



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

UNIVERSITA' DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria Industriale DII

Corso di Laurea Magistrale in Ingegneria Aerospaziale

Pushbroom sensor modelling for HYPSOS telescope calibration

Relatore:

Ch.mo Prof. Giampiero Naletto

Correlatori:

Ch.ma Dott.ssa Cristina Re

Ch.mo Dott. Gabriele Cremonese

Laureanda:

Chiara Doria

Matricola n. 1242419

Anno Accademico 2020/2021

Ai miei genitori
che mi hanno permesso di
raggiungere questo traguardo
e a cui devo tutto

A mio fratello
che mi ha insegnato a
non arrendermi mai

A Federico
che ogni giorno mi sopporta
e mi supporta

A voi va il mio ringraziamento più grande

Abstract

Nowadays, one of the goals that we always try to satisfy when facing a space mission that has as objective the observation of solid bodies, is to obtain the DTM, Digital Terrain Model, of the planet's surface that we are going to observe. The DTM model is a model that includes three-dimensional spatial characteristics of the region framed and it is the result of complex procedures starting from at least two images of the same region acquired from different point of views.

HYPSSOS, HYPerspectral Stereo Observing System, is an instrument that will employ a new technology able to give simultaneously both 3D spatial and spectral information of the observed features. With the spectral information and the spatial information, we can create a representation of the surface linked with the hyperspectral information emitted and diffused by the objects on the surface in 3D space. Thus, we name the resulting data product Hyperspectral Digital Terrain Model (HDTM).

An important step for the use of the stereo camera is the modelling and calibration. In this thesis I present the work done for realizing a technique of geometric calibration of HYPSSOS, which acquires the data by means of a pushbroom scanning system. The adopted technique follows the method introduced by Gupta and Hartley, that under some assumptions, allows to greatly simplify the computational steps usually involved in the pushbroom model calibration. This analysis allows to reproduce results at the sub-pixel scale, allowing also to model the non-linear effects induced by the camera movement. The accuracy of the model has been validated using different approaches: we used a series of simulated images obtained with a ray-tracing software and a numerical benchmark.

An important task of the work has been to implement the estimation of the model proposed by Gupta & Hartley for the linear pushbroom systems and validate the method: this has been done both with numerical inputs and with the extraction of the chessboard corners coordinates extracted from the ray-traced images by means of a Harris Corner detector. Furthermore, I searched for the optimal chessboard geometry to provide the smallest error in the detection.

Sommario

Al giorno d'oggi, uno degli obiettivi che si cerca sempre di soddisfare in una missione spaziale che ha come obiettivo l'osservazione di corpi solidi, è quello di ottenere il DTM, Digital Terrain Model, della superficie del pianeta da osservare. Il modello DTM è un modello che comprende le caratteristiche spaziali della regione analizzata, a partire da almeno due immagini della stessa regione acquisite da punti di vista differenti.

HYPSON, HYPerspectral Stereo Observing System, è uno strumento che impiegherà una nuova tecnologia che sarà in grado di restituire simultaneamente sia le informazioni spaziali 3D che le informazioni spettrali dell'oggetto osservato. Con le informazioni spaziali e spettrali, possiamo creare una rappresentazione della forma della superficie collegata insieme alle informazioni iperspettrali emesse e riflesse dagli oggetti ricoperti dalla superficie nello spazio 3D. Il modello che ne risulta viene chiamato Hyperspectral Digital Terrain Model (HDTM).

Un passaggio importante per l'utilizzo della stereo camera è la modellazione e calibrazione; In questa tesi presenterò il lavoro svolto per la calibrazione geometrica del sistema di acquisizione pushbroom di HYPSON, seguendo il metodo introdotto da Gupta e Hartley che, fatte alcune ipotesi, consente di semplificare notevolmente i passaggi computazionali solitamente coinvolti nella calibrazione del modello pushbroom. Questa analisi ha permesso di riprodurre i risultati alla scala sub-pixel, consentendo anche di modellare gli effetti non lineari indotti dal movimento della stereo camera. L'accuratezza del modello è stata validata utilizzando diversi approcci: abbiamo sfruttato una serie di immagini simulate ottenute con un software di ray-tracing e un benchmark numerico.

Un obiettivo fondamentale del lavoro è stato quello di implementare la stima del modello proposto da Gupta & Hartley per i sistemi pushbroom lineari e validare il metodo sia con input numerici, sia grazie alle coordinate dei corners della scacchiera estratti dalle immagini ottenute con ray-tracing per mezzo di un rilevatore Harris Corner. Inoltre, ho lavorato per capire quale potesse essere la migliore geometria della scacchiera per la calibrazione, che permettesse di ottenere un errore minore nel rilevamento.

Contents

List of Figures	9
List of Tables	11
Chapter 1: INTRODUCTION	
1.1 Context	13
1.1.1 Planetary photogrammetry	14
1.1.2 Stereo camera overview	16
1.2 HYPSSOS instrument concept	18
1.2.1 Optical prototype	19
1.3 Research objective	21
Chapter 2: LITERATURE OVERVIEW AND METHODOLOGY	
2.1 Overview of hyperspectral imaging	23
2.2 Camera calibration	24
2.2.1 Geometric parameters	25
2.2.2 Computer Vision coordinate system	25
2.2.2.1 World to camera reference system transformation	26
2.2.2.2 Camera to image reference system transformation	26
2.3 Camera models	28
2.3.1 Pinhole camera model	28
2.3.2 Pushbroom camera model	30
2.3.2.1 Camera matrix	32
2.3.2.2 Intrinsic parameters matrix	34

2.3.2.3 The DLT method	38
Chapter 3: STUDY SETUP	
3.1 Experimental setup	40
3.2 Bi-planar calibration gauge	43
3.2.1 Image acquisition	44
3.3 Harris corner detector	45
3.3.1 Python implementation	48
Chapter 4: RESULTS	
4.1 Results obtained with a numerical simulation	49
4.2 Results obtained with the ray-traced images	52
4.2.1 Nadiral image analysis on a set of corners	52
4.2.2 Inclined image analysis	54
4.2.3 Harris detected corners analysis	55
4.2.4 Instability of the detection process	56
4.3 Chebyshev polynomials refinement	61
4.3.1 Re-estimated error	63
4.4 Correlation between parameters	65
Conclusions	67
References	69

List of Figures

Chapter 1:

Figure 1.1: Disparity map creation combining Left and Right images.	15
Figure 1.2: Basic concept of the stereo vision.	16
Figure 1.3: HYPSSOS' Field of View ground projection.	19
Figure 1.4: Optical layout of HYPSSOS prototype.	20

Chapter 2:

Figure 2.1: Example of hyperspectral imaging approach on Earth.	24
Figure 2.2: Representation of a camera projection.	25
Figure 2.3: Representation of the image distortion: image without distortion, with Barrel Distortion and with Pincushion Distortion.	28
Figure 2.4: Schematic image of the pinhole camera model working.	29
Figure 2.5: Schematic projection of the pushbroom camera model acquisition system.	31

Chapter 3:

Figure 3.1: Schematic image of the laboratory set-up used to simulate the acquisition system of HYPSSOS.	41
Figure 3.2: Stereo Validation setup (both for STC and HYPSSOS).	41
Figure 3.3: The scheme of HYPSSOS acquisition during the calibration.	42
Figure 3.4: A schematic view of STC/HYPSSOS acquisition system on a simulated Mission.	43
Figure 3.5: Bi-planar calibration gauge used for the camera calibration.	44
Figure 3.6: Synthetic image process of acquisition with SurRender.	45
Figure 3.7: a) represents the flat region; b) represents the edge region. c) represents the corner region.	46
Figure 3.8: Eigenvalues for the different kind of region.	47
Figure 3.9: Harris Corner detection on Python applied on a simulated image of the bi-planar chessboard.	48

Chapter 4:

Figure 4.1: Plots of all the corners of the bi-planar chessboard.	50
Figure 4.2: Representation of the residual values for the eighteen points, obtained from the simulation for the nadiral image. (The value is multiplied by a factor 1.05)	53
Figure 4.3a: Histogram of the residual values (in pixel units) obtained from the primary analysis for all detected corners with Harris along u direction.	55

Figure 4.3b: Histogram of the residual values (in pixel units) obtained from the primary analysis for all detected corners with Harris along v direction.	56
Figure 4.4: Representation of the variation of the focal length due to an error of detection with an evaluation on 390 corners.	57
Figure 4.5: Representation of the variation of the focal length due to an error of detection with an evaluation on 4900 corners. Image a) with a dimension of the squares of the chessboard of 2 mm and b) a dimension of the squares of the chessboard of 5 mm.	58
Figure 4.6: Representation of the variation of the focal length due to an error of detection with an evaluation on 22500 corners. Image a) with a dimension of the squares of the chessboard of 2 mm and b) a dimension of the squares of the chessboard of 5 mm.	59
Figure 4.7: Representation of the variation of the focal length due to an error of detection with an evaluation on 800 corners.	60
Figure 4.8a: Histogram of the residual values (in pixel units) obtained from the refinement with Chebyshev for all detected corners with Harris corner detector, along u direction.	64
Figure 4.8b: Histogram of the residual values (in pixel units) obtained from the refinement with Chebyshev for all detected corners with Harris corner detector along v direction.	64

List of Tables

Chapter 1:

Table 1.1: Principal parameters of HYPSSOS optical prototype.

20

Chapter 4:

Table 4.1: Results obtained in Python with the sets of points produced with the numerical simulation with the MATLAB code for a nadiral acquisition. 51

Table 4.2: Results obtained in Python with the sets of points produced with the numerical simulation with the MATLAB code for a tilted acquisition of 20°. 51

Table 4.3: Values obtained with eighteen corners. (Nadiral image) 53

Table 4.4: Values obtained with eighteen corners. (Tilted Image). 54

Table 4.5: Values obtained with all detected corners. 55

Table 4.6: Coefficients for the Chebyshev polynomial fitting the residuals in u and v variables. 62

Table 4.7: Values obtained with a 2D Chebyshev polynomial refinement. 63

Chapter 1: INTRODUCTION

In this chapter I will shortly introduce the planetary photogrammetry process and the stereo camera concept. Then I will describe the optical prototype of HYPSSOS and will give an overview of what are the objective of my work and an outline of my thesis.

1.1 Context

HYPSSOS, HYPerspectral Stereo Observing System, as mentioned before, is a new kind of remote sensing pushbroom instrument born from the joining of two different instruments: a stereoscopic pushbroom camera system and a spectrograph. It is a very compact instrument so it will be very useful for applications on small size spacecrafts, or on interplanetary missions. This instrument collects light from two different perspectives, which allows to realize the tridimensional model of the observed surface, and then to extract the spectral information from each resolved element, thus giving a full 4-dimensional hypercube dataset.

In all modern planetary space mission, the objective is usually to provide the digital terrain model (DTM) of the surface which is usually obtained by a stereo camera onboard of the orbiter. The DTM is useful for morphological studies of the observed planet. At the same time, it is really important to also get information about the surface composition, for this reason it is necessary to also involve a spectrograph. Usually, the 4D (three spatial and one spectral) information is obtained by means of two different instruments, a stereo camera, and a spectrograph; unfortunately, this is rather difficult to perform because the two instruments have different Field of View, different pixel size and different covered region. This is the reason why HYPSSOS has been designed: the main task to achieve is the cross-calibration of the two instruments to get the spectral DTM, SDTM.

In this way a single instrument will provide all the data and information to fully characterize the observed planet surface without needing to merge the data collected from the two instruments.

Another main task will may be to get an accurate software that can integrate the information to get the SDTM, so there is for example the problem of the wavelength dependent characteristics of the instruments and of the spectral bidirectional reflectance distribution function of the observed surface that can impact on the quality level.

1.1.1 Planetary photogrammetry

Photogrammetry is a digital image processing technique in terms of measurement and interpretation, that allows to derive the shape and the location of an object, posed to a big distance, using one or more photograph. The primary purpose is to obtain a three-dimensional reconstruction of an object in a digital form or a graphical form.

If we talk about the acquisition of planetary bodies stereo-images, what we want to obtain is the DTM, Digital Terrain Model, deriving also from laser altimeter experiments on board the orbiting spacecraft.

When we reduce the three-dimensional object to a two-dimensional one, we have a loss of information. There may be areas in which the contrast is too small, or the size is limited, and for this, there will be a geometric changing in the object. There are geometric changes caused by the relative position between camera and object, optical lens defect, and the shape of the object itself; but there are also radiometric changes because, the electromagnetic radiation reflected from the object and recorded by the camera, may be affected by the transmission media, air or glass.

To define the reconstruction of the object from a stereo-image or a photograph it is important to describe the optical process of the image acquisition without neglecting the elements that contribute to the process, such as: light sources, properties of the surface of the observed object, the transmission media in which the light travels and the technology implied.

In the recent years, all the space missions that aim to explore bodies with a solid surface want to derive the topography of the surface with great precision, to provide important information for geologists studying the structure, the geomorphology and the physic of planets and asteroids, for these reasons different tools and methods have been implemented. In contrast to the mapping of the Earth surface, the other planets and satellites present the following problems:

- Planets or satellites surface has no definition of the geoid, a reference surface that is used for height and depth measurements. The definition of the geoid requires information on the internal and crustal structure of the planet, on density and gravity, and on any gravitational and density anomalies. Very often we do not have this information from other planets, or it is inaccurate.
- Their atmospheres present different characteristics respect to the Earth one.

So, considering all that have been said till now, the process to derive the DTM starting from raw data, is very complex and it must answer to two main requirements: an extremely large volume of data and a big accuracy. In particular, the baseline and the stereo-angle are crucial for the final accuracy. The stereo angle between the two images can be fixed by mechanical constraints (depending on the stereo camera configuration) as in the along-track mode or by the geometry of the photogrammetric block characteristics defined by the choice of the orbital images.

Despite the presence of new algorithms that allow better results in qualitative terms, continuous and smooth surface, and in terms of time for the process, usually the trend is to opt for classical area-based techniques refining the method improving the different phases of the process.

After the calibration procedure, that allows to know the characteristics of the imaging system, it is possible to start with the three-dimensional reconstruction. The first step is to find the disparity map, usually achieved with a stereo algorithm and allows to describe the correspondence between two images. The **disparity map** is an image where the value of each pixel is equal to the difference between the coordinates of different pixel location of the two images of the stereo pair (Figure 1.1).

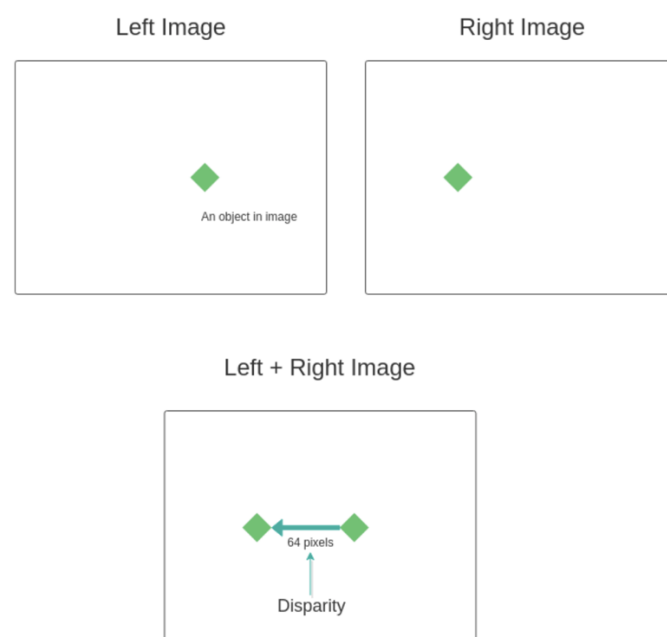


Figure 1.1: Disparity map creation combining Left and Right images.

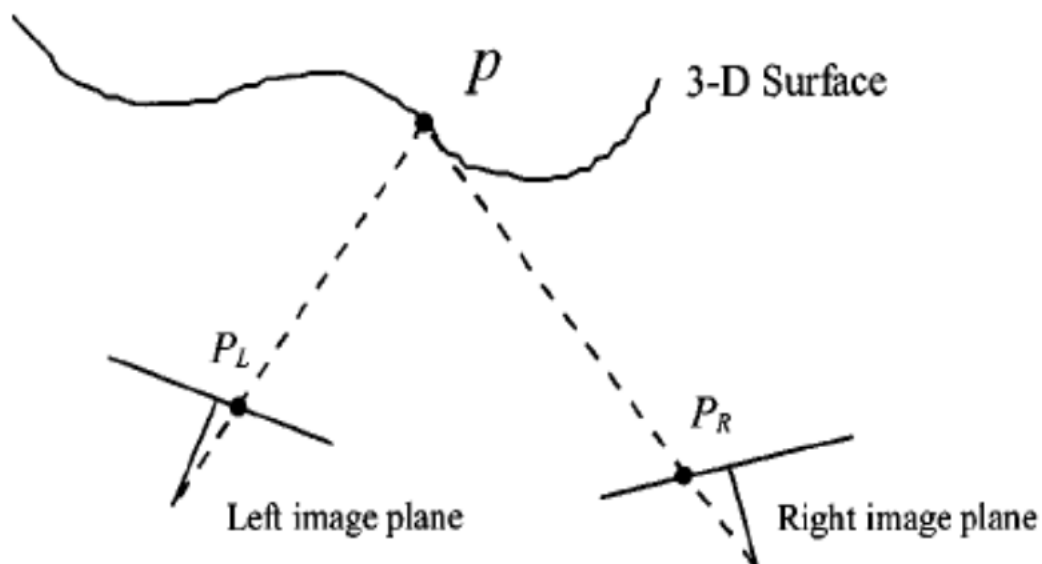


Figure 1.2: Basic concept of the stereo vision.

The concept is that an arbitrary point P of the surface is projected in different locations on two stereo images, left P_L and right P_R . Then, knowing the geometry of the imaging process, it is possible to estimate the 3D position. In particular, the position of P can be calculated knowing the intrinsic and extrinsic parameters of the cameras, such as the focal length and the position of the two cameras.

Once the disparity map is produced, the triangulation phase can be performed and the DTM can be produced.

1.1.2 Stereo camera overview

When in a mission we observe a planet, it is important to have the topography measurements of the surfaces. We can gain the global and local shape of the planets in two ways:

- Exploiting Laser Altimeter data.
- Using a photogrammetric procedure.

The first one provides data in the form of profiles along ground tracks, and it is very accurate, it suffers of problems in terms of spatial resolution and positional error. At the same time from the second one, that acquires stereo images, it is possible to obtain the Digital Terrain

Models (DTM) with flexibility and a better performance. From these models it is possible to gain information of the topography of the surface that allows to plan spacecraft missions and even missions with landers.

To obtain the three-dimensional reconstruction, it is necessary to have overlapping frames in order to achieve stereo-images pairs of the same area. Because of the parallax effect and the different acquisition points of view, we have the effect called disparity. At the state of art there are many technologies that allow to provide high quality stereo cameras for space mission, even if the 3D analysis and interpretation of planet surfaces require more studies. To provides good stereo-images, it is also important to keep in mind that photogrammetry must adapt to the resources allocated to the instrument and to the orbit. So, the optical design and the sensor must satisfy the requirements. Particularly when we talk about the constraints imposed by the spacecraft orbit and by the imaging geometry, we refer to the accuracy and precision of the stereo reconstruction.

The main elements we must keep in mind for a good accuracy are the baseline and the stereo-angle; the latter, measured between two images, can be fixed by mechanical constraint in the along-track mode or by the geometry of the photogrammetric block. It is useful sometimes to get a multi-images block of the same area to have a stable processing and more accuracy.

Nowadays there are two kinds of camera for the stereo-acquisition, the **pushbroom** cameras and the **frame-based** ones. The first kind are characterized by a capture of the images making the scan of one line at time, creating so an image meanwhile the satellite moves along the trajectory. For the second kind, the totality of the image is captured in one time, meaning that all the pixels are acquired simultaneously. The pushbroom camera is the most used and the simplest in terms of configuration; it is obtained involving two independent cameras tilted to gain the desired stereo-angle. In other cases, such us NAC camera on board of LRO, the two tilted optical channels have the scope to obtain an extended field of view and not a stereo reconstruction; the stereo pairs are created by images captured in two consecutive orbits. The left and right cameras have a small overlapping area but can still provide three or four models from which it is possible to obtain the elevation measurements. A different configuration is the one adopted for the HRSC camera; it consists of a multi-line pushbroom stereo camera that can provide up to 5 panchromatic multi-angle observation during Mars' orbit. The capability of simultaneous acquisition of stereo imagery avoids the changing of imaging conditions during successive orbital passes, arriving to a ground

resolution of 10 m/px. Another kind of configuration is the one used for the STC on board BepiColombo mission, we are talking about a push-frame configuration. This is a solution between the two-cameras and the single-camera; the detector is a bi-dimensional array, and the operations allows a variable overlap of the imaged regions in the along-track direction, increasing the image matching accuracy.

In the case of HYPSSOS the adopted solution consists in using a single stereo camera with two sub-channels, allowing to respect the constraints imposed by the mission and the stability of the camera. The final purpose will be to integrate to the stereo camera the characteristics of a hyper-spectral imager on both channels.

1.2 HYPSSOS instrument concept

HYPSSOS concept start from the idea of joining in one instrument a pushbroom stereo camera system and a spectrograph. To design the instrument, everything started from the classic configuration of a nadir pointing satellite and a pushbroom system of acquisition with his forward and backward channels tilted by an angle of $\pm 20^\circ$ with respect to nadir along the flying direction. This approach can be used for civilian applications to realize low-coast Earth 4D-observing system, like nanosats or Unmanned Aerial Vehicles.

For the realization of the optical design, it has been considered the stereo camera of SIMBIO-SYS, which is the imaging system on board of the ESA BepiColombo mission to Mercury. The technology applied to HYPSSOS is to send two separate beams of light along the same optical path, and to get the two images on the same focal plane; thanks to this the mass and the total envelope resources are minimized. To provide the spectral information, we need to englobe the spectrograph in such a way that the two collected images are sent to the entrance slit of the spectrometer. In the following Figure 1.3 it is possible to see the ground projected FoV's of a standard pushbroom stereo camera for a nadir pointing satellite; the sub-satellite track indicates the spacecraft on-ground trajectory and his associated motion orientation: Channel 1 represents the forward FoV and Channel 2 the backward FoV.

HYPSSOS stereo camera optical system has been designed to provide a 90° rotation of the FoV's and bring them onto the spectrograph entrance slit, which has the function of field stop. In this way, the two FoV's are entering the spectrograph in different portions of the entrance slit and, by means of a suitable imaging spectrograph, the spectra of the two FoV's are obtained.

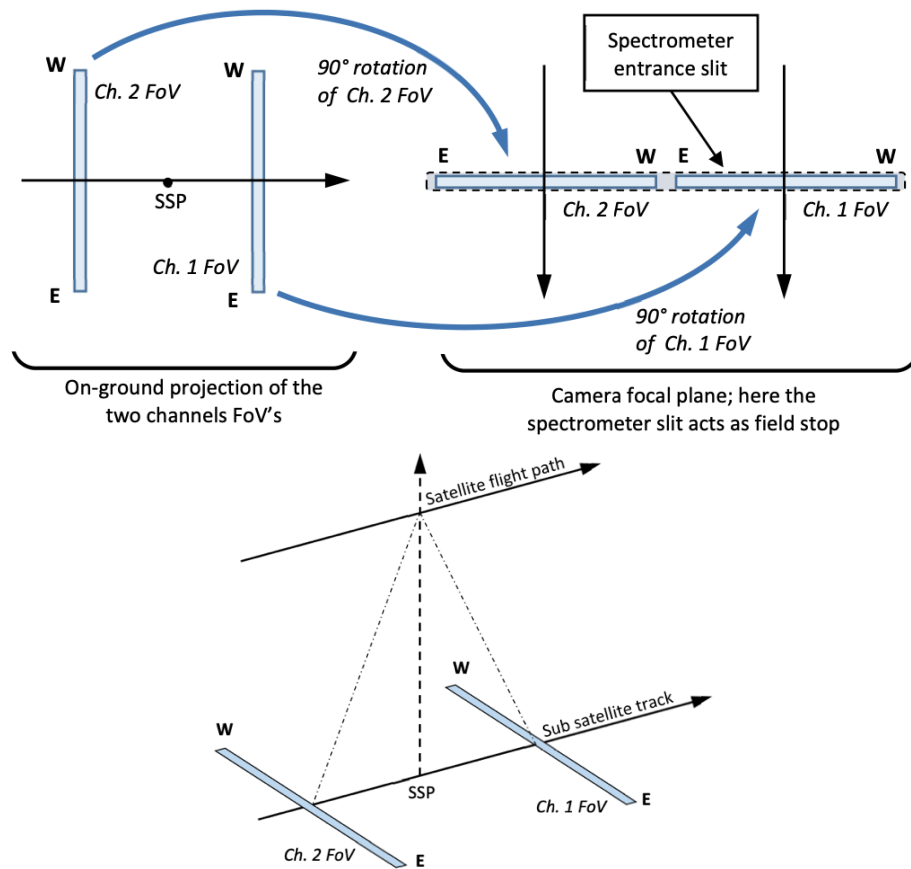


Figure 1.3: HYPSSOS' Field of View ground projection.

The advantage is that from the design concept it is possible to achieve all the information needed to have a 4D instrument at the same time:

→ from the two **stereo channel** we obtain the data needed to reconstruct the surface DTM.

→ from the **spectrometer** the spectral information for each point on the DTM.

From a primary analysis a good configuration can be obtained by using all-reflective telescope as stereo camera, in this way we avoid chromatic aberration, and the camera is wavelength independent. To validate the concept, we realized an instrument prototype.

1.2.1 Optical prototype

The optical configuration of the prototype is represented in Figure 1.4: a couple of flat mirrors collects the light from two different directions with an inclination of $\pm 20^\circ$ with the respect to the optical axis.

The two beams of light pass through two 45° tilted Schmidt- Pechan prisms¹ (SPP), which provide a 90° rotation of the FoV; in this way both lights path have in common the same optical elements after the SPP.

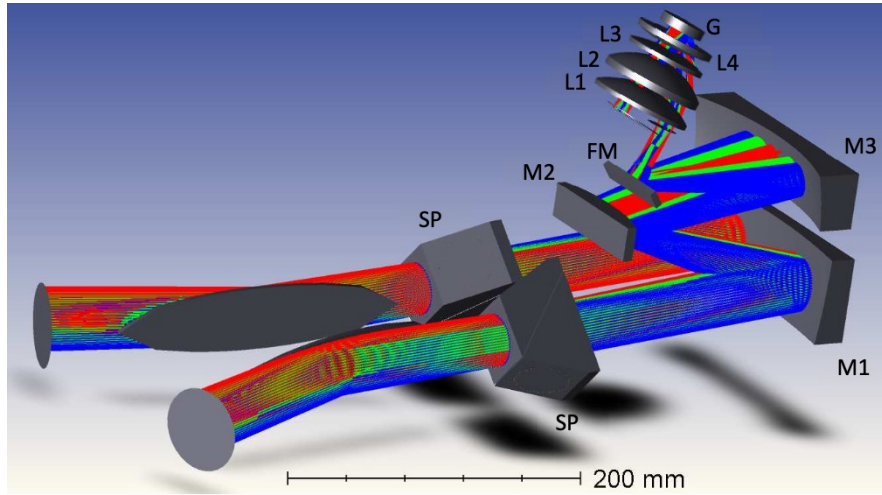


Figure 1.4: Optical layout of HYPSSOS prototype.

The light passes through a three-mirror anastigmatic (TMA) telescope (M1, M2, M3) and is folded by a flat mirror (FM) before reaching the focus. On the telescope focal plane, a slit defines the instrument FoV, as shown in Figure 1.3, with a separation between the two channels. Then the light passes through a double-pass imaging spectrometer composed by four lenses (L1, L2, L3, L4) and a concave reflection grating (G). As a final step a bidimensional detector collects the two separate spectra. In Table 1.1 all the principal parameters of the optical device are summarized.

Table 1.1: Principal parameters of HYPSSOS optical prototype.

M1	Radius of curvature: 467 mm Concave hyperbolic surface. conic constant: -1.474
M1-M2 distance	135 mm
M2	Radius of curvature: 141 mm Convex spherical surface
M1-M3 distance	135 mm

¹ The Schmidt-Pechan prism is a type of optical prism that rotate an image of 180°. Pechan design will invert or revert (flip) the image, depending on the orientation of the prism, but not both at the same time. The image's handedness is not changed by the Schmidt-Pechan.

M3	Radius of curvature: 117.5 mm Concave oblate ellipsoidal surface. conic constant: 0.198
M3-FM distance	225 mm
FM-focal plane distance	41.73 mm
Spectrograph entrance slit(s)	Two co-axial slits (22 μm x 8 mm each), 1 mm central separation.
L1	Meniscus R1: 562.7 mm; R2: 61.1 mm Thickness: 19 mm
L2	Plano convex R: 50.78 mm Thickness: 19 mm
L3	Plano convex R: 91.84 mm Thickness: 7.4 mm
L4	Meniscus R1: 104.4 mm; R2: 64.8 mm Thickness: 3.2 mm
G	Concave spherical Radius of curvature: 83.7 mm Ruling density: 678 mm^{-1}

1.3 Research objective

My thesis work is based on the calibration of HYPSSOS, and so there are two set of parameters that is necessary to find: intrinsic and extrinsic parameters. Intrinsic parameters are those parameters which are internal to the stereo camera such as focal length, principal point (\mathbf{P}), etc..., meanwhile extrinsic parameters are those parameters that allow as to know the location \mathbf{t} (translation vector) and orientation \mathbf{R} (rotation matrix) of the stereo camera with respect to an external coordinate system that we assume to be the coordinate system of the world.

So, starting from modelling and analyzing the pushbroom sensor, a Python code has been created. It was necessary to create simulated images of a bi-planar chessboard used as a target for the calibration; from this, using a Harris Corner detector, all corners of the chessboard representing the image coordinates (2D coordinates) have been detected. In a different way, using the geometry of the chessboard, also the object coordinates (3D coordinates) were extracted and using as an input both kind of coordinates for the pushbroom code, the camera matrix has been extracted. So, we can summarize all the parameters that are necessary dividing them into inputs and outputs:

INPUTS:

- 3D object coordinates of the checkerboard corners evaluated in the world reference system that consist of a high number of easily detectable, homogeneously distributed, and accurate reference points.
- 2D image coordinates that are semi-automatic points detected by using Harris corner detector algorithm.

OUTPUTS

- Focal length (f).
- Principal point (P).
- Starting point, Rotation matrix (R), Translation vector (T).

The last step has been to evaluate all the intrinsic parameters of the model and trying to check if the value obtained were close to the real ones.

Chapter 2: LITERATURE OVERVIEW AND METHODOLOGY

In this chapter I will firstly describe the hyperspectral imaging, the spectrum of application and the advantages of using it. Then I will give a short overview of the most used photogrammetric methods for camera calibration and a comparison between pinhole camera and pushbroom camera, dealing with the function and the mathematical model.

2.1 Overview of hyperspectral imaging

Remote sensing satellites can provide a great variety of data in panchromatic, multispectral, and hyperspectral modes.

Panchromatic images consist of only one spectral band, usually displayed as a gray scale image. The image can be interpreted as a black and white aerial photograph of the area. To interpret the panchromatic images is usually evaluated the radiometric information. Differently, a multispectral image consists of several bands of data, displayed one at time as a gray scale image or as a color composite image combining three bands at time. Multispectral sensor typically provides 15 bands meanwhile hyperspectral sensors can provide up to 100 spectral bands, which explains his name.

Hyperspectral imaging is a technology that has been mostly used in remote sensing to study the surfaces of different planet and other bodies in the solar system. The used instruments are imaging spectrometers, that can have a moderate spatial resolution ($\sim 1\text{-}30\text{ m}$) coupled with a regular sampling of a broad spectral range from ultraviolet ($\sim 0.35\ \mu\text{m}$) to thermal infrared ($\sim 12\ \mu\text{m}$) of the electromagnetic spectrum.

In this way we can say that imaging spectrometers are able to provide images with an extra dimension to conventional imaging: the spectral dimension. The physics behind the process is strictly connected with the principle of electromagnetic radiation and the way it interacts with different objects; we can have reflectance, absorption, and transmission of the incident flux. Hyperspectral remote sensing sample the spectral range of interest, viewing the surface as a series of contiguous and spectrally narrow image bands. We can see the function of the system in Figure 2.1: each spatial element of the image has an associated full resolution spectrum; the calibrated reflectance or emittance spectra collected will be compared with the one measured in the laboratory for different materials, in this way when it matches, we are able to identify the different materials composing the surface that we are observing.

Since today hyperspectral imagery has been used to detect and map a wide variety of materials and to detect surface properties. Other applications are the detection of topographical and geological features of a surface, the research of water sources and analysis of water quality, and vegetation biochemical and biophysical data detection.

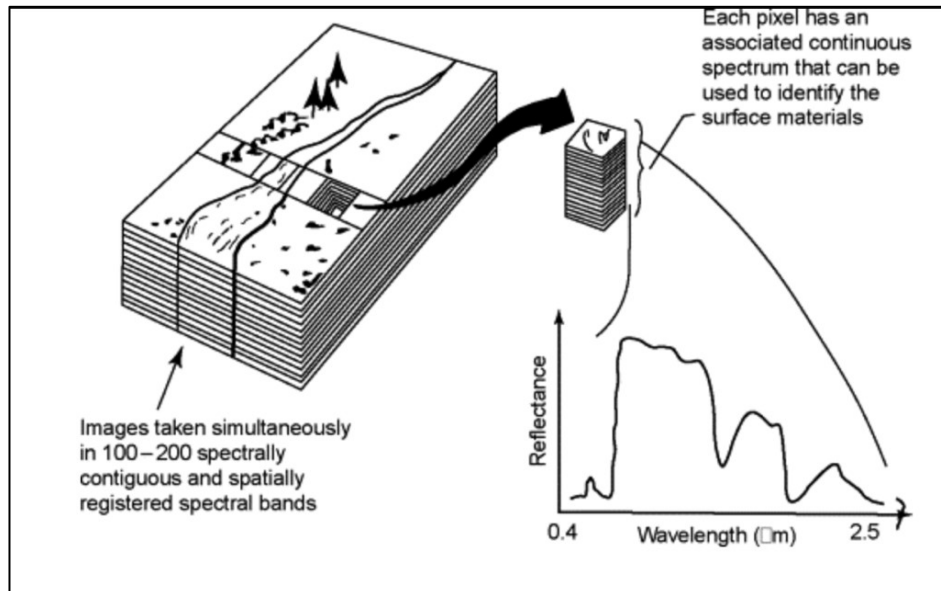


Figure 2.1: Example of hyperspectral imaging approach on Earth.

Hyperspectral imaging can give more benefits than multispectral as it covers a continuous spectral range without gaps allowing a uniform analysis of the mineralogical properties of a surface.

2.2 Camera calibration

The process due to evaluate all parameters of a camera necessary to give a correspondence between the 3D coordinates of an object in the real world and the image points (2D coordinates) is called **camera calibration**. The parameters can be divided into two categories:

- Intrinsic parameters, that are all the parameters that strictly depend on the camera that we are using (e.g., focal length).
- Extrinsic parameters, that allow us to have the correspondence between 3D and 2D coordinates.

2.2.1 Geometric parameters

The geometric parameters allow to understand the position of the image captured by the camera. They are in some way the connection between the points of the scene and the points on the image plane. They are the following parameters: the type of projection; the position and the orientation of the camera in the space (world); physics propriety of the matrix of the camera sensor; the distortion due to the optical design.

A common geometric model for a projection of a camera consists of a plane called **image plane** (π), with a **center of projection** (point O). The distance between this point O and π is called **focal length** (f).

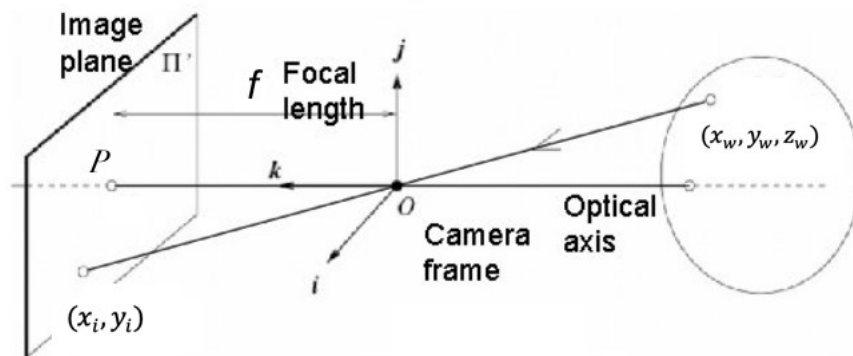


Figure 2.2: Representation of a camera projection.

We have the **optical axis**, that is the line perpendicular to the image plane that pass through the center of projection and intersect the image plane in the point called the **principal point** (P). If we want to get the projection of the point $(x_w, y_w, z_w)^T$, a generic world point, we must trace the line that pass-through P and O and the intersection with the image plane of this line will be the projection of the point in the image plane; in Figure 2.2 is represented by $(x_i, y_i)^T$.

2.2.2 Computer Vision coordinate system

The Computer Vision is the totality of the processes which leads us to have an approximate model of the 3D world starting from the 2D images. The real world is described by a series of equations that connect the image taken by the camera (one or more, it depends on the

model) to the coordinates of the scene that we want to observe. For this reason, we need first of all to decide a reference system for the camera, so usually is assumed that

1. We know the position of the camera system with respect to a reference system note, such as the world coordinate system for which we can easily decide the position of the center.
2. The coordinates of the image points of the camera reference system are connected to the pixel coordinates.

Starting from these two conditions, we need to know more parameters of the camera as possible. As said before the parameters can be divided in **extrinsic parameters**, that define the position, orientation of the camera system with respect to the world reference system; and **intrinsic parameters** that are those parameters that give the correspondence between the pixel coordinates of the image with respect to the camera reference system.

2.2.2.1 World to camera coordinate reference system transformation

The world coordinate system is a 3D reference system with an arbitrary origin. We can denote a point in this system like $\bar{x}_w=(x_w, y_w, z_w)$. Extrinsic parameters are used to correlate each other the reference system of the world to the reference system of the camera. This is easily done by means of a roto-translation. In fact, we can introduce:

- The **Translation vector** $T=(T_x, T_y, T_z)^T$, that represents the relative position between the two systems.
- The **Rotation Matrix** R

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

that is an orthogonal matrix which leads us to know how the two systems are each other oriented.

A point in the camera based reference system can be indicated like $\bar{x}_c=(x_c, y_c, z_c)$.

2.2.2.2 Camera to image reference system transformation

Intrinsic parameters can be defined as the set of parameters needed to determinate the optical, geometrical, and digital characteristics of our telescope, such as:

- The focal length f .
- The Field of view.
- Aperture.
- Resolution.

The focal length, as said before, can be defined as the distance between the optical center and the image plane (or plane of focus) that corresponds to the sensor.

The equations of the prospective projection are non-linear for the presence of z_c at the denominator

$$x_i = f \frac{x_c}{z_c} \quad \text{and} \quad y_i = f \frac{y_c}{z_c} \quad (2.1)$$

The third coordinate of the point projected in the image plane is equal to the focal length, and for the fact that it is a constant value, we must write the projected point like $\bar{x}_i = (x_i, y_i)^T$.

Introducing the coordinates of the **principal point** $P = (P_x, P_y)$, and (S_x, S_y) dimensions of the pixel, to get the transformation from camera coordinates of a point (x_i, y_i) to pixel coordinates (u, v) of the same point we need to evaluate the following equations

$$u = -(x_i - P_x)S_x \quad (2.2)$$

$$v = -(y_i - P_y)S_y \quad (2.3)$$

in which the sign minus is due to the opposite orientation of the x and y axis in the two reference systems. In conclusion the intrinsic parameters are f, P_x, P_y, S_x, S_y .

In many cases we need to consider also the optical distortion introduced by the optical lens which affects the images. The distortion is usually associated to an optical aberration that deforms and bends physically straight lines of the images transforming straight lines into curvy lines. This distortion is usually called curvilinear.

We can divide the optical distortion into two types:

- Barrel Distortion (or positive radial distortion), which leads to an inwards curve, particularly the extreme edges of the frame, in a shape of a barrel.

- Pincushion Distortion (or negative radial distortion), where lines are curved outward from the center.

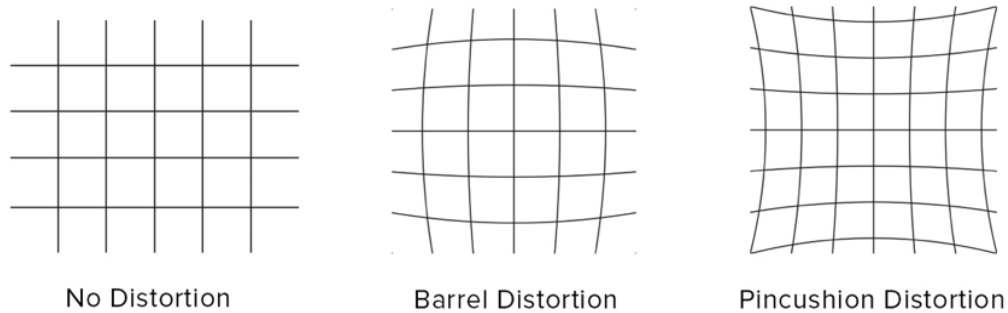


Figure 2.3: Representation of the image distortion: image without distortion, with Barrel Distortion and with Pincushion Distortion.

As it is possible to see from the Figure 2.3, the main distortion appears at the edges of the field of view, in other words we have that at the center of the image the distortion can be neglected but it increases when we move to the edges. These distortions for the pinhole on axis case, can be evaluated as a radial distortion and can be represented as a radial translation of the points of the image. Introducing K_1 and K_2 as parameters to evaluate the distortion we can write.

$$u = u_d(1 + K_1r^2 + K_2r^4) \quad \text{and} \quad v = v_d(1 + K_1r^2 + K_2r^4)$$

x_d and y_d are the coordinates of the distorted point and r is evaluated from $r^2 = x_d^2 + y_d^2$.

2.3 Camera models

2.3.1 Pinhole camera model

The pinhole is the most known and used camera model. If we design a simple camera system, which means a system that can record an image of an object or scene in the 3D world, we can think about a barrier with a small aperture (pinhole) between the object in the 3D world and an image sensor.

This means that now every point of the 3D object can emit many rays of light outwards; only the rays passing through the small aperture will reach the image sensor, as shown in Figure 2.4.

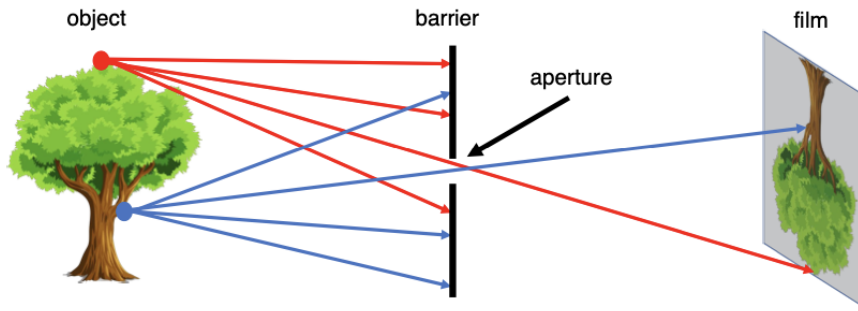


Figure 2.4: Schematic image of the pinhole camera model working.

If we map the points of the 3D object and the points in the sensor, we will obtain that the sensor gets exposed by an “image” of the 3D object due to this mapping.

If now we call \bar{x}_w a point in the 3D object visible to the pinhole camera, we have that this point will be projected on to the image plane, resulting in a 2D image point $\bar{x}_i = (u, v)^T$.

To project a world object point into the image plane, we need to firstly transform it into a camera point, using the following transformation

$$\bar{x}_c = R\bar{x}_w + T \quad (2.4)$$

Where R represents the rotational matrix and T the vector of the translation. Writing this equation in a matrixial way obtaining the extrinsic camera matrix 3x4 called \mathbf{E} .

$$\mathbf{E} = [R|T] \quad (2.5)$$

To provide a relationship between the camera points and the 2D image points, it is possible to define the intrinsic parameters matrix \mathbf{K} such that \bar{x}_i is proportional to \mathbf{K} . Defining

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

We can write

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \bar{\mathbf{x}}_c = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (2.6)$$

where u and v are the image coordinates and w is the factor used to normalize the vector $\bar{\mathbf{x}}_i$. Meanwhile f_x and f_y represents the focal length in terms of the x and y pixel dimension; P_x and P_y are the principal point coordinates.

The camera matrix \mathbf{M} (3x4) results from the product of matrix \mathbf{K} and \mathbf{E} . Finally, we obtain

$$\mathbf{M} = \mathbf{KE} \quad (2.7)$$

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2.8)$$

2.3.2 Pushbroom camera model

The pushbroom acquisition system is a commonly used sensor in satellite imagery. Modelling and analyzing the pushbroom acquisition is difficult and computationally intensive for the fact that we must combine the motion of the orbit of the satellite with respect to the rotating Earth and the non-linearity of the model that involves orbital dynamics. There are many mathematical models that has been studied to analyze the acquisition, models of varying complexity and accuracy. The main approaches used for high resolution satellite imagery are based on rational polynomials functions, affine projections, and direct linear transformation DLT.

In this work, I have studied the Gupta and Hartley model (Gupta R. & Hartley R., 1997), and I have implemented the estimation system in a Python code,

This model allows to give a very accurate results even if it is a simplified model. The key assumptions are:

- The sensor array is travelling in a straight line and constant velocity with respect to the world.
- The sensor orientation, and so the view plane, is fixed all over the image acquisition duration.

Before estimating the intrinsic and extrinsic parameters of the camera it is necessary to evaluate the mathematical model for the pushbroom system that, due to the system of acquisition, will be different to the one of the pinhole case.

The pushbroom system works thanks to the movement of the structure, projecting on the camera focal plane a linear scan of the ground surface while the platform travels along the path. The plane defined by the linear array of sensors and the optical center is called instantaneous view plane or simply **view plane**.

This optical system is mounted on the satellite and when the satellite moves, as we can see in Figure 2.5 the view plane scan a region of space keeping his direction almost perpendicular to the direction of motion (X axis).

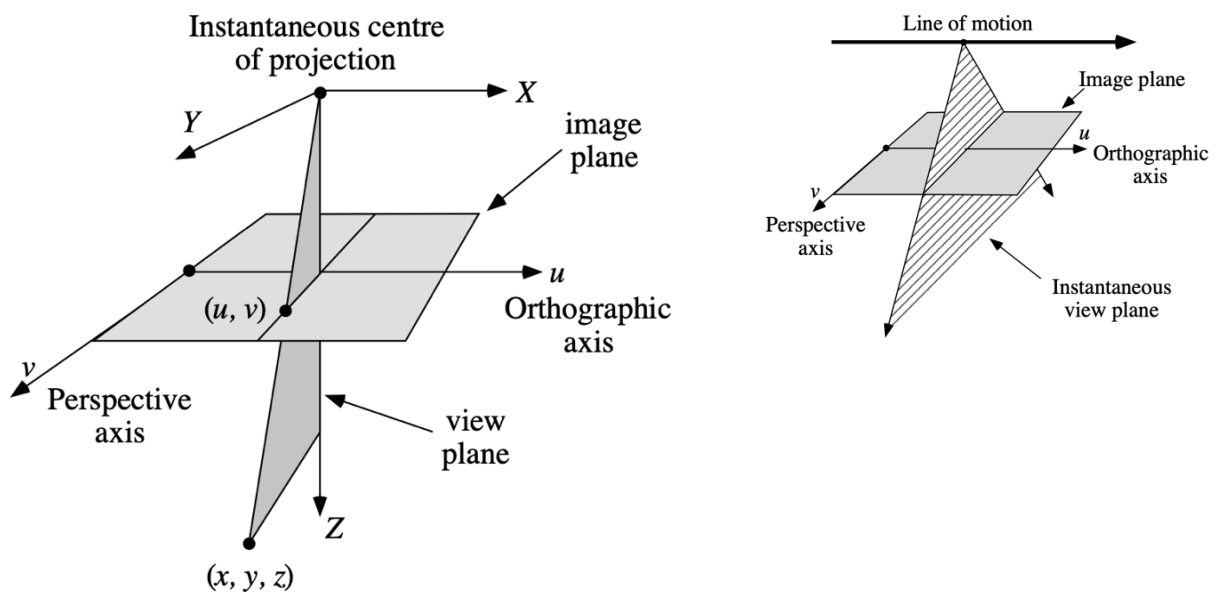


Figure 2.5: Schematic projection of the pushbroom camera model acquisition system.

This camera can be seen as a perspective camera that moves along a linear trajectory in space with constant velocity and fixed orientation. In this way the camera is constrained so that at any moment in time it images only the points lying in the view plane whose orientation is fixed. At any time, a 2D projection of the view plane into an image line takes place since the view plane scans the whole area of interest.

So, we can say that the coordinate u represents the time the arbitrary point \bar{x}_w is imaged, and v is the projection of the point on the image line. Considering an orthogonal coordinates frame attached to the moving camera with the origin of the coordinates system coincident with the center of projection, we have that the Y axis lies in the view plane parallel with the focal plane; the Z axis lies in the view plane perpendicular to the Y axis and the x coordinate is perpendicular to the view plane. In this way x , y and z axis form a right-handed coordinates frame.

2.3.2.1 Camera matrix

Starting from a 2D projection of a point with $(0, y_c, z_c)$ coordinates with respect to the camera frame we can write for the 1D projection of the v coordinate

$$\begin{bmatrix} wv \\ w \end{bmatrix} = \begin{bmatrix} f & p_y \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_c \\ z_c \end{bmatrix} \quad (2.9)$$

With f the focal length, P_y the principal point offset in Y direction and w is a scale factor.

To simplify the analysis, it is possible to consider a fixed camera and a world moving, instead of a stationary world and a moving camera. In this way a point in the space at the instant t will be represented as $\bar{x}(t) = (x(t), y(t), z(t))$. Assuming $-\mathbf{V} = -(V_x, V_y, V_z)^T$ the velocity vector of the point with respect to the camera (with the minus sign representing that the velocity of the camera with respect to the world is \mathbf{V}) and t_{im} the instant in which the moving point in space crosses the view plane, we can write the 2D projection (u, v) , where $u = t_{im}$:

$$\begin{bmatrix} u \\ wv \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_{im} \\ y_{im} \\ z_{im} \end{bmatrix} \quad (2.10)$$

The coordinates of the moving point \bar{x} as a function of the time can be written as

$$\bar{x}(t) = x_0 - tV = (x_0, y_0, z_0)^T - t(V_x, V_y, V_z) \quad (2.11)$$

Where $\bar{x}_0 = \bar{x}(t = 0)$.

At the time $t_{im} = x_0/V_x$, the point \bar{x} will be at the position $(0, y_{im}, z_{im})^T$, because the view plane is at the coordinate $x = 0$. This can be represented by

$$(0, y_{im}, z_{im})^T = (0, y_0 - x_0 V_y/V_x, z_0 - x_0 V_z/V_x)^T$$

Or like

$$\begin{bmatrix} t_{im} \\ y_{im} \\ z_{im} \end{bmatrix} = \begin{bmatrix} 1/V_x & 0 & 0 \\ -V_y/V_x & 1 & 0 \\ -V_z/V_x & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (2.12)$$

Combining equation 2.10 and equation 2.12 we obtain the final projection (u, v) on the image plane of $\bar{x}(t)$ in terms of the observed point in the camera reference frame at time $t = 0$.

$$\begin{bmatrix} u \\ wv \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/V_x & 0 & 0 \\ -V_y/V_x & 1 & 0 \\ -V_z/V_x & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (2.13)$$

Now, we must consider that generally the coordinates of the points are known in terms of some external fixed coordinate system $(x_w, y_w, z_w)^T$. Between the external coordinate system and the camera-based we have this relation:

$$\begin{aligned} (x_0, y_0, z_0)^T &= R \left((x_w, y_w, z_w)^T - (T_x, T_y, T_z)^T \right) \\ &= (R | -RT) (x_w, y_w, z_w, 1)^T \end{aligned} \quad (2.14)$$

With $\mathbf{T} = (T_x, T_y, T_z)^T$ which represents the translation vector (position of the camera at time $t=0$ in the external coordinate system), and \mathbf{R} is the rotation matrix. From this, we can finally write

$$\begin{aligned} \begin{bmatrix} u \\ wv \\ w \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/V_x & 0 & 0 \\ -V_y/V_x & 1 & 0 \\ -V_z/V_x & 0 & 1 \end{bmatrix} (R | -RT) \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \\ &= \mathbf{M} (x_w, y_w, z_w, 1)^T \end{aligned} \quad (2.15)$$

M is the camera matrix (or reprojection matrix) which leads us to have the final correlation between the 3D world coordinates and the 2D image coordinates.

It is important to say that we have two different transformations along the two directions, in particular:

- Perspective projection along the v direction.
- Orthographic projection along the u direction.

The Orthographic projection does leave parallel lines parallel, and it preserves relative distance between objects. The perspective projection doesn't preserve distances or angles, and parallel lines no longer remain parallel.

The camera matrix M for a linear pushbroom sensor can model translation, rotation and scaling of the 3D world coordinates as well as translation and scaling of the 2D image coordinates. It cannot take in account of the rotation of the image plane. The linear pushbroom camera matrix has in total 11 degrees of freedom: the first row, m_1^T , has 4 degrees of freedom; the other two rows, m_2^T and m_3^T , have 7 degrees of freedom.

The analysis of equation 2.15 reveals that multiplying M by a constant factor k is like multiplying the image coordinate u per k , while the coordinate v remains unchanged. This means that the last two rows of M can be multiplied by a factor k without changing the mapping.

It is also important to define the way we can determinate where the points are set with respect to the camera. To understand if the points are in front of the camera, we must analyze the value of the parameter w . Remembering that the positive z axis is directed in a way that the points in front of the camera have a positive z coordinate, the mapped point should also have $w > 0$.

For the fact that, as said before, we can multiply the last two rows of M by the factor k , if k is negative, then the points will not lie anymore in front of the camera because w will be negative too.

2.3.2.2 Intrinsic parameters matrix

M can be written as $M = (K | -KT)$ for a non-singular matrix K (3x3).

$$K = \begin{bmatrix} 1/V_x & 0 & 0 \\ -k(fV_y + p_v V_z)/V_x & kf & kp_y \\ -kV_z/V_x & 0 & k \end{bmatrix} R = LR \quad (2.16)$$

L is the matrix containing all the intrinsic parameters of the system, in particular

$$f = L_{22}$$

$$p_y = L_{23}$$

$$V_x = 1/L_{11}$$

$$V_z = -L_{31}/L_{11}$$

$$V_y = -(L_{21} + p_y L_{31})/(fL_{11}).$$

The mathematical method applied to solve the pushbroom Gupta and Hartley model is the DLT method, usually applied for the pinhole acquisition system.

Starting from eq. 2.15 we can write

$$u = m_1^T \bar{x}_w$$

$$wv = m_2^T \bar{x}_w \quad (2.17)$$

$$w = m_3^T \bar{x}_w$$

Combining the last two equations, we eliminate the factor w leading to

$$u = m_1^T \bar{x}_w$$

$$v m_3^T \bar{x}_w = m_2^T \bar{x}_w \quad (2.18)$$

If we have at least 11 sets of points, meaning a set of 11 correspondences between world coordinates (x_w, y_w, z_w) and image coordinates (u, v) , we can evaluate the matrix M . In particular we know that the first row of M , m_1^T rely only the u coordinates and it can be evaluated given four ground control points. Meanwhile the second and the third rows of M ,

m_2^T, m_3^T , need seven ground control points to be evaluated and rely only the v coordinates of the points. These 11 sets of points represent the 11 degrees of freedom of the system:

- 3 degrees of freedom related to the position.
- 3 degrees of freedom of orientation.
- 3 degrees of freedom related to the velocity.
- the focal length.
- the v -offset.

The \mathbf{M} matrix will be evaluated using the Direct Linear Transformation method, which is described in the following subchapter. Unfortunately, there is not a unique solution for the system of equations. Hence to determinate the right projective matrix, we need first to evaluate the position of the points with respect to the camera. For this, when we have evaluated the matrix \mathbf{M} , we chose one point \bar{x}_i and evaluate the product $(u_i, w_i v_i, w_i)^T = \mathbf{M}(x_w, y_w, z_w, 1)^T$; if w_i is negative than the last two rows of \mathbf{M} must be multiplied by -1. In this way all points are in front of the camera. In doing so all the parameters of the camera will now be evaluated using a variation of the standard QR factorization method², explained later. It is important to say that some parameters of the camera are for sure known from the manufacturing, for example, the focal length or the principal point offset. But unfortunately, these values cannot be inserted in the DLT method, and they can be used only in a second time to correct the values that we obtain from the resolution.

We start obtaining the parameters of the camera by determining the initial position at $t = 0$. Matrix M can be written as $M = (K \mid -KT)$ for a non-singular matrix K (3x3), that correspond to the left-hand 3x3 block of M . So, we can easily determine the translation vector \mathbf{T} by solving the linear equation

$$-K\mathbf{T} = -m_3^T \quad (2.19)$$

² The QR factorization, is a decomposition that allows to express a matrix A as a product between two separate matrixes, Q and R. Where Q is an orthogonal matrix and R is an upper/ right triangular matrix. This method is often used to solve the linear least squares problem.

\mathbf{K} is in the form written in eq. 2.16, and it is now known, so we need to evaluate \mathbf{L} and \mathbf{R} . The matrix of rotation \mathbf{R} is obtained by a sequence of rotation matrixes R_z, R_y, R_x . Denoting with $c = \cos(\theta)$, and $s = \sin(\theta)$

$$R_z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix};$$

$$R_y = \begin{bmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s & 0 & c \end{bmatrix};$$

$$R_x = \begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Now to evaluate the values of θ , we impose equal to zero the elements L_{12}, L_{13}, L_{32} . Defining the sequence of the intermediate products exploiting the QR factorization leads us to

$$P = L R_z, Q = L R_z R_y, K = L R_z R_y R_x.$$

Working on this we obtain

$$\theta_z = \arctan2(K_{12}, K_{11})$$

$$\theta_y = \arctan2(P_{13}, -P_{11})$$

$$\theta_x = \arctan2(Q_{32}, -Q_{33})$$

The rotation matrix is not uniquely determinate so we must check if L_{22} and L_{33} are both positive, if not it would contradict that $K = L_{33} > 0$ and the geometric requirement that $L_{22}/L_{33} = f > 0$. So, in case these quantities are negative, we must correct as follow: if $L_{33} < 0$ then we should apply a further rotation around the Y axis, R_y , of an angle of π ; if also $L_{22} < 0$, then it is needed to apply another rotation around the Z axis, R_z , again by an angle of π .

2.3.2.3 The DLT method

The Direct Linear Transformation (DLT) is the most common method applied to solve the 3D position of an object, and so to find the camera calibration. Normally this method is applied to the pinhole sensor, but with some changes we can apply it to the pushbroom one.

First of all, we have to consider n points (x_w, y_w, z_w) in the 3D world reference system and their correspondent 2D n points (u, v) in the image reference system; the correlation between these points must be as precise as possible because the correlation hardly affects the solution. The scope of the DLT is to find the matrix \mathbf{M} starting from the linear correlation between the coordinates.

The correlation between the coordinates is given by $(u, v, w)^T = \mathbf{M} (x_w, y_w, z_w, 1)^T$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

from which we can obtain the equations 2.17; that set of equations can be rewritten in the form of

$$[\mathbf{A}]\{\mathbf{q}\} = \{\mathbf{b}\}. \quad (2.20)$$

with \mathbf{A} the matrix $2n \times 11$ that represents the vectors spanning a $2n$ dimensional space; \mathbf{b} the $2n \times 1$ vector containing the u and v image coordinates in pixel; lastly \mathbf{q} is the 11×1 vector containing the 11 elements of M . The twelfth element m_{34} can be considered equal to 1 because the last two rows of M are free scaling, and the scale factor can be recovered later (k).

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_1 x_1 & -v_1 y_1 & -v_1 z_1 \\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & x_n & y_n & z_n & 1 & -v_n x_n & -v_n y_n & -v_n z_n \end{bmatrix} \begin{Bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ \vdots \\ \vdots \\ m_{34} \end{Bmatrix} = \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ \vdots \\ \vdots \\ v_n \end{Bmatrix} \quad (2.21)$$

The vector \mathbf{q} is the only unknown but due to pixel error \mathbf{b} will lie outside of the hyperplane spanned by \mathbf{A} . To solve the system at the least squares solution we have to use the Penrose solution, that allows to project the vector \mathbf{b} into the hyperplane:

$$\mathbf{q} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (2.22)$$

Hence, obtained \mathbf{q} we can rewrite the vector in the matrixial form adding the element m_{34} and we have the matrix \mathbf{M} .

Below the implementation of the DLT modified method extracted from the Python code
Construction of the matrix \mathbf{A} .

```

xt = (np.transpose(imjpoints)[0:len(imjpoints)])
Xt = (np.transpose(objpoints)[0:len(xt)])
one = np.ones((len(Xt), 1))
zeros = np.zeros((len(Xt), 1))
Xt_1 = np.hstack((Xt, one))
X1t = np.empty((len(Xt), 1))
X0t = np.empty((len(Xt), 1))
xXt = np.empty((len(Xt), 3))
for i in range(0, len(Xt)):
    xXt[i] = -xt[i][1] * Xt[i]
xXt_1 = np.hstack((xXt, one))
zero4 = np.array((0, 0, 0, 0))
zero3 = np.array((0, 0, 0))
A = np.array((len(Xt)*2, 11))
for i in range(0, len(Xt)):
    C = np.hstack((Xt_1[i], zero4, zero3))
    B = np.hstack((zero4, Xt_1[i], xXt[i]))
    C = np.reshape(A, (1, 11))
    B = np.reshape(B, (1, 11))
    if i == 0:
        A = np.vstack((C, B))
    else:
        A = np.vstack((A, C, B))

```

Definition of the vector \mathbf{b} , composed by the alternate u_i, v_i coordinates

```

b = np.empty((len(Xt)*2, 1))
for i in range(0, len(xt)):
    a = (xt[i, 0])
    c = (xt[i, 1])
    if i == 0:
        b = np.vstack((a, c))
    else:
        b = np.vstack((b, a, c))

```

Final estimation of the projective matrix \mathbf{M} , applying the pseudoinverse transformation.

```

#penrose pseudoinverse
A_Pseud = np.linalg.pinv(A.astype(float))
m = A_Pseud.dot(b)
mm = np.append(m, 1)
M = mm.reshape((3, 4))
print('matrice M', M)

```

Chapter 3: STUDY SETUP

In this Chapter, the arrangement of the laboratory setup able to simulate the acquisition mode of HYPSSOS is described; the photogrammetric procedures used to calibrate the system, both through a mathematical modelling (implemented in MATLAB) and with ray-tracing simulation process. Then, an automatic procedure to capture the corners of the chessboard using the Harris corner detector is described. Finally, the analysis of the results obtained from the running of the implemented pushbroom model estimation is discussed and summarized in the conclusions.

3.1 *Experimental setup*

The setup that will be used to validate the HYPSSOS concept and to generate the data is similar to the one used for the BepiColombo-STC stereo validation.

The experimental set-up for HYPSSOS takes many elements from the one used for the stereo camera STC of SYMBIO-SYS, on board of the BepiColombo mission. Some changes have been introduced in the setup due to the different acquisition mode of HYPSSOS (pushbroom with respect to the push frame).

To perform the geometric calibration of the HYPSSOS system on the optical bench in laboratory (exploiting many elements of the Stereo Validation System Setup (SVS) (Figure 3.1) of STC), we arranged a setup composed by:

- two rotational stages (devoted to reproducing the stereo angle of HYPSSOS),
- a translator that allows the acquisition in push-broom mode,
- a collimator lens (achromatic doublet with nominal focal length of 1m),
- an illumination source (halogen lamp)
- a biplanar calibration gauge (chessboard) useful for the calibration algorithm based on known reference points with corresponding projections in the image.

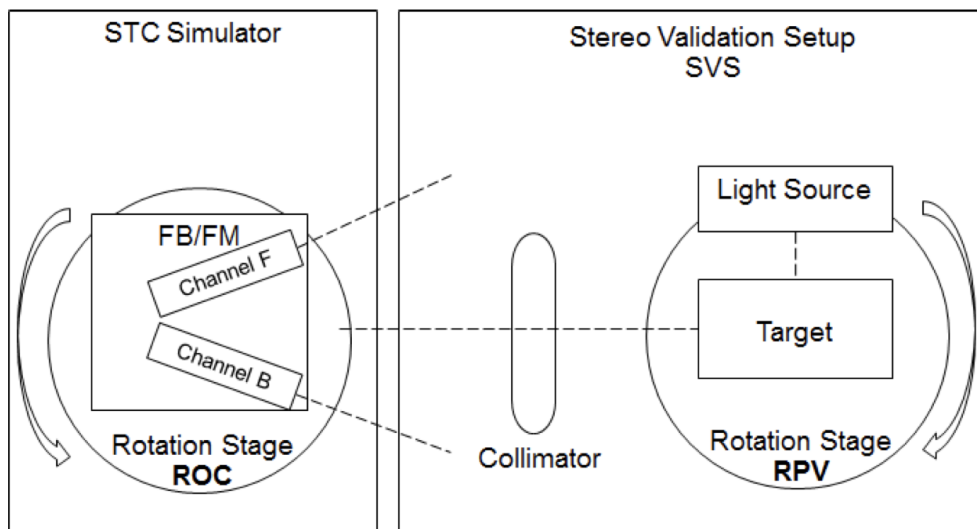


Figure 3.1: Schematic image of the laboratory set-up used to simulate the acquisition system of HYPSSOS.

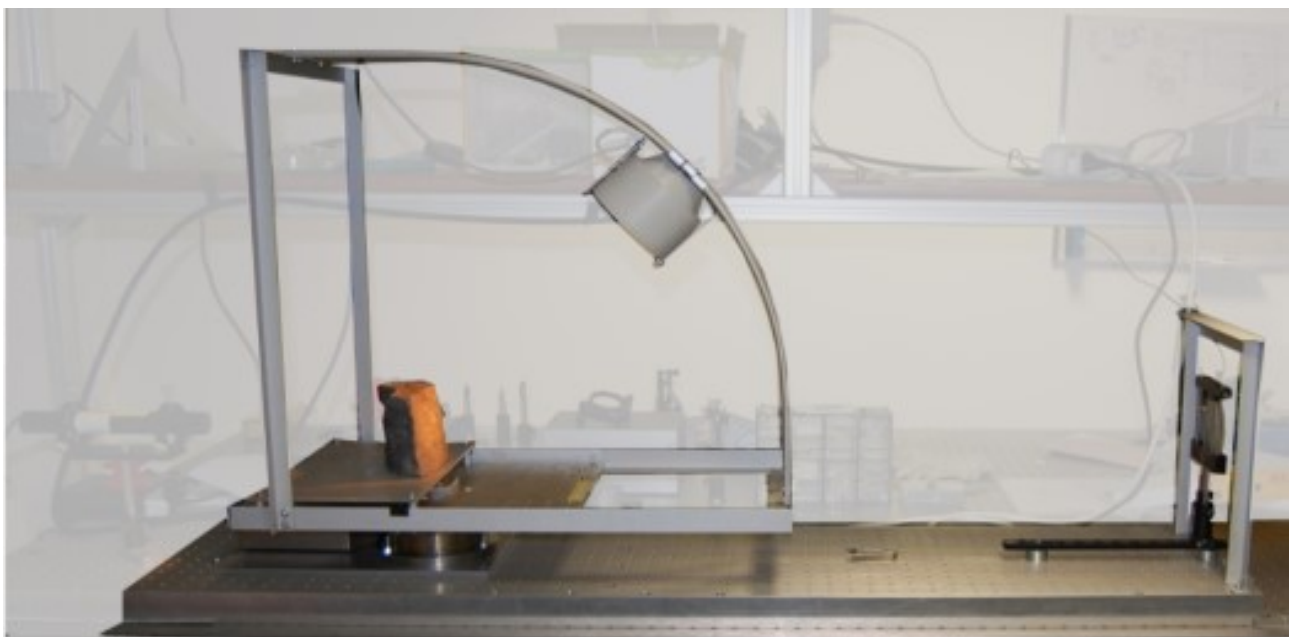


Figure 3.2: Stereo Validation setup (both for STC and HYPSSOS).

During the geometric calibration the **target** will be a bi-planar chessboard, as we can see in Figure 3.3; then it will be replaced by stones like Anorthosite and Basalt useful to simulate a planetary surface. This reference gauge, devoted to the photogrammetric calibration, has been designed to provide a high number of easily detectable points, homogeneously distributed on the complete scanning acquisition.

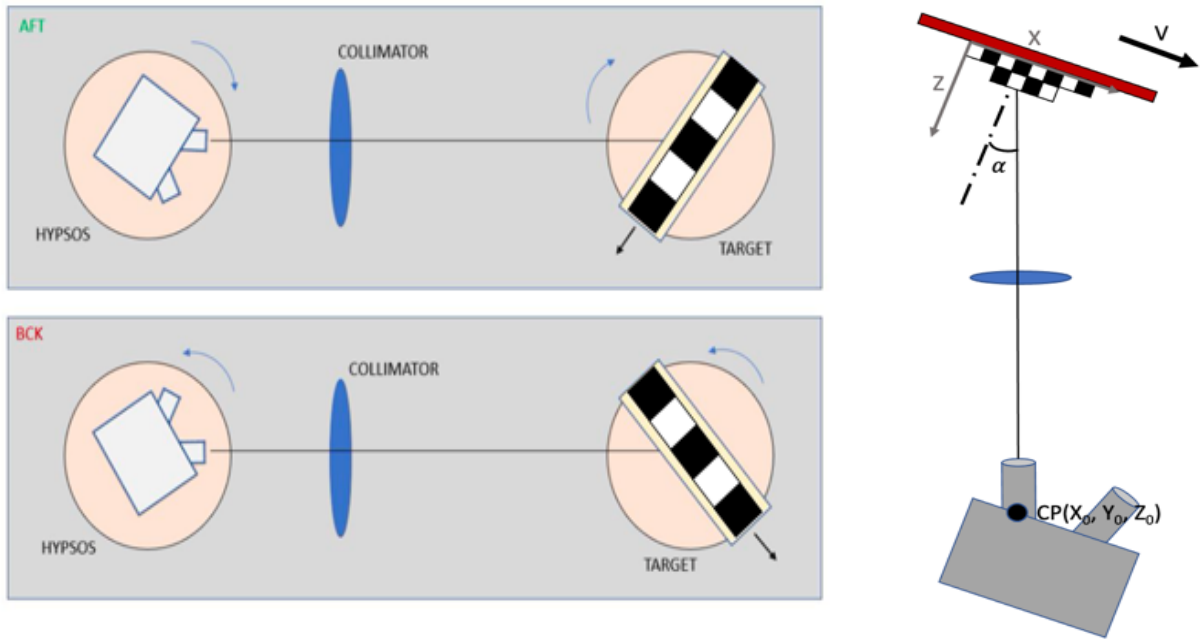


Figure 3.3: The scheme of HYPSSOS acquisition during the calibration.

The target is mounted on a translator that can simulate the acquisitions of the pushbroom model, moving forward the target.

It is important to fix a central point (CP in figure 3.3) using a fixed mark in the field of view of the stereo camera.

So, in laboratory the target is moved relatively to the fixed instrument meanwhile when we are in flight the target will be stationary relatively to the satellite that is moving along the orbit. We can see also the light source mounted over a curve rail that allows to maintain a constant lighting condition and avoids the shading effects, that introduce an error in the image correlations phase.

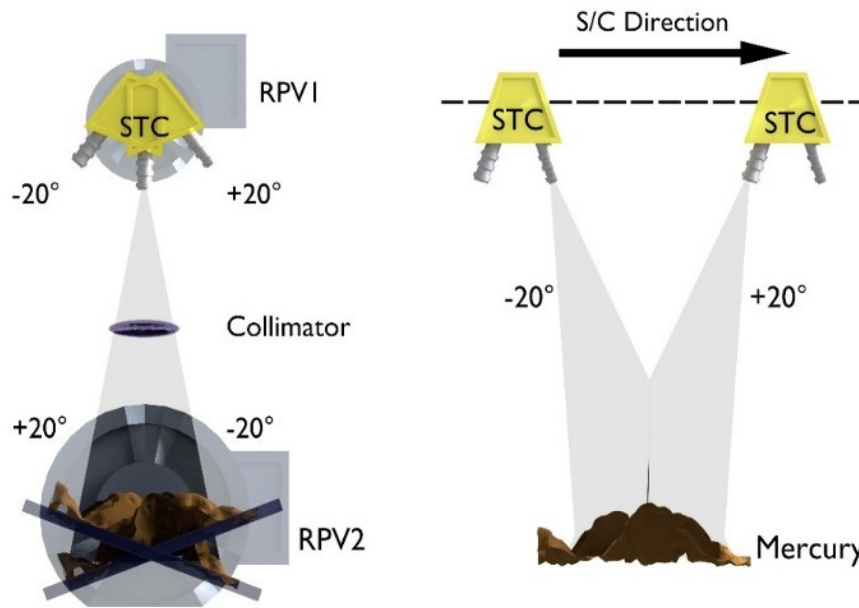


Figure 3.4: A schematic view of STC/HYPSOS acquisition system on a simulated Mission.

Figure 3.4 shows the acquisition of the two channels tilted by an angle of $\pm 20^\circ$.

The aim of the photogrammetric pipeline foreseen for HYPSOS, is the reconstruction of the scene with the generation of the Hyperspectral DTM of the target, meanwhile the spectral information is exported as orthorectified image cubes with the pixel size of the DTM.

The photogrammetric calibration is essential for the determination of the projection matrices that describe the projection system for the final triangulation that derives the 3D information from the stereo images.

The definition of the stereo camera model is mandatory, and the photogrammetric procedures for the estimation of the intrinsic and extrinsic parameters are also needed. Thanks to the acquisition with the stereo validation setup, all the calibration steps are permitted.

3.2 Bi-planar calibration gauge

In order to proceed with the photogrammetric calibration of the system, we decided to use a bi-planar laminated chessboard that provides high number of easily detectable corners, homogeneously distributed on the scanning acquisition, we decide to use a bi-planar laminated chessboard. Both the planes are paper printed with a regular chess pattern. A

first design of the calibration gauge that considers a good number of corners has a lower plane with a dimension of $4 \times 4 \text{ cm}^2$ and the upper plane of $2 \times 2 \text{ cm}^2$ with a height of 0.5 cm ; every square of the chessboard has a dimension of $0.2 \times 0.2 \text{ cm}^2$.

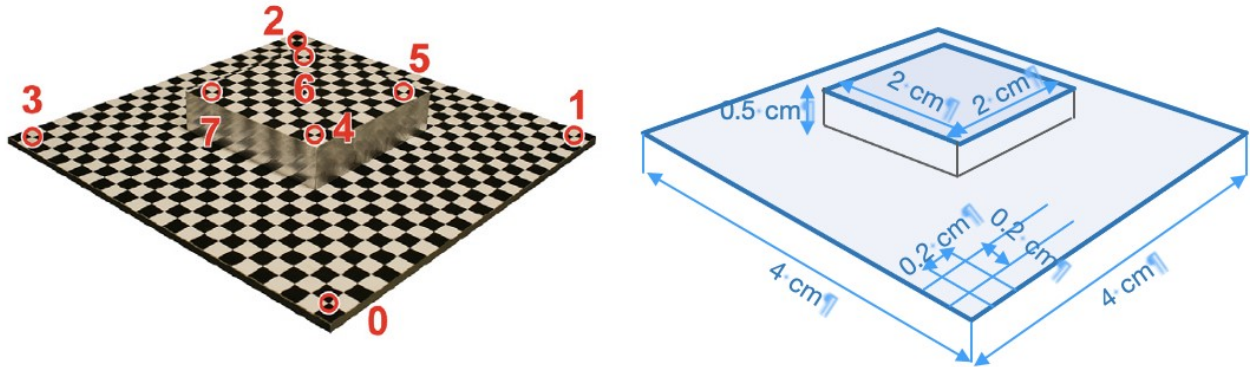


Figure 3.5: Bi-planar calibration gauge used for the camera calibration.

The DLT method requires a high resolution of the chessboard, because even a small error in the chessboard design could induce a large error in the resulting estimation. To evaluate the error introduced to the calibration by the chess squares design, we implemented a program in MATLAB that can provide the variations in the focal length estimating the error induced by the corner detection process.

3.2.1 Image simulation

In preparation of the actual image data, a simulated benchmark has been created exploiting a powerful Airbus ray-tracing software (SurRender, Brochard & al., 2018) that provides a useful set of synthetic images. Furthermore, a proper numerical simulation (in MATLAB code) based on the geometrical characteristics of the calibration setup has been used for further analysis.

Concerning the generation of the synthetic images thanks to SurRender software, we started with reproducing in the 3D modelling environment the 3D mesh of the designed reference gauge (Figure 3.5). Then defining the geometry of the SVS and creating a virtual stereo camera with the main characteristics of HYPSONOS, we were able to reproduce with high fidelity the line scanning, that, once mosaicked, have provided a good synthetic dataset for

the validation of the method for the camera model estimation. The entire workflow is summarized in the block diagram in Figure 3.6.

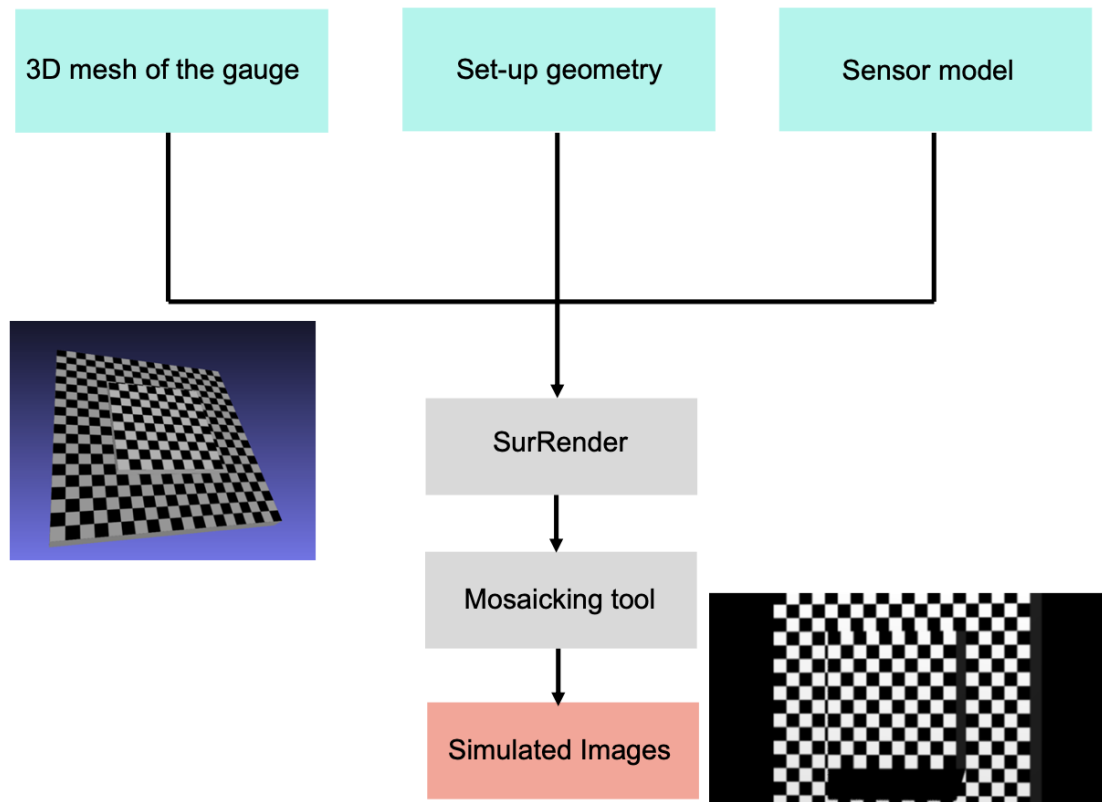


Figure 3.6: Synthetic image process of acquisition with SurRender.

For what concerns the mathematical simulation in MATLAB code that, starting from the known geometry of the setup, the code is able to provide both the 3D (world) and 2D (image plane) coordinates. Exploiting the two implemented codes we can validate the model with a double check: one with ray-tracing simulated data images and one with numerical set of data obtained with a mathematical simulation.

HYPSSOS works as a line scan stereo camera providing on the CCD array the spatial information on one direction and the spectral one in the other direction. The mosaics are realized putting together the lines acquired during the movement of the system in the along-track direction. The movement of the acquisition system is enabled by the movement of the translator.

3.3 Harris corner detector

Starting with the simulated images obtained with the ray-tracing software, the first step in the procedure consists in the definition of the image corners of the chessboard squares. To derive the 2D coordinates of the corners impressed on the simulated images we decided to implement a Harris corner detector (Juraneck L., & al.,2018).

The corners can be defined as important local features in the images; they are points with high curvature in a curve and are individuated in the zone where four regions of different brightness meet. There are many methods to detect the corners, but they can mainly be divided into contours based, that works on image edge data; and intensity based. Harris Corner method belongs to the second group: it works on a gray color image, and it detects the corners analyzing the different scale of gray in the image. The precision and the quality of the corner's coordinates detection is crucial for the calibration.

Before understanding where to find the corners coordinates, we have to define three kinds of regions in the gray image: the **flat** region (Figure 3.7a) where there is no gradient changes in all the directions; then we have the **edge** region (Figure 3.7b) where there is no change in gradient along the edge direction; and finally, the **corner** (Figure 3.7c) where there is a significant gradient variation in both directions.

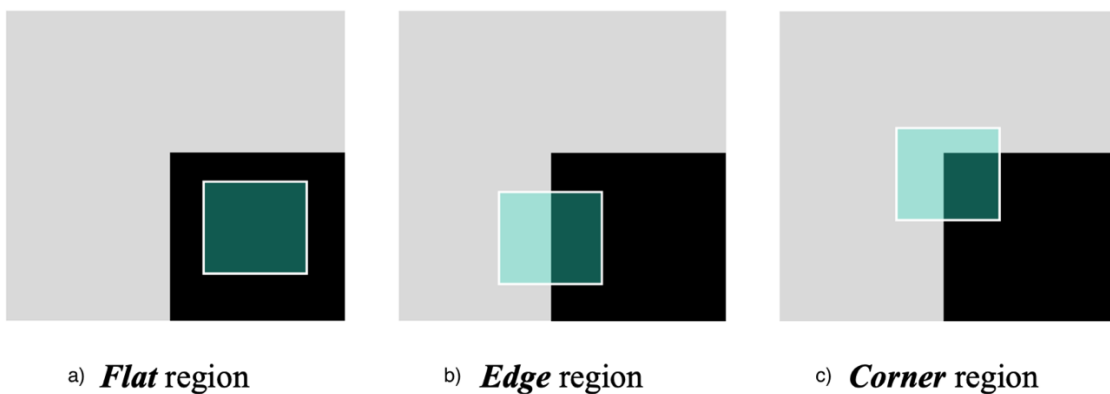


Figure 3.7: a) represents the flat region; b) represents the edge region. c) represents the corner region.

Harris corners are based on an error model, evaluated in the following equation

$$E(u, v) = \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2 \quad (3.1)$$

where I is the intensity of the image in the point (x, y) and u and v are the shifts in the two directions, and $w(x,y)$ is the window function, that is either a rectangular window or a Gaussian window which gives weights to pixels underneath.

We have now to maximize this function $E(u,v)$, in particular the second term. We can do this applying a Taylor expansion that leads us, for small shifts $[u, v]$, to

$$E(u, v) \approx [u \ v] \left(\sum w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.2)$$

If we call M the term inside the round parenthesis, we can rewrite the equation like $E(u, v) \approx [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}$. Now, finding the eigenvectors of M , we can determinate the different types of regions: flat, edge, corner. Evaluating the score R

$$R = \det(M) - k (\text{trace}(M))^2$$

$$\det(M) = \lambda_1 \lambda_2$$

$$\text{trace}(M) = \lambda_1 + \lambda_2$$

where k is an empirical constant factor (typically with a value of 0.04-0.06), and λ_1 and λ_2 are the two eigenvalues of M . so we have that:

- if $|R|$ is small, and so λ_1 and λ_2 are small, we are in the flat region.
- $R < 0$, that means that $\lambda_1 \gg \lambda_2$ or vice versa, we are in an edge region.
- $R > 0$, $\lambda_1 \sim \lambda_2$ and λ_1, λ_2 are big values, we have a corner.

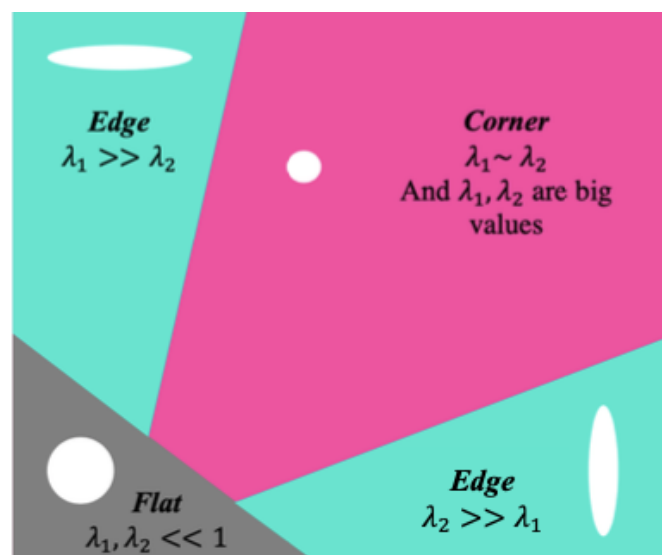


Figure 3.8: Eigenvalues for the different kind of region.

3.3.1 Python implementation

To implement the Harris Corner detector routine in Python we have used the `cornerHarris` function of the OpenCV toolbox: giving in input a gray color image and the 3D coordinates of the bi-planar chessboard it allows to obtain the detection of the corner coordinates. Here the code implementation

```
# #image definition
img = cv2.imread('img_tilt.png')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray = np.float32(gray)
#Harris Corner detector
dst = cv2.cornerHarris(gray, 2, 3, 0.04)
#centroid definition
ret, dst = cv2.threshold(dst, 0.01 * dst.max(), 255, 0)
dst = np.uint8(dst)
ret, labels, stats, centroids = cv2.connectedComponentsWithStats(dst)
```

To find the corners with a better accuracy we did also a refinement of the coordinates using another function, `cornerSubPix`, that provides a sub-pixel accuracy.

```
#subpixel refinement
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.001)
corners = cv2.cornerSubPix(gray, np.float32(centroids), (5, 5), (-1, -1), criteria)
```

Below there are the results obtained for a chessboard modelled with the SW SurRender.

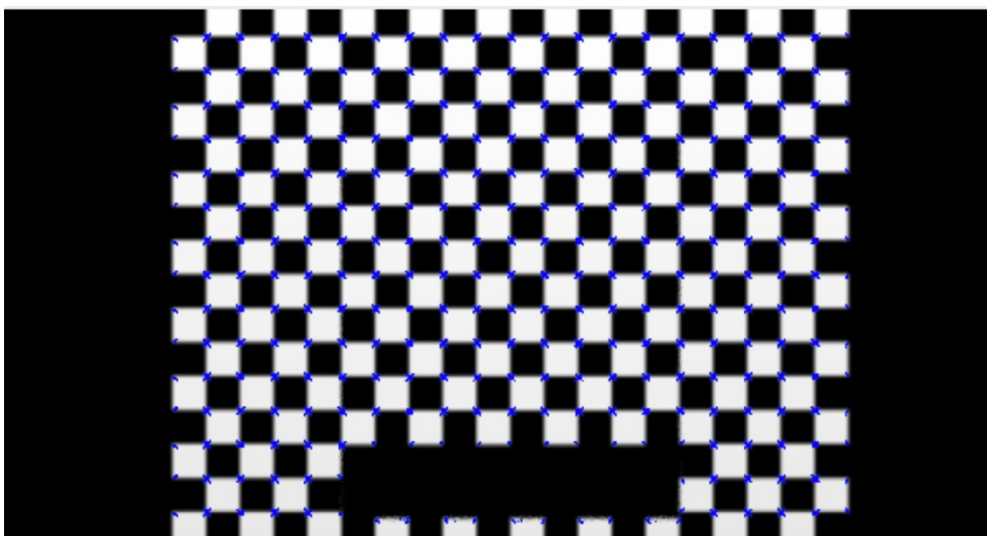


Figure 3.9: Harris Corner detection on Python applied on a simulated image of the bi-planar chessboard.

Chapter 4: STUDY SETUP

In this Chapter, I describe the results obtained from the analysis (camera model estimation) starting with both the images simulated with the ray-tracing software and with the numerical dataset generated in MATLAB (numerical simulation). The validation procedure and the main problems we had to face in terms of detection error have been presented together with the conclusions of the work.

4.1 Results obtained with a numerical simulation

To check the implementation of the Gupta method for the pushbroom model in Python, we started deriving a MATLAB code that would allow to get a numerical simulated dataset, not subjected to any source of errors introduced, for instance, by the detection of the corner coordinates by the Harris detector. This code allows to obtain a data set of correspondence between the 3D coordinates of the gauge and the 2D image coordinates.

This is the part of code implemented to obtain the 3D coordinates of the upper and lower planes defined by the number of squares of the chessboard (Nup, Ndown) and the single square dimension (CHESSBOARDDIM). For simplicity we decide to fix the second plane not in the middle of the first one but on a corner.

```
[xdw ydw]=meshgrid([1:Ndown1]*CHESSBOARDDIM,[1:Ndown2]*CHESSBOARDDIM);
z=xdw;
z(:)=0;
X=[xdw(:),ydw(:),z(:)];
[x y]=meshgrid([1:Nup1]*CHESSBOARDDIM,[1:Nup2]*CHESSBOARDDIM);
cond_x=(xdw<min2(x))|(xdw>max2(x));
cond_y=(ydw<min2(y))|(ydw>max2(y));
f=find(cond_x|cond_y);
z=x;
z(:)=H;
X=[X(f,:);[x(:),y(:),z(:)]];
```

This code allows to obtain the various intrinsic and extrinsic parameters that lead us to verify our pushbroom model implementation in Python. So, first of all, the matrix L (equation 2.16) can be defined, as follows:

$$[L] = \begin{bmatrix} 10.889 & 0 & 0 \\ 0 & 1.0208e + 4 & 3.840 \\ 0 & 0 & 1 \end{bmatrix}$$

L is obtained, in fact, by imposing a velocity V_x , of 7.0640 px/s, repetition time (Rt) of 0.013s and a focal length of 245 mm. L_{11} has been converted in mm dividing by Rt . Subsequently, considering the inclination of $\pm 20^\circ$, the V_x will be divided in the other two components along y and z directions.

The matrix M that we obtain considering the 3D coordinates and the 2D ones is:

$$[M] = \begin{bmatrix} 10.889 & 0 & 0 & 0 \\ 0 & 10208.333 & 3.840 & 3.84e - 13 \\ 0 & 0 & 1 & 1000 \end{bmatrix}$$

And the translation vector has been set to: $T = [0, 0, -1.000e+03]$.

Projecting the 3D points using the matrix M , we obtained all the direct correspondences of corners.

```
ijn2(:,1)=[Mmis(1,1:4)]*[X';ones(1,length(X))];
ijn2(:,2)=[Mmis(2,1:4)]*[X';ones(1,length(X))]./([Mmis(3,1:4)]*[X';ones(1,length(X))]);
```

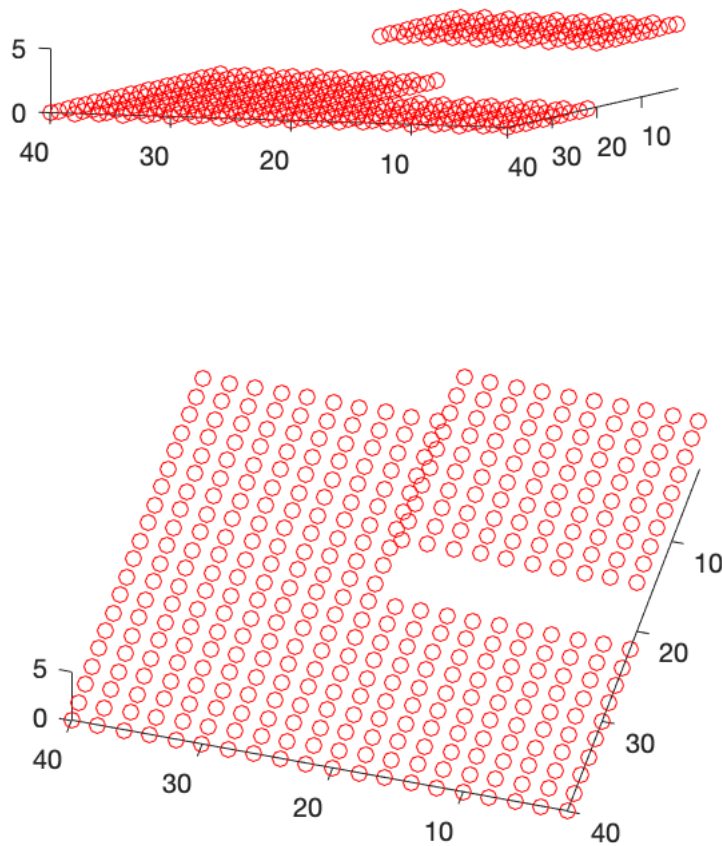


Figure 4.1: Plots of all the corners of the bi-planar chessboard.

As a preliminary analysis, giving as input these correspondences of points to the Python code, we were able to verify that the results are the same in terms of matrix M , Translation vector, Rotation vector, and focal length. Since the estimation provides the results that we expected for both the nadiral (Table 4.1) and tilted case of an angle of 20° (Table 4.2), the implemented Python Gupta model can be considered correct

Table 4.1: Results obtained in Python with the sets of points produced by the numerical simulation with the MATLAB code for a nadiral acquisition.

Reprojecting Error along u direction [px]	1.888e-15
Reprojecting Error along v direction [px]	2.108e-13
Focal length [mm]	244.999
Rotation Vector [deg]	[4.341e-11, 180.000, 3.513e-13]
Translation Vector [mm]	[1.068e-11, -1.231e-11,-1.000e+03]

Table 4.2: Results obtained in Python with the sets of points produced by the numerical simulation with the MATLAB code for a tilted acquisition of 20° .

Reprojecting Error along u direction [px]	3.654e-13
Reprojecting Error along v direction [px]	4.646e-11
Focal length [mm]	244.999
Rotation Vector [deg]	[-179,999,159,999, 1,400e-12]
Translation Vector [mm]	[-1.685e-08,-2.631e-11, 1.000e+03]

Further analysis will be made using the simulated images obtained starting from considering the i, j pixels indices of the image and working to obtain the Intensity matrix of the pixels. Starting from the matrix M , again we want to work to transforming the i, j into the 3D

coordinates. The first step is to create a planar image and so we have $Z_0 = 0$. To obtain X_0 and Y_0 it is necessary to solve the system of equations:

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} - jm_{31} & m_{22} - jm_{32} \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} = \begin{bmatrix} i - m_{14} \\ jm_{34} - m_{24} \end{bmatrix} \quad (4.1)$$

Therefore, evaluating the results for every pixel of the image it is possible to define the intensity like

$$is_x = \left| \left(\left(\frac{X_0}{2} \right), 2 \right) \right| \text{ and } is_y = \left| \left(\left(\frac{Y_0}{2} \right), 2 \right) \right| \quad (4.2)$$

And the Intensity matrix like

$$I(i, j) = ((is_x = 0) \& (is_y = 0)) | ((is_x = 1) \& (is_y = 0)) \quad (4.3)$$

The same procedure can be made with the second level of the chessboard, considering $Z_H = H$, with H the height of the second plane.

$$\begin{aligned} & \begin{bmatrix} m_{11} & m_{12} \\ m_{21} - jm_{31} & m_{22} - jm_{32} \end{bmatrix} \begin{bmatrix} X_H \\ Y_H \end{bmatrix} \\ & = \begin{bmatrix} i - m_{14} - Hm_{13} \\ jm_{34} - m_{24} - H(m_{13} - jm_{33}) \end{bmatrix} \end{aligned} \quad (4.4)$$

4.2 Results obtained with the ray-traced images

4.2.1 Nadiral image analysis on a set of corners

The first simulation, that has been made using the Harris corner detector, uses the simulated nadiral images obtained with SurRender. Remembering that the 3D world points coordinates are known and correct because they are obtained from the geometry of the chessboard, we decide to use a set of eighteen points, composed by the 3D coordinates and the respective 2D coordinates obtained with collineation. With this simulation we obtained quite good values:

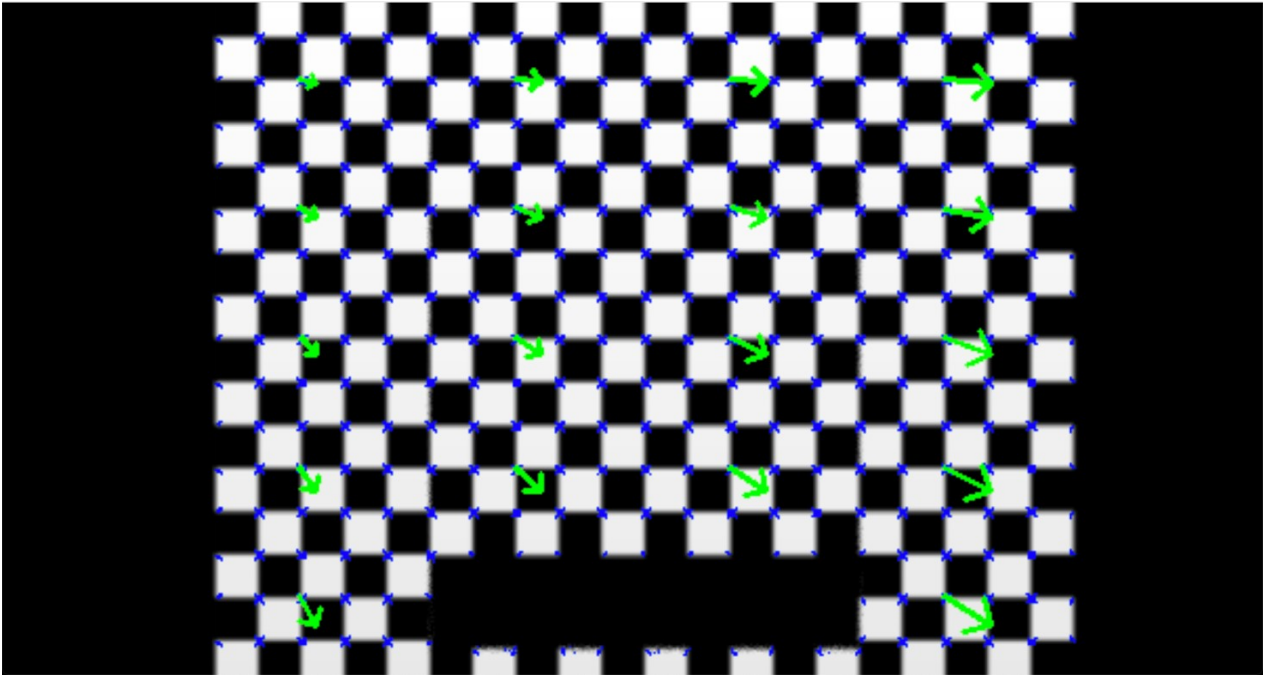


Figure 4.2: Representation of the residual values for the eighteen points, obtained from the simulation for the nadiral image. (The value is multiplied by a factor 1.05).

Table 4.3: Values obtained with eighteen corners (Nadiral Image).

Reprojecting Error along u direction [px]	0.048
Reprojecting Error along v direction [px]	0.026
Focal length [mm]	230.638
Rotation Vector [deg]	[179.197, -179.939, 0.0018]
Translation Vector [mm]	[-2.179, 3.657, 187.273]

The reprojecting error is evaluated as

```
sum = 0
for i in range(0, len(Xt)):
```

```
sum += np.square(xt[i][0]-x_reproj[i][0])
reproj_error_x = np.sqrt(sum)/len(Xt)
```

for both u and v directions. The focal length of 230.64mm is close to the real value of 245mm. The third component of the translation vector is still far from the real value of 1 meter; this is mainly due to the correlation between the parameters, so an error in one variable has an impact also in the other variables. The instability of the estimation reveals also that the number of the corners strongly influences the final estimation. The results obtained with the numerical dataset suggested that the larger the number of the corners more stable is the estimation. This issue has been analyzed in paragraph 4.2.4.

4.2.2 Inclined image analysis

The same analysis has been made for images acquired with a tilted view with respect to nadir, also obtained with the ray-tracing software. With these tilted images, the errors in the definition of the exact image coordinates of the corners increase.

These simulations are very important because they are representative of the way in which HYPSONS will acquire the images, that is with an inclination of 20° . This is the reason why we performed a deeper analysis with the tilted images created by the MATLAB code, described in the preview chapter.

Table 4.4: Values obtained with eighteen corners (Tilted Image).

Reprojecting Error along u direction [px]	0.279
Reprojecting Error along v direction [px]	0.023
Focal length [mm]	344.041
Rotation Vector[deg]	[170.317, -171.316, 0.449]
Translation Vector [mm]	[39.777, 2.997, 280.947]

The correlation between the parameters leads us to have significant errors especially in the rotational vector.

4.2.3 Harris detected corners analysis

The following step has been to evaluate the calibration using all corners of the chessboard. For this test, all points detected from the Harris corner detector program, have been used. In this way what we obtain is shown in Table 4.5 and in Figure 4.3

Table 4.5: Values obtained with all detected corners.

Reprojecting Error along u direction [px]	0.0064
Reprojecting Error along v direction [px]	0.0053
Focal length [mm]	355.995
Rotation Vector[deg]	[179.197, -179.939, 0.0018]
Translation Vector [mm]	[-10.358, 16.868, 1445.121]

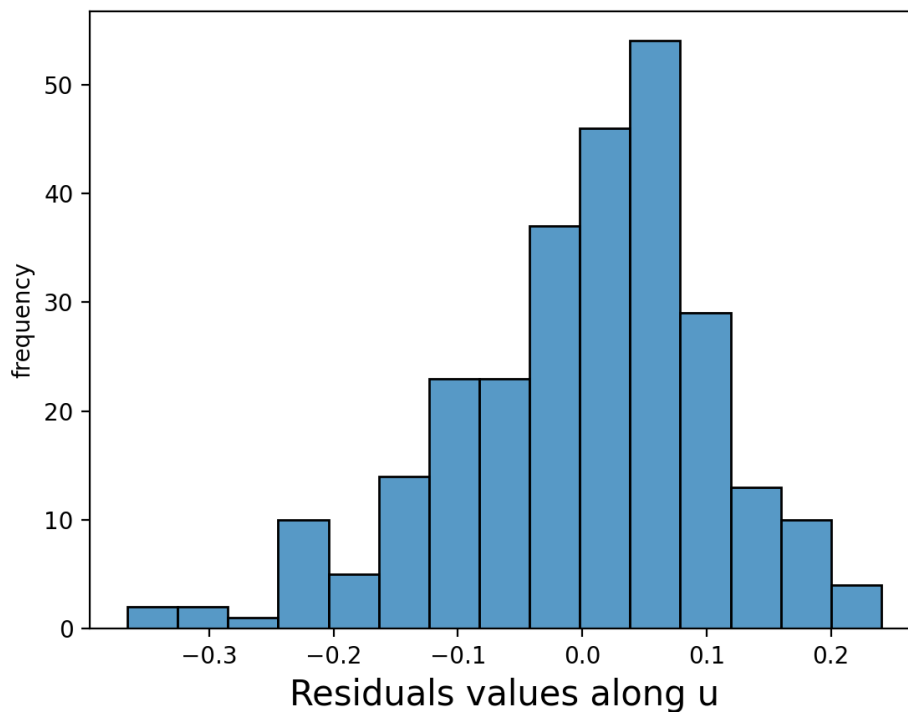


Figure 4.3a: Histogram of the residual values (in pixel units) obtained from the primary analysis for all detected corners with Harris along u direction.

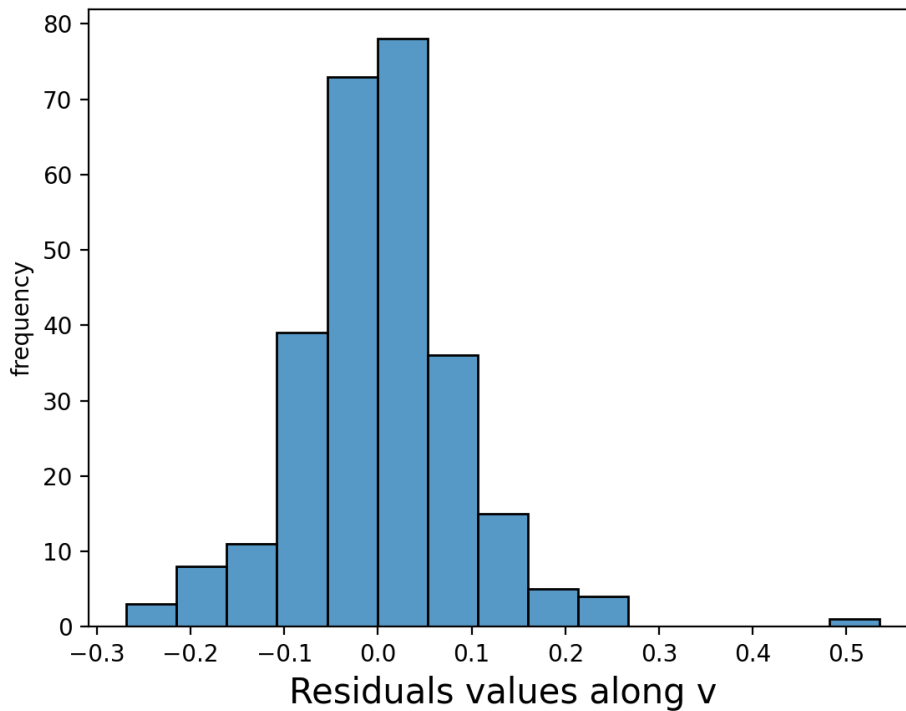


Figure 4.3b: Histogram of the residual values (in pixel units) obtained from the primary analysis for all detected corners with Harris along v direction.

The deviations of the corners coordinates are 0.38px for the u direction and of 0.28px for the v direction.

The estimation of the focal length is quite distant from what we were expecting, mainly due to the error in the detection of the corners by the Harris operator, of the order of few tenths of pixels. The number of the corners influences the stability of the system, together with the height of the second plane of the chessboard that can help to decouple the correlation between the parameters.

To better understand how much the detection affects the focal length estimation, we tested different configurations of the chessboard. The code is able to create a plot of the focal length varying the error detection.

4.2.4 Instability of the detection process

To analyze the instability of the estimation in relation with the detection of the corners, using the Harris Corner operator, as just said, we created a MATLAB code. The inputs of the code are the matrix L of the intrinsic parameters, that we were able to establish considering the

instrument and its characteristics; the 3D coordinates of the chessboard obtained by defining the number of the squares of the chessboard, the size of the single square, the height of the second plane, and finally a random error of detection taken between 0 px and 0.4 px from the literature. We obtain that the focal length is highly sensitive to the definition of the height (helping in decoupling the parameters) and to the number of corners.

The configuration of a chessboard as the one we used as a target (as a starting design), with 390 corners and a height of the second plane of 5 mm, gives as response a range of focal lengths that varies from 244.99 mm to 350.53 mm with several peaks.

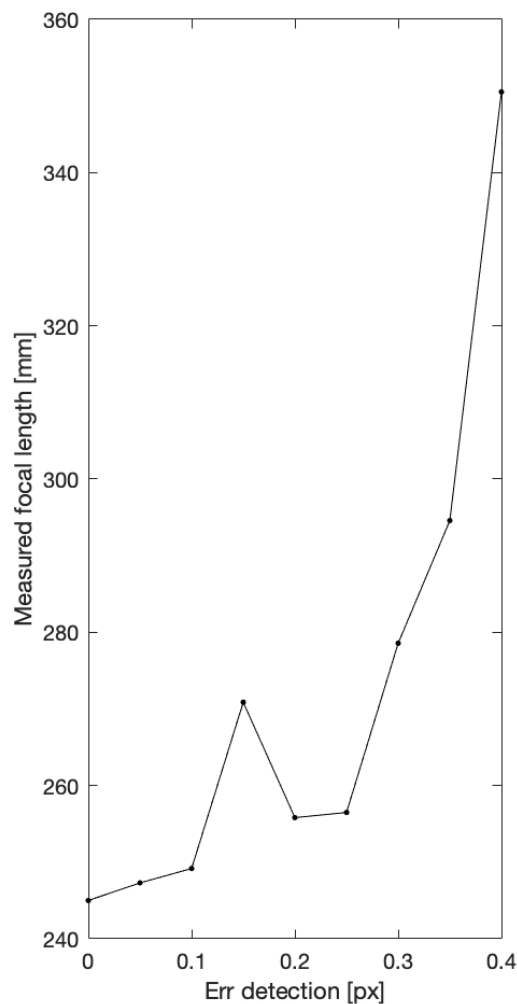


Figure 4.4: Representation of the variation of the focal length due to an error of detection with an evaluation on 390 corners.

Other configurations obtained changing the number of the corners and the size of the squares of the chessboard, without changing the height, are shown in the following plots. It is clear that the number of corners influences the focal length: for the case of 4900 points,

corresponding to a chessboard with 70x70 squares in the lower plane and 35x35 squares in the upper plane, with a dimension of the squares of the chessboard of 2mm (as the one we used) the focal length range varies from 244.99 mm to 271.04 mm; meanwhile in the case of again 4900 points but a size of the squares of the chessboard of 5 mm the range varies from 244.75 mm to 254.95 mm. These configurations are, without any doubt, more stable and would allow to reduce the impact of the error in the estimation.

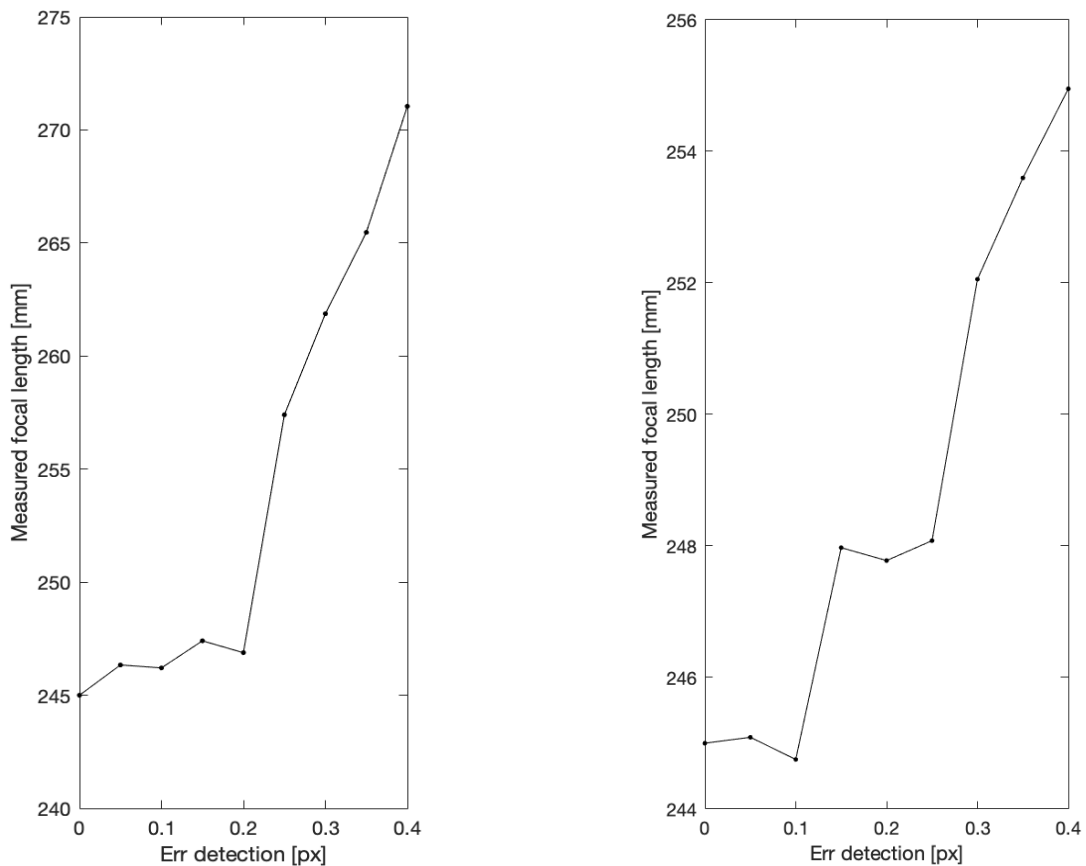


Figure 4.5: Representation of the variation of the focal length due to an error of detection with an evaluation on 4900 corners. Image a) with a dimension of the squares of the chessboard of 2 mm and b) dimension of the squares of the chessboard of 5 mm.

Other tests have been made, on a set of 22500 corners, corresponding to a chessboard with 150x150 squares in the lower plane and 75x75 squares in the upper plane, and a height of 100 mm, again with both 2 mm and 5 mm as sizes of the squares of the chessboard. These cases give the best results, in fact the range of the focal length variation for the first configuration is from 244.99 mm to 245.71 mm, instead for the second one from 245.00 mm to 245.27 mm. This is the most stable configuration, but unfortunately not reasonable to realize for the calibration. Most likely this result it is due to the bigger height of the second

plane that allows a better decoupling between the coordinates of the two planes and a better analysis.

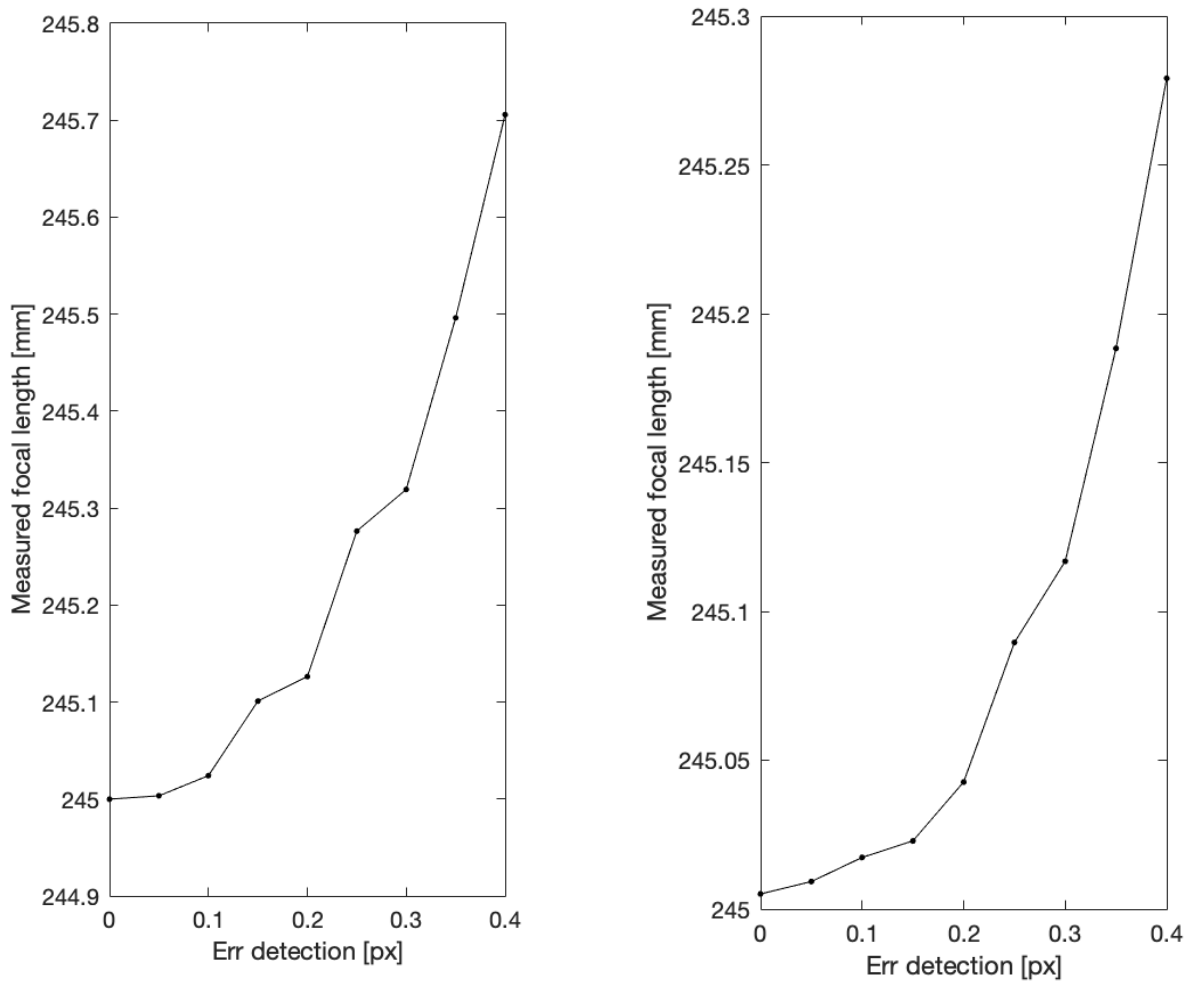


Figure 4.6: Representation of the variation of the focal length due to an error of detection with an evaluation on 22500 corners. Image a) with a dimension of the squares of the chessboard of 2 mm and b) with a dimension of the squares of the chessboard of 5 mm.

From these tests, in order to gain a more stable configuration for the calibration, we searched for a good compromise between the limits of realization of the gauge and the stability of the estimation procedure. A good solution, to gain a larger number of corners, would be to stretch the chessboard in the along-track direction, meanwhile we have a limit in the cross-track direction. What we obtained to fit all the requirements is a chessboard with the following geometry: we kept the size of the squares of the chessboard of 2 mm, the cross-track direction of 4 cm and we elongate the along-track direction to 8 cm, and raised the second plane to 1 cm. In this way we gained 800 corners. The fitted result is reported in the following plot.

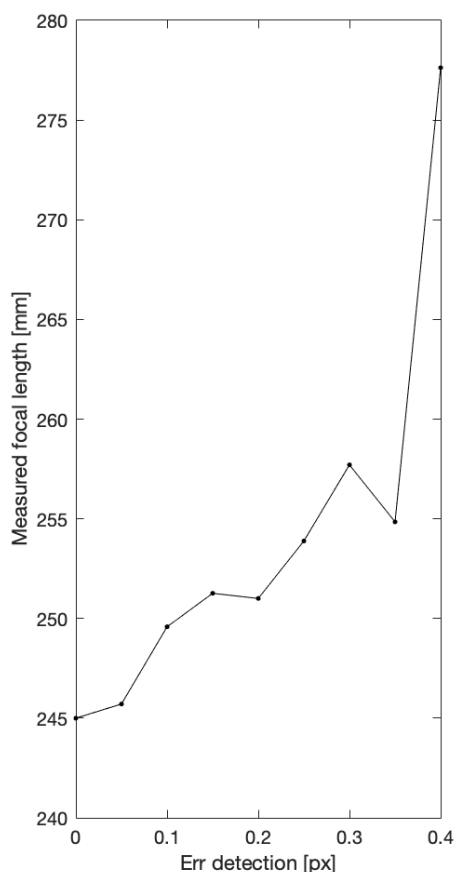


Figure 4.7: Representation of the variation of the focal length due to an error of detection with an evaluation on 800 corners.

As it is possible to see in Figure 4.7 the range of the focal length varies from 244.99 mm to 277.63. This last test could represent a good compromise for the stability of the system considering a chessboard of feasible realization.

The purpose of determining the individual camera parameters is to allow to know how the camera parameters influence the calibration of the stereo camera. For instance, the focal length and principal point offset of the camera may be known quite precisely from manufacturing specifications. In the DLT method for determining the camera matrix M , as described in section 2 there is no way to incorporate this information into the calibration process. One way to do that, however, is to get an initial solution for the camera matrix M , extract the parameters from the matrix, fix the known parameters to the known values, and finally carry out an iterative parameter fitting algorithm to get a more exact estimate of the camera mapping. This approach could allow any of the parameters to be fixed absolutely, or with a specified standard deviation and it will be possible to parametrize the camera in different ways to allow for different types of knowledge of the stereo camera setup.

This could be a possible solution for the estimation problems that we encountered so far, another kind of solution that we tested has been to try a refinement of the re-projection errors using Chebyshev polynomials as described in the next session.

4.3 Chebyshev polynomials refinement

The linear pushbroom stereo camera model is not able to capture the non-linear aspects and so it is possible to fit them to get the best residuals in the image coordinate projection. Using a single photogram for every pose and so not having a defined movement, imply the overfit of the model.

For these reasons we decided to work with a polynomial approximation. In this way we should obtain an image coordinate with stronger consistency with the linear pushbroom model and the coordinates of the object. In particular the Chebyshev polynomials of the first kind $T_n(x)$ have been used.

$$T_n(x) = \cos(n\theta)$$

with $x = \cos(\theta)$. If the range of the variables is in the interval $[-1, 1]$, the range of θ is $[0, \pi]$. Knowing that $\cos(n\theta)$ can be written as a polynomial of degree n of $\cos(\theta)$, the first few polynomials of Chebyshev are:

$$T_0(x) = 1;$$

$$T_1(x) = x;$$

$$T_2(x) = 2x^2 - 1;$$

$$T_3(x) = 4x^3 - 3x.$$

So, combining the polynomial of the third grade for two variables, in our case u and v , we obtain the following equation that we had evaluated to model the residuals in both directions obtaining two polynomials $P_u(u, v)$ and $P_v(u, v)$.

$$P(u, v) = a_0 + a_{10}u + a_{01}v + a_{11}vu + a_{20}(2u^2 - 1) + a_{02}(2v^2 - 1) + a_{12}v(2u^2 - 1) + a_{21}u(2v^2 - 1) + a_{30}(4u^3 - 3u) + a_{03}(4v^3 - 3v) \quad (4.5)$$

The a_{ij} coefficients have been estimated with a least squares optimization following the process as follows.

The corners are homogeneously distributed along the gauge and cover the whole FOV of the stereo camera so, the approximation function is used to model the residual between the detected image coordinates and the projected one. For this step it is important to underline the difference between the forward and the inverse mapping:

- The forward mapping allows the calculation of the new image coordinates (u_{new}, v_{new}) starting from the original coordinates. Obtained the polynomials for both directions we can write

$$\begin{bmatrix} u_{new} \\ v_{new} \end{bmatrix} = \begin{bmatrix} P_u(u_{or}, v_{or}) \\ P_v(u_{or}, v_{or}) \end{bmatrix} + \begin{bmatrix} u_{or} \\ v_{or} \end{bmatrix} \quad (4.6)$$

- The inverse mapping is evaluated as

$$\begin{bmatrix} u_{or} \\ v_{or} \end{bmatrix} = \begin{bmatrix} P_u(u_{new}, v_{new}) \\ P_v(u_{new}, v_{new}) \end{bmatrix} + \begin{bmatrix} u_{new} \\ v_{new} \end{bmatrix} \quad (4.7)$$

Considering the inverse error like $e_{inv} = Mx_w - x_{img}$.

The forward mapping is useful for computing the transformation of the points from the original location to the new one but not for resampling the image because it may create gaps in the resampled image; meanwhile the inverse mapping is useful for resampling the image.

Starting from these considerations we fitted the inverse error with the Chebyshev polynomial to obtain the coefficients. In the following table there are the results for both u and v directions.

Table 4.6: Coefficients for the Chebyshev polynomial fitting the residuals in u and v variables.

	Coefficients for $P_u(u, v)$	Coefficients for $P_v(u, v)$
a_0	8.507e-01	1.615e-01
a_{10}	- 1.040e-02	- 5.154e-04
a_{01}	5.540e-04	- 4.073e-03

a_{11}	1.150e-06	5.836e-06
a_{20}	9.127e-06	1.626e-07
a_{02}	- 1.914e-06	7.012e-06
a_{12}	- 3.258e-10	- 1.231e-09
a_{21}	3.175e-11	- 2.564e-09
a_{30}	- 6.075e-10	- 8.464e-13
a_{03}	3.087e-10	- 9.353e-10

The analysis of the coefficients reveals that the major influence is given from the a_0 coefficients.

4.3.1 Re-estimated error

According to the Chebyshev polynomial fit results, we added the evaluated error to the corners detected with Harris, in this way we were able to re-estimate the M projective matrix, giving as an input the new image coordinate, and so of the intrinsic parameters. We gained a better fit for the displacement residuals and even for the focal length.

Table 4.7: Values obtained with 2D Chebyshev polynomial refinement.

Reprojecting Error along u direction [px]	0.0035
Reprojecting Error along v direction [px]	0.0012
Focal length [mm]	329.489
Rotation Vector [deg]	[178.437, 179.998, 0.0069]
Translation Vector [mm]	[-9.954, 16.817, 1337.511]

The focal length is improved by the 7.4%, passing from the value of 355.99mm to 329.49mm. Meanwhile the residuals along the two directions yielded a significant improvement especially along the v direction: the maximal residuals length along the u direction is of 0.15px meanwhile along the v direction reaches the value of 0.07px.

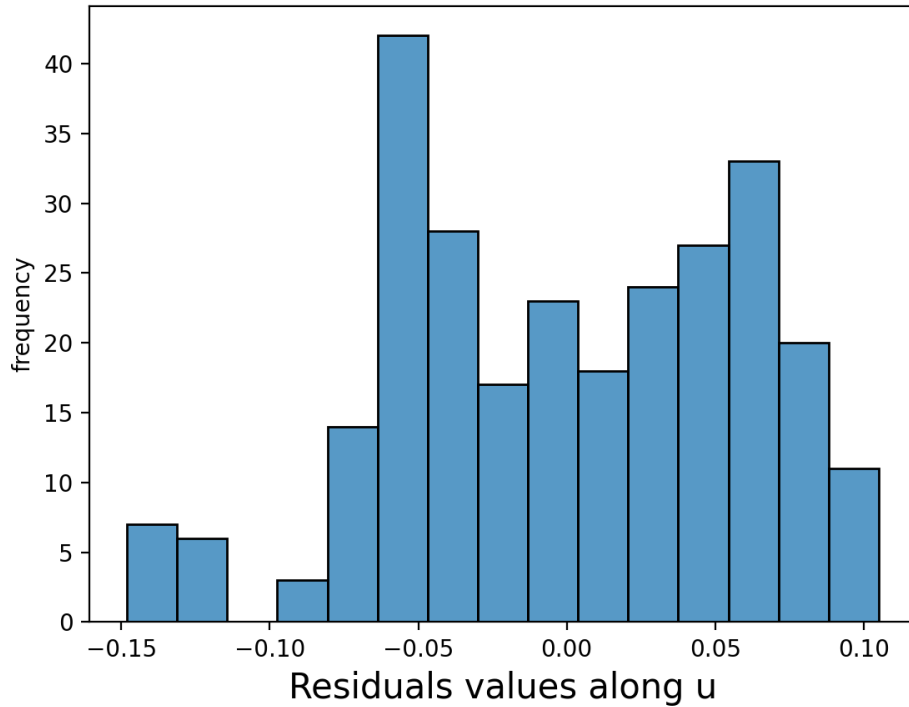


Figure 4.8a: Histogram of the residual values (in pixel units) obtained from the refinement with Chebyshev for all detected corners with Harris corner detector along u direction.

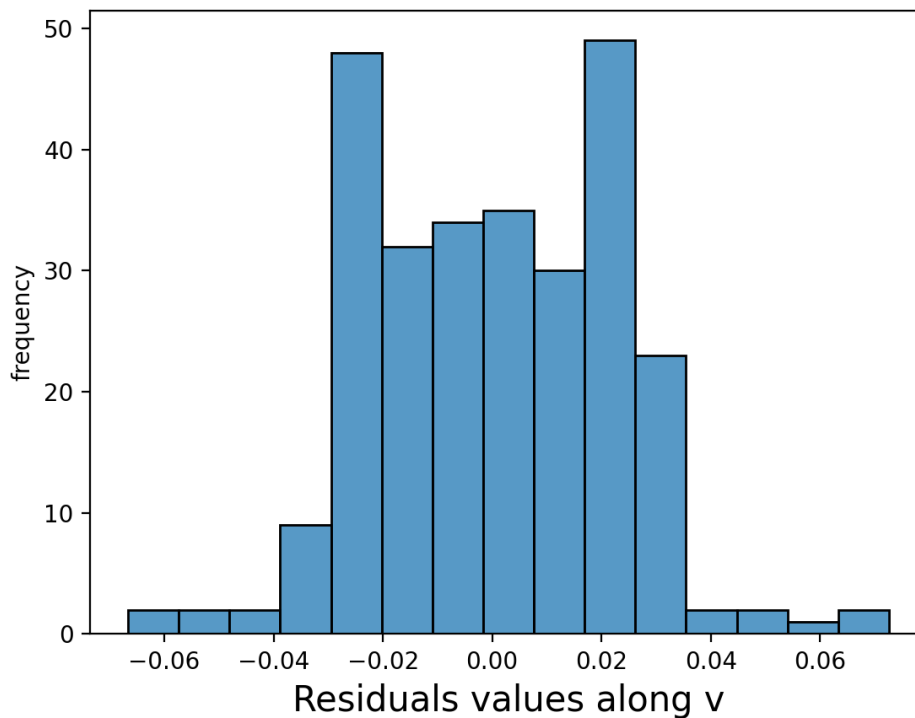


Figure 4.8b: Histogram of the residual values (in pixel units) obtained from the refinement with Chebyshev for all detected corners with Harris corner detector along v direction.

Considering the 2D Chebyshev polynomials (Fig. 4.8a and 4.8b), the larger residual errors along u direction indicates that the camera motion has a larger impact on the accuracy of

the corner coordinate definition. On the other hand, residual error that changes as a function of v direction indicates that the error is related to intrinsic properties of the camera itself.

4.4 Correlation between parameters

Another important point is to find how much the variation of one parameter influences the others. It is a statistic way to understand the connection between pairs of variables. At the same time this is a useful way to understand the stability of a system and the quality of the obtained data.

The correlation between parameters must be avoided as much as possible, because it will lead to instability of the process and incorrect solutions.

The covariance between two variables must be calculated before the correlation, and then the standard deviation. For example, considering two variables X , Y the correlation between the two is defined like

$$\text{covariance} = \rho = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \quad (4.8)$$

- If ρ is +1, means that the two variables have a perfect positive relationship; this means that when one variable moves higher or lower, the other variable do the same; the nearer to 1 is the value, the stronger get the correlation.
- If $0 < \rho < 1$, the two variables move in the opposite direction, and they have a negative correlation.
- If ρ is -1, they have a perfectly negative correlation.

So near to 1 or -1 the linear relationship is strong; near to 0 the linear relationship is weak, and we are in a better condition.

Evaluating this would allow us to understand in a better way the calibration stability, and it would be a good method for re-estimate the projection matrix constraining the necessary parameters, fixing the parameters that we know with higher precision.

Conclusions

This work aimed to calibrate a new concept of hyperspectral stereo imaging system (HYPSOS) following the method introduced by Gupta and Hartley for estimating the linear pushbroom stereo camera model.

The method has been implemented in a Python code and it has been validated using two kinds of datasets: a numerical dataset created with a MATLAB code and using a virtual dataset produced using a ray-tracing software able to generate simulated images.

Working with the numerically simulated dataset, the implemented method of Gupta & Hartley has been successfully validated. Then, we realized a series of chessboard simulated images exploiting the geometry of a bi-planar chessboard and the ray-tracing software, SurRender. This second analysis has been made extracting the 2D image coordinates, called corners, using the Harris Corner detector. With this analysis the strong correlation between the different parameters has been underlined, leading to some errors in the estimation of the intrinsic and extrinsic parameters, and at suggesting a certain instability of this estimation system.

The sensitivity in the focal length estimation has been evaluated varying the amount of the errors introduced by the corner detection process and the tests suggested that increasing in the number of the possible corners and/or the increasing in the height of the second plane of the gauge makes the estimation more robust and stable.

Several chessboard configurations have been tested in order to find the best trade-off between a good number of corners, a proper height for the second plane but in accordance with the limits imposed by the chessboard realization and the limits of the scanning system in the along-track direction.

As exposed in the reference paper of Gupta & Hartley, fix the known parameters to known values, and finally carry out an iterative parameter fitting algorithm to get a more exact estimate of the camera mapping is the best approach to overcome the instability problems of the system. Allowing any of the parameters to be fixed absolutely, or with a specified standard deviation, it is possible to parametrize the camera in different ways to allow for different types of knowledge of the stereo camera setup.

In this work, the possibility to fix some parameters has not been considered, but the possibility of determining the individual camera parameters, has allowed the knowledge of the camera parameters to influence the calibration of the stereo camera.

The work can be considered very useful to drive the choice of the best configuration for the design of a proper chessboard to be used for the geometric calibration of a pushbroom stereo camera such as HYPSONS. The method implemented will be the starting point for the procedures that will be adopted during the actual calibration of HYPSONS with the real images and the use of the simulated and synthetic datasets have provided to the team, involved into the project, a good benchmark for the identification of the critical aspects imposed by the complexity of the calibration of an innovative imaging system as HYPSONS.

References

- Ahmad Yousef, K.M.; Mohd, B.J.; Al-Widyan, K.; Hayajneh, T., Extrinsic Calibration of Camera and 2D Laser Sensors without Overlap, *Sensors* 2017, <https://doi.org/10.3390/s17102346>.
- Barbieux, K., Pushbroom Hyperspectral Data Orientation by Combining Feature-Based and Area-Based Co-Registration Techniques. *Remote Sens.* 2018, 10, 645.
- Brochard R., et al., Scientific image rendering for space scenes with the SurRender software, *arXiv preprint arXiv:1810.01423*, 2018
- Chen Chu-Song, Chen Yu-Ting, Huang Fay, Stitching and Reconstruction of Linear-Pushbroom Panoramic Images for Planar Scenes, Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan, R. O. C.
- Chen Jie, Zou Li-hui; Zhang Juan; Dou Li-hua, The Comparison and Application of Corner Detection Algorithms, *Journal of Multimedia*, 2009, Vol. 4 Issue 6, p435-441.
- Cucs-Naoyuki Ichimura, Naoyuki Ichimura, Shree K. Nayar, A Framework for 3D Pushbroom Imaging, 2003, Department of Computer Science, Columbia University New York, New York 10027, USA
- De Franchis Carlo, Meinhardt-Llopis Enric, Greslou Daniel, Facciolo Gabriele, Attitude Refinement for Orbiting Pushbroom Cameras: a Simple Polynomial Fitting Method, *Image Processing Online*. 2015. 328–361. 10.5201/ipol.2015.146.
- Donna Monica, Ahmad Matyanto, Wismu Sunarmodo, Mugroho Wifijatmiko, Geometric correction model for dual sensor pushbroom aerial camera, *IOP Conf. Series: Earth and Environmental Science* 500 (2020) 012058
- Drareni Jamil, Sturm Peter, Roy Sébastien, Plane-Based Calibration for Linear Cameras. OM- NIVIS 2008 - 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras, Rahul Swaminathan and Vincenzo Caglioti and Antonis Argyros, Oct 2008, Marseille, France.
- Gómez-Chova L., Alonso L., Guanter L., Camps-Valls G., Calpe J., Moreno J., Correction of systematic spatial noise in push-broom hyperspectral sensors: application to CHRIS/PROBA images, *Appl Opt.* 2008 Oct 1;47(28): F46-60. doi: 10.1364/ao.47.000f46. PMID: 18830284.
- Gupta R., Hartley R. I., Linear Pushbroom Cameras, *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL. 19, NO. 9, 1997.
- Gupta R., Hartley R., Camera Estimation for Orbiting Pushbrooms, General Electric corporate R&D, River Rd, Schenectady, NY 12390 USA, 2012.

- Harris, C.; Stephens, M., A combined corner and edge detector. In: Proceedings of the Alvey Conference, 1987, pp. 189-192.
- Hartley R. I., G.E. CRD, An Object-Oriented Approach to Scene Reconstruction, Schenectady, NY, 12301, 1996 IEE.
- Hirschmüller H., Scholten F., Hirzinger G., Stereo Vision Based Reconstruction of Huge Urban Areas from an Airborne Pushbroom Camera (HRSC), Published in DAGM-Symposium 31 August 2005.
- Jannati, M.; Valadan Zoej, M.J.; Mokhtarzade, M. Epipolar Resampling of Cross-Track Pushbroom Satellite Imagery Using the Rigorous Sensor Model. *Sensors* 2017, 17, 129. <https://doi.org/10.3390/s17010129>.
- Juranek L., Stastny J., Skorpil, V., Effect of Low-Pass Filters as a Shi-Tomasi Corner Detector's Window Functions. 2018, 10.1109/TSP.2018.8441178.
- Lebreton J., Brochard R., Baudry M., Jonniaux G., Salah Adrien H., Kanani K., Goff et al., Image simulation for space applications with the SurRender software, 11th International ESA Conference on Guidance, Navigation & Control Systems, 22 - 25 June 2021, <https://doi.org/10.48550/arxiv.2106.11322>.
- Lei Li, Zhang Qin, Huang Danfeng, et al. "A review of imaging techniques for plant phenotyping." *Sensors (Basel, Switzerland)* vol. 14,11 20078-111. 24 Oct. 2014, doi:10.3390/s141120078.
- Li Yi-bo, Li Jun-jun, Harris Corner Detection Algorithm Based on Improved Contourlet Transform, *Procedia Engineering*, Volume 15, 2011, Pages 2239-2243, ISSN 1877-7058.
- Mason J.C., Hanscomb D.C., Chebyshev Polynomials, 2002, Chapman and Hall_CRC.
- Morten H., Jakob W., Jeppe Revall Frisvald, Superaccurate Camera Calibration via Inverse Rendering, 2020.
- Naletto Giampiero, Agostini Livio, Brotto Francesca, Cremonese Gabriele, Faccioni Matteo, et al., Laboratory characterization of HYPSONS, a novel 4D remote sensing instrument, *Proc. SPIE 11858, Sensors, Systems, and Next-Generation Satellites XXV*, 118580E (12 September 2021); doi: 10.1117/12.2601091.
- Perrier R., Arnaud E., Sturm P., Ortner M.. Estimating Satellite Attitude From Push-broom Sensors. *CVPR 2010 - 23th IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2010, San Francisco, United States. pp.591-598.
- Pi Yingdong, Xie Baorong, Yang Bo, et al., On-orbit Geometric Calibration of Linear Pushbroom Optical Satellite Based on Sparse GCPs[J]. *Journal of Geodesy and Geoinformation Science*, 2020, 3(1): 64-75. DOI:10.11947/j.JGGS.2020.0107.

- Reed K. M., Solid Model Acquisition from Range Imagery, 1998.
- Ringaby, E., Geometric Models for Rolling-shutter and Push-broom Sensors (PhD dissertation, Linköping University Electronic Press), 2014, <https://doi.org/10.3384/diss.diva-110085>.
- Roberts, DA, Y Yamaguchi, and RJP Lyon. 1986. "Comparison of various techniques for calibration of AIS data." NASA STI/Recon Technical Report N 87.
- Suk-Kun, Hoon Chang, The Evaluations of Sensor Models for Push-broom Satellite Sensor, Korean Journal of Geomatics, Vol.4, No.1, November 2004, pp.31-37.
- Sung Woong Shin, B.Sc., M.Sc., Rigorous Model of Panoramic Cameras, the Ohio State University, 2003
- Suomalainen, Juha, Niels Anders, Shahzad Iqbal, Gerbert Roerink, Jappe Franke, Philip Wenting, Dirk Hünninger, Harm Bartholomeus, Rolf Becker, and Lammert Kooistra. 2014. "A lightweight hyperspectral mapping system and photogrammetric processing chain for unmanned aerial vehicles." Remote Sensing 6 (11):11013-11030.
- Tordi Massimiliano, Cremonese Gabriele, Naletto Giampiero, Marchiori Gianpietro, Re Cristina, Lucchetti Alice, Agostini Livio, HYPSPSOS: a HYPerspectral stereo observing system for solar system exploration, Proc. of SPIE Vol. 11443 114437C-1
- Wendel Alexander, Underwood James, Extrinsic Parameter Calibration for Line Scanning Cameras on Ground Vehicles with Navigation Systems Using a Calibration Pattern, Sensors 2017; doi:10.3390/s17112491.
- Weser T., Rottensteiner F., Willneff J., Fraser C. S., A generic pushbroom sensor model for high-resolution satellite imagery applied to SPOT5, Quickbird and ALOS data sets, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2007, Vol.36.
- Zhang Z., A flexible new technique for camera calibration, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, Nov. 2000, doi: 10.1109/34.888718.