

Université Pierre et Marie Curie  
Stage de master 2017

Ecole Normale Supérieure Paris-Saclay  
Cachan

Master de mathématiques et applications

Spécialité : Mathématiques de la Modélisation

Majeure : Contrôle, Optimisation, Calcul des Variations

3D reconstruction from stereo images: application to Mars digital  
terrain models

*AMMOR Amine*

Sous la supervision de Carlo de Franchis, Frédéric Schmidt et Jean-Michel Morel

## Remerciements

Mes remerciements s'adressent en premier lieu, à mes tuteurs de stage: Carlo de Franchis, Jean-Michel Morel et Frédéric Schmidt qui m'ont assisté tout au long de ce projet de recherche.

Je tiens à remercier Carlo de Franchis, pour son encadrement quasi-quotidien, pour m'avoir introduit les problématiques de reconstruction 3D en imagerie satellitaire, pour ses critiques des résultats qui m'ont permis d'identifier quelques erreurs de développement, ainsi que pour ses présentations lucides, sur les différents outils de géométrie projective et épipolaire, et sur les aspects théoriques et pratiques du logiciel s2p, sur lequel j'ai dû travailler.

Son aide a été indispensable pour les différents problèmes d'ordre logiciel auxquels j'ai été confronté à plusieurs reprises durant ce stage, et sans laquelle je n'aurais pas pu aboutir à ce manuscrit.

Je remercie également monsieur Jean-Michel Morel, pour m'avoir fait confiance et proposé ce sujet de stage, ainsi que pour les critiques et conseils qu'il m'a adressés sur mon travail. Les rendez-vous avec la présence de Carlo de Franchis m'ont souvent redonné du courage pour la suite, et ont abouti à divers algorithmes qui ont été des étapes indispensables tout au long de ce stage.

La manipulation des données complexes des images martiennes, et la compréhension du contexte, n'auraient pu être appréhendés efficacement sans l'assistance et la disponibilité de Frédéric Schmidt, qui m'a introduit divers logiciels indispensables pour le projet de recherche et aidé pour leur utilisation, et qui m'a aussi proposé différents articles sur l'état de l'art de la reconstruction 3D des images martiennes, ainsi que sur les propriétés des capteurs considérés, et des diverses données spécifiques à Mars.

Je le remercie également pour sa très grande disponibilité durant tout ce stage malgré sa distance de mon lieu de travail.

D'autres personnes m'ont assisté et aidé durant ce stage. Je remercie notamment Pauline Tan de m'avoir conseillé certains articles qui ont eu une influence importante sur mon travail. Je remercie aussi Enric Meinhardt-Llopis, pour m'avoir assisté et conseillé sur l'utilisation d'un de ces algorithmes.

Je remercie les différents membres du laboratoire, qui ont contribué à une ambiance chaleureuse au sein du lieu de travail, et en particulier les membres du GTTI (Groupe de Travail de Traitement d'Images) dont la participation aux rendez-vous hebdomadaires, de présentation de travaux, a été l'occasion de revoir et d'accroître ma connaissance sur l'état de l'art en traitement d'images.

Finalement, je remercie les professeurs du master "mathématiques de la modélisation", dont l'excellence des cours, m'a permis d'aborder des articles plus approfondis en mathématiques, en particulier les professeurs d'optimisation continue, d'analyse convexe et de recherche opérationnelle , sans lesquels, je n'aurais pu apprécier certains articles, qu'avec une très grande difficulté. Les connaissances acquises lors de ce master constituent une formation complémentaire à ma formation d'ingénieur, qui me seront sans aucun doute, d'une grande utilité, pour la carrière de chercheur à

laquelle j'aspire.

## Summary

One of the most important problems in computer vision (field whose aim is analyzing, processing and understanding the images taken by an acquisition system), is 3D reconstruction. This problem is made of many sub-problems that can be solved sequentially to produce a 3D model from a set of images covering the same scene. In the case of satellite imaging, the goal is to estimate the topography of the scene, called DEM (digital elevation model).

One of the theories that interests planetologists about the planet Mars is the presence on its surface of some possible sites of water erosion. It is thus useful to have a 3D Model of the surface, giving more information about these sites of interests.

The main objective of the internship was to clarify the geometric and optimization problems arising in the stereo-reconstruction of the images of the planet Mars. More particularly, we adapted the algorithmic chain developed by CNES and CMLA to images of a specific camera: the camera HiRISE, which is fixed to the Mars Reconnaissance Orbiter (MRO) satellite. This camera takes high resolution images (up to 30cm / pixel) and takes images of varying sizes due to its structure, covering areas up to 3 km wide on Mars.

The first part of the internship focuses on the (projective) geometrical problem of bundle adjustment. "Bundle adjustment" algorithms estimate the positions and orientations of the cameras from their images.

The second study focused on the stereo matching variationnal problem. "Stereo matching" algorithms produce disparity maps linking each pixel of the first image to its corresponding pixel in the other image. These maps are codified on images, and can be represented with a single image channel thanks to image rectification.

## Contents

<b>1. The 3D reconstruction problem: application to Mars images</b>	<b>9</b>
1.1. The HiRISE Camera . . . . .	9
1.1.1. Description of the components . . . . .	9
1.1.2. Acquisition of the image . . . . .	12
1.1.3. Extraction of the position and orientation of a pixel from the optical axis frame	13
1.1.4. Raw-data extraction for the production of the input Images of the s2p Algorithm	15
1.2. Stereoscopy and topography reconstruction . . . . .	18
1.3. Algorithmic Chain of s2p . . . . .	22
1.3.1. Pointing correction . . . . .	24
1.3.2. Local Stereo rectification . . . . .	27
1.3.3. Global pointing correction and triangulation . . . . .	27
<b>2. The problem of Bundle adjustment</b>	<b>29</b>
2.1. Principle . . . . .	29
2.2. MOLA Point . . . . .	33
2.3. Registration between the MOLA data and the HIRISE data . . . . .	35
2.3.1. Coarse Registration . . . . .	35
2.3.2. Finer Registration . . . . .	36
<b>3. The problem of Stereo Matching (As an optimisation problem)</b>	<b>39</b>
3.1. Principle and state of the art . . . . .	39
3.2. Global solutions of variational models of the stereo matching problem with convex regularization term . . . . .	46
3.2.1. Description of the method . . . . .	46
3.2.2. Numerical Implementation . . . . .	49
3.3. A faster algorithm, for the particular case of the total variation . . . . .	51
3.3.1. Transformation of the Saddle Point Problem to an explicit minimization problem . . . . .	53
3.3.2. Analysis of the explicit minimization problem . . . . .	54
3.3.3. Resolution of the strongly Convex Problem . . . . .	56
3.3.4. Algorithms for the resolution of the problem . . . . .	58
3.3.5. Description and analysis of the algorithm . . . . .	59
<b>4. Conclusion</b>	<b>69</b>
<b>A. Some Elements of convex analysis</b>	<b>73</b>
<b>B. Fista algorithm</b>	<b>74</b>
<b>C. Fista Algorithm with two coordinates</b>	<b>77</b>

D. Computation of the conjugate of the total variation	80
E. Elements of continuous optimization	82
F. Hausdorff measure	83
G. Projective Geometry	85
H. Epipolar Geometry	86
I. Camera models	89
J. DEM and standard geographical vertical references	92

## List of Figures

1.1. <i>illustration of the HiRISE's mode of acquisition (from [18])</i>	9
1.2. (A) <i>Layout of CCDs on the base plate.</i>	
(B) <i>Reference data on each Detector Chip Assembly (DCA), which consists of the CCD and CPMM (computer processing memory module which in particular, adds header data to the image data as it is transferred to the spacecraft ). Spacecraft motion would be down as shown here, although HiRISE will typically image on the ascending side of the orbit.</i>	11
1.3. <i>2D illustration of the HiRISE Frames</i>	12
1.4. <i>TDI Operation Using a 4-TDI Configuration to Illustrate Charge Accumulation</i>	13
1.5. (a) <i>image of CCD 0, and channel 0</i>	
(b) <i>image of CCD 0, and channel 1</i>	
(c) <i>zoom of a patch of the image of CCD 0, and channel 0</i>	
(d) <i>zoom of a patch of the image of CCD 0, and channel 1</i>	
(e) <i>obtained image after the radiometric calibration and the assembling of the two channels.</i>	
(f) <i>zoom of the previous image.</i>	18
1.6. (a) <i>obtained image after the radiometric calibration and the assembling of the two channels</i>	
(b) <i>the same image obtained after the step "noproj", the transformation applied to the image aims to re-sample it as if it was taken by the CCD 5</i>	
(c) <i>the final image obtained after the concatenation of the images of others strips</i>	18
1.7. (a) <i>image covering a patch of the ground of Mars</i>	
(b) <i>another image covering the same scene</i>	
(c) <i>a digital elevation model</i>	19
1.8. <i>illustration of the principle of triangulation: P is the position of the 3D point, <math>p_l</math> and <math>p_r</math> the corresponding projection on the two views, <math>O_l</math> and <math>O_r</math> are the centers of the cameras that produced the image (<math>E_L</math> and <math>E_R</math> are called the epipoles (more information about this points on appendix H, )</i>	19
1.9. <i>illustration of the occlusion phenomena: The section B cannot be seen by the camera on the right, and the section D cannot be seen by the camera on the left.</i>	20
1.10. (a) <i>illustration of uncertainty on the position of the imaged point: <math>P_i</math> and <math>Q_i</math> have the same projection</i>	
(b) <i>illustration of uncertainty on the position of the corresponding pixel, on the associated image: the point <math>X_D</math> lies on a line</i>	20
1.11. (a) <i>a digital elevation model</i>	
(b) <i>a digital elevation model with shading (the intensity is reversed), the intensity's color is reversed on the image</i>	21

1.12. (a) input image taken by the camera HiRISE	
(b) the same image after ortho-rectification: it is not visible in the images, but the distances are not conserved, and the covered zone is not rectangular anymore. . . . .	22
1.13. overview of the main execution steps of s2p . . . . .	23
1.14. (a) reference image (input $u$ on 1.13)	
(b) secondary image (input $v$ on 1.13) . . . . .	23
1.15. (a) the four red lines represent four epipolar curves of specific pixels, on the associated image.	
(b) the point $x$ and $x'$ are corresponding point on the two images, but $x'$ doesn't belongs to the epipolar curve computed by the telemetry data. . . . .	24
1.16. In the pinhole case (left image) the epipolar plane (appendix H) defines a correspondence between epipolar lines. In the push-broom case (right image) the projection of a 3-space ray on the secondary view generates a ruled quadric. The projection of this quadric on the reference view contains many epipolar curves: epipolar curves are not conjugate . . . . .	25
1.17. Effect of attitude errors on the localization function. The figure on the left shows a pitch error of $\epsilon$ . To first order, this induces a ground displacement of the sensor projection of $a\epsilon$ , where $a$ is the flying altitude of the sensor. The figure on the right shows a yaw error of $\epsilon$ . It induces a maximal ground displacement of $\frac{D}{2\epsilon}$ , obtained for the sensor endpoints, where $D$ denotes the swath width. . . . .	26
1.18. (a) digital elevation model on the whole input image $u$ ??	
(b) digital elevation Model on a limited zone . . . . .	28
2.1. illustration of the traditional tie points and the inter strip (inter-CCD) tie points . . .	32
2.2. illustration of the jitter observation . . . . .	33
2.3. illustration of the PEDR points . . . . .	34
2.4. (a) the DEM of HiRISE (produced by the s2p algorithm)	
(b) the DEM of MOLA (chooses in a region that contains the DEM of s2p [?]) . . . . .	35
2.5. the image at the left corresponds to the DEM of HiRISE after the application of the Gaussian filtering, and the DEM of the left corresponds to the DEM of Mola . . . . .	36
2.6. PEDR points represented in 3D, every (nearly) straight line, on the longitude/latitude plane represents an orbit . . . . .	37
2.7. profile of maximal inter-correlation using one orbit . . . . .	38
2.8. profile of maximal inter-correlation using all the orbits . . . . .	38
3.1. (a) First image of the scene Tsukuba from the data set Middlebury	
(b) Second image of the scene Tsukuba from the data set Middlebury	
(c) Ground Truth for the disparity image from the data set Middlebury	
(d) Computed Disparity image by the use of dynamic programming . . . . .	41
3.2. illustration of the cost aggregation method of SGM . . . . .	41
3.3. Difference between the MGM and SGM algorithm . . . . .	42
3.4. Star associated graphs associated in SGM to two adjacent pixels . . . . .	42

3.5. (a) First rectified image for the disparity computation	
(b) Second rectified image for the disparity computation	
(c) Disparity Image computed by MGM	
(d) Disparity Image computed by SGM . . . . .	43
H.1. epipolar geometry . . . . .	86
H.2. point correspondence geometry . . . . .	87
I.1. Pinhole camera geometry. . . . .	89

# 1. The 3D reconstruction problem: application to Mars images

## 1.1. The HiRISE Camera

This section introduces briefly the HiRISE Camera, the technology of acquisition, its architecture, and its location on the spacecraft.

### 1.1.1. Description of the components

The HiRISE camera doesn't take a single image of an entire scene all at once, but instead builds up a picture by "sweeping" a swath of Mars. As the MRO (Mars Reconnaissance Orbiter) spacecraft (the spacecraft to which it belongs) speeds over the surface of Mars, the camera builds up the picture by grabbing one row of pixels at a time (figure 1.1). This model of cameras is known as a **push-broom camera**, and it induces a substantial difficulty for the 3D reconstruction, because the position of the camera is not fixed during the whole acquisition. Nevertheless, the software presented after this section can handle the intrinsic difficulties with this kind of cameras.

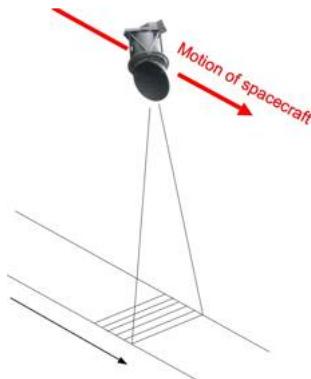


Figure 1.1.: illustration of the HiRISE's mode of acquisition (from [18])

The HiRISE Camera sweeps the imaged patch of interest on the ground (thus comes the name of push-broom camera), and can form an image arbitrarily long on the direction parallel to the movement of the device , which is limited only by the limited camera's internal memory . The contained data is afterward radio-backed to earth, and freed for other acquisitions.

The camera is characterized by high SNR (signal to noise ratio) due to its technology of acquisition and large image size in addition to high resolution. It's thus perfectly suitable for detailed exploration of Mars' ground.

Its focal plane contains a total of 14 charge-coupled device (CCD) arrays (figure 1.2), each of which operates as a 2048-pixel-wide line (of size  $12 \times 12 \mu m$ ) detector to build up an image in push-broom mode, with up to 128 lines of time delay and integration (TDI) (a technology of acquisition explained in the next subsection 1.1.2) to ensure a high SNR ratio.

The CCDs are staggered to provide a continuous coverage on the cross-track direction (dimension perpendicular to the motion of the camera) of 20,048 pixels wide without gap (1.2).

Ten of the detectors, filtered to accept only red wavelengths, overlap slightly in the cross-track direction

Additional detectors sensitive to blue-green and near-infrared wavelengths permit false-color imaging of the central 4000 pixels of the swath, but for the purpose of this project we are exclusively interested in the images produced by the previous CCDs: the wide image provided by the red detectors is sufficient for stereo-analysis.

Concerning the along-track length of images that can be acquired, it depends on the number of CCDs used, pixel binning (next subsection) and data compression, but can be as much as 120,000 lines at full resolution.

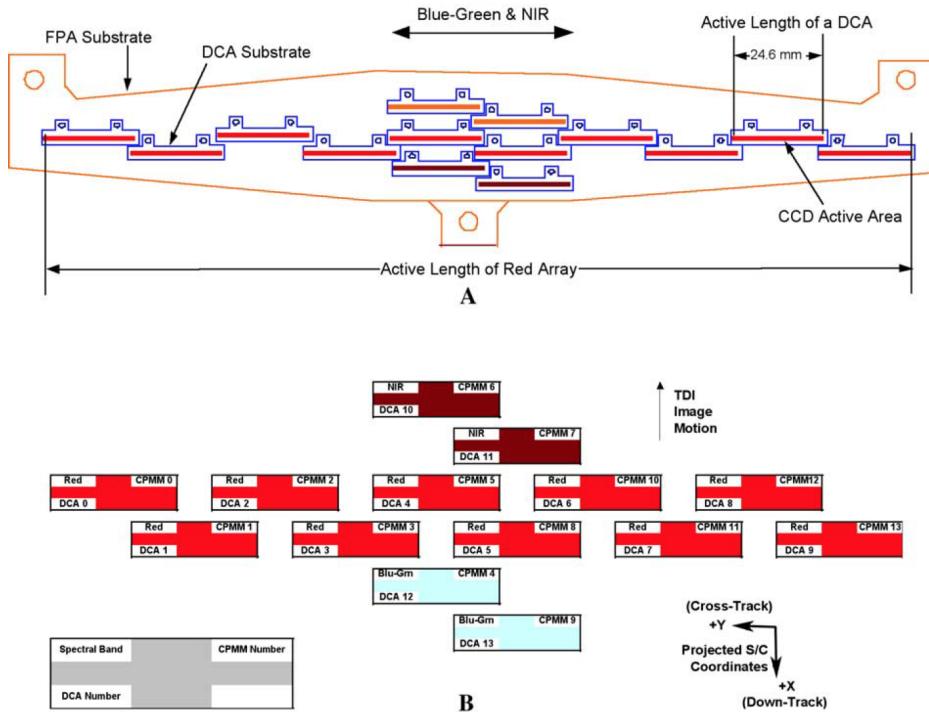


Figure 1.2.: (A) Layout of CCDs on the base plate.

(B) Reference data on each Detector Chip Assembly (DCA), which consists of the CCD and CPMM (computer processing memory module which in particular, adds header data to the image data as it is transferred to the spacecraft). Spacecraft motion would be down as shown here, although HiRISE will typically image on the ascending side of the orbit.

The HiRISE Camera is allowed to rotate with respect to the spacecraft. In fact, there are multiple frames defined in the spacecraft. This makes the position of the pixels, with respect to a fixed frame, and the whole relative geometry of the components, not straightforward to get. These informations are present partially in the raw images as meta-data, and also in other files called kernels (see <https://naif.jpl.nasa.gov/pub/naif/MRO/>).

The page [https://naif.jpl.nasa.gov/pub/naif/MRO/kernels/fk/mro\\_v15.tf](https://naif.jpl.nasa.gov/pub/naif/MRO/kernels/fk/mro_v15.tf) gives an insight about the multiple frames used. As we are interested in the HiRISE Camera, only a very limited number of frames would be of interest (figure 1.3). The different frames have numerical identifiers, but also specific names:

#### **-MRO\_HIRISE\_LOOK\_DIRECTION** (The HiRISE look direction):

It is nominally co-aligned with respect to the spacecraft frame ( $X_{sc}, Y_{sc}, Z_{sc}$ ), it has its  $+z$  direction ( $+Z_{hd}$ ) along the camera bore-sight, nominally defined as the direction of the pixel 0 of the detector pixel 0 of CDD 5/Chanel 1 (every CCD, is composed of two contiguous channels that acquire the image separately),  $Y$  direction ( $+Y_{hd}$ ) along the detector lines, and  $+X$  direction ( $+X_{hd}$ ) completing the right-hand frame.

#### **-MRO\_HIRISE\_OPTICAL\_AXIS** (The HiRISE optical axis frame)

It is the frame to which we directly links the index of pixel to its physical position. It is also fixed with respect to the MRO SPACECRAFT's frame and is nominally rotated from it by +0.45 degrees about +Y axis, it has +Z along the camera optical axis, +Y along the detector lines, and +X completing the right hand frame. Only its Z axis is represented on the figure (+Zh\_ oa)

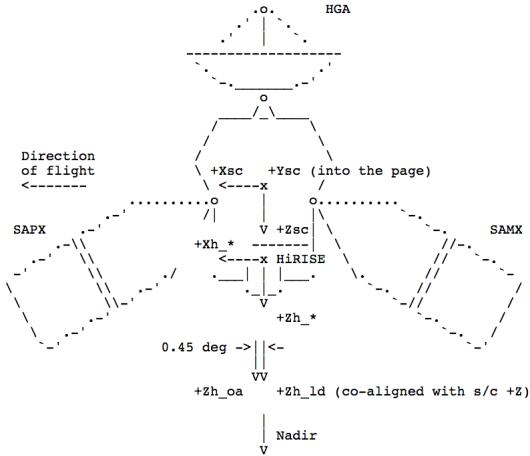


Figure 1.3.: 2D illustration of the HiRISE Frames

### 1.1.2. Acquisition of the image

A major challenge to high-resolution imaging for a fast ground track velocity is how to collect enough photons to make a high-quality image. One of the methods used for the image acquisition is the TDI (time delay integration). It is a solution for which each small patch of the ground is imaged up to 128 times and the signals are summed, to increase the exposure time, allowing to obtain both very high resolution and a high signal-to-noise ratio (there are in fact four different choices: 128, 64, 32 and 8, and the CCD's strips are composed of 128 lines of (physical) pixels).

As the spacecraft moves above the surface of Mars, TDI integrates the signal as it passes across the CCD detector by shifting the accumulated signal into the next row of the CCD at the same rate as the image moves (figure 1.4).

The pixel line time for each image is set to match the predicted ground velocity so that charge from one image pixel is sequentially clocked into the next corresponding element in the along-track direction.

Thence, TDI imaging require very precise stability and pointing control by the spacecraft. As each row is read, signals are shifted, the TDI readout should properly match the velocity of the spacecraft otherwise the image will be blurred.

In order to increases the camera's light sensitivity and improves the signal-to-noise ratio, another method is used, it is the **bining**, it is a process that allows charge from multiple adjacent pixels to be combined into one pixel. It makes the pixel brighter, but you have a smaller image. Square pixel

binning can be used in multiple modes (2x2,2x4,8x8,16x16).

In addition to that, the image can be compressed from 14 bits to 8 bits, which is detailed in the PDS file (a file containing the meta data about the image) as a lookup table (a list that associates interval of values in 14 bits to values in 8 bits). The 8 bit- image can be further compressed but with no information loss, by an intern software in the satellite named "FELICS".

Each CCD can have a different choice of binning, TDI, number of lines, and choice of LUT (look up table). The line time must be the same for all CCDs in an image because it is set to match the ground track speed. Also when using FELICS compression (the great majority of Mars Image), the data from all CCDs must first be LUT-compressed to 8 bits.

A "trim" function delays the start of forward clocking for each CCD so that they begin to collect data over the same location in the scene so that the assembled image will have a nearly straight beginning edge (to manage the difference between the setting of the different channels ).

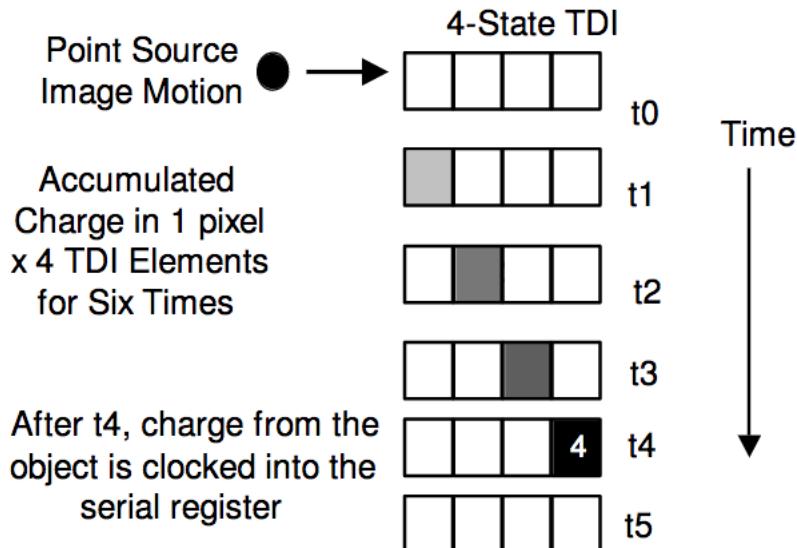


Figure 1.4.: *TDI Operation Using a 4-TDI Configuration to Illustrate Charge Accumulation*

### 1.1.3. Extraction of the position and orientation of a pixel from the optical axis frame

In order to perform 3D reconstruction from a pair of images, we need to obtain the orientation of the camera, or in more precise terms, the direction of light from the imaged patch of the ground to its corresponding pixels. For the terrestrial images, these informations are presents on a specific format that contains the orientation of the satellite. For the image of Mars, these informations aren't present on a simple format.

The extraction of the position of a pixel on the optical axis frame is not straightforward. The procedure is described in the section "A Recipe for Computing Observation Time and CCD Pixel View Direction" from the file "<https://naif.jpl.nasa.gov/pub/naif/pds/data/mro-m-spice-6-v1.0/>

mrosp\_1000/data/ik/mro\_hirise\_v12.ti", and unfolds on several steps, that are briefly summarized here:

#### **-Computation of the time for the middle point of the TDI line block:**

As the pixels of the image are not formed instantly, the first step concerns the computation of the time of acquisition of the pixel, and as the pixel is formed from a sequential acquisition of the same pixel multiple times (TDI method), there is an ambiguity on the time of its acquisition, this ambiguity is not a problem, once we focus on the unique associated patch of the ground.

We choose to consider the acquisition time as the time at the middle of the TDI block, (and at the middle of the corresponding pixel, or group of pixels if the binning mode is used), (for example in the case where we use 128 lines, it is the instant ,when 64 lines have been acquired). The acquisition time depends also in the chosen binning mode, on the line rate and require also to access to the instant of acquisition of the first line of the image. The acquisition time boils down to a linear equation of the line of the image (not precised here).

Thence for each image line, we can associate a time of acquisition, and this time of acquisition results on the orientation of the bore-sight direction (defined above). This information is present on other files, produced after each active phase of the satellite, and is not present on the meta-data of the image.In fact, the bore-sight direction is computed on specific moments, and the value on the moments of acquisition is deduced by interpolation.

#### **-Computation of the physical location of the pixel on the focal plane coordinates**

The HiRISE's image is not formed as a single image, but rather as multiple images formed on the 14 CCD that constitute the camera HIRISE, and each CCD is composed of two channels, for which each part of the image is formed independently.

Knowing the index of the pixel on the image, the ID of the CCD and the channel, and the index on the image of the physical center of the CCD, we can deduce the index of the physical pixel. After we get the 2D dimensional physical location on the CDD, it is possible to get the focal plane coordinates of every pixel of each CCD with respect to the center of the focal plane, by a relation of this form:

$$X = X_0 + \frac{\partial X}{\partial S} S + \frac{\partial X}{\partial L} L$$

$$Y = Y_0 + \frac{\partial Y}{\partial S} S + \frac{\partial Y}{\partial L} L$$

With L referring to the index of the line on the CDD , S referring to the index of the columns, (also named sample). The term  $\frac{\partial}{\partial}$  refers to coefficients that are specific to each CCD, these values are constant and stored on the file "hiriseAddendum003.ti".

#### **-Computation of the ideal focal plane coordinates from distorted coordinates :**

Knowing the physical location (x,y) of each pixel on the detector with respect to the center of the focal plane, we can deduce its location with respect to the frame "MRO\_HIRISE\_OPTICAL\_AXIS":

the location is  $[x, y, f_l]$  with  $f_l$ , the value of the focal length), it is also the direction of the ray converging to the pixel, if there was no distortion (if the light converged on rigorously straight lines).

A model of radial distortion is taken into consideration, for the computation of the exact location of the pixel: The model links the distorted (observed) parameters  $[x, y]$  to the ideal coordinates  $[x_p, y_p]$  by the following equations:

$$r^2 = \sqrt{x^2 + y^2}$$

$$dr = k_0 + r^2(k_1 + k_2 r^2)$$

$$x_p = (1 - \frac{dr}{r})x$$

$$y_p = (1 - \frac{dr}{r})y$$

The location on the frame "MRO\_HIRISE\_OPTICAL\_AXIS" is obtained rigorously as :  $[x_p, y_p, f_l]$

#### **1.1.4. Raw-data extraction for the production of the input Images of the s2p Algorithm**

##### **Description of the Raw-data, and its properties**

A HiRISE observation is contained on a set of files called EDR (experiment data record) products, with each EDR storing the instrument data of a single charge couple device (CCD) channel, which we can divide on three: image data, acquisitions of other sensors on the spacecraft and meta-data (for example the start time of acquisition, the TDI mode, the binning mode ...), with two output channels for each of CCD colocated on the instrument's focal plane array.

Up to 28 EDR products (depending on how many CCDs are commanded to operate) can be generated per observation.

The meta-data contained on the image are not complete, there are other informations contained on other files named "kernels", that contains in particular the position and orientation of the satellite, during its active phase, informations to converts between different clocks, and frames... etc

The extraction of these informations require the use of SPICE, which is an ancillary information system that manage this data. The manual extraction can be very delicate, as some "kernels" contains informations written in binary.

The software "s2p" that would be presented on the next section require as an input, a pair of two images, the images produced by the camera HIRISE, needs therefore to be stitched.

Nevertheless, these images are subjected to the "jitter effect", it consists of small motions of the spacecraft around its nominal pointing and of high frequency, that distort the scanned image.

The causes of jitter effect include spacecraft vibrations associated with solar panels or a particular instrument, thermal changes and others. These motions can have periods as short as 0.01 seconds

(which is the order of magnitude of a line acquisition 100 lines of TDI), and lower than the rate of measurement of the orientation and position of satellite (obtained from gyroscopes and star-tracker cameras, which is done during the active phase of satellite) every 0.1 s (about 1000 image line of the TDI), which is thence insufficient to interpolate the jitter effect on a polynomial expression ([1] and [7]).

It results on effects on the line of pixels it self, in form of blurring in the image, and also on some artifacts and discontinuities on the produced DEM ([11]).

The jitter have effects of the amplitude of 1 pixel, but can exceed more than 2 pixels, with implies a poor quality of the produced DEM (digital elevation model), even in the case of images with good quality, for the most common algorithm. The effects are in fact twofolds; on the quality of the image, and the precision of the attitude (orientation) of the satellite. Thence, the jitter effect is an important feature of the problem.

These distortions will occur at slightly different places on the images from different CCDs, complicating the assembly of the full image. This occurs because the detectors are displaced alternately forward and afterward in the focal plane as shown at the figure 1.2.

The position and orientation of satellite aren't known at each instant, these values are measured, at particular times during an active phase of the satellite. The resolution of pointing has a value of 0.035 m Rad, which correspond to 10m on the ground ([11], [1]).

These trajectories are modeled by second-order polynomials [31], and represent most of the movement of the satellite. It remains after the interpolation, (whose parameters are presents on the meta-data), a non-negligible error on pointing of magnitude, of the order of some pixels, implying that the jitter effect cannot be entirely modeled by a simple polynomial interpolation ([11]).

## Description of the technical solution for the extraction of the images from the Raw Data

A program named "hiedr2mosaic" contained on an other modular software named "ASP" for 3D reconstruction ([23]), but designed to work on images of other planets, apply the whole steps necessary for the extraction of the raw data.

The process is limited to the ten CCD, that acquire images on the red spectrum, and uses the programs of ISIS (Integrated Software for Imagers and Spectrometers), a digital image processing software that can manipulate, the EDR products and the software SPICE, in order to stitch the images handling the jitter effects.

These steps are the following (with their associated command on the ISIS Software) (also represented in figure 1.1.4):

-hi2isis: Conversion of the EDR products to Isis products, the files have an extension of the form ".cub"

-hical: Radiometric calibration correction of the image: In the obtained image after each step, it can be observed that the image intensity is not continuous between the borders of the different channels. The radiometric calibration aims at the removal of the atmospheric effects like illumination variations.

-histitch: assemble the image of two different channels into one image.

-spiceinit and spicefit: includes the data presents on the "kernels" files, to obtain an image, with a more complete meta-data.

-nopproj :project all images (of each CCD) into the perspective of the central CCD number 5 of the camera. The transformation on the image is assimilated to a distortion.

-hijitreg: it registers the overlapping section of two adjacent CCDs (there is at most 48 pixels in the across-track direction, but the section of overlap can be smaller), using a registration algorithm. The default used algorithm is the maximum of intensity correlation. (It should be noted that due to displacements between the CCD detectors, a same row, may be acquired, with a different orientation, on each CCDs, thence the registration may be incomplete, and lies on the smallness of the amplitude of the jitter)

Remark: As the jitter is of high frequency, its effect is not completely removed during this step: the row of the image aren't acquired at the same moment. The jitter effect should be handled in particular during **stereo matching** (discussed below).

-handmos: mosaic the image to a single image.

-cubenorm:normalize the values of the images.

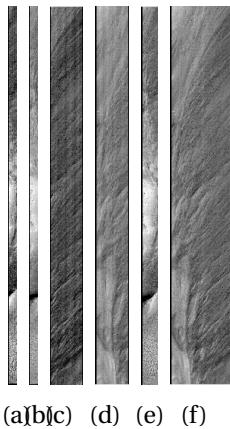


Figure 1.5.: (a) image of CCD 0, and channel 0  
 (b) image of CCD 0, and channel 1  
 (c) zoom of a patch of the image of CCD 0, and channel 0  
 (d) zoom of a patch of the image of CCD 0, and channel 1  
 (e) obtained image after the radiometric calibration and the assembling of the two channels.  
 (f) zoom of the previous image.

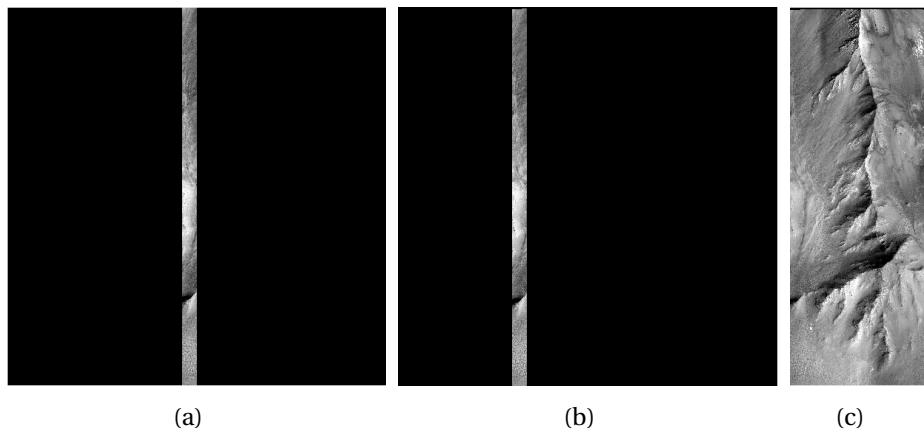


Figure 1.6.: (a) obtained image after the radiometric calibration and the assembling of the two channels  
 (b) the same image obtained after the step "noproj", the transformation applied to the image aims to re-sample it as if it was taken by the CCD 5  
 (c) the final image obtained after the concatenation of the images of others strips

## 1.2. Stereoscopy and topography reconstruction

Stereoscopy is a set of techniques for creating and enhancing the illusion of depth in an image by means of other images of the same scene, in order to produce a perception of depth, which is not exactly the 3D scene of the associated images, but rather a profile of elevation called Digital Elevation Model (DEM) (1.7).

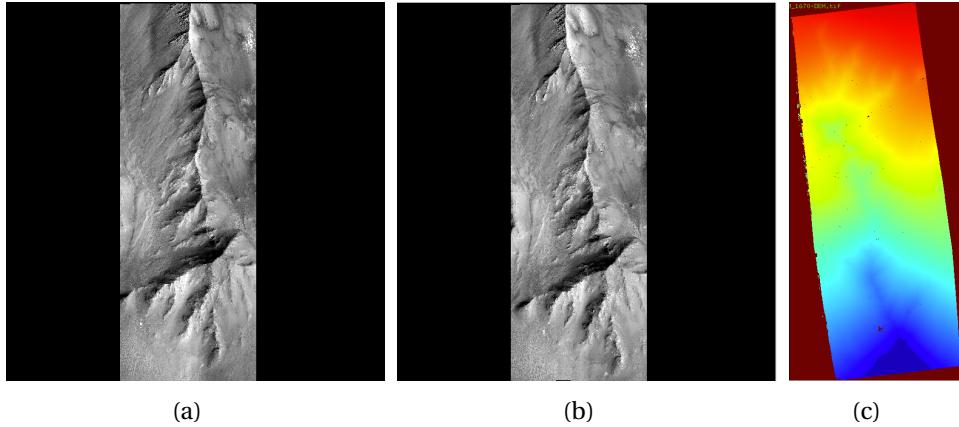


Figure 1.7.: (a) image covering a patch of the ground of Mars  
 (b) another image covering the same scene  
 (c) a digital elevation model

Stereoscopy relies on the principle of triangulation (figure 1.8): considering two different images of the same scene and knowing two corresponding points on the image  $p_l$  and  $p_r$ , we can reconstruct the 3D point  $P$  by following the reverse path of light, from each pixel.

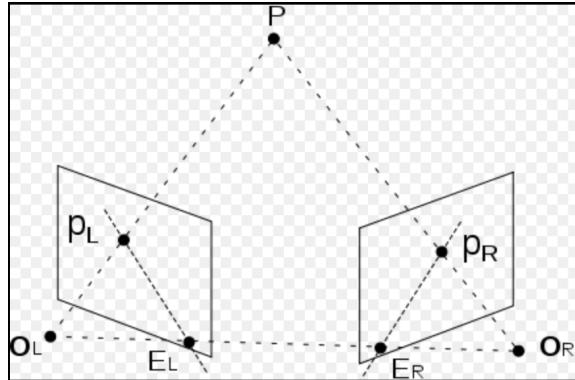


Figure 1.8.: illustration of the principle of triangulation:  $P$  is the position of the 3D point,  $p_l$  and  $p_r$  the corresponding projection on the two views,  $O_l$  and  $O_r$  are the centers of the cameras that produced the image ( $E_l$  and  $E_r$  are called the epipoles (more information about this points on appendix H, )

This problem seems trivial geometrically, but becomes very delicate once we take into consideration the incompleteness and inaccuracy of the data.

One of the common problems of stereoscopy is the problem of occlusions (figure 1.9). Aside from the case of a flat plane, there are parts of the scene that are observed in one image and not in the other. This makes the problem of reconstruction of an exact 3D model impossible to solve completely. This issue is reduced in the case of satellite images because the angles are small between the different views, compared to the distance to the imaged scene.

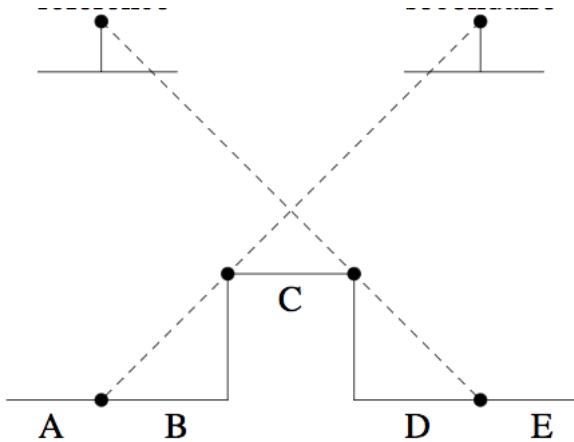


Figure 1.9.: illustration of the occlusion phenomena: The section B cannot be seen by the camera on the right, and the section D cannot be seen by the camera on the left.

For a specific point on the image, there is a one-dimensional uncertainty about the position of the corresponding point on the other image (figure 1.10), this problem, becomes more difficult, once we take into account that the corresponding point may not exist on the images. It is known as the problem of **stereo matching**.

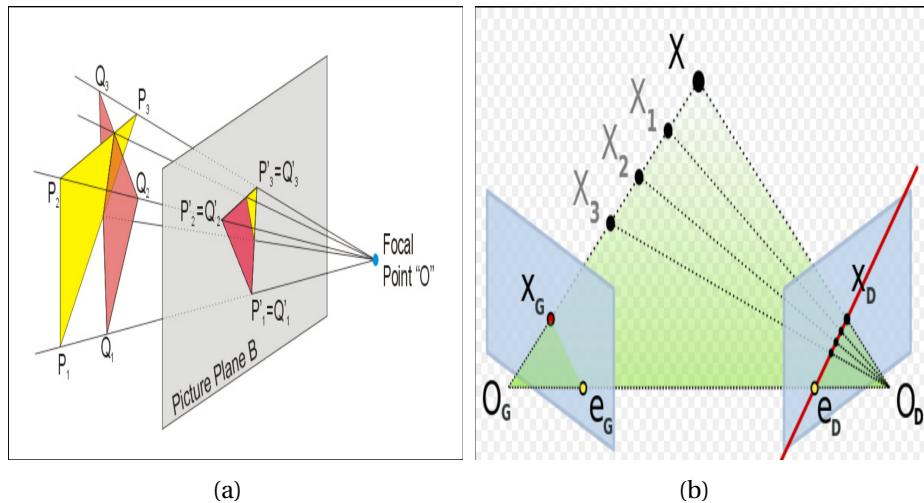


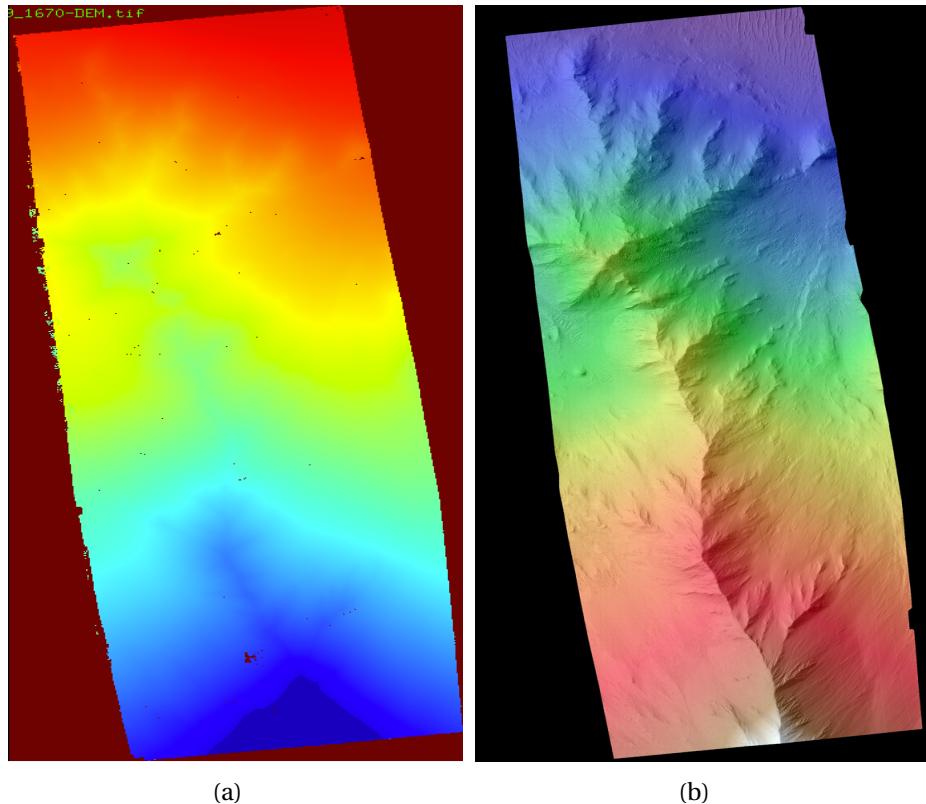
Figure 1.10.: (a) illustration of uncertainty on the position of the imaged point:  $P_i$  and  $Q_i$  have the same projection  
(b) illustration of uncertainty on the position of the corresponding pixel, on the associated image: the point  $X_D$  lies on a line

Another, crucial problem, for stereoscopy, is the accuracy of the orientation of the camera. For the satellite images, a very low modification of the orientation, brings a serious inaccuracy for the problem of **stereo-matching**. The problem of refining the orientation and position of the camera, is the problem of **bundle adjustment**.

The resolution of the problem, and the application of the triangulation, leads to a set of 3D geo-

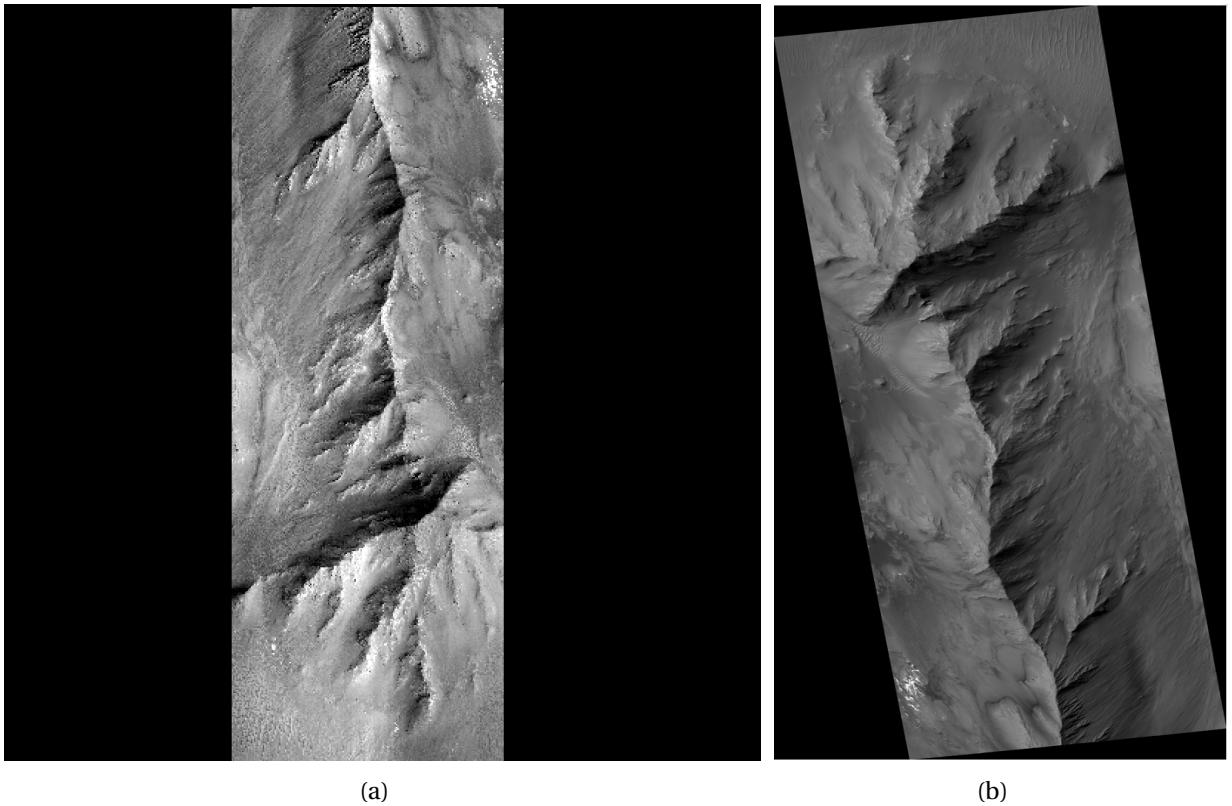
cated points. The set of 3D geolocated point, can thereafter, be projected, on a map (as described on appendix J), which leads to a figure similar to the image (a) of figure (1.11), that constitutes the DEM.

This product is texture-less: the DEM, on each pixel, gives the average value of the height, and does not shows the value of intensity of each pixel, nor the shading. The addition of shading gives, a more realistic stereoscopic product (image (b) of figure 1.11), it can be obtained from the direction of the sun, with respect to the geolocated path of the ground.



**Figure 1.11.: (a) a digital elevation model  
(b) a digital elevation model with shading (the intensity is reversed), the intensity's color is reversed on the image**

In order, to derive a 3D reconstruction, we can merge the DEM, with an ortho-rectified images 1.12, which is an image, the have been projected on a map. It leads to a 3D image (i.e., a 3D dimensional object, composed of voxels defined by a row a column and a height, which depict a 2D surface. Each voxel, that it crossed by this surface, has an intensity in the RED spectrum, the others are null.



(a)

(b)

Figure 1.12.: (a) input image taken by the camera HIRISE

(b) the same image after ortho-rectification: it is not visible in the images, but the distances are not conserved, and the covered zone is not rectangular anymore.

### 1.3. Algorithmic Chain of s2p

The software s2p is a modular and automatic pipeline that produces digital elevations models from satellite images.

The pipeline is illustrated in the figure (1.13), the details about the algorithm are present on the article [9].

The aim of this section is to present the main steps of the algorithm, and the modifications we wish to apply to the algorithm, for this project.

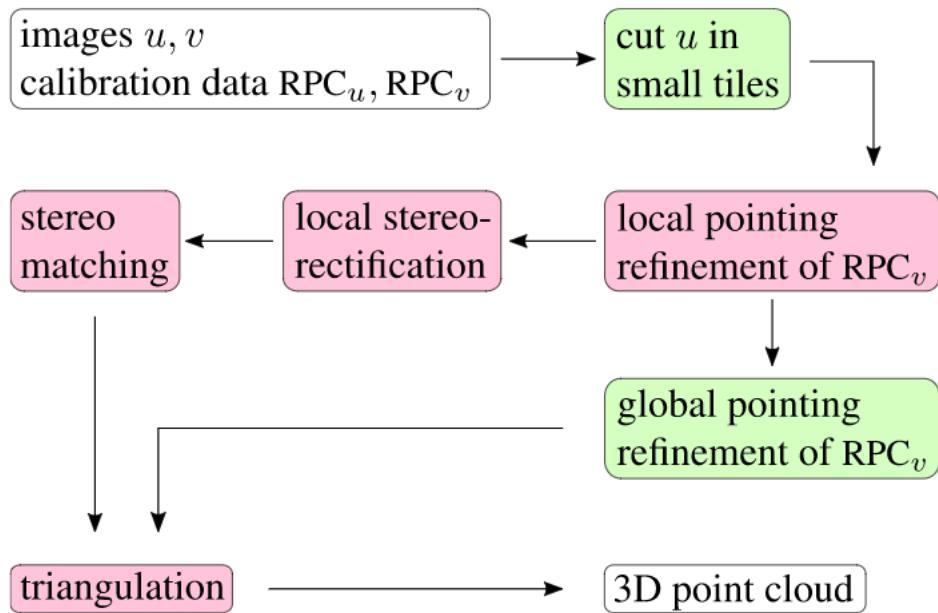


Figure 1.13.: overview of the main execution steps of s2p

For this work, the algorithm have been applied on the same input images. The two processed pairs are shown in figure 1.14. Although they may look similar, there is a small angular movement between the two pairs, that allows the 3D reconstruction. The amount of time between the acquisition of the both images, can last several days or several weeks, but as long as there is no human activity on Mars, we except small variations on the two images.

As shown on the figure 1.13, the input image  $u$  (the order matters) is cut in several rectangular tiles.

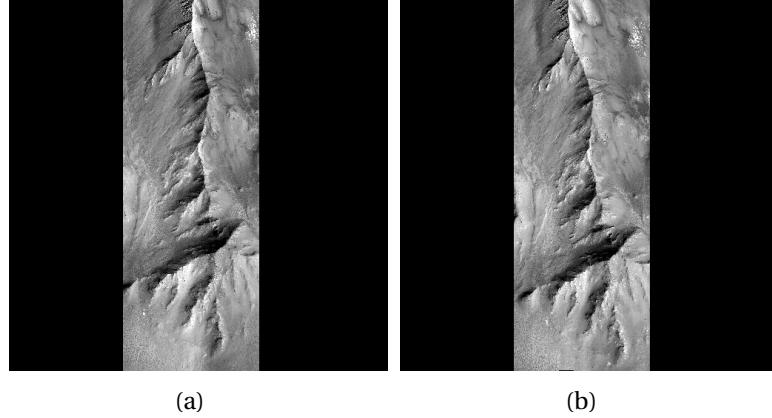


Figure 1.14.: (a) reference image (input  $u$  on 1.13)(b) secondary image (input  $v$  on 1.13)

### 1.3.1. Pointing correction

The RPC file on 1.13 contains informations about the orientation and the position of the satellite during the acquisition phase. It consists of polynomial coefficients, describing the latitude and longitude specific to each pixel as a polynomial interpolation of the orientation and position of the camera measured on the satellite, on specific moments (link between the variables described on 1.1.3)

Using these datas, we define the epipolar curve :

$$epi_{uv}^x(h) = RPC_v(RPC_u^{-1}(x, h))$$

where the calibration data RPC specific to each image links a specific latitude and longitude on the planet (Mars) to a specific pixel on the image. It is well defined once we fix the height.

The epipolar curve is a straight curve in the case of an affine camera model (appendix I). There are two important observations about these curves (in the case of the push-broom cameras). They are approximatively straight lines, almost parallel, and the distance between the epipolar curve of a pixel  $x$ , and its corresponding pixel  $x'$  on the other image is not null and can be in the order of few pixels (figure 1.15).

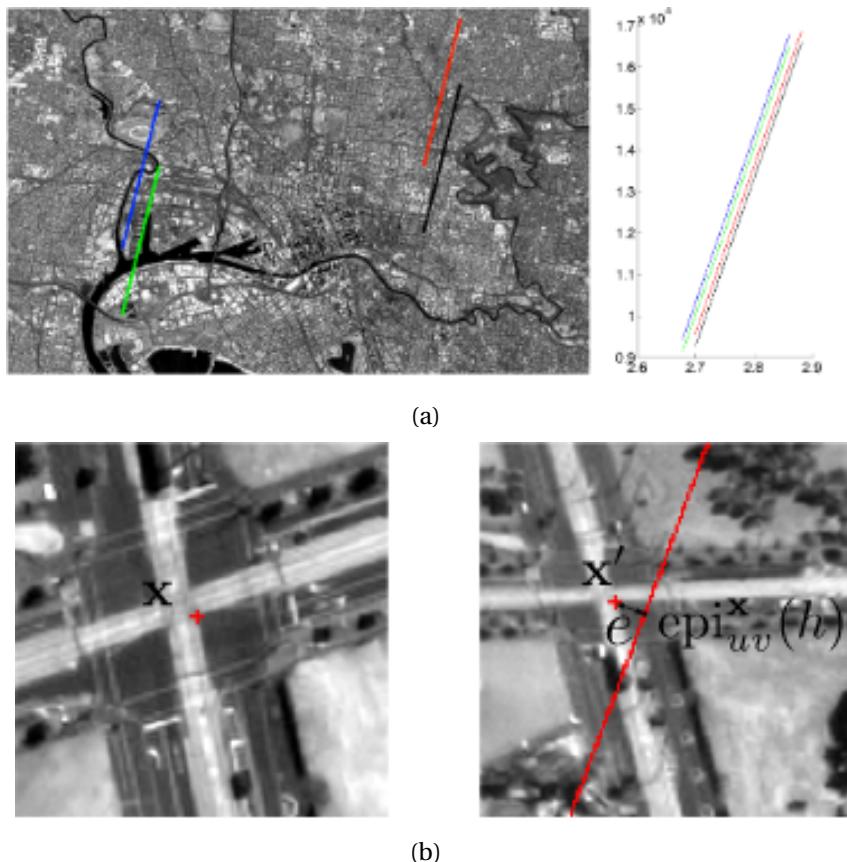


Figure 1.15.: (a) the four red lines represent four epipolar curves of specific pixels, on the associated image.  
(b) the point  $x$  and  $x'$  are corresponding point on the two images, but  $x'$  doesn't belongs to the epipolar curve computed by the telemetry data.

Given a list of correspondence  $(x_i, x'_i)$  (that can be computed by a famous algorithm in image processing: the algorithm SIFT [20]), we define the **relative pointing error**, between two images of the same scene as the mean distance between the epipolar curve of  $x_i$ , and  $x'_i$ , explicitly this value is:

$$\frac{1}{N} \sum_{i \in [|1, N|]} d(x'_i, epi_{uv}^{x_i}(\mathbb{R}))$$

The pinhole camera model, one of the simplest and most common camera model (appendix I), verify the property that the epipolar curves are straight lines, and permits the useful properties explained in the new subsection (appendix H). Though, this model can't be assumed in the case of the push-broom camera, because the epipolar curves are not conjugate (i.e the points contained on the epipolar curve of a point on the reference image, don't have the same epipolar curve on the reference image), as illustrated on the figure 1.16.

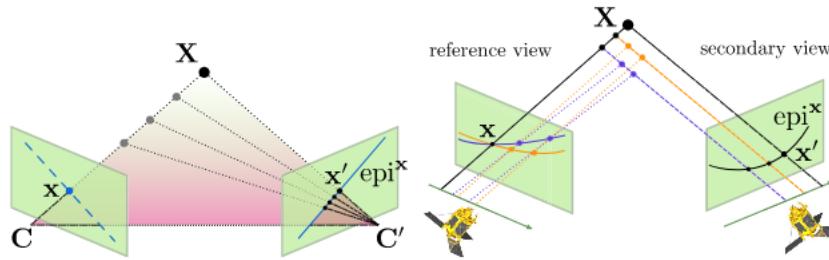


Figure 1.16.: In the pinhole case (left image) the epipolar plane (appendix H) defines a correspondence between epipolar lines. In the push-broom case (right image) the projection of a 3-space ray on the secondary view generates a ruled quadric. The projection of this quadric on the reference view contains many epipolar curves: epipolar curves are not conjugate

Nevertheless, it is shown, experimentally, that for small tiles of the input image (image u of 1.14), the camera can be approximated as an affine Camera (which is a limit case of the pinhole model, that preserves the property of conjugation). In this case, the curve  $epi_{uv}^{x_i}(\mathbb{R})$ , can be approximated by  $Fx_i$ , where F is the fundamental matrix between the two corresponding views (appendix H), that can be estimated by the computed matches.

The relative pointing error, is minimized independently on the tiles of the input image, and explicitly the problem considered on this step, is the following:

$$\min_T \frac{1}{N} d(Tx'_i, epi_{uv}^{x_i}(\mathbb{R}))$$

with T a homography (appendix G).

In fact, the error on local pointing (assuming that the 6 parameters are fitted with polynomials), reside mainly in sensor attitude (the orientation of the satellite), the position parameters are well fitted for the satellite's image of HiRISE([27]), and also for some terrestrial images ([3]).

More precisely, the error resides on the roll and pitch angles as illustrated on the figure (1.17), and

at first order, these errors are approximated on the ground, and on the image plane as translation of the epipolar lines.

Considering the affine approximation (appendix H.1), and the important distance of the imaged point with respect to the camera, the epipolar lines are approximatively parallel.

Thence as a mean first order approximation, we can assume that the relative pointing error, to be modeled by a mean translation  $T$ , applied to all the epipolar lines on the same direction.

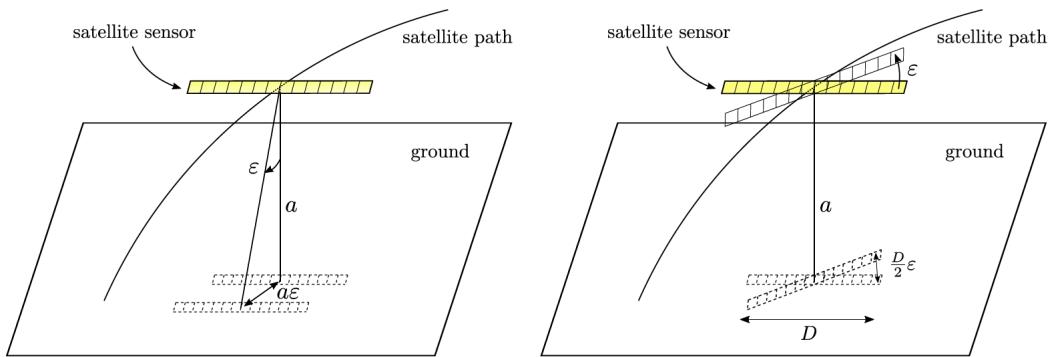


Figure 1.17.: Effect of attitude errors on the localization function. The figure on the left shows a pitch error of  $\epsilon$ . To first order, this induces a ground displacement of  $a\epsilon$ , where  $a$  is the flying altitude of the sensor. The figure on the right shows a yaw error of  $\epsilon$ . It induces a maximal ground displacement of  $\frac{D}{2\epsilon}$ , obtained for the sensor endpoints, where  $D$  denotes the swath width.

This correction is nevertheless relative and not absolute, its reduction will improve the consistency, between the sensor's data and the relative orientation of the sensors, yielding a better estimation of the epipolar lines, thus a better stereo-matching algorithm, but not necessarily correctly geolocated.

In order to have a better registration of the produced 3D model with its actual location on the ground, we need ground control points, which are points measured with a high resolution on the ground, and that can be manually retrieved on the images.

Unfortunately, these points doesn't exists for the planet Mars, but there are other points called the MOLA points, which are localized with high resolution, but cannot be manually retrieved on the images, and would be discussed on the next chapter.

This leads to one of the objective of this internship, which is the correction of the absolute bias error. The error of orientation between the camera (at each time of acquisition), and the ground, i.e the absolute orientation of the Camera.

### 1.3.2. Local Stereo rectification

The stereo matching process is described in the third chapter (about stereo matching). Before its application, a common used technique to simplify the problem, is the stereo image rectification, which simplify the search of corresponding points between the image of a stereo pair. This process, can be done in particular for the affine model (I), the assumption is assumed valid for the tiles, as stated in the last subsection.

Rectifying images is equivalent in practice to the search of two homographies (appendix G)  $H_1$  and  $H_2$  applied to each of the images.

This process may lead to a vertical disparity (perpendicular to the epipolar lines), referred as the epipolar error, that is evaluated by the fundamental matrix  $F$  (appendix H.1). (the error comes in particular from the affine approximation) Explicitly it has the form:

$$\max_{i \in \{1, \dots, N\}} \max d(x'_i, Fx_i), d(F^T x'_i, x_i)$$

Where  $F$ , in contrast with the previous section is computed by virtual matches, which are generated by the (corrected) RPC functions:

$$epi_{uv}^x(h) = RPC_v(RPC_u^{-1}(x, h))$$

$F$  is thence generated by the geometrical information only, and not the information substantial to the image. This metric quantify thence, the affine approximation. It is observed and experimentally proven that the epipolar error, is lower than 0.05 pixel (for terrestrial images).

The affine camera model find its validity on the assumption that during the acquisition, the sensor is assumed to have the same orientation and speed during the acquisition of the tile, thus come the importance of choosing small tiles during the application of the algorithm, for which the jitter effect would be minimized.

### 1.3.3. Global pointing correction and triangulation

The triangulation is the final step of the pipeline (figure 1.13), that produces from the disparity map (set of pixel's correspondences for each tile), and the RPC functions, a DEM.

It corresponds to the value of the altitude for each pixel of the image, producing an image similar to the image of figure (1.18)

In order to ensure the continuity of the produced 3D model, the refinement computed for each tile is averaged and applied to the initial RPC function of the secondary image (step "global pointing refinement" on 1.13), before the triangulation step.

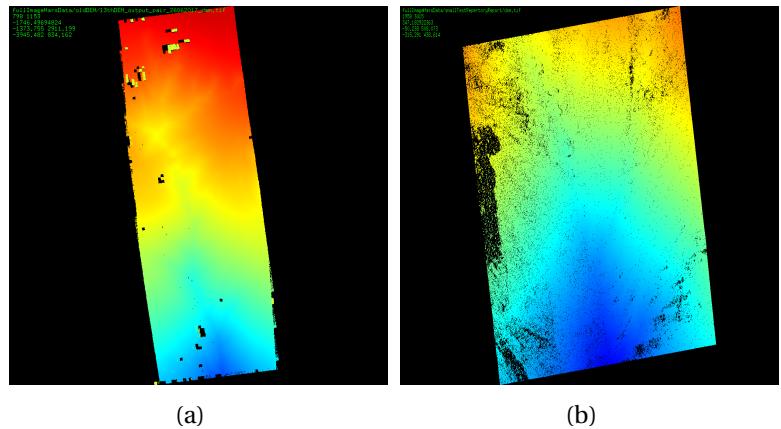


Figure 1.18.: (a) digital elevation model on the whole input image u ??

(b) digital elevation Model on a limited zone

## 2. The problem of Bundle adjustment

### 2.1. Principle

Given a set of images depicting a number of 3D points from different viewpoints, bundle adjustment can be defined as the problem of simultaneously refining the 3D coordinates describing the scene geometry, the parameters of the relative motion, and the optical characteristics of the camera(s) employed to acquire the images, according to an optimality criterion involving the corresponding image projections of all points.

More mathematically, considering a situation in which a set of 3D points  $X_i$  is viewed by a set of cameras with matrices  $P_i$ , and denoting by  $x_{ij}$ , the coordinates of the j-th point as seen by the i-th camera. We wish to solve the following reconstruction problem: given the set of image coordinates  $x_{ij}$ , find the set of camera matrices  $P_i$  ( where the camera matrix is defined in appendix I) and the points  $X_j$  such that:  $P_i X_j = x_{ij}$ .

Without further restriction on the  $P_i$ , such a reconstruction is a projective reconstruction (i.e For H homography, th points  $H X_i$  and the camera matrices  $P_i H^{-1}$  constitutes a second solution to the problem ).

The problem of bundle adjustment can't be reduced to this form, because the measurement  $x_{ij}$  are pixelised, their precision depends not only the resolution of the image, but also of the precision of the used algorithm for theirs extraction (one popular algorithm in the image processing community , that resolves this problem is the SIFT algorithm [20], this algorithm find more precisely, corresponding points in multiple views).

The image measurements are thence always noisy, and for each camera matrix  $P_i$ , point  $X_j$ , and eventual additional calibrations parameters  $C_c$ , we can define the **feature predictive error** :

$$\Delta x_{ij}(C_c, P_i, X_j) \stackrel{\Delta}{=} \underline{x}_{i,j} - x(C_c, P_i, X_j) \quad (2.1)$$

where  $\underline{x}_{i,j}$  is the measured value, which is also called an observation, and  $x(C_c, P_i, X_j)$  is its associated predictive model.

A predictive model, is generally an image projection, but in a more general framework, it can also be a 3D point, or every value that can be measured, and that depends of the values  $C_c, P_i, X_j$ .

We can replace, for example,  $x_{i,j}$  by a point on the space  $x(C_c, P_i, X_j)$  by a point on the space that depends on the cameras, and that should match the measured 3D point.

The first important feature of the problem is the optimal criterion, which constitutes the model of the problem, it is a function f, that depends on the "feature predictive error", for every point and

camera, and whose minimum should represents mostly the characteristics of the unknown solution.

As the measured values  $\underline{x}_{i,j}$  are constant during an optimization scheme, it can be represented as a function "f" of the "state vector" =  $x = (\underline{x}_{i,j})_{i,j}$  (which is concatenation of the predictive model for the various cameras and space point).

The second important feature of the problem, is the optimization scheme. The cost function may be as simple as a squared sum of errors, it's still a none obvious problem to resolve, because of the parameter space, which is generally a non linear manifold: a large Cartesian product of projective 3D feature, 3D rotation, and camera calibration manifolds, with nonlinear constraints: we do not minimize the function f as a function of the state x, but as a function of the parameters  $(C_c, P_i, X_j)$ , and in particular the state vector doesn't cover all the space uniformly.

As the problem is generally non-convex, the usual methodology in bundle adjustment is to provide, a good initialization, then to linearize the cost function f around, this state, to resolve a linear model. There is various method, in the state of Art, that allows a good initialization, in the case of Affine Camera ,and for a cost function f, equal to:

$$f(x) = \sum_{i,j,c} (\Delta x_{ij}(P_i, X_j))^2 = \sum_{i,j,c} \|\underline{x}_{j,nh}^i - (P_i X_j)_{nh}\|^2 = \sum_{i,j,c} \|\underline{x}_{j,nh}^i - (M^i X_{j,nh} + t^i)\|^2$$

Where  $_{nh}$  holds for non homogeneous coordinates, and the last equality comes from the form of an affine camera.

As an example, the factorization algorithm resolves the minimization problem by resolving the equations:  $\frac{\partial}{\partial t_0^i} \sum_{i,j} \|\underline{x}_j^i - (M^i X_j + t^i)\|^2 = 0$ , and  $\frac{\partial}{\partial X_{j_0}} \sum_{i,j} \|\underline{x}_j^i - (M^i X_j + t^i)\|^2 = 0 \forall i_0, j_0$ .

The resolution of these equations is reduced to the resolution of a linear system (section 18.2 of [15]), which involves the computation of the SVD decomposition, which can be done by factorization algorithm or iterative algorithms.

Though, in our case, the initialization algorithm, is not necessary , as we have an initial estimate of the orientation of the HiRISE Camera, from gyroscopes and star-tracker cameras.

In the bundle adjustment framework, we need equations that links the ground point  $X_j$  to the image coordinates  $x_{i,j}$ , these equations have the following form:

$$\begin{aligned} x_{i,c} + f \frac{a_{11,c}^i (X_i - X_i^c) + a_{12,c}^i (Y_i - Y_i^c) + a_{13,c}^i (Z_i - Z_i^c)}{a_{31,c}^i (X_i - X_i^c) + a_{32,c}^i (Y_i - Y_i^c) + a_{33,c}^i (Z_i - Z_i^c)} &= 0 \\ y_{i,c} + f \frac{a_{21,c}^i (X_i - X_i^c) + a_{22,c}^i (Y_i - Y_i^c) + a_{23,c}^i (Z_i - Z_i^c)}{a_{31,c}^i (X_i - X_i^c) + a_{32,c}^i (Y_i - Y_i^c) + a_{33,c}^i (Z_i - Z_i^c)} &= 0 \quad (2.2) \end{aligned}$$

,where  $(x_{i,c}, y_{i,c})$  is the pixel's position of the i-th tie point in the view c (=1 or 2),  $(X_i^c, Y_i^c, Z_i^c)$  is the ground coordinate of the camera center (the origin of the HiRISE optical frame), for each tie-point i,  $(X_i^c, Y_i^c, Z_i^c)$  is the ground coordinate of each tie-points , $(a_{i,j,c})_{i,j}$  is the rotation matrix formed

by the sensor pointing angle, and  $f$  is the focal length of the camera. These equations are called observations equations, they must be linearized, in order to be resolved.

In the case of a push-broom camera, we have additional equations, that models the positions of the camera centers as polynomial, equations of the time,:;

$$\begin{cases} X_i^c = a_0 + a_1 t + a_2 t^2 + a_3 t^3; & \omega_i^c = d_0 + d_1 t + d_2 t^2 + d_3 t^3 \\ Y_i^c = b_0 + b_1 t + b_2 t^2 + b_3 t^3; & \phi_i^c = e_0 + e_1 t + e_2 t^2 + e_3 t^3 \\ Z_i^c = c_0 + c_1 t + c_2 t^2 + c_3 t^3; & \kappa_i^c = f_0 + f_1 t + f_2 t^2 + f_3 t^3 \end{cases} \quad (2.3)$$

These equations are also called pseudo-observations. Normally the coefficients should be defined for each CCD, but since the CCDs are fixed to each other , it is sufficient to consider one strip as the reference, and deduce the others, by the offsets.

It is observed in ([27] and [1] ) that the polynomial model isn't sufficient to models the jitter effect completely, but is sufficient to compute the position of the satellite, with a high accuracy. Thence, a more rigorous approach is to consider, the following model:

$$\begin{cases} X_i^c = a_0 + a_1 t + a_2 t^2 + a_3 t^3; & \omega_i^c = d_0 + d_1 t + d_2 t^2 + d_3 t^3 + \Omega(t) \\ Y_i^c = b_0 + b_1 t + b_2 t^2 + b_3 t^3; & \phi_i^c = e_0 + e_1 t + e_2 t^2 + e_3 t^3 + \Phi(t) \\ Z_i^c = c_0 + c_1 t + c_2 t^2 + c_3 t^3; & \kappa_i^c = f_0 + f_1 t + f_2 t^2 + f_3 t^3 + K(t) \end{cases} \quad (2.4)$$

where we added the error subsequent to the polynomial interpolation.

Starting from the initial value of telemetry data , the optimization process is the following.:

After the measurement of the tie-points on the different views, we extract the line coordinates,from which we derive the six parameters 2.3 (from the telemetry data), which added to the calibrated interior orientation, gives a rigorous geometric model, to compute the approximate ground coordinates of the tie points (by simple space intersection), via the collinearity equation 2.2 (but with the previous value of the matrix rotation, the initial parameters are given by the telemetry data), this leads to equation of the form 2.2, that should be linearized then resolved, then new values of camera center's coordinates and orientation are obtained from which we deduce new values of the polynomial model 2.3.

In the particular case of the HIRISE Camera, there is two types of tie points, the traditional tie points, (one on each views), and the inter-strip tie points that are located in the same view of the HIRISE Camera, in the overlapping section between two CCDS (figure 2.1). What it interesting about this additional tie points, is that two corresponding points (that can be determined by the SIFT algorithm), are imaged in two different times, because the CCDs of HIRISE are arranged with a certain interval along the along-track direction (figure 1.2 and 2.2), hence the elimination of the disagreement between overlapping CCDs can contribute to the removal of the jitter effect.

The "bundle adjustment" between the two HiRISE Stereo images, is already handled by the "s2p"

algorithm. In the subsection , the CCDs are formed by the registration of the two overlapping CCDs, which leads to information loss, but simplify the adaptation of the s2p algorithm, and preserves its pipeline. And the processing of the images, is handled as a simple push-broom camera, formed of one CCD.

Concerning the geometric inconsistencies with the Mars' ground, it is not handled by the algorithm, and it is one of the objectives of this internship to study the resolution of an extended bundle adjustment's problem, which would lead to an absolute positioning of the HIRISE's DEM (digital elevation model), with the ground.

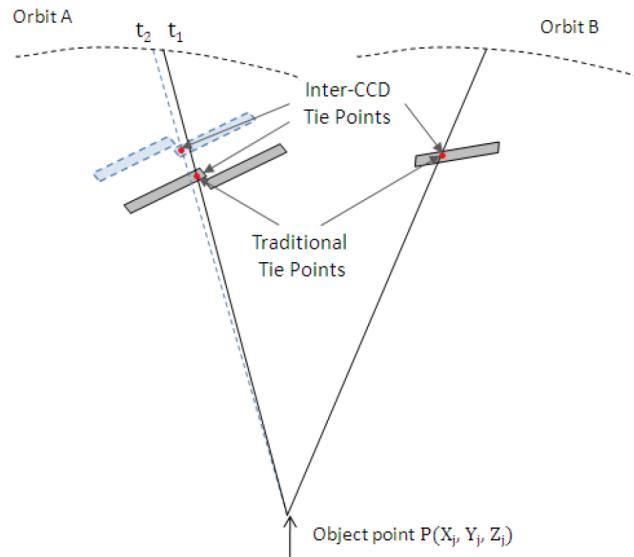


Figure 2.1.: illustration of the traditional tie points and the inter strip (inter-CCD) tie points

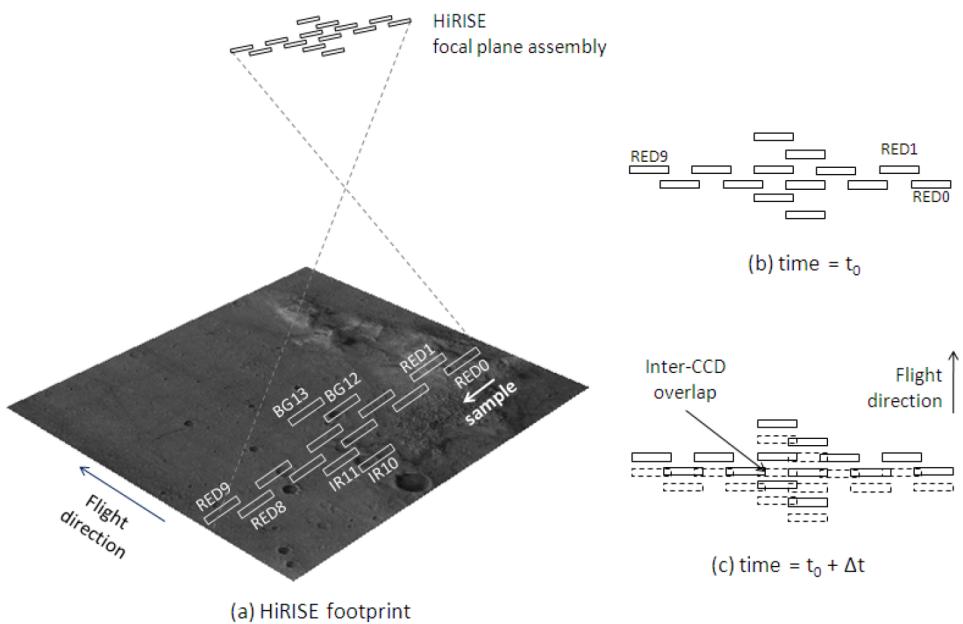


Figure 2.2.: illustration of the jitter observation

## 2.2. MOLA Point

There is no ground control point on the planet Mars, but there are points for which the altitude and positions, are known with high accuracy. These points are measured by an optical remote sensing instrument, the "Mars orbiter laser altimeter" (MOLA), loaded on another spacecraft, the Mars Global Surveyor (MGS).

A laser altimeter is an instrument that measures the distance from an orbiting spacecraft to the surface of the planet that the spacecraft is orbiting. The distance to the object can be determined by multiplying the round-trip pulse time by the speed of light and dividing by two.

The MOLA points as measured by the altimeter have a vertical accuracy of 1 meter, and are determined within approximately 160 m footprints ([10] [17]), in linear segments (2.3), and are called the MOLA-PEDR products.

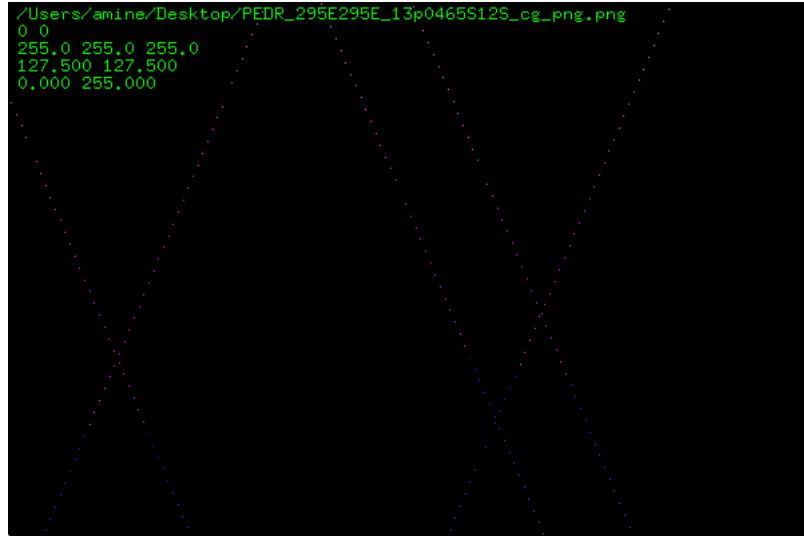


Figure 2.3.: illustration of the PEDR points

Other products exist for the MOLA data, and are generated from the PEDR points, like the MEGDR product, which is a DEM, with low resolution, about 463 meters of grid size, and with an accuracy of 10m, along the vertical.

Under the assumption that s2p, deals correctly with the bundle adjustment of exterior orientation parameters (2.3 in subsection 1.3.1 , the problem of bundle adjustment, can be reduced to the problem of reducing the feature prediction error (2.1) for the MOLA points.

Here  $\Delta x_{ij}(C_c, P_i, X_j) \stackrel{\Delta}{=} X_i^{MOLA} - X_i^{HIRISE}$ , where  $X_i^{HIRISE}$  is the latitude, longitude and height of the MOLA points (from the PEDR product), and  $X_i^{MOLA}$ , is the associated points on the HIRISE DEM that shares the same longitude and latitude.

In the case, where the cost function for bundle adjustment takes the form of the classical squared sum of errors (in fact this sum is always weighted in the general form, in order to avoid badly conditioned system: the weight is chosen with respect of the variance of the data. But in this case of the reduced form, where the observation data concerns only the MOLA DEM, there is one kind of data, then no added weight ), the problem becomes the problem of registering the HIRISE DEM, with the DEM of MOLA, as images.

Furthermore, as it is mentioned in the previous subsection (1.3.1), the parametric error, is mainly a translation between the DEM of MOLA and the DEM of HIRISE (appendix J), thence there is no need for a parametrization, and the pointing correction with the ground, as well as the absolute positioning, can be computed, with a simple registration algorithm.

This approach is certainly less robust and less rigorous, than an approach that would resolve, the whole bundle adjustment system, but is simpler to implement, and permits in particular to test if the MOLA data, are sufficient to register, high-resolution DEM such as those of HiRISE.

One approach for the registration, is to use the MEGDR product to perform a coarse registration between the HiRISE DEM and the MOLA DEM, then to use the PEDR product to perform a finer registration.

In order to test the data, for its potential usefulness of the absolute positioning, we will perform an exhaustive research for the cost, by translating the HiRISE DEM over the two directions, and computing the sum-squared difference with the MEGDR product , then the PEDR product on a finer scale.

Remark: Normally, the translation of the center of a DEM, should be associated with a distortion of the distances (appendix J). For the registration of the Hirise DEM with the mola DEM, we would assume that the distortion is negligible, for the registration with the mola Points, we would move the mola points rather, than the Hirise dem, to perform the interpolation, to avoid heavy computations.

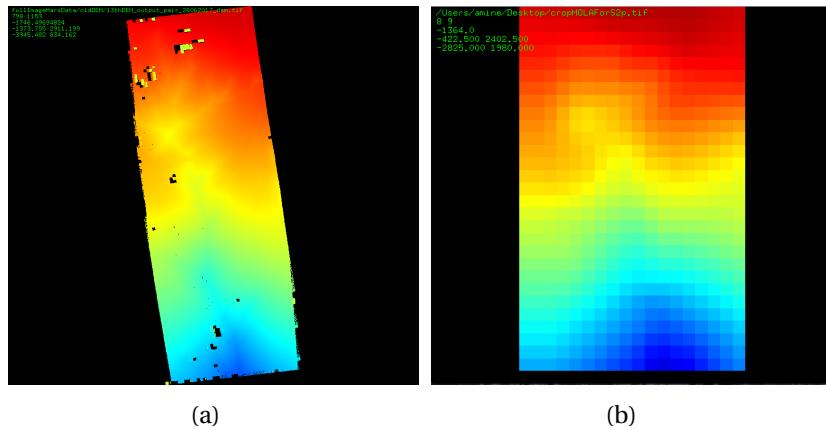


Figure 2.4.: (a) the DEM of HiRISE (produced by the s2p algorithm)  
(b) the DEM of MOLA (chooses in a region that contains the DEM of s2p [?])

## 2.3. Registration between the MOLA data and the HiRISE data

### 2.3.1. Coarse Registration

The first step of the registration consists on the inter-correlation between the DEM of HiRISE and the DEM of MOLA, the classical approach is to reduce the resolution of the DEM of HIRISE, applying a gaussian filter. The application of the gaussian filter involves that the two DEM should border the same region, this is done by the application of the command "gdalwarp" of the software "gdal" (a software that can manage the geographical data, in the form of DEM appendix 2.4).

As it is shown in the figure (image (a) of J), the HiRISE DEM, contains some "NaN" values, which corresponds to pixels for which there is no values, this happens for two reasons, it happens if the SIFT algorithm ([20]) didn't found enough tie points, which may leads to error on some tiles, or more generally during the stereo-matching step (next chapter).

There are two possibilities, the first one, is to replace the gaussian filtering  $g_\sigma * I$ , where  $g$  is the

gaussian function with variance  $\sigma$ , and  $I$  the intensity of the DEM, as an image (the height as a function of the pixel), by the operation  $\frac{g_{\sigma} * (I \cdot 1_{NoNaN})}{g_{\sigma} * (1_{NoNaN})}$ , where  $1_{NoNaN}$ , is equal to 1 when the pixel has a value on the DEM of HiRISE and 0 otherwise.

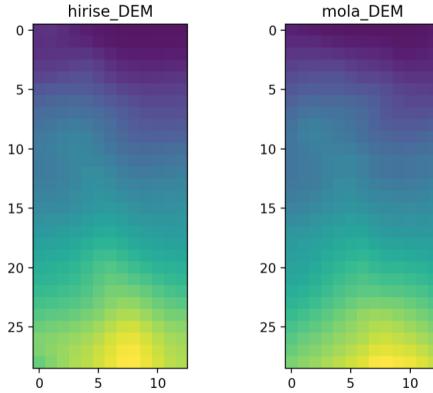


Figure 2.5.: the image at the left corresponds to the DEM of HiRISE after the application of the Gaussian filtering, and the DEM of the left corresponds to the DEM of Mola

In this case, the optimal translation remains the fixed translation.  
The second possibility would be to apply an interpolation on the Data, in order to erase the "NaN" values. This method didn't lead to any change, for the used data.

It would also be possible to apply an interpolation on the MEGDR product (MOLA's DEM) to obtain a finer DEM, in order to improve the resolution, but the MOLA DEM have a resolution of 463m per pixel in along the equator line in the equirectangular projection (appendix J), and the horizontal accuracy is expected to be 200m ([12]). The interpolation didn't change the results.

### 2.3.2. Finer Registration

The next step concerns, the interpolation of the HiRISE DEM's, with the MOLA points. The used cost function for the bundle adjustment is:

$$f(T) = \sum_i |X_i^{MOLA} - (TX^{HIRISE})_i|^2$$

where  $T$  is the translation applied to the HiRISE DEM, and the MOLA points are chosen on a coarse region on Mars that covers, the takes into account all the possible translations used for the inter-correlation (this set of possible translations is chosen approximatively at the same magnitude as the resolution of the MEGDR product) ,which is equal to:

$$f(T) = \sum_i \|X_i^{MOLA}\|^2 + \sum_i \|(TX^{HIRISE})_i\|^2 - 2 \sum_i X_i^{MOLA} (TX^{HIRISE})_i$$

Thence it is sufficient to maximize the inter-correlation  $\sum_i X_i^{MOLA} (TX^{HIRISE})_i$ .

A more robust approach, would be to minimize the normalized inter-correlation:

$$\frac{\sum_i X_i^{MOLA} (TX^{HIRISE})_i}{\sqrt{\sum_i \|X_i^{MOLA}\|^2} \sqrt{\sum_i \|(TX^{HIRISE})_i\|^2}}$$

The DEM points consists of measured points on linear segments (2.3.2 and 2.3)

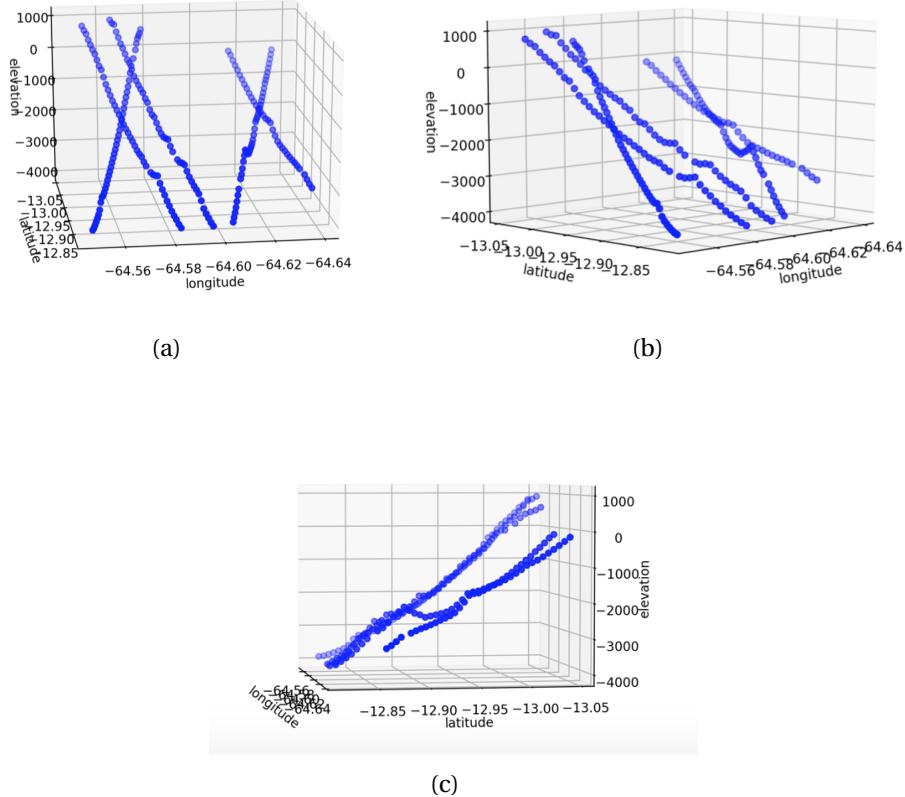


Figure 2.6.: *PEDR points represented in 3D, every (nearly) straight line, on the longitude/latitude plane represents an orbit*

If we choose, one orbit to maximize the inter-correlation, it leads clearly to a poorly discriminated maximum (figure 2.3.2), plotting the inter-correlation, using all the orbits, on the same region, leads to a more discriminative cost function (figure 2.3.2), but we observe clearly that one direction is more discriminative than the other. The more discriminative corresponds to the along track direction (along the rows), which is due to the particularity of the input image (DEM at image (a) of 2.4): the height increase strongly in this direction.

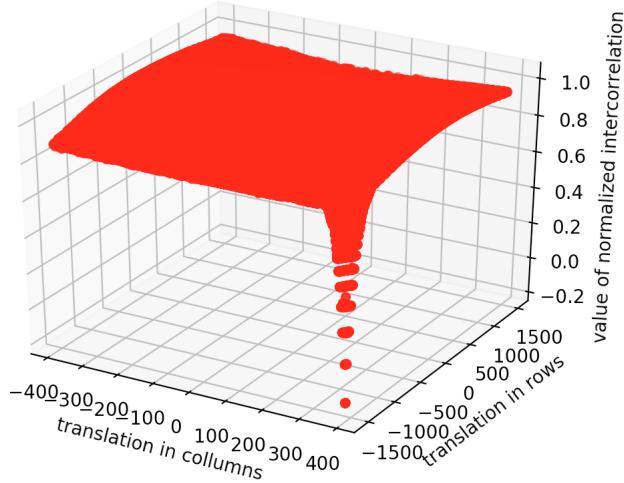


Figure 2.7.: profile of maximal inter-correlation using one orbit

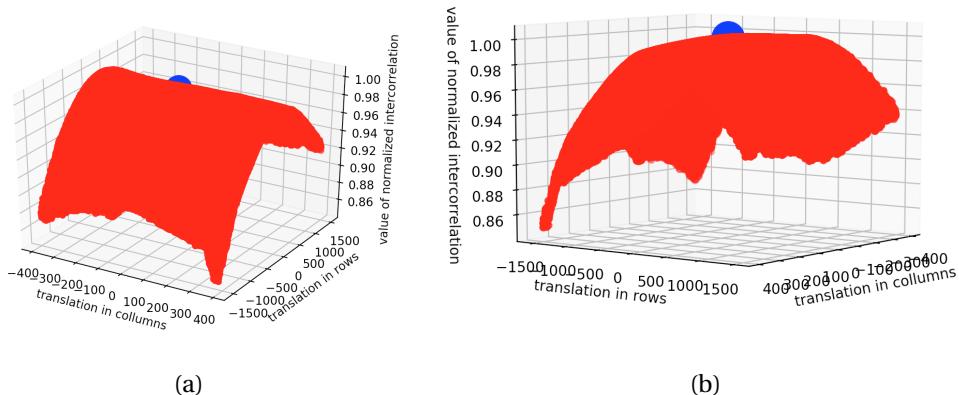


Figure 2.8.: profile of maximal inter-correlation using all the orbits

The particularity of the HiRISE data, is that there is only 2 images for each scene, and the HiRISE Camera covers only particular scene in the ground of Mars. The only available data, that covers the whole planet, is the MOLA data.

One possible way to improve the discriminative behavior of the cost function , would be to use the CTX images (obtained from the satellite) as it is the case in [30], which are images of intermediate resolution between the HiRISE Data and Mola Data, and obtained from another Camera, on the satellite.

### 3. The problem of Stereo Matching (As an optimisation problem)

#### 3.1. Principle and state of the art

In traditional stereo vision, two cameras, displaced horizontally from one another are used to obtain two differing views of a scene, in a manner similar to human binocular vision. By comparing these two images, the relative depth information can be obtained in the form of a disparity map, which encodes the difference in horizontal coordinates of corresponding image points. The values in this disparity map are inversely proportional to the scene depth at the corresponding pixel location.

In the general case, two images of the same scene taken by a camera can have arbitrary orientations. In this case a disparity map can be computed but the two images must be rectified. The process of rectification is explained in appendix H.

Image rectification consists of a pair of transformations applied to the pair of images in order to have the corresponding epipolar lines of the resulting images situated on the same horizontal line. Stereo matching is the process of computing the disparity map. In computer vision ,there are two approaches to handle this problem: the spatially discrete setting and the continuous one.

#### Discrete approach

In the discrete setting, the problem of Stereo Matching is formulated as the minimization of a functional of the form [8]:

$$E(D) = \sum_{p \in I} C_p(D_p) + \sum_{(p,q) \in \xi} V(D_p, D_q), \quad (3.1)$$

with  $D$  the disparity map to estimate, and  $G = (I, \xi)$  a graph that represents the image: the elements of  $I$  (the vertices of the graph) represent the pixels, and the elements of  $\xi$  (the edges of the graph) represent the connections between neighboring pixels.

The disparity  $D$  belongs here to a finite set of values. In a Markovian random field approach, the variable  $D$  is called a *label*. The term  $V$  is a regularizing term. It enforces the smoothness of the solution by penalizing changes of neighboring disparities on the edge set  $\xi$ . One of the most common regularizing terms is the total variation:  $V(D_p, D_q) = |D_p - D_q|$ . This term penalizes strong discontinuities for the disparity map  $D$ .

The term is also called prior term, because it gives an "a priori" about the form of the solution.

Some methods, qualified as *local*, minimize the cost  $C(p)$  independently for each pixel. These methods are generally used only for the initialization of the algorithm. It would be equivalent to

set  $V$  to 0 in these cases. The disparity map can thence have a very irregular form.

The algorithm of the discrete setting needs four elements:

#### -**The cost computation**

It means the definition of the cost function for a pixel  $p$ , displaced by the unknown disparity  $D(p)$ . The pixel  $p$  and the disparity  $D(p)$ , define a position of a pixel in the corresponding image's pair. The cost is what links the data to the model, one of the most simplest is the squared difference between the intensity of the two pixels:  $C_p(D_p) = ||I_1(p) - I_2(p_0 + Dp, p_1)||^2$ , where  $p_0$  is the index of the row, and  $p_1$  is the index of the column, and  $I_1, I_2$ , the intensities, respectively on the first image and the second image.

#### -**The cost aggregation**

It results on the term  $C_p$  of the model. The cost's computation is not always the term chosen on the model. Sometimes, in order to enforce the regularity of the cost, and reduce the effect of noise, we apply convolution of the computation's cost with a kernel, or we average the cost over a support region, which is a particular case.

#### -**Disparity computation and optimization**

One of the important drawbacks of the discrete setting (equation 3.1), is that for common classes of smoothness function, the problem is NP-hard.

And in order to resolve the problems, we can use meta-heuristics to find local minimums of the problem, or propose an algorithm that would compute an approximate solution of the problem.

#### -**Refinements of disparities**

For some applications, the disparity map may be not precise enough (in a low resolution). We can refine it by computing it, at higher spatial resolution, by interpolating the input images before applying the algorithm, but though, for computational time, it can be more efficient to refine the disparity map itself, afterwards. Besides sub-pixel refinements, we need to detect occluded areas, which is generally done by cross-checking (comparing left-to-right and right-to-left disparity maps).

One of the popular algorithm of the discrete approach is SGM. The key idea of the algorithm to overcome the NP-completeness, is that the resolution of the problem over 1 line can be resolved on a polynomial time by performing dynamic programming.

A First idea would be to use the dynamic programming ([2]) along the rows only due to the epipolar constraints  $H$ , but this leads to streaking effects as shown in figure 3.1.

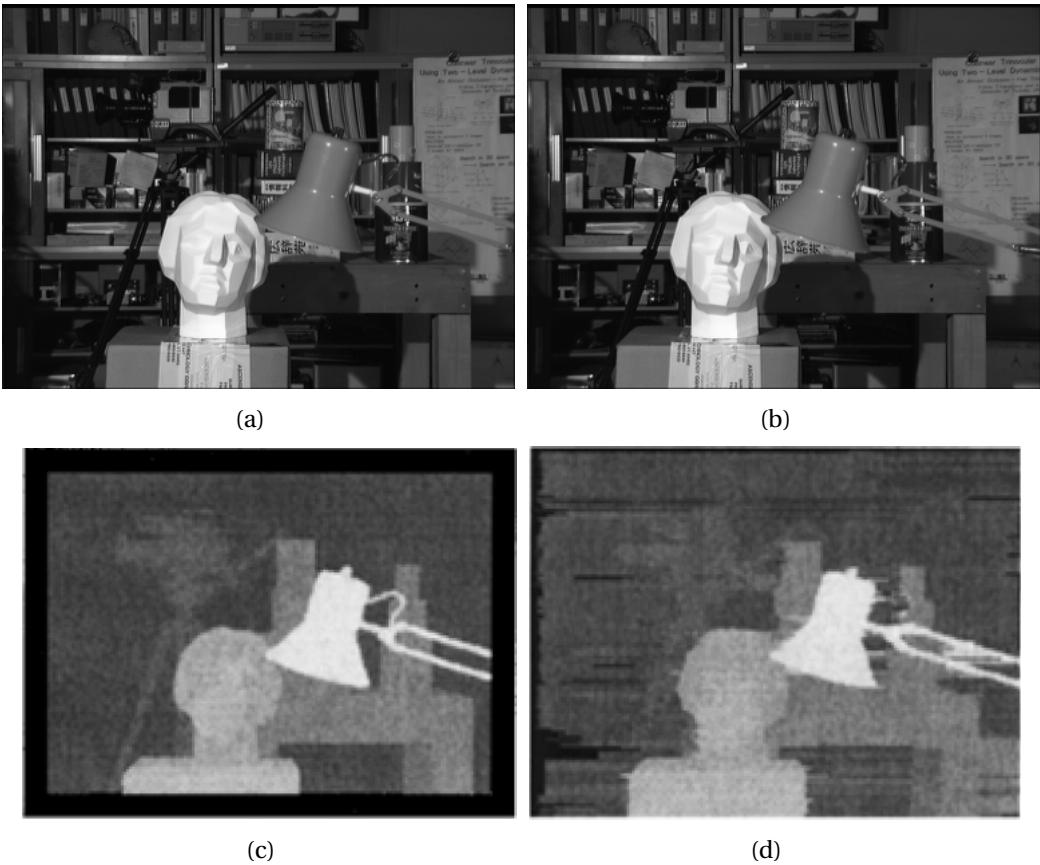


Figure 3.1.: (a) First image of the scene Tsukuba from the data set Middlebury  
 (b) Second image of the scene Tsukuba from the data set Middlebury  
 (c) Ground Truth for the disparity image from the data set Middlebury  
 (d) Computed Disparity image by the use of dynamic programming

What is proposed in the article, is to perform for every pixel  $p$ , with a value of disparity  $D(p)=d$ , 1D- dynamic programming over multiple lines that ends on that pixels as illustrated on the figure 3.2, and then to sum the contribution.

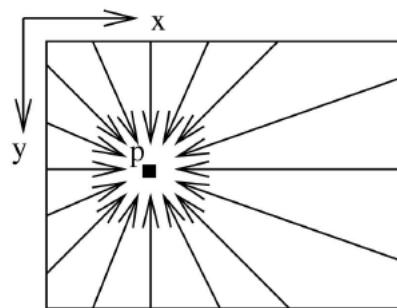


Figure 3.2.: illustration of the cost aggregation method of SGM

Thence, the value of the disparity that minimizes this sum is the chosen one.

Another algorithm, implemented on s2p, called "MGM" proposes ([13]), in the same spirit, an amelioration of the algorithm "SGM" by browsing not a line for each direction but a quadrant as illustrated on figure 3.3, and it imposes more regularity on the solution, and gives less artifacts. This browsing can be done on a Polynomial time by dynamic programming. One explanation of the gain reside in the figure 3.4, that shows that the browsed graph associated to two adjacent pixels are loosely related on the case of "SGM", whereas on "mwm", they are more related.

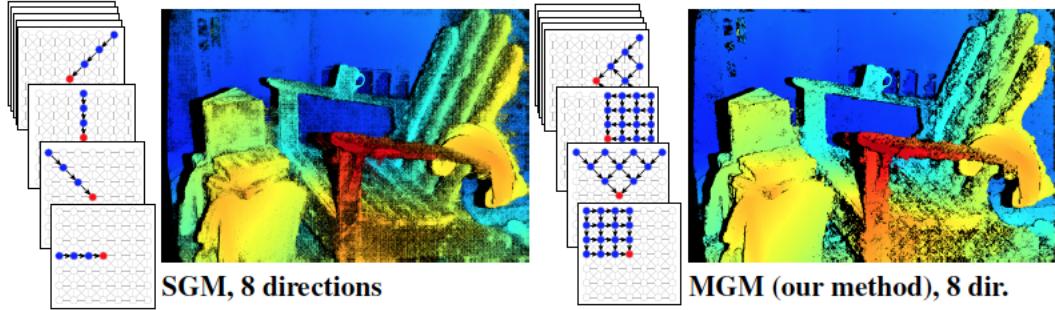


Figure 3.3.: Difference between the MGM and SGM algorithm

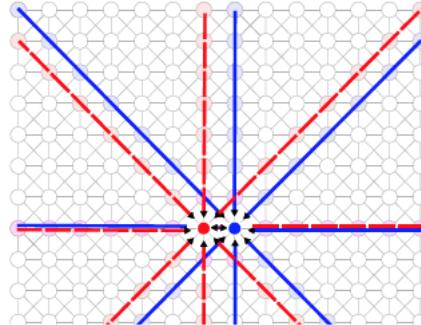


Figure 3.4.: Star associated graphs associated in SGM to two adjacent pixels

The figure 3.1 illustrates, this last improvement on the current data-set's cases. The black holes in the disparity map are issued from the refinement step, more particularly they originate from "cross-checking", that are implemented both on mgm and sgm.

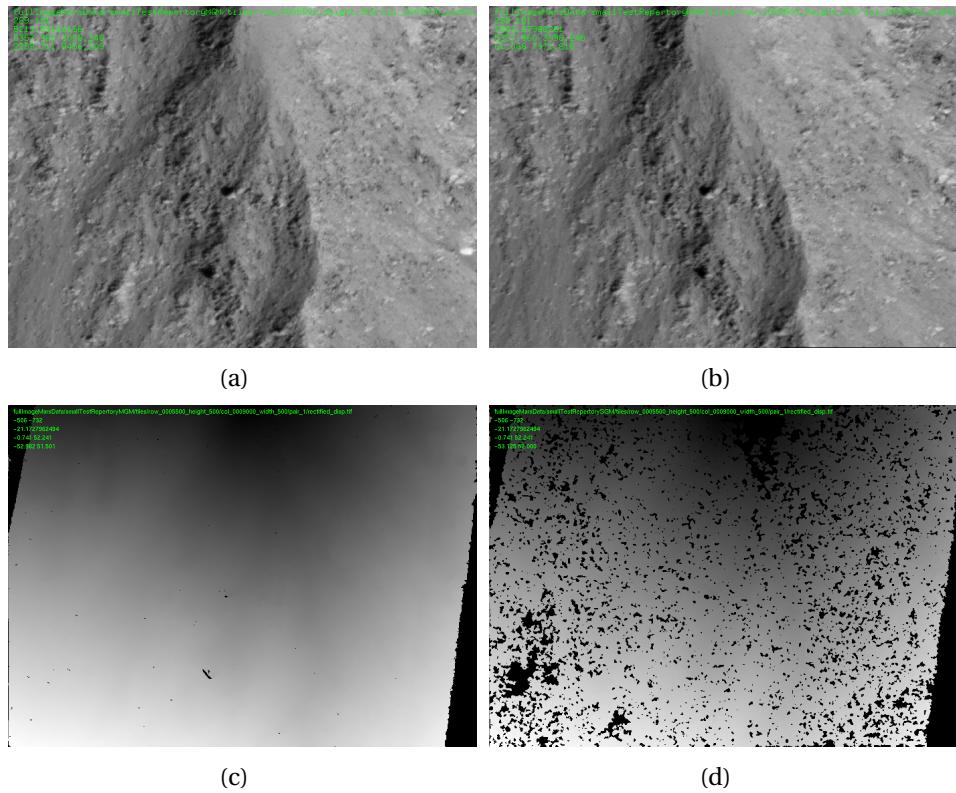


Figure 3.5.: (a) First rectified image for the disparity computation  
 (b) Second rectified image for the disparity computation  
 (c) Disparity Image computed by MGM  
 (d) Disparity Image computed by SGM

## Continuous approach

### Model

In the spatially continuous setting, the problem of stereo-matching is modeled in the following form: a disparity map is a continuous function of  $L^1(\Omega)$ , with  $\Omega$  the domain of the image, that minimizes a functional  $F$  of this form:

$$\min_{u \in L^1(\Omega)} \{F(u) = \int_{\Omega} f(x, u(x), \nabla u(x))\} \text{ with } f(x, t, p^x) = g(x, t) + h(p^x) \quad (3.2)$$

Where  $\Omega$  is a 2-dimensional bounded open subset of  $\mathbb{R}^2$  representing the image, it is in our case a rectangular domain.

The Lagrangian  $f(x, t, p^x)$  is the core of the stereo-matching model,  $g$  is the data-term (it is linked to the pair of images),  $h$  is the regularization term, also called the prior, and the function  $u$ , is the disparity map, which belongs to a functional space, that can be more restricted than  $L^1(\Omega)$ .

Various methods of this family, are inspired by variational methods applied to optical flow, which is a similar problem, that in contrast with the problem of stereo matching, does not study the change of camera orientation, but of the scene itself. This is the case of the article [24]

The prior term generally chosen is the term of "total variation", for which h is equal to the  $L_1$  norm on the continuous model 3.2, ( this term promote solutions piecewise continuous, which corresponds to one of the common properties of a disparity map) , or other variants:

-it is the case of the last article, who propose a regularized version of the total variation: the norm  $L_1(\Omega)$  is replaced by the term

$$f \rightarrow \int_{\Omega} \sqrt{f(x)^2 + \epsilon} dx$$

-In another approach [21], authors proposed the Nagel-enkelmann operator 3.3. This term is a generalization of the total variation, that minimizes the disparity in an anisotropic way near the edges, emphasizing the minimization in the directions parallel to the gradients of (intensity of the) image.

$$h(p^x) = (p^x)^T D(\nabla I) p^x \quad \text{where } D \text{ is the gradient of the image, and } h \text{ refers to 3.2} \quad (3.3)$$

Though, in this case, the regularizing term, is closer to the  $L_2$  norm.

-Another interesting variant is the one used on the article [26], called ITGV:

The total variation of a function on the set  $\Omega$  is defined generally by the expression  $V(u) = \int_{\Omega} |\nabla u(x)| dx$ , it can also be defined as  $V(u) = \sup \{ \int_{\Omega} u(x) \operatorname{div} \Phi(x) : \Phi \in C_c^1(\Omega, \mathbb{R}), \|\Phi\|_{L^\infty(\Omega)} \leq 1 \}$ , which is called the dual definition of TV.

This model favors in fact solutions which are piecewise constant.

The mathematical generalization of the total variation, is the TGV regularizing of order k that can be defined as:

$$TGV_{\alpha}^k(u) = \sup \left\{ \int_{\Omega} u \operatorname{div} v^k \Phi ds \mid \Phi \in C_c^k(\Omega, \operatorname{Sym}^k(\mathbb{R}^2)) : \|\operatorname{div} v^l \Phi\|_{\infty} \forall l \in [0, k-1] \right\}$$

Where  $\operatorname{Sym}^k(\mathbb{R}^2)$  is the space of symmetric tensors of order k with arguments in  $\mathbb{R}^2$ , more precisely:

$$\operatorname{Sym}^k(\mathbb{R}^2) = \{ \xi : \underbrace{\mathbb{R}^2 \times \dots \times \mathbb{R}^2}_{k \text{ times}} \rightarrow \mathbb{R} : \xi \text{ symmetric and } k \text{ linear} \}$$

and  $\alpha \in \mathbb{R}^k$  is a constant,

where  $C_c^k(\Omega, \operatorname{Sym}^k(\mathbb{R}^2))$  denotes the space of functions with support of compact adherence, that are k continuously differentiable in a sense clarified in the article [19]. The generalization of divergence for this objects is also defined in this article, and would be heavy to present in this section.

Nevertheless, what is interesting about the generalization  $TGV_{\alpha}^k$ , is that it is equal to  $\alpha TV(u)$  for k equal 1, and in the other cases, it favors piecewise polynomial of order k, which permits more general solutions.

In the case of k=2 ,  $TGV_{\alpha}^k$  can be rewritten on this form:

$$TGV_{\alpha}^2 = \min_{\omega \in \mathbb{R}^2} \{ \alpha_1 \int_{\Omega} |\nabla u - \omega| dx + \alpha_0 \int_{\Omega} |\nabla w| dx \}$$

which constitute its primal form.

The ITGV regularizer, which refer to image driven regularizing, is a version of the prior  $TGV_\alpha^2$  dependent to the data, explicitly, it is equal:  $ITGV_\alpha^2 = \min_{\omega \in \mathbb{R}^2} \{\alpha_1 \int_{\Omega} |D^{\frac{1}{2}} \nabla u - \omega| dx + \alpha_0 \int_{\Omega} |\nabla w| dx\}$

The operator  $D^{\frac{1}{2}}$  whose expression can be found on section II, and subsection 2 of the article [26], is an anisotropic diffusion tensor, which works similarly to the "Nagel-Enkelmann operator", but one can observes that in this case, the regularizer is closed to the total variation, while on the case 3.3, it is closer to a Tikhonov regularizer (which equivalent to use an  $L_2$  norm for the function h )

What is interesting about this regularizer, is that it is self-contained. Minimizing it over the dummy variable  $\omega$  and the variable  $u$  (the disparity map), it gives a self-contained model for stereo-estimation.

Concerning the data term , as in the discrete case, the most classical one is the SSD (the sum squared difference), on other articles, we choose to minimize the difference of the gradients, or a weighted sum of the two terms. The solutions that minimizes the gradients, are favored, because they register better the edge and permits, a more robust approach, toward the variation of illumination.

A famous data-term is the difference of the census term of the image, which is non continuous, but very robust toward variation of illumination, and present the virtuous property of being non parametric. The census transform and the SDD are implemented on the next algorithm, the census transform would be detailed on the first implemented algorithm.

### **Optimization methods**

Generally the continuous model 3.2 is not convex, , the non-convexity comes mostly from the data-term, an image's intensity which on the most general cases is not convex makes it intractable, to construct a convex data-term.

Thence, in order to resolves the problem, there are two approaches:

The first one resides on the resolution of Euler-Lagrange equations. the article [21] , use a multi-scale approach, where the disparity is refined progressively (section 2.5 of the article), by resolving numerically the equations of Eule-Lagrange iteratively, and by considering the disparity computed at the previous step as the initial disparity,

The second approach consists in the linearization of the Lagrangian (i.e the term "f" on the model 3.2), using a Taylor approximation, this is the approach used in [26], using the same iterative procedure for the first approach, that we designate as a coarse to fine procedure.

This procedure helps to avoid the convergence toward a local minimum, needs an initial disparity map, but doesn't gives necessarily the exact minimum.

One of the most important articles, that treats the continuous approaches is the article [14] , it gives a generic algorithm, that resolves the exactly the problem 3.2, it is generic in the sense,that it resolves the problems as long as the data-term is continuous, and the regularization term is convex. It would be the first implemented algorithm, whose implementation would be described and

analyzed on the next section. This algorithm is however slow, but in the particular case of the total variation, it can be shown, that the resolved problem, can be resolved faster.

### 3.2. Global solutions of variational models of the stereo matching problem with convex regularization term

The article [14] shows that a solution of the problem 3.2 (an exact global minimizer) can be obtained as the a solution of a convex problem.

It suggest to resolves the problem on the space  $W_{1,1}(\Omega) = \{u \in L^1 : Du \text{ exists and } L^1(\Omega)\}$ , with  $Du$  the derivative in the distributional sense.

The importance of that space is that its functions, can have discontinuities, which is one of the characteristics of the disparity maps, it is also contained by definition, on the set of the model 3.2.

The article proposes a method to transform the problem:

$$F(u) = \inf_{u \in W_{1,1}(\Omega)} \int_{\Omega} f(x, u(x), Du(x)) dx \quad (3.4)$$

with the hypothesis that  $f$  is continuous with respect to the two first variables, and convex with respect to the last one.

#### 3.2.1. Description of the method

The proof is divided on three steps, the first one consists one the convex representation of the functional of the model 3.2, the obtained model is resolved over a non convex set, the second step, consists on its relaxation to a convex set, and the third set, changes the model to more tractable one, using a duality approach.

#### Convex representation

The minimization problem:

$$\inf_{u \in W_{1,1}(\Omega)} F(u)$$

can be rewritten as an interfacial energy,of the boundary  $\Gamma_u$  of the subgraph of

$$1_u = \begin{cases} 1 & \text{if } u(x) > t \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

An interfacial energy is of the form  $\int_{\Gamma_u} h(x, t, v_{\Gamma_u}(x)) d\mathcal{H}^2(x)$ , with  $\mathcal{H}^2$  the Hausdorff measure.

The definition of the 2-dimensional Hausdorff measure on the space  $\mathbb{R}^3$  is postponed to the appendix F.

More precisely, choosing the convex function:

$$h(x, t, p) = \begin{cases} |p^t|f(x, t, \frac{p^x}{|p^t|}) & \text{if } p^t < 0 \\ f^\infty(x, t, p^x) & \text{if } p^t = 0 \\ +\infty & \text{if } p^t > 0 \end{cases} \quad (3.6)$$

with  $f^\infty(x, t, p^x) = \lim_{\lambda \rightarrow +\infty} f(x, t, \lambda p^x) / \lambda$ , (which is well defined, since by convexity:  $\lambda \rightarrow (f(x, t, \lambda p^x) - f(x, t, 0)) / \lambda$  is an increasing function). We get that :

$$\int_{\Omega} f(x, u(x), Du(x)) d\lambda = \int_{\Omega} h(x, u(x), (Du(x), -1)) d\lambda$$

with D the derivative in the distributional sense, and  $v_{\Gamma_u}(x) = \frac{1}{\sqrt{1+|\nabla u(x)|^2}} \begin{pmatrix} Du(x) \\ -1 \end{pmatrix}$

Using the property 16 in appendix F, we get that:

$$\int_{\Gamma_u} h(x, t, v_{\Gamma_u}(x, t)) d\mathcal{H}^2(x, t) = \int_{\Omega} h(x, u(x), (Du(x), -1)) d\lambda$$

Then :

$$\int_{\Omega} f(x, u(x), Du(x)) d\lambda = \int_{\Gamma_u} h(x, t, v_{\Gamma_u}(x)) d\mathcal{H}^2(x, t)$$

Defining  $\mathcal{F}(v) = \int_{\Omega \times \mathbb{R}} h(x, t, Dv) d\mathcal{H}^2(x, t) \quad \forall v \in BV_{loc}$ ,

with :  $BV_{loc}(\Omega) = \{u \in L^1_{loc}(\Omega) : V(u, \Omega) < +\infty\}$ ,  
and  $V(u, \Omega) = \sup \{\int_{\Omega} u(x) \operatorname{div} \Phi(x) dx : \Phi \in C_c^1(\Omega, \mathbb{R}), \|\Phi\|_{L^\infty(\Omega)} \leq 1\}$ .

One gets that for  $v = 1_u$  (defined from 3.5) with  $u \in W_{1,1}(v)$ :

$$\mathcal{F}(1_u) = \int_{\Gamma_u} h(x, t, v_{\Gamma_u}(x)) d\mathcal{H}^2(x, t) = \int_{\Omega} f(x, u(x), Du(x)) d\lambda$$

Then we deduce the following property:

$$\inf_{u \in BV(\Omega)} \mathcal{F}(1_u) = \inf_{u \in W_{1,1}(v)} F(u)$$

This constitute the convex representation of the problem of minimization of the non convex functional F:

Indeed, the function h (3.6) is shown to be convex, lower-semi-continuous and one-homogeneous with respect to its last variable ([22]).

The one-homogeneity, and convexity comes directly by construction, of the function h. Concerning the lower-semi-continuity with respect to the last variable, the proof is redirected to the article ([22]).

## Convex relaxation

By considering the set  $C = \{v \in BV(\Omega \times \mathbb{R}, [0, 1]) : \lim_{t \rightarrow -\infty} v(x, t) = 1 \text{ and } \lim_{t \rightarrow +\infty} v(x, t) = 0\}$ . We see that  $\{1_u : u \in W_{1,1}(\Omega)\} \subset C$ , thence minimizing  $\mathcal{F}$  on this last space constitute a convex relaxation.

The property that is partially shown in the section 3.1 of the article [14] is that in order to resolve the binary problem :

$$\inf_{u \in W_{1,1}(\Omega)} \mathcal{F}(1_u) \quad (3.7)$$

it is sufficient to resolve the problem

$$\inf_{v \in C} \mathcal{F}(v) \quad (3.8)$$

More precisely, we get that for a solution  $v^*$  of the problem 3.8 the element  $1_{\{v^* > s\}}$  (with the usual definition of an indicator function for the set  $\{v^* > s\}$ ) is a global minimizer of the problem 3.7 and thence a global minimizer of 3.4, for every  $s \in [0, 1]$ .

This implies particularly that  $\exists u \in W_{1,1}(\Omega) \quad 1_{v^* > s} = 1_u$  with the definition 3.5 , and then that the solution is non-increasing with respect to the last variable.

The resolution of the problem 3.4 is thence equivalent to the resolution of the problem 3.8.

### Resolution by duality

In order to simplify the resolution of the problem 3.8, it is rewritten as a saddle point problem. It is partially shown in theorem 3.2 of the article that:

$$\forall u \in W^{1,1}(\Omega, \mathbb{R}) \quad \mathcal{F}(1_u) = \sup_{\Phi \in K} \int_{\Omega \times \mathbb{R}} \Phi.D1_u \quad (3.9)$$

Where K is a convex set defined as:

$K = \{\Phi = (\Phi^x, \Phi^t) \in C_0(\Omega \times \mathbb{R}, \mathbb{R}^3) \quad \Phi^t \geq f^*(x, t, \Phi^x(x, t))\}$ , where  $f^*$  is the conjugate with respect to the third variable .

Then it is shown the following property, which constitutes the generalized co-area formula.

**Property 1.** *For functionals of the this form  $\mathcal{H} : v -> \int_{\Omega \times \mathbb{R}} h(x, t, v_v(x, t))$ , with h 1-homogeneous with respect to the last variable, we get this formula:  $\mathcal{H}(v) = \int_{-\infty}^{+\infty} \mathcal{H}(1_{v \geq s}) \forall v \in BV_{loc}(\Omega)$  And in particular,  $\mathcal{H}(v) = \int_0^1 \mathcal{H}(1_{v \geq s}) \forall v \in C$*

The proof is on the theorem 3.1 of the article [14].

Then, using this last property, the formula 3.9 is generalized to

$$\forall v \in BV_{loc}(\Omega \times \mathbb{R}) \quad \mathcal{F}(v) = \sup_{\Phi \in K} \int_{\Omega \times \mathbb{R}} \Phi.Dv \quad (3.10)$$

Finally, we get the following conclusion from the article [14]

**Theorem 1.** *A solution of the problem 3.4, can be obtained from a solution  $v^*$  of the problem:*

$$\inf_{v \in C} \sup_{\Phi \in K} \int_{\Omega \times \mathbb{R}} \Phi.Dv \quad (3.11)$$

with

$$C = \{v \in BV(\Omega \times \mathbb{R}, [0, 1]) : \lim_{t \rightarrow -\infty} v(x, t) = 1 \quad \text{and} \quad \lim_{t \rightarrow +\infty} v(x, t) = 0\}$$

and

$$K = \{\Phi = (\Phi^x, \Phi^t) \in C_0(\Omega \times \mathbb{R}, \mathbb{R}^3) \quad \Phi^t \geq f^*(x, t, \Phi^x(x, t))\}$$

by considering the function  $u^* : \Omega \rightarrow \mathbb{R}$ , such that  $1_{u^*} = 1_{v^* > s}$ , with  $s \in [0, 1]$

### 3.2.2. Numerical Implementation

#### Discretization of the problem

Numerically, we should resolve the following problem:

Find  $\phi^h$  ans  $v^h$  solutions of  $\min_{v^h \in C^h} \max_{\phi^h \in K^h} \langle \nabla^h v^h, \phi^h \rangle$ ,

with  $C^h = \{v^h \in [0, 1]^{N_x N_y N_t} : (0, 0, 0) < (i, j, k) < (N_x, N_y, N_t)\}$

and  $K^h = \{\phi^h = (\phi_x^h, \phi_y^h, \phi_t^h) \in (\mathbb{R}^3)^{N_x, y, t} : (\phi_{i,j,k}^h)_{(i,j,k) \in N_{x,y,t}} \in K\}$  with  $K$  defined in (3.11)

and  $G^h = \{(i h_x, j h_y, k h_t + offset) : (0, 0, 0) < (i, j, k) < (N_x, N_y, N_t)\}$  the grid that would represent any function  $g$  in the discrete form with the following notation  $g_{i,j,k} = g(x_i, y_j, k h_t + offset)$ , where the index  $h$  stands for the step  $h = (h_x, h_y, h_t)$ ,

$N_{x,y,t} = [|1, N_y|] \times [|1, N_x|] \times [|1, N_t|]$ , (the  $y$  coordinate is associated with the row, and the  $x$  coordinate with the column)

$$\text{and } (\nabla^h v)_{i,j,k} = \begin{cases} (\nabla_y^h v)_{i,j,k} \\ (\nabla_x^h v)_{i,j,k} \\ (\nabla_t^h v)_{i,j,k} \end{cases}$$

$$\text{With } (\nabla_y^h v)_{i,j,k} = \begin{cases} \frac{(v_{i+1,j,k} - v_{i,j,k})}{h_y} & \text{if } i < N_y - 1 \\ 0 & \text{else} \end{cases}$$

$$\text{With } (\nabla_x^h v)_{i,j,k} = \begin{cases} \frac{(v_{i,j+1,k} - v_{i,j,k})}{h_x} & \text{if } i < N_x - 1 \\ 0 & \text{else} \end{cases}$$

$$\text{With } (\nabla_t^h v)_{i,j,k} = \begin{cases} \frac{(v_{i,j,k+1} - v_{i,j,k})}{h_x} & \text{if } i < N_t - 1 \\ 0 & \text{else} \end{cases}$$

#### Correspondence between the indexes of the grid and the data :

$N_x h_x$  and  $N_y h_y$  are the dimensions of the (two) images, one possible choice that would simplify the algorithm is :  $N_y, N_x$  equal respectively to the number of rows and columns of the image, and  $h_x = h_y = 1$ . In order to apply the algorithm for bigger values of  $h$ , it would be simpler and equivalent to perform an interpolation, on the image before its implementation.

The values of  $offset.h_t$  and  $(offset + N_t)h_t$  should correspond to the extremal values of the interval of disparity:

The set  $C^h$  is a discretization of the set  $C = \{v \in BV(\Omega \times \mathbb{R}; [0, 1]) : \lim_{t \rightarrow -\infty} v(x, t) = 1 \text{ and } \lim_{t \rightarrow +\infty} v(x, t) = 0\}$

$0\}$  with  $\Omega$  the region of the image, where we seek to compute the disparity map.

Since the function  $v(x, \cdot)$  is defined on  $\mathbb{R}$ , we should be restrained to a set of bounded functions in order to represent them numerically

From the sub-subsection "convex relaxation" we have that the solution  $v^*$  of the theorem 3.11 verify that

$$1_{v^* > s} = 1_{u^*} \quad (3.12)$$

with  $u^*$  the disparity map, in its continuous form .

Knowing that  $1_{v^* > s}$  has the expression:

$$\begin{cases} 1 & \text{if } v^*(x, t) > s \\ 0 & \text{else} \\ \forall (x, t) \in \Omega \times \mathbb{R} \end{cases}$$

And is equal to :

$$\begin{cases} 1 & \text{if } u^*(x) > t \\ 0 & \text{else} \\ \forall (x, t) \in \Omega \times \mathbb{R} \end{cases}$$

The equality gives the equivalence:

$$\forall t \in \mathbb{R} \quad \forall x \in \Omega \quad u^*(x) > t \Leftrightarrow v^*(x, t) > s$$

Thence

$$u^*(x) = \sup\{t \in \mathbb{R} : v^*(x, t) > s\} \quad (3.13)$$

and since, the obtention of the disparity map is our final result, is it sufficient to discretise  $v^*$ , on the maximal extension of disparity, it is thence one of the inputs of the algorithm.

We also have from the equation 3.12 that  $u^* = \int_{t \in \mathbb{R}} 1_{v^*(x, t) > s}$ , numerically, it is simplified to :  $u^*(x) = \sum_{k \in \text{disparity indexes}} 1_{v^*(x, k) > s} \forall x \text{pixel of the image}$

Remark: it is not specified which value of  $s$  to choose , a smaller value of  $s$  would give a field of values  $u$  punctually greater or equal to the previous one, because  $t \rightarrow v^*(x, t)$  is non-increasing, and from 3.13, we get the decrease of  $u^*$  with respect to  $s$ .

### Choice of the data-term and the regularization term

The data term should be continuous, in order to ensure that the lagragian is continuous , with respect to the last variable.

The two implemented data-terms are the following: (with respect to the model 3.2)

- The absolute difference between the intensity, of the images:  $g(x, t) = (I_r(x + t) - I_l(x))$  ( $I_r$  and  $I_l$  refers to the left and right image, this distinction is not very relevant in our case )
- The census transform ([25]), which is very robust with respect to the illumination's variations, but which is very irregular:  $g(x, t) = \Delta(C(I_L; x + u), C(I_R; x))$

with  $\Delta(p, q) = \sum_{p_i \neq q_i} 1$

and  $C(I; x) = \otimes_{y \in N(x), x \neq y} s(I; x, y)$ , where  $\otimes$  denotes the concatenation

$$s(I; p, q) = \begin{cases} 0 & \text{if } I(p) - I(q) < 0 \\ 1 & \text{if } I(q) - I(p) > 0 \end{cases}$$

One solution, to remove the regularity, would be to apply a convolution to the data-term  $g$ , which would ensure the continuity, here, we simply consider the continuous interpolate of the function.

Concerning the regularization term, the total variation was implemented, in this case the value of  $h$  is:  $h(p^x) = |p^x|$

### Calculus of the operator's projection

In the article, we needed to perform two projections:

-  $\text{proj}_{C^h}(v)$  may be computed by affecting  $v_{i,j,k}$  with the value 1, if it is greater than 1 , 0 if it is smaller than 0 , and the same element otherwise. This operation is called a clipping on the interval  $[0, 1]$

This constructed element verify that  $\forall y \in C_h, \langle y - \text{proj}_{C^h}(v) | v - \text{proj}_{C^h}(v) \rangle$ , (with  $C_h$  defined on sub-subsection 3.2.2) , thence this is the unique projection on  $C^h$

-  $\text{proj}_{K^H}(V)$ : For this term, we should go back to the initial constraint

$$K = \{\phi \in C_0(\Omega \times \mathbb{R}, \mathbb{R}^3) : \phi^t(x, t) \leq f^*(x, t, \phi^x(x, t)) \quad \forall x, t \in \Omega \times \mathbb{R}\}$$

which can be rexritten as

$$K = \{\phi \in C_0(\Omega \times \mathbb{R}, \mathbb{R}^3) : \phi^t(x, t) + g(x, t) \geq h^*(\Phi^x(x, t)) \quad \forall x, t \in \Omega \times \mathbb{R}\}$$

which can be red as a punctual constraint:

$$K(x, t) = \{\Phi(x, t) \in \mathbb{R}^3 : \Phi^t(x, t) + g(x, t) \geq h^*(\Phi^x(x, t))\}$$

with continuity hypothesis of  $\Phi$ .

In the case of the total variation, we find the following expression:

$$K(x, t) = \{\Phi(x, t) = (\Phi^x(x, t), \Phi^t(x, t)) \in \mathbb{R}^3 : \Phi^t(x, t) + g(x, t) \geq 0, |\Phi^x(x, t)| \leq 1\}$$

If we go back to the discrete setting, the projection can again be computed punctually:

$$\text{proj}_{K^H}(q)_{i,j,k} = (q_{i,j,k}^x / \max(1, q_{i,j,k}^x), \max(0, q_{i,j,k}^t) - g_{i,j,k}^h)$$

with  $q_{i,j,k} = (q_{i,j,k}^x, q_{i,j,k}^t) \in \mathbb{R}^2 \times \mathbb{R}$

### 3.3. A faster algorithm, for the particular case of the total variation

No result for the rate of convergence is mentioned about the last algorithm, and in our attempt to implement the algorithm for the data-set, and the purpose of this work, the algorithm seemed to

---

**Algorithm 1** algorithm of computation of the disparity  $u$  by the minimization of  $\min_{v^h \in C^h} \max_{\phi^h \in K^h} \langle \nabla^h v^h, \phi^h \rangle$

with  $C^h = \{v^h \in [0, 1]^{N_x N_y N_t} : (0, 0, 0) < (i, j, k) < (N_x, N_y, N_t)\}$

et  $K^h = \{\phi^h = (\phi_x^h, \phi_y^h, \phi_t^h) \in (\mathbb{R}^3)^{N_x N_y N_t} : (\phi^h)_{i,j,k} \in K\}$  with  $K$  the convex set determined by the data-term and the regularization term, and such that the indexes represents the points  $G^h = \{(ih_x, jh_y, kh_t + offset) : (0, 0, 0) < (i, j, k) < (N_x, N_y, N_t)\}$

---

**Require:** two rectified images covering the same scene, parameter  $s$ , the value  $N_t$  and the offset  $s$ .

**Ensure:** Disparity map  $u$

1: random initialization of the dual variables  $(\phi^h)^0 \in K^h$  and the primal variables  $(v^h)^0 \in C^h$  by the application of the Monte-Carlo method with rejection (though in the case of the total variation the set  $K^h$  has a simple form). Concerning the set  $C^h$  it is sufficient to choose an element between 0 and 1 for all the points.

2: initialization of the variable  $(\bar{v}^h)^0 = (v^h)^0$

**iterative instructions to resolve the numerical problem :** (as exposed in the article [14] , the proof is in the appendix of the article [28])

3:  $L = \sqrt{\left(\frac{4}{h_x^2} + \frac{4}{h_y^2} + \frac{4}{h_t^2}\right)}$  with  $h_x = h_y = 1$  as explained above,  $h_t$  fixed as 1 in the implemented version of the algorithm. choice of  $\tau$  and  $\sigma$  such that  $\tau\sigma L^2 < 1$

4: apply the instructions, until a fixed step is reached or the primal dual gap  $\sup_{\phi^h \in K^h} \langle \nabla^h v_c^h, \phi^h \rangle - \min_{v^h \in C^h} \langle \nabla^h v^h, \phi_c^h \rangle$  (is under a ratio compared to its initial value) ,with the index  $c$  standing for the current value

$$\begin{cases} (\phi^h)^{n+1} = \text{proj}_{K^h}((\phi^h)^n + \sigma(\Delta^h(\bar{v})^n)) \\ (v^h)^{n+1} = \text{proj}_{C^h}((\phi^h)^n - \tau(\text{div}^h(\phi^h)^{n+1})) \\ (\bar{v}^h)^{n+1} = 2(v^h)^{n+1} - (v^h)^n \end{cases}$$

{the numerical scheme concerns the finite difference , and the term  $\text{div}^h$  is the adjoint operator of the discretized gradient operator  $\nabla^h$  }

5: thresholding operation:

The value of the returned disparity map is equal to :

$$u^*(x) = \sum_{k \in \text{disparity indexes}} \mathbf{1}_{v^*(x, k) > s} \forall x \text{pixel of the image}$$


---

be not fast enough to the rectified images. In fact, one of the main drawbacks of the continuous method compared to the discrete methods is the computation time.

In the case of the total variation, the value of  $h$  (3.2) is  $h(x, y) = |(x, y)| = \sqrt{x^2 + y^2}$ , we obtain that  $h^* = \delta_{(x, y) \in \mathbb{R}^2, |(x, y)| \leq 1}$ , with [

$$\delta_F(x) = \begin{cases} 0 & \text{if } x \in F \\ +\infty & \text{else} \end{cases}$$

In this case the set  $K^h$  (3.2.2) becomes:

$$K^h = \{\Phi \in (\mathbb{R}^3)^{N_{x,y,t}} : \Phi_{i,j,k}^t + g_{i,j,k} \geq 0 \quad \text{and} \quad |(\Phi_{i,j,k}^x, \Phi_{i,j,k}^y)| \leq 1 \quad \forall (i, j, k) \in N_{x,y,t}\}$$

It can be rewritten in this form:

$$K^h = \{\Phi \in (\mathbb{R}^3)^{N_{x,y,t}} : \Phi_{i,j,k} \in K_{i,j,k} \quad \forall (i, j, k) \in N_{x,y,t}\} \text{ with}$$

$$K_{i,j,k} = \{\Phi = (\Phi_1, \Phi_2, \Phi_3) \in \mathbb{R}^3 : \Phi^3 + g_{i,j,k} \geq 0 \quad |(\Phi_1, \Phi_2)| \leq 1\}$$

and the notation  $g_{i,j,k} = g(x_i, y_j, t_k)$ . And in this particular case, the term  $\max_{\Phi \in K} \langle \nabla v, \Phi \rangle$ , can be easily evaluated, to obtain a more explicit convex problem to be resolved. In the next section, we will present this calculus. After that we obtain, a new convex problem, we will again transform this problem into a new problem which is strongly convex. A strongly convex problem can be resolved, with a rate of  $\frac{1}{k^2}$  using the famous algorithm FISTA, which present a serious advantage in our case. We will, then detail the algorithm used to resolve the problem.

### 3.3.1. Transformation of the Saddle Point Problem to an explicit minimization problem

As mentioned above, in order to perform the transformation, we need to compute

$$\max_{\Phi \in K} \langle \nabla v, \Phi \rangle \quad \forall v \in C$$

For  $v \in C$ , we have that:  $\langle \nabla v, \Phi \rangle = \sum_{i,j,k} [\langle \begin{pmatrix} (\nabla_x v)_{i,j,k} \\ (\nabla_y v)_{i,j,k} \end{pmatrix}, \begin{pmatrix} (\Phi^x)_{i,j,k} \\ (\Phi^y)_{i,j,k} \end{pmatrix} \rangle + (\nabla_t v)_{i,j,k} \cdot \Phi_{i,j,k}^t]$

Then from the independence of the constraints on the terms  $\Phi_{i,j,k}$ , we have that:

$$\max_{\Phi \in K} \langle \nabla v, \Phi \rangle = \sum_{i,j,k} \max_{\Phi_{i,j,k} \in K_{i,j,k}} [\langle \begin{pmatrix} (\nabla_x v)_{i,j,k} \\ (\nabla_y v)_{i,j,k} \end{pmatrix}, \begin{pmatrix} (\Phi^x)_{i,j,k} \\ (\Phi^y)_{i,j,k} \end{pmatrix} \rangle + (\nabla_t v)_{i,j,k} \cdot \Phi_{i,j,k}^t] \quad (3.14)$$

Again from the independence of the components  $\Phi_{i,j,k}^t$  and  $(\Phi_{i,j,k}^x, \Phi_{i,j,k}^y)$  in the set  $K_{i,j,k}$ , we have that

$$\max_{\Phi \in K} \langle \nabla v, \Phi \rangle = \sum_{i,j,k} \max_{|(\Phi_{i,j,k}^x, \Phi_{i,j,k}^y)| \leq 1} [\langle \begin{pmatrix} (\nabla_x v)_{i,j,k} \\ (\nabla_y v)_{i,j,k} \end{pmatrix}, \begin{pmatrix} (\Phi^x)_{i,j,k} \\ (\Phi^y)_{i,j,k} \end{pmatrix} \rangle] + \max_{\Phi_{i,j,k}^t \geq -g_{i,j,k}} [(\nabla_t v)_{i,j,k} \cdot \Phi_{i,j,k}^t]$$

From 3.2.2, we can deduce the following property on the solution, it is non-decreasing, on its third component. So in the vicinity of the solution, we have that  $(\nabla_t v)_{i,j,k} \leq 0$ , we deduce then that

$$\max_{\Phi_{i,j,k}^t \geq -g_{i,j,k}} [(\nabla_t v)_{i,j,k} \cdot \Phi_{i,j,k}^t] = |(\nabla_t v)_{i,j,k}| g_{i,j,k}.$$

We also have (from Cauchy-Schwartz inequality )that :

$$\max_{|(\Phi_{i,j,k}^x, \Phi_{i,j,k}^y)| \leq 1} [\langle \frac{(\nabla_x v)_{i,j,k}}{(\nabla_y v)_{i,j,k}}, \frac{(\Phi^x)_{i,j,k}}{(\Phi^y)_{i,j,k}} \rangle] = |(\nabla_x v)_{i,j,k}, (\nabla_x v)_{i,j,k}|$$

We deduce then that:

$$\max_{\Phi_{i,j,k} \in K_{i,j,k}} \langle \nabla v, \Phi_{i,j,k} \rangle = |(\nabla_x v)_{i,j,k}, (\nabla_x v)_{i,j,k}| + |(\nabla_t v)_{i,j,k}| g_{i,j,k} \quad \forall (i, j, k) \in N_{x,y,t}$$

We deduce then from 3.14:

$$\min_{v \in C} \max_{\Phi \in K} \langle \nabla v, \Phi \rangle = \min_{v \in C} \sum_{i,j,k} |(\nabla_x v)_{i,j,k}, (\nabla_x v)_{i,j,k}| + |(\nabla_t v)_{i,j,k}| g_{i,j,k}$$

Then the new explicit problem in this case is the following:

$$\min_{v \in C} I(v) = \min_{v \in C} TV_{2Dij}(v) + TV_{1DK,g}(v) \quad (3.15)$$

with,  $TV_{2Dij}(v) = \sum_{i,j,k} |(\nabla_x v)_{i,j,k}, (\nabla_y v)_{i,j,k}|$  (with the same notation above,  $(\nabla_x v)_{i,j,k}$  is null if the component if the index is out of the grid)

and  $TV_{1DK,g}(v) = \sum_{i,j,k} |(\nabla_t v)_{i,j,k}| g_{i,j,k}$ .

### 3.3.2. Analysis of the explicit minimization problem

One remark, about the problem

$$\min_{v \in C} I(v) = \min_{v \in C} TV_{2Dij}(v) + TV_{1DK,g}(v)$$

is that the cost of the function is not strictly convex, which makes it hard to resolves it, rapidly .

In the article [6], an interesting procedure is mentioned, in order to transform the convex problem

$$\min_{v \in C} I_1(v) \quad (3.16)$$

to a strongly convex problem:

$$\min_{v \in C} I_1(v) + \frac{\lambda}{2} \|v - f\|^2 \quad (3.17)$$

with  $I_1(v) = TV_{1Di}(v) + TV_{1Dj}(v) + TV_{1DK,g}(v)$  and

$TV_{1Di}(v) = \sum_{i,j,k} |(\nabla_y v)_{i,j,k}|$ ,

$TV_{1Dj}(v) = \sum_{i,j,k} |(\nabla_x v)_{i,j,k}|$ , and

$TV_{1DK,g}(v) = \sum_{i,j,k} |(\nabla_t v)_{i,j,k}| g_{i,j,k}$  and with f defined:  $f_{i,j,k} = \begin{cases} \gamma & \text{if } k = 0 \\ -\gamma & \text{if } k = N_t \\ 0 & \text{else} \end{cases}$

By transforming a minimization problem, we mean that, we can obtain a solution of the first problem, by applying a transformation to the (unique) solution of the second problem.

More precisely, we show two properties: If  $\gamma$  is big enough, the solution  $v_1$  of 3.17 verify that  $(v_1)_{i,j,k} > 0$  and  $(v_1)_{i,j,N_t}$  for all the possible indexes  $i, j$ . In this case, the second property state that

, a solution  $u$  of the problem 3.16 can be obtained by the operation  $u_{i,j,k} = \begin{cases} 1 & \text{if } (v_1)_{i,j,k} \geq 0 \\ 0 & \text{if } (v_1)_{i,j,k} < 0 \end{cases}$

The first property can be shown for  $I$  proper, convex and lower-semi-continuous, and continuous on 0. ( all the definitions and properties necessary to follow the proofs are on the appendix)

**Property 2.** *If  $I$  is proper, convex and continuous , then for  $\gamma$  sufficiently big ( $\gamma > 2 * ||\partial I(B(0, \rho))||_\infty$ ), we have that  $(v_\gamma^*)_{i,j,0} > 0$  and  $(v_\gamma^*)_{i,j,N_t} < 0$*

*Proof.* Let's note  $L(v) = I(v) + \frac{\lambda}{2}||v - f||^2$ . Let  $v_\gamma^*$  be the minimum of  $L$ . We have from the Law of Fermat (7) that:  $0 \in \partial L(v_\gamma^*)$ , we also have from property (5)  $\partial L(v_\gamma^*) = \partial I(v_\gamma^*) + \partial(\frac{\lambda}{2}||v - f||^2)(v_\gamma^*)$ , then  $0 \in \partial I(v_\gamma^*) + \{v_\gamma^* - f\}$ , then  $\exists p_{v_\gamma^*} \in \partial I(v_\gamma^*)$ , such that  $v_\gamma^* = f - p_{v_\gamma^*}$ . In particular, we have that  $(v_\gamma^*)_{i,j,0} = \gamma - (p_{v_\gamma^*})_{i,j,0}$ , and  $(v_\gamma^*)_{i,j,N_t} = -\gamma - (p_{v_\gamma^*})_{i,j,N_t}$ , then choosing  $\gamma > 2 * ||\partial I(B(0, \rho))||_\infty$ , imposes that  $(v_\gamma^*)_{i,j,0} > 0$  and  $(v_\gamma^*)_{i,j,N_t} < 0$   $\square$

The second property is proven in [5], and is assumed for the rest on this section.

The problem 3.16 is similar to the problem 3.15, and presents a structure that makes it solvable rapidly, with the algorithm FISTA presented in appendix B,(Corollary 3) , we will then choose to solve this problem rather than (3.15), assuming that the solution would be similar. The article [6] presents an algorithm to resolve this problem, the rest of this section will be dedicated to the analysis of this algorithm.

### 3.3.3. Resolution of the strongly Convex Problem

The problem 3.17 is in the following form:

$$\min_{v \in X} (f_1(v) + f_2(v) + f_3(v) + \frac{\lambda}{2} \|v - f\|^2), \text{ with } X = \mathbb{R}^{N_{x,y,t}}$$

with  $f_1(v) = TV_{1Dk,g}(v) = \sum_{i,j,k} |(\nabla_t v)_{i,j,k}| g_{i,j,k} \forall v \in X$ ,  $f_2(v) = TV_{1Di}(v) = \sum_{i,j,k} |(\nabla_y v)_{i,j,k}|$  and  $f_3(v) = TV_{1Dj}(v) = \sum_{i,j,k} |(\nabla_x v)_{i,j,k}|$ .

This problem is  $\lambda$  strongly convex, and can thence be resolved by the FISTA Algorithm (B), but it involves that  $\text{prox}_{f_1+f_2}$  can be computed efficiently. Another efficient algorithm using the FISTA trick is presented in [6],and would be used here, but in order to use it, we will show its equivalence with the dual problem.

This problem can be rewritten:  $\min_{v, v_1, v_2, v_3 \in X} \max_{x_1, x_2, x_3 \in X} (f_1(v_1) + \langle x_1, v - v_1 \rangle + f_2(v_2) + \langle x_2, v - v_2 \rangle + f_3(v_3) + \langle x_3, v - v_3 \rangle + \frac{\lambda}{2} \|v - f\|^2) = \min_{v, v_1, v_2 \in X} \max_{x_1, x_2 \in X} L(v, v_1, v_2, v_3, x_1, x_2, x_3)$

With :  $L(v, v_1, v_2, v_3, x_1, x_2, x_3) = f_1(v_1) + \langle x_1, v - v_1 \rangle + f_2(v_2) + \langle x_2, v - v_2 \rangle + f_3(v_3) + \langle x_3, v - v_3 \rangle + \frac{\lambda}{2} \|v - f\|^2$ .

$L$  is in  $\Gamma_0$  and strongly convex (appendix A) with respect to the variable  $u$  ,  $u_1$   $u_2$  and  $u_3$ , then from property 2 ,  $L$  admit a unique minimizer for  $x_1$ ,  $x_2$  and  $x_3$  fixed, we have then that:

$\exists v_x, v_{x,1}, v_{x,2}$  and  $v_{x,3}$  such that

$$\begin{aligned} \max_{v, v_1, v_2, v_3 \in X} \min_{x_1, x_2, x_3 \in X} L(v, v_1, v_2, v_3, x_1, x_2, x_3) &= \max_{v, v_1, v_2, v_3 \in X} L(v_x, v_{x,1}, v_{x,2}, v_{x,3}, x_1, x_2, x_3) \geq \\ &\min_{x_1, x_2, x_3 \in X} \max_{v, v_1, v_2, v_3 \in X} L(v, v_1, v_2, v_3, x_1, x_2, x_3) (*). \end{aligned} \quad (3.18)$$

We also have that:

$$\forall v, v_1, v_2, v_3 \in X \quad \min_{x_1, x_2, x_3 \in X} \max_{v, v_1, v_2, v_3 \in X^3} L(v, v_1, v_2, v_3, x_1, x_2, x_3) \geq \min_{x_1, x_2 \in X} L(v, v_1, v_2, v_3, x_1, x_2, x_3)$$

Then we deduce:

$$\min_{x_1, x_2, x_3 \in X} \max_{v, v_1, v_2, v_3 \in X} L(v, v_1, v_2, v_3, x_1, x_2, x_3) \geq \max_{v, v_1, v_2, v_3 \in X} \min_{x_1, x_2, x_3 \in X} L(v, v_1, v_2, v_3, x_1, x_2, x_3) (**)$$

Remark: This last property is more general and doesn't depends on the form of  $L$ , (the primal of a problem is always greater than its dual).

From (\*) and (\*\*), we obtain that

$$\min_{x_1, x_2, x_3 \in X} \max_{v, v_1, v_2, v_3 \in X} L(v, v_1, v_2, v_3, x_1, x_2, x_3) = \max_{v, v_1, v_2, v_3 \in X} \min_{x_1, x_2, x_3 \in X} L(v, v_1, v_2, v_3, x_1, x_2, x_3)$$

By the independence of the constraints, we obtain then, that:

$$\begin{aligned} & \max_{v, v_1, v_2, v_3 \in X} \min_{x_1, x_2, x_3 \in X} f_1(v_1) + \langle x_1, v - v_1 \rangle + f_2(v) + \langle x_2, v - v_2 \rangle + f_3(v_3) + \langle x_3, v - v_3 \rangle + \frac{\lambda}{2} \|v - f\|^2 \\ &= \max_{v, v_1, v_2, v_3 \in X} - \max_{v_1 \in X} (\langle x_1, v_1 \rangle - f_1(v_1)) - \max_{v_2 \in X} (\langle x_2, v_2 \rangle - f_2(v_2)) - \max_{v_3 \in X} (\langle x_3, v_3 \rangle - f_3(v_3)) + \\ & \quad \min_{v \in X} (\langle x_1 + x_2 + x_3, v \rangle + \frac{\lambda}{2} \|v - f\|^2) \end{aligned}$$

Then, we have that:

$$\begin{aligned} & \max_{v, v_1, v_2, v_3 \in X} \min_{x_1, x_2, x_3 \in X} f_1(v_1) + \langle x_1, v - v_1 \rangle + f_2(v) + \langle x_2, v - v_2 \rangle + f_3(v_3) + \langle x_3, v - v_3 \rangle + \frac{\lambda}{2} \|v - f\|^2 \\ &= \max_{x_1, x_2, x_3 \in X} - f_1^*(x_1) - f_2^*(x_2) - f_3^*(x_3) + \min_{v \in X} (\langle x_1 + x_2 + x_3, v \rangle + \frac{\lambda}{2} \|v - f\|^2) \end{aligned}$$

The minimum of the expression  $\min_{v \in X} (\langle x_1 + x_2 + x_3, v \rangle + \frac{\lambda}{2} \|v - f\|^2)$  is reached for the variable  $v$  at the point:  $v = f - \frac{(x_1 + x_2 + x_3)}{\lambda}$ .

We obtain then that:

$$\begin{aligned} & \max_{v, v_1, v_2, v_3 \in X} \min_{x_1, x_2, x_3 \in X} f_1(v_1) + \langle x_1, v - v_1 \rangle + f_2(v) + \langle x_2, v - v_2 \rangle + f_3(v_3) + \langle x_3, v - v_3 \rangle + \frac{\lambda}{2} \|v - f\|^2 \\ &= \max_{x_1, x_2 \in X} - f_1^*(x_1) - f_2^*(x_2) - f_3^*(x_3) - \frac{1}{2\lambda} \|x_1 + x_2 + x_3\|^2 + \langle x_1 + x_2 + x_3, f \rangle \end{aligned}$$

The dual problem is then in the following form :

$$(D_1): \min_{x_1, x_2, x_3 \in X} [G(x_1, x_2, x_3) = g_1(x_1) + g_2(x_2) + g_3(x_3) + \frac{1}{2\lambda} \|x_1 + x_2 + x_3\|^2] \quad (3.19)$$

with  $g_1(x_1) = f_1^*(x_1) - \langle x_1, f \rangle$ ,  $g_2(x_2) = f_2^*(x_2) - \langle x_2, f \rangle$  and  $g_3(x_3) = f_3^*(x_3) - \langle x_3, f \rangle$ .  
This problem can be rewritten in the form:

$$(D_1): \min_{x_1, x_2, x_3 \in X} [G(x_1, x_2, x_3) = g_1(x_1) + g_{2,3}(x_{2,3}) + \frac{1}{2} \|A_1 x_1 + A_{2,3} x_{2,3}\|^2] \quad (3.20)$$

with  $A_1$ , an operator defined on the space  $X$  with values on  $X$ , with  $A_1(x_1) = \frac{1}{\lambda} x_1$ , and  $A_{2,3}$  an operator defined in the space  $X \times X$  with values on the space  $X$ , with  $A_{2,3} x_{2,3} = \frac{1}{\lambda} (x_2 + x_3)$ , with the notation  $x_{2,3} = (x_2, x_3)$ , and with  $g_{2,3}(x_{2,3}) = g_2(x_2) + g_3(x_3)$

This last problem is very similar to the primal (3.17), at the difference that the number of variables has tripled. The FISTA Algorithm applied to the problem (3.20)

---

**Algorithm 2** FISTA Algorithm with two variables alternating descent

---

**Require:**  $f_1$  and  $f_2$  convex functions

**Ensure:** : The minimizer of  $\min_{x_1, x_2 \in X} g_1(x_1) + g_2(x_2) + \frac{1}{2\lambda} \|x_1 + x_2\|^2$  with  $g_1 = f_1^* - \langle \cdot, g \rangle$  and  $g_2 = f_2^* - \langle \cdot, g \rangle$

1: choose  $x^0 \in X$  and  $x^{-1} = x^0$  and  $t_0 = 0$

2:  $\forall k \geq 0$  {for a fixed number of iteration, or until the gap between the dual and the primal (property 4) is below a certain threshold do }

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$$

$$\bar{x}_2^k = x_2^k + \frac{t_k - 1}{t_{k+1}} (x_2^k - x_2^{k-1}) \quad \{ \text{there is no need to compute } \bar{x}_1^k \}$$

$$x_1^{k+1} = \underset{x_1}{\operatorname{prox}_{\lambda g_1}}(-\bar{x}_2^k) = \underset{x_1}{\operatorname{argmin}} g_1(x_1) + \frac{1}{2\lambda} \|x_1 + \bar{x}_2^k\|^2$$

$$\text{and } x_2^{k+1} = \underset{x_2}{\operatorname{prox}_{\lambda g_2}}(-x_1^{k+1}) = \underset{x_2}{\operatorname{argmin}} g_2(x_2) + \frac{1}{2\lambda} \|x_2^{k+1} + x_1^{k+1}\|^2$$


---

### 3.3.4. Algorithms for the resolution of the problem

The FISTA algorithm, with two variables descent, whose proof is presented in property (12) appendix C, is presented in algorithm 2.

This last algorithm is qualified abstract, because we should provide, algorithms to compute the recurrent instructions of step 2.

The computation of the two recurrent steps, is not always easy to solve. Solving the two instructions as continuous optimization problems may be inefficient, even if, they can be resolved, on a  $\frac{1}{k^2}$  rate, because the upper limit depends of the initial value of the argument, which could arbitrary be far from the optimum. Making the computation time of every iteration of the main algorithm difficult to estimate.

Furthermore, this kind of resolution, wouldn't give the exact solution of the problem which would impose an additional error term, on the iterative instructions.

It would, be thence more interesting, to emphasize on algorithms that gives the exact solution.

In the case, where these problems aren't possible to compute, the article [6], propose an algorithmic framework to solve them.

It resolves the problems of the form:

$$\min_{x=(x_i)_{i=1:N}} (\xi(x))$$

with

$$\xi(x) = \left( \sum_{i=1:N} f_i(x_i) \right) + \frac{1}{2} \left\| \sum_{i=1:N} A_i x_i \right\|^2.$$

In our case 3.20, we want to resolve a problem with two variables, the algorithm, takes, the form (algorithm 7), and we get similarly to the previous calculus, a generalized property:

**Property 3.** *The element  $x_k$  obtained in the iteration  $k$  from the algorithm 7 verify the following inequality:*

$$\xi(x_k) - \xi(x^*) \leq \frac{2}{(k+1)^2} \left( \|x_1^0 - x_1^*\|_{B_1}^2 + \frac{\|x_2^0 - x_2^*\|}{\tau_2} \right)$$

with  $x^*$ , the minimizer of  $\xi$ ,  $B_i = \frac{I}{\tau_i} I - A_i^* A_i$  that should be positive, the notation  $x = (x_1, x_2)$  and  $|x|_A = \langle Ax, x \rangle$ .

*Proof.* The proof is similar to the previous ones, it is in the section 3 of the article [6]. This article describe also more general cases that wouldn't be used here.  $\square$

---

### Algorithm 3 proximal alternating descent With two variables

---

**Require:**  $f_1$  and  $f_{2,3}$  convex functions defined on Hilbert space  $X_1$  and  $X_{2,3}$

**Ensure:** : The minimizer of  $\min_{x_1, x_2 \in X_1 \times X_{2,3}} g_1(x_1) + g_{2,3}(x_{2,3}) + \frac{1}{2} \|A_1 x_1 + A_{2,3} x_{2,3}\|^2$  with  $g_1 = f_1^* - \langle \cdot, g \rangle$

and  $g_{2,3} = f_{2,3}^* - \langle \cdot, g \rangle$

1: choose  $x^0 \in X$  and  $x^{-1} = x^0$  and  $t_0 = 0$

2: **for all**  $k$  until fixed number of iteration, or until the gap between the dual and the primal (property 4) is below a certain threshold do **do**

3:    $t_k = \frac{k+1}{2}$  {which is a possible choice that allows :  $t_k^2 \geq t_{k+1}(t_{k+1} - 1)$  and  $t_k > 1 \quad \forall k \geq 1$ }

4:    $x_{2,3}^k = x_{2,3}^k + \frac{t_k - 1}{t_{k+1}} (x_{2,3}^k - x_{2,3}^{k-1})$

5:    $\bar{x}_1^k = x_1^k + \frac{t_k - 1}{t_{k+1}} (x_1^k - x_1^{k-1})$

6:    $x_1^{k+1} = \underset{x_1 \in X}{\operatorname{argmin}} (g_1(x_1) + \langle x_1, A_1^*(A_1 \bar{x}_1 + A_{2,3} \bar{x}_{2,3}) \rangle + \frac{\|x_1 - \bar{x}_1\|^2}{2\tau_1})$

7:    $x_{2,3}^{k+1} = \underset{x_{2,3} \in X^2}{\operatorname{argmin}} (g_{2,3}(x_{2,3}) + \langle x_{2,3}, A_{2,3}^*(A_1 \bar{x}_1 + A_{2,3} \bar{x}_{2,3}) \rangle + \frac{\|x_{2,3} - \bar{x}_{2,3}\|^2}{2\tau_{2,3}})$

8: **end for**

---

The algorithm 3 is the algorithm that would be implemented, for which a more explicit version is on 7, and is explained, in the following sections.

#### 3.3.5. Description and analysis of the algorithm

This last algorithm is a twofold generalization of the algorithm 2, the first generalization concerns the problems to minimize. In the first case, the choice of  $A_i$  is the identity's operator. The second generalization concerns, the computation of the proximal operator for which, we choose a value  $\tau_i$ , for which  $\frac{I}{\tau_i} \geq A_i^* A_i$ , and  $g_i$  is "simple" ( $I$  is the identity of the operator of the current space). "Simple" means here a function for which the minimization can be computed easily. In the particular case  $\frac{I}{\tau_i} = A_i^* A_i$ , we get back the value of the proximal operator:  
(ie: For example for the first step:

$$\begin{aligned} x_1^{k+1} &= \operatorname{argmin} (g_1(x_1) + \langle x_1, A_1^*(A_1 \bar{x}_1 + A_{2,3} \bar{x}_{2,3}) \rangle + \frac{\|x_1 - \bar{x}_1\|^2}{2\tau_1}) \\ &= \operatorname{argmin} (g_1(x_1) + \langle x_1, A_1^* A_{2,3} \bar{x}_{2,3} \rangle + \frac{\|x_1\|^2}{2\tau_1}) \\ &= \operatorname{argmin} (g_1(x_1) + \frac{1}{2\tau_1} (\|x_1\|^2 - 2\langle x_1, \tau_1 A_1^* A_{2,3} \bar{x}_{2,3} \rangle)) \\ &= \operatorname{argmin} (g_1(x_1) + \frac{1}{2\tau_1} (\|x_1 - \tau_1 A_1^* A_{2,3} \bar{x}_{2,3}\|^2)) = \operatorname{prox}_{\tau_1 g_1}(\tau_1 A_1^* A_{2,3} \bar{x}_{2,3}) \end{aligned}$$

)

(The article [6] treat a more general case where the term  $\frac{\|x_i - \bar{x}_i\|^2}{2\tau_i}$  is replaced by  $\frac{M_i(\langle x_i - \bar{x}_i \rangle, \langle x_i - \bar{x}_i \rangle)^2}{2\tau_i}$ , and the condition  $\frac{I}{\tau_i} \geq A_i^* A_i$  is replaced by  $\frac{M_i}{\tau_i} \geq A_i^* A_i$ , but that won't be used here)

From the property 3, it is clear that the computation of the proximal operator gives a sharper inequality, it is thence better to compute the proximal operator when possible.

### Computation of the iterative instructions:

The operator  $prox_{\lambda g_i}(x_0)$ , can be transformed:

We have that:

$$\begin{aligned} prox_{\lambda g_i}(x_0) &= \underset{x_1}{\operatorname{argmin}}(g_1(x_1) + \frac{1}{2\lambda}\|x_0 - x_1\|^2) = \underset{x_1}{\operatorname{argmin}}(f_1^*(x_1) - \langle x_1, g \rangle + \frac{1}{2\lambda}\|x_0 - x_1\|^2) \\ &= \underset{x_1}{\operatorname{argmin}}(f_1^*(x_1) - \langle x_1 - x_0, g \rangle + \frac{1}{2\lambda}\|x_0 - x_1\|^2) = \underset{x_1}{\operatorname{argmin}}(f_1^*(x_1) + \frac{1}{2\lambda}\|x_1 - x_0 - g\|^2) \end{aligned}$$

Then  $prox_{\lambda g_i}(x_0) = prox_{\lambda f_1^*}(x_0 + g)$ .

Thence, it is sufficient to be able to compute  $prox_{\lambda f_1^*}$  and  $prox_{\lambda f_2^*}$

**Computation of the step**  $x_1^{k+1} = \underset{x_1}{\operatorname{argmin}}(g_1(x_1) + \langle x_1, A_1^*(A_1 \bar{x}_1 + A_{2,3} \bar{x}_{2,3}) \rangle + \frac{\|x_1 - \bar{x}_1\|^2}{2\tau_1})$  **of algorithm 7:**

The explicit expression of  $f_1$  is  $f_1(v) = TV_{1Dk,g}(v) = \sum_{(i,j,k) \in N_{x,y,t}} |(\nabla_t v)_{i,j,k}| g_{i,j,k} \quad \forall v \in X$

Its conjuguate is:

$$\begin{aligned} f_1^*(x) &= \sup_{v \in X} (\langle x, v \rangle - f_1(v)) \\ &= \sup_{v \in X} (\sum_{(i,j,k) \in N_{x,y,t}} x_{i,j,k} v_{i,j,k} - \sum_{(i,j,k) \in [|1, N_y|] \times [|1, N_x|] \times [|1, N_t - 1|]} |(\nabla_t v)_{i,j,k}| g_{i,j,k}) \\ &= \sup_{v \in X} (\sum_{(i,j) \in [|1, N_y|] \times [|1, N_x|]} (\sum_{k \in [|1, N_t|]} (x_{i,j,k} v_{i,j,k} - |(\nabla_t v)_{i,j,k}| g_{i,j,k}))) \\ &= \sum_{(i,j) \in [|1, N_y|] \times [|1, N_x|]} (\sup_{v \in \mathbb{R}^{[1, N_t]}} (\sum_{k \in [|1, N_t|]} (x_{i,j,k} v_k - |(\nabla v)_k| g_{i,j,k}))) \end{aligned}$$

With  $(\nabla v)_j = (v_{j+1} - v_j)$  with the convention that this term is null every time  $v_j$  or  $v_{j+1}$  is out of the grid.

Then

$$f_1^*(x) = \sum_{(i,j) \in [|1, N_y|] \times [|1, N_x|]} TV_{g_{i,j,.}}^*(x_{i,j,.})$$

With  $TV_c(v) = \sum_{k \in [|1, N_t - 1|]} |(\nabla v)_k| c_k$ , and the notation, that will be used in all this article, for  $y \in N_{x,y,t}$ ,

$y_{i,j,.} = (y_{i,j,k})_{k \in [|1, N_t|]}$ ,  $y_{.,j,k} = (y_{i,j,k})_{i \in [|1, N_y|]}$  and  $y_{.,j} = (y_{i,j,k})_{i \in [|1, N_y|]}$ .

The computation of  $TV_c^*$  is in appendix D:

$$TV_c^*(x) = \delta_{\{y \in \mathbb{R}^N : \sum_{j_1=1}^{j_1=j-1} y_{j_1} \leq c_{j-1} \forall j \in [|2, N|] \text{ and } \sum_{j_1=1}^{j_1=N} y_{j_1} = 0\}}(x)$$

Thence

$$f_1^*(x) = \delta_{\{y \in \mathbb{R}^{N_{x,y,t}} : \sum_{k_1=1}^{k_1=k} y_{i,j,k_1} \leq c_k \forall k \in [|1, N_t - 1|] \text{ and } \sum_{k_1=1}^{k_1=N_t-1} y_{i,j,k_1} = 0\}}(x) \quad \forall (i,j) \in [|1, N_y|] \times [|1, N_x|]$$

The set  $C = \{y \in \mathbb{R}^{N_{x,y,t}} : [ \sum_{k_1=1}^{k_1=k-1} y_{i,j,k_1} | \leq c_{k-1} \forall k \in [|2, N_t|] \text{ and } \sum_{k_1=1}^{k_1=N} y_{i,j,k_1} = 0 ] \quad \forall (i, j) \in [|1, N_y|] \times [|1, N_x|] \}$ , is a closed convex non-empty set and the proximal operator of  $\delta_K$ , has a known form (appendix E).

$f_1^*(x) = \text{prox}_{f_1}(x)$ . The convex set if of the form  $K = \{x \in X : Ax \leq b\}$ , it can be also written in the form of  $K = \bigcap_{i \in N_{x,y,t}} K_i$ , with  $K_i = \{x \in X : a_i x \leq b_i\}$ . The projection on the sets  $K_i$ , can be evaluated analytically. Thence one way to compute the projection, it to apply the algorithm "POCS" [16].

But the POCS algorithm doesn't give generally the exact solution of the projection.

Another way to compute the proximal operator is by using the Moreau Identity (property 15) in appendix E.

It is thence sufficient to be able to compute the term  $\text{prox}_{f_1}$ .

We denote  $N_{x,y} = [|1, N_y|] \times [|1, N_x|]$  for the rest of this report.

### Computation of $\text{prox}_{f_1}$

By definition:

$$\text{prox}_{f_1}(u) = \underset{v \in X}{\operatorname{argmin}} \sum_{(i,j,k) \in N_{x,y,t}} |(\nabla_t v)_{i,j,k}| g_{i,j,k} + \frac{1}{2} \|u - v\|^2$$

More explicitly:

$$\text{prox}_{f_1}(u) = \underset{v \in X}{\operatorname{argmin}} \sum_{(i,j,k) \in N_{x,y,t}} |(\nabla_t v)_{i,j,k}| g_{i,j,k} + \frac{1}{2} \sum_{(i,j,k) \in N_{x,y,t}} |u_{i,j,k} - v_{i,j,k}|^2$$

$$\text{prox}_{f_1}(u) = \underset{v \in X}{\operatorname{argmin}} \sum_{(i,j) \in N_{x,y}} \left[ \sum_{k \in [|1, N_t - 1|]} |v_{i,j,k} - v_{i,j,k+1}| g_{i,j,k} + \frac{1}{2} \sum_{k \in [|1, N_t|]} |u_{i,j,k} - v_{i,j,k}|^2 \right]$$

Thence:  $\forall (i, j) \in N_{x,y}$

$$\text{prox}_{f_1}(u)_{i,j,.} = \underset{v_{i,j,.} \in \mathbb{R}^{N_t}}{\operatorname{argmin}} \left( \sum_{k \in [|1, N_t - 1|]} |v_{i,j,k} - v_{i,j,k+1}| g_{i,j,k} + \frac{1}{2} \sum_{k \in [|1, N_t|]} |u_{i,j,k} - v_{i,j,k}|^2 \right) \quad (3.21)$$

The problem is equivalent, to the problem of solving:

$$ROF(x) = \underset{v \in \mathbb{R}^N}{\operatorname{argmin}} \left( \sum_{k \in [|1, N-1|]} |v_k - v_{k+1}| c_k + \frac{1}{2} \sum_{k \in [|1, N|]} |u_k - v_k|^2 \right)$$

This problem can be resolved as a dynamic programming algorithm, which in terms of messages has the following form:

Nevertheless, the case that interests u is a particular case, for which the functions  $f_i$  and  $f_{i,i+1}$  are convex. Furthermore,  $f_{i,i+1}$  is of the form

$$f_{i,i+1} = \begin{cases} w_{i,i+1}^- x & \text{if } x < 0 \\ w_{i,i+1}^+ x & \text{if } x \geq 0 \end{cases}$$

with  $w_{i,i+1}^- \leq w_{i,i+1}^+$

The article [29] presents an efficient algorithm for its resolution (the algorithm 5 below). The principal idea is that in the convex case, the message  $M_i$  are convex, so we use the sub-gradients for the resolution of the minimization , (appendixA), which simplifies it:

---

**Algorithm 4 Dynamic programming algorithm**


---

**Require:** Function  $f_i$  and functions  $f_{i,j}$

**Ensure:** : value  $x_i$  that minimizes  $\sum_{1 \leq i \leq N} f_i(x_i) + \sum_{1 \leq i \leq N} f_{i,i+1}(x_{i+1} - x_i)$

1: For  $i = 1 : N$

$$\begin{cases} M_i(x_i) = f_i(x_i) + M_{i-1,i}(x_i) \\ M_{i,i+1}(x_{i+1}) = \min_{x_i \in \mathbb{R}} (M_i(x_i) + f_{i,i+1}(x_{i+1} - x_i)) \end{cases}$$

{ $M_{i,i+1}$  should be computed for every value possible value of  $x_{i+1}$ , hence the algorithm presented this way is an abstract algorithm}

2: find  $x_n$  that minimizes  $M_N$

3: For  $i = 1 : N$  find  $x_i = \arg \min_{x \in \mathbb{R}} (M_i(x) + f_{i,i+1}(x_{i+1} - x_i))$

---

Indeed, in this case  $M_{i,i+1}(x_{i+1}) = \min_{x_i \in \mathbb{R}} (M_i(x_i) + f_{i,i+1}(x_{i+1} - x_i))$  is equivalent to find  $x_i$  that verifies  $0 \in \partial M_i(x_i) + \partial f_{i,i+1}(x_{i+1} - .)(x_i)$ , which is equivalent to:

$$\begin{cases} 0 \in \partial M_i(x_i) + \{-w_{i,i+1}^+\} & \text{if } x_i < x_{i+1} \\ 0 \in \partial M_i(x_i) + \{-w_{i,i+1}^-\} & \text{if } x_i > x_{i+1} \\ 0 \in \partial M_i(x_i) + [-w_{i,i+1}^+, -w_{i,i+1}^-] & \text{if } x_i = x_{i+1} \end{cases} \quad (3.22)$$

Thence, has minimizer is necessarily its sub-gradient on the interval  $[-w_{i,i+1}^+, -w_{i,i+1}^-]$ . We denote  $[\lambda_{i,i+1}^-, \lambda_{i,i+1}^+]$  the biggest interval such that  $w_{i,i+1}^- \in \partial M_i(\lambda_{i,i+1}^-)$  and  $w_{i,i+1}^+ \in \partial M_i(\lambda_{i,i+1}^+)$ .

Due to the monotony of the sub-gradients (property 9), we have that, for  $x \leq y$ ,  $\partial M_i(x) \leq \partial M_i(y)$ , (with the notation  $A \leq B : \forall (x, y) \in A \times B x \leq y$ ), hence from 3.22, all the minimizers are necessarily on this interval.

We denote  $m_i(x)$ , a sub-gradient of  $M_i$  on the point  $x$ .

-If  $m_i(x_{i+1}) > w_{i,i+1}^+$ , then  $m_i(x_{i+1}) > m_i(x_i)$ , with  $x_i$ , a minimizer of  $M_i(x_i) + f_{i,i+1}(x_{i+1} - x_i)$ , then  $x_{i+1} > x_i$ .

From 3.22, the minimizer  $x_i$ , that verify this last condition, verify necessarily that  $w_{i,i+1}^+ \in \partial M_i(x_i)$

We also have that:  $M_i(x_i) + f_{i,i+1}(x_{i+1} - x_i) = M_i(x_i) + w_{i,i+1}^+(x_{i+1} - x_i)$ , then every element that verify  $w_{i,i+1}^+ \in \partial M_i(x_i)$  is a minimizer of the expression.

$\lambda_{i,i+1}^+$ , verify this condition, thence, we can write:

$$M_{i,i+1}(x_{i+1}) = M_i(\lambda_{i,i+1}^+) + w_{i,i+1}^+(x_{i+1} - \lambda_{i,i+1}^+)$$

The function is thence linear and a sub-gradient of  $M_{i,i+1}$ , verify that

$$m_{i,i+1}(x_{i+1}) = w_{i,i+1}^+ = \text{clip}_{\{w_{i,i+1}^-, w_{i,i+1}^+\}}(m_i(x_{i+1})).$$

a minimizer of the expression is  $x_i = \lambda_{i,i+1}^+$

-If  $m_i(x_{i+1}) < w_{i,i+1}^-$ , we get a similar result:

$$m_{i,i+1}(x_{i+1}) = w_{i,i+1}^- = \text{clip}_{\{w_{i,i+1}^-, w_{i,i+1}^+\}}(m_i(x_{i+1})).$$

and the element  $x_i = \lambda_{i,i+1}^-$  is a minimizer of the expression .

-In the case  $m_i(x_{i+1}) \in [w_{i,i+1}^-, w_{i,i+1}^+]$ , this imply:  $0 \in \partial M_i(x_{i+1}) + [-w_{i,i+1}^+, -w_{i,i+1}^-]$  then  $0 \in \partial(M_i + f_{i,i+1}(x_{i+1} - .))(x_{i+1})$ , thence  $x_{i+1}$  is a minimizer, then

$$m_{i,i+1}(x_{i+1}) = m_i(x_{i+1}) = \text{clip}_{\{w_{i,i+1}^-, w_{i,i+1}^+\}}(m_i(x_{i+1}))$$

and the element  $x_i = x_{i+1}$  is a minimizer of the expression.

We get then that the equality

$$m_{i,i+1}(x_{i+1}) = \text{clip}_{\{w_{i,i+1}^-, w_{i,i+1}^+\}}(m_i(x_{i+1}))$$

and that the minimizer  $x_i$ , verify  $x_i = \text{clip}_{[\lambda_{i,i+1}^-, \lambda_{i,i+1}^+]}(x_{i+1})$

#### **Algorithm 5 Dynamic programming algorithm for the convex case, total variation as prior term $f_{i,i+1}$**

**Require:** Function  $f_i$  and functions  $f_{i,i+1}$  convexes functions, with  $f_{i,i+1}$  of the form 3.22

**Ensure:** : value  $x_i$  that minimizes  $\sum_{1 \leq i \leq N} f_i(x_i) + \sum_{1 \leq i \leq N} f_{i,i+1}(x_{i+1} - x_i)$

1: add a node to the chain  $f_{N,N+1}$  with  $w_{N,N+1}^- = w_{N,N+1}^+ = 0$  {it is necessary to}

2: For  $i = 1 : N$

$$\begin{cases} m_i(x_i) = g_i(x_i) + m_{i-1,i}(x_i) \\ m_{i,i+1}(x_{i+1}) = \text{clip}_{\{w_{i,i+1}^-, w_{i,i+1}^+\}}(m_i(x_{i+1})) \\ \text{find interval } [\lambda_i^-, \lambda_i^+] \text{ such that } w_{i,i+1}^- \in \partial M_i(\lambda_{i,i+1}^-) \text{ and } w_{i,i+1}^+ \in \partial M_i(\lambda_{i,i+1}^+) \end{cases}$$

{ $g_i$  is the a derivative of  $f_i$ : in our case , it is continuous}

3: choose  $x_n$  in  $[\lambda_N^-, \lambda_N^+]$

4: For  $i = 1 : N$   $x_i = \text{clip}_{[\lambda_{i,i+1}^-, \lambda_{i,i+1}^+]}(x_{i+1})$

The algorithm is still an abstract algorithm, because the calculus must be performed, for all the possible value of  $x_{i+1}$  on step 2. If we restrict the algorithm, for the case  $f_i(z) = \frac{1}{2}a_i z_i^2 - b_i z_i$ , the functions  $M_i$  are differentiable, and the derivatives that we denote  $m_i$  are piecewise affine-warped and non-increasing functions.

It is the same case for the function  $m_{i,i+1}$ , the article [29] proposes in this particular case, a concrete algorithm 6, to compute the minimum.

The messages are represented in this form by elements  $(s_0, \lambda_1, s_1, \dots, s_{N-1}, \lambda_N, s_N)$  and  $\bar{a}$ , with  $s_i + \bar{a}$ , the slopes and  $\lambda_i$  , the breakpoints, the points of discontinuity.  $\bar{a}$  is an artifice that limits the computation time.

The algorithm 6 is the algorithm used to resolve the problem, it computes the value  $\text{prox}_{TV_c}$  with a complexity of  $O(n)$ . From 3.21 , the computation of  $\text{prox}_{f_1}$  has a complexity of  $O(N_x N_y N_t)$ .

---

**Algorithm 6 Dynamic programming algorithm for the convex case, total variation as prior term  $f_{i,i+1}$ , and  $f_i$  quadratic**


---

**Require:** Function  $f_i$  and functions  $f_{i,i+1}$  convexes functions, with  $f_{i,i+1}$  of the form 3.22, and  $f_i(x) = \frac{1}{2}a_i^2x^2 + b_i x \forall x$

**Ensure:** : value  $x_i$  that minimizes  $\sum_{1 \leq i \leq N} f_i(x_i) + \sum_{1 \leq i \leq N} f_{i,i+1}(x_{i+1} - x_i)$

- 1: add a node to the chain  $f_{N,N+1}$  with  $w_{N,N+1}^- = w_{N,N+1}^+ = 0$
- 2:  $\bar{a} = a_1$  and  $m = (0)$
- 3: **for all i do**
- 4:    $\bar{a} = \bar{a} + a_i$
- 5:   Find biggest  $\lambda_p$  such that  $m_i(\lambda_p < w_{i,i+1}^-)$  by the following procedure: compute  $m_i(\lambda_1) = w_{i-1,i}^- + a_i \lambda_1 - b_i$  (which is true by the definition of  $M_i$ ), then compute until condition broken  $m_i(\lambda_{p+1}) + m_i(\lambda_p) + (s_p + \bar{a}_i)(\lambda_{p+1} - \lambda_p)$ , by adding 1 to p.
- 6:   Find smallest  $\lambda_r \geq \lambda_{p+1}$  such that  $m_i(\lambda_r > w_{i,i+1}^+)$  by the following procedure: compute  $m_i(\lambda_N) = w_{i-1,i}^+ + a_i \lambda_N - b_i$  (which is true by the definition of  $M_i$ ), then compute until condition broken  $m_i(\lambda_r) + m_i(\lambda_{r+1}) + (s_r + \bar{a}_i)(\lambda_{r+1} - \lambda_r)$ , by subtracting 1 to r.
- 7: **end for**
- 8: substitute  $(s_0^c, \lambda_1^c, s_1^c, \dots, s_{N-1}^c, \lambda_N^c, s_N^c)$  (with  $c$  standing for the current value) by  $(-\bar{a}, \lambda_l^-, s_{l+1}^c, \dots, s_{r-1}^c, \lambda_r^c, \bar{a})$
- 9: choose  $x_N \in [\lambda_N^-, \lambda_N^+]$
- 10: For  $i=1:n-1$   $x_i = clip[\lambda_{i,i+1}^-, \lambda_{i,i+1}^+](x_{i+1})$

---

We can thence describes, the computation of the step, via the computation of  $prox_{f_1^*}$  that is computed by the Moreau identity 15, and the ROF algorithm described just above:

We observe that  $A_1^* A_1$  is the identity operator, we can then choose  $\tau_1 = 1$ , such that  $A_1^* A_1 = \frac{1}{\tau_1} I$ . This choice implies that the instruction is equivalent to the computation of the proximal operator: Indeed, the step  $x_1^{k+1} = argmin_{x_1 \in X} (g_1(x_1) + \langle x_1, A_1^*(A_1 \bar{x}_1 + A_{2,3} \bar{x}_{2,3}) \rangle + \frac{\|x_1 - \bar{x}_1\|^2}{2\tau_1})$  is in this case equivalent to :

$$\begin{aligned} x_1^{k+1} &= argmin_{x_1 \in X} (g_1(x_1) + \langle x_1, (\bar{x}_1 + \bar{x}_2 + \bar{x}_3) \rangle + \frac{\|x_1 - \bar{x}_1\|^2}{2\tau_1}) \\ x_1^{k+1} &= argmin_{x_1 \in X} (g_1(x_1) + \frac{1}{2}(\|x_1 - \bar{x}_1\|^2 + 2\langle x_1, (\bar{x}_1 + \bar{x}_2 + \bar{x}_3) \rangle)) \\ x_1^{k+1} &= argmin_{x_1 \in X} (f_1^*(x_1) + \frac{1}{2}(\|x_1 - \bar{x}_1\|^2 + 2\langle x_1, (\bar{x}_1 + \bar{x}_2 + \bar{x}_3 - f) \rangle)) \\ x_1^{k+1} &= prox_{f_1^*}(f - \bar{x}_2 - \bar{x}_3) \end{aligned}$$

**Computation of the step  $x_{2,3}^{k+1} = argmin_{x_{2,3}} (g_{2,3}(x_2) + \langle x_{2,3}, A_{2,3}^*(A_1 \hat{x}_1 + A_{2,3} x_{2,3}^-) \rangle + \frac{\|x_{2,3} - \bar{x}_{2,3}\|^2}{2\tau_{2,3}})$  of algorithm 7:**

The algorithm, impose the choice of a value for  $\tau_{2,3}$  that verify  $\frac{I}{\tau_{2,3}} \geq A_{2,3}^* A_{2,3}$  Let's compute the value of  $A_{2,3}^* A_{2,3}$ :

We have that :

$$\forall x \in X \quad \forall y_{2,3} \in X^2 \quad \langle A_{2,3}^* x, (y_2, y_3) \rangle = \langle x, A_{2,3}(y_2, y_3) \rangle$$

then  $\langle x, y_2 + y_3 \rangle = \langle (x, x), (y_2, y_3) \rangle$

So we deduce that:  $A_{2,3}^* x = (x, x)$  Then:

$$\begin{aligned} \forall x_{2,3} \in X^2 \quad \langle A_{2,3}^* A_{2,3} x_{2,3}, x_{2,3} \rangle &= \langle (x_2 + x_3, x_2 + x_3), (x_2, x_3) \rangle \\ &= \|x_2 + x_3\|^2 \leq 2(\|x_2\|^2 + \|x_3\|^2) = \frac{1}{\tau_{2,3}} \langle (x_2, x_3), (x_2, x_3) \rangle \end{aligned}$$

with  $\tau_{2,3} = \frac{1}{2}$

The step takes this explicit form:

$$x_{2,3}^{k+1} = \underset{x_{2,3} \in X^2}{\operatorname{argmin}} (TV_{1Di}^*(x_2) - \langle x_2, f \rangle + TV_{1Di}^*(x_3) - \langle x_3, f \rangle + \langle x_2 + x_3, \bar{x}_2 + \bar{x}_3 + x_1^{k+1} \rangle + \frac{\|x_{2,3} - \bar{x}_{2,3}\|^2}{2\tau_{2,3}})$$

As long as there is not constraint, between the two variables:  $x_2$  and  $x_3$ , we get :

$$\begin{aligned} x_2^{k+1} &= \underset{x_2 \in X^2}{\operatorname{argmin}} (TV_{1Di}^*(x_2) - \langle x_2, f \rangle + \langle x_2, \bar{x}_2 + \bar{x}_3 + x_1^{k+1} \rangle + \frac{\|x_2 - \bar{x}_2\|^2}{2\tau_{2,3}}) \\ x_3^{k+1} &= \underset{x_3 \in X^2}{\operatorname{argmin}} (TV_{1Dj}^*(x_3) - \langle x_3, f \rangle + \langle x_3, \bar{x}_2 + \bar{x}_3 + x_1^{k+1} \rangle + \frac{\|x_3 - \bar{x}_3\|^2}{2\tau_{2,3}}) \end{aligned}$$

The two equations become:

$$\begin{aligned} x_2^{k+1} &= \underset{x_2 \in X}{\operatorname{argmin}} (TV_{1Di}^*(x_2) + \frac{1}{2\tau_{2,3}} [\|x_2 - \bar{x}_2\|^2 + 2\langle x_2 - \bar{x}_2, \tau_{2,3}(\bar{x}_2 + \bar{x}_3 + x_1^{k+1} - f) \rangle]) \\ x_3^{k+1} &= \underset{x_3 \in X}{\operatorname{argmin}} (TV_{1Dj}^*(x_3) + \frac{1}{2\tau_{2,3}} [\|x_3 - \bar{x}_3\|^2 + 2\langle x_3 - \bar{x}_3, \tau_{2,3}(\bar{x}_2 + \bar{x}_3 + x_1^{k+1} - f) \rangle]) \end{aligned}$$

This is equivalent to:

$$\begin{aligned} x_2^{k+1} &= \underset{x_2 \in X}{\operatorname{argmin}} (TV_{1Dj}^*(x_2) + \frac{1}{2\tau_{2,3}} [\|x_2 - (\bar{x}_2 - \tau_{2,3}(\bar{x}_2 + \bar{x}_3 + x_1^{k+1} - f))\|^2]) \\ x_3^{k+1} &= \underset{x_3 \in X}{\operatorname{argmin}} (TV_{1Di}^*(x_3) + \frac{1}{2\tau_{2,3}} [\|x_3 - (\bar{x}_3 - \tau_{2,3}(\bar{x}_2 + \bar{x}_3 + x_1^{k+1} - f))\|^2]) \end{aligned}$$

Then finally:

$$\begin{aligned} x_2^{k+1} &= \operatorname{prox}_{\tau_{2,3}TV_{1Di}^*}(\bar{x}_2 - \tau_{2,3}(\bar{x}_2 + \bar{x}_3 + x_1^{k+1} - f)) \\ x_3^{k+1} &= \operatorname{prox}_{\tau_{2,3}TV_{1Dj}^*}(\bar{x}_3 - \tau_{2,3}(\bar{x}_2 + \bar{x}_3 + x_1^{k+1} - f)) \end{aligned}$$

---

**Algorithm 7 proximal alternating descent With two variables (Explicit algorithm)**


---

**Require:** cost function  $g$ , and value of  $\lambda \in \mathbb{R}^+$

**Ensure:** : The minimizer of  $\min_{x_1, x_2 \in X_1 \times X_{2,3}} g_1(x_1) + g_{2,3}(x_{2,3}) + \frac{1}{2\lambda} \|A_1 x_1 + A_{2,3} x_{2,3}\|^2$  with  $g_1(x_1) = f_1^*(x_1) - \langle x_1, g \rangle$  and  $g_{2,3}(x_{2,3}) = f_2^*(x_2) - \langle x_2, g \rangle + f_3^*(x_3) - \langle x_3, g \rangle$ , with  $f_1(v) = TV_{1DV,g}(v)$  and  $f_2(v) = TV_{1Di}(v) + TV_{1Dj}(v)$ .

1: choose  $x^0 \in X$  and  $x^{-1} = x^0$  and  $t_0 = 0$

2: **for all**  $k$  until fixed number of iteration, or until the gap between the dual and the primal (property 4) is below a certain threshold do **do**

3:    $t_k = \frac{k+1}{2}$  {which is a possible choice that allows :  $t_k^2 \geq t_{k+1}(t_{k+1} - 1)$  and  $t_k > 1 \quad \forall k \geq 1$ }

4:    $\bar{x}_{2,3} = x_{2,3}^k + \frac{t_k - 1}{t_{k+1}}(x_{2,3}^k - x_{2,3}^{k-1})$

5:

$$x_1^{k+1} = prox_{TV_{1Dv,g}^*}(f - \bar{x}_2 - \bar{x}_3)$$

6:

$$\tau_{2,3} = \frac{1}{2}$$

7:

$$x_2^{k+1} = prox_{\tau_{2,3} TV_{1Di}^*}(\bar{x}_2 - \tau_{2,3}(\bar{x}_2 + \bar{x}_3 + x_1^{k+1} - f))$$

8:

$$x_3^{k+1} = prox_{\tau_{2,3} TV_{1Dj}^*}(\bar{x}_3 - \tau_{2,3}(\bar{x}_2 + \bar{x}_3 + x_1^{k+1} - f))$$

{use the algorithm 6 applied to each term of (equation 3.21) to compute  $prox_{\lambda_1 TV_{1D}}$ , then deduce  $prox_{\lambda_2 TV_{1D}^*}$  using the Moreau identity (property 15) }

9: **end for**

---

From, the explicit description of all the steps, we deduce the algorithm 7

### The gap between the dual and the primal

The problem 3.20, has the following expression:

$$D_1(x_1, x_2, x_3) = \langle x_1 + x_2 + x_3, f \rangle - f_1^*(x_1) - f_2^*(x_2) - f_3^*(x_3) - \frac{1}{2\lambda} \|x_1 + x_2 + x_3\|^2$$

It is the dual form of the primal problem:

$$P_1(u) = f_1(u) + f_2(u) + f_3(u) + \frac{\lambda}{2} \|u - f\|^2$$

with  $f_1 = TV_{1Dk,g}$ ,  $f_2 = TV_{1Di}$  and  $f_3 = TV_{1Dj}$

We have the following property:

**Property 4.**  $\forall u \in X \forall (x_1, x_2, x_3) \in X^3$

$$P_1(u) - D_1(x_1, x_2, x_3) \geq \frac{1}{2\lambda} \|x_1 + x_2 + x_3 + \lambda(u - f)\|^2$$

And

$$P_1(u) - D_1(x_1, x_2, x_3) \geq \frac{\lambda}{2} \|u - u^*\|^2$$

with  $u^*$  the infimum of the primal thence, the problem 3.17.

*Proof.*

$$\begin{aligned} P_1(u) - D_1(x_1, x_2, x_3) &= f_1(u) + f_2(u) + f_3(u) + \frac{\lambda}{2} \|u - f\|^2 - \langle x_1 + x_2 + x_3, f \rangle + \\ &\quad f_1^*(x_1) + f_2^*(x_2) + f_3^*(x_3) + \frac{1}{2\lambda} \|x_1 + x_2 + x_3\|^2 \\ &= [f_1(u) + f_1^*(x_1)] + [f_2(u) + f_2^*(x_2)] + [f_3(u) + f_3^*(x_3)] + \frac{\lambda}{2} \|u - f\|^2 + \frac{1}{2\lambda} \|x_1 + x_2 + x_3\|^2 - \langle x_1 + x_2 + x_3, f \rangle \end{aligned}$$

From the property 10 , we deduce:

$$\begin{aligned} P_1(u) - D_1(x_1, x_2, x_3) &\geq \langle u, x_1 \rangle + \langle u, x_2 \rangle + \langle u, x_3 \rangle + \frac{\lambda}{2} \|u - f\|^2 + \frac{1}{2\lambda} \|x_1 + x_2 + x_3\|^2 - \langle x_1 + x_2 + x_3, f \rangle \\ &\geq \langle u - f, x_1 + x_2 + x_3 \rangle + \frac{\lambda}{2} \|u - f\|^2 + \frac{1}{2\lambda} \|x_1 + x_2 + x_3\|^2 \\ &\geq \frac{1}{2\lambda} (\|x_1 + x_2 + x_3\|^2 + \lambda^2 \|u - f\|^2 + 2\langle \lambda(u - f), x_1 + x_2 + x_3 \rangle) \end{aligned}$$

Then

$$P_1(u) - D_1(x_1, x_2, x_3) \geq \frac{1}{2\lambda} \|x_1 + x_2 + x_3 + \lambda(u - f)\|^2$$

Denoting  $(x_1^*, x_2^*, x_3^*)$  the argument that maximize the dual, and since  $u^*$  is the infimum of the primal, and  $\inf P_1 = \sup D_1$ , we get from the inequality that  $u^* = f - \frac{x_1^* + x_2^* + x_3^*}{\lambda}$ .

The second result, comes from the definition of  $u^*$ , we have that:

$$P_1(u) - D_1(x_1, x_2, x_3) \geq P_1(u) - D_1(x_1^*, x_2^*, x_3^*) \geq \frac{1}{2\lambda} \|x_1^* + x_2^* + x_3^* + \lambda(u - f)\|^2 = \frac{\lambda}{2} \|u - u^*\|^2$$

□

Defining, the primal variable as  $u = f - \frac{x_1 + x_2 + x_3}{\lambda}$ , and since  $(x_1, x_2, x_3)$  converges to  $(x_1^*, x_2^*, x_3^*)$ ,  $u$  converges to  $u^*$ . By continuity , we get that  $P_1(u) - D_1(x_1, x_2, x_3)$  converges to 0, and this gap is also an indicator of the convergence of the algorithm.

## 4. Conclusion

Two subjects of 3D reconstruction, were tackled during this internship, the first one concerns the subject of bundle adjustment. The chosen solution assume that there is no geometrical inconsistencies, in the relative orientation between the two orientations of the cameras, during their respective acquisition.

The solution does not incorporate the effect of jitter, and doesn't include the method of inter-strip-ties (figure 2.1 and [1]), for its removal. It is also,a solution that can be applied, after the computation of the DEM, and thence simpler to integrate to the s2p pipeline.

In the case of s2p, the bundle adjustment (for the relative orientation), is replaced by the local pointing correction applied to each tiles, and the global pointing refinement step.The integration of the constraint of inter-strip-tie points is not trivial, as they don't belong to all the tiles.

Qualitatively,the particular stereo pair of images used for this work, seems to be not very affected by the jitter effect. From [11], there is small fraction of images for which, the jitter amplitude is greater than 1 pixel, and jitter less than about 2 pixels is tolerable, for the DEM production.

It would be interesting to test the actual algorithm for images affected by the jitter effect, and to integrate the inter-strip-ties points in the stereo pipeline, to study, possible improvements.

The registration between the adjacent CCDs, using registration between the overlapping section, is not a very rigourous approach, since one row of formed pixels, may be obtained under two differents direction (There is a delay in the formation of the same row between two adjacent CCDs, due to the displacement between them in the focal plane as shown in figure 1.2), and thence a more rigourous approach for bundle ajdustment and stereo-matching, would take into account the possibility that for some tiles, the image may be distorted, and the orientation badly defined, if they contain the image of two adjacent CCDs.

The difference in resolution, between the Mola point:160m between, two successive points, and the resolution of the HiRISE images (and dem): 30 cm per pixel, brings us to the following question: are the MOLA points sufficient, to register between the HiRISE DEM with the ground?

The HiRISE's data-set, is also very limited, there is 2 number of views per scene, and the whole image covers a very limited, and sparse part of the Mars' ground. One possible solution would be to exploit the CTX data, in the bundle adjustment process, and to apply two successive registration.

The second subject tackled during, this internship, concerns the subject of stereo matching. Two algorithms, were developed, the first one is an algorithm, that resolves a very spread model among the computer vision's community, but is computationally very low. The second algorithm, is an approximated version of the first algorithm, that is shown to be faster.

The validation, of the two algorithms, was qualitative, they weren't integrated yet in the s2p pipeline. One possible improvement for the two algorithms, would be to use the result obtained from a discrete algorithm (these algorithms are known to be faster), and to use it as an input.

Another improvement, would be to use the output of the second algorithm, as an input of the first algorithm (and compare the two results).

Another possible improvement for the two algorithms, would be to change the convex prior term, used for the two algorithms, for another non-convex term, that takes into account, the particularity of the ground of Mars, but this solution would involves, the development of another algorithm, because the prior term may be not convex.

## Bibliography

- [1] Precision processing of hirise stereo orbital images for topographic mapping on mars. *ASPRS 2010 annual conference*, April 2010.
- [2] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.
- [3] Daniel Greslou Carlo de Franchis, Enric Meinhardt-Llopis and Gabriele Facciolo. Attitude refinement for orbiting pushbroom cameras: a simple polynomial fitting method.
- [4] Antonin Chambolle. Continuous optimization ,an introduction. *Course*, 2016.
- [5] Antonin Chambolle and Jérôme Darbon. Image processing and analysis with graphs:a parametric maximum flow approach for discrete total variation regularization. *CRC Press*, 2012.
- [6] Thomas Pock Antonin Chambolle. A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. *MAI-Journal of computational mathematics*,, page 29–54, 2015.
- [7] Yunhang Chen. Bundle adjustment of mars hirise orbiter stereo images based on the rigorous sensor model.
- [8] Richard Szeliski Daniel Scharstein. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 2002.
- [9] Carlo de Franchis et al. An automatic and modular stereo pipeline for pushbroom images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.
- [10] David E.Smith et al. Mars orbiter laser altimeter: Experiment summary after the first year of global mapping of mars.
- [11] Kirk et al. Ultrahigh resolution topographic mapping of mars with mro hirise stereo images: Meter-scale slopes of candidate phoenix landing sites. *JOURNAL OF GEOPHYSICAL RESEARCH*, 2008.
- [12] Klaus Gwinner et al. Derivation and validation of high-resolution digital terrain models from mars express hrsc data.
- [13] Enric Meinhardt Gabriele Facciolo, Carlo de Franchis. Mgm: A significantly more global matching for stereovision. 2015.
- [14] D. Cremers H. Bischof T. Pock and A. Chambolle. Global solutions of variational models with convex regularization. *SIAM J. Imaging Sci.*,, pages 1122–1145,, jan 2010.
- [15] Richard Hartley. *Multiple view geometry in computer vision*. 2000.

- [16] JONATHAN M. BORWEIN HEINZ H. BAUSCHKE. On projection algorithmes for solving convex feasibility problems. *SIAM*, page 367–426, September 1996.
- [17] NASA <http://ode.rsl.wustl.edu/mars/dataPointSearch.aspx>.
- [18] HiRISE Operations Center <https://hirise.lpl.arizona.edu/>.
- [19] Karl Kunisch Kristian Bredies and Thomas Pock. Total generalized variation.
- [20] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*.
- [21] Javier Sanchez Luis Alvarez, Rachid Deriche and Joachim Weickert. Dense disparity map estimation respectingimage discontinuities: A pde and scale-space based approach.
- [22] Gianni Dal Maso. Integral representation on  $\text{bv}(\omega)$  of  $\gamma$ -limits of variational integrals.
- [23] NASA. *The Ames Stereo Pipeline*: April 3 2017.
- [24] Andrés Bruhn Natalia Slesareva and Joachim Weickert. Optic flow goes stereo : A variational method for estimating discontinuity-preserving dense disparity maps.
- [25] John Woodfill Ramin Zabih. Non-parametric local transforms for computing visual corre-spondance. *Computer Vision — ECCV'94*.
- [26] Thomas Pock René Ranftl, Stefan Gehrig and Horst Bischof. Pushing the limits of stereo using variational stereo estimation.
- [27] Yunhang Chen Rongxing Li, Juwon Hwangbo and Kaichang Di. Rigorous photogrammetric processing of hirise stereo imagery for mars topographic mapping. *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, july 2011.
- [28] H. Bischof T. Pock, D. Cremers and A. Chambolle. An algorithm for minimizing the mumford-shah functional. *International Conference on Computer Vision*, 2009.
- [29] Michal Rolinek Vladimir Kolmogorov, Thomas Pock. Total variation on a tree. *SIAM*, February 2015.
- [30] Yiran Wang and Bo Wu. Investigation of boresight offsets and co-registration of hirise and ctx imagery for precision mars topographic mapping.
- [31] Yoon and Shann. Photogrammetric analysis of the mars global surveyor mapping data.

## A. Some Elements of convex analysis

**Definition 1.** iff is proper, we define  $\partial f(x) = \{p \in X : \forall y \in X : f(y) \geq f(x) + \langle p, y - x \rangle\}$

**Definition 2.**  $f$  is coercive:  $\lim_{x \rightarrow +\infty} f(x) = +\infty$

**Definition 3.**  $f$  is lower-semi-continuous (lsc) if and only if:  $\forall x \in X \forall (x_n)_{n \in \mathbb{N}}$  that converges toward  $x \in X$ , we have that:  $f(x) \leq \liminf_{x_n \rightarrow x} f(x_n)$

**Definition 4.**  $f$  is proper:  $-\infty \notin \text{dom}(f)$  and and  $f$  not identically equal to  $+\infty$  :  $\text{dom}(f) \neq \emptyset$

We denote by  $\Gamma_0$  as the set of functions lower-semi-continuous, convex and proper, it is also a crucial set of functions on optimization.

**Property 5.** If  $f$  and  $g$  are in  $\Gamma_0$ , and  $(ir(\text{dom}(f)) \cap ir(\text{dom}(g))) \neq \emptyset$ , then  $\partial(f+g)(x) = \partial f(x) + \partial g(x) \quad \forall x \in X$ . it can be condensedly rewritten  $\partial(f+g) = \partial f + \partial g$

**Definition 5.**  $f$  is strongly convex:  $\exists \mu$  such that  $f$  is  $\mu$  strongly convex  $\Leftrightarrow f - \mu \frac{\| \cdot \|_2^2}{2}$  is convex

**Definition 6.**  $f$  is coercive:  $\lim_{x \rightarrow +\infty} f(x) = +\infty$

**Property 6.** If  $f$  is  $\mu$  strongly convex then it is coercive

**Property 7. The Law of Fermat**

for  $f$  proper function , we have that:  $0 \in \partial f(x) \Leftrightarrow x$  minimum of  $f$

**Property 8.** iff  $f$  is  $\mu$  strongly-convex then  $\forall (x, y) \in X \quad \forall p \in \partial f(x) : f(y) \geq f(x) + \langle p, y - x \rangle + \frac{\mu}{2} \|y - x\|^2$

**Corollary 1.** iff  $f$  is  $\mu$  strongly-convex then and  $\hat{x}$  is a minimizer of  $f$ , we have the practical corollary  $\forall y \in X : f(y) \geq f(\hat{x}) + \frac{\mu}{2} \|y - \hat{x}\|^2$

**Theorem 2.** For  $E$  reflective Banach space (a Hilbert space is in particular a reflective Banach space), and  $f : C \rightarrow \mathbb{R}$ , and  $C$  convex closed, non empty set included on  $E$ .

If  $f$  is in  $\Gamma_0$  then:

if  $C$  is bounded or  $f$  is coercive then,  $\exists x \in C$  that resolves  $\min_{x \in C} f$

If  $f$  is strongly convex or strictly convex, then the minimizer is unique

Remark: We have the following result:

**Corollary 2.** If  $f$  is in  $\Gamma_0$  and is strongly convex then, it has a unique minimizer , in every convex closed, none empty set

**Property 9.** For a proper function  $f$ :

$\forall (x, y) \in X \times X \quad \forall (p, q) \in \partial f(x) \times \partial f(y) : \langle p - q, x - y \rangle \geq 0$

**Definition 7.** For a function  $f$ , we define its conjugate:  $f^*(x) = \arg \min_{v \in X} (\langle v, x \rangle - f(v))$

**Property 10.**  $\forall u, x \in X^2 f^*(u) + f(x) \geq \langle u, x \rangle$

## B. Fista algorithm

- The proximal operator of a function  $f \in \Gamma_0$  is defined as:  $\text{prox}_f(x) = \underset{y \in X}{\arg\min}(f(y) + \frac{1}{2}||x - y||^2)$  It is well defined by the property 2.

The FISTA Algorithm resolves a problem with the following assumptions:

$$\min_{x \in X} F(x), \text{ with } F = f + g.$$

$f$  is a smooth convex function of type  $C^{1,1}$ , continuously differentiable, with lipshitz gradient  $L(f)$ :

$$i.e ||\nabla f(x) - \nabla f(y)|| \leq L(f) ||x - y||, \text{ with } L(f) > 0$$

$g$  is a continuous convex function which is possibly non smooth.

The algorithm is as follows ([4]):

FISTA Algorithm with fixed step (value of  $\tau$ )

**Require:**  $f$  and  $g$  with the above assumptions

**Ensure:** : A value on  $X$  near the minimizer of a problem of the form  $\min_{x \in X}(f(x) + g(x))$

1: choose  $x^0 \in X$  and  $x^{-1} = x^0$  and  $t_0 > 0$

2:  $\forall k \geq 0$  (until a fixed step) do

$$y^k = x^k + \beta_k(x^k - x^{k-1})$$

$$x^{k+1} = \text{prox}_{\tau g}(y^k - \tau \nabla f(y^k))$$

3: We would use  $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$  and  $\beta_k = \frac{t_k-1}{t_{k+1}}$

---

In this case, we have the following result:

**Theorem 3.** Assume  $t_0 = 0$  and let  $x^k$  be generated by the above algorithm , then, we have that:

$$F(x^k) - F(x^*) \leq \frac{2}{(k+1)^2} ||x^0 - x^*||^2$$

$$F(x^k) - F(x^*) \leq \frac{2}{k^2 \tau} ||x^0 - x^*||^2$$

(There are other variants of the algorithm (faster ones) in the case where  $f$  and  $g$  are strongly convex [4], but the rate doesn't exceed the value  $\frac{1}{k^2}$ )

**Corollary 3.** One particular case to is the case when  $F$  is  $\mu$  strongly convex.  $F$  can be rewritten as  $F=f+g$ , with  $g$  convex function, and  $g$  a quadratic function of the form  $x \rightarrow \mu \frac{||x||^2}{2}$ , in this case the Lipshitz constant  $L$  is equal to  $\mu$ , and  $\tau$  can be equal at most to  $\frac{1}{\mu}$ . Thence, the more the coefficient of the strongly convex function is big, the better the rate of the FISTA algorithm (with fixed rate) is big.

*Proof.* Assuming that  $\tau \leq \frac{1}{L}$  , we deduce from property (11) (below) that:  $F(x) + \frac{||x-x^k||^2}{2\tau} \geq F(x^{k+1}) + \frac{||x-x^{k+1}||^2}{2\tau}$

Replacing  $x$  , by the generic variable  $\frac{(t-1)x^k+x}{t}$  which is a convex combination of a generic variable  $x$  and  $x^k$ , with the generic scalar  $t \geq 1$ , we obtain , that:

$$t(t-1)(F(x^k) - F(x)) + \frac{\|(t-1)x^k + x - ty^k\|^2}{2\tau} \geq t^2(F(x^{k+1}) - F(x)) + \frac{\|(t-1)x^k + x - tx^{k+1}\|^2}{2\tau} \quad (\text{B.1})$$

The idea behind the FISTA algorithm's result is to choose the variable parameter  $t = t_{k+1}$  such that  $t_k^2 \geq t_{k+1}(1 + t_{k+1})$  (Generally, we choose the both term equal) and choose  $y^k$  to ensure that the term  $(t_{k+1} - 1)x^k + x - t_{k+1}x^{k+1}$  in the right side is the same as the term  $(t_{k+1} - 1)x^k + x - t_{k+1}y^k$  of the left hand side at the previous iterate, in order have an inequality of the form:  $X_n \geq X_{n+1}$ .

Precisely it means that  $(t_{k+1} - 1)x^k + x - t_{k+1}y^k = (t_k - 1)x^{k-1} + x - t_kx^k$ . Which gives the expression of  $y_k = \beta_k(x^k - x^{k-1}) + x^k$ , with  $\beta_k = \frac{t_k - 1}{t_{k+1}}$

We obtain then that

$$t_k^2(F(x^k) - F(x)) + \frac{\|(t_k - 1)x^k + x - t_kx^k\|^2}{2\tau} \geq t_{k+1}^2(F(x^{k+1}) - F(x)) + \frac{\|(t_{k+1} - 1)x^{k+1} + x - t_{k+1}x^{k+1}\|^2}{2\tau} \quad (\text{B.2})$$

Then summing the expression for  $k$  from 0 to  $n-1$ , and choosing (knowing that  $t_0 = 0$  and  $x^{-1} = x^0$  from the hypothesis) we obtain:

$$F(x^n) - F(x) + \frac{1}{2t_n^2\tau} \|(t_n - 1)x^n + x - t_nx^n\|^2 \leq \frac{1}{2t_n^2\tau} \|x^0 - x\|^2$$

Choosing  $t_{k+1} = t_k(1 + t_k)$ , we get  $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$ , then by induction, we can show that  $t_k \geq \frac{k}{2}$ , then, we get :

$$F(x^n) - F(x) \leq F(x^n) - F(x) + \frac{1}{2t_n^2\tau} \|(t_n - 1)x^n + x - t_nx^n\|^2 \leq \frac{1}{2t_n^2\tau} \|x^0 - x\|^2$$

Then by choosing the minimizer  $x^*$ , we get :

$$F(x^n) - F(x^*) \leq \frac{1}{2t_n^2\tau} \|x^0 - x^*\|^2$$

The, due to  $t_n \geq \frac{n}{2}$ , we get :

$$F(x^n) - F(x^*) \leq \frac{2}{n^2\tau} \|x^0 - x^*\|^2$$

□

**Property 11.** If  $\hat{x} = \text{prox}_{\tau g}(\bar{x} - \tau \nabla f(\bar{x}))$ , and  $x \rightarrow \nabla f(x)$  is L lipschitz then:

$$F(x) + \frac{\|x - \bar{x}\|^2}{2\tau} \geq F(\hat{x}) + \frac{1 - \tau L}{\tau} \frac{\|\hat{x} - \bar{x}\|^2}{2} + \frac{\|x - \hat{x}\|^2}{2\tau}$$

*Proof.* Since  $\hat{x}$  is the minimizer of  $x \rightarrow g(x) + \frac{1}{2\tau} \|\bar{x} - \tau \nabla f(\bar{x}) - x\|^2$  (by the definition of the proximal operator), we have by developing that  $\hat{x}$  is also the minimizer of

$$x \rightarrow g(x) + f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x} \rangle + \frac{\|x - \bar{x}\|^2}{2\tau} \quad (\text{B.3})$$

This last function is a  $\frac{1}{\tau}$  strongly convex function. By convexity of  $f$  applied at the point  $\bar{x}$  (property 8), we have that:

$$f(x) \geq f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x} \rangle$$

We have then that by adding a term on the both side:

$$F(x) + \frac{\|x - \bar{x}\|^2}{2\tau} \geq g(x) + f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x} \rangle + \frac{\|x - \bar{x}\|^2}{2\tau}$$

And since  $\hat{x}$  is a minimizer of the  $\frac{1}{\tau}$  strongly convex function B.3, by the property (1), we get :

$$g(x) + f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x} \rangle + \frac{\|x - \bar{x}\|^2}{2\tau} \geq g(\hat{x}) + f(\bar{x}) + \langle \nabla f(\bar{x}), \hat{x} - \bar{x} \rangle + \frac{\|\hat{x} - \bar{x}\|^2}{2\tau} + \frac{\|x - \hat{x}\|^2}{2\tau}$$

Applying it to the previous inequality, we get:

$$F(x) + \frac{\|x - \bar{x}\|^2}{2\tau} \geq g(\hat{x}) + f(\bar{x}) + \langle \nabla f(\bar{x}), \hat{x} - \bar{x} \rangle + \frac{\|\hat{x} - \bar{x}\|^2}{2\tau} + \frac{1}{\tau} \frac{\|x - \hat{x}\|^2}{2}$$

And finally, since  $\nabla F$  is L-lipshitz, we have that:  $f(\bar{x}) + \langle \nabla f(\bar{x}), \hat{x} - \bar{x} \rangle \geq f(\hat{x}) - \frac{L}{2} \|\hat{x} - \bar{x}\|^2$ , which leads to:

$$F(x) + \frac{\|x - \bar{x}\|^2}{2\tau} \geq g(\hat{x}) + f(\hat{x}) - \frac{L}{2} \|\hat{x} - \bar{x}\|^2 + \frac{\|\hat{x} - \bar{x}\|^2}{2\tau} + \frac{1}{\tau} \frac{\|x - \hat{x}\|^2}{2}$$

We obtain finally (by reordering) our result:

$$F(x) + \frac{\|x - \bar{x}\|^2}{2\tau} \geq F(\hat{x}) + \frac{\|x - \hat{x}\|^2}{2\tau} + \frac{1 - \tau L}{\tau} \frac{\|\hat{x} - \bar{x}\|^2}{2}$$

□

## C. Fista Algorithm with two coordinates

There is another interesting variant of the Fista algorithm, that resolves problems of the form.

$$\min_{x_1, x_2 \in X} F(x_1, x_2), \text{ with } F(x_1, x_2) = g_1(x_1) + g_2(x_2). \text{ with:}$$

$f_1$  is a smooth convex function of type  $C^{1,1}$ , continuously differentiable, with lipshitz gradient  $L(f_1)$ : i.e  $\|\nabla f_1(x) - \nabla f_1(y)\| \leq L(f_1)\|x - y\|$ , with  $L(f_1) > 0$

$f_2$  is a continuous convex function which is possibly non smooth.

In the case of this report, the particular case of interest is the following:

$$(D_1): \min_{x_1, x_2 \in X} [G(x_1, x_2) = g_1(x_1) + g_2(x_2) + \frac{1}{2\lambda}\|x_1 + x_2\|^2] \quad (\text{C.1})$$

The FISTA Algorithm presented in appendix B, applied to the problem (3.20) with the variable  $(x_1, x_2)$  would give the result  $G(x_1^k, x_2^k) - G(x_1^*, x_2^*) \leq 4 \frac{|x_1^* - x_1^0|^2 + |x_2^* - x_2^0|^2}{k^2}$ , with  $G(x_1, x_2) = g_1(x_1) + g_2(x_2) + \frac{1}{2\lambda}\|x_1 + x_2\|^2$ .

It is possible to exploit the same developments used in the proof for the convergence of the algorithm FISTA (B), in order to have a sharper inequality:

The proof is from the article ([6]), and is rewritten here.

**Property 12.** Let  $x_0 = x_{-1}$  and  $\forall k$  defines recursively  $\bar{x}_2^k = x_2^k + \frac{t_k - 1}{t_{k+1}}(x_2^k - x_2^{k-1})$ ,  $x_1^{k+1}$  that minimizes  $G(., \bar{x}_2^k)$  and  $x_2^{k+1}$  that minimizes  $G(x_1^{k+1}, .)$ , then:

$$G(x^k) - G(x^*) \leq 2 \frac{|x_2^* - x_2^0|^2}{\lambda k^2}$$

, with  $x^*$  the minimum .

*Proof.* The case  $\lambda \neq 1$ , can be deduced from the case  $\lambda = 1$ , by switching  $G$  with  $\lambda G$

It is thence, sufficient to treat the case  $\lambda = 1$ :

From the property below (13) applied to  $G$ , we obtain that:

$$G(x) + \frac{1}{2}\|x_2 - \bar{x}_2^k\|^2 \geq G(x^{k+1}) + \frac{1}{2}\|x_2 - x_2^{k+1}\|^2$$

Replacing the variable  $x$  by the generic variable  $\frac{t-1}{t}x^k + \frac{1}{t}x$   $\forall t > 1$ , and using the convexity of  $G$ , we deduce:

$$\frac{t-1}{t}G(x^k) + \frac{1}{t}G(x) + \frac{1}{2}\left\|\frac{t-1}{t}x^k + \frac{1}{t}x_2 - \bar{x}_2^k\right\|^2 \geq G(x^{k+1}) + \frac{1}{2}\left\|\frac{t-1}{t}x_2^k + \frac{1}{t}x_2 - x_2^{k+1}\right\|^2 \quad \forall x \quad \forall t > 1.$$

Then:  $\frac{t-1}{t}(G(x^k) - G(x)) + \frac{1}{2}\left\|\frac{t-1}{t}x^k + \frac{1}{t}x_2 - \bar{x}_2^k\right\|^2 \geq (G(x^{k+1}) - G(x)) + \frac{1}{2}\left\|\frac{t-1}{t}x_2^k + \frac{1}{t}x_2 - x_2^{k+1}\right\|^2$ .  
Multiplying by  $t^2$ , we get:

$$(t-1)t(G(x^k) - G(x)) + \frac{1}{2}\|(t-1)x_2^k + x_2 - t\bar{x}_2^k\|^2 \geq t^2(G(x^{k+1}) - G(x)) + \frac{1}{2}\|(t-1)x_2^k + x_2 - tx_2^{k+1}\|^2$$

Replacing again  $t$  by the sequence  $t_{k+1}$ , and imposing the recurrence definition :  $t_k^2 = t_{k+1}(t_{k+1} - 1)$  (there is always a solution strictly than 1, thus the equation is valid for each  $k$ , more precisely, we have that the solution  $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$ , is strictly bigger than 1 if  $t_k \neq 0$ ), we obtain:

$$t_k^2(G(x^k) - G(x)) + \frac{1}{2}||(t_{k+1}-1)x_2^k + x_2 - t_{k+1}\bar{x}_2^k||^2 \geq t_{k+1}^2(G(x^{k+1}) - G(x)) + \frac{1}{2}||(t_{k+1}-1)x_2^k + x_2 - t_{k+1}x_2^{k+1}||^2$$

In order to have a recurring proof like in the proof in appendix, we can impose:

$$(t_{k+1}-1)x_2^k + x_2 - t_{k+1}\bar{x}_2^k = (t_k-1)x_2^{k-1} + x_2 - t_kx_2^k$$

$$\text{This is equivalent to the choice of } \bar{x}_2^k = x_2^k + \frac{t_k-1}{t_{k+1}}(x_2^k - x_2^{k-1}).$$

We obtain then the recurrence :

$$t_k^2(G(x^k) - G(x)) + \frac{1}{2}||(t_k-1)x_2^{k-1} + x_2 - t_kx_2^k||^2 \geq t_{k+1}^2(G(x^{k+1}) - G(x)) + \frac{1}{2}||(t_{k+1}-1)x_2^k + x_2 - t_{k+1}x_2^{k+1}||^2$$

Then ,choosing  $t_0 = \epsilon$ , and  $x_2^{-1} = x_2^0$  imply that  $t_k > 1$  for all the values of  $k$ , we get then :

$$\epsilon^2(G(x^k) - G(x)) + \frac{1}{2}||x_2 - x_2^0||^2 \geq t_k^2(G(x^k) - G(x)) + \frac{1}{2}||(t_k-1)x_2^{k-1} + x_2 - t_kx_2^k||^2$$

The values  $t_k$  are continuous with respect to  $\epsilon$ , converging  $\epsilon$  to 0, gives:

$$\frac{1}{2}||x_2 - x_2^0||^2 \geq t_k^2(G(x^k) - G(x)) + \frac{1}{2}||(t_k-1)x_2^{k-1} + x_2 - t_kx_2^k||^2$$

Then, we get

$$\frac{1}{2t_k^2}||x_2 - x_2^0||^2 \geq (G(x^k) - G(x))$$

For the particular choice of  $x = x^*$ , we obtain:

$$G(x^k) - G(x^*) \leq \frac{1}{2t_k^2}||x_2^* - x_2^0||^2$$

We have the recurrent definition  $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$ , the if  $t_k \geq \frac{k}{2}$ , we get  $t_{k+1} \geq \frac{k+1}{2}$ , and since  $t_0 = 0$ , we obtain finally , the result:

$$G(x^k) - G(x^*) \leq \frac{2}{k^2}||x_2^* - x_2^0||^2$$

, which prooves the case  $\lambda = 1$  then the remaining cases.

□

**Property 13.** If we consider the transform  $T\bar{x} = \hat{x}$ , such that:  $\hat{x}_1 = \underset{x_1}{\operatorname{argmin}} g_1(x_1) + \frac{1}{2\lambda}||x_1 + \bar{x}_2||^2$

$$\hat{x}_2 = \underset{x_2}{\operatorname{argmin}} g_2(x_2) + \frac{1}{2\lambda}||\hat{x}_1 + x_2||^2$$

with the notation  $x = (x_1, x_2)$

and  $\lambda \in \mathbb{R}^+$

Then, we have the following inequality (similar to property 11 in the case  $\tau = \frac{1}{L}$ ):

$$\lambda G(x) + \frac{1}{2}||x_2 - \bar{x}_2||^2 \geq \lambda G(\hat{x}) + \frac{1}{2}||x_2 - \hat{x}_2||^2$$

*Proof.* The case  $\lambda \neq 1$  can be deduced from the case  $\lambda = 1$  by replacing  $G$  to  $\lambda G$ . So it is sufficient to treat the case  $\lambda = 1$ .

Since  $\hat{x}_1$  is the minimizer of  $x_1 \rightarrow g_1(x_1) + \frac{1}{2}||x_1 + \bar{x}_2||^2$ , which is a  $1$ -strongly convex function, which leads to the inequality (by the property 1):

$$g_1(x_1) + \frac{1}{2}||x_1 + \bar{x}_2||^2 \geq g_1(\hat{x}_1) + \frac{1}{2}||\hat{x}_1 + \bar{x}_2||^2 + \frac{1}{2}||x_1 - \hat{x}_1||^2$$

$\hat{x}_2$  is the minimizer of  $x_2 \rightarrow g_2(x_2) + \frac{1}{2}||\hat{x}_1 + x_2||^2$ , which is a 1-strongly convex function, which leads to a similar inequality:

$$g_2(x_2) + \frac{1}{2}||\hat{x}_1 + x_2||^2 \geq g_2(\hat{x}_2) + \frac{1}{2}||\hat{x}_1 + \hat{x}_2||^2 + \frac{1}{2}||x_2 - \hat{x}_2||^2$$

Summing the two inequalities and adding the term  $\frac{1}{2}||x_1 + x_2||^2$ , gives:

$$\begin{aligned} G(x) &= g_1(x_1) + g_2(x_2) + \frac{1}{2}||x_1 + x_2||^2 \geq g_1(\hat{x}_1) + \frac{1}{2}||\hat{x}_1 + \bar{x}_2||^2 + \frac{1}{2}||x_1 - \hat{x}_1||^2 + g_2(\hat{x}_2) \\ &\quad + \frac{1}{2}||\hat{x}_1 + \hat{x}_2||^2 + \frac{1}{2}||x_2 - \hat{x}_2||^2 - \frac{1}{2}||\hat{x}_1 + x_2||^2 - \frac{1}{2}||x_1 + \bar{x}_2||^2 + \frac{1}{2}||x_1 + x_2||^2 \end{aligned} \quad (\text{C.2})$$

Which is the same as:

$$G(x) \geq G(\hat{x}) + \frac{1}{2}||\hat{x}_1 + \bar{x}_2||^2 + \frac{1}{2}||x_1 - \hat{x}_1||^2 + \frac{1}{2}||x_2 - \hat{x}_2||^2 - \frac{1}{2}||\hat{x}_1 + x_2||^2 - \frac{1}{2}||x_1 + \bar{x}_2||^2 + \frac{1}{2}||x_1 + x_2||^2 \quad (\text{C.3})$$

Then :

$$\begin{aligned} G(x) &\geq G(\hat{x}) + \frac{1}{2}||x_1 - \hat{x}_1||^2 + \frac{1}{2}||x_2 - \hat{x}_2||^2 \\ &\quad + \frac{1}{2}||\hat{x}_1 + \bar{x}_2||^2 - \frac{1}{2}||\hat{x}_1 + x_2||^2 - \frac{1}{2}||x_1 + \bar{x}_2||^2 + \frac{1}{2}||x_1 + x_2||^2 \end{aligned} \quad (\text{C.4})$$

The term  $\frac{1}{2}||\hat{x}_1 + \bar{x}_2||^2 - \frac{1}{2}||\hat{x}_1 + x_2||^2 - \frac{1}{2}||x_1 + \bar{x}_2||^2 + \frac{1}{2}||x_1 + x_2||^2$  is equal to  $(x_1 - \hat{x}_1)(x_2 - \bar{x}_2) = \frac{1}{2}||x_1 + x_2 - (\hat{x}_1 + \hat{x}_2)||^2 - \frac{1}{2}||x_1 - \hat{x}_1||^2 - \frac{1}{2}||x_2 - \bar{x}_2||^2$

The previous inequality boils down to:

$$\begin{aligned} G(x) &\geq G(\hat{x}) + \frac{1}{2}||x_1 - \hat{x}_1||^2 + \frac{1}{2}||x_2 - \hat{x}_2||^2 + \frac{1}{2}||x_1 + x_2 - (\hat{x}_1 + \hat{x}_2)||^2 - \frac{1}{2}||x_1 - \hat{x}_1||^2 - \frac{1}{2}||x_2 - \bar{x}_2||^2 \\ &= G(\hat{x}) + \frac{1}{2}||x_2 - \hat{x}_2||^2 + \frac{1}{2}||x_1 + x_2 - (\hat{x}_1 + \hat{x}_2)||^2 - \frac{1}{2}||x_2 - \bar{x}_2||^2 \end{aligned} \quad (\text{C.5})$$

We get then the result:

$$G(x) + \frac{1}{2}||x_2 - \bar{x}_2||^2 \geq G(\hat{x}) + \frac{1}{2}||x_2 - \hat{x}_2||^2 + \frac{1}{2}||x_1 + x_2 - (\hat{x}_1 + \hat{x}_2)||^2 \geq G(\hat{x}) + \frac{1}{2}||x_2 - \hat{x}_2||^2 \quad (\text{C.6})$$

□

## D. Computation of the conjugate of the total variation

In this appendix, we compute  $TV_c^*$ , with:  $TV_c(v) = \sum_{k \in [|1, N-1|]} |(\nabla v)_k| c_k$ . and  $N$  the size of the one-dimensional vector  $\nabla v$ , and  $c$ .

$$TV_c^*(x) = \sup_{v \in \mathbb{R}^{[1, N]}} \sum_{j=1:N} x_j v_j - \sum_{j=1:N-1} |v_j - v_{j+1}| c_j.$$

The application that associates  $v$  to  $V$  such that  $V_j = v_j - v_{j-1} \forall j \in [2, N]$  and  $V_1 = v_1$  is a bijective application.

We have in particular:  $v_j = \sum_{j_1 \in [|1, j|]} V_{j_1} \forall j \in [|1, N|]$

Applying this transformation on the expression of  $TV_c^*$  gives:  $TV_c^*(x) = \sup_{V \in \mathbb{R}^{[1, N]}} \sum_{j=1:N} x_j (\sum_{j_1=1:j} V_{j_1}) - \sum_{j=1:N-1} |V_{j+1}| c_j$

Then:

$$TV_c^*(x) = \sup_{V \in \mathbb{R}^{[1, N]}} \sum_{j_1=1:N} (\sum_{j=j_1:N} V_{j_1} x_j) - \sum_{j=2:N} |V_j| c_{j-1}$$

$$TV_c^*(x) = \sup_{V \in \mathbb{R}^{[1, N]}} \sum_{j=1:N} V_j (\sum_{j_1=j:N} x_{j_1}) - \sum_{j=2:N} |V_j| c_{j-1}$$

Then,

$$TV_c^*(x) = \sup_{V \in \mathbb{R}^{[1, N]}} V_1 (\sum_{j_1=1:N} x_{j_1}) + \sum_{j=2:N} V_j (\sum_{j_1=j:N} x_{j_1}) - \sum_{j=2:N} |V_j| c_{j-1}$$

Then,

$$TV_c^*(x) = \sup_{V \in \mathbb{R}^{[1, N]}} V_1 (\sum_{j_1=1:N} x_{j_1}) + \sum_{j=2:N} [V_j (\sum_{j_1=j:N} x_{j_1}) - |V_j| c_{j-1}]$$

Choosing  $V_j = 0 \forall j \neq 1$ , gives

$$TV_c^*(x) \geq \sup_{V_1 \in \mathbb{R}} V_1 (\sum_{j_1=1:N} x_{j_1})$$

Then  $TV_c^*(x) = +\infty$  if  $\sum_{j_1=1:N} x_{j_1} \neq 0$ .

Again, choosing  $V_j = 0 \forall j \neq j_0$

$$TV_c^*(x) \geq \sup_{V_{j_0} \in \mathbb{R}} [V_{j_0} (\sum_{j_1=j_0:N} x_{j_1}) - |V_{j_0}| c_{j_0-1}] \geq \sup_{\lambda \in \mathbb{R}^+} [\lambda (\sum_{j_1=j_0:N} x_{j_1}) - c_{j_0-1}]$$

If  $|\sum_{j_1=j_0:N} x_{j_1}| > c_{j_0-1}$ , then  $TV_c^*(x) = +\infty$

If  $|\sum_{j_1=j:N} x_{j_1}| \leq c_{j-1} \forall j \in [|2, N|]$  and  $\sum_{j_1=1:N} x_{j_1} = 0$ , then:

$$\begin{aligned} TV_c^*(x) &= \sup_{V \in \mathbb{R}^{[1, N]}} V_1 (\sum_{j_1=1:N} x_{j_1}) + \sum_{j=2:N} [V_j (\sum_{j_1=j:N} x_{j_1}) - |V_j| c_{j-1}] = \\ &\quad \sup_{V \in \mathbb{R}^{[1, N]}} \sum_{j=2:N} [V_j (\sum_{j_1=j:N} x_{j_1}) - |V_j| c_{j-1}] \end{aligned} \quad (\text{D.1})$$

The term  $[V_j(\sum_{j_1=j:N} x_{j_1}) - |V_j|c_{j-1}] \leq 0$  from the hypothesis, then

$$TV_c^*(x) \leq 0$$

And choosing  $V_j = 0 \forall j$ , we deduce the bound is reached, thence:

$$TV_c^*(x) = \delta_{\{y \in \mathbb{R}^N : |\sum_{j_1=j}^{j_1=N} y_{j_1}| \leq c_{j-1} \forall j \in [|2, N|] \text{ and } \sum_{j_1=1}^{j_1=N} y_{j_1} = 0\}}(x)$$

Which is equivalent to:

$$TV_c^*(x) = \delta_{\{y \in \mathbb{R}^N : |\sum_{j_1=1}^{j_1=j} y_{j_1}| \leq c_j \forall j \in [|1, N-1|] \text{ and } \sum_{j_1=1}^{j_1=N} y_{j_1} = 0\}}(x)$$

## E. Elements of continuous optimization

**Property 14.** For  $K$  closed convex set,  $\delta_K$ , is a lower-semi-continuous, convex and proper function, the proximal operator is well defined, thence:

$$\begin{aligned} \text{prox}_{\lambda\delta_K}(x) &= \underset{v \in X}{\operatorname{argmin}} \lambda\delta_K(v) + \frac{1}{2}\|x - v\|^2 \\ &= \underset{v \in K}{\operatorname{argmin}} \|x - v\|^2 = \text{proj}_K(x) \end{aligned}$$

**Property 15.** Moreau identity For flower-semi-continuous, convex and proper defined for  $X$  Hilbert Space, we have the identity

$$\forall x \in X \quad x = \text{prox}_{\tau f}(x) + \text{prox}_{\frac{1}{\tau}f^*}\left(\frac{x}{\tau}\right)$$

## F. Hausdorff measure

Defining

$$H_\delta^2(S) = \frac{\alpha_2}{4} \inf \left\{ \sum_{i=1}^{\infty} \text{diam}(U_i)^2 : S \subset \bigcup_{i=1}^{\infty} U_i \text{ and } \text{diam}(U_i) < \delta \right\}$$

with  $\alpha_2 = \pi$  the volume of the unit ball in the 2-dimensional space.

and with  $\text{diam}(U) = \sup\{\|x - y\| : \forall x, y \in U \subset \mathbb{R}^3\}$ , and  $\text{diam}(\emptyset) = 0$

We define the 2-dimensional Hausdorff measure in the three dimensional space as

$$\mathcal{H}^2(S) = \sup_{\delta>0} H_\delta^2(S) = \lim_{\delta \rightarrow 0} H_\delta^2(S)$$

We can define the 2-dimensional Hausdorff measure for every possible space of dimension greater or equal 2.

The Hausdorff measure generalizes the Lebesgue measure, and gives a number in  $[0, +\infty]$  for every set of  $\mathbb{R}^n \forall n \in \mathbb{N}$ .

This measure gives a number from 0 to  $+\infty$  to each set of  $\mathbb{R}^3$ . For a continuous set contained on a 2-dimensional affine space, this measure gives its area, for a set which is not contained on a 2 dimensional space this measure gives  $\infty$ , and for a set included in a one dimensional affine space, the measure gives the value 0.

This definition implies in particular that for every Borel set  $\mathcal{H}^2(S) = \lambda_2(S)$ , with  $\lambda_2$  the Borel measure (not proven here).

Now let's prove the following property:

**Property 16.** *for  $h$  continuous for the two first variable, convex , one homogeneous with respect to the last variable, and such that the function  $x \rightarrow h(x, u(x), (\nabla u(x), -1))$  is integrable , we have that:*

$$\int_{\Gamma_u} h(x, t, v_{\Gamma_u}(x)) d(\mathcal{H}^2)(x) = \int_{\Omega} h(x, u(x), (\nabla u(x), -1)) d\lambda$$

with  $v_{\Gamma_u}(x) = \frac{(\nabla u(x), -1)}{\sqrt{1 + \|\nabla u(x)\|^2}}$ ,  $\lambda$  the Lebesgue measure, and  $\Gamma_u$ , the boundary of the function:

$$1_u = \begin{cases} 1 & \text{if } u(x) > t \\ 0 & \text{otherwise} \end{cases}$$

*Proof.* We have that the term  $\int_{\Omega} h(x, u(x), (\nabla u(x), -1))$  is well defined from the integrability hy-

pothesis, then by 1-homogeneity:

$$\begin{aligned}
\int_{\Omega} h(x, u(x), (\nabla u(x), -1)) d\lambda &= \int_{\Omega} \sqrt{1 + ||\nabla u(x)||^2} h(x, u(x), v_{\Gamma_u}(x)) d\lambda \\
&= \int_{\Omega \times 0} \sqrt{1 + ||\nabla u(x)||^2} h(x, u(x), v_{\Gamma_u}(x)) d\lambda \\
&= \int_{\Omega \times 0} \sqrt{1 + ||\nabla u(x)||^2} h(x, u(x), v_{\Gamma_u}(x)) d\mathcal{H}^2(x, 0)
\end{aligned}$$

Where  $\lambda$  in the penultimate line, is the Lebesgue measure on the space  $\mathbb{R}^3$ .

The last equality comes from the equivalence between the Hausdorff and the Lebesgue measure for the Borel set  $\Gamma$ , the equality can be proven, by writing the integral as the limit of integral of sum of weighted indicator function (i.e of the form  $1_{A_i}$ ).

Then:

$$\int_{\Omega} h(x, u(x), (\nabla u(x), -1)) d\lambda = \int_{\Omega} \sqrt{1 + ||\nabla u(x)||^2} h_1(\Phi^{-1})(x, 0) d\mathcal{H}^2(x, 0)$$

With:  $\Phi : \Gamma_u \subset \mathbb{R}^3 \rightarrow \Omega \times \{0\} \subset \mathbb{R}^3$ , such that  $\Phi(x, t) = (x, 0)$

And  $h_1(x, t) = h(x, t, v_{\Gamma_u}(x))$ .

As, we consider the functions of  $BV(\Omega)$ , these functions are continuous almost everywhere, because they are differentiable almost everywhere, thence for nearly every  $x \in \Omega$  :  $(x, u(x)) \in \Gamma_u$ .

Then, we have that  $\Phi^{-1}$  is well defined, with  $\forall x \in \Omega \quad \Phi^{-1}(x, 0) = (x, u(x))$ , then  $\Phi$  is bijective, differentiable everywhere, and its inverse is differentiable almost everywhere, then  $\Phi$  is a  $C^1$  diffeomorphism.

We have that  $J_{\Phi^{-1}}(x, 0) = \sqrt{1 + ||\nabla u(x)||^2}$ , then, getting back to our equation, it results:

$$\int_{\Omega} h(x, u(x), (\nabla u(x), -1)) d\lambda = \int_{\Omega} J_{\Phi^{-1}}(x, 0) h_1(\Phi^{-1})(x, 0) d\mathcal{H}^2(x, 0)$$

With a change of variable, by the  $C^1$  diffeomorphism  $\Phi^{-1}$  we get that:

$$\int_{\Omega} h(x, u(x), (\nabla u(x), -1)) d\lambda = \int_{\Gamma_u} h_1(x, t) d\mathcal{H}^2(x, t) = \int_{\Gamma_u} h(x, t, v_{\Gamma_u}(x)) d\mathcal{H}^2(x, t)$$

That concludes the proof  $\square$

## G. Projective Geometry

In order to represent a pixel of an image, we will choose to represent them on homogeneous coordinates.

Homogeneous coordinates are obtained by a mapping between the set  $\mathbb{R}^2$  and  $\mathbb{R}^3 - (0,0,0)$  that associate to each point  $(x,y)$  of the image a point  $(x,y,1)$  on the second set. We define a set of equivalence on the arrival set, by the relation equivalence " $=$ " such that  $(x, y, 1) = (\lambda x, \lambda y, \lambda) \forall \lambda \in \mathbb{R}^* \forall x, y$ . The quotient space is called the projective space, denoted  $\mathbb{P}^2$ , and all the variable would be represented on that space.

This mapping is clearly an invertible one, when the arrival space is replaced by  $\mathbb{P}^2$ .

The interpretation on terms of a camera model is the following:

If  $(0,0,0)$  corresponds to the center of the camera on the projective space, the homogeneous coordinate  $(x,y,1)$  represents the line passing through  $(x,y)$  and the center of the camera, thence, the physical ray coming toward the point  $(x,y)$ .

A physical image is obtained by a clipping of the space  $\mathbb{R}^2$  on a rectangular domain or its clipping  $\mathbb{P}^2$ , on its equivalent, thence when we apply a transformation on an image, it is applied only on this domain.

We represents also the lines as point on the homogeneous space. The line defined implicitly by the equation:  $ax + by + c = 0$  can be represented as  $l=(a,b,c)$  involving the following property.

**Property 17.** *If  $x$  lies on a line  $l$ , then  $l.x=0$  which is equivalent to  $x^T.l = 0$*

We also have this useful property:

**Property 18.** *If  $x$  and  $x'$  lies on a line  $l$ , then  $l = x \times x'$*

For the purpose of image-rectification (presented in appendix H), we need to represent, the transformation between two images. We will be interested on a particular set of transformations, the projective transformations.

A projective transformation  $h$  , is an invertible mapping from  $\mathbb{P}^2$  to  $\mathbb{P}^2$  such that:

If  $x_1, x_2$  and  $x_3$  are collinear then  $h(x_1), h(x_2)$  and  $h(x_3)$  are also collinear.

A theorem from the reference [15], stipulates that :

**Theorem 4.** *A mapping  $h: \mathbb{P}^2 \rightarrow \mathbb{P}^2$  is a projectivity if and only if there exists a non singular  $3 \times 3$  matrix such that for any point in  $\mathbb{P}^2$  represented by a vector  $x$  it is true that  $h(x) = Hx$*

A projective transformation is on the form of a  $3 \times 3$  matrix determined by 8 coefficients, (there is an ambiguity with one coefficient since it is applied on the projective space).

A change of the orientation and position of a projective camera, is in particular a projective transformation, thus comes the interest for this group of transformations.This is though not valid for a push-broom camera.

This projective transformation are though only valid for a projective camera.

Remark: all the properties are issued from the book [15].

## H. Epipolar Geometry

In order to understand the notion of image rectification, it is important to define some properties of epipolar geometry

The epipolar geometry is the intrinsic projective geometry between two views. It is independent of scene structure, and only depends on the cameras' internal parameters and relative pose.

Suppose that we have a point  $X$  in 3-space that lays on the ground, who is imaged in the two cameras.

Denoting its projection on the first Camera's  $x$ , and its projection on the second camera  $x'$ . we see that these 3 points and the camera center  $C$  and  $C'$ , are co-planar, as it is shown in picture (a) of figure (H.1), we denote this plan ( $\pi$ ) The line joining the camera's center is called the **baseline**, and it intersects each of the image planes on the points  $e$  and  $e'$ , who are called the epipoles, denoted by  $e$  and  $e'$  on the picture (b) of figure (H.1).

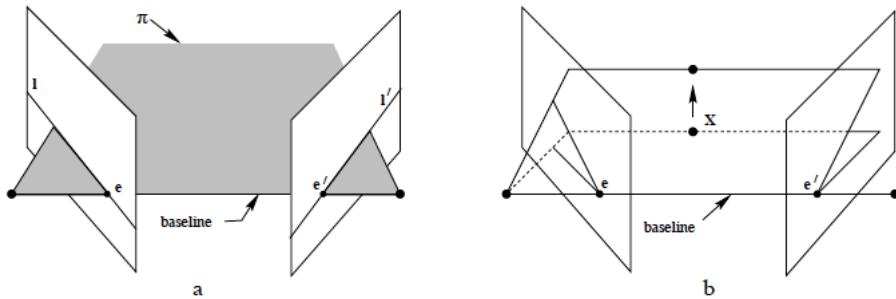


Figure H.1.: epipolar geometry

An epipolar plane is a plane containing, the baseline. The plane ( $\pi$ ) is in particular an epipolar plane.

An epipolar line is the intersection of an epipolar plane with an image plane. All the epipolar lines of one image intersects at their epipole.

Supposing that we only know  $x$ , we may ask how the corresponding point  $x'$  is constrained. The plane ( $\pi$ ) is determined by the baseline and the ray defined by  $x$  as we may see in the figure H.2. From above, we know that the point  $x'$  is contained on the intersection of the plan ( $\pi$ ) and the image containing the point  $x'$ . It is thence an epipolar line.

For each point  $x$  in the one image, we can associate an epipolar line in another (image) plane, and in particular for each epipolar line on an image , we can associate a unique epipolar line, as long as the baseline is fixed, which is the case when the parameters of the camera are fixed.

It is important to notice that the epipolar line encodes the depth of the physical point.

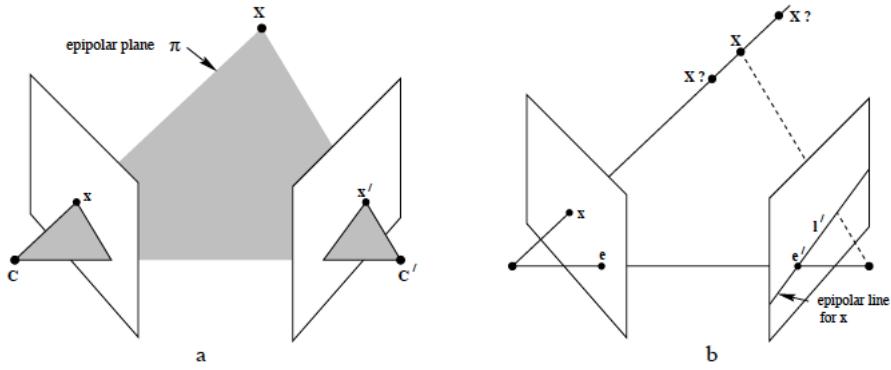


Figure H.2.: point correspondence geometry

Thence, we can restrict the search for the corresponding points of a pixel of an image, on its epipolar line on the corresponding pair.

In the particular case where the epipolar lines are horizontal with respect to the rectangular borders of each image, we can encode the transformation passing from the pixel of an image A to a corresponding image B, on a single image called the "disparity map".

Image rectification is the process of re-sampling pairs of stereo images taken from widely differing viewpoints in order to produce a pair of "matched epipolar projections", which contains in particular the previous particular case, because in that case, the epipolar line associated to a point  $x$ , pass from the corresponding point  $x'$ , whose epipolar line pass through  $x$  and is horizontal, thence, for each horizontal line on an image A, all the correspondences are situated on a horizontal line on image B, and vice-versa.

In order to have a simple representation of the disparity map, we seek for a transformation for which the epipolar are horizontal, and covers the index of line on the two images.

The formalization this transformation, needs the introduction of the concept of fundamental matrix. It is the algebraic representation of epipolar geometry. From figure (H.2), we see that for each point  $x$  on the first image corresponds a line  $l'$  on the second image, on which lies the searched point  $x'$ .

Knowing a set of correspondences  $(x_i, x'_i)$  between the two images, we can construct a 2D homography, that links each point  $x$  to a point  $x'$  on the second image. Denoting this homography  $H_\pi$ , we get that  $x' = H_\pi x$ , as the epipolar line  $l'$  passes from  $x'$  and the epipole  $e'$ , we get that:

$$l' = e' \times$$

Then:

$$x' = [e']x'$$

With the notation: with  $[a] = \begin{pmatrix} 0, -a_3, a_2 \\ a_3, 0, -a_1 \\ -a_2, a_1, 0 \end{pmatrix}$  for an element  $a = (a_1, a_2, a_3)$

Then  $l' = [e'] H_\pi x$

The element  $F = [e'] H_\pi$  is called the fundamental matrix associated to a pair of image.  
Since  $x'$  lies on  $l'$ , we also have that  $x'^T \cdot l' = 0$ , then:

$$x'^T F x = 0$$

This equation , encapsulates the link between two corresponding views on an image.

The process of image rectification is the process of searching two 2D homography:  $H$  and  $H'$  to apply to the images in order to have that  $F_1 = H'^{-T} F H^{-1}$  the fundamental matrix in the new coordinates, verify that: if  $x$  of the form  $(x_0, \lambda, 1)$  (i.e  $x$  on the horizontal line defined by the pixel  $x_0$ ) such that  $x_0$  constant, then the element  $x'$ , is necessarily of the form  $(x_0, \lambda', 1)$ .

## I. Camera models

A camera, is mathematically represented as a  $3 \times 4$  matrix, which associates to every point in the space  $\mathbb{R}^3$ , or the associated projective space  $\mathbb{P}^3$ , a unique point in the associated projective space to the image plane  $\mathbb{P}^2$ .

The pinhole camera model does not include, for example, geometric distortions or blurring of unfocused objects caused by lenses and finite sized aperture. It also does not take into account that most practical cameras have only discrete image coordinates. It is a first order approximation of the mapping of a 3D scene to a 2D image.

In its most basic form, it associates to a point  $(X, Y, Z)$  on the space  $\mathbb{R}^3$  (in this case, it is represented in the frame associated to the camera), the point  $(fX/Z, fY/Z)$ , on the image plane (figure I.1), where  $f$  is the focal length.

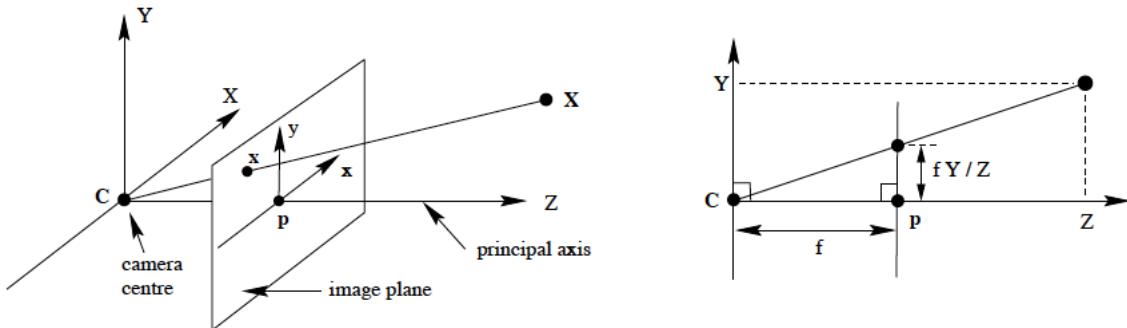


Figure I.1.: Pinhole camera geometry.

Thence to the point  $(X, Y, Z, 1)$  on the space  $\mathbb{P}^3$ , we associate the point  $(fX/Z, fY/Z, 1)$ , on the space  $\mathbb{P}^2$  (the last coordinate 1 is equal to 1, when the two first coordinates corresponds to the position of the pixel on the image plane).

In terms of camera matrix, it could be rewritten as  $\begin{pmatrix} f, 0, 0, 0 \\ 0, f, 0, 0 \\ 0, 0, 1, 0 \end{pmatrix}$

This model assume that the origin of coordinates in the image plane, is the principal point, which is the orthogonal projection of the center of Camera C, on the image plane(figure I.1). A more general form of the pinhole model, is thence:  $\begin{pmatrix} f, 0, p_x, 0 \\ 0, f, p_y, 0 \\ 0, 0, 1, 0 \end{pmatrix} = [K, O]$  where  $(p_x, p_y)$  is the coordinate of the principal point on the image plane, and  $K = \begin{pmatrix} f, 0, p_x \\ 0, f, p_y \\ 0, 0, 1 \end{pmatrix}$  is called the camera calibration matrix.

**UPMC**  
Sorbonne Universités

In the case of CCDs cameras, the pixels are not necessary squared, it implies a more general form for the matrix K:

$$K = \begin{pmatrix} fm_x, 0, p_x m_x \\ 0, fm_y, p_y m_y \\ 0, 0, 1 \end{pmatrix}$$

Generally, points in space may be expressed in other frames, as the frame linked to the planet Mars. In our case, a more appropriate model for the camera matrix, would be of the form:  $K[R, t]$ , where R is a rotation matrix of size  $3 \times 3$ , and t a translation matrix of size  $3 \times 1$

This matrix K is supposed constant for the HiRISE Camera, the values can be obtained from the reference ([11]).

In particular cases, an other parameter "s", called the skew parameter, can be added in the form of K:

$$K = \begin{pmatrix} fm_x, s, p_x m_x \\ 0, fm_y, p_y m_y \\ 0, 0, 1 \end{pmatrix}$$

(this value is different from 0 when the x-axis and y-axis of the CCD array are considered to be not perpendicular,in this case).

A camera of this form, is called a finite projective camera, it has 11 parameters, and every camera matrix has the left hand  $3 \times 3$  if and only if it is a finite projective camera ([15]).

A general projective camera has the form  $P=[M, P_4]$ , where M is a general matrix of size  $3 \times 3$ , and  $P_4$  is a general matrix of size  $3 \times 1$ .

A particular case, of the general projective camera is the case of an affine camera, it is a camera for which the last row is of the form  $(0,0,0,1)$ .

This form is a limit form obtained when, we magnify the zoom of a finite projective camera (i.e  $fm_x$  and  $fm_y$  are sent to infinity),and when we increase the distance between the camera center and the work coordinate frame.

It is in particular a convenient model for a high-resolution camera, loaded on a spacecraft, and imaging a planet,it is not convenient for a push-broom camera, though, but rigorously valid for each line of the produced image, and shown to be experimentally valid for small tiles of a push-broom camera ([9])

The general form of the camera matrix for an affine Camera, is  $P = \begin{pmatrix} fm_x, s, p_x m_x \\ 0, fm_y, p_y m_y \\ 0, 0, 1 \end{pmatrix} \begin{pmatrix} r^1, t_1 \\ r^2, t_2 \\ (0,0,0), 1 \end{pmatrix}$

where  $r^1$  and  $r^2$  are of size  $3 \times 1$  (equation 6.25 of [15])and are the two first rows of the matrix rotation between the coordinate world frame, and the optical frame of HiRISE.  $t_1$  and  $t_2$  are the two first coordinates of the translation between the two frames.

In the case of this model, the vector  $t^3$  have no effect on the produced image, but the element

$r^3$  is uniquely defined by the equations ( $r^3.r^1 = 0$  and  $r^3.r^2 = 0$ ).

Considering the assumption that the camera calibration matrix is well-calibrated for HiRISE, it is sufficient for the bundle adjustment to compute finer values for the rotation matrix and the translation between the optical axis' frame and the world coordinate frame, for each line, and more generally in the case of the "RPC model", it is sufficient, to compute finer values for the coefficients of the polynomials.

It is important to notice , that the pinhole camera model, incorporates every finite projective camera, and that the property of conjugation of the epipolar curves is in particular still valid for the affine camera model H.

## J. DEM and standard geographical vertical references

After the triangulation step, we obtain a 3D cloud point from the stereo pair of images. This 3D cloud point, can be represented, only after we choose a Datum. It is not convenient to represent this cloud point, on the coordinates of the frame linked to Mars. A most convenient representation would be to show, to represent any point, by its latitude , longitude, and its height from the surface.

The longitude is the angle between the perpendicular to the surface, and the meridian plan, which is an arbitrary fixed plan on Mars that passes through its center. The ambiguity in this definition, is that the perpendicular is not defined. On a simple spherical model of the planet, it is the line that passes from the center of the earth.On an ellipsoid model of the planet, it's the perpendicular to the model.

The model (or geodesic system) that allows a rigorous representation of the points in the form of a latitude longitude and height, is called a Datum. The model include sometimes some approximations.

It is the case of the "areoid",the equipotential surface of Mars (which is not necessarily equal to the topographical surface of Mars).In this case, the total uncertainty is about 3m.It is also in this areoid, that the DEM of MOLA is represented.

The informations about the Datum are efficiently encapsulated on the RPC data (1.13), as this function associates to each pixel on the image, and a particular height, the value of the latitude and longitude.

The RPC data for the case of the Camera of HiRISE are obtained from, a tool of the ASP software: "cam2rpc", that produces for a specific height range and range of latitude and longitude, the associated polynomial coefficients of the RPC model.It is computed, with the Datum "D\_MARS", which is a spherical datum , with radius equal to (3396190 meters).In order to register it with the MOLA DEM, and Mola points, the datum is converted to the areoid, another on-shelf tool in the ASP software is used: "dem\_geoid".

The DEM is the final product produced by s2p, it is obtained by a transformation that associated to every point in the geodetic coordinates, a location on a "map". The most simple transformation, used for all the DEM, is the equi-rectangular projection map, that associates to each latitude  $\lambda$  and longitude  $\phi$ , the points:

$$x = (\lambda - \lambda_0) \cos(\phi_1)$$

$$y = (\phi - \phi_0)$$

Where  $(\phi_0 , \lambda_0)$  represents the center of the map, and  $\phi_1$  corresponds to the standard parallel, which is the parallel for which the scale of the projection is true. This transformation distort the distance between the points: points situated at the same latitude  $\phi$ , would be closer the more we

go farther from the latitude  $\phi_1$ .

Generally,  $\phi_0$  and  $\phi_1$  are equal, and they are different when we want to emphasize a particular region of the planet, without changing the center of the map.