LIÈGE université Library

http://lib.uliege.ac.be http://matheo.uliege.be

Master thesis : Study and implementation of a High-Dynamic Range (HDR) imaging algorithm for a space telescope

Auteur : Gava, Sébastien
Promoteur(s) : Van Droogenbroeck, Marc; Denis, Francois
Faculté : Faculté des Sciences appliquées
Diplôme : Master en ingénieur civil électricien, à finalité spécialisée en "electrical engineering"
Année académique : 2017-2018
URI/URL : http://hdl.handle.net/2268.2/4511

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.

Faculty of Applied Science
Department of Electrical engineering and computer science

Master thesis

Study and implementation of a High-Dynamic Range (HDR) imaging algorithm for a space telescope

Sébastien **Gava**

Master thesis submitted in partial fulfillment of the requirements for the Degree of Master of Science in Electrical Engineering.

Supervisor: Marc **Van Droogenbroeck**
Internship supervisor: François **Denis** at Centre spatial de Liège

Centre spatial de Liège

Liège, academic year 2017-2018

Contents

1 Introduction **4**

2 Background **5**
2.1 Coronagraph . . . 5
2.1.1 PROBA-3 . . . 6
2.2 High Dynamic Range imaging . . . 8
2.2.1 Dynamic range . . . 8
2.2.2 Exposure time: . . . 8
2.2.3 HDR imaging . . . 9
2.2.4 Misaligment and ghost effect . . . 10
2.2.5 Dynamic range and human eye . . . 11
2.2.6 Tone mapping . . . 11
2.3 Objectives . . . 11

3 Data **13**
3.1 Images . . . 13
3.2 Occulter . . . 15
3.3 Exposure time . . . 17
3.4 Other data set . . . 19
3.5 Conclusion . . . 19

4 Occulter detection **20**
4.1 Circle Hough transform . . . 20
4.1.1 Tests . . . 21
4.1.2 Performances . . . 22
4.2 Daugman circular integro-differential operator . . . 23
4.2.1 Optimisation . . . 23
4.2.2 Implementation . . . 24
4.2.3 Tests . . . 24
4.2.4 Performances . . . 26
4.2.5 Verification on real images . . . 26
4.2.6 Optimisation . . . 29
4.3 Conclusion . . . 29

5 Image alignment **31**
5.1 Review of image registration/alignment techniques . . . 31
5.1.1 Feature based algorithms . . . 32
5.1.2 Intensity based algorithms . . . 33
5.2 Development of an alignment technique . . . 35
5.2.1 Contours . . . 35
5.2.2 Alignment of contours of images with same exposure time . . . 39
5.2.3 Alignment of contours of images with different exposure time . . . 45
5.2.4 Conclusion . . . 54
5.3 Conclusion on image alignment . . . 54

6 Image composition **55**
6.1 Image calibration 55
6.2 Pixels weighting 57
6.3 Conclusion 62

7 Conclusion **64**
7.1 Complete algorithm 64
7.2 Alignment 66
7.3 occulter detection 67

8 Annex **68**
8.1 Centre spatial de Liège 68
8.2 Structure 68
8.3 Internship 69

Chapter 1

Introduction

The sun's corona exhibits a very high brightness gradient. The use of HDR (High dynamic range) imaging technique is an asset at improving the quality of the resulting image. By taking successive images with different exposure times, it is possible to enhance the contrast of the image. Short exposure time images being adapted for very bright parts of the corona but inadequate for dark parts. Whereas long exposure time images are adapted for dark parts but inadequate for bright parts where saturation may occur. Combining images of different exposure time allows to take advantage of both and is at the basis of HDR techniques.

The combination of images requires a perfect alignment of the sun's coronal structures. Images of different exposure time are taken at successive instants which causes misaligment due to the relative motion of the objects in the scene. The alignment is difficult due to images having a different content (different exposure time) in addition of a translation due to the motion of the scene between successive shots.

This work is a study of existing techniques and development of an image alignment technique as well as an HDR image composition technique for a spatial coronagraph. It is applied in the case of the **PROBA-3** satellite project: a new type of external coronagraph. But the method will stay generic to avoid case specificity. Rather than giving one method to achieve the goal, it is a study of the different possibilities and will serve as a guide of possible techniques for further implementation.

It is part of the internship made at the *Centre spatial de Liège (CSL)*.

Chapter 2

Background

Space weather is a branch of space physics and aeronomy concerned with the time varying conditions within the Solar System [27]. Including the study of solar activity. The forecast of events such as coronal mass ejections, solar flares, geomagnetic storms can be done through solar observation. Large coronal mass ejections can have dramatic impact on the electrical grid on earth or other terrestrial infrastructures . The study of the corona (the outer part of the sun) is of great interest for astrophysicists.

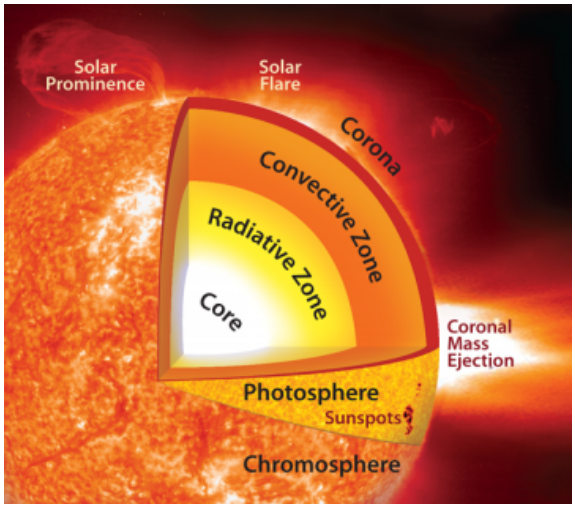


Figure 2.1: Structure of the sun: the outer part is the corona where coronal mass ejections, solar flares, solar prominences are observable

2.1 Coronagraph

The observation of the sun can be performed with the use of a telescope and a camera sensor. But the very high brightness of the central part of the sun makes it impossible to capture details in the corona close to the sun. The solar disc is approximately a million times brighter than the inner part of the corona. The glare of the sun makes it thus impossible to observe it correctly. It is such that *the innermost part of the corona is observable with a sufficiently high resolution only during total solar eclipses* [3]. Natural total solar eclipses observed being a rare phenomenon.

*A **coronagraph** (coronogpraphe in french) is a telescopic attachment designed to block out the direct light from a star so that nearby objects – which otherwise would be hidden in the star's bright glare – can be resolved* [20]. It produces an artificial solar eclipse which, otherwise, is a very rare phenomenon. The coronagraph discussed in this work is only for the observation of the corona of the sun. Coronagraphs can also be used for exoplanets detection for instance by covering bright

stars. While the means of achieving the results can differ, the goal is similar: cover the bright source of light (star) to allow weaker surrounding sources of lights (exoplanets, corona, ...) to be visible. There exists multiple technologies for blocking the direct light from the sun: Band-limited coronagraphs, use of Lyot stops, phase-mask coronagraphs, Optical vortex coronagraph (for stars), ... Optical systems can be very complex but a deeper understanding of these technologies is not necessary as a new type of coronagraph will be studied (see section 2.1.1).

Figure 2.2: Simplistic representation of a coronagraph. The occulter (green) blocks out the direct light from the sun to create an artificial solar eclipse. It allows other parts of the corona to be visible. Copyright ESA–D. Galano

Coronagraphs can be ground-based or in outer space. Cornonagraphs in outer space are placed on satellites in orbit around the earth. They present advantages over the same instruments if they would be if located on the ground. *This is because the complete absence of atmospheric scattering eliminates the largest source of glare present in a terrestrial coronagraph* [20].

But even outer space coronagraphs have a limited functioning due to the technology used for blocking the light. Occulter inside the optics or external but attached on the same instrument have a limited functionning. The problem is that stray light bends around the edge of the occulting disc. *It sets a limit of how close to the solar disc, the corona can be observed. SOHO's coronagraph, an outer-space intrument, can observe no closer in than 1.1 Sun-diameters* [7].

2.1.1 PROBA-3

PROBA-3 (Project for On-Board Autonomy) is a satellite mission. *It will be composed of two independent, three-axis stabilized spacecraft flying at 150 meters to one another with the ability to accurately control the attitude and separation of the two craft. It will be maintained for 6 hours, creating "artificial solar eclipse" for the satellite below* [25]. *The mission will demonstrate formation flying in the context of a large-scale science experiment. The paired satellites will together form a 150-m long solar coronagraph to study the Sun's faint corona* (proba 3).

Replacing an occulter in the optic of the telescope with an external occulter will allow better observations of the corona by overcoming current limitations. But the challenge of aligning two satellites to cover the sun has never been done before as this is a world's first attempt.

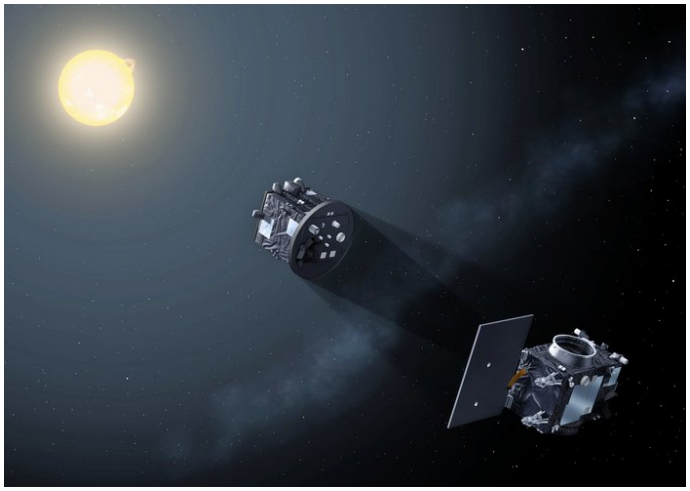


Figure 2.3: Artistic representation of PROBA-3, the circular spacecraft covers the disk of the sun. The distance between the two satellites is not to scale, actual distance is 150m. Source: ESA

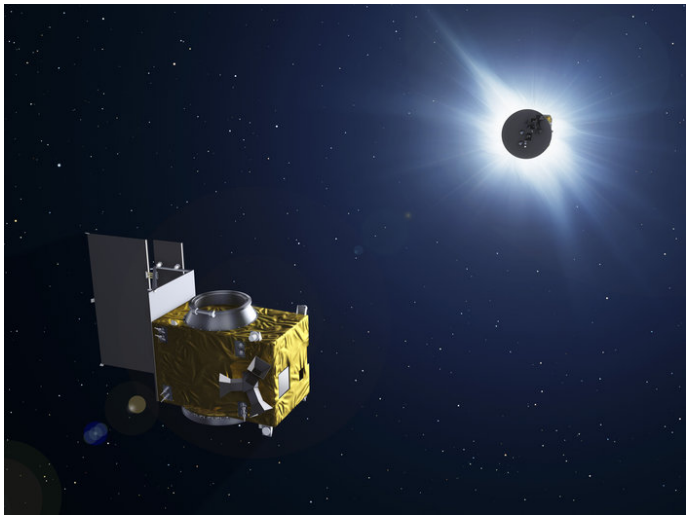


Figure 2.4: View from another angle of the artificial total solar eclipse created. Copyright ESA - P. Carril, 2013

The alignment between the two telescopes is performed through a system of light sensors placed on the spacecraft which carries the camera. A feedback signal is sent to the other spacecraft to perform the alignment.

The spacecraft pair will fly a highly elliptical orbit divided between periods of accurate formation flying around apogee. It will be maintained for 6 hours, creating 'artificial solar eclipse' for the satellite below. [25]

	Proba-3 facts
Launch date:	Late 2020
Mass:	Coronagraph spacecraft 340 kg; Occulter spacecraft 200 kg
Orbit:	High Earth orbit, 19.7 hours orbital period, 60 530 km apogee, 600 km perigee
Instrument:	External coronagraph
Ground station:	Redu, Belgium

Source: [25]

2.2 High Dynamic Range imaging

HDR is an acronym widely used these days and a commercial attribute in many consumer electronics devices (television sets, smartphone screens, cameras, etc). But it can refer to different things. HDR in this report refers to HDR imaging (or photography) which is a technique to capture a greater dynamic range of luminance during a photography. Not to be confused with HDR rendering which is related to the quality of the display of an image or video.

2.2.1 Dynamic range

The dynamic range in a general sense is the ratio between the largest and the smallest value of a quantity. In the case of images, the quantity is luminance. It is also named contrast ratio in the literature. Contrast ratio is more often used to describe a display, it is the ratio between the brightest and darkest value of luminance it can render. The higher the dynamic range of a sensor, the better the scene can be captured and stored on a medium. It means more details in the image and more information for a telemetric application.

With a digital camera, the maximal dynamic range of the image is linked to the number of bits per pixel that the sensor is able to capture. The number of bits per pixels defines the quantization levels of a digital image. Hence, $2^{nb\_bits} - 1$ levels of quantization. For instance, a monochromatic camera with 8bits/pixels has a maximal dynamic range of $255 : 1$. The more bits, the more information can be stored in the image. But the dynamic range depends on the quality of the parameters for the shot. An image very bright everywhere will have a very low dynamic range in the same way as a completely dark image. Independently of the image data representation or sensor quality. This is because the smallest and largest values in the image are close to one another thus the ratio close to 1.

Therefore, parameters of the camera have to be chosen adequately in order to obtain the maximal dynamic range. The lowest value that the camera is able to capture has to be as close as possible to the lowest luminance of the scene. And the maximal value ($2^{nb\_bits} - 1$ for a digital sensor) has to correspond to the highest luminance of the scene. If this condition is met then the dynamic range of the image is fully used to capture the scene (Fig 2.5).

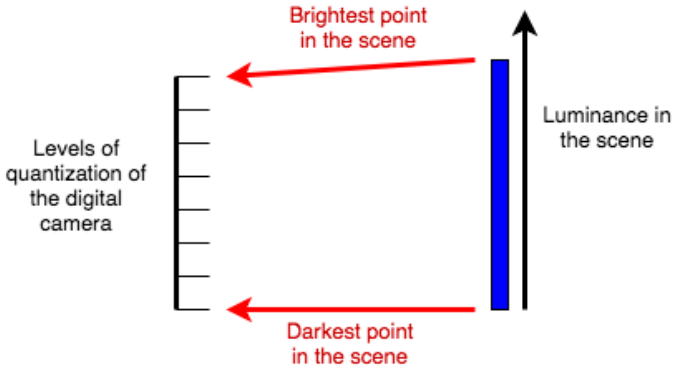


Figure 2.5: Optimal condition for image capture of a scene with a single shot digital camera

There exists multiple parameters in photography to allow this condition to be (partially) full filled: exposure time, ISO and aperture. But in the case of HDR imaging, only exposure variations are performed. Modifying the aperture also effects the depth of field which does not allow combination of multiple images as their content would be altered. [22]

2.2.2 Exposure time:

*The **exposure time** - also named shutter speed - is the length of time when the digital sensor inside the camera is exposed to light. The amount of light that reaches the film or image sensor is proportional to the exposure time* [26].

A long exposure time image will have an overall greater luminance. It will detect smaller amount of light sources even in darker areas. But the picture might saturate in bright parts of the scene due to a too large amount of light for the sensor. On the other hand, short exposure time image will be appropriate for bright parts of the scene but will not be able to detect weaker light sources. Exposure time values are often refered to as 'stops'. A stop consists in doubling the exposure time (relative to a reference time). +1 stop is doubling, +2 stops is times 4, and -1 is halving the exposure time.

Figure 2.6: Pictures of the same scene taken with an increasing exposure time. The increase between each scene is of +2 stops i.e 4 times longer. Source: wikipedia *https://en.wikipedia.org/wiki/High-dynamic-range_imaging*

2.2.3 HDR imaging

HDR imaging consists in combining multiple low dynamic range images (LDR) with different exposure time in a single High dynamic range image (HDR). In the case of a LDR, if the condition of maximal dynamic range for the shot is met(Fig 2.5), the dynamic range of the image is still bounded by the sensor capability. For instance a camera with 8 bits/pixel can only capture ratios up to 255:1. The resolution, in terms of contrast, is limited. But it is possible to achieve a greater dynamic range using the same camera by combining multiple shots with different exposure time.

Figure 2.7: HDR image composited from multi-exposure images in Fig 2.6.

Each LDR image will cover a different part of the luminance of the scene. This allows to have a greater resolution on the luminance captured . Images with a short exposure time will be adapted for capturing very bright parts of the scene but will fail to capture darker parts. Long exposure time images being the opposite. They will saturate in bright parts of the image (Fig 2.6) All intermediate images captured with various exposure time will help cover the whole range of luminance. The final composite image will have a greater dynamic range than it is achievable with a single shot by the camera.

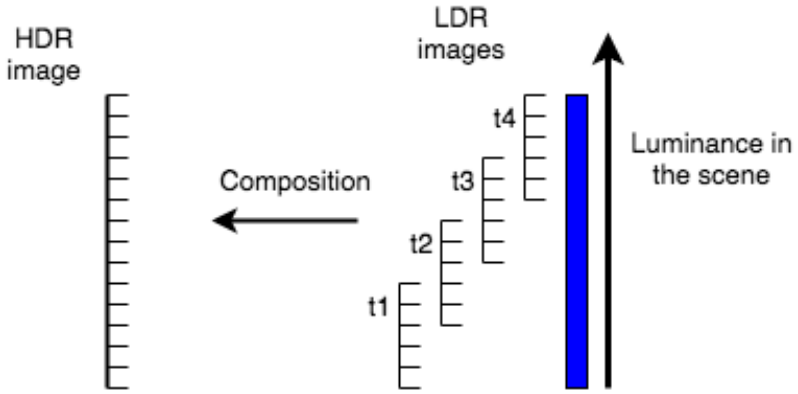


Figure 2.8: Schematic representation of the necessity of multiple images with different exposure time for capturing a scene with a high contrast ratio.

The sun's corona exhibits a large contrast ratio which is difficult to obtain through a single image. *For successful total eclipse photography it is necessary to achieve the brightness ratio up to 1 : 1 000 000* [3]. The use of compostion of images is a great improvement for the quality. And a necessity to capture a brightness ratio up to 1 : 1 000 000. Sensors are not capable of achieving such performances with a single shot.

2.2.4 Misaligment and ghost effect

HDR imaging requires multiple shots to be taken. In most cases, it is done with a single camera. The different shots are thus taken successively in time and not a single instant. The scene can not always be supposed fixed in time between the different shots. Especially if the time of exposures are long for the shots (and time between the shots). The differences can either be: a variation in the luminance in the scene, a movement of the object in the scene or a movement of the camera. This misalignment between shots is critical for HDR imaging as it will create a blur in the composite image if not corrected (Fig 2.9)

But the relative movement in the scene is not always possibly corrected. If the background is fixed in the image but an object is moving fast compared to the time of exposure and duration between shots, it will create a 'ghost effect'. The moving object appears blurred.

Figure 2.9: `Left:` The composed HDR image is blurred due to the motion of the camera during the capture. This is caused by the misalignment between successive shots. `Right:` The movement of the person during the capture of the successive shots causes a 'ghost' to appear in the composite HDR image. Source: *https://iphonephotographyschool.com/hdr-iphone/*

2.2.5 Dynamic range and human eye

The human eye can perceive a limited dynamic range. Since the human eye perception can vary a lot in different conditions as the eye actively adapts itself, it is impossible to provide a single value. Multiple sources giving different answers, no value will be given as it is not essential for the rest. But it is important to know that our perception of contrast ratio is limited.

2.2.6 Tone mapping

The composite image is stored in a medium with a greater amount of bits/pixels than that of a single LDR image. It is necessary in order to achieve a greater dynamic range. But the human eye is not always able to see such high contrast ratios. The display is also not always able to render such contrasts as it has a lower dynamic range. The process of tone mapping consists in reducing the dynamic range of the HDR image into an image understandable by the human eye or displayable by a display while keeping most details in all parts. (Fig 2.7 is actually a tone mapped image in order to be displayed correctly)

This process can be highly non linear depending on the result expected. Some artistic photography will create surrealistic effects using this technique but it will alter the data. For a telemetric application, this is not the result expected as tone mapping will try to reproduce the original scene as accurately as possible.

The image after tone mapping contains less information than the composite image due to the reduction of information. But it displays the image in a convenient way for seeing details in both dark and bright parts according to the human perception of contrasts.

2.3 Objectives

The idea of this project is to study the possibilities to design an HDR algorithm targeted for an embedded device in the context of the **PROBA-3** project. The composition of images on board of the satellite instead of a ground station presents several advantages. The process can be automated instead of commanding the satellite for different images manually for multiple exposure time images. It is interesting from a data transfer point of view. The amount of data to transfer a composite image is less than for multiple images sent successively (see chapter 6) . With the limited power of a satellite and the distance, the bit rate transfer is relatively small. An on-board image processing would be an advantage over a ground based processing for allowing more images taken in the same amount of time.

But making this procedure on an embedded device on a satellite is challenging. Electronics for spacecrafts face a lot of constraints. The reliability of the electronic systems must be much higher than any consumer electronics devices. Obviously, a repair is not an option. Electronics undergo many extreme environmental conditions. Temperatures can vary from very low to very high values. Convection does not occur in empty space which causes self-heating in the circuit. The acceleration at launch can reach enormous values. Circuits withstanding such accelerations have to be designed specifically. Radiations in outer space are way higher than on earth. It can cause corruption of the data as particles (heavy ions) can cause a change in a logical value inside the electronics (a 1 becomes a 0 or inversely). Risk mitigation techniques in general like redundancy, shields, spacing between hardware cells, etc. Causes spatial electronics to evolve at a slower innovation rate.

All constraints previously mentioned lead to the conclusion that computational power is very limited in spatial electronics. The chips that are certified for spatial electronics are relatively 'old technology' compared to what is found on terrestrial applications.

The objective is therefore to make algorithms that are suitable for an embedded device. Keeping in mind the limited computational power and limited memory is important for the choices in this work.

The problem is twofolds: The relative movement of the satellites causes misalignment between the successive images and an image registration technique has to be developed. Then a study and implementation of a composition technique for successive images into a single HDR composite image is performed. The former being the most preponderant part of the work. This work can be divided into these two parts independently.

Chapter 3

Data

One of the major difficulty of this work is that no data is yet available for a spatial coronagraph based on a pair of satellites. As **PROBA-3** is a future project and that no similar project is existing at the moment. Therefore, in order to test the algorithms it is necessary to have data that is representative of the conditions of **PROBA-3** and in a sufficient amount (for performance evaluation). A thorough research of available data that could be used as a test set needed to be performed.

Norikura Observatory [14] has ground-based coronagraphs but image quality and inconsistency makes the use of such images impossible. Plus ground-based coronagraphs do not reproduce the conditions expected with a pair of satellites (movements). There are multiple space coronagraphs but they do not represent a realistic setup either. The occulter is fixed at the center of the image as it is not a moving part. LASCO (Large Angle and Spectrometric Coronagraph), a spatial coronagraph, contains three instruments on board: C1, C2, and C3. C2 and C3 are coronagraphs for large radii respectively 1.5 to 6 solar radii and 3.7 to 30 solar radii[23] far from what this works is intended to. C1 is for smaller radii values but still, data is unavailable.

Data from photographs of natural eclipses (from a lunar eclipse, not a coronagraph) are too scarce to have a sufficient data set for performance analysis . The behavior is also not the same as a spatial observation. In addition of having really few data, there is no ground truth values available for alignment or occulter detection. However images from natural total solar eclipses are the closest samples to what is expected as data and can be used as models. The moon (which acts as the occulter) is moving between successive shots which corresponds better to our case.

In addition to that, finding images of coronagraphs with multiple exposure values usable for an HDR algorithm has been unsuccessful. Except for rare image with a different exposure time in LASCO images but inconsistent for a real analysis.

It is thus necessary to generate data for the occulter and for the different exposure time. Data generation is necessary for this study but the major drawback is the reliability of the results. The assumptions made for the generation of data must be taken critically as there is inevitably differences with real images. But the idea is to make choices that will keep the algorithm general and have as little influence on the end results. Therefore it was decided to use images from spatial solar telescopes. The eclipse is virtually created on the image afterwards (see section 3.2).

3.1 Images

There are multiple sources for spatial telescope observing the sun. Not all of them have data publicly available. Or for some of them, images are already processed (aligned, enhanced, ...) or with more sophisticated image processing which makes the data unusable. Finally, the satellite **PROBA-2** was selected. It has on board (among other instruments) the **SWAP**(Sun Watcher using Active Pixel System Detector and Image Processing) instrument which is monitoring the sun

since 2009. It was developed at the Centre spatial de Liège. It is also the previous version of the future **PROBA-3**. This instrument observes the sun in Extreme UV.

There are several advantages of using this particular data. A large database is available online for the **SWAP** telescope (around a million of images !)[16]. The advantage of this database is that raw data is available publicly while many others only have processed data. It allows more flexibility for the tests and it is also more realistic as this work is intended to work on raw data on board of the satellite.

Multiple data files are available for one image capture. The 'level 0' files consist in raw images from the sensor received from the satellite. The 'level 1' are calibrated images They are post-processed: bad pixels corrections, dark currents subtraction, aligned and rotated to have the north of the sun up, etc. This processing is performed in a ground observatory by the *Observatoire Royal de Belgique*. This post-processing is not necessary for an HDR algorithm (except the alignment of images). Plus it requires computational power which is more suited for ground stations than an on board processor with limited power.

Data is available in the FITS (Flexible Image Transport System) format. This format contains the image as well as metadata in a header file which are valuable. It contains a lot of information which allows different testings setups.

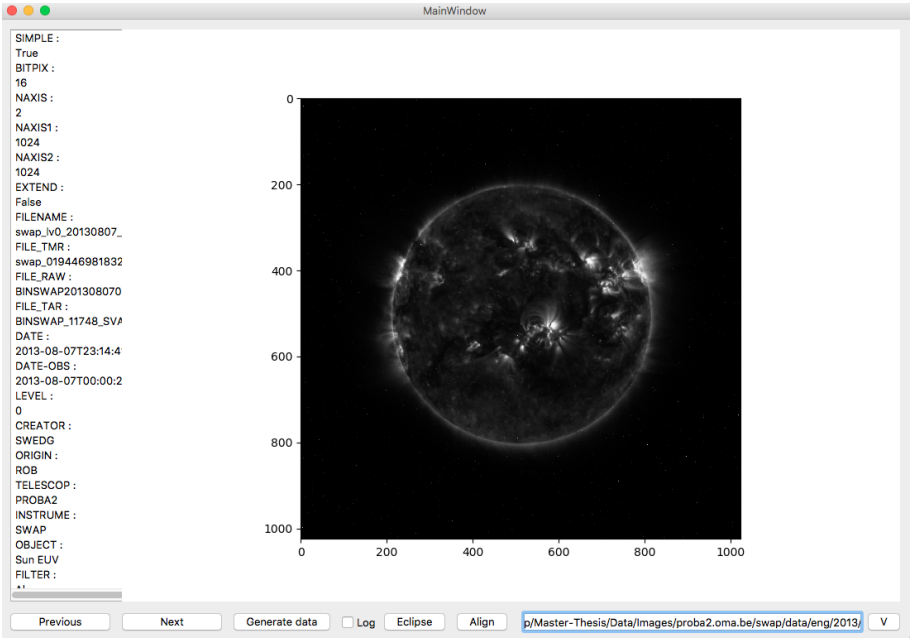


Figure 3.1: GUI developed in Python to visualise multiple FITS files efficiently and to handle the header file's data for generation and testing. Image of the sun that will be used as test data.

Multiple informations are available such as the date and time of the shot, the size of the image,... But most importantly the 'level 1' files (that are aligned images) contain information about the sun's location in the raw data 'level 0' files. This information is crucial to test and evaluate the image registration technique detailed in chapter 5. Therefore it is possible to have the raw unaligned image with the information of the sun's location in x and y coordinates.

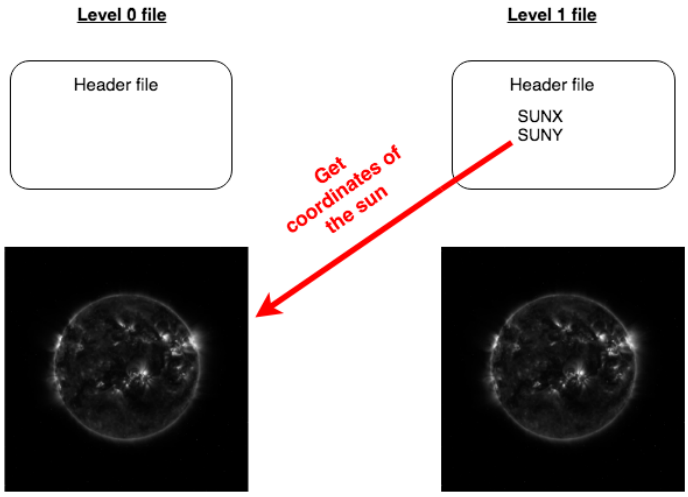


Figure 3.2: Level 0 files are unaligned, unprocessed images. Level 1 files are aligned and processed images. Level 0 files will serve as test images with the information of the correct translation to apply, retrieved from Level 1 files.

From this information, the translation between one image to the next can be retrieved easily.

$$\begin{aligned} t_x &= SUNX_1 - SUNX_2 \\ t_y &= SUNY_1 - SUNY_2 \end{aligned} \tag{3.1}$$

Where t stands for translation and $SUNX_i$ for the center of the sun in image i. This translation data will be used later as ground truth for assessing the image alignment method in chapter 5.

Images coming from this source have the following characteristics:

Image type	Grayscale image
Size	1024×1024 pixels
Number of bits per data pixel	16
Minimal value of a pixel	10
Maximal value of a pixel	3588
Effective number of bits per data pixel	12
Storage size	2,1 Mo

It is important to note that all algorithms discussed here are for grayscale images. This is the case for all spatial telescopes which work with monochromatic data. Although a conversion of RGB data to a grayscale image when working with other types of images can be performed easily before.

3.2 Occulter

Images from PROBA-2, SWAP are full images of the sun in the EUV (Extreme Ultra Violets). In order to simulate data coming from a coronagraph, an occulter has to be added on the image. As previously explained, the occulter in this case is a moving satellite. It has a circular shape and is flying at a distance of 150m from the other satellite containing the camera. It will cover the sun with a radius very close to that of the sun. Therefore the occulter is assumed to be in a range of radii limited. The relative distance between the camera and the occulter is not assumed to be perfectly controlled. So to stay as generic as possible, the radius of the occulter (in the image, the physical radius is fixed) is not considered constant over time. A scaling effect is possible with the relative motion.

To simulate a realistic total solar eclipse, it is necessary to cover correctly the sun. The information about the center of the sun discussed previously can be used for this purpose. The center of the occulter is defined as the center of the sun plus a random variable along x and y axis in order to simulate the motion. As the behavior is not known a priori, the random variable is a uniform distribution with a maximal value (to have realistic images). This has no real impact on the rest of the tests and stays generic. The motion is supposed to be unpredictable, hence the random variable. No speed or direction of displacement is taken into account between shots as that would be a big assumption.

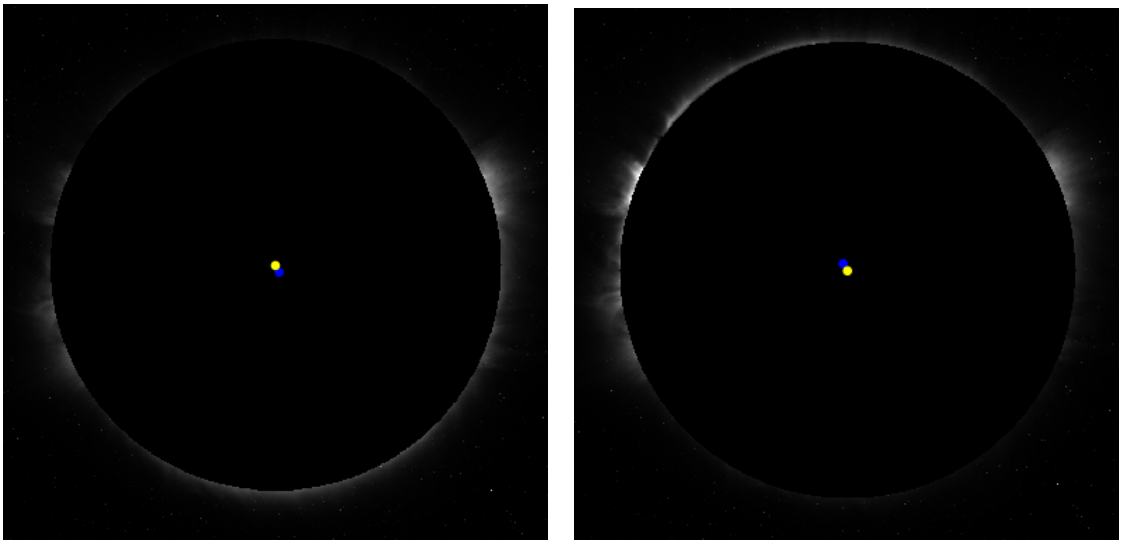


Figure 3.3: Two **identical** solar images where the occulter has been generated differently. The blue dot represents the center of the sun, the yellow dot represents the center of the occulter. The two centers do not correspond

Using a virtual occulter to make the tests is necessary, the few images coming from actual solar eclipses are too rare and do not offer the possibility of having a ground truth data for evaluation later on. But it shows its limitations, of course using an image of the sun and covering it after is not equivalent to a real solar eclipse. Optical phenomenons such as light bending or local glares could cause various differences between simulated data and real data. But a quick comparison with real solar eclipse shows that it is not too far from reality.

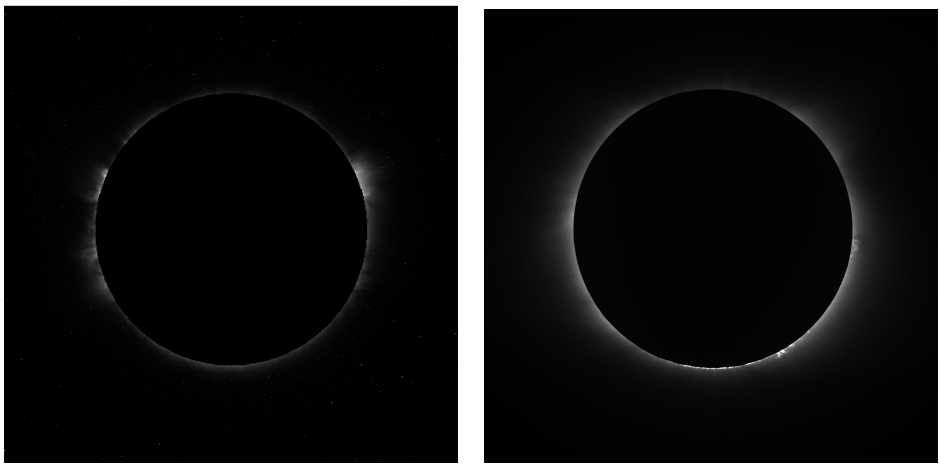


Figure 3.4: `Left:` Virtual eclipse created with the addition of an occulter `Right:` Natural solar eclipse captured on earth with a DSLR. ajouter source

3.3 Exposure time

No consistent data of multi exposure images for a coronagraph nor solar observation have been found. Or at least with a sufficient amount to consider an HDR algorihm or alignment performance evaluation. The database from **PROBA-2** used in the previous section are all at the same exposure time (10 seconds). It is thus also necessary to simulate the different exposure time virtually.

To simulate multiple exposure data, the idea is to make the opposite of an HDR algorithm. The goal is to separate the original image -which is considered to have a high dynamic range in this case - into multiple lower dynamic range images which will simulate the different exposure time.

Starting from images described in the previous section, table from 3.1 indicates grayscale levels ranging from 10 to 3588. Thus the image is encoded on $\lceil log_2(3588 - 10) \rceil = 11.8 \simeq 12$ bits/pixels. To create LDR images, certain parts of the 12 bits representing the data are selected. Most significant bits (MSB) represent bright parts of the corona, they will be used to create images of low exposure time. On the other hand, Least significant bits (LSB) represent finer details and weaker sources of light. They will be used to represent high exposure time images. Basically all intermediate bits selected will be for intermediate times of exposure.

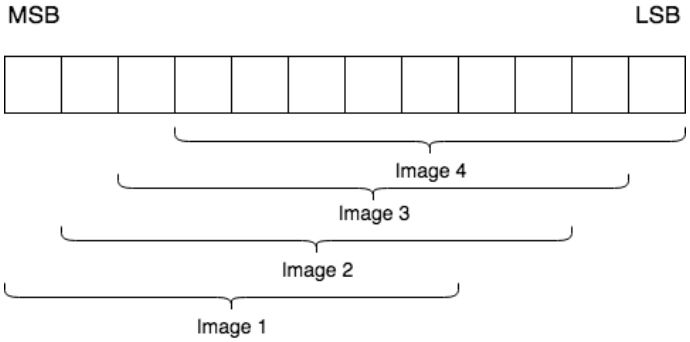


Figure 3.5: 12 Bits of a pixel in the image, used to virtually create images of 9 bits/pixel with different exposure time by selecting bits in the data

The bits were selected by 9 at a time, this results in 4 images with 9 bits/pixels. There is no particular reason to chose 9, it is just to have a convenient number of images and it has not much impact on results. But to construct an image like image 4 (see 3.6), it is important to first make a saturation of pixels that are above the maximal selected bits. Otherwise, least significants bits of bright points will be taken into account which is not realistic and will create irrelevant images. So the generation can be described in two steps:

- Shift to the right the vector of bits
- If value $> 2^9 - 1$ then value $= 2^9 - 1$ to saturate pixels of higher intensity

Shifting by one bit in the data is equivalent to a division by 2. Except the division is not on floating numbers but integers thus losing data. This method is more realistic than simply make a division by 2 with floating numbers because images with different exposure are expected to have a different content. Not simply be multiples of each others which otherwise would not recreate the difficulties of working with different exposure time images. This way it is possible to create data (see Fig 3.6).

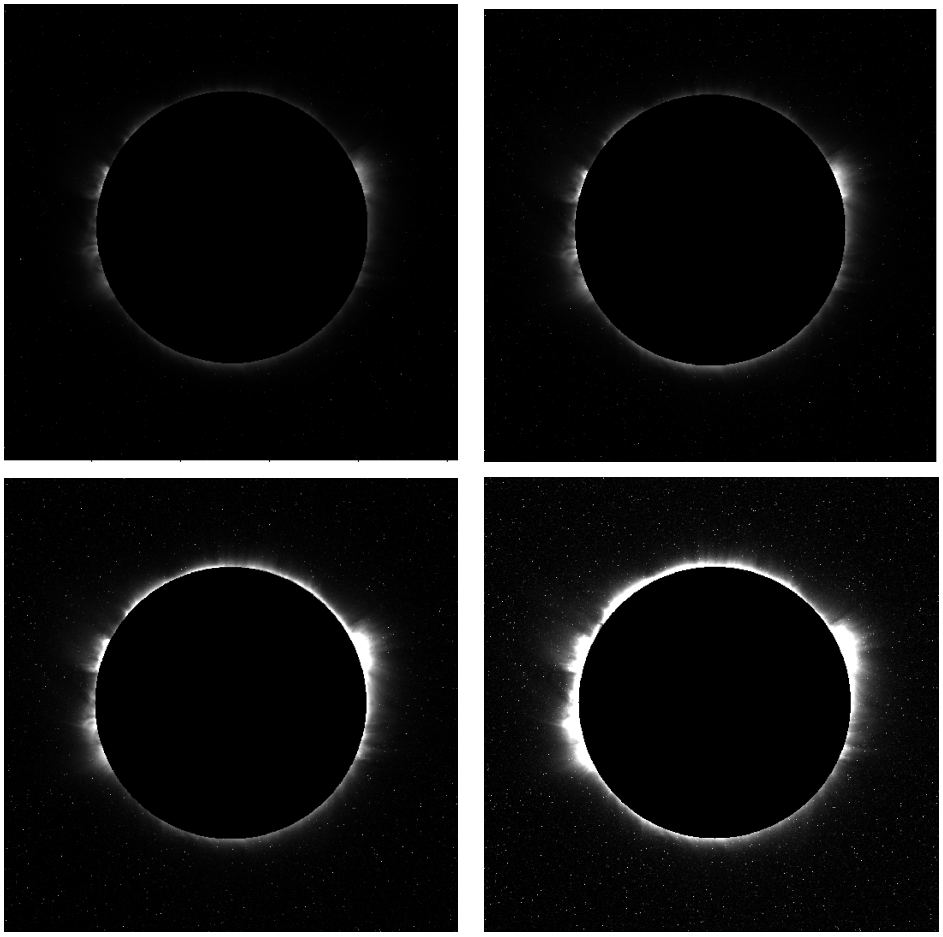


Figure 3.6: Images of 9 bits/pixel from low to high exposure created with the method described in Fig 3.5.

Image 4, which takes the least significant bits of the data, picks also more noise - noise is mostly present in the least significant bits. It is not the case for images like image 1 where noise is much smaller. This is however not very realistic, as noise should be constant on average over all images taken from an identical camera. But it should not have a great impact on the results.

Image 4 has a lot of saturation on bright points, which corresponds to the conditions of a long exposure image.

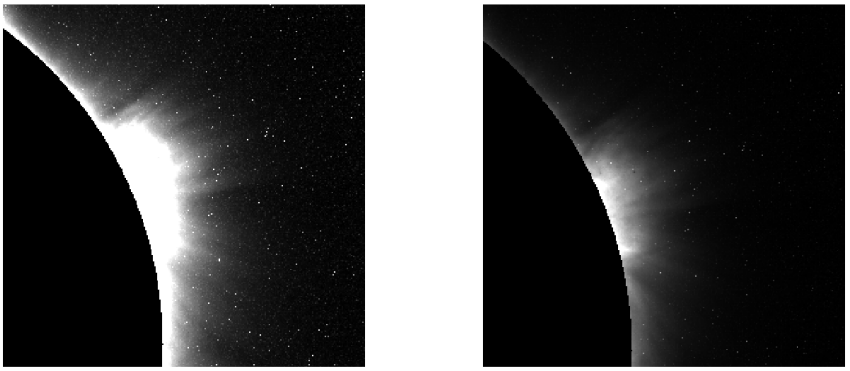


Figure 3.7: `Letf:` Image 4 is saturated in bright parts but more noise is visible in the background `Right:` Image 2 is not saturated in those regions

3.4 Other data set

One data set of multi-exposure image from a natural total solar eclipse is available (see Fig 3.8)[12]. It will serve as a guide for verifying the assumptions made for the data of virtual images. This data can not be used effectively for the tests of image alignment as the author claims to have aligned them manually but no ground truth data is available. Thus it is not sufficiently reliable for testing as there can be some errors. No information is given other than the time of exposure, which makes this data insufficient for a true performance analysis.

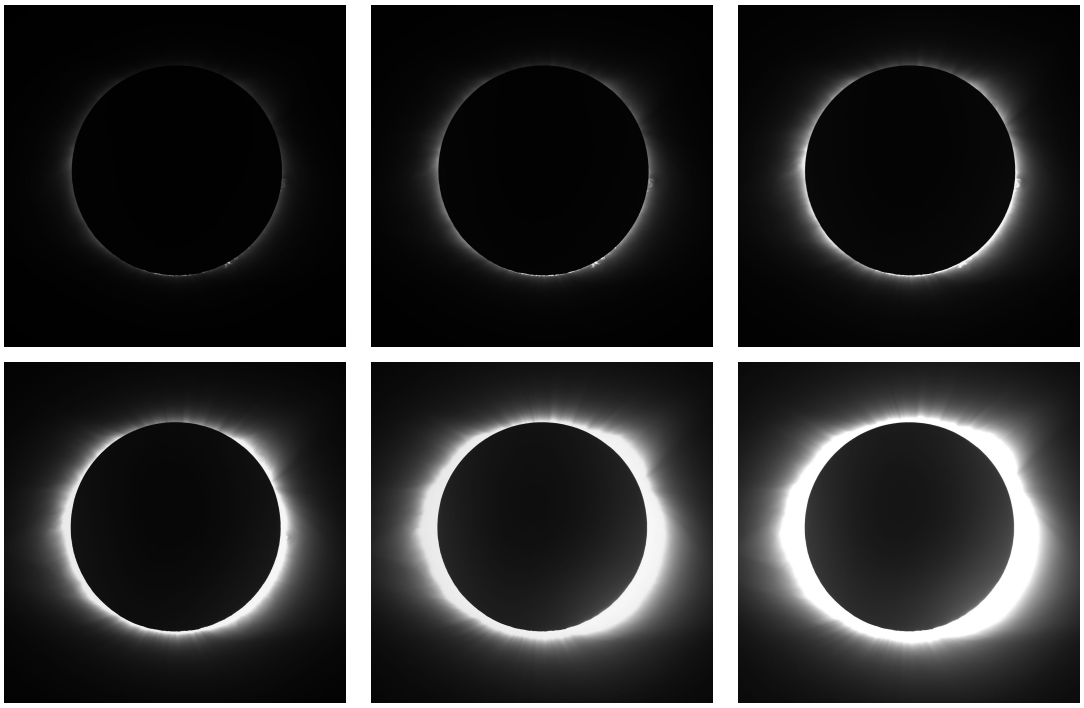


Figure 3.8: Images from low to high exposure of a natural total solar eclipse (moon) observed on earth with a DSLR. Time of exposure are respectively 16ms, 32ms, 64ms, 128ms, 256ms, 512ms

One can see the similarities between this data set and the simulated data in Fig 3.6

3.5 Conclusion

Images have been generated and they have the following characteristics: The sun - and its coronal structures - is misaligned between successive shots. An occulting disk is added covering the sun which is translated randomly between successive shots. Successive images have a different exposure time.

Chapter 4

Occulter detection

The detection of the occulter is necessary for both image alignment and image composition. It is necessary to detect correctly the pixels belonging to the occulter as they carry no information and must be discarded in all posterior steps. Therefore it is presented here as a preliminary step. It is important to reject pixels to avoid a ghost effect in the image due to the movement of the occulter.

The occulter is assumed to be circular. The detection can be reduced to determining three parameters: the x and y coordinates of the center of the disk and its radius. The assumption of a perfect circle is quite realistic and it will be shown that the algorithm can still work correctly if it is not too far from a perfect circle (The detected occulter will still be circular (x,y) and r parameters).

4.1 Circle Hough transform

The circle hough transform is a basic technique for detecting circular objects in a digital image. It is a variant of the Hough transform. The idea is to find the best circle candidate through a voting procedure. A circle in a two dimensional space can be described by:

$$(x-a)^2 + (y-b)^2 = r^2 \tag{4.1}$$

The three parameters (a, b, r) are sufficient to fully describe the circle. Or equivalently using parametric equations:

$$\begin{aligned} x &= a + rcos(\theta) \\ y &= b + rsin(\theta) \end{aligned} \tag{4.2}$$

Where theta ranges from 0 to 360 degrees. It is possible to go through all points (x, y) belonging to the circle described by (a, b, r) by varying θ. First a so-called accumulator matrix A is built. It is a 3 dimensional matrix where an element represents a triplet (a, b, r). All values of the accumulator are initialised to 0.

Let us show an example for a circle with a fixed and known radius R. For each point belonging on the circle (left image), a circle of radius R is drawn in the hough space (right image). The points corresponding to the circle in the Hough space receive a vote i.e are incremented by 1. The result is that the point which received most votes in the Hough space (the accumulator) is declared as the center of the circle.

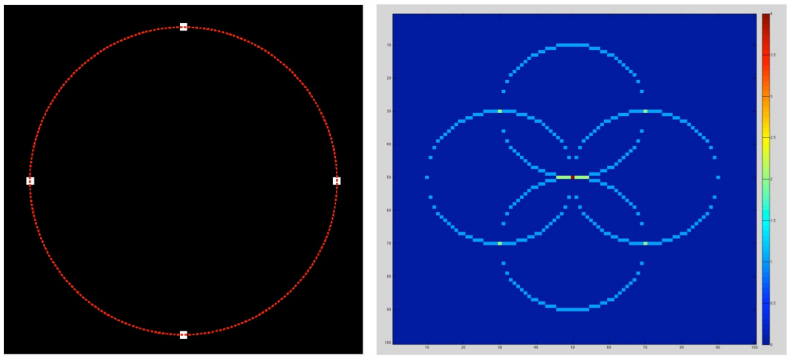


Figure 4.1: Source: [18], `Left:` 4 points belonging to a circle with a known radius for which the center has to be determined `Right:` Hough space where the maximum value corresponds to the center of the circle

For a fixed radius, the accumulator is reduced to a 2D matrix. But for an unknown radius, the accumulator is of 3 dimensions. The additional dimension being for the radius. The Hough transform is performed for each value of R and stored in the accumulator successively. Then the global maximum of the accumulator corresponds to the detected circle. The parameters of this circle (a, b, r) are then simply its coordinates in the accumulator matrix.

But the circle Hough transform works with circles. In the case of the occulter detection, it is a disk in the image. Therefore the common operation preceding the voting procedure is an edge detection. This edge detection can be performed in different ways: Canny edge detector, thresholding and/or morphological operations. Then all points of the edges are considered as candidate points of the circle. The hough transform is performed and the maximum in the accumulator is the detected circle. Of course, as most images contain various object different than circles, the accumulator contains a lot of noise coming from edges of other objects. The difficulty is thus to have a sufficiently detectable circle that corresponds to the maximum of the accumulator.

4.1.1 Tests

The tests were performed using a library in matlab [13]. Some simplifications are made possible because the detection here is limited to a single circle. Therefore the detection is simply the global maximum of the accumulator.

First the edge detection is performed on the input image (Fig. 4.2). Pixels in the corona as well as noise in the image are detected as edges. Then the Hough transform is performed and its peak detected (Fig. 4.3). Points not part of the circle create some noise around the global maximum. But it is still correctly detected within a pixel of precision.

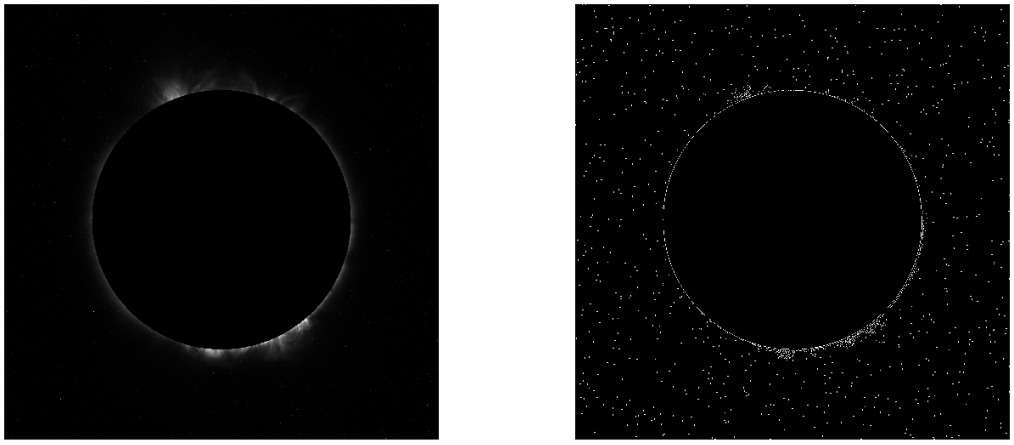


Figure 4.2: `Left:` Input image. `Right:` Edges of the image using Sobel operator by default. The image is fed as input of the hough circle transform

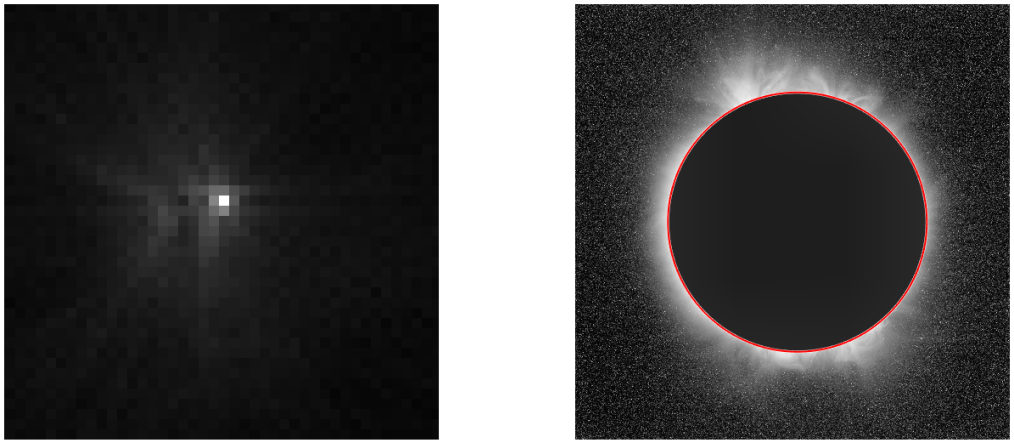


Figure 4.3: `Left:` Zoom on the peak of the accumulator of the circular Hough transform at fixed radius corresponding to the occulter. The peak is located at pixel [524, 513] which corresponds to the center to be detected. `Right:` Detected circle (red) on the image (tonemapped for a better visualisation)

4.1.2 Performances

The circle Hough transform requires a lot of computation. For a square image of size N, the complexity is: N^2 for browsing each pixel and $\frac{N}{2}$ values of radii to be tested. Thus it is of order $\mathcal{O}(N^3)$. Reducing the range of radii to be tested strongly reduces the computation time but it supposes an initial guess which is not always possible or too case dependent. Even with a reduced range of radii, the computational cost is unusable for 1024×1024 images. For a single value of radius, the algorithm takes 132 seconds to exexcute ! This is unusable as such in practice. The complexity is very size dependent but downsizing the image is not possible as the wanted precision is of the order of a pixel.

Reducing the range of possible center candidates is not that easy with the CHT - compared to the method described in the next section. The quality of the detection also relies on the quality of the edge detection. Parameters have to be tuned to correctly make the detection. And it can be difficult for low exposure images where the edge of the occulter is not strong compared to the background.

While improvements in efficiency and reduction of computation time are possible, another method can be investigated.

4.2 Daugman circular integro-differential operator

The problem of circle detection has been solved in a completely different application by John Daugman [2]. In iris recognition, the first step consists in an accurate segmentation of the iris and the pupil . This problem is somehow closely related to the detection of the occulter.

The detection of a circle in the image can be performed using the so-called Daugman integro-differential operator which is defined as:[2]

$$\max_{(r,x_0,y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right| \tag{4.3}$$

where I is the input image. *The operator searches over the image domain* (x, y) *for the maximum in the blurred partial derivative with respect to increasing radius, of the normalized contour integral of* $I(x_0, y_0)$ *along a circular arc* ds *of radius* r *and center coordinates* (x_0, y_0)*. The symbol* $*$ *denotes convolution and* $G_\sigma(r)$ *is a smoothing function such as a Gaussian of scale* σ*. The complete operator behaves as a circular edge detector, blurred at a scale set by, searching iteratively for the maximal contour integral derivative at successively finer scales of analysis through the three parameter space of center coordinates and radius* (x_0, y_0, r) *defining a path of contour integration.* [2]

The operator can be adapted to a discrete set of coordinates as images are represented in a discrete grid of pixels. It is not necessary to make this assumption for the values of intensity . The line integral $\oint_{r,x_0,y_0} ds$ becomes a discrete sum over a circle discretised over the grid of pixels. The derivative over the radius is replaced by a difference operation along radius values. For each center (x_0, y_0), an array of values is computed by the line integral and are stored. Then the difference operation is performed. This operation reveals edges along a radial change. A gaussian operation is performed on this array before taking the maximum.

$$\max_{(r,x_0,y_0)} \left| G_\sigma(r) * \Delta_r \sum_{r,x_0,y_0} \frac{I[x,y]}{2\pi r} \right| \tag{4.4}$$

algorithm outputs the coordinate (x_c, y_c) of the center and the radius r of the circle that fits at best the occulter.

4.2.1 Optimisation

The algorithm is heavy in computation as the raw algorithm requires a computation over every point x_0, y_0 and every radius r possible. Considering an image of 1024 by 1024 pixels, this would mean $1024 * 1024 * 512 = 0.5 * 10^{10}$ computation of discrete line integrals in the image. This is not achievable in reasonable time especially if the target is an embedded device with limited power. Therefore several optimisations have been made on the algorithm.

'Limited range' Instead of taking every pixel as a possible center for the sun, it is possible to limit the range of pixels where the sun's center is expected. Indeed the sun is supposed to be around the center. Therefore the region for the search of the sun's center can be reduced to a bounding box. The size of this is not known a priori and can only be defined empirically.

The same approach can be done with the radius. It is possible to define a range of valid values for the radius to limit the computation cost. There appears a trade-off between the computation cost/time of the algorithm and the cases that need to be covered.

Line integral The line integrals requires to compute the sum of every point over a circular contour. But this condition can be relaxed. Instead, only a certain number of points need to be computed. This number is given as a parameter for the alogrithm. A trade-off appears between computation cost and accuracy. The higher the number of points, the more accurate the detection is but the higher the computation cost.

Thresholding Even with a reduced number of points per line integral, the algorithm is still costly. But another optimisation is possible in the computation of this line integral. The image is stored on an integer of 16 bits/pixel. Making a 16 bits addition for each pixel takes a lot of CPU time. But by binarising the image b, this can be reduced to a binary sum.

The difficulty of image thresholding is to define the threshold to make a correct segmentation of the image. Indeed a threshold with a value too small will discard part of the . But with the integrodifferential operator, it is not a problem. The algorithm is able to correctly detect the occulter even with an incomplete circle. It fits the best possible circle inside the data (see later section 4.6).

The edge detection on Daugmans's operator is performed through a variation of the radius, as the differential operator is performed radially ($\frac{\partial}{\partial r}$). On the other hand, the Hough circle transform requires an edge detection as a primary operation.

4.2.2 Implementation

The implementation of the algorithm has been made using *matlab*. The pseudo-code of the actual implementation describes how it has been achieved. The code has been made from scratch and is entirely specific to the detection described for this particular problem. Although some parts have been inspired by a code for iris segmentation [15].

input image: I
foreach (x_0, y_0) *in range* **do**
  foreach R *in radius range* **do**
    for $\theta = 0$ *to* 2π *with n intermediate steps* **do**
      $x = x_0 – r * sin(\theta)$
      $y = y_0 + r * sin(\theta)$
      compute the line integral: $lineInt[R] += I(x, y)$
    end
  end
  Take the maximum index of the gaussian filtered discrete derivative of $lineInt[R]$
  Store the max value and the corresponding indices x_0, y_0, R
end
Output the indices of the maximal value as the detected circle x_c, y_c, R

4.2.3 Tests

First the input image is binarized with a given threshold chosen empirically with not much impact on the end result.

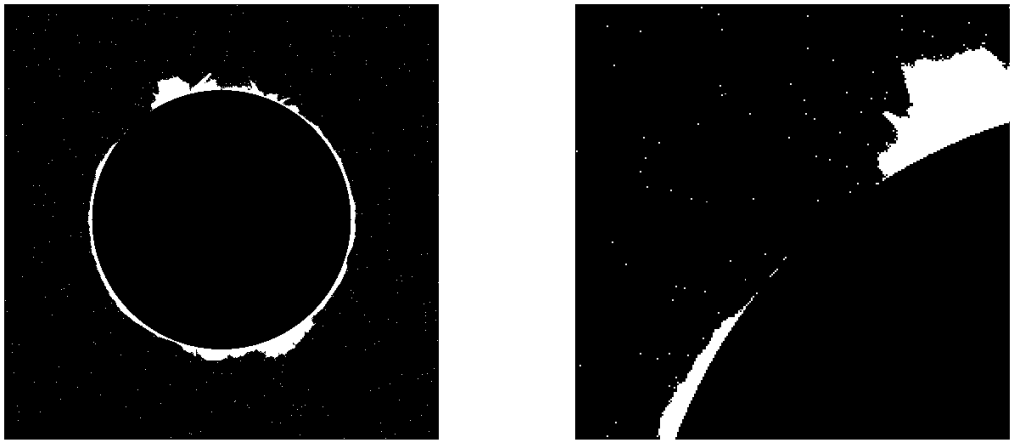


Figure 4.4: `Left`: binarised image fed to the input of the Daugman's integro-differential operator algorithm. `Right`: the thresholding operation causes part of the of the edges of the occulter's disk to be incomplete

The algorithm is ran over the binarized image and the result is displayed on the image. The detected circle parameters are $x_c = 512, y_c = 512, r = 305$ which corresponds exactly to that of the circle in the image. A pixel precision is achieved.

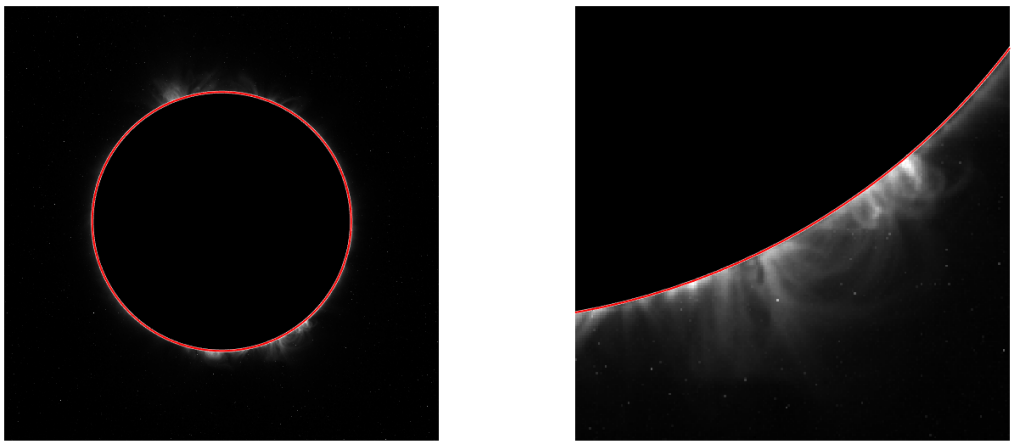


Figure 4.5: thresholded images

A higher threshold value leads to less points belonging to the circle to be present in the image. Almost 50 % of the points from the edges of the circle are discarded by the thresholding operation as can be seen on Fig. 4.6.

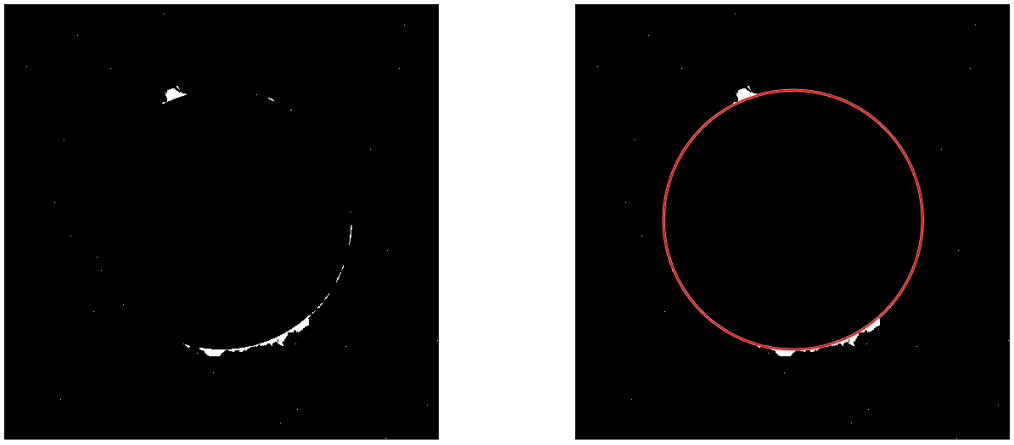


Figure 4.6: thresholded images

The detected parameters for the circle are $x_c = 513, y_c = 507, r = 305$, this represents only a one pixel error ($x_c = 512$). Despite the very limited points that are present in the image after thresholding. This is an extreme case, in practice a lower value of thresholding should be more adapted for all data. But it shows that the algorithm is not too dependent of particular values. This is especially important as the circle detection is performed with images of different exposure time. With high values of exposure time this is not very problematic as the occulter and the background will have a large contrast. But it is not the case for low exposure time images where the occulter might be blended and indistinguishable from the background in some parts.

This demonstrates the robustness of the algorithm to recover circle in imperfect conditions.

4.2.4 Performances

The algorithm has been tested over 200 images from the generated data set of different exposure time. The results are satisfactory as 10 images have an error of 1 pixel in the x or y coordinates detected. This gives a 95% accuracy with a 0 pixel error and 5% with a 1 pixel error, no error above 0 pixel is made. For the detected radius, the accuracy is 95.5% of the radii are correctly detected with 0 pixel error, and 4.5% with an error of 1 pixel, no error above 1 pixel is made.

This shows the robustness of the method against the movement of the occulter - the position can vary with up to 40 pixels in the image - and against the changes in exposure time of the image.

4.2.5 Verification on real images

Since the occulter was added virtually, it is necessary to stay critique about the results. The disk is added as a perfect circle with a single value in the image. This could correspond to an ideal case for which the algorithm's functioning is limited to. Therefore the algorithm might not be able to translate towards a real set up and its imperfections. To prove that the algorithm works correctly, it can be ran over the small data set of total solar eclipse of multi exposure (see section 3.4).

The first tested image is a short exposure time image of a total solar eclipse.

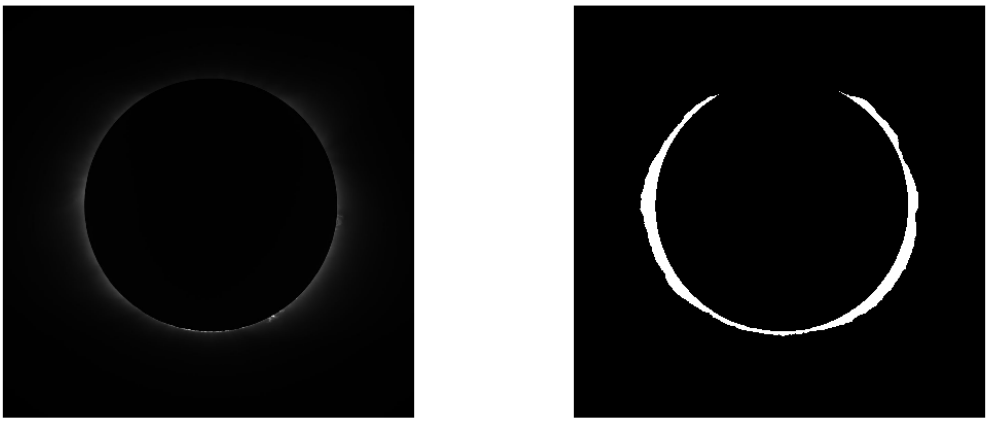


Figure 4.7: `Left:` short exposure image of total solar eclipse. `Right:` binarized image after thresholding, part of the circle is discarded due to the background being very dark in this region

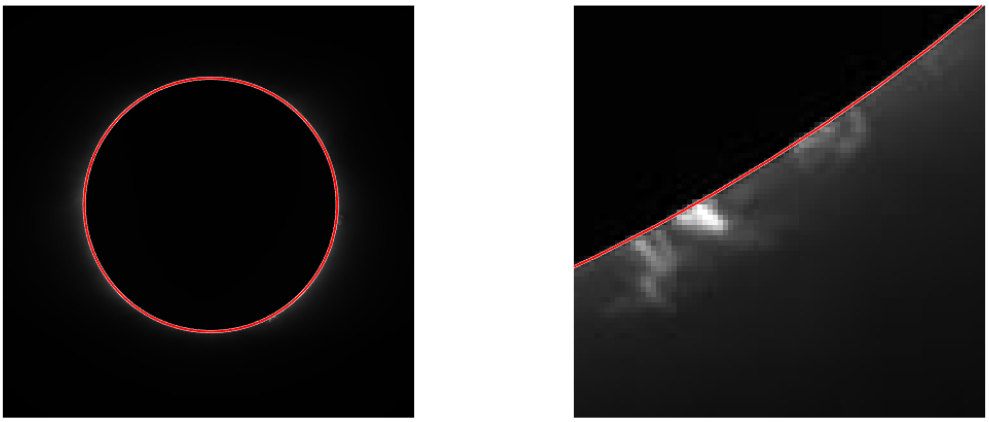


Figure 4.8: `Left:` detected occulter (red) of the short exposure image `Right:` zoom on the detected occulter(red)

The detected circle corresponds precisely to the edge of the occulter despite the low contrast between the occulter and the background. The precision is assessed visually.

The algorithm is also tested on a long exposure image.

Figure 4.9: `Left:` binarized image of long exposure. `Right:` zoom on a part of the circle where glare of light causes the circle to be imperfect

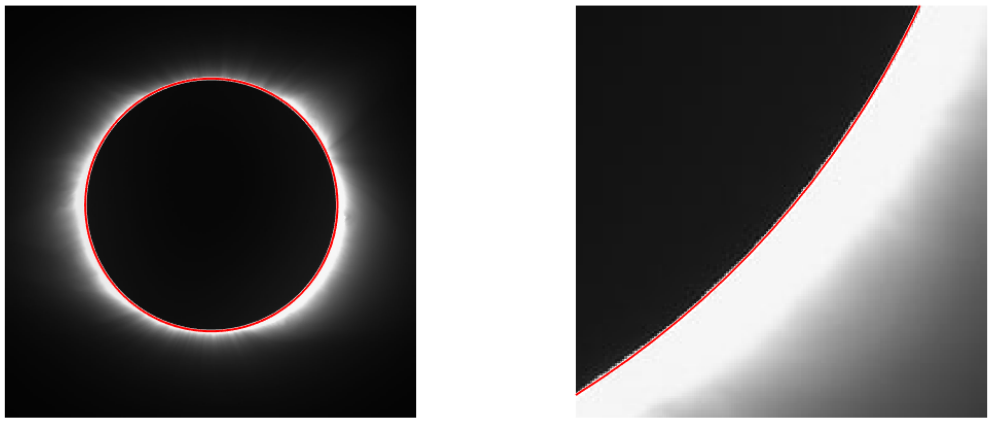


Figure 4.10: `Left:` detected occulter (red) of the long exposure image `Right:` zoom on the detected occulter (red) of the long exposure image

The verification can only be done manually as there is no ground truth data for this data set. But the results show that on a real total solar eclipse, the algorithm is capable of retrieving imperfect circles even if glares are present in the image or a blur of the edge of the circle. Of course, this is not a proof sufficient given the limited data and some fine tuning can be necessary depending on the data but this verification on real images after the simulated ones is the minimal step.

The tests performed on this data set also exposes the relative movement of the occulter and the necessity of a correct detection. The relative center of the occulter has been translated by 3 pixels along the x-axis and 1 along the y-axis between the two images tested. It is is even larger if the comparison is made between the first and last image of the sequence (5 pixels along x-aixs). Therefore verifying the assumptions made during data generation about translation of the occulter (see section 3.2).

It also verifies the assumption made on the circularity of the occulter. Strictly speaking, the occulter of the real images are not perfect circles (not exactly verifiying equation 4.1). But the algorithm still finds the optimum disk covering it without a significant impact on the result.

4.2.6 Optimisation

For both circle Hough transform and integro-differential operator, it is quickly noticeable that the computation cost remains way too high if the range of values is not strongly limited. This causes some problems as for an efficient calculation, it is necessary to know a precise range. But when tested over different data sets, it does not translate very well as the range depends on the size of the image and the characteristics of the image itself.

A simple, yet effective, trick is possible to reduce this cost. The complexity is highly dependent on the size of the image. Resizing the image reduces the relative range to be tested by the same factor. However, the precision required for a correct detection of the occulter has to be of the order of a pixel. When resizing, this pixel precision is not verified. But the result of the detection of the resized image can be used as a first rough estimate.

The first detection of the occulter is performed on the resized image by a scale factor s. It yields the results (x_r, y_r, R_r) the center and the radius of the resized image. Then the ranges of values to be tested on the full image is $x_r - \Delta \to x_r + \Delta$, $y_r - \Delta \to y_r + \Delta$ and $R_r - \Delta \to R_r + \Delta$. Where Δ is selected to ensure that the range comprises the circle. In practice, $\Delta \simeq \frac{1}{s}$ is sufficient.

So performing twice the circle detection actually decreases the cost of computation and avoids case specificity of the data set.

Figure 4.11: Detection of the circle on a resized image. Despite the lower resolution, the circle is correctly detected and serves as a first approximation.

The computation time has been decreased from a few tens of seconds - depending on the range tested - when treating images of size 1024 with a range of 100 for each x_0, y_0, r tested to under a second. Without any decrease in the accuracy of the method. And without having to give a range before hand. The method could still further be accelerated by using multiple steps of resizing in cascade. But it is sufficient for the current application and proves its feasibility for an embedded device with limited computational power.

4.3 Conclusion

The modified version of Daugman's intergo-differential operator is the method of choice. It detects circle within a pixel precision while keeping the computation time reasonable. Considering the size

of the images, this result is satisfying. It is robust and consistent against different settings and different data sets.

Chapter 5

Image alignment

The first and the most critical step of the total solar eclipse image processing is a precise image alignment[3]. For natural total solar eclipses, *neither the edge of the Moon nor stars are suitable for the high precision alignment of images due to their relatively fast motion during a total eclipse.* The case encountered here is similar. It is therefore necessary to use coronal structures themselves for the alignment. The alignment of images is not trivial as the images are not only translated but are different 'in content'. The classical techniques of image registration and template matching usually work well with (nearly) identical images that underwent an affine transformation. But more and more differences in image content such as saturation and intensity differences between images of different exposure time makes this task difficult.

It is important to first state what has to be achieved. Images from the data set are translated because there is a certain amount of time between shots taken by **PROBA-2** which causes a displacement of the sun in the image. Only translation has to be taken into account, rotation of the spacecraft is monitored with sensors therefore does not have to be corrected in this case. The alignment can not rely on the occulter as it is not fixed in the image nor aligned with the sun's center.

5.1 Review of image registration/alignment techniques

Image alignment also called image registration is necessary due to the movement of the telescope. But current existing coronagraphs or solar observers already align images for the observation of data. So one might ask how this alignment is performed for existing instruments even if it is performed on ground station. First this information is not always publicly available which makes it more difficult. **PROBA-2** uses the detection of the solar disc to perform the alignment. But it is of course not possible in our case as the solar disc is completely hidden in the case of a coronagraph. Some coronagraphs like LASCO use stars to align images but stars are not always available for example in PROBA-2 no stars are visible in the image. Plus using stars when dealing with images of different exposure time is not a good idea as low exposure time images might not detect the light from stars. And for high exposure time images, stars can be hidden by bright parts of the corona itself.

Some total solar eclipse photographs are simply aligned by hand by the user. This is the case for the data set from section 3.4 which was aligned by visual estimation.

More importantly, none of the instruments make an image alignment on board of the satellite, it is performed as a post-processing step in ground stations. Thus computation power is not a limitation.

Other means of image alignment have to be used. It is necessary to make a review of the possible techniques for Image alignment algorithms. They can be classified into two categories: Intensity

based and feature based algorithms. Let us review the different possibilities of image registration techniques and their advantages/disadvantages in the context of solar eclipse images.

5.1.1 Feature based algorithms

The first category of image alignment technique is based on features. The general procedure can be described as follows [6]

- *First search for the feature points in the image/template*
- *Then find the best matching between feature points in the image and the template*
- *And finally compute the warping based on the best feature points matching*

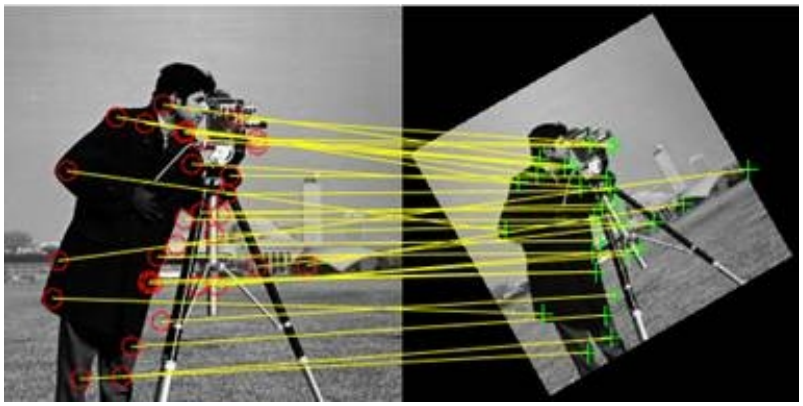


Figure 5.1: Example of feature points in the image used for matching two images. The warping here is a translation, a rotation and scaling. Source: https://nl.mathworks.com/discovery/image-registration.html

The problem with total solar eclipses, is the lack of features. The occulter's edges can not be used as a feature due to its motion. Features have to be found in the corona itself. But the lack of characteristic details in those parts makes it difficult as the light is very diffused. The only characteristic features can be found in the inner part of the corona but those features do not translate well with different exposure time. A feature based alignment algorithm requires feature points that are "unique and discriminatory" [6] between images. Most common feature detectors are: Harris corner detector, SIFT, SURF,...[6]

The use of such algorithms has also been discouraged by Druckmüller in the case of total solar eclipse alignment: *Methods based on finding corresponding points to align images are generally not adequate because of the absence of contrasty features in the images* [**Druck¨2**]. Common features are edges, corners, blobs [21] which are not very present nor distinctive in the case of solar eclipse images.

Most algorithms used for image alignment of multi-exposure images use feature based algorithms. Because features (corners, edges, etc) are often independent of the exposure time which make them very practical for alignment. Examples applied for digital photography alignment of multi-exposure images can be found [17].

Unfortunately, this leads to finding other means of alignment while this method is the most common and most documented especially in the case of multi-exposure images.

5.1.2 Intensity based algorithms

Intensity based algorithms use a measure of similarity and dissimilarity to find the best match between two images. *A similarity measure S is considered a metric if it produces a higher value as the dependency between corresponding values in the sequences increases.* [5]. This metric can be used in order to find the best matching between sequences.

For images, it can be used to find the translation that maximizes a metric.

Euclidean distance

A naive solution to compute the translation between images relies on the euclidean distance[6]. The distance between two images I_a and I_b can be computed as[6],

$$dist(I_a, I_b) = \sum_{i=0}^{N}\sum_{j=0}^{N}(I_a(i,j) - I_b(i,j))^2 \tag{5.1}$$

It computes the distance for each pixel in the image I_a with the corresponding in I_b. The distance gives a single integer.

In order to retrieve the translation, one of the image is translated with a known translation. Then the distance is computed and stored for each translation value. This allows to build a map containing the distance for each translation. Therefore the resolution of the problem consists in finding the minimal value of distance and get the related translation.

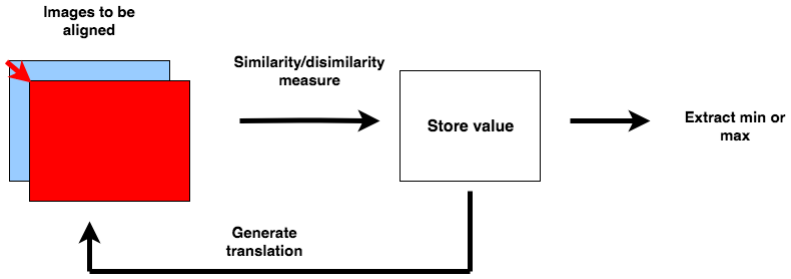


Figure 5.2: General overview of algorithms using similarity/disimilarity measures for image registration.

This algorithm is very expensive in computation. It requires a scanning of the images to compute the distance $\mathcal{O}(N^2)$ and an iterative search for the translation that minimizes the distance. If translations can range from $-\frac{N}{2}$ to $\frac{N}{2}$ in both directions than it leads to a complexity of $\mathcal{O}(N^4)$! This is a brute force algorithm.

It is possible to limit the complexity by limiting the range of values of translation to be tested. But this requires to know the expected maximum of translations a priori which is very limiting. Even in the case of a limited range, the calculation is still very expensive. For 1024 by 1024 images the computation is really high.

Unfortunately, while this method gives results, it is not a viable option for due to its very large computational power required.

Other measures of similarity/dissimilarity such as Pearson correlation coefficient, Spearman's rho, Kendall's tau, etc [6] exist. But they all share the same problem of computational power as all translations of the image have to be tested.

Phase correlation

Phase correlation is a more efficient way of computing the translation between images. It does not require to test for every point the similarity between the images which can be very inefficient. The calculation is performed in the frequency domain instead. The first step is therefore to convert the

two images to be aligned into their frequency domain representation. This is done with the use of the discrete 2D Fourier transform. For an image of size $N \times N$ with values $f(m, n)$, the fourier transform can be computed as:[8]

$$F(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) e^{-j2\pi(\frac{km+ln}{N})} \tag{5.2}$$

$F(0, 0)$ represents the DC component of the image. The image representing the fourier transform is not easily interpretable. Plus it is composed of complex values so the images have to be separated into real and imaginary parts. Or more commonly into amplitude and phase. In practice, all values are shifted in the image representation in order to have the DC component $F(0, 0)$ at the center.

The inverse discrete 2D fourier transform, which converts back to the spatial domain, is defined as:

$$f(a,b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l) e^{-j2\pi(\frac{ka+lb}{N})} \tag{5.3}$$

For the phase correlation method, both images I_a and I_b are thus transformed into their fourier transform G_a and G_b:

$$G_a = \mathcal{F}\{I_a\} \qquad G_b = \mathcal{F}\{I_b\} \tag{5.4}$$

Then the normalized cross-power spectrum can be computed: [3]

$$C = \frac{G_a G_b}{|G_a||G_b|} \tag{5.5}$$

Where the multiplication is performed elementwise. Then the inverse fourier transform gives the normalized cross-correlation.

$$c = \mathcal{F}^{-1}\{C\} \tag{5.6}$$

In order to understand how this can be used to retrieve the translation, let us take an example[24]. If I_b is circularly shifted with respect to I_a by a translation of a (t_x, t_y). Then their fourier transform will be related by:

$$G_b(k,l) = G_a(k,l) e^{-2j\pi(\frac{kt_x+lt_y}{N})} \tag{5.7}$$

Due to the property of the fourier transform that a shift in sapatial domain corresponds to a multiplication by an exponential in the frequency domain. The cross power spectrum is then computed:

$$C = \frac{G_a G_b}{|G_a||G_b|} = \frac{G_a G_a(k,l) e^{-2j\pi(\frac{kt_x+lt_y}{N})}}{|G_a||G_b|} = e^{-2j\pi(\frac{kt_x+lt_y}{N})} \tag{5.8}$$

The inverse fourier transform of this complex exponential gives:

$$c = \delta(x + t_x, y + t_y) \tag{5.9}$$

Where δ is a Kroenecker delta[24] i.e a single peak in the cross-correlation located at (t_x, t_y). The translation is therefore simply given by the location of the peak in the cross-correlation.

However the assumptions made for the example are too restrictive for a practical case of image registration. First, images are, in general, not circular shifts but linear shifts of each other. Edges of the images have to be correctly handled. Images are also not exact replicate of each other but carry noise which causes differences between them. All imperfections and differences between images lead to the cross-correlation to not be a perfect peak but rather a peak among a lot of noise. The correct detection of this peak is the main difficulty of this method.

The method of phase correlation applied to the alignment of total solar eclipse has been covered by Druckmuller in 'Phase correlation method for the alignment of total solar exlipse images'[4]. But it shows that the simplest version of the phase correlation can not be applied directly. The images contain noise and dissimilarities due to the brightness differences. The problem of maximum detection becomes difficult. It is not ensured that this peak is even a global maximum in the phase.

The modified phase correlation by Druckmuller applies multiple pre-processing steps to the image to solve that problem:

The normalized cross-power spectrum is modified to avoid dividing by values close to zero. A low pass gaussian filter is applied on the frequency domain of the cross-power spectrum to filter high frequencies. This requires a parameter to be tuned for the low pass filter. Low frequencies are filtered in the space domain by a filter which uses polar coordinates and the sun's center to achieve the results. This filter is particularly difficult to implement and requires parameters to be tuned correctly. And an unsharp mask is applied to remove the occulter. [4]

But this method has limitations. *The only disadvantage of the method, at present, is that it is not possible to automate it. Finding the algorithm for automatic estimation of the parameters [...] is the subject of future research*[3]. Therefore this method applied for an embedded device remains uncertain. The maximum location is difficult and unsatisfying in most cases.

Tests have shown that this method is very difficult to reproduce consistently over a data set of images. Many problems make the peak detection difficult. Mainly, bad pixels that are present in the images tend to give a peak for no translation in order to align those bad pixels. Edges from the removal of the occulter cause high frequencies to be present in the image. The removal with a low pass filter is imperfect and makes the peak detection difficult if not tuned perfectly. While this method gives supposedly good results - subpixel accuracy according to [4], it was not applied successfully over multiple images even of single exposure. This is due to the difficulty of tuning the different parameters that work for a given data set.

Other ideas

One idea that has come up during the research is to use the data of the sun's center of the data set. Instead of trying to determine directly the alignment, the problem becomes determining the sun's center to deduce afterwards the translation to apply to align the centers. This task is however very difficult, the solar disc is supposedly perfectly covered. And even if at a given moment it is not, its very high brightness makes an edge detection impossible in a real scenario. Therefore the sun's center has to be inferred from the corona itself.

One way to solve this problem could be to use machine learning techniques. The input would be the image pixels and the output, the coordinates of the sun's center. Given the large database of images and the ground truth values available for the coordinates, training a neural network for this task could be possible.

However, this task is very complex and uncertain about the achievable precision of the results. Plus the precision of estimating over a single photograph instead of two (when determining the translation) is expected to be less. Especially since a correct alignment requires a precision of the order of one pixel. This method has not been further tested.

5.2 Development of an alignment technique

The idea of determining the sun's center from one photograph is probably too ambitious (if not impossible) but it leads to other observations. The sun's brightness follows a radially decreasing gradient. Even if prominences are present in the corona. Using these geometrical observations can help finding other means of image alignment.

Methods described previously that are classical for image registration were not satisfying. Another one has been developed to achieve the results based on contours of intensity.

5.2.1 Contours

The specificity of the problem suggests that it is possible to use the radial gradient of brightness of the corona. Intensity decreases as the distance increase with respect to the occulter. Contour lines are such that *a **contour** line of a function of two variables is a curve along which the function has*

a constant value, so that the curve joins points of equal value.[19]. The two variables being the x and y coordinates of the image and the function is the intensity of each pixel of the image itself. The particular case of the sun suggests that it is possible to get a circular-shaped contour around the sun following iso value lines. The closer to the center, the higher the value of the contour.

A first contour test over a raw image using matlab's function `contour` can be performed.

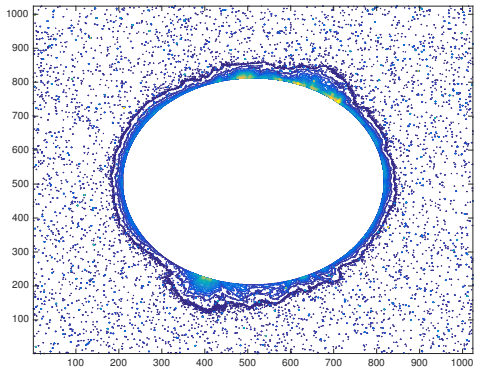


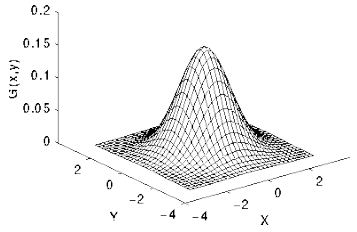
Figure 5.3: Contours generated using matlab

Multiple contours are drawn but they are affected by noise. Multiple isolated pixels are contoured due to noise and circular contours around the sun are too noisy as well.

One way to reduce this problem is to use a 2D gaussian filter over the image. It will filter the noise by averaging values. A 2D gaussian filter is defined as

$$G(x,y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5.10}$$

For an image, which is composed of discrete pixels, the gaussian function can be discretised over a mask of limited size. A gaussian is of infinite size theoretically but small values can be discarded. Then this mask can be applied to the image using convolution.



$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Figure 5.4: `Left:` 2D gaussian function from equation 5.10 `Right:` Discretised mask of the gaussian function to be applied on an image. Source:[10]

The resulting image is blurred.

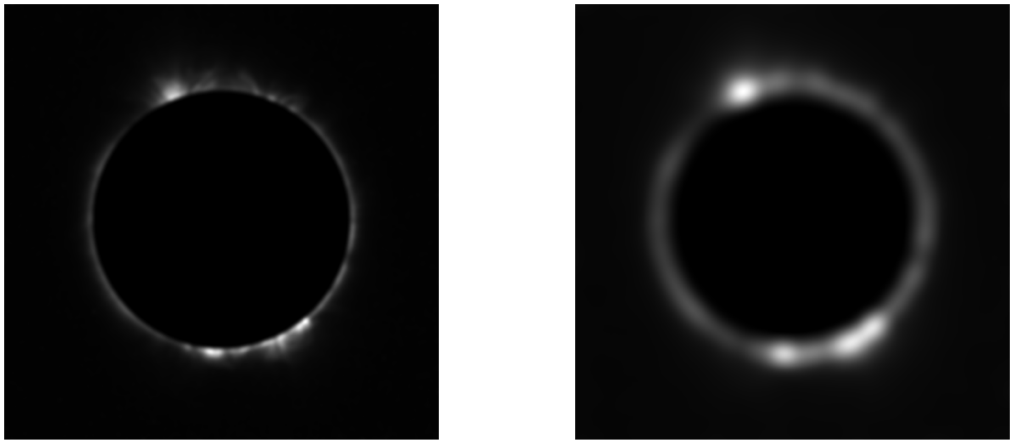


Figure 5.5: Images blurred with the use of a gaussian filter `Left:` $\sigma = 5$ `Right:` $\sigma = 20$

σ, the standard deviation, is a parameter that defines the intensity of the blur. In practice, it was found that performing a logarithmic operation on the blurred image helps getting better contours.

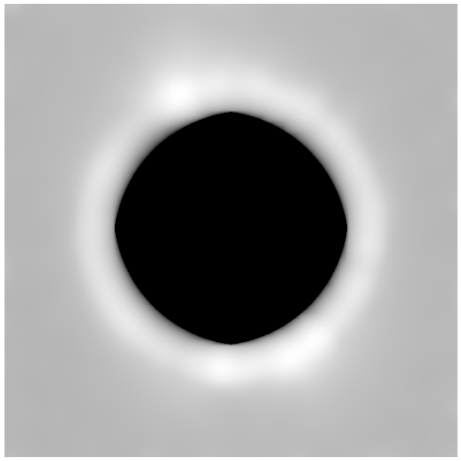


Figure 5.6: Image blurred with a gaussian filter and where each pixel value has been replaced by its natural logarithm

Contours can now be applied to this image blurred and that underwent a logarithmic operation.

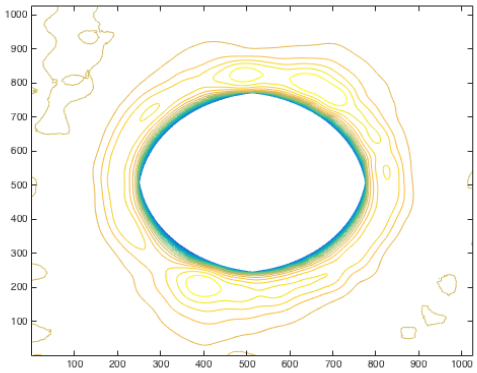


Figure 5.7: Contours of the image from fig 5.6 using matlab's function

Contours are now much smoother thanks to the blur and are less sensitive to noise. External contours are closed and with a circular-like shape.

Contours derived from matlab's library show multiple isovalue lines and are great for a quick test. But they do not allow much freedom for the data usage and especially to select the level of isovalue that is interesting for performing an alignment. Because only one contour can be selected and will be used for later operations. Many contours are bad or irrelevant and have to be discarded.

There exists multiple algorithms for contours. The **marching squares algorithm** can be used for generating contours. It is an efficient algorithm for computing the contours in 2D scalar field. An actual implementation from a library is used for the testing[11]. It allows to simplify the procedure. It takes as input the image (processed like in 5.6) and a value (a floating number) of the isovalue for which the algorithm computes the contour. The higher the value, the closer the contour is to the center and inversely as the intensity decreases radially. A further understanding of the algorithm is not necessary, it is only useful to understand the inputs and outputs.

The result is a binarized image which contains the contour for the isovalue selected. A first visual guess over multiple successive images of same exposure coming from **PROBA-2** shows that the lines are quite invariant between images. This is the case only if σ of the gaussian blur is sufficiently high. Otherwise contours will be too sensitive to noise in the images.

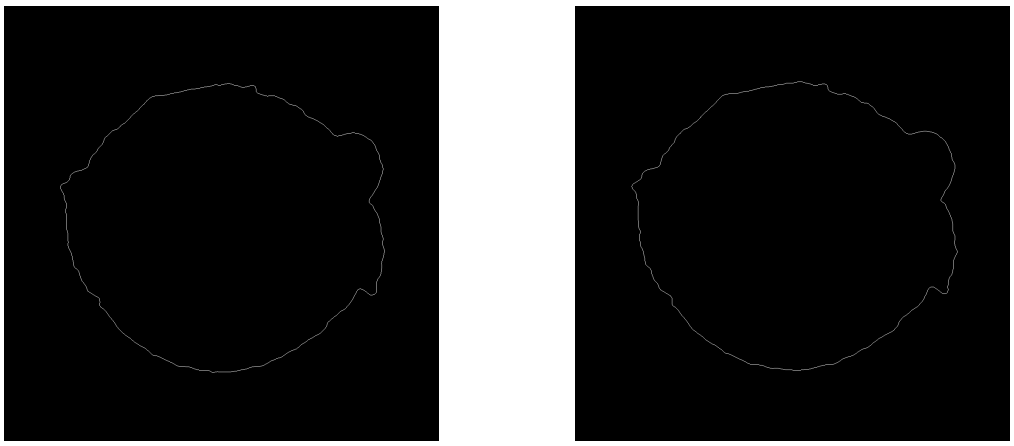


Figure 5.8: Contours obtained with the marching square algorithm with an isovalue of 5 of two successive (different) images

If this contour is sufficiently constant between images then it could be used for alignment of images. Due to the blur and the presence of the occulter in the image, an unwanted contour appears on the inside of the image. It does not represent any interesting information therefore has to be discarded. This can be easily done because the information about the position of the occulter is known (see chpater 7.3)

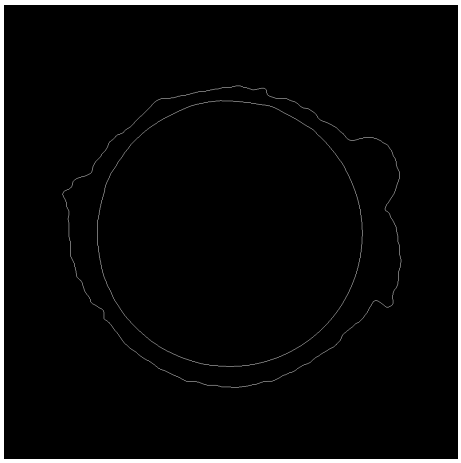


Figure 5.9: Contour of the occulter is present when applying the marching square algorithm. It can be removed thanks to the detection of the occulter previously discussed.

The detection of contours presents a huge advantage. They are independent of the position of the occulter as can be seen on figure 5.8. Therefore the alignment is not influenced by the position of the occulter. They are apparently constant between successive images.

These contours can be used to determine their center of gravity. This characteristic point - which is different than the sun's center - can be computed as x, y coordinates. Then the translation can be determined simply by:

$$\begin{aligned} t_x &= x_{c,1} - x_{c,2} \\ t_y &= y^1_{c,1} - y_{c,2} \end{aligned} \tag{5.11}$$

where $x_{c,i}$ is the center of gravity of the contour of image i. However this method has shown to be unreliable. Small differences in the contour lines cause a displacement of the center of gravity sufficiently high to make the alignment not precise enough.

5.2.2 Alignment of contours of images with same exposure time

The calculation of the center of gravity is too simplistic to allow a correct alignment but a more sophisticated one could be possible. The problem of alignment of images has been covered previously. The alignment can now be performed on the contours instead of the images themselves. This is equivalent to finding the translation (t_x, t_y) that maximizes the overlap between contours of the different successive images. This translation (t_x, t_y) is hopefully the same as that to apply to the successive images to perform the alignment. Contours of successive images, in addition of being translated, are not strictly equal due to the slight differences and the noise in the two images.

The most efficient way is to apply a phase correlation where the two input images are contours (G_a, G_b) of the images to be aligned. From section 5.1.2.

$$c = \mathcal{F}^{-1}\{\frac{G_a G_b}{|G_a||G_b|}\} \tag{5.12}$$

In this case, the application of the phase correlation is reduced to its simplest form. The problems encountered during the application of the phase correlation on full images are avoided in this case. (No filtering, no noise related problems, etc).

Tests

Let us test the method with two input images.

Figure 5.10: Two input images that have to be aligned

The misalignment of the coronal structures can be visualised by overlaying the two images. The colors (pink and green) show the differences between images (Matlab's function imfuse). As can be seen, the coronal structures are quite severly unaligned as the presence of color indicates. The color also indicates the displacement of the occulter between images(green disk).

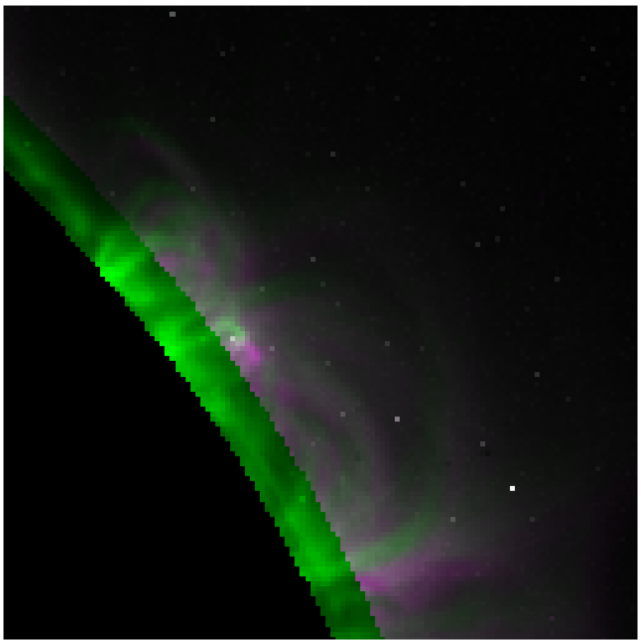


Figure 5.11: Overlaid unaligned images with matlab's function imfuse. Green and pink regions show differences between the two images. The coronal structures can be seen as unaligned.

Contours are thus computed over these images with an isovalue selected empirically to obtain a correct contour. A visual observation shows the similarity between the two contours. But a more detailed look of the overlaid contours (using imfuse) shows the displacement between the two.

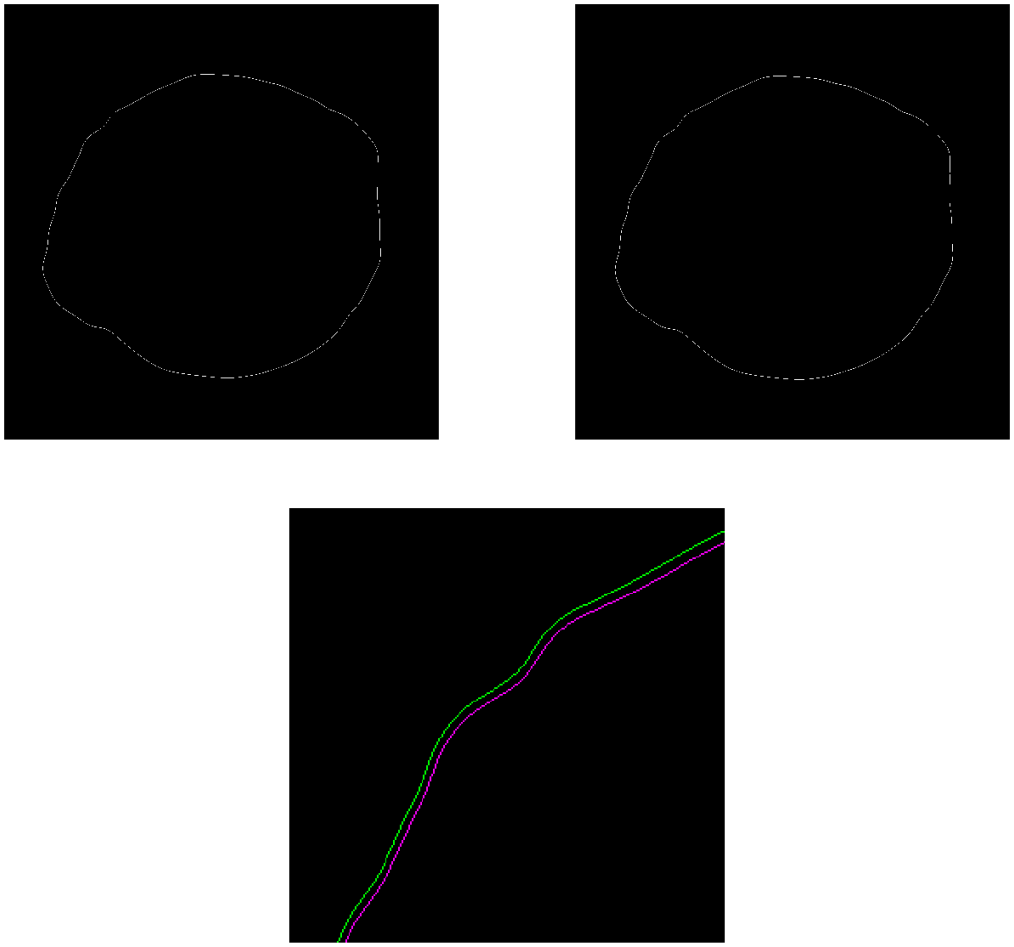


Figure 5.12: `Top:` Contours of the successive images to be aligned. The contour of the occulter is removed with the detection. `Bottom:` Zoom on the contours overlaid in one image, the translation between the two is visible.

Then the phase correlation method is applied with both contours as input images (equation 5.12). The result of the phase correlation can be visualized simply as an image of the same size as the images. The location of the peak will indicate the translation.

Figure 5.13: Zoom on the output of the phase correlation method applied to contours.

The red pixel indicates the center of the image (matlab's indices start at 1, thus $(513, 513)$ is the center pixel of images with size 1024), the peak of the cross correlation method is very distinct and is located at $(510, 509)$. Thus the retrieved translation between the images is $[510, 509] - [513, 513] = [-3, -4]$.

If the second image is now translated by this amount, let us verify the result visually. By using again matlab's function *imfuse* to merge the images.

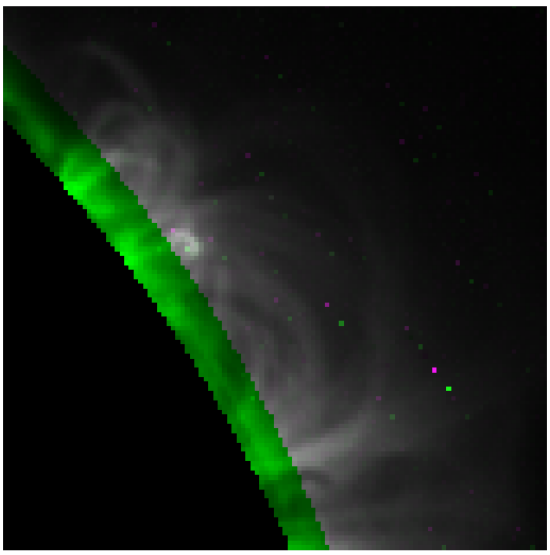


Figure 5.14: Overlaid aligned images. Coronal structures are now aligned as the lack of colors indicates that almost no difference is present between the two images.

The resulting image is almost only grey in the coronal region. Which indicates that there is almost no difference between the two images aligned images (to compare with Figure 5.11). The alignment of the structures is successful. The green region indicates that this part of the image was covered by the occulter in one of the two images but it does not prevent the correct alignment that was performed. The bright pixels of colors that can be seen on the right correspond to bad pixels that have been translated. The method is also robust against these values which can prevent a correct alignment.

In addition of a visual assessment, the value of the translation can be retrieved from the sun's location (see 7.3). In this case, the value for the two images are respectively $[517.57, 520.42]$ and $[520.36, 524.4]$ thus the ground truth translation to apply is $[517.57, 520.42] - [520.36, 524.4] = [-2.7900, -3.9800] \simeq [-3, -4]$. The ground truth translation matches with the result obtained by the method.

Performances

Now that the algorithm has been demonstrated and verified over a pair of images, it is necessary to assess its performances over a larger data set. This is where the generated data set and the corresponding ground truth values described in section 3 becomes very important. As a visual verification would not allow a practical way of assessing the performances. Plus the values of the retrieved translation come from an external observatory (*Observatoire royal de Belgique*) Making it a more a reliable and trustworthy assessment.

The process is automated and ran over the large data set. Images are stored in a chronological order, successive images are aligned by pairs. To perform the alignment tests, images are taken by pair. Thus image 1 is taken as the reference then the translation of image 2 is computed. Image 2 becomes the reference and the translation of image 3 is computed and so on until the last image. Actually, the computation of the contour can be performed only once per image and stored for the alignment with the next image to allow faster computation.

Values of the ground truth translation that are computed in the header files are floating point values. It means that the ground truth translation will also be a floating point value. But the predicted translation with the method is an integer since the calculation is made on the discrete grid of pixels.

One solution to measure the error could be to round the value of the ground truth translation and then compute the difference. But this method would not represent correctly the error. For instance, if the ground truth translation is 0.45 pixel to the right, rounded to 0. And the predicted translation is 1 to the right then the error would be of 1 pixel. Which is higher than the actual error of 0.55.

Therefore the error is computed as a distance:

$$E = \sqrt{(t_{x,true} - t_{x,pred})^2 + (t_{y,true} - t_{y,pred})^2} \tag{5.13}$$

A table showing the typical results obtained when running the algorithm is shown here:

True translation	predicted translation	Difference	Error
$[-0.85 - 2.98]$	$[-1 - 3]$	$[0.15 \quad 0.02]$	0.151
$[0.97 \quad 0.009]$	$[10]$	$[-0.03 \quad 0.01]$	0.031
$[-0.47 - 2.52]$	$[0 - 2]$	$[-0.47 - 0.52]$	0.701

To better visualize the results, an histogram can be plotted with the errors over the whole data set tested. The histogram can be read as follows, the x-axis represents the error (in pixels) and the y-axis, the number of images that are aligned with this error. The data set contains 200 successive images. They are taken in series of 5 or 6 then are rotated and a new sequence starts. Each sequence (of 5 or 6 images) will be aligned separately then will start again. The results of the alignment over 173 images (200 images excluding the first one of each sequence) gives:

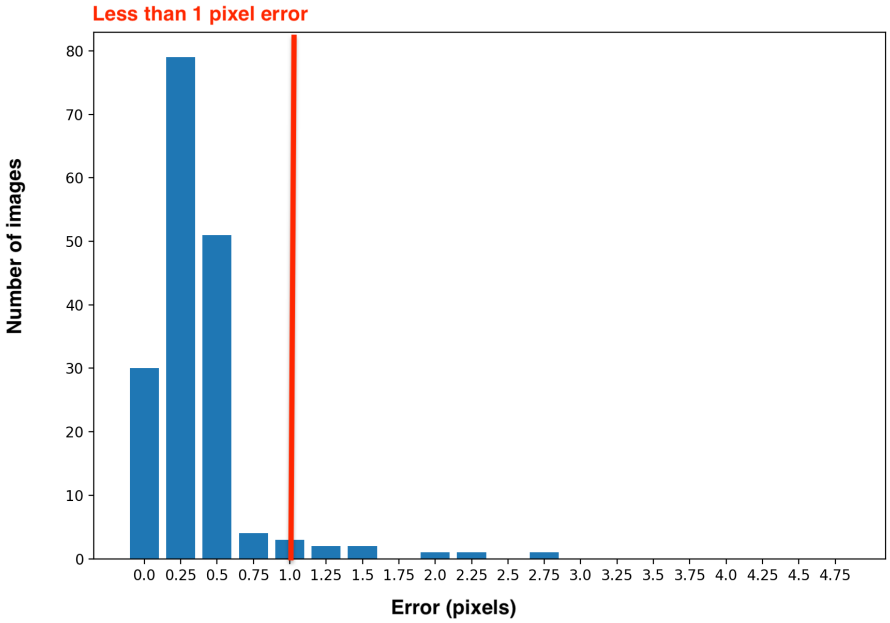


Figure 5.15: Histogram of the errors made during the alignment of the data set of images (200).

As can be seen the alignment performed is satisfactory. If an error of less than one pixel is considered as a correct alignment then for this data set, 167 images have been aligned with a precision of 1 pixel or under. This makes an accuracy of $\frac{167}{173} = 96.5\%$ of images aligned with under a pixel of error. The outliers correspond to images where a lot of noise was present in the contour, this corresponds to images of a lesser quality.

To show the capability of the algorithm, here is the misalignment error that would be made on images if no correction was performed (scale on x-axis is much larger than the previous histogram).

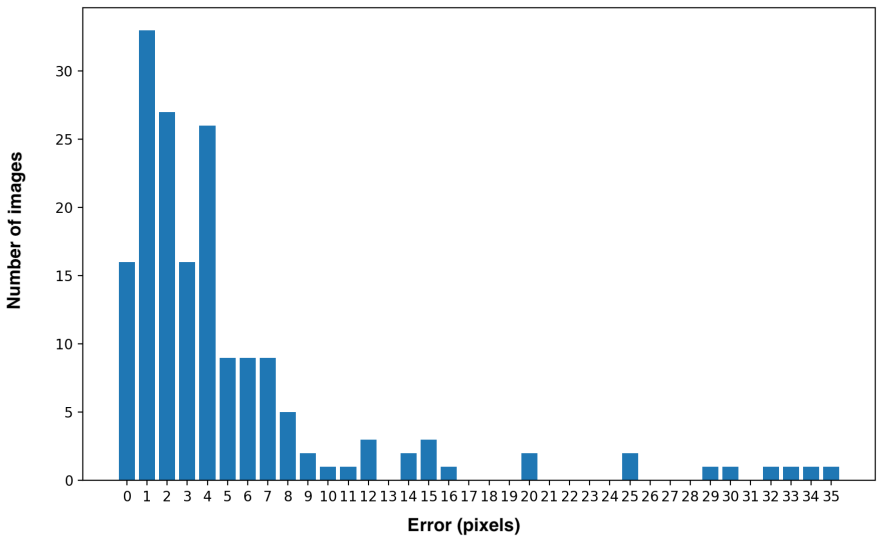


Figure 5.16: Histogram of the misalignment error on the images if no correction is applied. The necessity of image alignment can be seen as errors up to 35 pixels can be done otherwise.

It is to be noted that the **performances are independent of any assumptions made during data generation**. Indeed, in the case of single exposure images, the addition of the occulter does not have an impact on the contour used for the alignment. And the data used comes from real raw unaligned images of the satellite **PROBA-2** Therefore, it proves that this method of alignment can be used in a real context of single exposure images. It allows a correct alignment while using only a limited part of the image to compute the contour.

Two parameters are important for the correct functioning of the algorithm. The first is σ, the level of the blur applied with the gaussian function. The second is the isovalue that defines the level at which the contour will be drawn. σ is not too sensitive for the results, in practice, a value greater than 3 and up to 20 gives similar results. Greater than that, details in the contour become less distinguishable and the alignment is worse. A value of $\sigma = 5$ can be selected and fixed for the rest without impact on the resluts. The isovalue is also not too sensitive as long as it selected in a correct range in order to give contours that are: Far enough from the occulter to not be covered by it between succesive images, not too far in order to not be located in the noisy part of the image. This isovalue has to be selected empirically which makes automating this task difficult over multiple data sets. This value is defined by hand in this work. An automatic way of determining a correct isovalue could be possible.

5.2.3 Alignment of contours of images with different exposure time

Now that the technique for image alignment has been applied to successive images of a single exposure time, it is important to consider how the algorithm can translate in the case of multi-exposure images. Intuitively, the contour is directly linked to intensities in the images therefore not a constant over images especially with different exposure time. If the contour is directly applied on two images with different exposure time with the same parameter, the contour will inevitably be different.

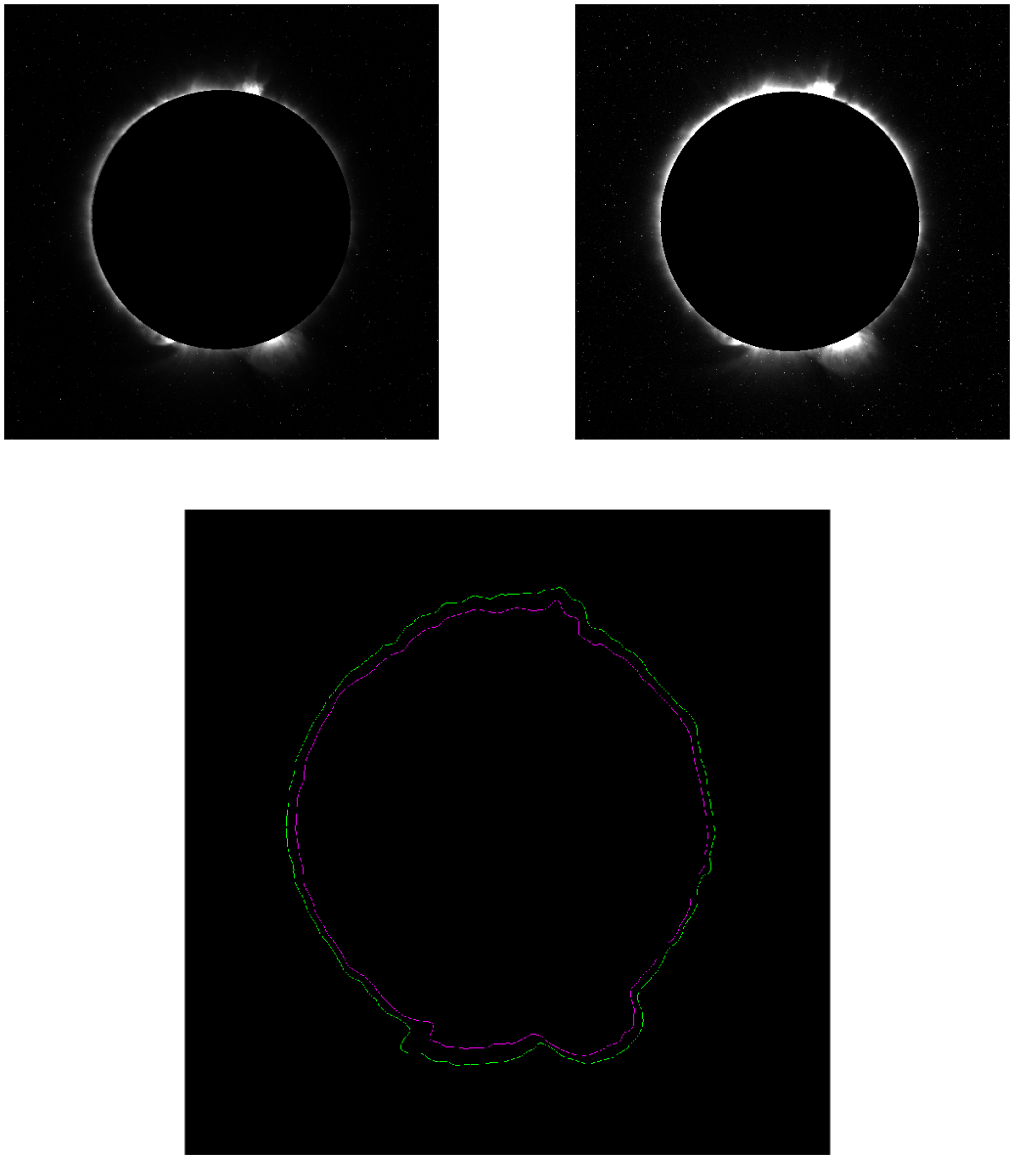


Figure 5.17: `Top:` Input images taken at successive instants with a different exposure time `Bottom:` Contours of the uncalibrated input images with a same isovalue. Contours can not be used for alignment.

The first step consists in calibrating the images with respect to the exposure time. If the exposure time is doubled, then it corresponds to a doubling in the intensity a priori (more details in section ..). Therefore, the intensity is first corrected to have an equivalent level between images. In other words, images that have a time of exposure that is half of the other image must first be calibrated by multiplying all the intensities of the pixels by 2. Images are always aligned by pairs with successive exposure time. Therefore the calibration consists in multiplying by two the intensities.

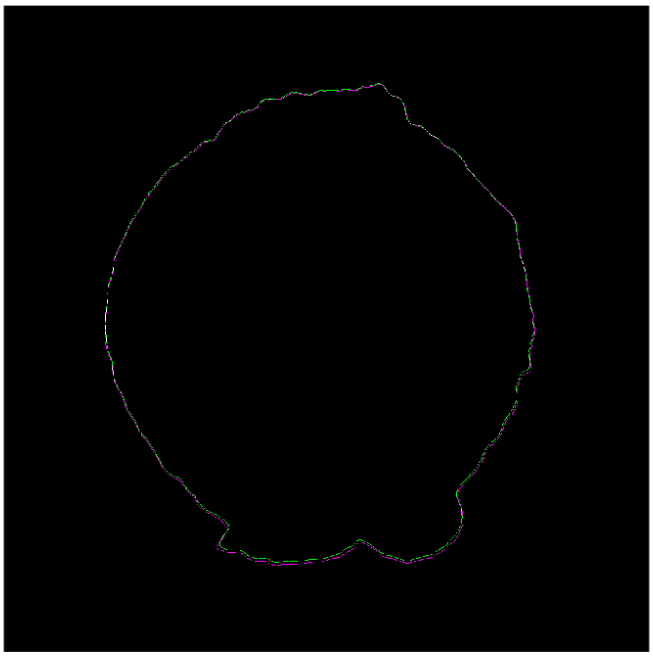


Figure 5.18: Contours of the calibrated input images with a same isovalue. The alignment can now be performed as previously.

Contours in the internal part of the corona can not be taken due to the saturation present. Contours have to be taken in more distant regions. A direct application of the method described in the previous section is performed except that the images are scaled before taking the contours gives the following results.

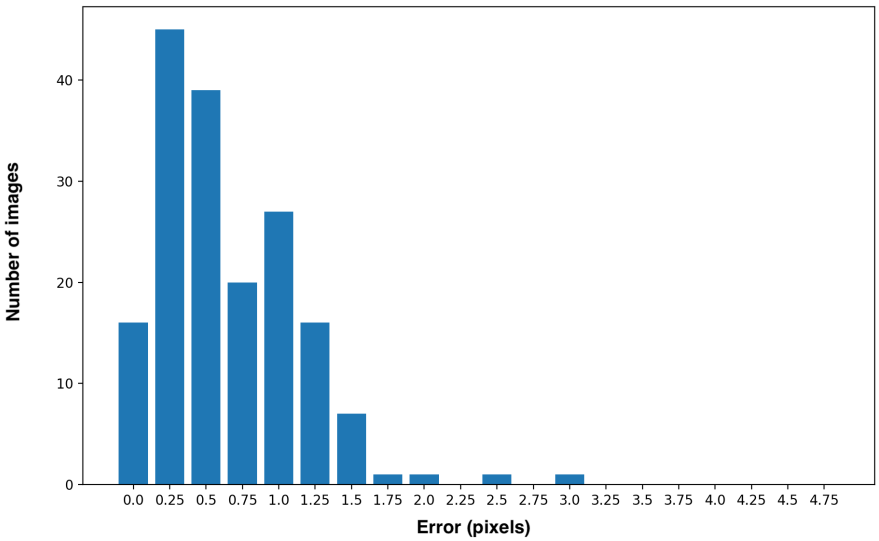


Figure 5.19:

Results are quite good as the correction of very large translation is performed. However, one can see that the errors are larger than for the case of single exposure images. This result can be

explained by the fact that images are not exact multiples of each other. Indeed as explained during data generation, images with a greater exposure time will have a better resolution of the contrast in the image. Therefore when doubling the intensities, contours are not exactly identical with the given isovalue.

One way to solve this problem consists in finding the optimal isovalue that matches the contours of both images. This value is not known a priori and can not be deduced simply as it varies from images to images. As it is impossible to find the optimal value simply, an iterative search can be performed.

Let us say that the reference has a contour with an isovalue i. Then the image that needs to be aligned will have contours with isovalues ranging from $i - \Delta$ to $i + \Delta$. For each value of isovalue, the phase correlation is performed. The value of translation is computed and stored as well as the value of the peak of the phase correlation. The optimal isovalue is thus selected as the one that gives the maximum in the peak of the phase correlation. The computed translation is then simply the translation that was computed with this isovalue.

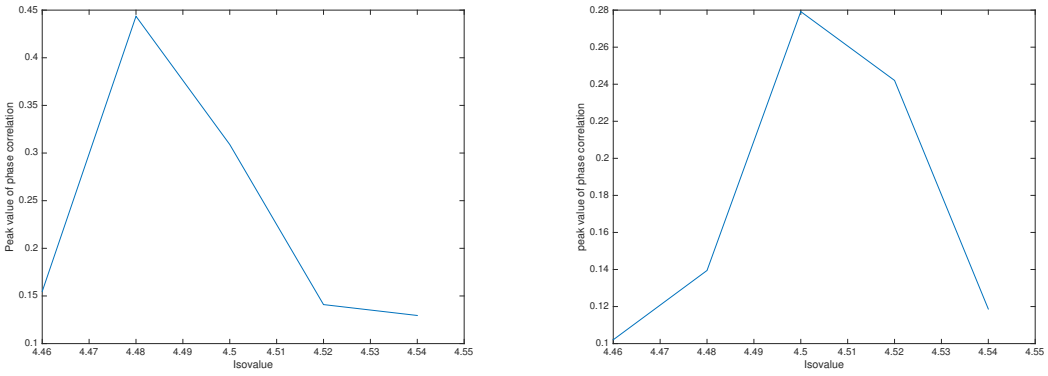


Figure 5.20: Graph showing the variation of the maximal value of the peak of the phase correlation on contours when varying the isovalue. The best isovalue is selected and the translation is computed from. Left and righ graphs show that for a different pair of images, the maximal value is different.

A graph showing the evolution of the isovalue and the value of the peak of the phase correlation can be drawn. This calculation is performed for each pair of images. This increases the computation power required. However with only 6 intermediate steps of isovalues, it is possible to get accurate results. Thus making the additional cost of computation for finding the optimal relatively low while keeping a good accuracy.

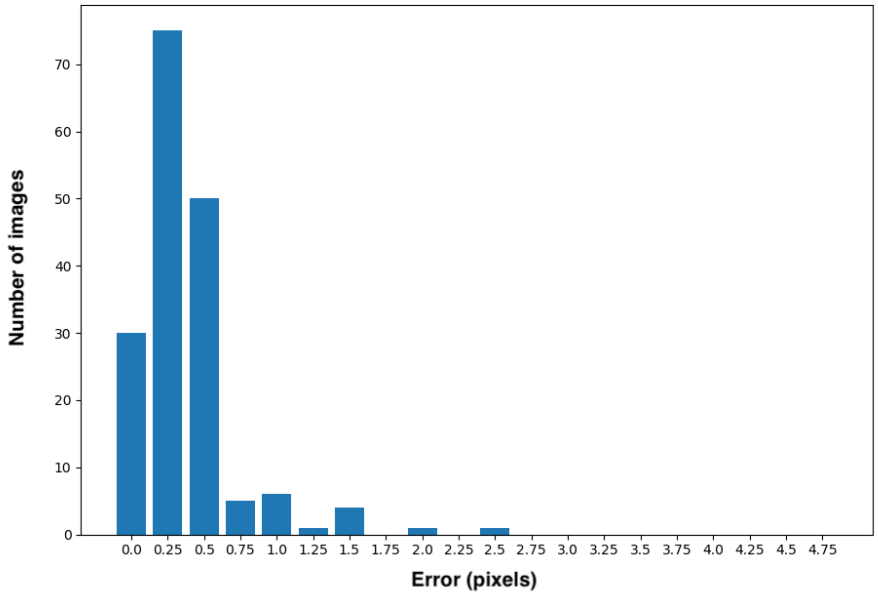


Figure 5.21: Histogram of the error when performing an iterative search for the optimal isovalue with only 6 values tested.

Therefore determining the translation is now performed in a 3D space. Much like the accumulator in the occulter detection (see chapter 4). It consists in finding the maximum in x, y coordinates for the translation but also according to the maximum of the contour level. It is the global maximum of the 'accumulator'. Although the contour level value is not relevant for any further calculation other than finding the translation.

Results obtained with this method are thus satisfying as it allows an error of less than 1 pixel of the order of 90%. Given the difficulty of aligning images with different exposure time, this is quite satisfactory. The error made if images were not aligned is the same as for previous tests (see Figure 5.16). Thus correcting large errors up to 35 pixels is performed.

Verification

In the same way as for the occulter detection, this method can be verified over the small dataset of real images of solar eclipse (see section 3.4). It allows to verify that the assumptions made during the tests can translate over other images. Especially the fact that contours are supposedly constant over different exposure time images.

The problem with this set of images is that it was aligned by hand (as stated by the author). Therefore the reliability is not 100% guaranteed. As it can be observed, a shift of a single pixel is often indistinguishable. Therefore the precision is not expected to be that of a pixel.

Since images are already aligned, to test the alignment algorithm, images were translated by a known amount. Then the capability of the algorithm to recover this translation is estimated.

First two input images of different exposure time are selected. The second image is shifted by a translation $[10, 10]$ (see Figure 5.23).

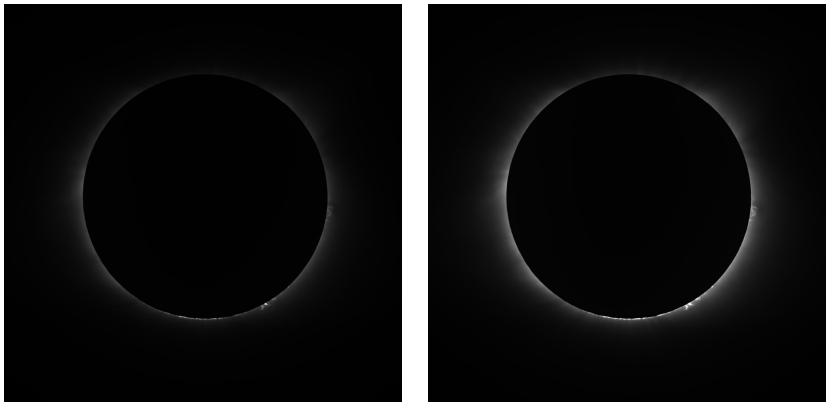


Figure 5.22: Images of respectively 16ms and 32ms of exposure time from real natural total solar eclipse.

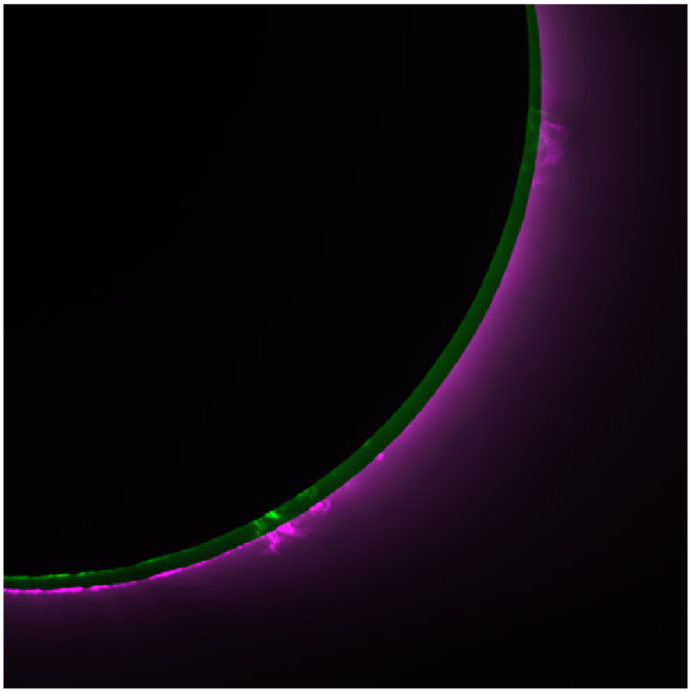


Figure 5.23: Zoom on the overlaid misaligned images. Coronal structures are clearly misaligned.

The first image is calibrated i.e its values are doubled. Then contours are computed with an isovalue selected empirically for this data set. And the search for the optimal isovalue is performed as previously explained. As can be seen, the assumptions that contours are relatively identical between images of different exposure time is verified. As this is real data and not generated data.

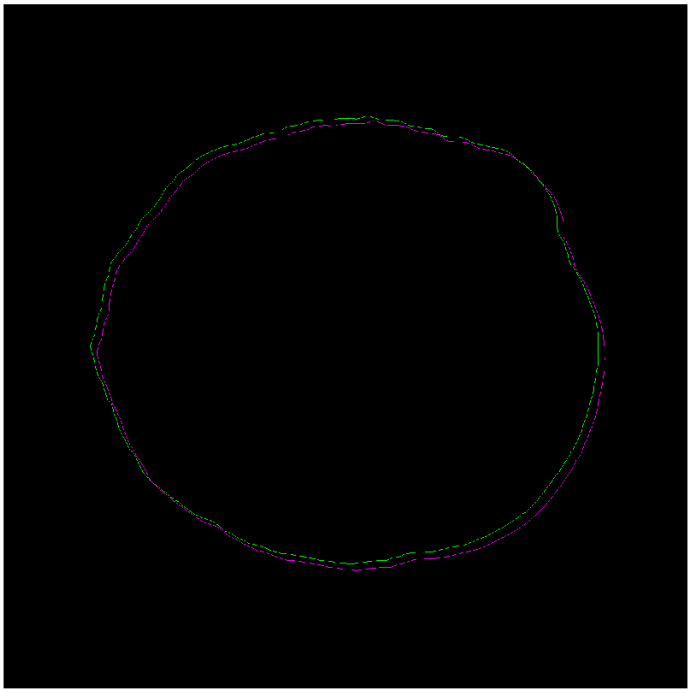


Figure 5.24: Contours of the images of different exposure time

The phase correlation method with the optimal isovalue gives the predicted translation. If this translation is applied on the image, one can see that the misalignment is corrected. Coronal structures are aligned as it can be seen on structures indicated by the arrows.

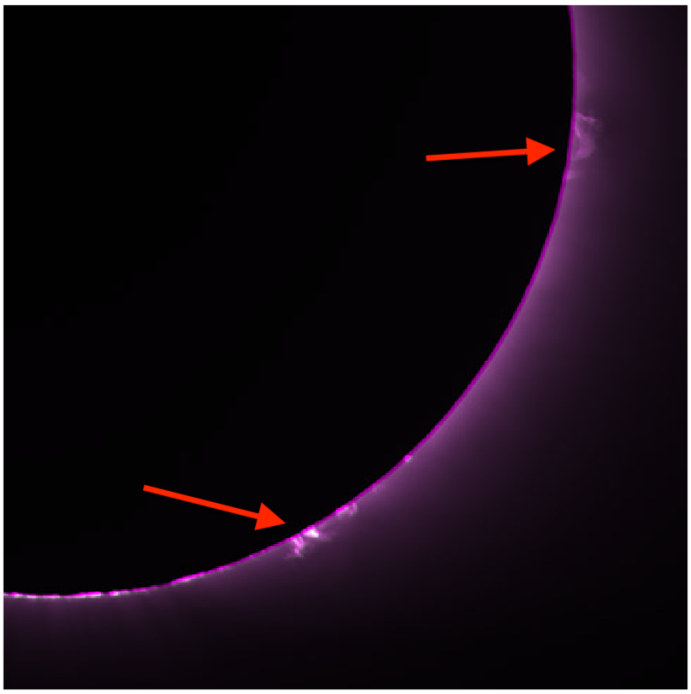


Figure 5.25: Input images aligned and overlaid. The red arrows indicate regions where it is easy to see the correct alignment of the images in comparison with Figure 5.23

The retrieved translation calculated with this method is $[-11, -11]$ whereas the translation that was applied on the images was $[-10, -10]$. Again as there is no guarantee that images were correctly aligned in the first place, this is satisfactory. As a visual observation shows the quality of the alignment performed.

Let us show another example with two input images from the same data set but with larger exposure time. The second image is tranlasted by $[-15, 10]$.

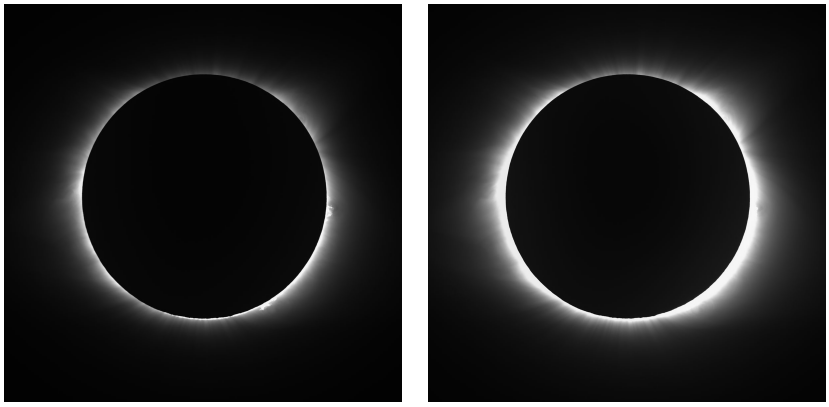


Figure 5.26: Images of respectively 64ms and 128ms of exposure time from real natural total solar eclipse.

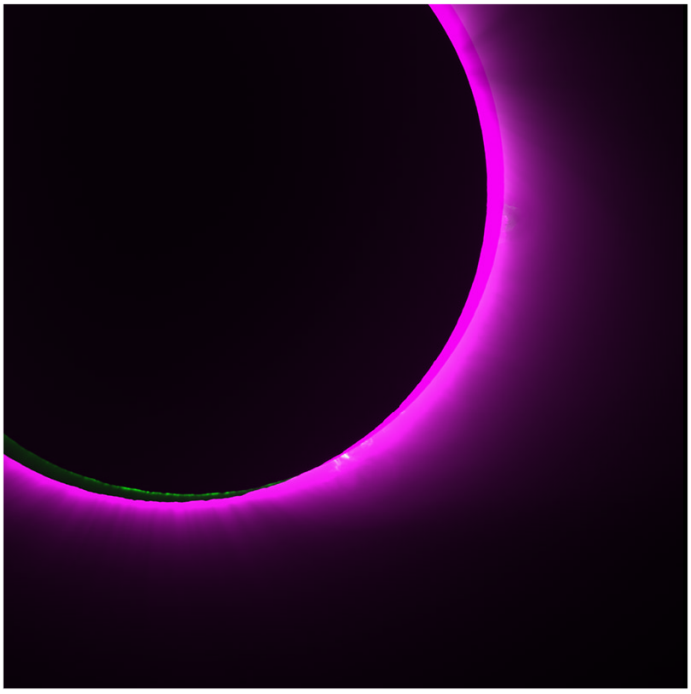


Figure 5.27: Zoom on the overlaid misaligned images

The detected contours over these two images are still very identical despite the visible difference between the two images due to the change of exposure. The phase correlation method is applied and the detected translation computed.

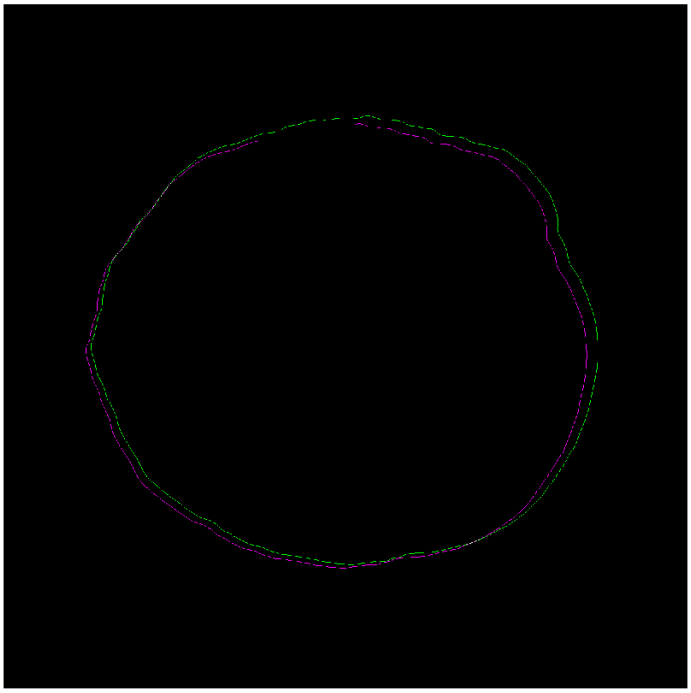


Figure 5.28: Contours of the images of different exposure time

The result is visualized by overlaying the two images but it is more difficult to assess in this case. As images are much more different, the saturated part of the second image covers almost all details. But the predicted translation is $[14, -11]$ which corresponds to that applied to the initial image with 1 pixel error. Therefore the alignment can be considered successful.

Figure 5.29: Overlaid aligned images.

The verification of the algorithm over real images has shown the reliablity of the method and that the assumptions made for the data generation were realistic. As similar results are obtained with images of total solar eclipse. The only parameter that had to be changed over the data set is the isovalue. Although it works for a wide range of values for the isovalue and is not too dependent on a fine tuning.

5.2.4 Conclusion

This method of contour alignment is specific to the application of sun's corona alignment. It is not supposed to be working in a more general case of image alignment. This procedure of alignment of contour of intensity is not detailed anywhere in the literature - at least that I had access to - therefore its efficiency was a big question mark. However tests have shown that this method is applicable with satisfactory results. It is impossible to know with confidence how this method could translate into other images but the assumptions made for the different exposure time should not have too much effect. The images used are still real images (from PROBA-2) therefore the quality of the method is still convincing as it allows the alignment of real data.

This method is also not dependent on the occulter as the contour is performed far from the central part. This is important in order to keep consistent results. Since the occulter is virtually added, if results were too dependent over the occulter, they would not be reliable in a real case. But here it is not the case. Also effects appearing around the occulter such as glares, blurs or light bending effects could cause differences in real images but contour alignment is not sensitive to those effects.

The method uses the geometry of the current problem meaning that it is not a universal method for alignment. But it works in the case of total solar eclipse images.

5.3 Conclusion on image alignment

Multiple image alignment techniques have been reviewed. Feature-based algorithms have been ruled out due to their non applicability in this context. The phase correlation method (and its modified version) is difficult to implement and did not give consistent results. Plus its dependency of parameters makes it non adequate for an embedded device. However this method, if correctly implemented and tuned, is supposedly more precise than the contour alignment method as the author claims a sub-pixel accuracy.

The contour alignment method is the one which gives the best results. A lot of trial and error has been made on other methods but only this one gives consistent and reproducible results. This method presents several advantage. It is more robust against bad pixels in the image. As a lot of bad pixels were present in the images but the algorithm still performed contrary to the phase correlation which is more sensitive to bad pixels.

This method can be more efficient in computation cost than the phase correlation (of full images) as less operations are necessary before hand. The phase correlation of contours can be heavily optimized as the fourier transform (equation 5.2) is computed over a binary image in comparison of a full image. This optimisation has not been implemented and is left as a suggestion.

Method	Performance	Computational cost
Euclidean distance	Ok	Too large
Phase correlation	Difficulty to have results	Ok
Contour alignment	Ok	Ok

Chapter 6

Image composition

Once the images are correctly aligned, it is possible to proceed to the composition of the HDR image. For each pixel (i, j) in the final image, a value has to be attributed taking into account all the pixels at (i, j) in the N images of different exposure time. The composition can not be performed directly on the data as the values are not calibrated. The different exposure time in each image causes a value at a pixel (i, j) to represent a different radiance in the scene. The first step is to calibrate the images in order to have all images representing equivalent pixel values for the same radiance in the scene. Then the composition is performed by attributing more importance to pixels that are correctly exposed and less to those that are overexposed or underexposed pixels.

6.1 Image calibration

As previously mentioned the exposure time of an image is the length of time when the sensor is exposed to light. For a given radiance in the scene, one might expect that doubling the exposure time will double the pixel value recovered by the sensor (given that no saturation occurs). But in practice it is not necessarily the case. Non linearities can be present, due to the camera, between radiance in the scene and the pixel value. This non linear mapping can be characterised by the characteristic curve - or Hurter–Driffield curves, H–D curves, D–logE curves- of the camera. This non linearity is especially present in cameras that use films. But digital cameras are not exempt. 'Digital cameras, which use charge coupled device (CCD) arrays to image the scene, are prone to the same difficulties. *Although the charge collected by a CCD element is proportional to its irradiance, most digital cameras apply a nonlinear mapping to the CCD outputs before they are written to the storage medium* [1]. Although in some cases, a simple multiplication is performed as a calbration[3].

This process of recovering the characteristic curve response of a camera has been covered [1]. A short explanation is detailed here.

For a pixel value $I_{i,j}$, a radiance E_i and an exposure time t_j. The pixel value $I_{i,j}$is related by[1]:

$$I_{i,j} = f(E_i t_j) \tag{6.1}$$

An ideal characteristic curve $f()$ would be such that $I_{i,j} = AE_i t_j$, it is directly proportional to exposure and radiance. But this is not the case in practice as non linearities appear. Equation 6.1 can be rewritten as:

$$f^{-1}(I_{i,j}) = E_i t_j \tag{6.2}$$

Taking the natural logarithm:[1]

$$\ln f^{-1}(I_{i,j}) = \ln E_i + \ln t_j \tag{6.3}$$

It is possible to retrieve this characteristic curve for a given camera by only using images of different exposure time. By selecting N pixels at various positions in P photographs of the same scene with

different exposure time. The curve can be estimated using the least square error method. $lnf^{-}1()$ is denoted g, then the problem becomes[1]:

$$\mathcal{O} = \sum_{i=1}^{N}\sum_{j=1}^{P}[g(I_{i,j}) - \ln E_i - \ln t_j]^2 + \lambda \sum_{z=Z_{min}}^{Z_{max}-1} g''(z)^2 \tag{6.4}$$

The first term is the least square and the second term is a smoothness term to ensure the calculated curve to be smooth and avoid overfitting. Where N is the number of photographs and P the number of pixels (fixed between photographs) that will serve for calculating the curve.

This method has been applied for recovering the characteristic curve of the data set of images of natural solar eclipse (see chapter 3). 100 pixels were selected randomly in the image. Then equation 6.4 has been applied and solved numerically using the procedure described in [1].

The result can be

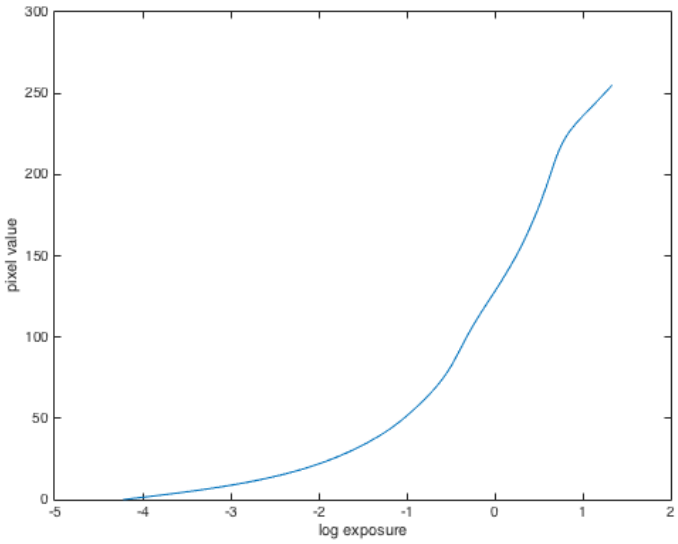


Figure 6.1: Graph of the retrieved characteristic function g().

The central part is the linear region where the curve is correctly defined. The top right part where non-linearities occur is the shoulder. And the left bottom part of the curve is the toe.

But this is where the limitations of simulating data is encountered. Since all images in the generated data set come from an image that has been truncated of its pixels (see section 3). It is not realistic nor correct to try to compute this characteristic curve on generated data. It has no real meaning due to the artificial procedure. This has been demonstrated on real images.

The characteristic curve is intrinsic to the camera itself therefore this determination of the characteristic curve can be performed before the device is in actual use. It is not an operation that has to be performed in real time during the image captures as the characteristic curve is supposedly constant over time. But the correction has to be applied to correctly calibrate images rather than a simple multiplication between images.

For the rest of the tests, a multiplication factor between images will serve as a sufficient calibration. For instance, if images with an exposure time of respectively $32ms$ and $64ms$ have to be composed, the intensity levels of the $32ms$ image will be doubled. But it is important to keep in mind that a more advanced calibration method can be necessary and is highly dependent on the sensor used. With the lack of information about any sensor and to stay generic, the calibration will remain simple.

6.2 Pixels weighting

Images are now aligned and calibrated. The last step is to compose the images into a single one. For N images, a simple idea would be to compute the average value for each pixels over all images [3]

$$C = \frac{1}{N}\sum_{i=1}^{N} I_i \tag{6.5}$$

But this method is not suited at all. Overexposed and underxposed pixels carry no information and will alter the data. Also as the occulter moves, the values can not be added or else it will create a ghost effect. Finally, not all pixels are of the same quality, pixels far from the center, especially at low exposure, are carrying noise. Pixels belonging to the corona and correctly exposed however are more important.

One way to solve this problem is to attribute weights to the pixels[3][1]. Before proceeding to the composition of the images.

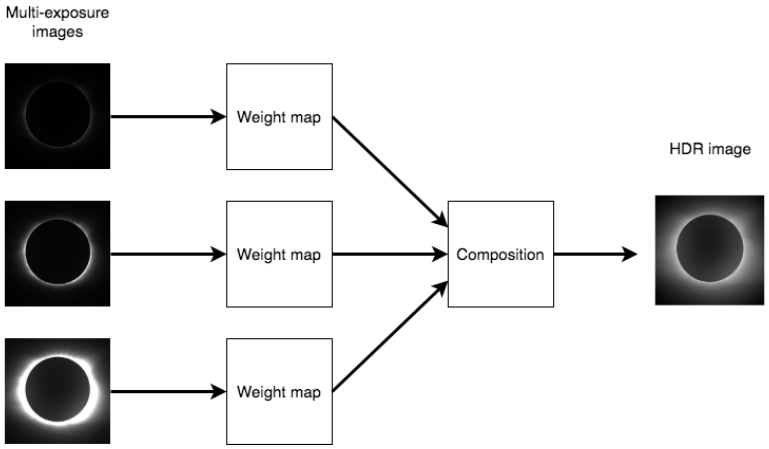


Figure 6.2: Schematic view of the image composition procedure: Each pixel from the multi-exposure images are weighted before the composition.

The composite image C will be such that for each pixel (i, j), k being the index representing the exposure of the N images[3][1].

$$C(i,j) = \frac{\sum_{k=1}^{N} I(i,j)_k w(i,j)_k}{\sum_{k=1}^{N} w(i,j)_k} \tag{6.6}$$

It is the weighted sum of the pixels of all images. Giving more importance to pixels with a high weight.

In the case of a coronagraph, weight attribution can be summarized as follows:

$$w(i,j)_k = \begin{cases} 0 & \text{if} \quad I(i,j)_k \in \text{occulter} \\ f(I(i,j)_k) & \text{else} \end{cases} \tag{6.7}$$

0 weights are attributed to the pixels belonging to the occulter, this ensures that ghost effects are not present due to its movement. This can be easily done as the occulter detection has already been performed (see chapter 4). Therefore, weights equal 0 if

$$(i - x_{occ})^2 + (j - x_{occ})^2 < r_{occ}^2 \tag{6.8}$$

But the determination of the function $f()$ is not trivial. This problem still remains open: *Namely how to find such values of pixel weights which would minimize the noise in the resulting image.* [3]. No objective definite solution is given for the case of total solar eclipse.

If this problem is strictly applied to the observation of solar eclipses, it seems useful to attribute weights in a radially decreasing way and depending on the exposure time since intensity is decreasing radially. Using the geometry of the problem for solving. However, weight attribution according to geometrical shape is not reliable due to the presence of prominences in the corona. Coronal mass ejection can change drastically the image content and make the weight attribution wrong if based on geometry. It seems much more effective to use pixels values themselves to attribute weights. Small values of pixels are essentially noise and have to be less important in the image composition. And high values of pixels that are close to saturation will also face non-linearities.

The use of a triangular shaped weight curve gives most importance to the pixels that are at middle of the dynamic range of the image[1]. With the assumption that these pixels are the best to represent the data. It is indeed the case as shown in previous section, the central part of the values of pixels correspond at best to the linear region of the characteristic curve.

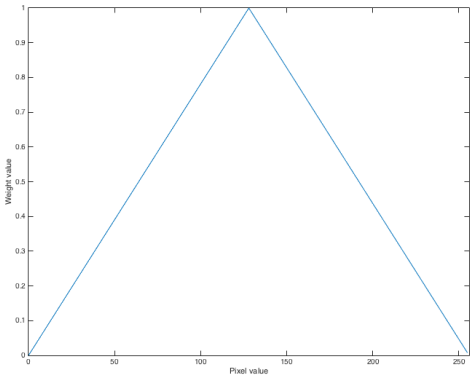


Figure 6.3: Triangular function for attributing weights to the pixels of an image. Pixels that are at the center of the dynamic range get the largest values. (Here for pixels between 0 to 255)

This method seems quite over-simplistic, It gives still a lot of importance to pixels that are far from the center of the curve (noise) and very few pixels value get a high weight (correct exposure). Therefore, the use of a gaussian function can give more flexibility. The standard deviation of the gaussian function will essentially give a range of pixels values that are considered as good. And higher and lower values of pixels are given lower weights (noise and non-linear region).

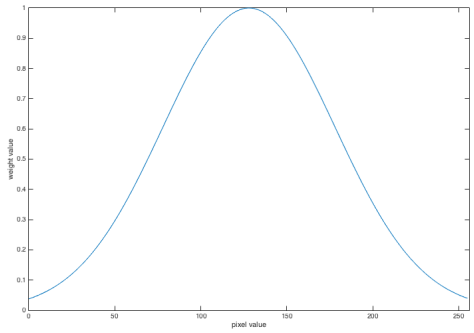


Figure 6.4: Gaussian function for attributing weights to the pixels of an image. Pixels that are at the center of the dynamic range get the largest values. (Here for pixels between 0 to 255) The standard deviation dictates the width of the 'correct' range of pixels values.

To further improve the quality of weight attribution - once the gaussian function has been applied on each pixels values to give the weight map - a gaussian blur can be applied. This avoids

having too much case specificity for each pixel weights. Instead, weights are spread over a region to indicate what 'region' of the image is interesting to retrieve.

The results of this method can be demonstrated here. Using 4 images of different exposure time correctly aligned with the method described previously, the procedure of weight attribution is performed.

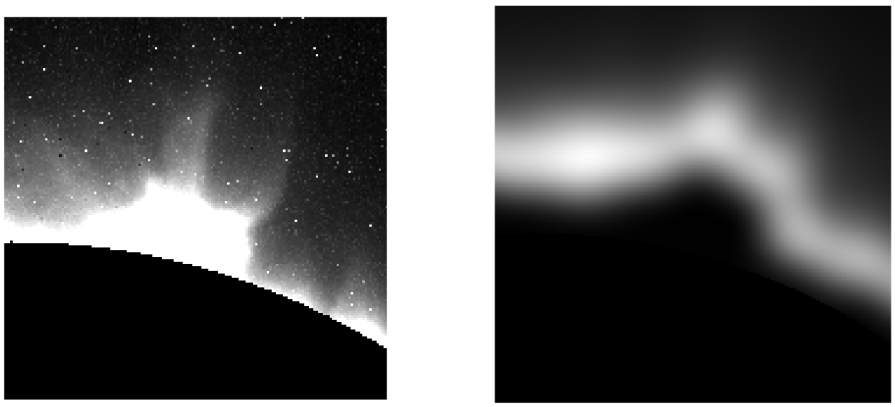


Figure 6.5: `Left:` Zoom on the image with the highest exposure time `Right:` Zoom on the weight map attributed to the image.

As can be seen, the occulter is given a zero value in the weight map. The high exposure image is highly saturated, all pixels in the saturated region are attributed a weight of 0 as they do not represent a correct data. The interesting part is thus in between the saturated region and before the noisy part (the more external part) of the image.

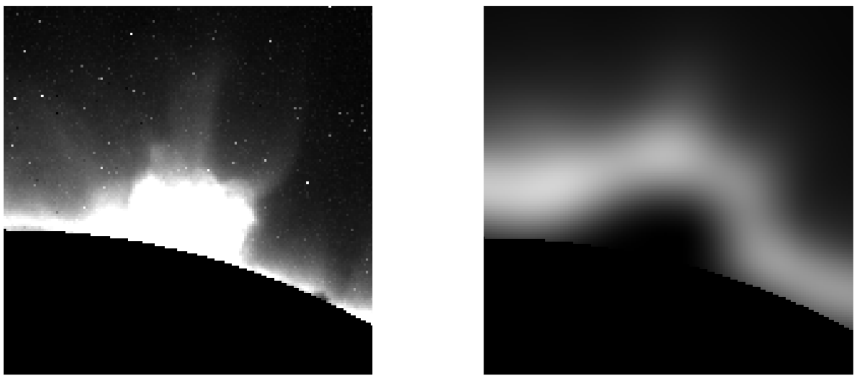


Figure 6.6: `Left:` Zoom on the image with a lower exposure time `Right:` Zoom on the weight map attributed to the image.

The results are quite similar to the previous, except in this case the saturated region is smaller as the exposure time is decreased.

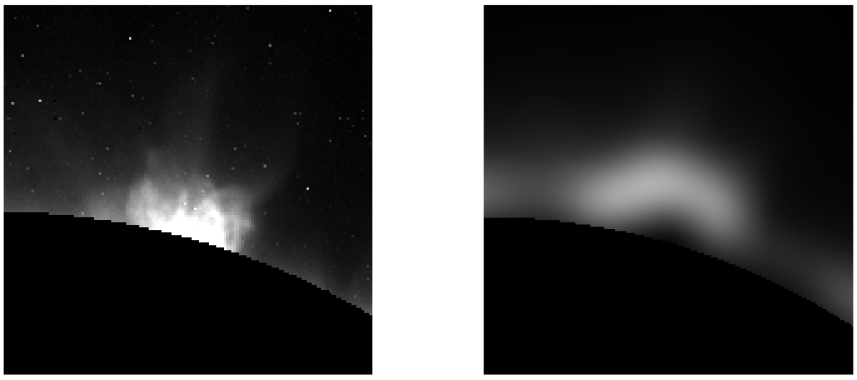


Figure 6.7: `Left:` Zoom on the image with a lower exposure time `Right:` Zoom on the weight map attributed to the image.

With an even lower exposure time image, details closer to the very bright part of the corona become visible.

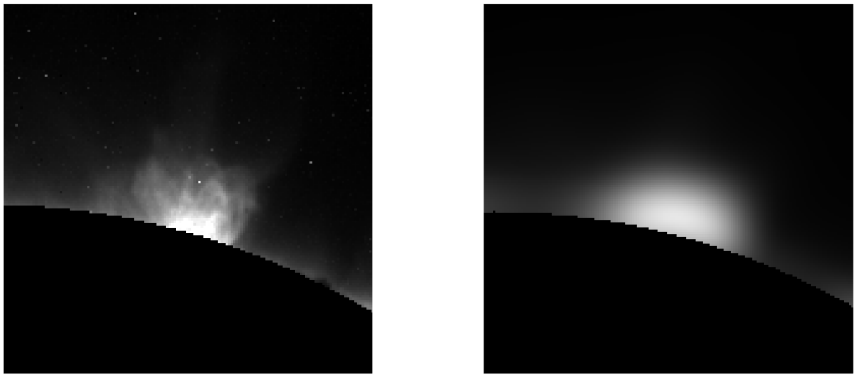


Figure 6.8: `Left:` Zoom on the image with a lower exposure time `Right:` Zoom on the weight map attributed to the image.

In this case, the brightest part of the corona is now visible. It can be seen that all images combined cover all regions of the corona and with correctly exposed values. This is the use of combining multiple images. The standard deviation of the gaussian blur has to be sufficiently high to make sure all parts are covered.

Finally, the composed image is created using equation 6.7. The result is an HDR image as can be seen on figure 6.9. But as already mentioned, it is not possible to see correctly the impact of the composition of the image. As the human eye and/or the display is not capable of seeing or displaying such contrasts ratios. A simple tone mapping can be performed by using a logarithmic operation (see figure 6.9). All the details of the corona are actually present in the composite image.

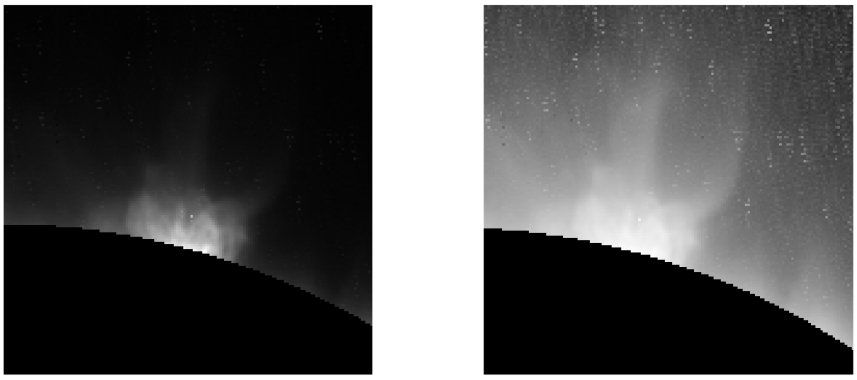


Figure 6.9: `Left:` Zoom on the composite HDR image `Right:` Zoom on the tone-mapped HDR image

Here is a view of the full image, the importance of tone-mapping is again shown in this case.

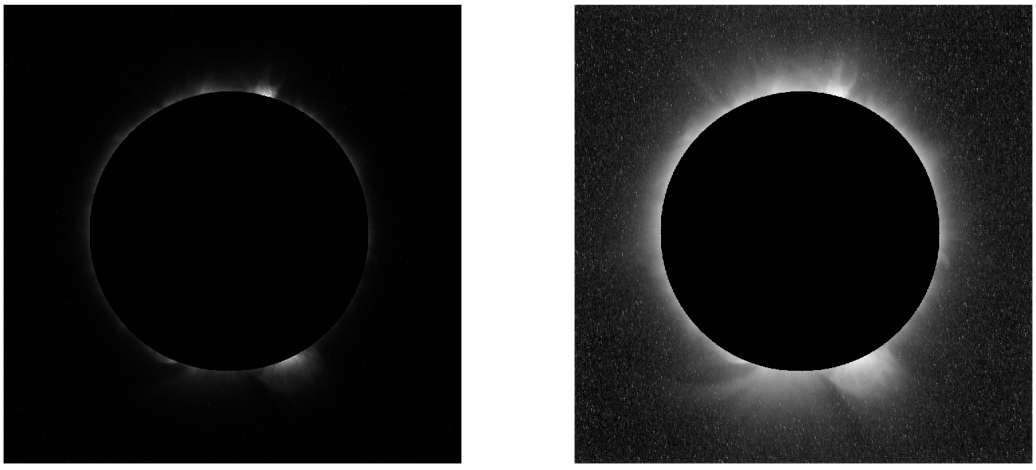


Figure 6.10: `Left:` Composite HDR image `Right:` Tone-mapped HDR image

Another way of understanding the role of the composition is to look at the histogram of the image. The histogram of the LDR images ranges to a maximal value of 511 as the images are stored with 9 bits/pixels. On the other hand, the composite HDR image has a large histogram containing information of all the LDR images. The maximal value is 2906, the HDR image is thus $log_2(2906) = 11.5 \simeq 12$ bits/pixels. The generated 9 bits/pixels images are derived from images of 12 bits/pixels. Therefore the intial dynamic range of the images is retrieved correctly using image composition.

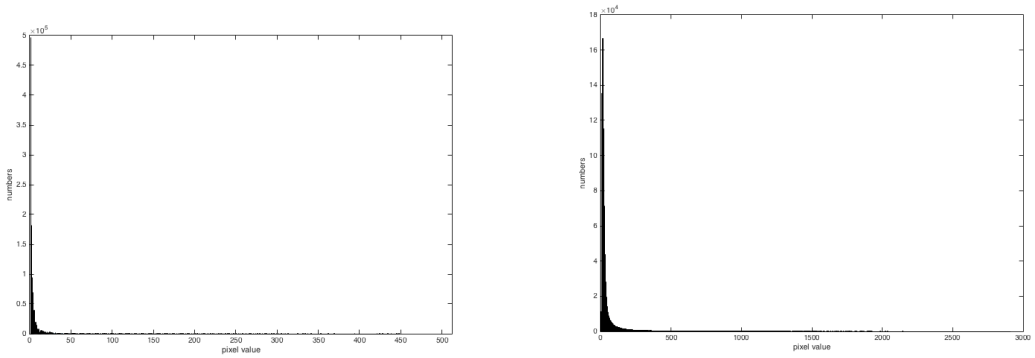


Figure 6.11: `Left:` Histogram of a LDR image. `Right:` Histogram of a HDR image.

The method can be applied also for the images of total solar eclipse. In this case, the correction for the calibration step with the characteristic curve can be performed[1]. The result is a high dynamic range image covering all parts of the corona. The image is tone-mapped to correctly see all the range that is performed with the composition. A comparison with the raw images in chapter 3 shows the utility of the method. Saturated regions of images are correctly discarded while keeping details in the inner part of the corona.

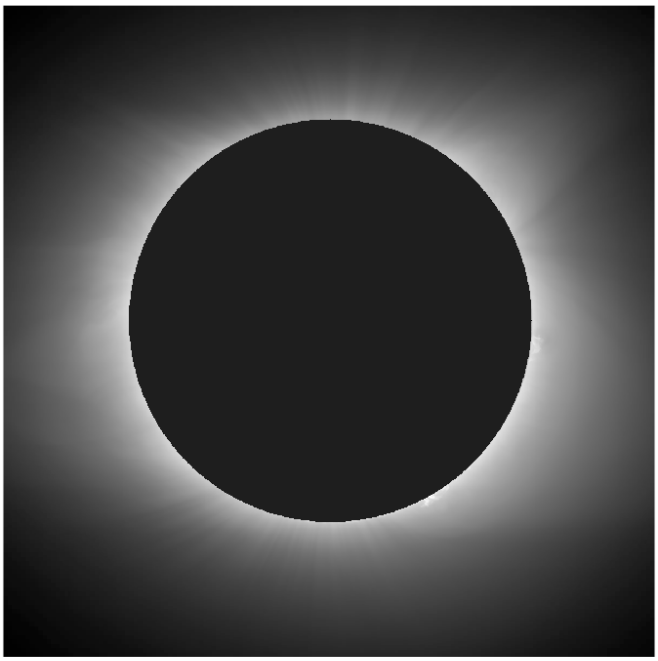


Figure 6.12: Tone-mapped HDR image of the total solar eclipse composed with 6 images of different exposure time.

6.3 Conclusion

The problem of image composition is relatively simple in its formalization. Yet the problem of weight attribution can be more subjective. The weight attribution method used in this case is specific due to the presence of the occulter. A gaussian function for weighting and a gaussian blur is performed on pixels values. As there is no easy objective way to assess the performances achieved in the composition of the HDR image, it is difficult to compare methods. However, this

can be performed with good results in a relatively simple way. The saturated regions are correctly discarded in the composite image and the correctly exposed pixels well preserved.

The non-linearities between time of exposure have to be taken into account for a more accurate HDR algorithm. But its application with generated data is not possible. The global goal has been achieved for image composition but a more advanced way of assessing the performances of the method is left open.

Chapter 7

Conclusion

An image alignment method has been successfully implemented with a pixel accuracy for multi-exposure images from a spatial coronagraph to correct for translation movements. The method is based on alignment of contours of intensity of the image. This particular technique is innovative as no similar methods are used for image registration. It can be applied to various data sets but with the only drawback that an initial value for the isovalue lines has to be selected.

The detection of the occulter is performed successfully with a pixel accuracy. The algorithm runs with a reasonable amount of computation time using the specificity of the problem. It is able to recover incomplete circles and is robust against changes of time of exposure.

An image composition method has been tested and implemented. The result is an HDR image composed from multi-exposure images. Which allows scene of high contrast to be taken with a camera of lower dynamic range.

Initial goals have been reached and a successful implementation has been made for a complete HDR algorithm. The data used for testing was mainly simulated data which makes the results questionable. But choices and tests have been made very critically during the research to make it as reliable as possible.

The next step of the work is an implementation on an embedded device Where many optimisations are possible for an efficient calculation. But this part is out of the scope of this project. All tests and performance evaluation have been made using `matlab` and `Python` for achieving the results.

7.1 Complete algorithm

The following is a block diagram summarizing the complete procedure described in the previous sections.

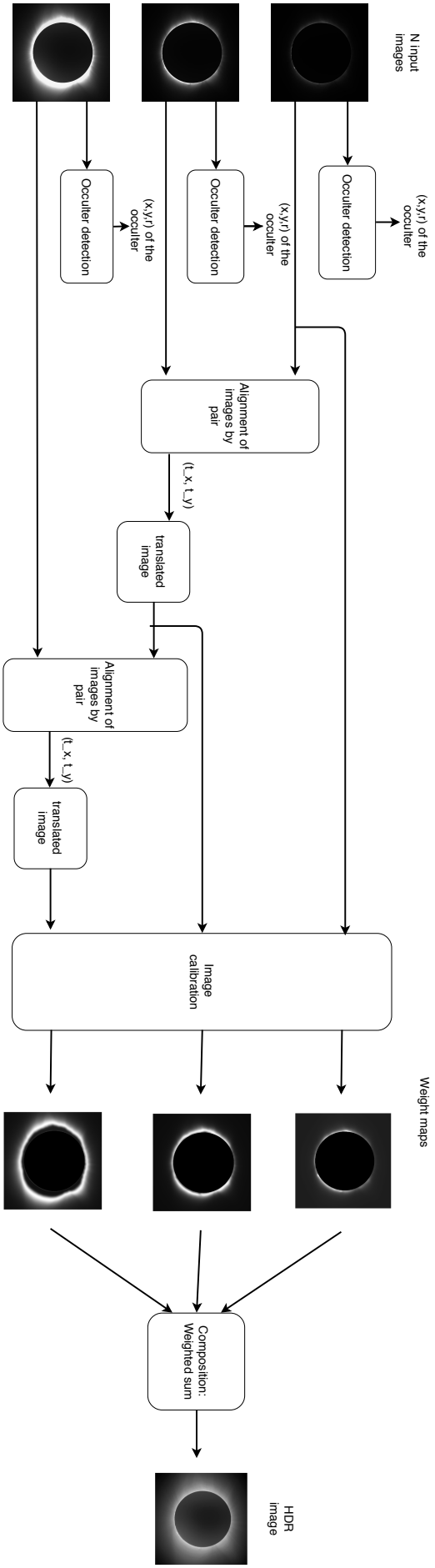

N input images
Occulter detection
(x,y,r) of the occulter
Occulter detection
(x,y,r) of the occulter
Occulter detection
(x,y,r) of the occulter
Alignment of images by pair
(t_x, t_y)
translated image
Alignment of images by pair
(t_x, t_y)
translated image
Image calibration
Weight maps
Composition: Weighted sum
HDR image

Figure 7.1:

7.2 Alignment

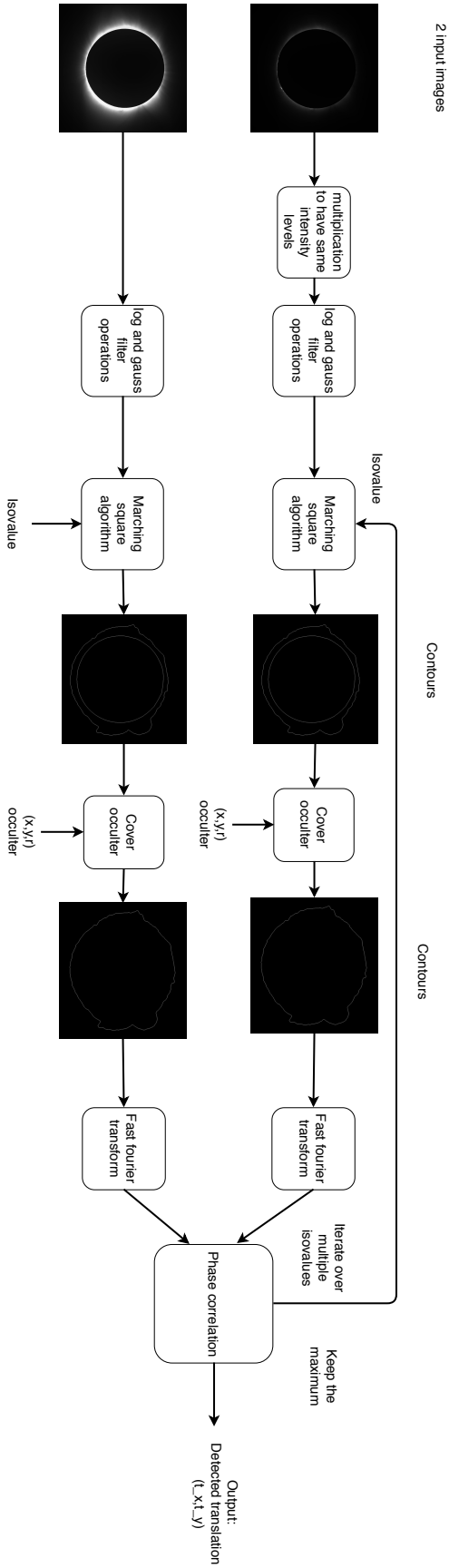


Figure 7.2:

7.3 occulter detection

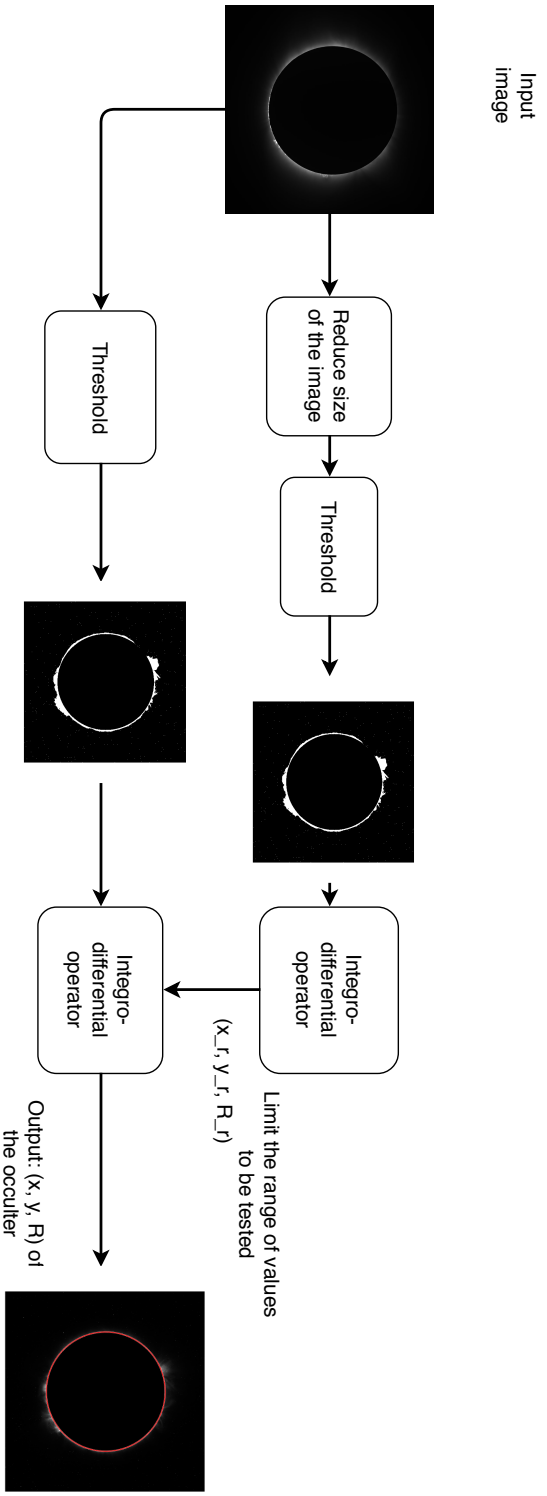


Figure 7.3:

Chapter 8

Annex

This chapter is dedicated to summarizing the internship made and the structure of the organization. It is non-technical and can be separated from the rest of the work.

8.1 Centre spatial de Liège

The Centre Spatial de Liège (CSL) is an applied Research Center owned by the University of Liège, focused on design, integration and calibration of space observation instruments[9]. It is located in the 'Scientific park' of Angleur. It employs about 100 employees.

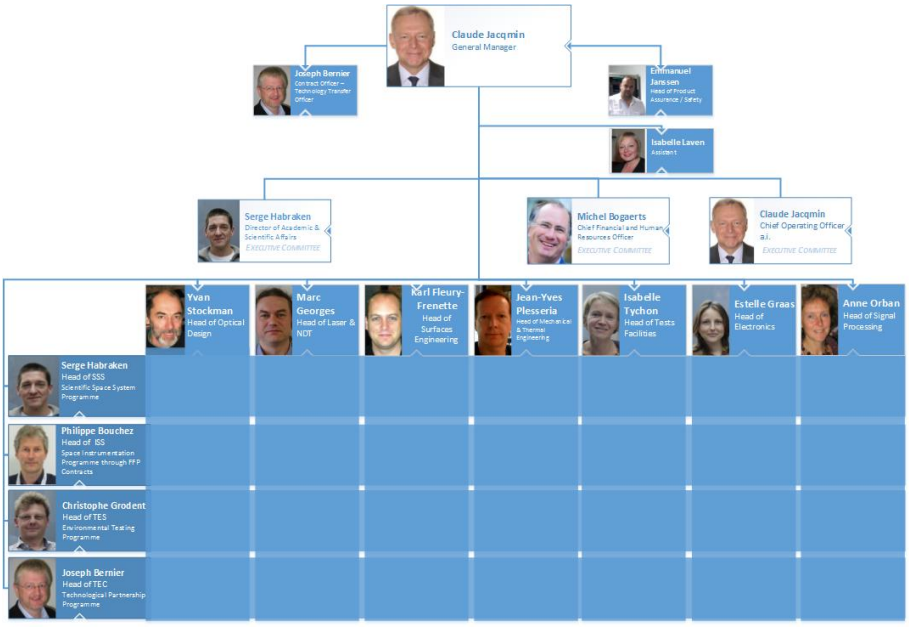
CSL owns state of the art test facilities for testing extreme conditions. Instruments, parts, optical systems or satellites can be tested by reproducing conditions encountered by a spatial device. Extreme Temperatures, pressures, vibrations, radiations can be reproduced in the facilities.
This has allowed CSL to be part of renown space missions.

CSL has several areas of expertise [9]: optical design and metrology, non destructive testing, surface micro-texturing, sensors and components working under extreme environment, lasers, cryogenics, thermal and mechanical design, signal processing...
Its highly skilled employees can work and collaborate on different projects: Regional, national, european research projects. Or private companies on very diverse areas.

8.2 Structure

CSL is structured as a matrix. There is a departmentalization based on the skills of the employees. Depending on the domain of expertise of the worker, it will be a (optics, electronics, mechanics, ...). For each department, a head of department is responsible of the laboratory. But there is also a structuration around the output of the work. Depending on the client, th. Employees are attributed to tasks inside

The horizontal division is quite low as employees can change the project depending on their personal preferences. There is also an interaction between department as they can work on an identical goal which reduces horizontal division and collaboration inside the organization. Vertical division is also low - at least in the electronics laboratory where the internship was made - as the technical core works actively on projects and on its development together with higher levels of qualifications.



8.3 Internship

The internship made lasted 80 days inside the organization. Inside the laboratory of electronics, the internship was 100% dedicated to R&D. The work performed was a precursory research of a new technology in hope of integrating it in later applications. It was started from scratch and without any expertise from the laboratory. The research was really in the spirit of academic research for which innovation, out-of-the box thinking and interdisciplinary.

A lot of experience was acquired during this period and a positive outcome as a first experience inside a professional structure.

Bibliography

[1] Jitendra Malik Paul Debevec. "Recovering High Dynamic Range Radiance Maps from Photographs". In: (1997).

[2] J. Daugman. "How Iris Recognition Works". en. In: *IEEE Transactions on Circuits and Systems for Video Technology* 14.1 (Jan. 2004), pp. 21–30. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2003.818350. URL: http://ieeexplore.ieee.org/document/1262028/ (visited on 01/04/2017).

[3] M. Druckmüller, V. Rusin, and M. Minarovjech. "A new numerical method of total solar eclipse photography processing". In: (Aug. 2006).

[4] M. Druckmüller. "Phase correlation method for the alignment of total solar eclipse images". In: (Nov. 2009).

[5] A. Ardeshir Goshtasby. *Image registration : principles, tools and methods.* London : Springer, July 2012. ISBN: 1-4471-2458-8; 9786613574206; 1-280-39628-8.

[6] Marc Van Droogenbroeck and Philippe Latour. "Computer Vision". In: (2017), pp. 437–518.

[7] URL: https://phys.org/news/2016-09-proba-3set-verge-sun.html.

[8] URL: https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm.

[9] *CSL*. URL: http://www.csl.uliege.be/jcms/c_5414/en/about-us.

[10] *Gaussian filter*. URL: https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm.

[11] Dirk-Jan Kroon. *Isocontour*. URL: https://nl.mathworks.com/matlabcentral/fileexchange/30525-isocontour.

[12] Jean-Marc Lecleire. *Solar eclipse 2017*. URL: https://www.hdrsoft.com/resources/hdr-for-solar-eclipse.html.

[13] Kobi Nistel. *Circle Detection Using Hough Transforms*. URL: https://nl.mathworks.com/matlabcentral/fileexchange/35223-circle-detection-using-hough-transforms?focused=5234934&tab=function.

[14] Norikura Observatory. URL: http://solarwww.mtk.nao.ac.jp/en/norikura/nkr10cmnew.html.

[15] Anirudh Sivaraman. *Iris segmentation using Daugman's integrodifferential operator*. URL: https://nl.mathworks.com/matlabcentral/fileexchange/15652-iris-segmentation-using-daugman-s-integrodifferential-operator.

[16] SWAP. *SWAP is a project of the Centre Spatial de Liege and the Royal Observatory of Belgium funded by the Belgian Federal Science Policy Office (BELSPO)*. URL: http://proba2.oma.be/data/SWAP.

[17] Anna Tomaszewska and Radoslaw Mantiuk. "Image Registration for Multi-exposure High Dynamic Range Image Acquisition". In: ().

[18] Wikipedia. *Circle Hough transform*. URL: https://en.wikipedia.org/wiki/Circle_Hough_Transform.

[19] Wikipedia. *Contour line*. URL: https://en.wikipedia.org/wiki/Contour_line.

[20] Wikipedia. *Coronagraph*. URL: https://en.wikipedia.org/wiki/Coronagraph.

[21] Wikipedia. *Feature detection*. URL: https://en.wikipedia.org/wiki/Feature_detection_(computer_vision).

[22] Wikipedia. *High-dynamic-range_imaging*. URL: https://en.wikipedia.org/wiki/High-dynamic-range_imaging.

[23] Wikipedia. *Large Angle and Spectrometric Coronagraph*. URL: https://en.wikipedia.org/wiki/Large_Angle_and_Spectrometric_Coronagraph.

[24] Wikipedia. *Phase correlation*. URL: https://en.wikipedia.org/wiki/Phase_correlation.

[25] Wikipedia. *PROBA-3*. URL: https://en.wikipedia.org/wiki/PROBA-3.

[26] Wikipedia. *Shutter speed*. URL: https://en.wikipedia.org/wiki/Shutter_speed.

[27] Wikipedia. *Space Weather*. URL: https://en.wikipedia.org/wiki/Space_weather.