- *All points are in the same plane.*
- *The distance between two consecutive points is known.*

As the iterative algorithm used to solve the problem (2.18) has been detailed in [37] and already implemented in *OpenCV*, no further detail will be given but understanding how the problem is defined will be useful in the following sections.

2.2.3 Stereoscopic Cameras Calibration

In a stereoscopic calibration, the same thought process can be applied as in the previous paragraph except that *intrinsic* and *extrinsic* matrices for both cameras must be estimated in the same time. Indeed, we start from the equation system (2.8), which gives after rewriting:

$$\begin{pmatrix} p_L \\ p_R \end{pmatrix}^i = \mathcal{F}_{stereo}(\theta_{stereo}) \tag{2.19}$$

Where:

$$\theta_{stereo} = \begin{pmatrix} f_U^L, f_V^L, c_U^L, c_V^L, s_{UV}^L, k_1^L, k_2^L, k_3^L, k_4^L, k_5^L, P_L^i \\ f_U^R, f_V^R, c_U^R, c_V^R, s_{UV}^R, k_1^R, k_2^R, k_3^R, k_4^R, k_5^R, P_R^i \\ M_L R \end{pmatrix} \tag{2.20}$$

The minimization problem is then:

$$\theta_{stereo}^i = \texttt{arg}\ \ \mathsf{min}\left\{ \sum_{i=1}^{m} \left\| \mathcal{F}_{stereo}(\hat{\theta}_{stereo}^i) - \begin{pmatrix} p_L \\ p_R \end{pmatrix}^i \right\|^2 \right\} \tag{2.21}$$

The constraints are now:

- *All points are in the same plane.*
- *The distance between two consecutive points is known.*
- *Two epipolar lines cross in one single point (or the error is minimized in the case of a complex model).*

2.2.4 Multi-Sensors Calibration

In this third and last part of calibration, the estimation of the extrinsic matrix between the ORF and the stereoscopic cameras is performed. As explained in the literature review, this method proposed in [6] takes advantage of all the information given by the sensors and not only of *visual images*. Figure 2-6 shows the architecture of this method and theoretical details are given in the following paragraphs.

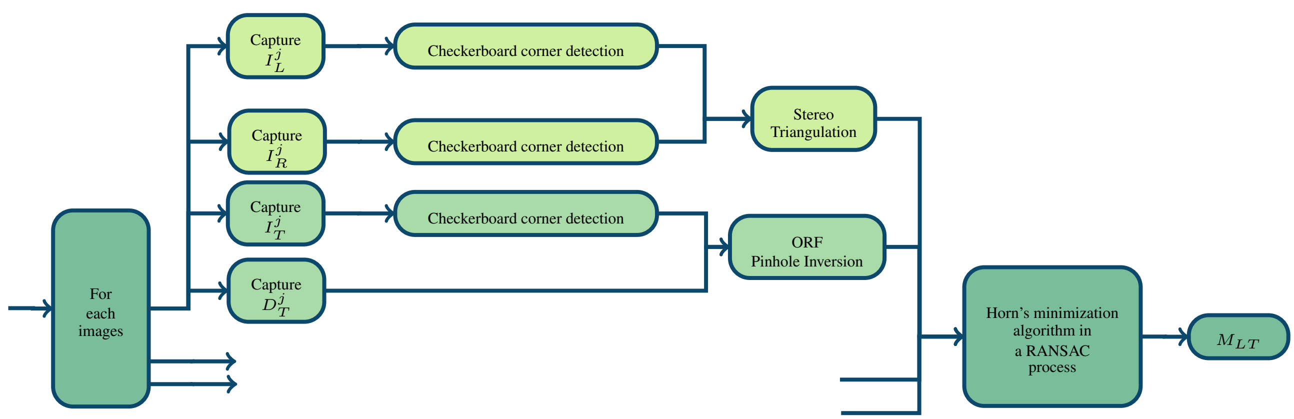


Figure 2-6: Architecture of the multi-sensors extrinsic calibration algorithm.

Checkerboard Corners Detection

Let us consider a system composed by two traditional cameras L and R and an ORF (or ToF) camera T (figure 2-7). To keep this method universal, no hypothesis is made on the geometry between the cameras. L and R produce respectively an image matrix I_L and I_R whose resolution is good and T produce a *depth matrix* D_T, a *visual image matrix* I_T and a *confidence matrix* C_T with a lower resolution.

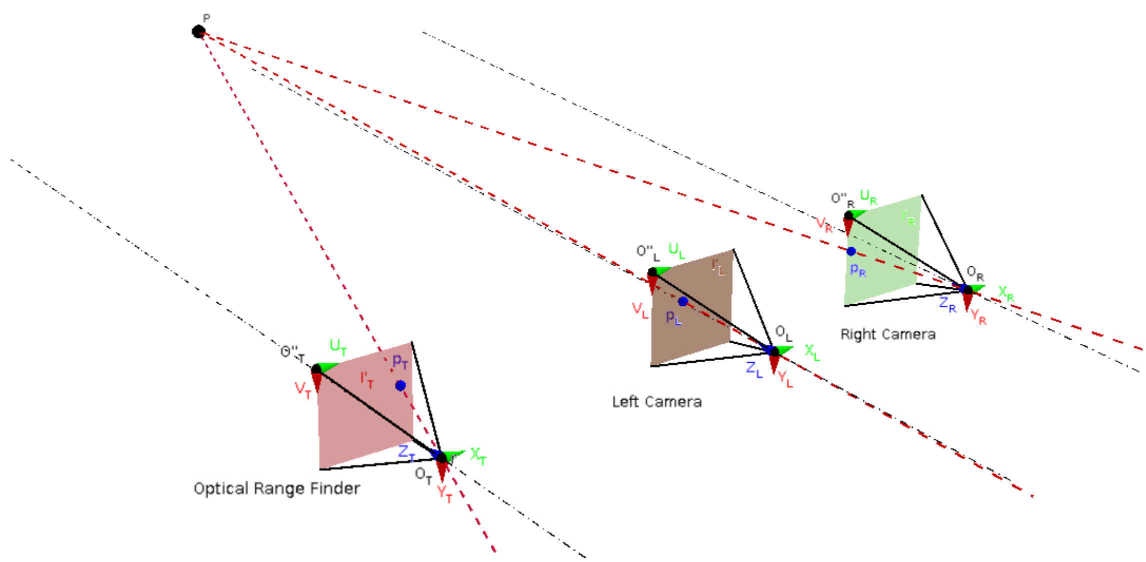


Figure 2-7: A point P of the real world is framed by three cameras: the ORF, left camera and right camera.

The prerequisite of this calibration are the intrinsic parameters of L, R and T and the extrinsic parameters between L and R computed in the methods introduced previously. The plan is to present a checkerboard in a region framed simultaneously by all the three cameras and capture images from all the sensors on time t_j where $j = 1..m$. In practice, we must also ensure that all images are reliable (no noise, no movement, good ranges, good confidence C_T,...) since the calibration process is sensible to the exactness of the measurements. By using *OpenCV* detection methods described in [29], we can obtain a set of i checkerboard corner $i = 1..n$ where n is equal to the number of images m times the number of corner per checkerboard k. Those points are represented by:

$$p_T^i = \begin{pmatrix} u \\ v \end{pmatrix}_T^i \qquad p_L^i = \begin{pmatrix} u \\ v \end{pmatrix}_R^i \qquad p_R^i = \begin{pmatrix} u \\ v \end{pmatrix}_R^i \tag{2.22}$$

Stereoscopic Triangulation

Once we have the set of points in the projection images of each camera, we use the stereoscopic cameras as well as their calibration matrices previously computed to triangulate them into the real world. In this project, given the relatively good alignment of VERTIGO cameras, a simplified *triangulation* is used but we could have used the full one:

$$\begin{cases} \begin{pmatrix} x_L \\ y_L \\ z_L \\ 1 \end{pmatrix}_L^i = \begin{pmatrix} f & 0 & c_U & 0 \\ 0 & f & c_V & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}_L^i \\ \\ \begin{pmatrix} x_L \\ y_L \\ z_L \\ 1 \end{pmatrix}_R^i = \begin{pmatrix} f & 0 & c_U & t_{RL} \\ 0 & f & c_V & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}_R^i \end{cases} \tag{2.23}$$

Which can be simplified in:

$$\begin{cases} z_L = \dfrac{t_{RL}}{u_L - u_R} \\ x_L = \dfrac{(u_L - c_U)z}{f} \\ y_L = \dfrac{(v_L - c_V)z}{f} \end{cases} \tag{2.24}$$

Optical Range Finder Pinhole Inversion

Similarly to the *triangulation* of the stereo assembly, we proceed to a *reconstruction* of points p_T^i in 3D. In contrast with the stereo *triangulation* method, a depth can be directly read in the D_T *depth matrix* for each of the points i. This lead to the simple euclidean geometry problem presented in figure 2-8 where we have:

- Three inputs: u_T^i, v_T^i, d_T^i
- Three outputs: x_T^i, y_T^i, z_T^i

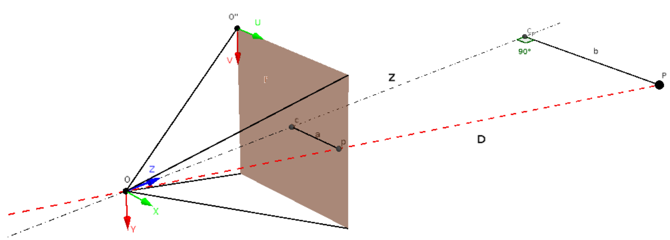


Figure 2-8: To compute the (x_T, y_T, z_T) coordinates of the point P given (u_T, v_T) and d_T, we use the Pythagorean and Thales theorems.

In figure 2-8, for each point p_T^i:

$$x_T = \frac{(u_T - c_U)}{z_T} f \quad \text{(Pinhole model)}$$

$$y_T = \frac{(v_T - c_V)}{z_T} f \quad \text{(Pinhole model)}$$

$$\frac{f}{z_T} = \frac{a}{b} \quad \text{(Thales)}$$

$$d^2 = z_T^2 + b^2 \quad \text{(Pythagoras)}$$

Hence, after simplifications:

$$\begin{cases} z_T = \sqrt{\dfrac{d_T^2}{1 + (u_T - c_U)^2 + \dfrac{(v_T - cV)^2}{f^2}}} \\ x_T = \dfrac{(u_T - c_U)}{z_T} f \\ y_T = \dfrac{(v_T - c_V)}{z_T} f \end{cases} \tag{2.25}$$

Pose Estimation

Now that the *triangulation* from the stereo cameras and the *reconstruction* from the ORF have been performed, we have a set of i 3D points in two different coordinates systems T and L and we want to find the matrix M_{LT} which minimizes the global errors between all those couple of points. The optimization problem can therefore be defined as an iterative:

$$\hat{M}_{LT} = \texttt{arg min} \left\{ \sum_{i=1}^{n} \|P_T^i - M_{LT} * P_L^i\|^2 \right\} \tag{2.26}$$

This problem will be solve using Horn's absolute orientation algorithm [16] into a Random Sample Consensus process (RANSAC) [7]. Without going into further details, the idea behind Horn's algorithm is the following:

- Compute the centroids for L and T points, find the translation between those centroids.
- Find the scale between the two points clouds.
- Rewrite the problem using quaternion so that the minimization can be reduced to finding eigenvalues of a matrix.

The RANSAC process aims at rejecting false positive measurements from the estimation scheme. For instance, if the checkerboard corner detection fails in one image and detects wrong points, we do not want those false positives to influence the total estimation. Intuitively, we know they can be spotted very quickly because they produce results far from those computed with all other points. We present the algorithm as summarized in [27]:

Extrinsic Parameters Computation

In the end, from the transformation matrix M_{LT} and the stereo *extrinsic matrix* M_{LR}, all *extrinsic parameters* can be computed:

$$M_{TL} = M_{LT}^{-1} \qquad M_{RL} = M_{LR}^{-1} \qquad M_{RT} = M_{RL} * M_{LT} \qquad M_{TR} = M_{RT}^{-1} \tag{2.27}$$

Algorithm 1 RANSAC: Random sample consensus.

```
function RANSAC(data, maxiterations, mininliers)
    iteration ⟵ 0;
    bestmodelpoints ⟵ emptyset;
    while iterations < maxiterations OR bestmodelpoints < mininliers do
        randompoints ⟵ select minimum number of points randomly;
        hypothesismodel ⟵ estimate modelparameters using randompoints;
        hypothesispoints ⟵ randompoints;
        for all points except randompoints do
            if point fits hypothesismodel with small error then
                Add point to hypothesispoints;
            end if
        end for
        if hypothesispoints has more points than bestmodelpoints then
            bestmodelpoints ⟵ hypothesispoints;
        end if
        iterations ⟵ iterations + 1;
    end while
    bestmodel ⟵ estimate model parameters using bestmodelpoints;
    Return bestmodel;
end function
```

2.3 Fusion Algorithm

2.3.1 Literature Overview

The goal of the fusion algorithm is to compute 3D frames of the environment, taking advantage of each sensor features in order to improve the global accuracy and completeness. Many articles have studied the realization of stereo and ToF cameras fusion algorithm using various alternate methods. In order to understand more clearly, we can divide them into two categories:

- In the first category, the stereo algorithm is performed alone without including the ToF informations. As an output of this algorithm, we obtain a 3D map which can be compared to the ToF 3D map to give a refined solution. This analyze is proposed in [19] where a final 3D map is deduced from the ToF 3D map and the stereo 3D map by *Winner takes it all* or *Simulated annealing* strategies. According to their results, it gives real-time performances with a computation time lower than one second which is a true asset. In [1], 3D maps are fused along many patchlets areas using a *Gauss-Markov Model*.
- The second category implies incorporating the 3D information of the ToF sensor directly in the stereo algorithm. Those methods seem to give good results but no information about the processing time is given. In [6], a full probability model is computed and an estimation maximizing the joint probability is then selected. The same kind of approach is used in [38]. In [12], the ToF data are exploited in the initialization phase of a *Dynamic Programming (DP)* algorithm developed in [33]. In another field of application, [18] creates a joint probability of the two sensors to fulfill an occupancy grid and reconstruct a 3D object from multiple views.

In this section, we constructed a new algorithm based essentially on the work in [6]. Indeed, in the beginning of this internship, it seemed an elegant way to improve each pixel accuracy (and only accuracy) by taking advantage of both sensors. However, the characteristics of the setup we are using are quite different and chapter 4 will discuss in detail what that involves.

2.3.2 Background

Sensors

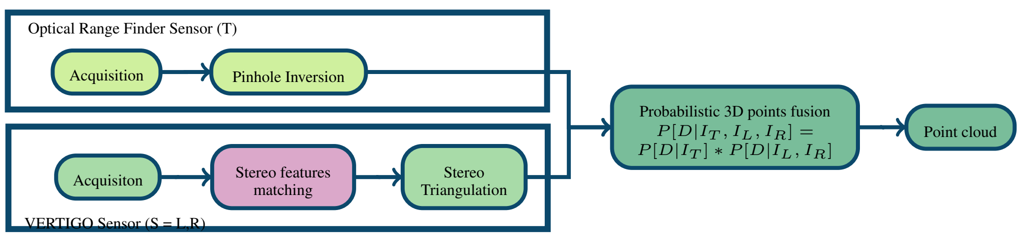


Figure 2-9: The physical sensors of the INSPECT project are represented with their software in a *common representation format* which enables a probabilistic fusion [26]. N.B.: This is not exactly the way the fusion is done here (see figure 2-14).

If we adopt the representation and the notation in [26], a *sensor* E is a system (hardware and software) characterized by its state, function, performance, output and energy type. In this case, we then consider two external sensors dedicated to navigation through 3D environment reconstruction whose output is a 3D point cloud combined with a monochromatic image (figure 2-9). The performances are summarized in table 2.1.

Performance	ORF	VERTIGO Sensors
Accuracy	Omnipresent noise	Good (if results)
Repeatability	Very good	Very good
Linearity	Non-linear with object speed and distance	Non-linear with object textures
Sensitivity	Sensible to luminous noise, saturation and movement	Sensible to the lack of texture in the environment
Resolution	Medium spatial resolution and good temporal resolution	Good spatial resolution and bad temporal resolution (computation)
Reliability	A result is guaranteed (but with variable quality)	Result not always guaranteed
Range	Very limited	Good

Table 2.1: Performances of the two sensors according to the notation in [26].

For each sensor, we can also define a *sensor observation* O:

$$O = \langle E, \mathbf{x}, t, \mathbf{y}, \boldsymbol{\Delta}\mathbf{y} \rangle \tag{2.28}$$

Where:

- E: the entity name which include the name of the physical property measured by the sensors as well as the units. Here, we measure *3D point clouds* with their luminous intensity. In this work, we do not try to measure a spectral intensity as we will do next when the thermographic camera will be added, we just need to know if the point exists or not (boolean measurement).

- $\mathbf{x}$: the spatial location of the measurement (the 3D coordinates of each point in an absolute coordinate system).
- $\mathbf{y}$: the value of the measurement (exists or not).
- $\mathbf{\Delta y}$: a generic term which includes many types of errors including measurement, calibration, location,...

Fusion

To perform fusion between them, the two *sensors observations* must be represented in the same *common representational format* which means we shall perform:

- *Spatial Alignment*: This process also called *registration* consists of matching both sensors point clouds together. This task is one of the most important in the fusion and involves *triangulation* and *re-projection* using the calibration matrices as discussed in sections 2.1.3 and 2.1.4. This fusion algorithm is a bit special in a way we follow the process:

  2D points in T reference system ⇒ Reconstruction ⇒ 3D points in T ⇒ 3D points in L ⇒ Projection ⇒ 2D points in L and R

- *Temporal Alignment*: It is the transformation of the local times t to a common time axis. In this work, we can estimate the image acquisition to be synchronous between the two devices (a system of time stamps has been implemented to guarantee this approximation) and we will not focus on this point.
- *Sensor Value Normalization*: As explained in [26], normalizing two sensors with different physical properties is generally done by converting them into *a posteriori probabilities*. In that case, estimated outputs are given by the application of

the Bayes theorem. Here we can assimilate the computation of a 3D cloud to the research of a depth distribution D which will be estimated by the system as a distribution $\hat{D}$ maximizing the probability:

$$\hat{D} = \mathtt{arg}\ \ \mathsf{max}\, P[D|I_T, I_L, I_R] = \mathtt{arg}\ \ \mathsf{max}\, \frac{P[I_T, I_L, I_R|D]P[D]}{P[I_T, I_L, I_R]} \tag{2.29}$$

Since $P[I_T, I_L, I_R]$ is not D-dependent, it can be replaced by any constant without changing the problem. Within those constants, let choose $P[I_T]P[I_L, I_R]$:

$$\hat{D} = \mathtt{arg}\ \ \mathsf{max}\, \frac{P[I_T, I_L, I_R|D]P[D]}{P[I_T]P[I_L, I_R]} \tag{2.30}$$

As the camera depth distribution can be considered uniform, we can also write:

$$\hat{D} = \mathtt{arg}\ \ \mathsf{max}\, \frac{P[I_T, I_L, I_R|D]P[D]P[D]}{P[I_T]P[I_L, I_R]} \tag{2.31}$$

One major hypothesis in the fusion is that $\{I_T, D\}$ and $\{I_L, I_R, D\}$ are independent [6]. It leads to:

$$\hat{D} = \mathtt{arg}\ \ \mathsf{max}\, \frac{P[I_T|D]P[D]}{P[I_T]} * \frac{P[I_L, I_R|D]P[D]}{P[I_L, I_R]} \tag{2.32}$$

$$\hat{D} = \mathtt{arg}\ \ \mathsf{max}\, P[D|I_T] * P[D|I_L, I_R] \tag{2.33}$$

Finally, a fusion algorithm is characterized in terms of representation, certainty, accuracy, completeness. This will be discussed again with the introduction of the results in chapter 4 but we can already say:

- **Representation**: This feature characterizing the abstraction level increasse from the algorithm inputs (2D matrices) to the algorithm outputs (3D point clouds).
- **Certainty**: The gain in measurement certainty illustrates the growth between $P[D|I_T]$ and $P[D|I_L, I_R]$ on one hand and $P[D|I_T, I_L, I_R]$ on the other hand. It will be discussed in chapter 4.
- **Accuracy**: The gain in measurement accuracy before and after the fusion will be discussed in chapter 4.
- **Completeness**: The completeness will be affected by the limited FoV of each cameras and the geometry between them. This feature is environment-dependent and will depend on the target size and distance and the application we want to perform.

2.3.3 Architecture

All the concepts above leads to the architecture in figure 2-14 that shows the big scheme of the fusion algorithm implemented in this thesis. However, there is a big difference compared to figure 2-9. Indeed, we want to avoid *feature matching* between stereo images which leads to bad results in poor textured environment. A single stream process is then use and each major step is presented in the following paragraphs.

Determination of the Points Framed by all Cameras

In order to perform *registration*, the determination of the domain framed by both devices is the first step of the algorithm. We then only keep the points framed by the two cameras before proceeding to the fusion. Figure 2-10 shows the three devices FoV in the XZ plane.

The hatched zone corresponds to the domain where fusion will take place. As we consider the three cameras have the same Y coordinate, the Y domain is given by the minimal FoV. Basic analytical calculations give:

$$z_i \in \left[min\left(\frac{t_{TR}\tan(\rho)\tan(\tau)}{\tan(\rho)+\tan(\tau)}, range_{min}\right); range_{max}\right] \tag{2.34}$$

$$x_i \in \left[\frac{-z_i - t_{TR}\tan(\rho)}{\tan(\rho)}; \frac{z_i}{\tan(\tau)}\right] \tag{2.35}$$

$$y_i \in \left[min\left(\frac{-z_i}{\tan(\rho)}, \frac{-z_i}{\tan(\tau)}\right); min\left(\frac{z_i}{\tan(\rho)}, \frac{z_i}{\tan(\tau)}\right)\right] \tag{2.36}$$

Where $range_{min}$ and $range_{max}$ represent the limited Z range of the ORF (see section 1.2.1).

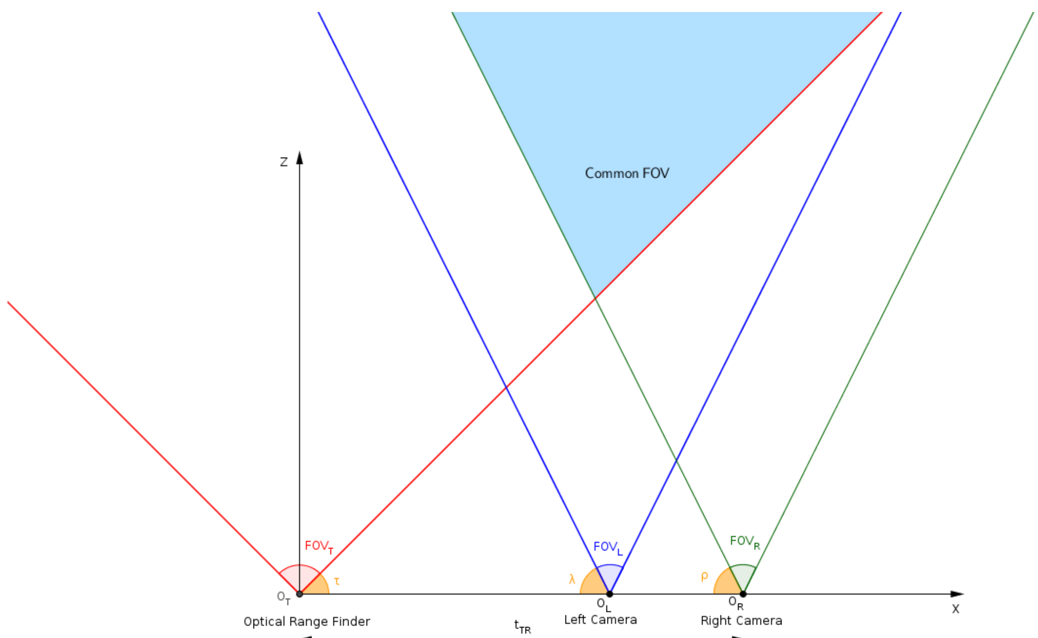


Figure 2-10: The region framed by the three cameras depends on the *baseline* between the cameras as well as their FoV.

Optical Range Finder Noise Model

According to [17] and [6], the error in the depth measurement $\hat{D}_T$ for each pixel captured by the ORF camera can be mainly described as the sum of the following components:

- A *thermal noise* with a Gaussian distribution.
- A *quantization error*.
- A *photon shot noise* with a Poisson distribution.
- A *scattering generated noise* especially in presence of discontinuities.

According to the same sources, they consider the *quantization error* and the *photon shot noise* can be neglected in front of the *scattering* and *thermal noises*. The latter can be represented by a Normal distribution around the measured depth:

$$\mathcal{N}(d_T^i, \sigma_t^2) \tag{2.37}$$

Where σ_t is inversely proportional to the confidence of the pixel given in the pixel map C_T. In our experiment setup, we selected a factor $\sigma_t^2 = (range_{max} - C_T)/10$. Concerning the scattering noise, [5] shows that it could also be expressed with a normal distribution:

$$\mathcal{N}(d_T^i, \sigma_s^2) \tag{2.38}$$

Where σ_s^2 is the variance of the measured depths in the second order neighborhood of the considered pixel p_T. Finally, the thermal noise is neglected near discontinuities and scattering noise far from discontinuities which gives:

$$P[d^i|I_T] \sim \mathcal{N}(d_T^i, \sigma_w^2) \tag{2.39}$$

Where:

$$\sigma_w = \mathsf{max}(\sigma_t, \sigma_s) \tag{2.40}$$

Probabilistic Space Discretization

Once the noise has been determined for each pixel, we can assume the real depth is situated at maximum $3\sigma_w$ from the measured depth:

$$d^i \in [d_T^i - 3\sigma_w;\ d_T^i + 3\sigma_w] \tag{2.41}$$

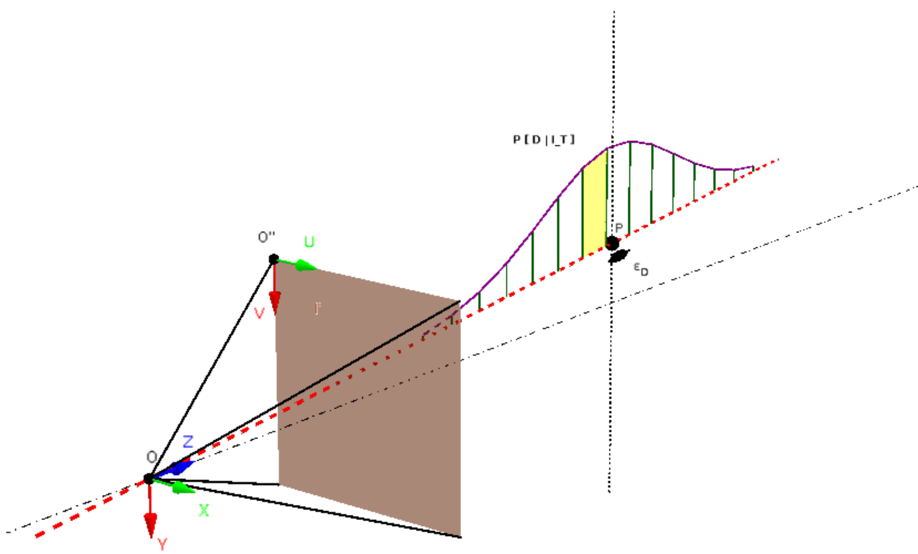


Figure 2-11: The interval around the measured depth d_T^i is divided in j steps of length ϵ_D.

As presented in figure 2-11, this interval is then discretized around $\hat{d}_T^i$ with a step equal to the stereoscopic accuracy sub-sampled by a factor k_{int} ($k_{int} = 4$ in this work). In [10] and [20], this resolution is equal to:

$$\epsilon_d = \frac{d^2}{b\,f}\,\epsilon_p \tag{2.42}$$

Where d is the depth, b is the *baseline* between L and R cameras, f is the focal length (assumed to be the same for L and R) and ϵ_p is the distance between two pixels on the camera as shown in figure 2-12.

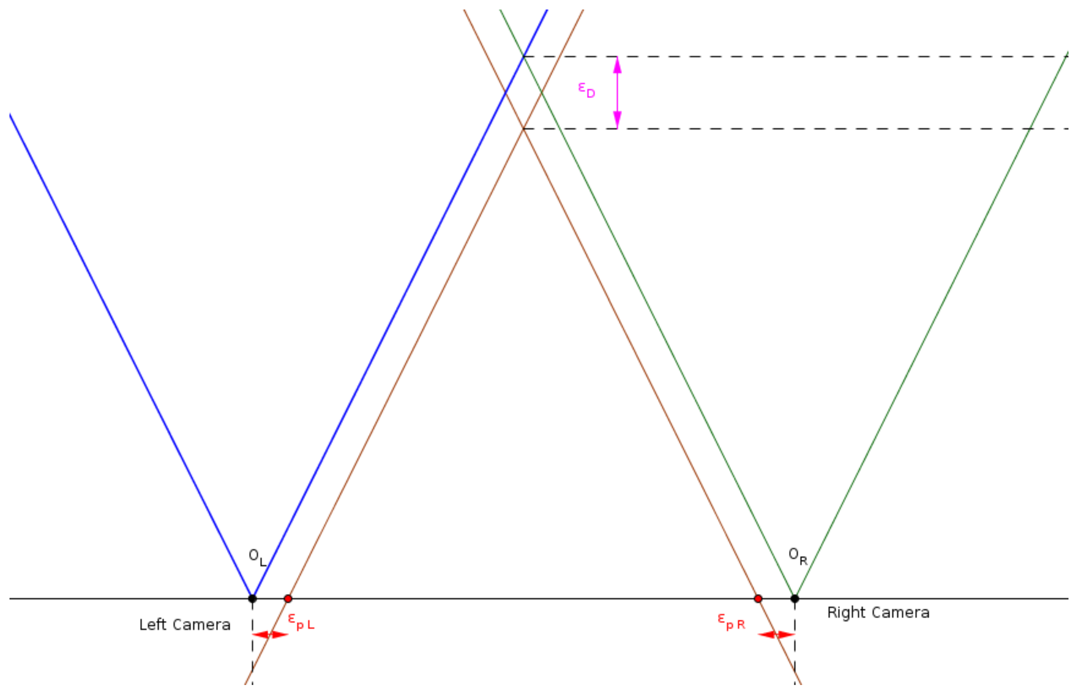


Figure 2-12: The theoretical depth error for stereo devices ϵ_d depends on the depth d, the baseline between cameras b, the pixel size ϵ_p and the focal length f.

Optical Range Finder Probabilistic Model

For each pixel of the depth map d_T^i ($i = 1..n$), we have constructed a sample of j points $d_T^{i,j}$ ($j = 1..m$) which are the candidates to represent the final estimation $\hat{d}^i$. As explained in section 2.3.2, the next step to perform fusion consists of determining $P[d^i = d^{i,j}|I_T]$ and $P[d^i = d^{i,j}|I_L, I_R]$. As shown previously, we can directly write:

$$P[d^i = d^{i,j}|I_T] = \frac{1}{\sigma_w^i \sqrt{2\pi}} e^{-\frac{(d^{i,j}-d^i)^2}{2\sigma_w^{i\,2}}} \quad (2.43)$$

Stereoscopic Cameras Probabilistic Model

The computation of the probability $P[d^i = d^{i,j}|I_L, I_R]$ involves the reconstruction of the 3D coordinates obtained for all $d^{i,j}$ from the ORF (figure 2-13). It is a way to avoid straightforward feature matching whose results depend on the texture level of the scene as explained previously.

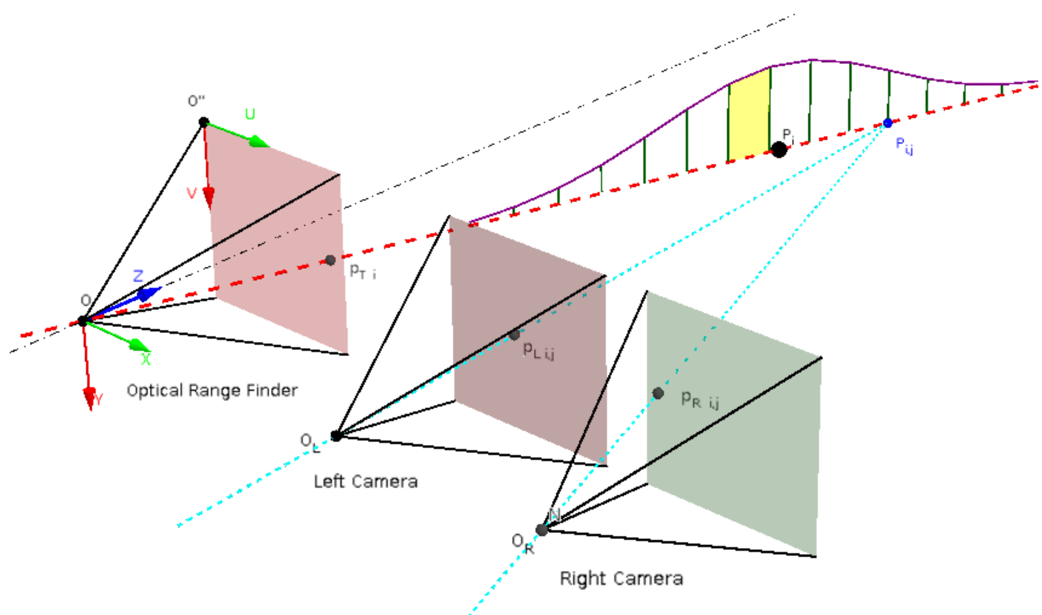


Figure 2-13: For each pixel, each $P_T^{i,j}$ is reprojected in L and R image planes. From those coordinates, a probability will be established to correct the depth measured by the ORF.

After this reconstruction, the stereo probability is computed as an exponential distribution which decreases according to a factor $c_{i,j}$ representing the degree of likeness between a point $P^{i,j}$ projected in L and R:

$$P[d^i = d^{i,j} | I_L, I_R] = e^{\frac{-c_{i,j}}{\sigma_w^i}} \tag{2.44}$$

In this way, when we project all the j points discretized from the ORF Normal probability of a pixel i, we can compute a degree likeness for each pair in L and R. As a pixel is very tiny, a small offset could incur a big mismatch between L and R projection, that is why we consider a rectangular window of size (k, l) around this point and we compute $c_{i,j}$ as a truncated sum of the absolute difference (*L1-norm*) between each point of the window [9]:

$$c_{i,j} = \mathsf{min}\Big\{ \sum_k \sum_l \big| I_L(u_{i,j} + k, v_{i,j} + l) - I_R(u_{i,j} + k, v_{i,j} + l) \big| \,, \, thresh \Big\} \tag{2.45}$$

Where the value $thresh$ will be discussed in the results.

Joint Probability Computation

For each pixel i, we deduce the joint probability:

$$P[d^i = d^{i,j} | I_T, I_L, I_R] = P[d^i = d^{i,j} | I_T] \; * \; P[d^i = d^{i,j} | I_L, I_R] \tag{2.46}$$

Depth Estimation Selection

To create an improved 3D map, the last step consists of selecting the maximal probability in the selected interval for each pixel i:

$$\hat{d}_i = \mathsf{max}_j \Big\{ P[d^i = d^{i,j} | I_T, I_L, I_R] \Big\} \tag{2.47}$$

3D Coordinates Computation

From the variables $(u_T^i, v_T^i, \hat{d}_i)$, we recover $\hat{P}^i = (\hat{x}^i, \hat{y}^i, \hat{z}^i)$ by inverting the pinhole equation as described in 2.2.4.

Point Cloud Creation

As a last step, a point cloud $\hat{\mathbf{P}}$ is composed by the assembly of every pixel $\hat{P}^i$

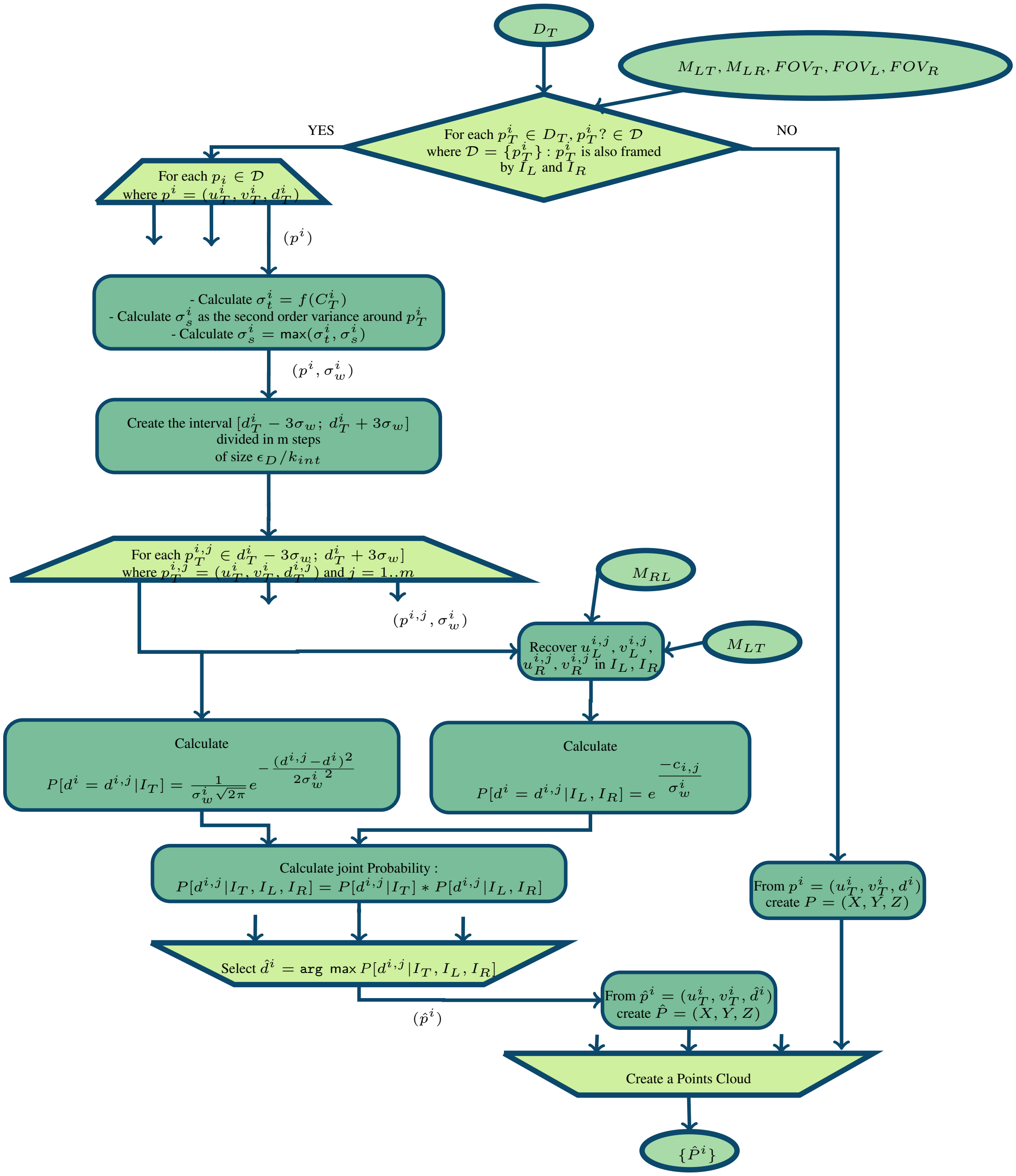


Figure 2-14: The global architecture of the fusion algorithm. Fusion is not performed straightforwardly to avoid feature matching in poor-textured environment.

Chapter 3

Implementation

3.1 Language and Libraries

The algorithms detailed in the previous chapter have been implemented to be tested with the sensors presented in the first chapter. The choice of *C++* as a programming language came naturally for the following reasons:

- This language is used inside the VERTIGO, INSPECT and SPHERES cores.
- *C++* is the most common language on Linux and Ubuntu 12.04 is the operating system of the VERTIGO computer.
- *C++* benefits from multiple libraries.
- The cameras constructors provided *C++* API on Linux for their hardware.

To simplify the implementation and avoid useless developments, we also used third-part libraries:

- *MESA-Imaging API* for the ToF camera.
- *IDS-Imaging uEye API* for the stereo cameras.

- *PLEORA eBUS SDK* for the thermographic camera.
- *OpenCV* for image processing, matrix operations and result display.
- *Point Cloud Library ()PCL)* for 3D points cloud manipulation and display.
- *Boost* for matrix operations and numerical solvers.
- *Eigen* for matrix operations and numerical solvers.

3.2 Software Architecture

Beside a few tests on Matlab and with *OpenCV* as well as the camera acquisition programming in the INSPECT core, this project has focused on a software implementing live and post-processing of images captured by the ToF and the stereo devices. The sources and tutorials of this software are available in the repository:

`https://github.com/Gabs48/Inspect_sensor_fusion`

In its architecture presented in figure 3-1, the classes *Camera*, *ORF* and *Thermocam* directly implement the acquisition using the constructors API. Alternatively, they can be initialized to read files in a directory to perform post-processing instead. *Halo* is a class representing the Halo hardware and implements the *Camera*, the *ORF* and the *Thermocam* devices. Each of those four classes inherits from *Calibration* which allows to compute calibration parameters when they are not available yet in calibration files. *StereoTriangulation* and *OrfTriangulation* implement the projection and triangulation process for the two sensors. Finally, *HaloTriangulation* contains the fusion algorithm itself since it reconstructs 3D points from the sensors data. *Utils* provides a few tools like the *TimeStamp* class to compare acquisition time of sensors between each others.

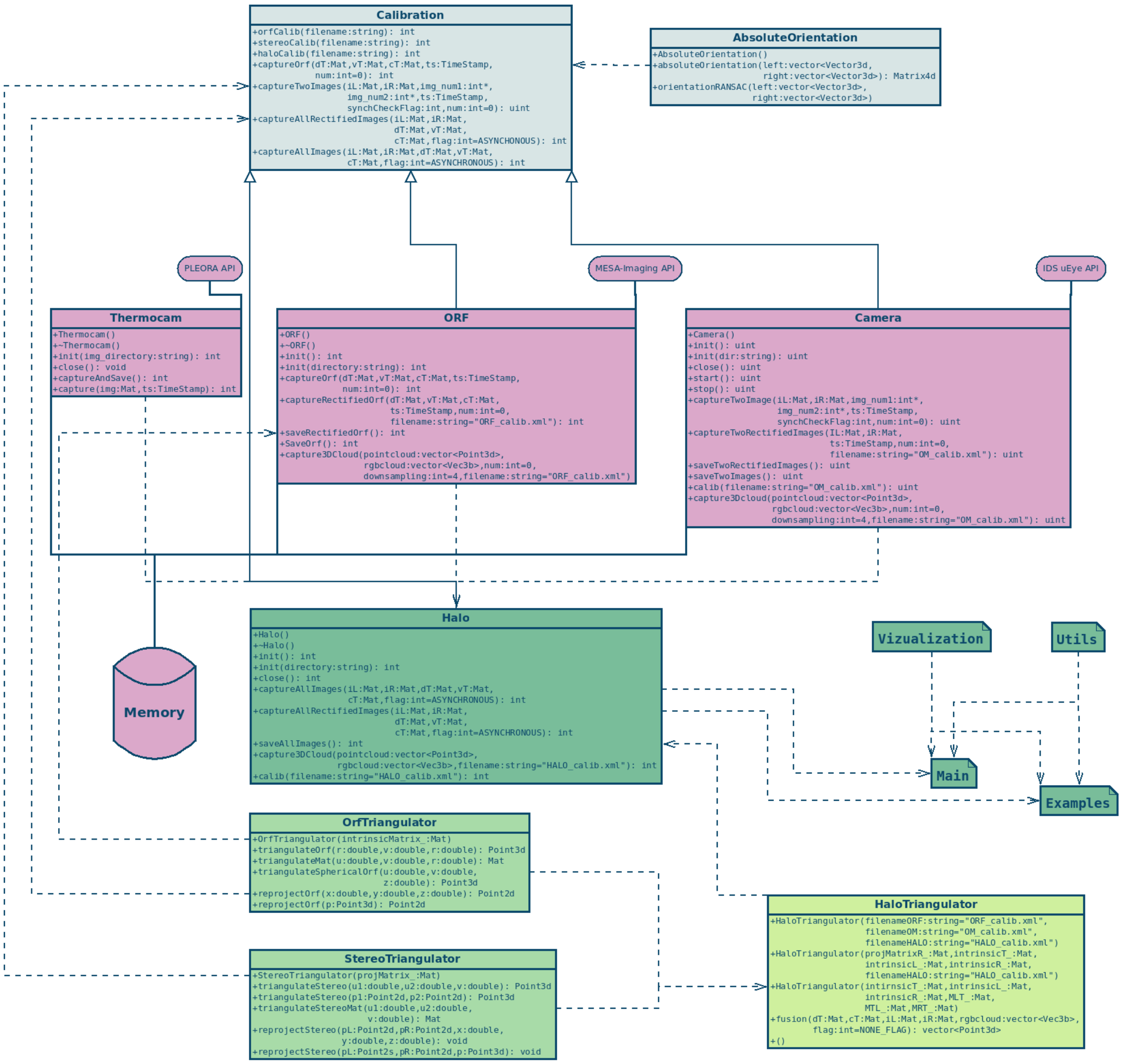


Figure 3-1: The UML diagram of the Inspect Sensor Fusion software available in the public repository `https://github.com/Gabs48/Inspect_sensor_fusion`. Pink correspond to acquisition classes; Light blue to calibration classes; Very light green to fusion class; Green a bit darker to projection and triangulation classes; Remaining classes are in dark green.

Chapter 4

Results

In this chapter, the results obtained in the laboratory and during an RGA test sessions are presented and analyzed in order to make conclusions about the performance of the algorithms elaborated in this project. The two first sections put forth the sensors performance and give a qualitative appreciation for each of them motivating the use of a fusion algorithm. The third section takes an interest in the full system calibration and, through images and measurements, highlights the accuracy of the algorithm. Finally, in the fourth section, we show the current results from the multi-sensor fusion algorithm obtained during ground test sessions and also briefly RGA sessions. We then try to explain why they may be not as good as expected theoretically.

4.1 ORF Acquisition

The *ORF C++ API* as proposed by the constructor MESA-Imaging allows the user to get three different images in a single capture:

- A *depth image* D_T coded in 10 bits from which a distance in meters can be computed for each pixel.

- A *visual image* of the environment V_T coded in 16 bits.
- A *confidence* image acting as a indicator of confidence in the depth measurement coded in 16 bits.

Thanks to the pinhole inversion method presented in chapter 2, a 3D point cloud containing visual information can be created. In this section, we will present the image acquisition, the ORF calibration, the depth measurement accuracy and the 3D points cloud reconstruction before to conclude on the ORF imperfections.

4.1.1 Acquisition

Figure 4-1 shows three typical images captured in a stationary environment. We can see that further the distance, the lighter the depth picture. We can also notice the increasing noise farther from the center of the camera, the lower confidence on the objects edges (due to discontinuities) and the poor quality of the visual image (partly due to image equalization to adapt to changing lighting conditions).

Figure 4-1: *Left*: depth image - the darker, the nearer. *Center*: visual image. *Right*: confidence image - the variance is high on objects rims (discontinuities), on the picture edges (sensor limitations) and on some surfaces (thermal noise).

The limited range is highlighted in figure 4-2 where an object is put a few centimeters away from the camera. The confidence becomes extremely bad (black) as the measured distance is clearly far from reality.

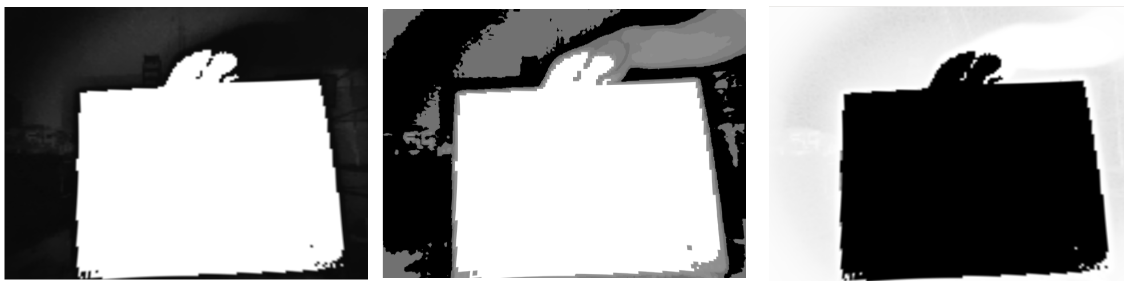


Figure 4-2: *Left*: depth. *Center*: visual. *Right*: confidence. When the object exceed the range limitation of the camera, the measurement can be twisted.

In figure 4-3, the same object is held a little farther so it can be measured correctly. However, the low luminosity induces the automatic exposure regulation to wait a little longer during the capture in order to make a good measurement, leading to very noisy pictures where the environment is brighter.

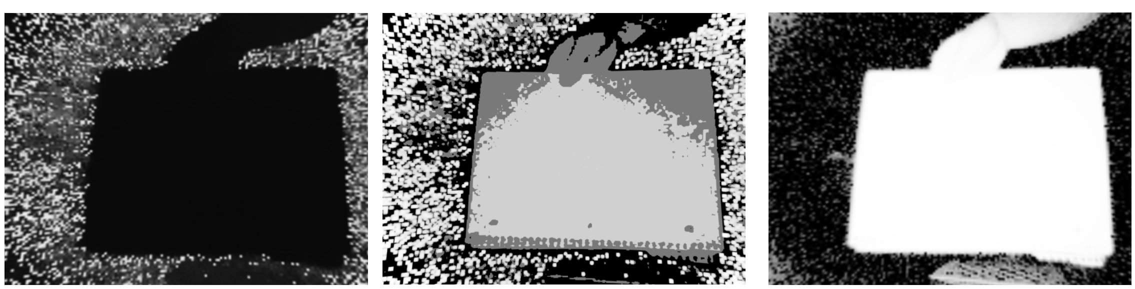


Figure 4-3: If the object is too close (even if respecting the range limitation), auto-exposure can lead to very high noise in the background.

Figure 4-4 illustrates that the quality of the measurement is material-dependent. Here, the reflection of an aluminum slab corrupts the depth and visual pictures when the computation of confidence does not highlight the problem everywhere on the slab which means that the measurement of the quality cannot rely on the confidence picture only.

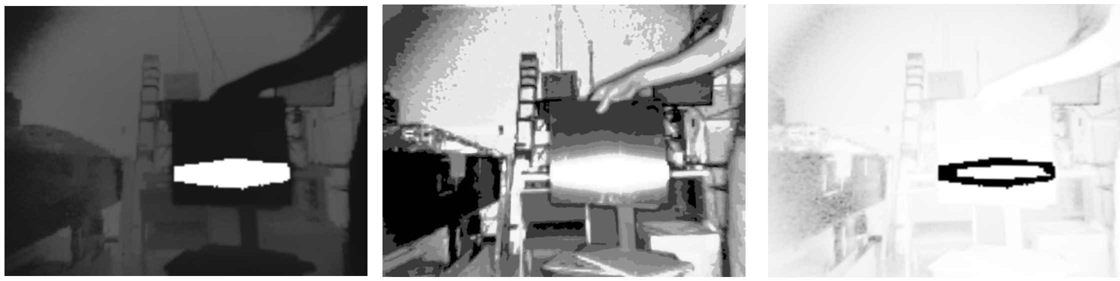


Figure 4-4: The nature of the material (here some aluminum) changes the reflection properties, hence the measurement quality.

Let us also mention the importance of the ORF IR emission time. In figure 4-5, we compare *confidence images* taken in the same conditions but where the IR emits either very shortly during the capture, either constantly.

Figure 4-5: *Left*: short IR emissions during capture leads to bad confidence image. *Right*: leaving the IR emitter constantly ON produce a better confidence.

Last but not least, the problem of relative motion between the camera and the environment is presented in figure 4-6. On those pictures, a checkerboard is moved with a speed barely higher than a few centimeters per second and leads to very bad results. This can be a very important issue in space where the studied system is supposed to track and analyze the properties of spinning and tumbling targets.

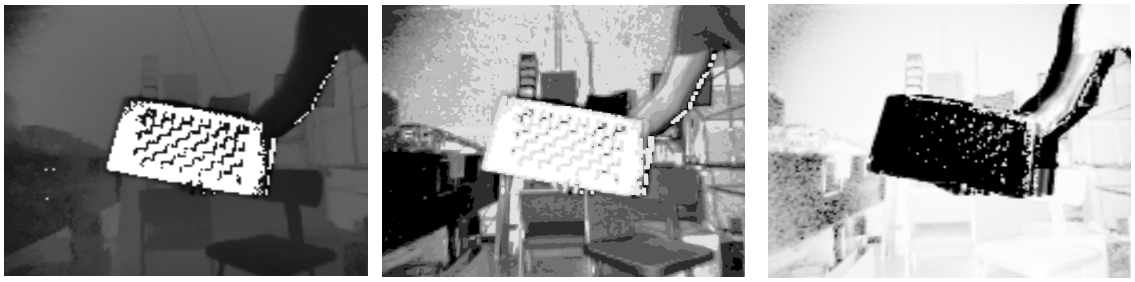


Figure 4-6: The movement is a decisive factor driving the measurement accuracy. A relative speed of a few dozens of centimeters per second causes a substantial error in the checkerboard depth measurement.

4.1.2 Calibration and distortions Rectification

In figure 4-7, many typical pictures are taken in the limits of the space where checkerboards can be detected using only the *visual image*.

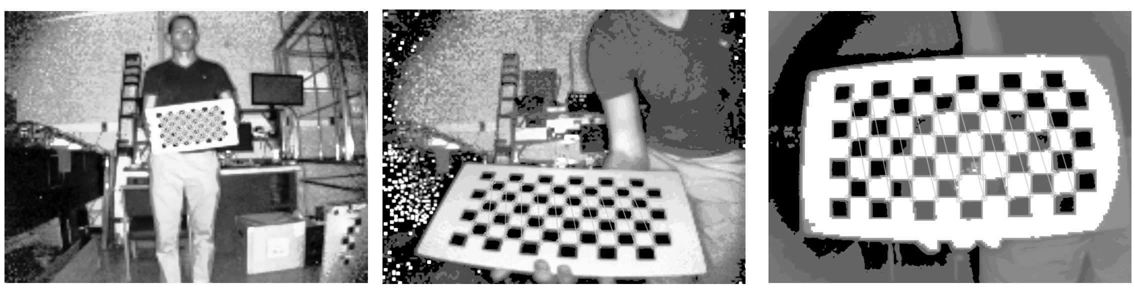


Figure 4-7: We vary the checkerboard orientation to the limits of detection to cover the space at maximum. *Left*: the farthest distance. *Center*: the maximal inclination. *Right*: the closest distance.

From the coordinates of the points, the software gives the following *intrinsic matrix*:

$$\begin{pmatrix} 532.74 & 0 & 308.64 \\ 0 & 490.43 & 222.22 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.1}$$

Where the distances are in pixels. Given a pixel size of $40\mu m$ (see chapter 1) and the picture scaling ratio of 3.68 along U and 3.33 along V, we deduce $f_U = 5.9mm$ and $f_V = 5.9mm$ where the small differences with the values as provided by the constructor can be explained by a variable focal length. Besides, $c_U = 308$ and $c_V = 222$ are not far from the theoretical center $c_U = 320$ and $c_V = 240$. The distortion coefficients are equals to:

$$\begin{pmatrix} -3.45e-01 & 1.44e-01 & -2.20e-04 & 1.91e-03 & 5.181e-02 \end{pmatrix} \tag{4.2}$$

Figure 4-8 shows the result after the distortion rectification for a very close object.

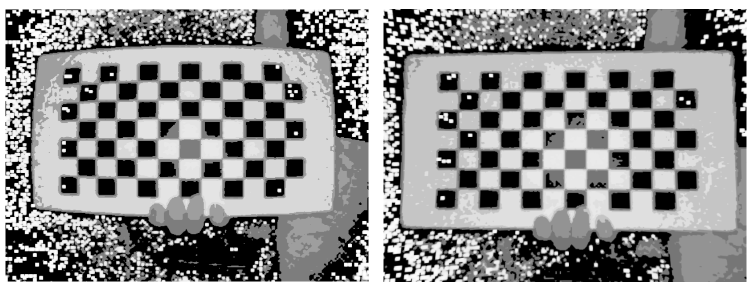


Figure 4-8: *Left*: a checkerboard before rectification. *Right*: the same checkerboard after the distortions rectification. The edges are now straight and parallel.

4.1.3 Absolute Depth Measurement

In this paragraph, one of the examples implemented with this project software is used to compute the depth of a calibration target situated $1m$ away from the ORF camera (figure 4-9).

To take the noise into account, the capture of figure 4-10 is realized several times to com-

Figure 4-9: The setup used for ORF calibration.

pensate the thermal noise of $1.026mm$. This difference of a few millimeters is explained by the inaccuracy of the setup itself, the material reflectivity, and the fact that the measured point is not directly along the focal axis (we then measure the hypotenuse as presented in figure 2-8).

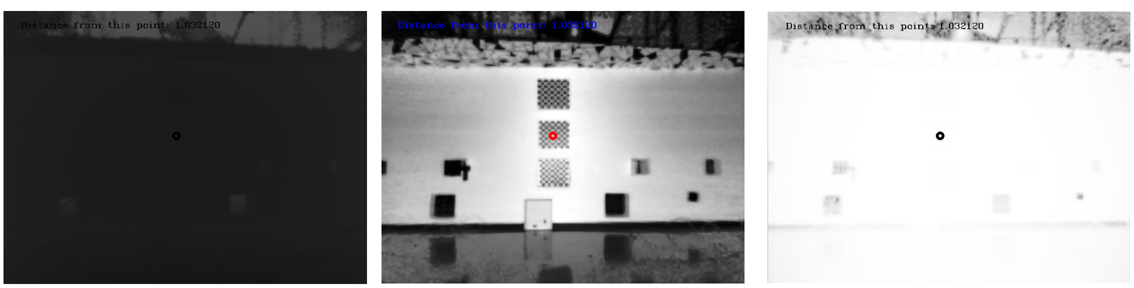


Figure 4-10: Several measurement are effectuated in the center of the target and their mean is computed to compensate Gaussian noise.

This last effect is showed in figure 4-11, where the measured depth is equal to $1.159m$ when the Z coordinate should be still about $1m$.

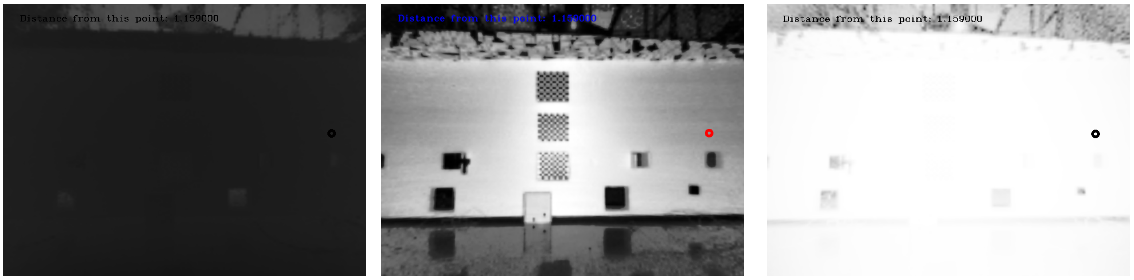


Figure 4-11: The same measurement are made on the right of the target to illustrate the problem of XYZ reconstruction from the depth.

4.1.4 Relative Depth Measurement

Using the data of the two last paragraphs, we can also compute XYZ coordinates in the T coordinate system using pinhole inversion. This time, we perform six captures in three targets presenting a relative difference of $2 inches$ in Z coordinates and we compute this relative distance to get rid of the systemic error in the setup accuracy (figure 4-12). We obtain then $4.75cm$ which represents only a few millimeters accuracy compared to the theoretical $5.08cm$.

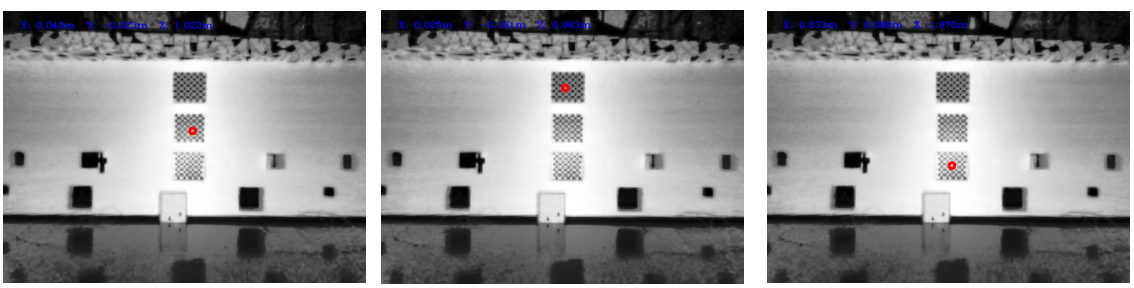


Figure 4-12: The mean XYZ coordinates are computed for three different points to deduce the relative Z distance between them and compare it to the reference benchmark.

To be more general and verify that X and Y coordinates are also correct (they depends on f_U, f_V, c_U, c_V unlike the Z coordinate) we can also measure the euclidean distance between two vertexes of a target with a known geometry framed by a camera with a random angle and distance (figure 4-13). Once again, this experiment is repeated several times and the mean distance is computed:

$$\bar{d} = \sqrt{(\bar{x_1} - \bar{x_2})^2 + (\bar{y_1} - \bar{y_2})^2 + (\bar{z_1} - \bar{z_2})^2} = 6.24cm \tag{4.3}$$

Which is once again the same rough accuracy of a few millimeters compared to the $7.12cm$ we were supposed to obtain.

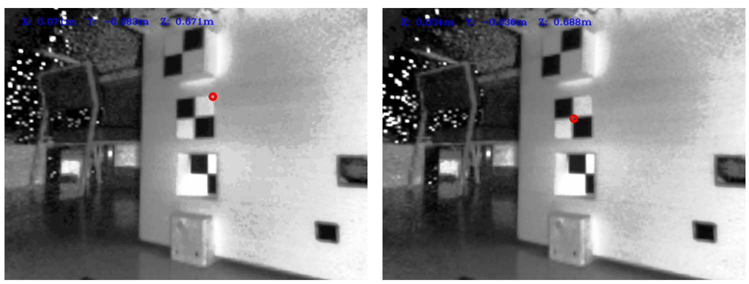


Figure 4-13: The mean XYZ coordinates are computed for two different points to deduce the euclidean distance between them and compare it to the reference benchmark.

4.1.5 3D Points Cloud Construction

In this last experiment with the ORF, a point cloud is constructed using the *PCL library* tools included in the software developed in this project. In figure 4-14, two views of the same points cloud are given. We can notice that the *visual image* is included in the process since we can see the pattern on the checkerboard. The importance of the scattering and the thermal noise can be highlighted especially if we look precisely to the edges of the checkerboard and its shape from above (which is supposed to be flat).

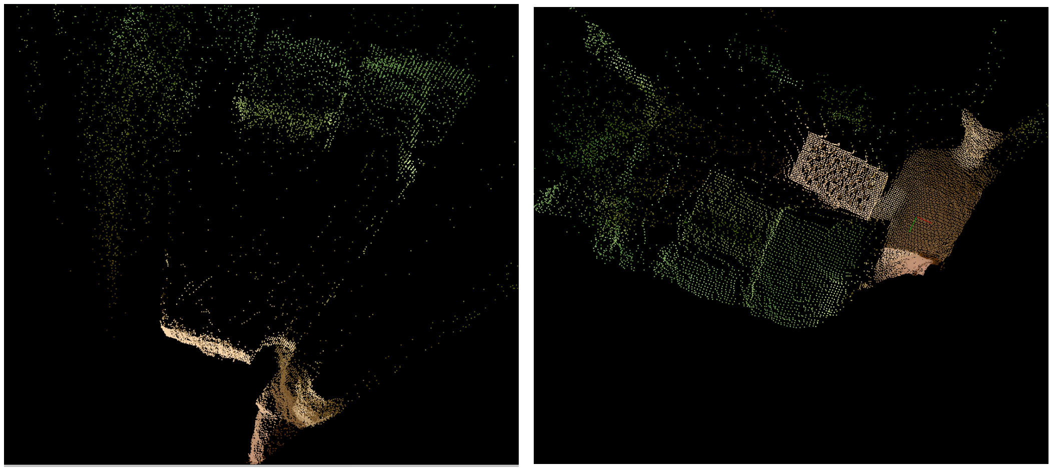


Figure 4-14: A 3D points cloud with visual information reconstructed from ORF pictures. Scattering around edges and thermal noise in the background can be observed.

Figure 4-15 illustrates the noise significance due to auto-exposure when an object is very close to the lens.

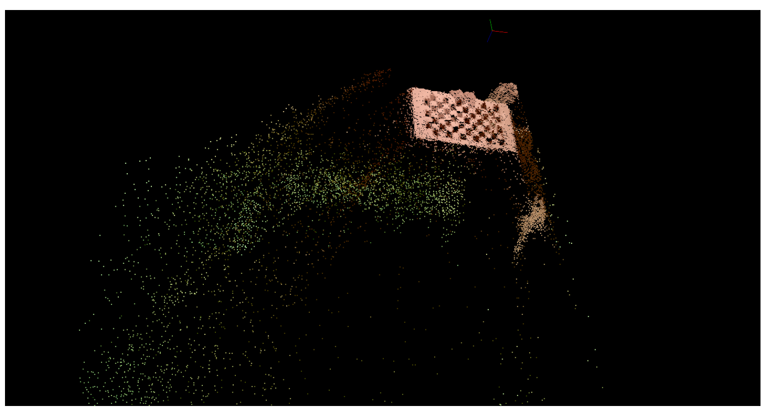


Figure 4-15: When the target is too close from the camera, the background becomes very noisy because of the auto-exposure.

4.2 Stereo Acquisition

In this section, we will discuss the efficiency of the VERTIGO sensors and algorithm. As a lot of work has already been done on the subject, for instance in [32], [27] and [23], this project simply focused on the comparison with the ORF camera and points out the complementarity of the two systems.

The reconstruction of a 3D points cloud from stereo sensors is performed through the elaboration of a *disparity map*, whose information is quite similar to that of the ORF depth map. To create this *disparity map*, different techniques can be used, a taxonomy of those is given in [30]. Basically, the process always implies the following steps: *features detection and matching*, *aggregation*, *disparity computation* and *optimization*. The quality of the *disparity image* is dependent on the number of matchable features, called *supporting points*, in the left and right pictures: if the considered environment is highly textured, this will lead to accurate 3D reconstruction while uniform patches will not give reliable results. The interstices between *supporting points* is filled by *aggregation* methods around those points which have also a lot of influence on the final *disparity map*. In figure 4-16, the results of a block matching stereo function provided in *OpenCV* libraries illustrate our remarks.

However, if the stereo vision algorithms suffer from this texture dependency, they do not encounter the same problems as the ORF in term of movement and luminosity conditions. This complementarity has been proved qualitatively with the use of a setup involving the ORF and the VERTIGO goggles capturing simultaneous pictures of the same object.

In figure 4-17, the movement reliability is highlighted: a checkerboard animated with a speed of the few dozen of centimeters per second is captured by the two sensors. Even if the stereo images could be considered a little more blurred than in the static case, they

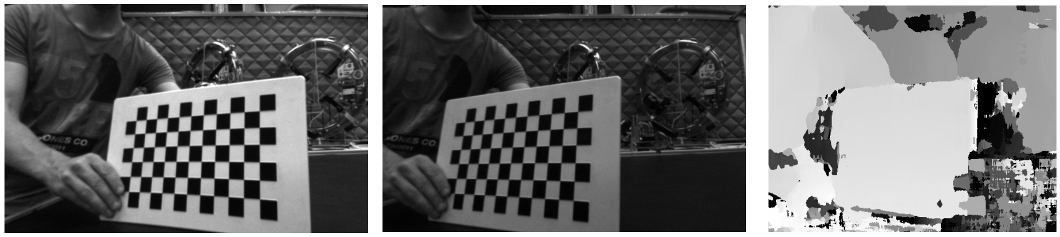


Figure 4-16: *Left*: left image. *Center*: right image. *Right*: disparity map reconstructed with *OpenCV* stereo block matching function - the checkerboard (high textured) gives a good result when the patch in the bottom right corner (uniform) is badly represented.

Figure 4-17: *Above*: visual, confidence and depth images captured with the ORF - the movement induces a error in the depth measurement. *Below*: left and right images of the stereo sensor and the disparity map - even with the movement, the result is acceptable for the checkerboard.

are still sufficient to detect features and a *disparity map* can be more reliable for textured pattern than the ORF *depth map*. An illustration of the range reliability is given in figure 4-18 where the ORF shows difficulty to estimate correctly the depth of a checkerboard too close to the camera when the stereo sensors will not notice any difference.

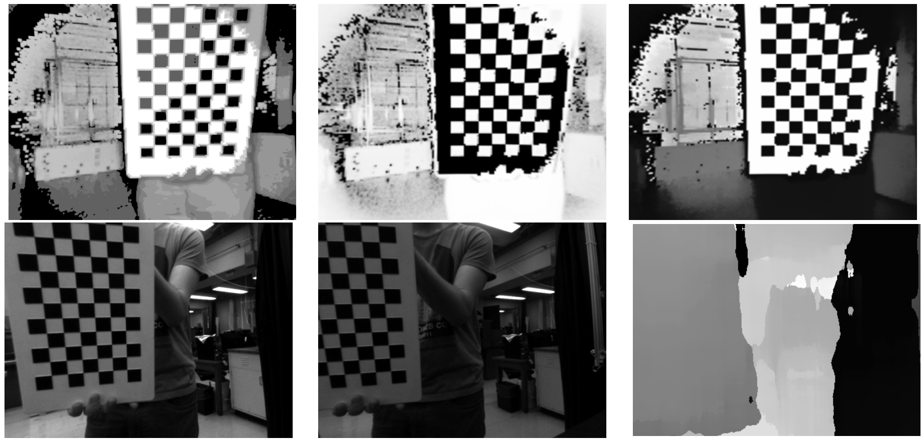


Figure 4-18: *Above*: visual, confidence and depth images captured with the ORF - as the checkerboard is too close, the depth measurement is very bad. *Below*: left and right images of the stereo sensor and the disparity map - the checkerboard depth measurement is still correct near the camera.

4.3 Multi-sensors Calibration

To determine the performances of the calibration algorithm we discussed in section 2.2.4, we connect the ORF and the stereo cameras to the VERTIGO computer and we acquire synchronous images of a checkerboard (figure 4-19). It is important for each capture to make sure the checkerboard is not moving and the luminous conditions are sufficient by checking the confidence picture (figure 4-20).

We then proceed to corner detection in those images and reconstruct separately the 3D corresponding points for the ORF (in the T coordinate system) and the stereo cameras (in the L coordinate system). The transformation matrix between them being still unknown, those points are represented in the same 3D visualization assuming a common origin (figure 4-21). It is important to understand though that this has no physical meaning, it is just a way

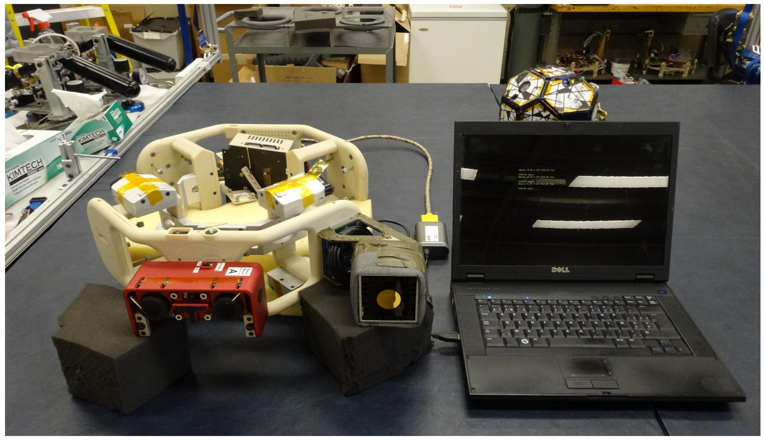


Figure 4-19: The experiment setup: VERTIGO and the ORF camera fixed to the Halo and plugged to the embedded computer.

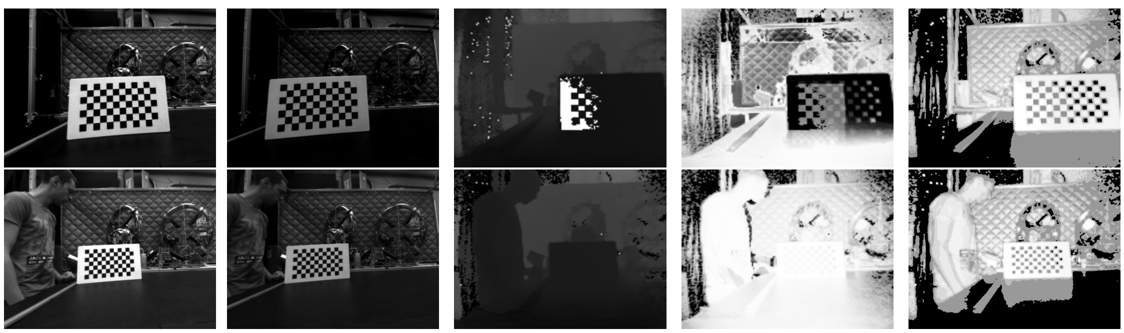


Figure 4-20: *Above*: a bad capture for calibration (from left to right: left image, right image, depth image, confidence image, ORF visual image). *Below*: a good capture (same disposition).

to verify a few parameters like the size of the cloud, the regularity of the checkerboard, their straightness or the noise. For example, we can notice the noise is more important with the ORF as predicted theoretically (figure 4-22). The RANSAC scheme will help to put aside the points that are too distant each other. The points evaluated by the stereo sensors are more accurate, since the triangulation process is based on *feature matching*, which is

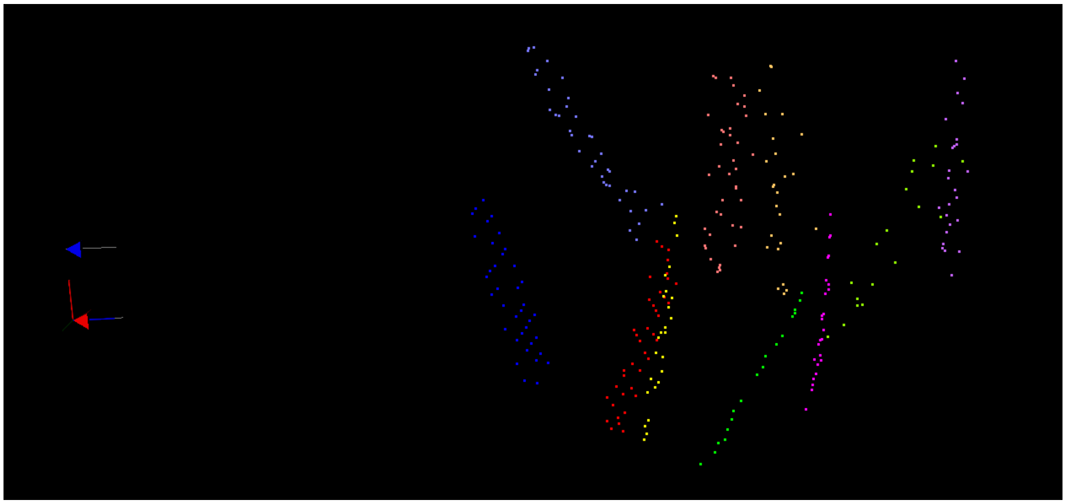


Figure 4-21: ORF and stereo reconstructed checkerboard are represented on the same image. Dark blue: stereo board 1. Light blue: ORF board 1. Red: stereo board 2. Rose: ORF board 2. Yellow: stereo board 3. Orange: ORF board 3. Dark green: stereo board 4. Light green: ORF board 4. Dark violet: stereo board 5. Light violet: ORF board 5.

an easier problem for checkerboard corner detection. Hence, the accuracy is driven by the theoretical definition of equation (2.42), though the *focal length* and the *baseline* values depend on the stereo calibration efficiency. A way to check the reliability of the 3D reconstruction is to compute the euclidean distances between each point and compare them to the theoretical $25mm$ of the real checkerboard. On the whole, those distances are always situated around $30mm$ and are bigger for ORF which makes theoretical sense, since the noise, naturally higher for the ORF, always has a positive contribution on distance measurements.

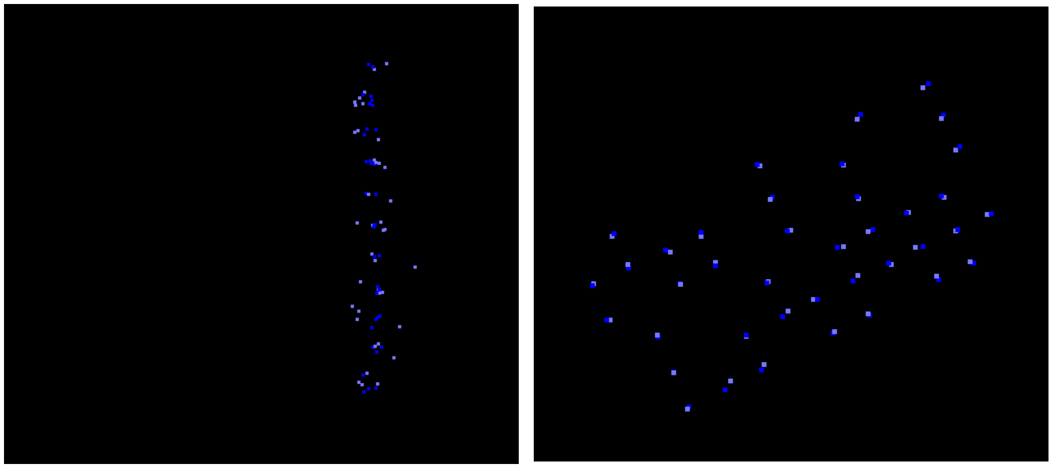


Figure 4-22: *Left*: the ORF reconstructed checkerboard (light blue) is noisier than the stereo one (dark blue) in the Z direction. *Right*: this effect is less pronounced in the XY plane. The holes are due to rejection of the points considered as too noisy.

In the next part of the process, the sum of the distances between the range finder and the stereo device is minimized and extrinsic matrices are computed. Here, we display the M_{TL} matrices of a calibration using 7 checkerboard capture with the sensors plugged on the Halo. With RANSAC:

$$\begin{pmatrix} 0.998 & 0.0465 & 0.0294 & -0.166 \\ -0.0456 & 0.999 & -0.0299 & 0.00767 \\ -0.0308 & 0.0285 & 0.999 & -0.0589 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.4}$$

Without RANSAC:

$$\begin{pmatrix} 0.991 & 0.128 & -0.0435 & -0.14 \\ -0.128 & 0.992 & -0.00107 & -0.00672 \\ 0.043 & 0.00663 & 0.999 & -0.0846 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.5}$$

In those matrices, we can see that the sensors are pretty aligned (rotation coefficient almost equal to 1 on the diagonal and 0 elsewhere) and we can compare the translations with the theoretical one measured on the CAD model of the INSPECT project:

	CAD Model		**Calibration Results**	
	in *inches*	in cm	with RANSAC (in cm)	without RANSAC (in cm)
X	6.3345	**16.09**	**16.6**	14
Y	0.7903	**2.01**	**0.77**	-0.67
Z	1.8662	**4.74**	**5.89**	8.46

Table 4.1: Comparison of the theoretical and calibration results of the translation between the ORF and the left camera.

Given the quality of the sensors and the global mechanical accuracy of the setup, we can consider those results to be quite good. However, it may be important to discuss the limitations:

- **Stereo calibration parameters dependency**: First, as we just reminded, the stereo precision is a function of the *focal length* f and the *baseline* b between left and right cameras. Yet, those values are extracted from the stereo calibration and small imprecision can strongly affect the error on the final Halo calibration. Indeed, during the *triangulation*, errors on b and f will cause the reconstructed points cloud to inflate or deflate (figure 4-23). As the minimization process of this calibration calculates the rotation and translation between the geometric center of each cloud, the calibration will be then dependent on the localization of observed checkerboards (directly influencing the position of the cloud's geometric centers). A checkerboard situated essentially along the X axis would lead to a higher error on X. In our example, we observe mainly checkerboard along Z but one of them is shifted on the Y axis, which explain the higher error on Y. It is important to understand that this problem is only due to the stereo imprecision, though. Because of this effect, the global observation drawn from all the calibration tests shows a bigger error in Z. As a matter of fact, if the checkerboards photographed during the calibration process are near the center of the XY plane, they by necessity have a positive Z coordinate.

- **ORF noise**: Secondly, the noise of the ORF is directed along the depth axis and is a function of the luminosity and the materials in the environment. Once again, as the cameras look at checkerboard with a positive Z, this component is more concerned by the calibration errors. However, the random nature of this process leads to an error with a zero mean value and then have a lower influence than the previous effect.

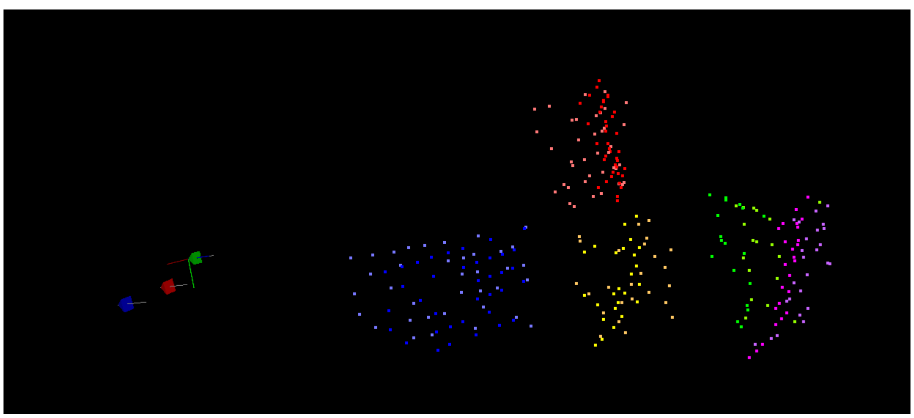


Figure 4-23: The ORF and stereo reconstructed checkerboard are now superposed thanks to the computed transformation matrix. If we look closely, the ORF checkerboard near the camera is shifted toward the camera when the farthest ORF checkerboard is shifted in the other direction. This highlights the fact that the stereo points cloud is too small, due to stereo calibration parameters inaccuracies.

4.4 Multi-Sensor Fusion

4.4.1 Ground Results

In this section, the results provided by images captured in the ground laboratory are analyzed. We use the setup in figure 4-19 to acquire simultaneous pictures from the ORF and VERTIGO (figure 4-24).

Figure 4-24: A typical sample of input pictures for fusion. *From left to right*: ORF depth, ORF visual, ORF confidence, left image, right image.

With the data provided by the range finder, a 3D points cloud is built as in section 4.1.5 (figure 4-27). To gain in clearness, the 3D cloud is sub-sampled, though it would have been possible to do the same with each object in the FoV of the three cameras without sub-sampling. The next step consists of computing the variance σ_w^i of the ORF total noise (thermal and scattering) assigned to each 3D point by using its depth map and the confidence map. Once this variance is computed, we can represent the interval $[d_T^i - 3\sigma_w^i;\ d_T^i + 3\sigma_w^i]$ around each point in the direction of the depth, defined by the axis linking this point and the optical center of the ORF. This interval is discretized into steps whose width is equal to one quarter of the stereoscopic precision (figure 4-27).

Then, the coordinates of those points $p_T^{i,j}$ are moved in the left camera coordinate system L, and their projection on the left and right image planes of the stereo sensor is computed (figures 4-25 and 4-26).

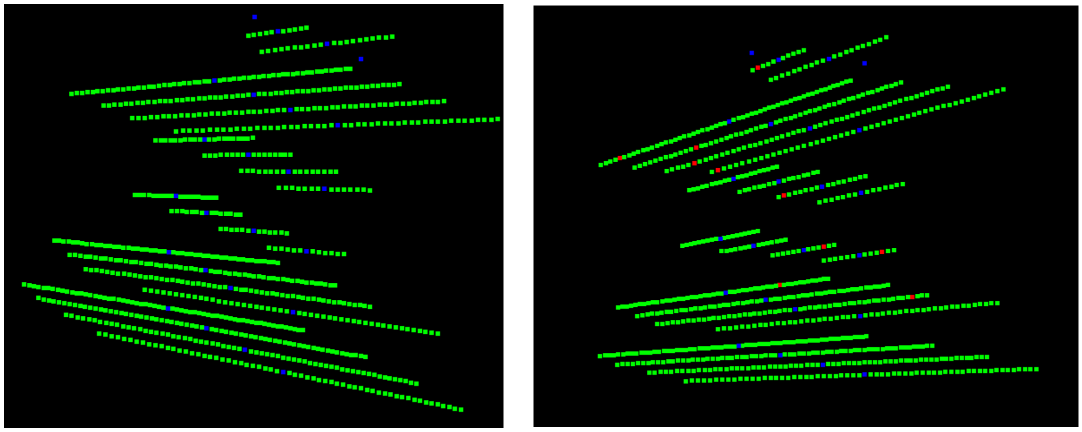


Figure 4-25: *Left*: in blue, the 3D ORF points; in green, a interval has been constructed around those points in function of the noise. *Right*: in blue and green, idem; in red, the 3D points computed in the end of the fusion.

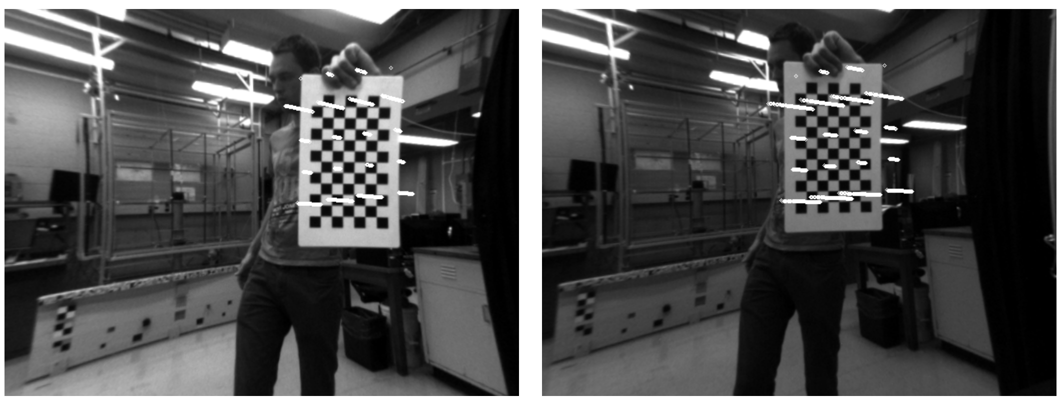


Figure 4-26: The interval around each 3D point are reprojected in the stereo images (subsampled in the picture).

It is therefore possible to calculate the a-priori probability for the ORF ($P[p^{i,j}|I_T]$) and the stereoscopic system ($P[p^{i,j}|I_L, I_R]$) and the joint probability ($P[p^{i,j}|I_T, I_L, I_R]$) which is minimized to find the new point $\hat{p}^i$ arising from the fusion (figure 4-27). If we compare the results before and after the fusion, we can see that not only is the computation time at best ten times higher than the acquisition on a laptop computer, but the accuracy is worse

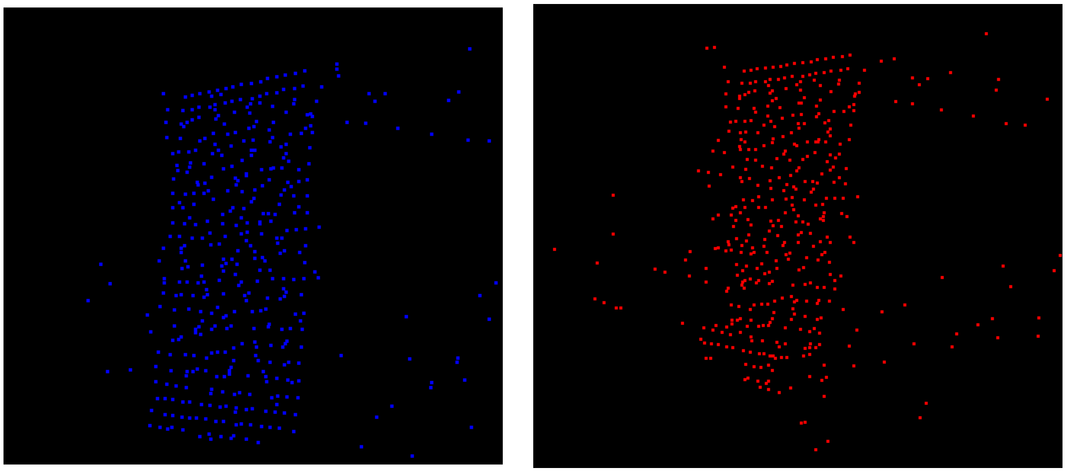


Figure 4-27: *Left*: a flat checkerboard before fusion. *Right*: the same checkerboard after the fusion. Unlike the first theoretical assumptions, the fusion algorithm produces noise.

after the fusion. We will discuss the reason further in this document.

4.4.2 RGA Results

As for the ground pictures, the fusion algorithm has been tested in a zero gravity environment provided by the NASA RGA. To be able to use the data collected further in the project, the thermocam acquisition has been also added. To better understand the intent of this experiment, it may be important to remind the context of the project. Indeed, as mentioned in the introduction, this fusion algorithm is part of a SLAM algorithm aiming at localizing, mapping and understanding the motion parameters of the objects around. Moreover, this algorithm is intended to be part of a control process meant for operation in an environment that cannot be reproduced outside a zero gravity environment. Tests in a parabolic flights are then important to prove the reliability of the sensor fusion and acquisition as part as a full system as well as its performances in front of six DoF moving objects. However, given the amount of data, lot of post-processing has still to be done but we can already see the complementarity of VERTIGO and ORF in the datasets meaning that an adapted algorithm could recover the best knowledge of the aircraft inside by using stereo and ToF cameras in the same time.

4.4.3 Discussion

In this section, we showed that, despite the demonstrated complementary operation of VERTIGO and the ORF, the new fusion algorithm does not take full advantage of this complementarity. To discuss the explanations, we classified them into four categories.

ORF default can induce fusion failure

If this algorithm is supposed to enhance the accuracy of 3D cloud built with the ORF measurements, the image capture makes it impossible for the stereo system to recover from bad images obtained by by the ORF. This has been mainly observed during RGA tests sessions in which saturation, significant relative speeds, and a too short range have provided bad results but that could be improved using VERTIGO reliability. Using the notations of section 2.3.2, we can say that **certainty** is not assured to increase with the fusion.

Limited range

As the size of the merged points cloud is defined by the subspace where all camera FoV are crossing, this subspace is then smaller than VERTIGO or the ORF alone (hardware issue). In other words, the **completeness** decreases during the fusion process.

Non-linearity of the stereo process

The most important phenomenon that seems to explain the bad results of the algorithm is the following: to calculate the stereo probability, we use the projected points in the left and right image by associating a *likeliness coefficient* to each couple of points (assuming a rectangular window around). However, it is notable in figure 4-26 that small calibration errors lead to small shifts between left and right images (in other words, the projection does not point exactly the same detail as it should). When we build a 3D map from stereo pairs in a traditional way, we first match features then project them, which will cause small imprecisions we can deal with; the error can be linearized in this way. On the other hand, in the fusion algorithm case, we perform feature matching after the projection. Therefore, the matching can diverge in case of small calibration offsets; the error cannot be linearized in that way (figure 4-28). In conclusion, the mechanical configuration and the camera resolution of the INSPECT system leads to calibration errors and then corrupts the final result. This loss in **accuracy** could be avoided in other types of algorithms.

4.5 Future Work

4.5.1 Improving Calibration

As we concluded previously, the Halo calibration method that has been designed in this thesis looks promising. Ameliorations that could be implemented in the future are:

- Improving the triangulation: if a simple method of triangulation is sufficient to process data in live, calibration must be very accurate at the expense of computation time. Many problems were discovered throughout the project due to the lack of calibration parameter accuracy.

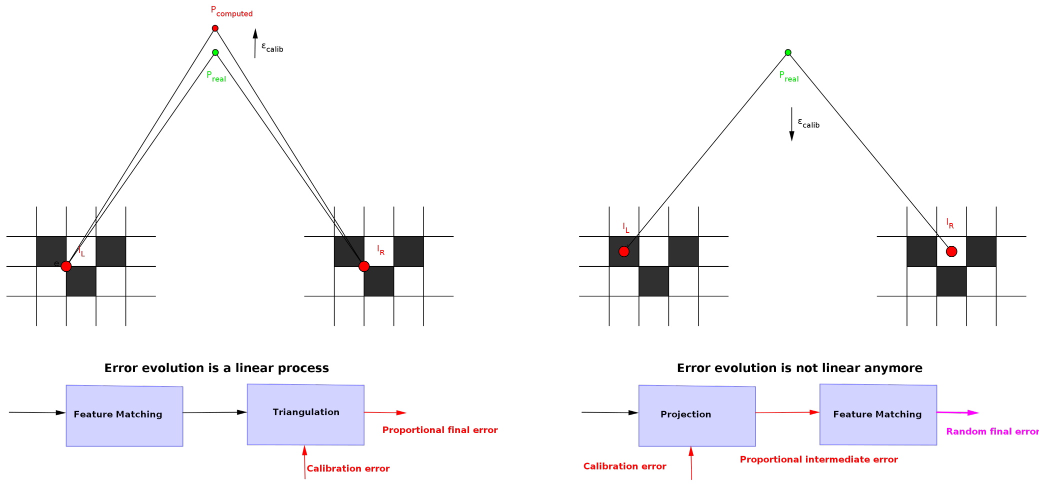


Figure 4-28: When the calibration error has a proportional impact on the result in the standard stereo 3D construction, this is not true in our algorithm.

- Integrate the thermocam: to perform thermocam calibration, we can rely on the same methods as developed in the current version of the software. Intrinsic calibration could use the same OpenCV functions but a IR calibration pattern would be needed [34], [36]. Concerning the extrinsic calibration, with the same pattern, a solution would be to integrate it into the stereo calibration and use a N-camera OpenCV function instead of the classical stereo calibration function.

4.5.2 Improving this Fusion Algorithm

Despite the observed lack of robustness of the algorithm, it is possible to improve performance through:

- Find a better function expressing the link between the confidence image and the thermal noise variance.
- Minimize the error during the computation of the stereo probability by the use of a

more reliable definition of the matches between stereo points despite of the calibration uncertainty.

- At the end of the algorithm, build a thermal 3D point cloud by re-projecting the 3D point cloud in the thermocam image plane, read the value and combine it with the point position.

4.5.3 Building a New Fusion Algorithm

Even if the issue concerning the stereo probability computation is overcome, there is still the problem of the lack of certainty: it requires good images from the ORF as an input which seems not to be always true. Several tracks can be therefore explored:

- **Keep a single stream**: the idea would be then to perform stereo triangulation first and improve it with the help of the ORF which does not require any matching function. However, the theoretical accuracy of the ORF is supposed to be less and, once again, in case of bad images from stereo cameras (not enough textures), the entire process would suffer.
- **Time division**: However, this would result in a fusion from a spatial point of view only and this does not correspond to the goals of the INSPECT project.
- **Spatial division**: The idea is to reconstruct two parallel 3D clouds with VERTIGO and the ORF in two different streams then use confidence and range to split the space into different parts. This time, the fusion is only temporal, which don not really match the goal neither.
- **Other methods**: Quantity of promising methods have been found in [8] in the end of this internship. When moving forward in the INSPECT project, refinements could be done to select the most interesting. To name but one, [28] propose a fusion method adapted for fast SLAM algorithms.

Chapter 5

Conclusions

This thesis is the result of a final project at Institut Supérieur de l'Aéronautique et de l'Espace achieved within the Massachusetts Institute of Tehcnology Space Systems Laboratory. It aimed at exploring new promising techniques for visual navigation of automated space systems and more especially the fusion of Time-of-Flight and stereoscopic cameras data. During this internship, literature review, theoretical development, software implementation and practical test sessions on the ground and inside parabolic flights have been part a bottom-up approach to improve autonomy efficiency in zero gravity robotics.

One of the main contribution of this work is the adaptation of a calibration algorithm to a cheap hardware including Time-of-Flight and stereoscopic cameras without any prior requirements about the geometric configuration and the mechanical accuracy. Even if could be still improved in different ways, involving the incorporation of a thermographic camera and the refinement of the triangulation methods, this process has proven to give good results with different setups and is a major step towards the success of multi-sensors fusion algorithms in space systems and robotics in general.

Even if it performed better results in other fields of application, the fusion implemented in this thesis has turned out to be unsuited for the low precision of the setup, the modest resolution of the sensors and the limited computation performances. However, it guided the research towards new types of algorithms that seem very promising for SLAM processes in the domain of visual navigation in robotics. Instead of trying to improve the accuracy of a full three dimensional map in a single stream, the idea would be to match interesting feature points and compute their 3D coordinates in two different streams before merging them from a probabilistic point of view. Once again, IR informations could be integrated in this process to enrich the variety.

Bibliography

[1] Christian Beder, Bogumil Bartczak, and Reinhard Koch. A combined approach for estimating patchlets from pmd depth images and stereo intensity images. In *Pattern Recognition*, pages 11–20. Springer, 2007.

[2] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263–296, 2008.

[3] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[4] D. C. Brown. Decentering distortion of lenses. *XXXII(3):444-462*, 1966.

[5] Nicola Brusco, Pietro Zanuttigh, David Taubman, and Guido Maria Cortelazzo. Distortion-sensitive synthesis of texture and geometry in interactive 3d visualization. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 256–263. IEEE, 2006.

[6] Carlo Dal Mutto, Pietro Zanuttigh, and Guido M Cortelazzo. A probabilistic approach to tof and stereo data fusion. *3DPVT, Paris, France*, 2, 2010.

[7] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[8] Sergi Foix Salmerón, Guillem Alenyà Ribas, Carme Torras Genís, et al. Exploitation of time-of-flight (tof) cameras. 2010.

[9] Andrea Fusiello, Vito Roberto, and Emanuele Trucco. Symmetric stereo with multiple windowing. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(08):1053–1066, 2000.

[10] David Gallup, J-M Frahm, Philippos Mordohai, and Marc Pollefeys. Variable baseline/resolution stereo. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[11] S Burak Gokturk, Hakan Yalcin, and Cyrus Bamji. A time-of-flight depth sensor-system description, issues and solutions. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 35–35. IEEE, 2004.

[12] Sigurjon Arni Gudmundsson, Henrik Aanaes, and Rasmus Larsen. Fusion of stereo vision and time-of-flight imaging for improved 3d estimation. *International Journal of Intelligent Systems Technologies and Applications*, 5(3):425–433, 2008.

[13] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[14] Richard I Hartley and Peter Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

[15] Janne Heikkila and Olli Silvén. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112. IEEE, 1997.

[16] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987.

[17] Timo Kahlmann and Hilmar Ingensand. Calibration and development for increased accuracy of 3d range imaging cameras. *Journal of Applied Geodesy*, 2(1):1–11, 2008.

[18] Young Min Kim, Christian Theobalt, James Diebel, Jana Kosecka, Branislav Miscusik, and Sebastian Thrun. Multi-view image and tof sensor fusion for dense 3d reconstruction. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1542–1549. IEEE, 2009.

[19] Klaus-Dieter Kuhnert and Martin Stommel. Fusion of stereo-camera and pmd-camera data for real-time suited precise 3d environment reconstruction. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4780–4785. IEEE, 2006.

[20] Mikko Kytö, Mikko Nuutinen, and Pirkko Oittinen. Method for measuring stereo camera depth accuracy based on stereoscopic vision. In *IS&T/SPIE Electronic Imaging*, pages 78640I–78640I. International Society for Optics and Photonics, 2011.

[21] Marvin Lindner and Andreas Kolb. Calibration of the intensity-related distance error of the pmd tof-camera. In *Optics East 2007*, pages 67640W–67640W. International Society for Optics and Photonics, 2007.

[22] Marvin Lindner, Ingo Schiller, Andreas Kolb, and Reinhard Koch. Time-of-flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding*, 114(12):1318–1328, 2010.

[23] Konrad Makowka. Visual based navigation with an experimental satellite testbed. *Master Thesis, Lehrstuhl für Raumfahrttechnik, Technische Universitat Munchen, Germany*, 2012.

[24] MESA Imaging. *SR4000 Data Sheet*, 2011. Rev. 5.1.

[25] Ouk Choi Radu Horaud Miles Hansard, Seungkyu Lee. *Time-of-Flight Cameras: Principles, Methods and Applications*. Springer, 2012.

[26] Harvey B Mitchell. *Multi-sensor data fusion*. Springer, 2007.

[27] Elias Müggler. Visual mapping of unknown space targets for relative navigation and inspection. *Master Thesis, Mechanical Engineering, ETH Zurich, Switzerland*, 2012.

[28] Chayakorn Netramai, Melnychuk Oleksandr, Chanin Joochim, and Hubert Roth. Motion estimation of a mobile robot using different types of 3d sensors. In *Autonomic and Autonomous Systems, 2008. ICAS 2008. Fourth International Conference on*, pages 148–153. IEEE, 2008.

[29] Martin Rufli, Davide Scaramuzza, and Roland Siegwart. Automatic detection of checkerboards on blurred and distorted images. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3121–3126. IEEE, 2008.

[30] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[31] Ingo Schiller, Christian Beder, and Reinhard Koch. Calibration of a pmd-camera using a planar calibration pattern together with a multi-camera setup. *The international archives of the photogrammetry, remote sensing and spatial information sciences*, 37:297–302, 2008.

[32] Brent Edward Tweddle. Computer vision-based localization and mapping of an unknown, uncooperative and spinning target for spacecraft proximity operations. *PhD, Massachusetts Institute of Technology, Cambridge, MA*, 2013.

[33] Geert Van Meerbergen, Maarten Vergauwen, Marc Pollefeys, and Luc Van Gool. A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal of Computer Vision*, 47(1-3):275–285, 2002.

[34] Stephen Vidas, Ruan Lakemond, Simon Denman, Clinton Fookes, Sridha Sridharan, and Tim Wark. A mask-based approach for the geometric calibration of thermal-infrared cameras. *Instrumentation and Measurement, IEEE Transactions on*, 61(6):1625–1635, 2012.

[35] Gang Xu and Zhengyou Zhang. *Epipolar geometry in stereo, motion and object recognition: a unified approach*, volume 6. Springer, 1996.

[36] Will Warren Yuncong Chen. 3d fusion of infrared images with dense rgb reconstruction from multiple views - with application to fire-fighting robots. *USCD*, 2013.

[37] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

[38] Jiejie Zhu, Liang Wang, Ruigang Yang, and James Davis. Fusion of time-of-flight depth and stereo for high accuracy depth maps. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.