Université de Montréal

Reconstruction of the surface of the Sun from stereoscopic images

by

Vlad - Andrei Lazăr

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en informatique

Decémber, 2007

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé :

Reconstruction of the surface of the Sun from stereoscopic images

présenté par

Vlad - Andrei Lazăr

a été évalué par un jury composé des personnes suivantes:

Pierre Poulin
(Professeur)

Max Mignotte
(Professeur)

Sébastien Roy
(Directeur du recherche)

Mémoire accepté le _____

RÉSUMÉ

Cette thèse s'intéresse à la reconstruction stéréoscopique dans des environnements contenant des objets transparents, comme la couronne solaire. Les données pour ce projet, images stéréoscopiques du soleil, ont été fournies par la NASA grâce à la mission STEREO. Ce mémoire propose une nouvelle méthode de rectification sphérique ainsi qu'un nouvel algorithme pour la reconstruction dense sans aucune hypothèse préalable sur la forme ou la transparence des objets dans la scène.

Premièrement, les paramètres des caméras sont estimés, et une étape de raffinement suit pour obtenir un alignement presque parfait entre les images. Dans l'étape suivante, les images sont rectifiées pour réduire l'espace de recherche de trois à deux dimensions. Les densités le long des lignes épipolaires sont ensuite estimées.

La reconstruction des scènes transparentes est encore une problème ouvert et il n'y a pas de méthodes générales pour résoudre la transparence. Les applications pour cet algorithme sont nombreuses, comme la reconstruction des traces de fumée en soufflerie, le design optimal des chambres à combustion, la realité augmentée, etc.

Mots clés: vision par ordinateur, rectification, stéréoscopie, transparence, estimation de profondeurs multiples, soleil, physique solaire

ABSTRACT

This thesis concentrates on the stereoscopic reconstruction of environments containing transparent objects. The data used to test the algorithms is graciously provided by NASA through the STEREO mission. This thesis proposes a new spherical rectification technique as well as a dense reconstruction algorithm without making any prior assumptions on the shape or transparency of objects inside the scene.

Firstly, the camera parameters are estimated, following a refinement step to get seamless alignment between images. In the next step the images get rectified in order to be able to restrict the search space to 2D rather than the full 3D. Afterwards the density at along each epipolar line gets estimated.

The reconstruction of transparent scene is still largely an open problem and there are no general methods to deal with transparency. The applications of such an algorithm are numerous, ranging from reconstruction of smoke trails inside wind tunnels, optimal design of combustion chambers, augmented reality, etc.

Keywords: computer vision, rectification, transparency, stereoscopy, multiple depth estimation, Sun, solar physics

TABLE DES MATIÈRES

LIST OF FIGURES

Chapter 1

INTRODUCTION

The field of machine vision aims at developing algorithms that mimic functions of the human visual system. Using data from sensors (imaging, range scanners, etc.), the algorithms are trying to get information about the surrounding physical world. Each of the sensors observes merely just a "projection" of the real world so this information must be merged to recover the world coordinates. Out of the machine vision problems, the one that received most of the attention is 3D reconstruction. Applications are numerous, ranging from metrology, navigation and adaptive multimedia systems.

In this thesis we attempt to develop a reconstruction scheme for solar coronal loops using extreme ultraviolet images taken by the STEREO mission, while making just standard smoothness/sparsity assumptions. For the first time we have simultaneous satellite images from two vantage points using identical instruments. Previous attempts at reconstruction used single vantage point images spaced in time, using the solar rotation to provide different views of the features.

The STEREO mission will provide an important tool to validate the theoretical models of magnetic fields and plasma flows on the Sun. The holy grail of solar physics is the accurate prediction of the space weather, which has a strong influence on our day to day activities. The coronal loops have a major influence on this phenomena. The loops on the surface of the Sun sometimes erupt outside the corona and escape the Sun's gravity. This creates the aurora Borealis/Australis and disrupts satellites and radio communications. Prediction of such phenomena relies on accurate 3D models of such loops, which is the main concern of this thesis.

There are multiple approaches to the 3D reconstruction problem. The simplest of which, uses pixel matching techniques along epipolar lines and together with the projection model, one can triangulate the world 3D position of each pixel. In this method one chooses a reference view and the resulting reconstruction is from this point of view. An alternate, but similar method, is *volumetric reconstruction*. The reconstruction volume gets discretized into volume elements, and the value at each voxel is dictated by an average of the pixel values from all views where the voxel is visible. This can accommodate an arbitrary number of views. Usually a voxel is either fully transparent or opaque leading to a single depth for a pixel inside the images. The success of this method is strongly influenced by our occlusion/visibility modelling.

Another family of methods, used commonly in medical imaging, is the *tomographic reconstruction*. Given a large number of projections of the object one can reconstruct the object with low error. Normally we will settle for a few hundred projections in order to obtain good results. This method used certain properties of the Fourier transform of the projections to perform the reconstruction. Usually an orthographic projection model is assumed.

The current algorithms cannot reconstruct reliably transparent environments unless an unreasonable number of input images is used or an a priori knowledge of the shape of objects is available. We will have to cope with as little as two or three images (if we use SOHO images as well). The solar loops are short lived phenomena, thus preventing us from using images taken at different instances of time.

The method proposed in this thesis is a hybrid between the volumetric and tomographic reconstructions: like the tomographic reconstruction we are looking for a certain "matter density" inside each voxel, but the original rectified images are used directly rather than the Fourier transform of its projection. The problem poses itself as a constrained minimization problem. The constraints are provided by the available views together with the corresponding projection models. The function to be

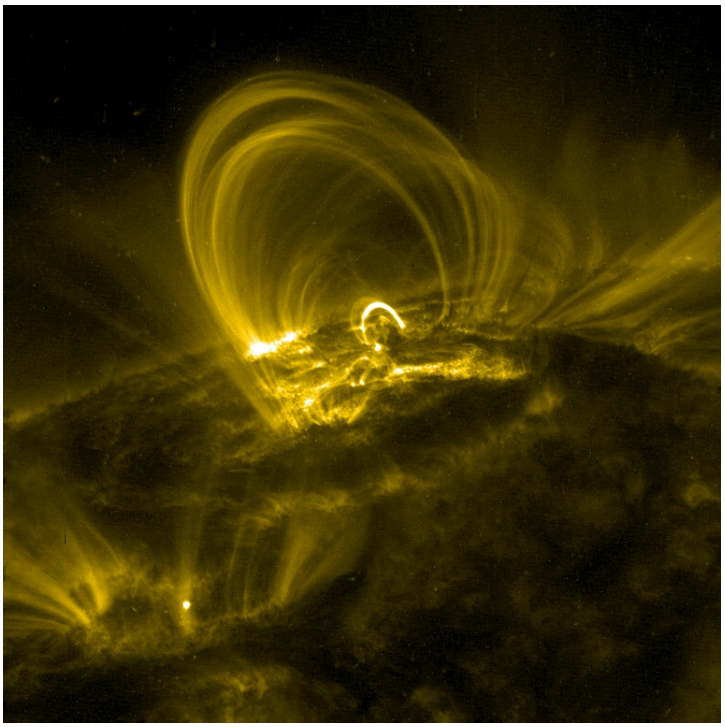

Figure 1.1. Coronal loops captured by the TRACE mission $284\mathring{A}$

minimized, provides some kind of regularization, helping us to impose certain properties of the solution. The problem is massively underconstrained: given a uniform discretization of n in each dimension, our reconstruction volume has n^3 cells (voxels) with $O(n^2)$ equations given by the views). This algorithm will be applied on solar coronal loops captured by STEREO (Fig. 1.1). The problem of transparent stereo matching is extremely challenging and there exists no current solution which is satisfactory. Because of this the results presented here are far from perfect.

A secondary contribution of this project is a rectification scheme named *spherical rectification*, which has all the good properties of state of the art rectification methods such as the ability to rectify any camera configuration outputting a finite image size, but is particularly useful for objects which are on spheres.

1.1 Outline

The thesis is organized as follows: in chapter 2 we present a brief history of solar observation, some current open research topics and a bit of physical background that will be useful in the later parts. In chapter 3 we introduce the standard 3D reconstruction toolkit. Chapter 4 introduces the fundamentals of rectification as well as some small results of our own rectification method. In chapter 5 we introduce the standard methods to reconstruct transparent environments and our proposed method. Chapter 6 presents some results of our reconstruction with both synthetic and read data, and in chapter 7 we suggest some future improvements.

Chapter 2

ASTRONOMICAL AND SOLAR IMAGING

The Sun has been a source of fascination for mankind before the dawn of history. Numerous historical discoveries stand witness that prehistoric people had basic knowledge of solar system planetary cycles.

It was not until around the year 1600 that the first Earthbound solar telescope was built by Galileo Galilee. He was the first to observe the solar dark spots. During the 19^{th} century, the German astronomer Heinrich Schwabe observed that the number of spots increases and decreases with time. He was the first to observe that the period of this solar activity oscillation is about 11 years.

Probably the greatest contribution to solar observations was brought by George Ellery Hale in the 20^{th} century. He discovered that the sunspots were cooler than the surrounding matter, and thus darker (the magnetic field inside the sunspots is strong enough to prevent convection, so hot matter from the inner Sun cannot reach the surface). Another important contribution was the observation that every 11 years the solar magnetic poles get reversed, giving birth to a more fundamental solar cycle of 22 years[2].

Since the beginning of the space age, the knowledge about the Sun has increased exponentially. This was powered by both recent theoretical physics and technological developments. Using airborne/spaceborne observatories has improved the quality of data by removing the effects of the atmosphere that could corrupt the data. Observations of certain wavelengths, such as X rays, are impossible inside the Earth's atmosphere because of its high absorption rate.

The motivation for the special interest in the Sun is fairly straightforward: it is our only source of high resolution data of the physical processes inside stars. The

activity on the Sun has a strong influence on our day to day activities as well, giving us more pragmatical reasons for its study. High energy particles ejected by the Sun into outer space - the solar wind - change on a global scale the Earth's climate, the most visible effect being *Aurora Borealis/Australis*. Other bad effects include disruption of geostationary satellites, pipelines, electrical power grids and increased levels of radiation. The generation of solar wind follows an extremely complicated mechanism, not entirely known.

The Sun provides a lot of information about processes that are not easily replicated by man made experiments. In elementary particle and nuclear physics the benefits were numerous. With the help of solar data, about 30 years ago the *neutrinos* were discovered. Up until the year 2002 there was a major discrepancy between the predicted and observed neutrino amounts. Finally two new types of neutrinos have been discovered (with much lower probability of interaction).

The bulk part of the solar energy is generated thorough the *CNO cycle* (Carbon, Nitrogen, Oxygen), in which stars convert through fusion Hydrogen into Helium, a phenomenon which is still not totally understood.

In the field of plasma physics the most important contributions were wave propagation and magnetic field generation.

One of the largely open problems is *the coronal heating problem*. The solar corona is the outer most atmosphere. This extends from R_{sun} to about $2-3$ solar radii. The mystery behind the corona pertains to its heating mechanism. It is about 200 times hotter than the *photosphere* - the next inner layer. The temperature of the corona rises from $5\ 000^{\circ}K$ to about $1\ 000\ 000^{\circ}K$ within 200 000 Km. There is still no generally accepted theory regarding the energy transfer mechanisms from the photosphere to the corona. The two most prevalent theories are the *wave transport theory* and the *magnetic reconnection theory*. The second one has the greater support and we will base our investigations on it. In short this theory claims that the heating is due to the magnetically induced electrical currents. When magnetic fields change

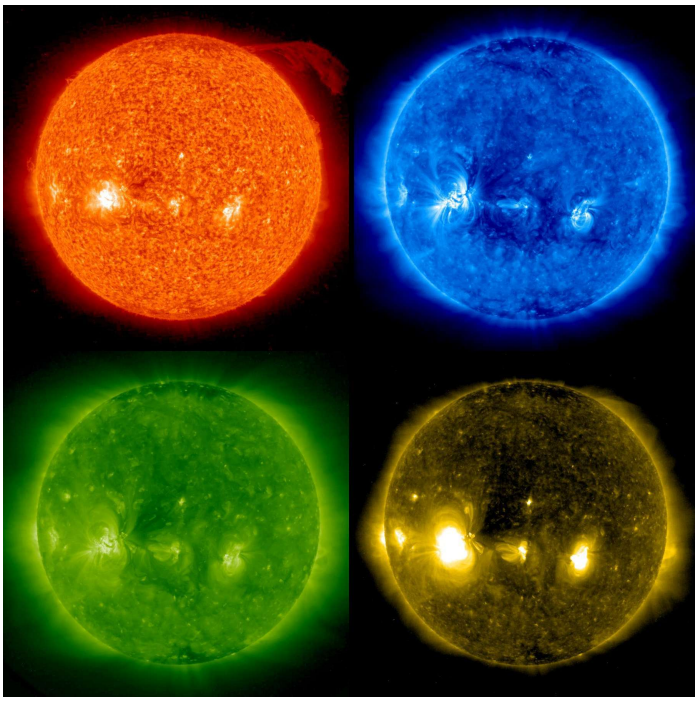

Figure 2.1. The 4 wavelengths captured by STEREO

topology (they merge or divide) a certain amount of energy gets released. In our project we will try reconstructing these field lines.

Because of the extreme temperature most of the matter is ionized. This is fortunate since this matter will gather around the magnetic field. The equation of motion for a charged particle inside magnetic field is given by the Lorentz equation:

$$\overrightarrow{F} = q \cdot \overrightarrow{V} \times \overrightarrow{B} \tag{2.1}$$

where q is the particle charge, V is its velocity and B is the magnetic field. Since there is a cross product, the particle will follow a helical motion around the field line. These particles provide an outline of the magnetic field, otherwise invisible (Fig 1.1).

For more detailed information on solar and stellar phenomena please refer to [3–6].

2.1 STEREO mission

In December 2006, NASA launched its third Solar Terrestrial Probe called STEREO (Solar TErrestrial RElations Observatory). The mission consists of two identical probes orbiting around the Sun, one in front and the other trailing behind the Earth, providing the first true stereoscopic view of the Sun.

The whole mission was designed to provide data for a period of 5 years with its main scientific objective being the better understanding of CMEs (Coronal Mass Ejections). CMEs are important to study since they have a direct impact on our day to day life. Once they escape the solar gravitational field they turn into solar wind and can disrupt satellites orbiting Earth, telecommunications, and even the terrestrial electrical power grids.

The mission carries a broad range of instruments. This project will be using the instruments contained in the SECCHI package (Sun Earth Connection Coronal and Heliospheric Investigation). Each satellite contains a EUV (extreme ultra violet) imager that takes images in the wavelengths of 171, 195, 284 and 304 Å. Since different emission lines get formed at different temperatures, different images provide insight at different depths inside the Sun (Fig. 2.1), ranging from R_{sun}, up to approximately $2R_{sun}$.

The satellites orbit in a heliocentric trajectory (around the Sun), allowing the satellites to separate more and more as time passes, since one is closer to the Sun and thus moving faster. The current separation between the satellites is about 25° and growing by a rate of about 6° per month. The satellites are situated about 10^9 meters away from the Sun. The field of view of the satellites is around 1.5°, so the projection model being close to an orthographic camera model.

The data comes in the FITS format. This is a general purpose format used in Solar and stellar astronomy, that can handle time series, images, or multidimensional data. The FITS files also contain a header where one can accomodate ancillary information

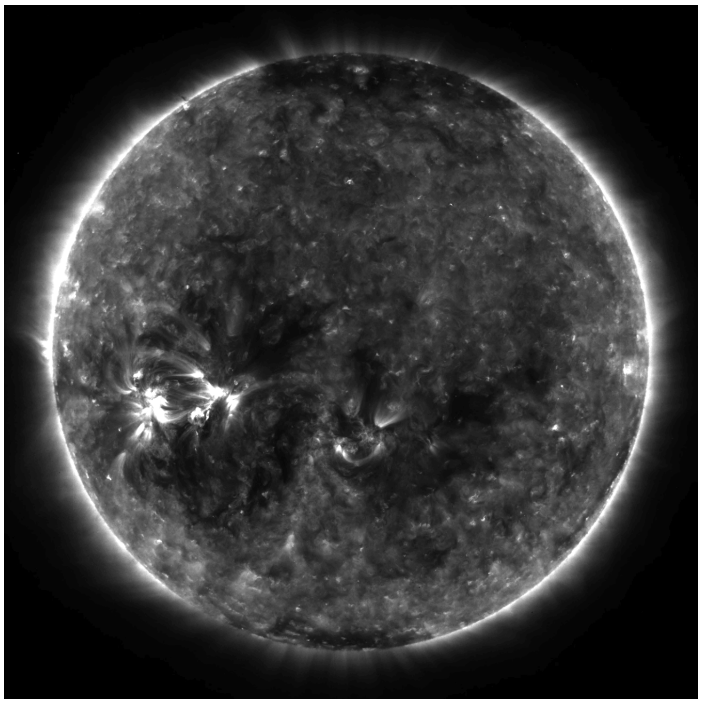

Figure 2.2. Image of the Sun as seen by STEREO-B in the $195\mathring{A}$

about the conditions in which the data was recorded. SECCHI provides its data as 16bit integer 2D images, (Fig. 2.2).

A more detailed mission description can be found in [7, 8].

2.2 Coordinate systems

In order to represent the positions of far astronomical bodies, they are considered as belonging to a sphere of infinite radius - *the celestial sphere*. In such a system, the parallax is virtually zero. The position of objects in such a system is fully determined by two angle parameters, the *right ascension* and *declination* (or galactic latitude and longitude). This sphere has its center located at the center of the Earth and its equator in the same plane as Earth's equator - the *celestial equator*. In a similar fashion coordinates on the surface of Earth are represented by two coordinates, latitude

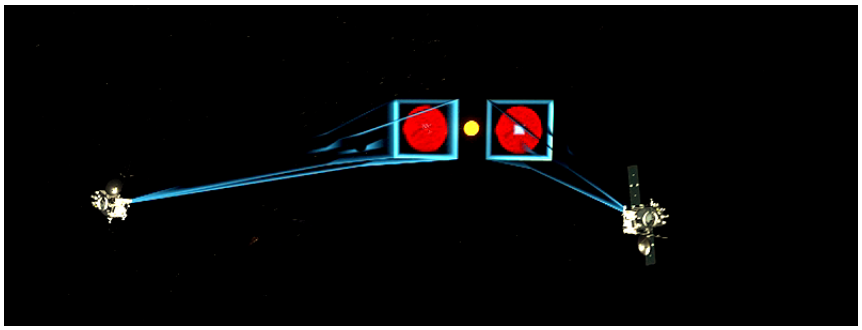

Figure 2.3. Sun seen from the two STEREOs

and longitude.

Since the Sun is close enough and the resolution of the observations permits us to resolve smaller features, it is crucial to introduce a third coordinate to accurately describe the phenomena occurring on the Sun. As we will see in the chapter about camera models (chapter 3), the third coordinate gets lost due to projection onto the imaging sensor. Because of this, at least two views are needed to recover the whole 3D geometry of phenomena.

Another difficulty in positioning objects onto the Sun is caused by the fact that there are no stationary points that could serve as reference. The Sun turns at different rates at different latitudes because of centrifugal and magnetic forces. Some coordinate systems will be rotating with respect to each other, thus it is necessary to take also time into consideration.

2.2.1 World coordinate systems

Since the STEREO is observing from two very different vantage points it is necessary to incorporate the instrument viewpoint (3D position) into the coordinate system (Fig. 2.3).

To be able to pass from 3D world coordinates to pixel coordinates inside images we need to pass through two levels of coordinate systems. Firstly, the 3D positions

and orientations of the satellites have to be known (in total 6 parameters, 3 for translations and 3 for rotations). These are the "external" parameters. After this we have to establish a set of 2D transformations that map the coordinates of the Sun to pixel coordinates of the sensors (the "internal" parameters).

In order to represent the 3D world position of the satellites, the FITS headers provide coordinates in a multitude of coordinate systems. To uniquely define a coordinate system we have to pinpoint its origin as well as choose two axes (the third is derived from these axes since we assume a right handed coordinate system). We use heliocentric coordinate systems, so the origin is at the center of the Sun. The most useful orientations of the axes are:

1. Heliocentric Aries Ecliptic
    - X axis points towards the First Point of Aries
    - Z axis points towards the ecliptic north pole
2. Heliocentric Earth ecliptic
    - X axis points towards the Earth
    - Z axis points towards the ecliptic north pole

The *ecliptic plane* is the plane in which the Earth rotates around the Sun. The *ecliptic north pole* direction is perpendicular to the ecliptic plane.

The first *point of Aries*, Fig 2.4, is the point in space where galactic longitude is considered 0. This is one of the points where the celestial (Earth's) equator plane intersects the ecliptic plane. Whenever the Sun is in one of these two points, an equinox occurs. The first point of Aries has been chosen as the *vernal equinox* that occurred in 1950. This points towards somewhere in the *Pisces* constellation. The first point of Aries moves at a constant rate of about one degree every 71 years. This

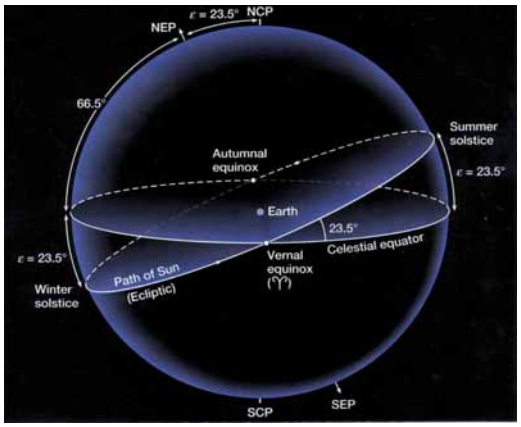

Figure 2.4. Celestial and ecliptic planes together with the equinoxes

movement is small enough to be considered constant considering the typical timescale of observed solar phenomena, which does not usually go over 6 months.

Both these coordinate systems have the origin at the center of the Sun, thus the name - heliocentric. These coordinate systems are used to represent the 3D position of the satellites. The satellites are designed to look towards the center of the Sun, making the remaining 3 rotation parameters known. Details on how to compute them will be given in chapter 4.

2.2.2 Image coordinate systems

Next we have to deal with conversion from 3D world coordinates to 2D pixel coordinates. This is accomplished by using the helioprojective system. Even though they are full 3D systems, they are not very useful to express real 3D world point as they are mostly tied to the Sun, thus changing fairly rapidly with time. Another characteristic that makes then unsuitable for this task is the fact that the observer point of view is not included in the system, making it impossible to compare two images taken from two different locations.

While not being used in practice, the helioprojective cartesian coordinates pro-

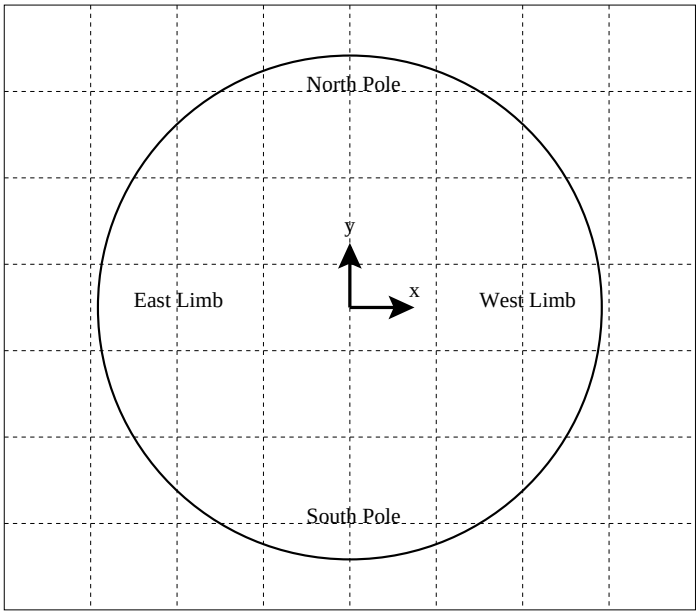

Figure 2.5. Heliocentric cartesian coordinate system

vide the background necessary for the other coordinate systems (Fig. 2.5). In this coordinate system the Z-axis is defined as the observer-Sun line pointing towards the observer. The X-axis is defined perpendicular to the plane defined by Z and the Solar North pole (point around which the Sun rotates). The Y-axis is defined as the cross product between the other two.

2.2.3 Helioprojective coordinate systems

Stars are usually considered far, flat, virtually positioned at infinity. This is not the case for the Sun, therefore we need a more specialized (accurate) coordinate system to express positions on the surface of the Sun (a sphere).

These coordinate systems mimic the heliocentric coordinates with the difference that their distances are replaced by angles. The origin of this coordinate system is located at the Sun's center. The Y-axis points towards the solar North pole and the X-axis towards the west solar limb. The solar north/south poles direction is defined

similarily to Earth as the direction perpendicular to the plane of solar rotation. We could define the Z-axis to be the vector product between Y and X, giving us a **left handed coordinate system**. In practice the third coordinate is fairly useless. The conversion between heliocentric Earth equatorial and helioprojective cartesian coordinates is one to one:

$$x \approx D\left(\frac{\pi}{180^\circ}\right)\phi_x \tag{2.2}$$

$$y \approx D\left(\frac{\pi}{180^\circ}\right)\phi_y \tag{2.3}$$

$$\tag{2.4}$$

where x and y are the heliocentric coordinates, ϕ_x and ϕ_y are the two helioprojective coordinates, D is the distance from the observer to solar center. The system assumes implicitly the observation is carried out from Earth. This system is nothing more than a spherical coordinate system analogous to one on the Earth. The system can be extended by adding the 3^{rd} coordinate $\xi \equiv D - d$, where d is the distance between the feature and the observer. In the vicinity of the Sun we can consider that $\xi \approx z$.

In practice in order to convert from pixel coordinates to helioprojective system and other way around, we need three extra parameters: the center of the Sun in pixels, a rotation around the satellites Z-axis (the yaw angle) needed to bring the solar north to the top of the picture, and a scale, the number of degrees/pixel.

The only place where the helioprojective coordinate system is used is in solar observations. The astronomers prefer most of the time to replace the true angles by some pseudoangles. The pseudoangles are defined as the projection of a feature onto the $z = 0$ plane expressed in angles. The pseudoangles vary with the tangent of the real angle. Since the apparent angular size of the Sun, from Earth is around 1°, the pseudo and true angles differ only at the fifth decimal place.

This approximation is also used when one is observing a spherical surface with a flat sensor and is called the TAN projection model.

More informations about common coordinate systems used in astrophysics can be

found in [9].

2.3 Influence of magnetic field

The magnetic field is of paramount importance for both theoretical understanding of and data processing. Once we have a model of the magnetic field which is simple enough, we could use it to help us identify features inside the images provided by STEREO.

The full dynamics of matter under magnetic and electric fields is described by a system of 8 coupled partial differential equations called the magneto-hydrodynamic equations (MHD). Since these equations are fairly hard to resolve, an acceptable subset of equations chosen to model the magnetic fields in the corona are the 4 Maxwell equations (the hydrodynamics is considered negligible as the density inside the corona is minimal):

$$\nabla \cdot E = 4\pi\rho_E \tag{2.5}$$

$$\nabla \cdot B = 0 \tag{2.6}$$

$$\nabla \times E = \frac{1}{c}\frac{\partial B}{\partial t} \tag{2.7}$$

$$\nabla \times B = \frac{1}{c}\frac{\partial E}{\partial t} + 4\pi j \tag{2.8}$$

where E and B are the electric and magnetic fields, ρ_E is the electric charge density, c is the speed of light and j is the electric current density. We can introduce further simplifications. Since we consider the fields as being in equilibrium, the time derivative terms are negligible.

If we consider that the magnetic field is a potential field, it can be written in terms of gradient of another field $B = \nabla\phi$. We get the potential field approximation of the field: $\nabla \times \nabla \cdot \phi = 0$ where $B = \nabla \cdot \phi$, $\nabla^2 \cdot B = 0$. Standard methods on how to solve such equations are described in [10]. The solution to the potential field approximation of the problem is the lowest energy configuration possible. This approximation holds

only inside regions on the Sun where activity is very low [11].

For regions with stronger activity, the model of choice is the linear force free model. The equation of this model is

$$\nabla \times B = \alpha B \tag{2.9}$$

With some further approximations this becomes $\nabla^2 \cdot B + \alpha^2 B = 0$, known as the Helmholtz equation. The parameter α can give us a measure of how unstable the region is (likelihood of a solar flare for example). For $\alpha = 0$ we are back to our potential field model.

The widely available magnetic data that is available from the MDI mission (Michelson-Doppler Interferometer) provides us with just the normal component of the magnetic fields on the surface of the Sun. Note however that the magnetic field is a vector function $B = (B_x, B_y, B_z)$ each component depending on (x, y, z). The data available from MDI is $B_z(x, y, R_{Sun})$. We need to propagate the information we have throughout the whole volume of interest (extrapolate the field) in order to use it at a later stage. In Fig 2.6 we have an example of a magnetogram provided by MDI. Red patches represent fields that exit the surface of the Sun and green patches where fields enter the Sun.

Fourier space methods recently developed in [12–14] provide very efficient ways to extrapolate linear force free magnetic fields. It can be shown that the solution of the Helmholtz equation can be expressed in terms of the Fourier transform of the normal

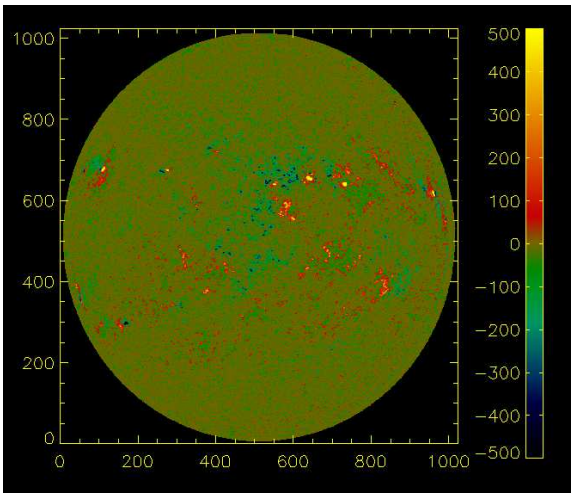

Figure 2.6. Magnetogram provided by MDI

component:

$$B_x(x,y,z) = \sum_{m,n=1}^{\infty} \frac{C_{mn}}{\lambda_{mn}} \exp(-r_{mn}z) \cdot \left[\alpha \frac{\pi n}{L_y} \sin\left(\frac{\pi m x}{L_x}\right) \cos\left(\frac{\pi n y}{L_y}\right) - r_{mn} \frac{\pi m}{L_x} \sin\left(\frac{\pi n y}{L_y}\right) \cos\left(\frac{\pi m x}{L_x}\right)\right] \quad (2.10)$$

$$B_y(x,y,z) = \sum_{m,n=1}^{\infty} \frac{C_{mn}}{\lambda_{mn}} \exp(-r_{mn}z) \cdot \left[\alpha \frac{\pi n}{L_y} \cos\left(\frac{\pi m x}{L_x}\right) \sin\left(\frac{\pi n y}{L_y}\right) - r_{mn} \frac{\pi m}{L_x} \cos\left(\frac{\pi n y}{L_y}\right) \sin\left(\frac{\pi m x}{L_x}\right)\right] \quad (2.11)$$

$$B_z(x,y,z) = \sum_{m,n=1}^{\infty} C_{mn} \exp(-r_{mn}z) \cdot \sin\left(\frac{\pi m x}{L_x}\right) \cdot \sin\left(\frac{\pi n y}{L_y}\right) \quad (2.12)$$

with $\lambda_{mn} = \pi^2(m^2/L_x^2 + n^2/L_y^2)$ and $r_{mn} = \sqrt{\lambda_{mn} - \alpha}$, and image sizes are L_x and L_y. We can find the coefficients C_{mn} by choosing $z = 0$ in the B_z formula and taking the FFT of $B_z(x, y, 0)$ (our image provided by MDI). In practice we have to do an antisymmetric mirroring of B_z before computing the FFT to get the identical formula:

$$B_z(-x,y) = -B(x,y) \quad (2.13)$$

$$B_z(x,-y) = -B(x,y) \quad (2.14)$$

Fig 2.7 contains an input image and a few traced lines from the resulting extrapolated magnetic field by the method developed by [14]. Also notice that lines which start very close to the edge of the image exit outside the frame due to periodicity of the discrete Fourier transform.

Since the coronal loops follow the magnetic field lines, we could use the extrapolated field lines to perform a feature based reconstruction of loops.

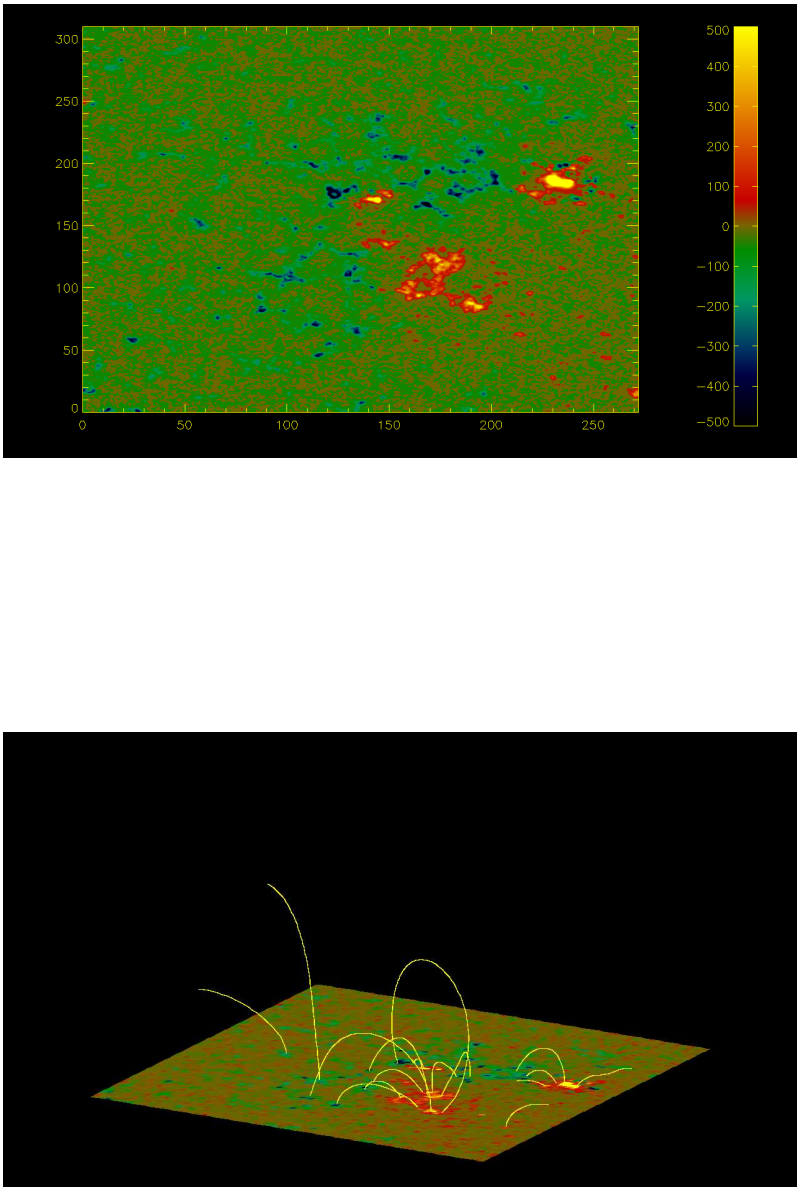

Figure 2.7. Linear force free magnetic field reconstruction

Chapter 3

3D GEOMETRY AND RECONSTRUCTION

This chapter will introduce some essential tools used in computer vision that will be used in the chapters to come. For a more in-depth introduction refer to [15, 16].

3.1 Homogeneous coordinates

The equation of a line in two dimensions is given by: $ax+by+c=0$, different choices for a, b and c generate different lines. It is also possible to rewrite this equation by using inner product:

$$\begin{pmatrix} a & b & c \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0 \tag{3.1}$$

The point $(x, y, 1)^T$ on the line is said to be the homogeneous representation of the 2D point (x, y). Clearly if such a point $(x, y, 1)^T$ belongs to the line, so will the point $(kx, ky, k)^T$. Thus we have an equivalence relation between all points that satisfy the equation of the line (a, b, c), $(x, y, 1) \equiv (kx, ky, k), \forall k \neq 0$. The concept of homogeneous coordinates, which are also called projective coordinates, can be expanded in a similar fashion to spaces of higher dimensions. The conversion between homogeneous and Euclidean points is straightforward: just multiply the point by constant such that the last coordinate becomes 1 and drop it:$(x, y, k) \sim (x/k, y/k, 1) \longrightarrow (x/k, y/k)$. It is important to note that even though the 2D homogeneous coordinates have 3 components, the dimension of the space is still two. One advantage of using the homogeneous coordinates is the ability to represent points and lines at infinity. This is simply done by letting the scale factor k tend to 0.

Another advantage of using this representation is the ability to represent the rotation and translation of a coordinate system as a linear operator. In case of 3D homogeneous coordinates this looks like:

$$\mathbf{P_c} = \mathcal{R}(\mathbf{P_w} - \mathcal{T}) \tag{3.2}$$

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{3.3}$$

3.2 Camera models

In most computer vision applications the data used is produced by cameras. Therefore it is crucial to be able to model the image formation. Throughout this section we will gradually develop the model for a perspective camera.

In its purest form, a camera consists of a focal point where all light rays intersect and a focal (imaging) plane where the image is formed, lying at a certain distance (focal length) (see Fig. 3.1).

The center of projection is called camera center. A line of sight is selected as the principle axis, that contains the camera center. Usually it is perpendicular to the image plane. The intersection of the principle axis with the imaging plane is called the principle point.

There are three coordinate systems tied to cameras that present importance (world, camera and image coordinate systems). The first one is the world coordinate system. To pass from the world system to the camera coordinate system we use the external parameters. The internal parameters allow us to pass from the camera system to the image coordinate system. The image coordinate system has the origin in the bottom left corner of the image (unlike image processing softwares that consider the origin in the top left corner). The Y axis is increasing upwards and the X

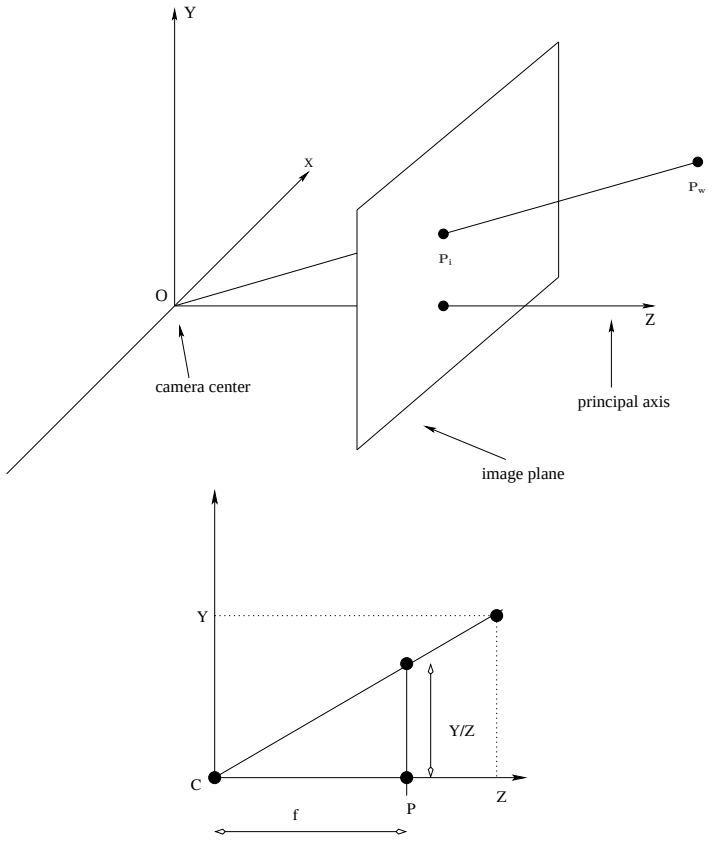

Figure 3.1. Pinhole camera projection model

from left to right. By convention the camera is observing the world in the negative Z direction.

Under this model, a point in the world $\mathbf{P_w} = (x, y, z)^T$ is mapped to a point on the image $\mathbf{P_i}$ that lies at the intersection of the line defined by the camera center and the point in the world, and the image plane. It is easy to notice that the point in the world $(x, y, z)^T \mapsto (fx/z, fy/z, f)^T$ under the previous projection. If we exclude the last coordinate we get: $(fx/z, fy/z)^T$. Defining depth as being $d = 1/z$ we get fdx, fdy.

If the world and image points are expressed in projective coordinates we can write

the mapping as:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & \cdot & \cdot & 0 \\ \cdot & f & \cdot & 0 \\ \cdot & \cdot & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.4}$$

This is a mapping from 3D projective to 2D projective space. The result we get is the same as before $(fx, fy, z)^T \sim (fx/z, fy/z, 1)^T$.

The previous projection model assumed that the origin of the coordinates in the image plane coresponds to the principle point. A more general form of the mapping is $(x, y, z)^T \rightarrow (fx/z + p_x, fy/z + p_y)^T$, with (p_x, p_y) being the coordinates of the central point. In matrix form this becomes:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & \cdot & p_x & 0 \\ \cdot & f & p_y & 0 \\ \cdot & \cdot & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.5}$$

The matrix:

$$K = \begin{bmatrix} f & \cdot & p_x \\ \cdot & f & p_y \\ \cdot & \cdot & 1 \end{bmatrix} \tag{3.6}$$

is called internal parameter matrix. This matrix captures intrinsic properties of the camera like the field of view and the position of the sensor with respect to the principle line (given by the optics). In mathematical terms this simply does a rescaling and shift of the points.

This projection model assumes that the world reference frame in which 3D points in the world are expressed coincides with the coordinate system of the camera. In general this is not the case so we are forced to do another transformation to align the coordinate systems. This transformation is described in equation (3.3). The rotation

and translation that are needed to align the camera with the world reference frame are called the external parameters matrix **M**.

Putting all these transformations together from world to the image we obtain:

$$\mathbf{P_c} = KM\mathbf{P_w} \tag{3.7}$$

$$\mathbf{P_c} = \begin{bmatrix} f & \cdot & p_x & 0 \\ \cdot & f & p_y & 0 \\ \cdot & \cdot & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \mathbf{P_w} \tag{3.8}$$

It is worth noting that a rotation around the Z camera axis is in fact a 2D transformation and can be perceived as being an external parameter or an internal one (a physical rotation of the CCD sensor). In this project we have considered this rotation as part of the internal parameters matrix. In this case the internal parameters matrix **K** becomes

$$\mathbf{K} = \begin{bmatrix} a & b & p_x \\ c & d & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

The upper 2x2 block does a Z-rotation and a scaling.

3.3 Radial distorsion

The assumptions so far were that the linear camera projection model is accurate. This remains valid for high-end lens with large focal lengths. When this is not the case, radial distorsion becomes apparent. This manifests itself by rendering straight lines in the world as curved, as illustrated in Fig. 3.2.

The position where the 3D points are projected gets affected by a non-linear function L, which depends only on the distance to a certain distorsion center. In camera coordinates (before applying the internal parameters) the distorsion model looks like this:

$$\begin{pmatrix} x \\ y \end{pmatrix} = L(\tilde{r}) \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \tag{3.10}$$

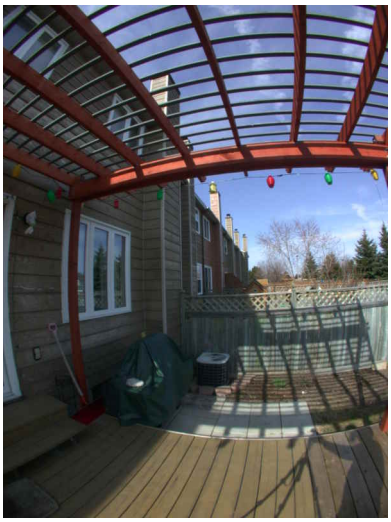

Figure 3.2. Radial distorsion [1]

where x, y are the coordinates flawed by radial distorsion and $\tilde{x}, \tilde{y}$ are the coordinates of the linear camera and $\tilde{r}$ is the distance to the distorsion center. This takes advantage of the fact that the optical center (and most of the time distorsion center) has the coordinates $(0, 0)$. In pixel coordinates the relation becomes:

$$x = x_c + L(\tilde{r})(\tilde{x} - x_c) \tag{3.11}$$

$$y = y_c + L(\tilde{r})(\tilde{y} - y_c) \tag{3.12}$$

with x_c, y_c being the distorsion centers. If the aspect ratio of the images is not 1, we need to multiply one of the coordinates by a scalar to bring it to 1, apply inverse distorsion and multiply by the inverse.

The radial distorsion function is defined only for positive values of r and $L(0) = 1$ such that the distorsion center does not get affected by the transformation. The function $L(r)$ is generally unknown (unless we have some prior knowledge about the optical system of the camera). An approximation to this is given by the Taylor expansion: $L(r) = \sum_{i=1}^{\infty} k_i r^i$. In practice three or four terms are enough to achieve good enough results. We consider the even expansion of the radial distorsion function

for negative values (since the distance is always positive). This means that if we consider only even power of r we will achieve same accuracy but with less parameters to estimate. In a similar fashion once could take odd powers if we consider the function to be odd.

The easiest way to estimate the parameters k_i for the radial distorsion is to minimize some cost based on derivation of some linear operator like a homography between a planar scene and an image. If we need to compute the distorsion centers as well x_c and y_c, we need to iterate between finding the distorsion center and reestimating the k_i's.

3.4 Planar homographies

A homography is a general planar (two dimensional) projective transformation. Homographies are extremely useful in practice as they enable us to rectify images such that they have certain properties like fronto-parallelism (views that differ just by a translation), useful for planar panorama making and for stereoscopic reconstruction. Also given enough homographies of the same camera with different planes one can compute most camera parameters (like internal parameters, essential matrix, etc.). Formally a homography is defined as a linear transformation: $H : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ that takes a point p_i to point p'_i, $p'_i = Hp_i$. In this formulation vectors which have the good orientation but differ in magnitude do not obey the equation as they should (since we are dealing with projective vectors). An alternative formulation of a homography is: $p'_i \times Hp_i = 0$. This leads to a set of linear equations that can be easily solved. Specifics can be found in [15].

3.5 Stereoscopic reconstruction

The general problem of stereoscopic reconstruction can be posed as: *given a set of images of the same scene, taken from different positions, recover the 3D information*

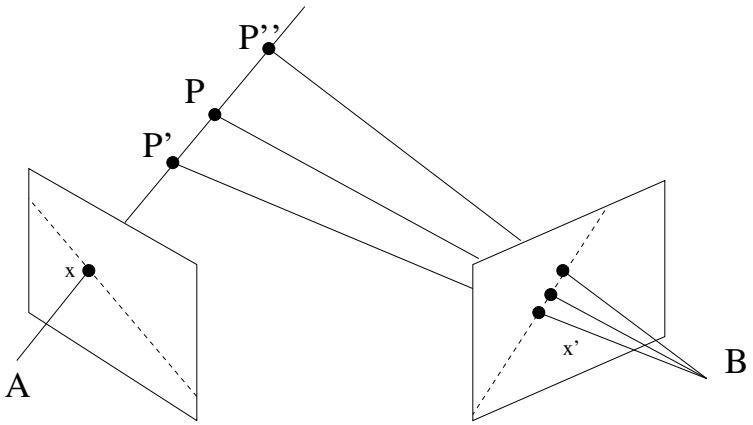

Figure 3.3. Triangulation

of each pixel in the image.

As you have noticed in the chapter 3.2, the only unknown for each pixel in the image is the Z coordinate of the 3D world point that generated the images. Therefore knowing the projection model for each camera (camera matrices) and the position of the cameras with respect to each other and pixel correspondences, one can calculate the missing coordinate by using triangulation.

As illustrated in Fig. 3.3, once we have managed to establish that the 3D world point P corresponds to point x in the reference image A and x' in image B it is fairly straightforward to solve the problem. If we know that a pixel in the first view corresponds to another pixel in the second view we can compute the position of the point in the world using triangulation. In order to obtain the pixel coordinates inside a camera with matrix M of a 3D projective point, we simply multiply the point with the matrix and divide by the third coordinate. Similarily to deproject an image point at depth d we simply multiply this point by the inverse camera matrix. The 3D projective coordinates of a pixel (i_x, i_y) in an image at depth d are: $(i_x, i_y, 1, d)$. The third coordinate of a pixel inside an image is equal to 1 since by convention the imaging plane is at $z = 1$ in the camera coordinate system.

In stereo, we pick x in the first image and by associating different depths d and reprojecting into the second view at x' and see if we have a good match.

The process of deprojecting a camera pixel (i_x, i_y) at depth d and reprojecting in a second image is called *triangulation*. Given a point in the first camera and that two camera matrices the deprojecting and reprojecting is done by computing:

$$M_b \cdot M_a^{-1} \begin{pmatrix} i_x \\ i_y \\ 1 \\ d \end{pmatrix} \tag{3.13}$$

We choose the depth of the pixel as being the one that minimizes the distance between our expected position and the actual position in the second image.

The correspondence estimation problem is far from being a trivial one. Besides the fact that noise can very quickly degrade our solution, we might encounter *occlusions*. In Fig. 3.4 you have an example of occlusion. Point B is visible from both camera, whereas because of the depth difference, point C is occluding A. Other complications include specularity and transparency of surfaces (which this project was aimed to deal with).

3.5.1 Epipolar geometry

The epipolar geometry between two views is the geometry that describes the relative positions of two cameras. It essentially describes for a point x in one image, the potential locations of matches in the second image. Observe in Fig. 3.5 that the two image points and camera centers are coplanar with the world point P. Similarly the backprojected rays that pass through x and x' are coplanar and intersect at P. This last property is of paramount importance to the correspondence problem as it limits the matches along a line. When epipolar lines are horizontal, the stereo process is greatly simplified to 1D horizontal searches. In this thesis a method is presented for rectification of solar images such that the epipolar lines are horizontal.

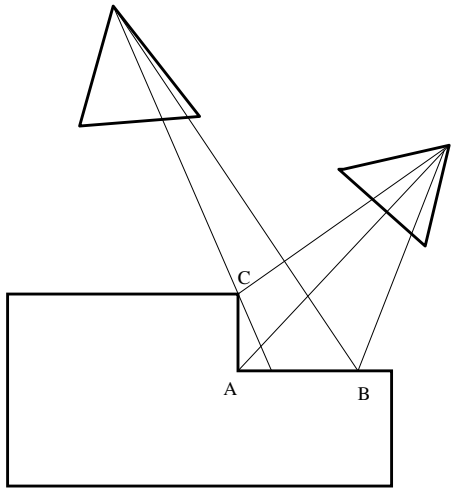

Figure 3.4. Occlusions: A is partially occluded, B is fully visible and C is an occluder

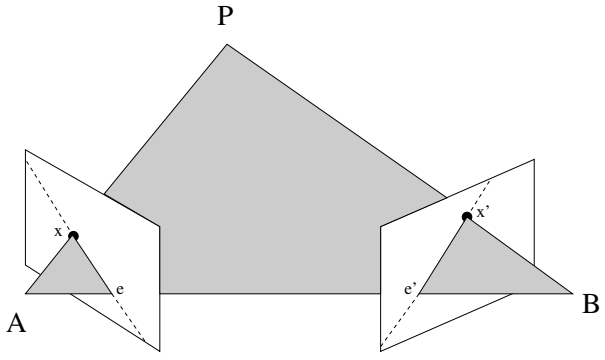

Figure 3.5. Epipolar geometry

The epipolar geometry is governed by the following parameters:

- The **epipole** e, e' is the intersection of the baseline AB with the two image planes.
- The **epipolar plane** is the plane that contains the baseline. This has one free parameter, the angle.
- The **epipolar lines** are the intersection of the epipolar plane with the two imaging planes. This gives correspondences between lines.

The method how to derive formulas for the epipolar planes will be given in the chapter 4.

3.5.2 Establishing correspondences

In order to match pixels along an epipolar line, we define the similarity of two pixels in terms of a cost function. Common choices for cost functions are:

$$c = \sum \parallel v_i - v_r \parallel_n \tag{3.14}$$

$$c = \sum \parallel v_i - \overline{v} \parallel_n \tag{3.15}$$

where v_i is the pixel intensity value in the i^{th} camera, v_r is the reference pixel value and $\overline{v}$ is an average pixel value. $\| \cdot \|_n$ is the L_n norm. Common choices for n are 1 or 2. In order for such cost functions to work one has to make the following assumptions:

- The objects are opaque
- Constant intensity in all views (a world point projects to the same intensity value in both images)
- Lambertian[1] surface

[1] Lambertian surfaces reflect light the same way regardless of the viewing angle

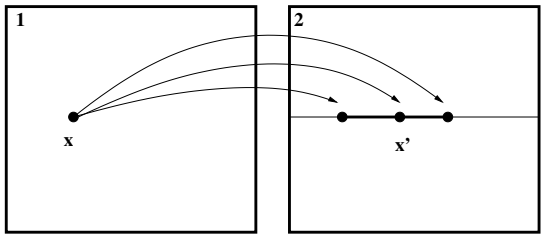

Figure 3.6. Correspondence of x and x' on an epipolar line

- No occlusions

The simplest method is to choose one pixel in the first image and search inside an interval in the second image for the best match according to our cost function (Fig. 3.6). This approach was proposed by Kanade [17]. This method calculates correspondences of each pixel independent, giving a noisy estimate. In practice neighboring pixels usually have the same value, depth (not considering discontinuities) and adding a smoothing cost will greatly improve solution.

Since real world surfaces tend to be smooth we can include a smoothing cost by matching two whole epipolar lines together. The new energy function will be of the following form:

$$E = E_c + E_s \tag{3.16}$$

The first term, E_c, is the correspondence cost, defined earlier. The second term, E_s, penalizes the difference of depth between neighboring pixels along an epipolar line. This can be again some norm of the difference between the disparity of the current pixel and that of its neighbors. Such problems can be easily solved using *Dynamic Programming* (see Fig. 3.7). The cost $Cost(x, d)$, of the pixel x in the first image to

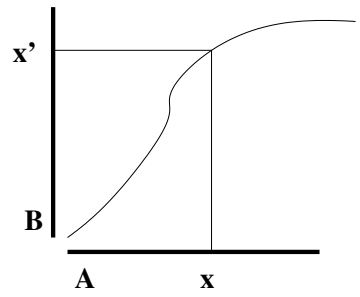

Figure 3.7. Dynamic programming

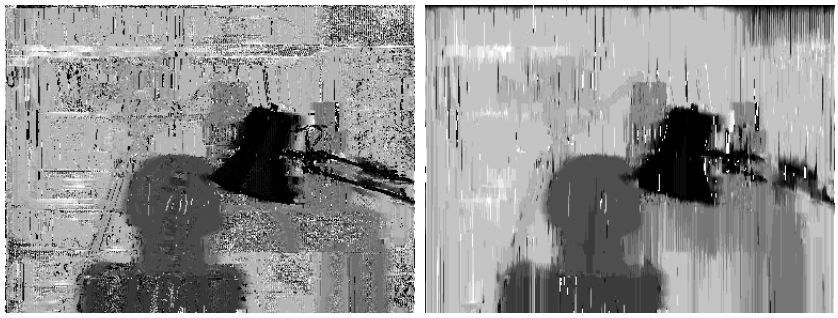

Figure 3.8. Tsukuba dataset: Left - direct search, Right - dynamic programming

match at $x + d$ in the second one is:

$$Cost(0, d) \quad = c(0, d) \tag{3.17}$$

$$Cost(x, d) \quad = \min_{d'} [c(x, d') + Cost(x - 1, d') + S(d, d')] \tag{3.18}$$

where $c(x, d)$ is the correspondence cost and $S(d, d')$ is a smoothing cost.

In Fig. 3.8 you can observe the resulting depth map of the two algorithms on the famous *Tsukuba* dataset. The direct search method result is much noisier than the dynamic programming one. You can observe some "streaks" in the dynamic programming solution, as the smoothing is imposed only along horizontal epipolar lines.

3.5.3 Volumetric reconstruction

The approach that was outlined, goes by the name of *stereoscopic reconstruction*. A reference view was chosen and the scene was reconstructed from the point of view of this camera. However this becomes impractical as the number of views grows. Because of the occlusions, this method works only if all cameras are situated on the same side of the object. Also this method breaks down if two cameras are facing each other.

To get rid of these limitations the problem can be approached from a slightly different angle. Instead of choosing a reference view, we discretized the 3D reconstruction space into voxels. Each voxel can be projected in each camera. The color of each voxel can be taken as the average of the colors in the cameras that see this voxel. Occlusions can cause a lot of problems since the views are often very separated. One of the most popular volumetric reconstruction algorithms is the *space carving algorithm* proposed in [18].

3.6 Satellite camera calibration

In this section we will present how the two satellite camera matrices are computed. In order to calibrate the external parameters of the camera, we need to find the three translation components and the orientation information (rotation with respect to the world coordinate system). For the internal parameter matrix we need one focal length, two values for the optical center (in pixels) and one parameter which is the rotation around the camera Z axis (the Z rotation can be considered as either internal or external parameter as it is a two dimensional transformation). Since all parameters provided by the mission, contain a fair amount of error we will introduce an extra matrix that corrects the value for all linear acting parameters. Additionally we want to calibrate for radial distorsion so 3 extra parameters are needed (more parameters do not introduce significant improvements). Since radial distorsion is not

linear in nature, it is impossible to express it as a matrix operator.

3.6.1 External parameters

We choose the HAE (Heliocentric Aries Ecliptic) system as the world coordinate system. The reason for this is that this system is most stationary of the ones given by the NASA and most other spaceborne missions have their coordinates in this system as well. This system has its origin at the center of the Sun, the X axis points at the *first point of Aries*, Z towards the ecliptic north pole, and the Y axis is defined as a cross product of the other two to end up with a right-handed coordinate system.

The three components of translation are given already in the header of the images as **HAEX_OBS**, **HAEY_OBS**, **HAEZ_OBS**.

From the mission description we know that the satellites are looking approximately towards the center of the Sun (origin). To find the rotation we will procede by a constructive approach. Since the camera looks towards the negative Z axis, the camera Z axis should be equal to normalized translation vector. We have computed the camera coordinate system up to a rotation around the Z axis. We choose the camera X axis to be perpendicular to the plane formed by the world Z and camera Z axis. The camera Y axis is just the cross product between the camera Z and X axes.

Once we have the new coordinate system, the rotation matrix between the standard (canonical) coordinate system and an arbitrary one is just a stacking of the axis

vectors.

$$c_z = T \tag{3.19}$$

$$c_x = c_z \times [0, 0, 1] \tag{3.20}$$

$$c_y = c_z \times c_x \tag{3.21}$$

$$R = \begin{bmatrix} c_x^T \\ c_y^T \\ c_z^T \end{bmatrix} \tag{3.22}$$

where T is the translation, c_x, c_y, c_z are the camera coordinate system axis and R is our rotation matrix. With these parameters computed, the external parameter matrix is simply $M = [R \mid T]$.

3.6.2 Internal parameters matrix

All parameters for the internal matrix are given in the FITS headers, but in a form which is not really usable for computer vision. The internal parameter matrix normally produces a shift and rescale between the image and camera coordinate system. Additionally, in case of STEREO camera there is an extra rotation around the optical axis. The location of the optical center is given by the **CRPIX1** and **CRPIX2** header keywords. The image scale is given in arcseconds/pixel is given by the **CDELT1** and **CDELT2** keywords. This value has to be multiplied by $\frac{\pi}{3600 \cdot 180}$ in order to get radians/pixel. The Z rotation matrix components is also given as **PC1_1**, **PC1_2**, **PC2_1**, **PC2_2**. With this the internal parameters matrix is [9]:

$$M_i = \begin{bmatrix} -\mathbf{PC2\_1}/\alpha & \mathbf{PC1\_2}/\alpha & \mathbf{CRPIX1} - 1 & 0 \\ \mathbf{PC2\_1}/\beta & \mathbf{PC1\_1}/\beta & \mathbf{CRPIX2} - 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.23}$$

$$\alpha = \mathbf{CDELT1}\,(\mathbf{PC1\_2} \cdot \mathbf{PC2\_1} - \mathbf{PC1\_1} \cdot \mathbf{PC2\_2}) \tag{3.24}$$

$$\beta = \mathbf{CDELT2}\,(\mathbf{PC1\_2} \cdot \mathbf{PC2\_1} - \mathbf{PC1\_1} \cdot \mathbf{PC2\_2}) \tag{3.25}$$

The upper 2x2 block does the rotation and scaling and the other two entries the shift. The matrix has this complicated form since all parameters given in the header perform the conversion from image to camera coordinates, but the internal parameter matrix is supposed to perform the conversion in the other direction.

3.6.3 Corrections matrix

The STEREO B satellite is assumed to have accurate internal parameters. We are to find the corrections to the internal parameters for the STEREO A images such that the alignment fits best. Note that the radial distorsion is assumed to be the same for both images. We introduce the following linear correction parameters:

- two parameters for the optical center
- one parameter for the scale factor
- one rotation angle around the Z axis

We have also tried optimizing for rotations around the X and Y axis, but the effect is almost totally explainable by the shift of optical center since the field of view is very small.

With this new matrix, the projection model becomes:

$$P_i = M_{shift} M_{scale} R_z M_{int} R_x R_y M_{ext} P_w \tag{3.26}$$

where P_i is a point in the image, P_w is a 3D world point, M_{int} internal parameter matrix, M_{ext} external parameter matrix, M_{scale} matrix adds a multiplier to the calculated focal length and M_{shift} changes the position of the optical center. In all that follows R_x and R_y are considered identity because of their insignificant effect.

We can group all the non-identity matrices into one $M_{corr} = M_{shift} M_{scale} R_z$:

$$M_{shift}(dx, dy) = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.27}$$

$$M_{scale}(sx, sy) = \begin{bmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.28}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.29}$$

$$\tag{3.30}$$

We notice that all correction parameters are linear Euclidean two dimensional transformations. The chosen objective function is the mean squared sum of differences between the two images inside a patch in the $304\mathring{A}$ wavelength (orange images). These images provide a view of the surface of the Sun. At this depth there are not many proeminences, and the rectification is made in such a way that objects at R_{Sun} will not exhibit any parallax.

The radial distorsion is not a linear transformation giving a very different effect from a scale/shift transformation. For this reason the problem is easily optimized (the cost function does not have valleys if one considers any pair of variables). This cost function contains 7 variables (2 for shift, 1 for scale, 1 for Z rotation and 3 for radial distorsion).

The deformation model for the radial distorsion is taken as in [19]:

$$L(r) = 1 + k_1 r^2 + k_2 r^4 + k_3 r^6 \tag{3.31}$$

This correction is applied in the end:

$$x = x_c + L(\tilde{r})(\tilde{x} - x_c) \tag{3.32}$$

$$y = y_c + L(\tilde{r})(\tilde{y} - y_c) \tag{3.33}$$

with $P_i = [x, y, 1, a]^T$, previously defined and $r = \sqrt{x^2 - y^2}$, where r is taken as the distance to a distorsion center, the center of the image in our case.

In the next chapter we introduce a method to rectify images taken by the STEREO mission where structures on the surface of the Sun are situated on the zero disparity surface (there is no motion parallax). This is particularly useful to align the two available views (that observe the surface of the Sun).

To compute the parameters for the correction matrix and radial distortion we try to minimize the sum of square differences between pixels of the 2 views taken in the $304\mathring{A}$. This wavelength gets formed very close to surface of the Sun, thus carrying very little depth information.

There are times when the minimization algorithm does not converge to the global minimum since the cost function might become very noisy because of the non-linear parameters (radial distorsion or Z-rotation). When this occurs we will perform the minimization in two steps: first start minimize the linear parameters setting the non-linear ones to 0. In the second step we set the linear term to the optimal values and minimize the non-linear terms. This ensures that the starting point for non-linear parts is close to the true solution.

An alternative is to minimize all variables at once and use some probabilistic minimization algorithm like simulated annealing, but this is extremely slow.

Chapter 4

RECTIFICATION

4.1 Related work

In section 3.5 we introduced the concept of *epipolar geometry*. This makes it possible to reduce the stereo search space from two dimensions to one. Since matching along horizontal epipolar lines is very desirable, we will rectify the solar images. All rectification methods require that the cameras to be calibrated (internal and external parameters), which was described in the previous chapter.

The first rectification method we will be presenting is introduced in [20] and is by far the simplest method but does not work for all camera configurations. Next we will present a brief introduction to the cylindrical rectification method [21] which resolves the previously mentioned problems. In the end we will present a rectification scheme that is specifically adapted to the case of spherical objects.

The rectification can be characterized in general terms as a succession of following operations:

- rotation of a pencil plane around an axis (baseline) and intersection with the two imaging planes
- mapping of an epipolar line onto a surface with a specific discretization

4.2 Planar rectification

The planar rectification is also known as rectification with homographies. The goal of rectification is making all epipolar lines parallel to each each other and aligned with one axis of the image (see Fig. 4.1). In order for the lines to be parallel the epipoles

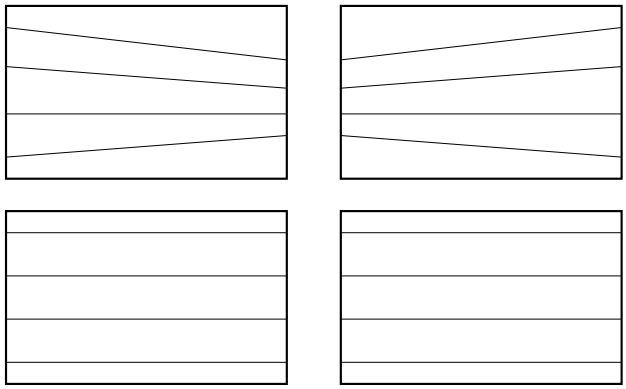

Figure 4.1. Original and rectified epipolar lines

have to be mapped at infinity. This is being realized by remapping the images onto two fronto-parallel views (two planes that differ just by a translation).

Without loss of generality we assume the following:

- R, T and principle point for both cameras are known (camera matrices).
- the origin of the image coordinate system is at the principal point of the right camera.
- both cameras have focal length f.

The algorithm consists of finding the rotation matrix such that the epipoles in both cameras go to infinity horizontally. Next we compute a second rotation, between the two cameras and align them to be fronto-parallel. As a last step we have to adjust the scales of the images.

In order to find the rotation matrix to make the views fronto-parallel we have to find 3 mutually orthogonal vectors e_1, e_2, e_3. This problem is underconstrained so we have to make an arbitrary choice for vectors. The vector e_1 is given by the epipole, which is actually the translation between cameras:

$$e_1 = \frac{T}{\| T \|} \tag{4.1}$$

We choose e_2 as being perpendicular to e_1 (we have one degree of freedom). For this we can take the cross product between the optical axis of one camera and the vector e_1. This gives the vector e_2 perpendicular to the plane formed by the optical axis and e_1:

$$e_2 = \frac{1}{\sqrt{T_x^2 + T_y^2}} \begin{bmatrix} -T_y & T_x & 0 \end{bmatrix}^T \tag{4.2}$$

The third vector e_3 is simply the normalized cross product of e_1 and e_2, $e_3 = e_1 \times e_2$. Once these vectors are computed, the rotation, R_{rect}, that makes the epipolar lines go to infinity is:

$$R_{rect} = \begin{pmatrix} e_1^T \\ e_2^T \\ e_3^T \end{pmatrix} \tag{4.3}$$

The rectification algorithm in short follows the following steps:

- compute R_{rect} from equation 4.3.
- compute the rotation matrices for left and right cameras $R_l = R_{rect}$, $R_r = RR_{rect}$, where R is the rotation matrix of the left camera.
- multiply each pixel $p = [x, y, f]^T$ from the left and right images by the appropriate rotation matrices, R_r, R_l, $R_l P = [x', y', z']$.
- rescale left and right images according to $p'_l = f/z'[x', y', z']$.

The pixel coordinates obtained through rectification will probably not be integer. In order to maintain the image quality it is better to perform the rectification the other way around: for each pixel in the final rectified image, one should apply the inverse transformation and end up with fractional coordinates in the original image, which can be interpolated.

One problem when rectifying images is that the image bounds will not be the same. If the original and final images have to have the same size, one can change the scale applied to the image.

There are certain camera configurations that are impossible to rectify by this method. One is if one camera can "see" the other camera's optical center inside the image (the translation in the camera Z direction is significant). In such a case the epipolar lines are radial around a point called the focus of expansion (FOE). In this case rectified images have infinite size. There is no rotation matrix that can rectify such pairs of images. Also the distorsion of images through the rectification process is a concern when the cameras are approaching this degenerate configuration.

4.3 Cylindrical rectification

The previous rectification method remaps images onto two fronto-parallel planes. While this is a very simple and efficient method as all operations are linear 2D projective, it has some problems. The method proposed by [21] and slightly modified by [22] employs 3D projective transformations. We will just provide the outline for the methods as they are more complicated.

As the name suggests this method remaps the images from image planes to a unit radius cylinder that has its axis aligned with the baseline (the line defined by the two camera optical centers). The method proceeds in a similar fashion as for the planar rectification. The rectification is done in three steps:

- each epipolar line gets rotated to be get parallel with the baseline.
- a translation is applied to change the reference system from each camera to the cylinder.
- a scaling is applied to bring the line to the unit radius cylinder.

While the planar rectification applies a global linear transformation, the cylindrical rectification method needs one linear transformation per epipolar line. This method guarantees that the final image will be of finite size regardless of the camera configuration. The resulting image is of minimal size such that there is no loss of information through the transformation. The length of the epipolar lines is preserved but unfortunately the straight lines which are not parallel with the epipolar lines are not preserved.

This method can handle arbitrary camera geometries, but not panoramic cameras that have a viewing angle of 180° or more.

4.4 Spherical rectification

An equally good rectification surface would be a sphere. Besides being able to handle an arbitrary camera configuration and keeping rectification images bounded it adds a few useful properties when the observed object is spherical. Unfortunately no straight lines inside the images will be preserved after rectification, unless they are the epipolar lines themselves. However the transformation is totally reversible.

This new rectification scheme has the following properties:

- zero disparity surface should be on the Sun.
- voxels that are induced by the rectification, that are further from the Sun should always project inside the images on integer pixel coordinates.

As you will see in chapter 5 the second property will be very useful when computing each voxel contribution to an image pixel.

The rectification is illustrated in Fig. 4.2 for two cameras observing a spherical object.

If we discretize the common visible surface of the Sun and the cast the rays that join each point on the Sun with the two cameras, we obtain a mesh that satisfies

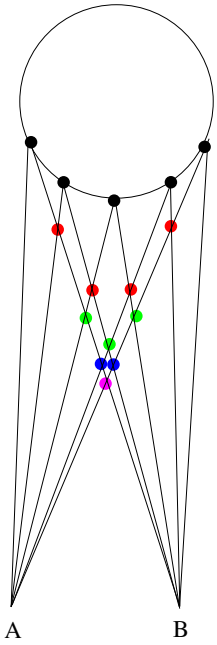

Figure 4.2. Epipolar line with spherical rectification

the first requirement: if all matter is concentrated on the surface of the Sun (no transparency on top) the stereo algorithm will match pixel i in the first image with pixel i in the second image (first ray cast from camera A will intersect first ray from camera B). In order to satisfy the second requirement we define the grid at height k as being the place where of intersection between i^{th} ray from camera A and $(i-k)^{th}$ ray from camera B. Since the grid is defined by the rays from the surface of the Sun to the two cameras, all higher voxels will project to the point on the images. In Fig. 4.2 you can clearly see the five levels of the mesh, corresponding to the 5 disparities. One can easily notice that this way of building the mesh has two unwanted properties:

- the k^{th} level has k less voxels than the zero height (surface of the Sun).
- voxels close to the middle of the grid expand faster near the side.

In order to avoid the use of an unstructured grid we can repeat k times the first

point to be able to keep the number of voxels constant for all layers.

The second unwanted property can be easily fixed by choosing a non-uniform discretization (we introduce more points in the middle).

In order to rectify the images we will rotate a pencil plane around the baseline and intersect it with the sphere. There are two basic geometric problems that we will encounter multiple times throughout this rectification scheme: equation of the circle that is generated by intersecting a sphere with a plane and tangent lines to the sphere that are contained inside a plane.

4.4.1 Intersection of a plane with a sphere

First we have to find out the center of the circle. If the plane is defined by its normal, the center of the circle is simply the position on the plane where the normal passes through the center of the sphere. Notice in Fig. 4.3 that a right triangle is formed by the center of the sphere, center of the circle and any point on the circle. From this triangle we can find the radius of the circle. Now to generate a circle we just have to rotate around the plane normal vector:

$$Circle = Center + R * [\cos a, \sin a, 0] \cdot [n_x, n_y, n_z]^T \tag{4.4}$$

where $[n_x, n_y, n_z]$ is the plane normal and a spans $[-\pi, \pi]$. The basics are illustrated in Fig. 4.3.

4.4.2 Tangent line to a sphere

We need to find the angle at which a line contained inside a plane becomes tangent to the sphere. The plane has the normal parallel to the Z axis. Consider the problem of finding the angle α at which a line becomes tangent to the circle shown in Fig 4.4. There are two coordinate systems that are important: given an X axis (which is the baseline in our case), one coordinate system has Y pointing towards the center of the sphere - $CS_1(P, x_1, y_1)$ and another that differs by a rotation around the Z axis,

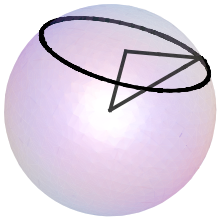

Figure 4.3. Sphere with corresponding circle

$CS_2(P, x_1, y_2)$. In Fig 4.4 you can see the 2D problem we have to solve (the plane defined by CS_2).

The angle α inside the CS_1 coordinate system is just given by $\cos^{-1}(\frac{AC}{PC})$, where AC is the radius of the circle resulting from intersecting the sphere with the plane defined by CS_2. The vectors $v_1 = Ry_1$ and $v_2 = R^T y_1$ will become collinear with PA, PB respectively (R is the rotation matrix around the axis z by α). The angles of tangency inside CS_2 are given by: $\tan^{-1}(\frac{v_1 \cdot y_2}{v_1 \cdot x_2})$ and $\tan^{-1}(\frac{v_2 \cdot y_2}{v_2 \cdot x_2})$ respectively. Note that generally, the two angles of tangency are equal inside CS_1, but not inside C_2. It is important to ensure that all returned angles are inside the interval $[-\pi, \pi]$.

In order to rectify with this method we have to go through the following stages:

- find the common visible region from the two views (latitude angles) and discretize.
- for each generated plane find the common longitude angles and discretize.
- project images onto the the discretized sphere (latitude and longitude).

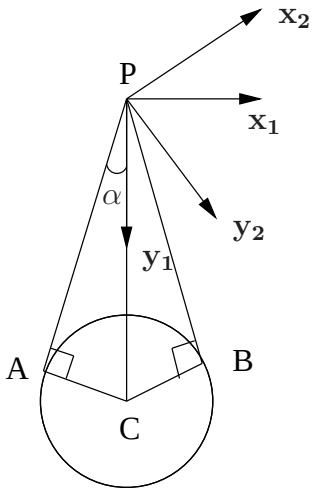

Figure 4.4. Two dimensional circle tangent problem.

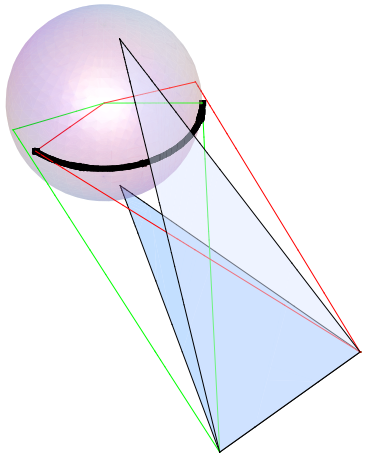

Figure 4.5. Solar rectification. Satellite A and B with the highest and lowest epipolar planes (tangets to the Sun)

4.4.3 Common latitude range

We consider the following coordinate axis for the plane: x-axis is pointing along the baseline from satellite A to B, Z axis is perpendicular to the plane formed by the x-axis and the translation of satellite A and Y perpendicular to the other two. The Y-axis is the perpendicular to the baseline and passes through the center of the sphere.

With this coordinate system, the problem of finding the latitude range amounts to finding the tangents to the sphere that are contained in the YZ plane and can be carried out by the above mentioned method. We can place the origin of the coordinate system anywhere onto the baseline without changing the result.

4.4.4 Common longitude range

We will have to solve for the range of latitude angles to remain onto the sphere for both satellites. Afterwards we compute the overlapping interval. We have to compute this range for each latitude angle.

The coordinate system differs from the previous one simply by the fact that it is rotated around the x-axis by the chosen latitude. The origin is chosen in turn at the position of satellite A and B.

The problem is simply to find the two tangents to the sphere that are inside the XY plane.

In Fig. 4.5 the two satellites are shown together with the Sun and the common latitude. The thick line on the Sun represents the intersection of a pencil plane that rotates around the baseline and the Sun (one epipolar line). To rectify the whole image the plane sweeps the whole interval of common latitude.

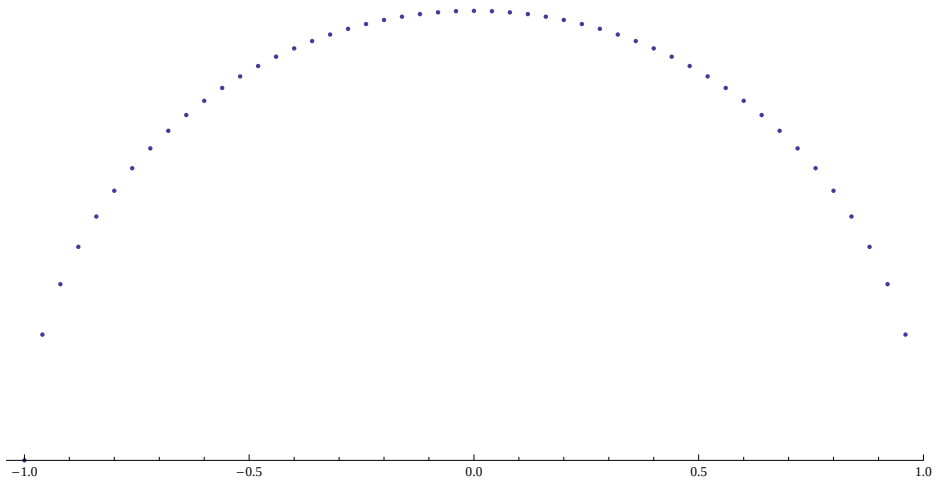

Figure 4.6. Cartesian discretization of a circle

4.4.5 Discretization of angle ranges

The easiest way to subdivide the angle range is select points with an uniform arc spacing. There are however some disadvantages to this. We have observed in Fig. 4.2 that points which are closer to the middle of the angle range move much faster away from the Sun when we increase the grid layer. This is inconvenient, since we will have a very low resolution inside our reconstruction grid for the center points (the resolution is limited by the fastest moving point). A simple solution is to sample finer towards the center of the interval. Such sampling functions will generally have discontinuities at the ends of the interval. Consider the cartesian representation of the positive half of a unit circle: $y = \sqrt{1 - x^2}$, shown in Fig. 4.6.

Notice that points at the sides are spaced very far away if we look at the arclength even though the original grid was uniform. The behavior is captured by the derivative of such function:

$$\frac{d}{dx}\sqrt{1 - x^2} = \frac{x}{\sqrt{1 - x^2}} \tag{4.5}$$

This function clearly has discontinuities at ± 1.

Since our angle range are not from -1 to 1, we start with the uniformly spaced

interval $(-1, 1)$. Afterwards we apply our resampling function (the derivative of the the circle) and rescale the result to $(-1, 1)$ since the first and last points will tend to infinity. Next we simply rescale these points to the interval $[\alpha_1, \alpha_2]$ (the two tangent angles).

We can choose the amount of "non-uniformity" by starting with the interval $(-k, k)$, $k < 1$ rather than $(-1, 1)$. If k is very small we end up in the uniform sampling case as the circle does not vary very much around 0.

4.5 Some results

In Fig. 4.7 you can see some images resulting from rectification using $k = 0.7$.

Notice that in the case of uniform sampling the edges are extremely stretched. This is due to the fact that pixels which are close to the sides of the Sun have the same size as pixels in the center, which is not backed by the observation model. For an example of both left and right images refer to Fig. 6.8

For other spherical rectification models check [23, 24]

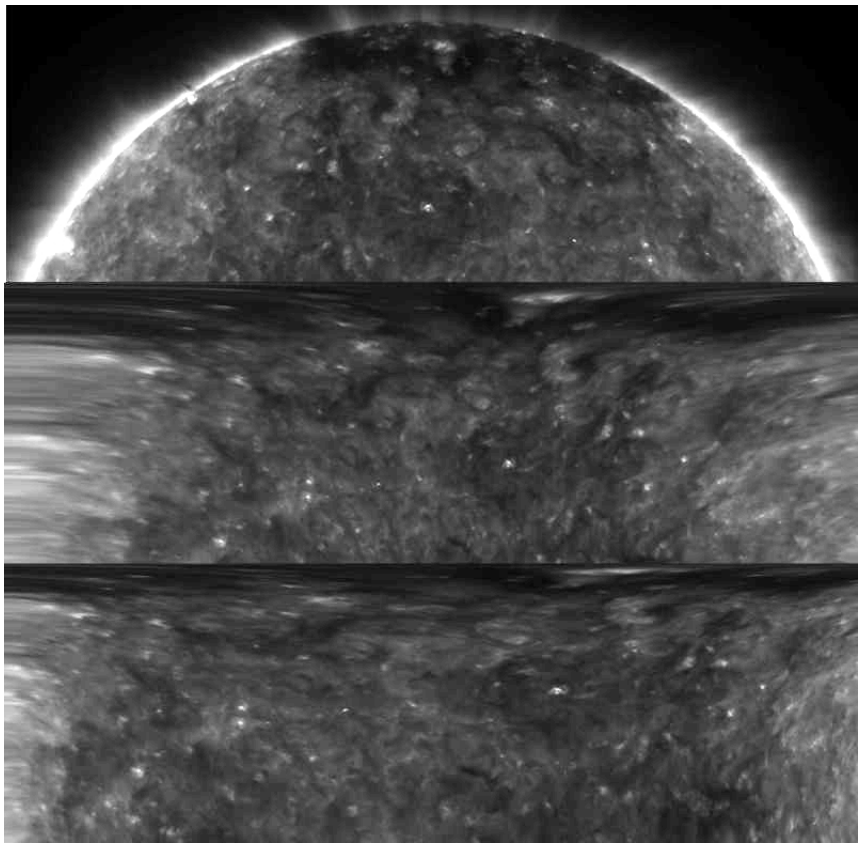

Figure 4.7. Top: original image, Middle: uniform sampling, Bottom: non uniform sampling

Chapter 5

RECONSTRUCTION OF SEMI-TRANSPARENT VOLUMES

While there are a lot of algorithms that provide excellent reconstruction methods for objects that are opaque and Lambertian, the problem of transparency or specularity is still largely open. As is the case with most inverse problems, it is ill posed.

5.1 Related work

The problem of estimating multiple depths inside transparent scenes has been widely studied. The main contributions come from the fields of medical imaging, atmospheric science and combustion. Most algorithms designed to handle transparency were conceived to use large number of views or make a lot of assumptions about the shape of the objects being reconstructed. We want to develop an algorithm that can provide satisfactory results with just two or three views and make only minimal assumptions about the observed objects.

5.1.1 Medical imaging

The problem of 3D reconstruction of transparent objects received most interest in the context of computerized tomography. There are methods that work with as few as two images but they produce just binary segmentation maps rather than a full reconstruction [25–27]. In order to obtain full reconstructions, the number of views needed ranges from tens to hundreds views. Even with additional regularization assumptions the problem still becomes unsolvable when a very small number of views

is available.

In [28] a method to reconstruct plasma (aurora Borealis/Australis) is given. It was conceived to work with images from the IMAGE mission (Imager for Magnetopause-to-Aurora Global Exploration), a mission that was supposed to provide insight into the connections between the solar and earth magnetic fields. The method presented in [28] uses a version of the tomographic reconstruction method, with additional symmetry assumptions on the solution. The input consists of images from a single satellite that are separated in time.

5.1.2 Computer vision

Probably the article that pioneered the study of transparency in the context of computer vision was [29]. The algorithm iterates between the following steps: initial disparity is estimated, visibility/transparency map is updated and last the color information at each depth gets updated. The algorithm works in a 4D space, the dimensions being x and y (image coordinates), d (number of disparities), k (number of available views).

To compute the initial disparities, the 4D space gets populated with color intensities:

$$c(x, y, d, k) = \mathcal{W}_f(c_k(u, v); H_k + t_k[0, 0, d]) \tag{5.1}$$

with $c_k(u, v)$ the k-th image, $\mathcal{W}_f(\circ; H_k)$ is a linear operator that rectifies each image from the point of view of a virtual/reference camera, $H_k + t_k[0, 0, d]$ is a homography that maps from camera k onto the d-th homography plane and $c(x, y, d, k)$ is the pixel color projected onto the 4D space. Next we can compute some statistics on color distribution such as μ, σ^2 over the k-dimension for each (x, y, d). If we just choose for each x, y, the d that minimizes the σ^2 we are replicating the direct search algorithm presented in the introductory chapter, however visibility is not being taken into account and neither is transparency, thus the results will be disappointing.

Now we consider that each of the input images is formed by stacking d semi-transparent layers. For this we apply the inverse mapping $\mathcal{W}_b$ from the virtual camera back to the input image k.

$$\tilde{c}_k(u,v,d) = \mathcal{W}_b(\hat{c}_k(x,y,d); H_k + t_k[0,0,d]) \tag{5.2}$$

where $\hat{c}_k = [r,g,b,\alpha]^T$ is the color information at (x,y,d) and $\tilde{c}$ is the color information in the k-th camera coordinate system and α is the corresponding transparency.

Next we have to compute $\tilde{c}_k(u,v)$, the composite of the "transparent sheets" into each view k and compare with the original data. For this we need to define the visibility of each pixel $V_k(u,v,d)$:

$$V_k(u,v,d-1) = V_k(u,v,d)(1-\tilde{\alpha}(u,v,d)) \tag{5.3}$$

$$= \prod_{d'=d}^{d_{max}} (1-\tilde{\alpha}_k(u,v,d')) \tag{5.4}$$

where $\tilde{\alpha}_k$ is the opacity of $\tilde{c}_k(u,v,d)$. Initially all visibilities $V_k(u,v,d_{max}) = 1$ and they are propagated from front to back. The moment one pixel becomes fully visible ($V_k = 1$), it will obstruct all pixels behind it. Now we have an easy way to composite images for each view k:

$$\tilde{c}_k(u,v) = \sum_{d=d_{min}}^{d_{max}} \tilde{c}_k(u,v,d) V_k(u,v,d) \tag{5.5}$$

As a last step we have to update the color information (see how far away are we from the k input images). This problem can be posed as a non-linear minimization problem with 3 terms:

$$\mathcal{C}_1 = \sum_{(u,v)} w_k(u,v) \rho_1 \left(\tilde{c}_k(u,v) - c_k(u,v)\right) \tag{5.6}$$

with w_k being a weighting function that gives more importance to certain cameras depending on their proximity to the virtual reference camera.

$$\mathcal{C}_2 = \sum_{(x,y,d)} \rho_1 \left(\triangle \hat{c}_k(x,y,d)\right) \tag{5.7}$$

with $\triangle\hat{c}_k(x,y,d)$ being the Laplacian of the color and transparency information. This enforces smoothness.

$$\mathcal{C}_3 = \sum_{(x,y,d)} \phi\left(\alpha(x,y,d)\right) \tag{5.8}$$

In the above formulas ρ_1,ρ_2 are quadratic or robust penalty functions. The function ϕ increases sparsity of the solution preferring solutions where matter is fully opaque or transparent $\phi(x) = x(1-x)$. The total cost function is:

$$\mathcal{C} = \lambda_1\mathcal{C}_1 + \lambda_2\mathcal{C}_2 + \lambda_3\mathcal{C}_3 \tag{5.9}$$

which can be easily solved with a conjugate gradient-like algorithm.

This whole algorithm provides acceptable results but the fact that we have to fill in initially the whole 4D space makes it very expensive in practice. We are also unable to make sure that the reprojected colors and opacities $\hat{c}_k(x,y,d)$ in the k available views, $\tilde{c}(u,v)$ are consistent with the images. Also the fact that for each voxel in the 4D space we compute both a color and alpha information makes the problem very hard to minimize as the model is too flexible. In fact we can always exchange a pixel with certain color and alpha by another one with lower color value and higher alpha or vice-versa.

Another important contribution from the field of computer graphics, and the inspiration for our approach, is introduced by [30]. A method is developed for reconstructing flames from as few as two views. The solution provided by the method exhibits some nice properties:

1. concentrates matter along continuous surfaces

2. is photoconsistent

3. most spatially compact distribution

The reconstruction problem is reduced to finding a convex combination of sheet-like densities derived from the two input views. The method assumes a linear image

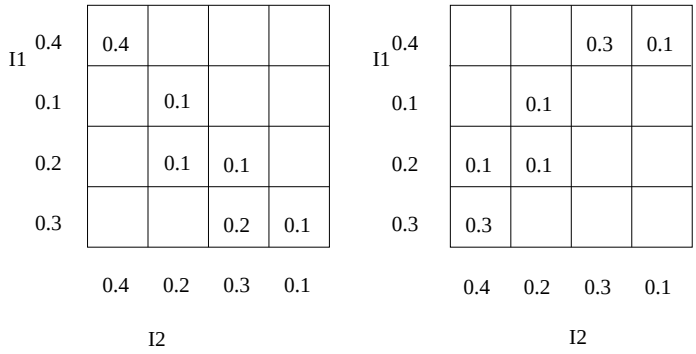

Figure 5.1. Density sheet reconstructions generated by two orthogonal views

formation mode:

$$I = \int_z D(z)dz \tag{5.10}$$

The observed intensity I is the integral along the line of sight of all densities. The method makes some implicit assumptions. Negligible scattering, means that only line of sight voxels contribute to the intensity. Constant emissivity assumes that each voxel emits a constant amount of light and a higher intensity implies more matter. Also each voxel just emits light and does not absorb it.

The two initial sheets are chosen to be monotonous curves in the (I_1, I_2) space (Fig 5.1).

We compute the density sheets for each pair of orthogonal views. With this the problem becomes:

$$I_m(p) = \sum_{r,c} w(r, c, p)D(r, c) \tag{5.11}$$

where the unknown is w and D is a density sheet. This is equivalent to finding the solution in terms of the basis formed by all the density sheets and w represents the size of the projection of the solution onto the corresponding element. Stacking all these equations together we are left with a problem of the kind:

$$\text{minimize } || \mathcal{F}x - I ||$$
$$\text{subject to } \sum x = 1,\ x \geq 0$$

The density sheet basis can be extended leading to a better reconstruction, but the number of such basses increases exponentially with the number or views.

Other more exotic approaches include [31], in which a matching operator is defined for the standard stereo match problem. This operator gets generalized for matching multiple disparities for each pixel. The problem gets transformed into finding the roots of the operator, one root for each layer that is sought. This problem becomes very fast ill conditioned and is unsolvable in practice for anything more than three sheets.

A filter response method is proposed in [32]. The method uses a combination between the images response to certain quadrature filters and canonical correlation analysis. The filter's phase caries the information on the multiple depths at each pixel. In the end, the problem is equivalent to finding the peaks of the filters. While this method is suitable for finding a small number of layers, like images that contain semitransparent mirrors or views taken through a glass window, but fails for larger numbers of depths.

5.1.3 Results from physics

There have been some attempts in physics to reconstruct solar coronal loops in three dimensions. All approaches produce sparse reconstructions (depths are estimated just for phenomena and not for background or inactive regions). All these methods use multiple EUV images of the Sun separated in time and some use magnetic information as well. They start by detecting the loops through some image processing method, like specialized edge detection filter [33]. Since the output is extremely noisy, a stage of "cleanup" follows usually involving magnetic field modelling. In [34] after applying an edge detection filter, an iterative method that eliminates pixels deemed as being noise and joins/splits features based on the magnetic field magnitude in the region (in principle close to the Canny edge detector). In [35] an extension is presented that takes into account some physical constraints such a curvature together with matching

temperatures of loops.

In [36–38], after applying standard image processing methods, the extrapolated magnetic field is used as a proxy to match features from the two images. A two step minimization method is employed. First stage the features are matched and in second the parameters for the magnetic field get updated (namely the free α constant in the linear, force free model).

The approach presented in [39] does not use any magnetic field information. The method assumes the coronal loops are characterized just by footpoint positions (where the loops disappear inside the photosphere) as well as the vertical and azimuthal angles. Solely image processing techniques are employed, but a full image formation model using all 4 EUV is needed.

The method presented in [40] thresholds the input images as a first stage to distinguish features from static regions. The active parts of the images get reprojected in 3D and intersected in 3D. This approach is similar to the silhouette reconstruction algorithm presented in [41].

5.2 Image formation model

We briefly presented in the previous chapter a linear formation model in the context of reconstruction of fire. While we will use something similar, in our implementation it is important to show the connections with the full emissivity model as given by plasma physics.

5.2.1 Plasma emissivity model

In order to be able to pass from the pixel intensity values in the four available EUV images to the quantities pertinent to physics, one has to establish an image formation model. We have to model how is the passing done from the physical quantities to the luminous energy and then the conversion from energy to pixel values (done by

the sensor/optical filters). The first part of the problem is ill posed. We can assume that the camera model is linear since the images are calibrated by the mission team.

From physics, the observed intensity is given by the plasma radiative transfer law:

$$I(x,y,\lambda) = \int_z \int_T A(T)\Lambda(N_\varepsilon(x,y,z), T(x,y,z), \lambda) N_\varepsilon(x,y,z) N_H(x,y,z) dT dz \quad (5.12)$$

where I is called observed intensity in wavelength λ. $A(T)$ is called the element abundance relative to hydrogen. $\Lambda\left(N_\varepsilon(x,y,z), T(x,y,z), \lambda\right)$ is called the radiative loss function of the plasma and it contains the statistical information, the probabilities to emit light at this certain wavelength given the temperature T, hydrogen density N_H and electron density N_ε. Given the extreme temperatures inside the coronal, one could consider $N_H \approx N_\varepsilon$.

In order to be able to compute the intensity observed at a certain wavelength, we need to reconstruct in 3D both the electron/hydrogen density and temperature profile. The function Λ does not have an analytical form and needs to be computed experimentally. Through a lot of experimentation, the behavior of our four available wavelengths with respect to density and temperature can be computed. This function acts like a convolution kernel on the (z, T) dimensions of N_ε and T. Through various deconvolution techniques we are able to retrieve from $I(x,y)$, $T(x,y)$ and $N_H(x,y)$, the integrals along the line of sight. Notice that the initial problem is not linear where as the last one is. Usually, the quantity of interest is N_H.

Like in the case of fire, matter only emits light and does not absorb, thus having negligible opacity. Also the scattering is negligible as we are computing only along the line of sight (there are more complicated models that take into account also scattering). Unlike the fire model there is no constant self-emissivity as the energy emitted due to high density or high temperature.

For more details on the plasma emissivity and ways to perform the deconvolution refer to [10, 42–44]

5.2.2 Linear model

Since the previous model is fairly complicated to implement, contains another ill posed problem and the convolution kernel's accuracy is still being debated, we will settle for a simpler model. We assume a purely additive linear model in which we have transparent matter in front of an opaque background.

$$I(x,y) = \int_z D(x,y,z)dz + I_{bg} \tag{5.13}$$

The assumptions made are the usual constant emissivity and transparency and negligible scattering.

5.3 Problem statement

We have two or more views of a 3D volume of matter which is represented by its density $D(x,y,z)$. The image formation model is that outlined above and the cameras are looking along the z axis. Our goal is to compute this 3D density distribution such that it projects into our views consistently with the images (photoconsistency). If we discretize the reconstruction volume as a cube with the side of length N, we have N^3 unknowns and only N^2 constraints. Since the images are rectified in a convenient matter we can take one epipolar line at a time in each image (corresponding to an epipolar plane in 3D) and we have N^2 unknowns with N constraints per image. This means that there are a lot of density distributions that projected give the same images, but not all are equally good reconstructions. For this we will also test different constraints like smoothness, sparsity and layer distribution.

5.4 Reconstruction volume

The choice for the shape of the zero disparity surface and the rectification scheme dictates the shape of the reconstruction volume. Normally the zero disparity surface (infinite Z) is a plane fronto-parallel to the reference view and the reconstruction

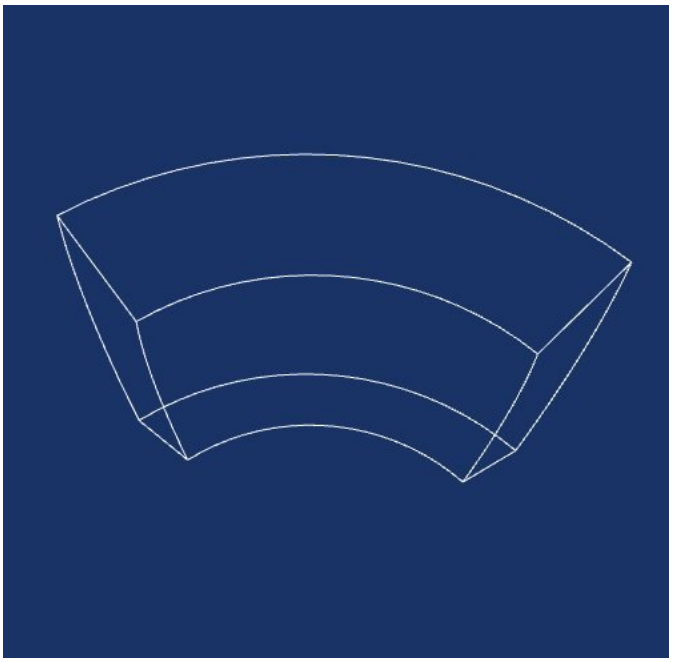

Figure 5.2. Reconstruction volume

surface is a uniformly sampled cube. This does not work in the case of the Sun because of its spherical shape. The zero disparity surface we wish to obtain is the surface of the Sun. The reconstruction volume that we obtain is shown in Fig. 5.2. Since we rectified the images beforehand we can take one epipolar line at a time. In Fig. 5.3 you can see the grid generated by two epipolar lines.

When we rectified the images we said that points on the surface of the Sun are defined by the intersection of ray i from STEREO-A and ray i from STEREO-B, corresponding to a disparity of zero. The points which are k layers above the surface are defined by the intersection between ray i from STEREO-A with ray $i - k$ from STEREO-B, $k < i$. This means that the valid region of reconstruction (on and above the surface of the Sun) is given by the lower half of the square (shown as the shaded region in Fig. 5.3). If each epipolar line has n points there are n^2 points on the full grid and $n(n+1)/2$ in the shaded region.

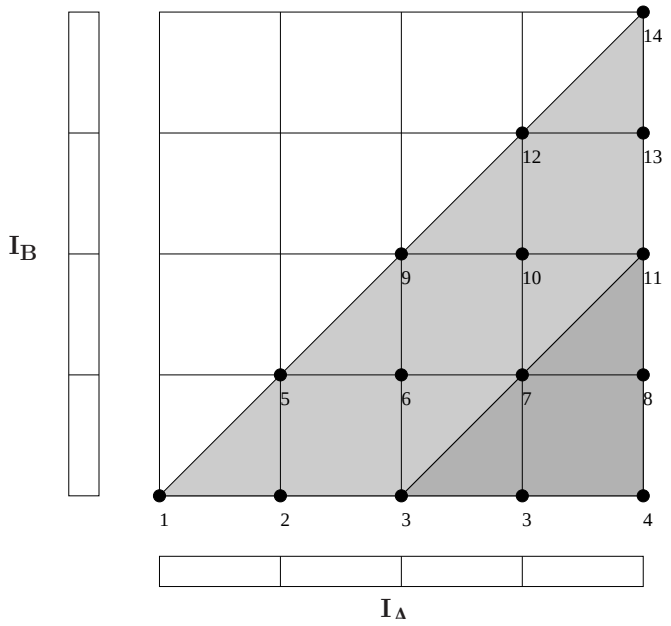

Figure 5.3. Reconstruction grid. Gray is valid region.

Note that the diagonal represents a line on the surface of the Sun and each line parallel to it is further and further from the Sun. The point labeled as 4 has the largest distance from the Sun and is potentially very far. Since the region of interest does not span further than $2R_{Sun}$ it does not make sense to reconstruct the whole possible volume so we will just take up to some maximum distance from the Sun. The new reconstruction grid is shown as light gray in Fig. 5.3.

5.5 The minimization problem

The satellite images I_A and I_B, represent the constraints to the problem. They are the sums along columns and rows of the reconstruction grid. If we collect all the points in the grid in the vector $x = (x_1, x_2, \ldots, x_n)$ we can represent our constraints

as two matrix equations, $S_{row}x = L_A$ and $S_{col}x = L_B$:

$$S_{row} = \left(\begin{array}{c|c|c|c|c|c} & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ I_n & & & & 0 & 0 \\ & I_{n-1} & I_{n-2} & I_{n-3} & 0 & 0 \\ & & & & I_2 & 0 \\ & & & & & 1 \end{array}\right) \quad (5.14)$$

$$S_{col} = \begin{pmatrix} \underbrace{1\ldots1}_{n} & 0\ldots0 & 0\ldots0 \\ 0\ldots0 & \underbrace{1\ldots1}_{n-1} & 0\ldots0 \\ \vdots & & \vdots \\ 0\ldots0 & 0\ldots0 & 1 \end{pmatrix} \quad (5.15)$$

where S_{row} and S_{col} do a sum along rows and columns of our grid and L_A, L_B are the epipolar lines in our two views. If we stack S_{row} and S_{col} on top of each other into A and also stack L_A and L_B into B we get our constraints in the form: $Ax = B$. The matrix has size $2n \times n(n+1)/2$, giving an underconstrained problem.

With this we can express the reconstruction as a constrained optimization problem:

$$\begin{gathered} \text{minimize } f(x) \\ \text{subject to } Ax = B,\ x > 0 \end{gathered}$$

The second positivity constraint, $x > 0$ comes from the fact that all the light gets transmitted and never absorbed. The function $f(x)$ penalizes certain unwanted features of the solution like: low sparsity, high spatial discontinuity, etc.

In practice the problem might be unfeasible (the constraints cannot be satisfied) since the images contain noise and the image formation model is not ideal. In this

case we have two choices: eliminate bad constraints or loosen the constraints. To eliminate the unsatisfiable constraints we do a first stage optimization:

$$\begin{gathered}\text{minimize } \|t\| \\ \text{subject to } Ax = B + t,\ x > 0\end{gathered}$$

The new variable t is a vector with $2n$ entries. We remove one by one columns from A that correspond to the greatest entry in t until the equation $Ax = B$ has a solution. In practice this is not very desirable since we are loosing constraints to an otherwise weakly constrained problem. We can however loosen the constraints and modify the minimization problem:

$$\begin{gathered}\text{minimize } f(x) + \mathbb{I}(\|t\|) \\ \text{subject to } Ax = B + t,\ x > 0\end{gathered}$$

$$\mathbb{I}(x) = \begin{cases} 0 & x = 0 \\ \infty & x \neq 0 \end{cases} \tag{5.16}$$

The indicator function, $\mathbb{I}$ makes it very costly to break the constraints.

5.6 Cost functions

Next we have to investigate possible cost functions. If the cost function is convex, there are a lot of very efficient methods for solving these minimization problems. For a proper background in convex optimization refer to [45]. One of the most popular choice for cost functions are norms. These functions have nice continuity properties and most importantly, they are convex. The convex cost functions will be minimized in practice with the CVX package for Matlab [46].

5.6.1 Choosing the norm

Any function $f : \mathbb{R}^n \to \mathbb{R}$ that has two properties can be considered as a norm function:

- Positive scalability $f(ax) = \|a\|f(x)$
- Triangle inequality $f(x+y) \leq f(x) + f(y)$

As a consequence of the previous two properties, follows that $f(x) = 0$ iff $x = 0$ and $f(x) > 0, \forall x \in \mathbb{R}^n_*$. Common choices for norm functions are:

- l_p norm defined as: $l_p(x) = |x|^p$
- deadzone-linear with deadzone width $a > 0$

$$\phi(x) = \begin{cases} 0 & |x| \leq a \\ |x| - a & |x| > a \end{cases} \tag{5.17}$$

- log barrier with limit $a > 0$

$$\phi(x) = \begin{cases} -a^2 \log\left(1 - (u/a)^2\right) & |x| \leq a \\ \infty & |x| > a \end{cases} \tag{5.18}$$

In Fig. 5.4 you can see the l_1, l_2 norms, the deadzone linear with $a = 0.25$ and the log-barrier with $a = 1$. These cost functions express the penalties we wish to impose on the current errors. Note that if we scale these functions, the final error will be scaled itself, but the solution towards which it converges is identical. The ratio of the penalty for large to small errors gives us the behavior of the cost function.

For residuals in $(-1, 1)$ the l_1 function penalizes more than the l_2 norm. This means that large errors on certain components of x are accepted more than small errors. On the other hand the l_2 norm penalizes a lot large errors compared to small ones. The deadzone error function penalizes just errors that are bigger than some amount linearly. The log-barrier function is similar to the quadratic error for small errors, but has infinite penalty for values larger than a (errors larger than a are unacceptable).

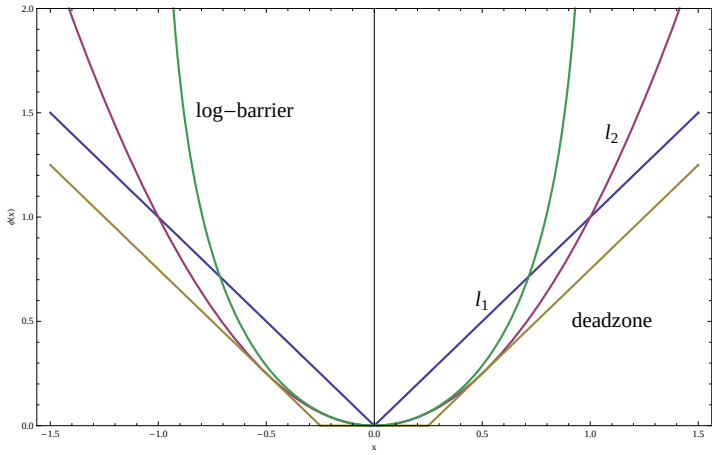

Figure 5.4. Norm functions

5.6.2 Increasing sparsity

Generally we prefer solutions where the matter is as packed as possible and still obey the constraints, meaning we prefer the solution with a high sparsity pattern.

A common practice for learning algorithms where sparsity is a must, is to use the l_0 norm defined as $\sum_i x_i^0$ and define $0^0 = 0$. This is just a measure of number of non-zero elements of x. Is is an abuse of language calling it norm since it is not positive scalable. Also this function is not convex.

Another way to enforce sparsity is by minimizing the l_1 norm. Since it "dislikes" small errors, it is likely to put small residues to zero and accept large ones. However this is not the optimal solution since medium errors are still accepted. To improve we could employ an iterative minimization scheme of a weighted l_1 norm:

1. initialize $W = \frac{1}{x_0}$
2. minimize $|Wx|$ with constraints
3. $W = \frac{1}{x}$

with x_0 being an initial solution which we want to increase sparsity (can be with initial weights all being 1). We iterate between steps 2 and 3 until the sparsity does not improve anymore. We consider a component as being zero when it drops below a certain threshold. Notice that components which are small in the solution will cost a lot and will be eliminated in the next iteration if possible since their weight will be quite high. This iterative scheme does not guarantee that the cost for the solution drops continuously from one iteration to another, but it is guaranteed that the sparsity of the solution will always increase.

An alternative is to maximize the infinity norm $l_\infty = max(x_i)$ with constraints. This is a concave function. We generally choose our cost function as a sum of convex functions. The sum of convex and concave functions is not convex, creating a very hard minimization problem.

There exists also an analytical form for the sparsity of a vector, introduced in [47, 48], $sp : [-1, 1]_*^n \longrightarrow [0, 1]$:

$$sp(x) = \frac{1}{\sqrt{n} - 1} \left(\sqrt{n} - \frac{\|x\|_1}{\|x\|_2} \right) \tag{5.19}$$

This function has the value 0 for a constant vector. For a vector with a single non-zero entry equal to 1 the function equals 1. This function has quite a few drawbacks. It is neither convex, nor concave, so our simple and efficient methods do not work and it has a discontinuity in 0^n.

There are other choices for cost functions that are known to increase sparsity like p-norms with $p < 1$ (including negative p's), Shannon and Gaussian entropy. For a comparison of these functions please refer to [49, 50].

These basic functions provide inspiration but cannot be used directly with our "flattened" grid since there should be no interaction between certain elements (even though they are neighbors in the "flattened" grid as they are not adjacent in our original 2D grid). We will use as cost function the sum of costs for each row in our 2D grid.

Chapter 6

RESULTS

In this section we will present some practical results using the theoretical concepts developed in the previous chapters. We will start by validating the reconstruction techniques using some synthetic examples where the ground truth is known. As a second step we will test the reconstruction on synthetically generated images to be able to reproduce certain situations one would encounter on the Sun. As a last step we will test the reconstruction on real solar images after they have been rectified.

6.1 Synthetic results

6.1.1 Known ground truth

For the synthetic examples with known ground truth we will test the following cost functions based on: l_2 norm, iterative minimization with l_1 norm, and non-convex optimization of the sparsity function from equation 5.19. If we were to apply the previous cost functions directly on our one dimensional flattened grid, there would be an "interaction" between nodes that are adjacent in the flattened grid but not in the original triangular one (for example node 4 and 5 in Fig. 5.3). One simple modification meant to exclude the non-existing interaction between nodes is to consider the sum of cost functions applied to each row in the triangular 2D grid. Since the sparsity measure is defined just for vectors with elements smaller than 1, we will add this extra constraint. Also note that the new analytical sparsity measure takes values in the range $[0, r]$, where r is the number of rows (the function is the sum of row sparsities each taking values in the range $[0, 1]$).

The first synthetic example with known ground truth is shown in Fig. 6.1. The

0.5	-	-	-	-	0.5
0.5	-	-	-	0.5	0
1	-	-	0.5	0	0.5
1		0.5	0	0.5	0
1	0.5	0	0.5	0	0
	0.5	0.5	1	1	1

Figure 6.1. Ground truth. Sparsity of 1.31

0.5	-	-	-	-	0.5
0.5	-	-	-	0.31	0.18
1	-	-	0.45	0.33	0.2
1		0.29	0.35	0.24	0.1
1	0.5	0.2	0.19	0.1	0
	0.5	0.5	1	1	1

Figure 6.2. l_2 norm minimization. Sparsity of 3.28

first step is getting a solution to our problem that obeys the constraints, which will get refined in the next steps. In Fig. 6.2 you can see the solution when we minimize with the l_2 norm. This solution respects the constraints but has very low sparsity (matter is very spread). We take this solution and we iteratively minimize the weighted l_1 norm to increase sparsity. After 4 iterations the result is shown in Fig. 6.3. Notice that this has exactly the same sparsity measure as the ground truth (number of non-zero entries), but the solution is different, but equivalent since the problem is underconstrained. Next we will minimize the sparsity function directly with a non-convex minimizer. The result is shown in Fig. 6.4. Notice that this solution satisfies the same constraints as the true solution and its sparsity is even higher than ground truth's.

0.5	-	-	-	-	0.5
0.5	-	-	-	0.5	0
1	-	-	0	0.5	0.5
1		0.5	0.5	0	0
1	0.5	0	0.5	0	0
	0.5	0.5	1	1	1

Figure 6.3. Iterative minimization with the weighted l_1 norm. Sparsity of 1.31

0.5	-	-	-	-	0.5
0.5	-	-	-	0.5	0
1	-	-	1	0	0
1		0.5	0	0.5	0
1	0.5	0	0	0	0.5
	0.5	0.5	1	1	1

Figure 6.4. Minimization with the non-convex sparsity measure. Sparsity of 0.74

It is easy to observe that the current sparsity enforcement model cannot impose a certain sparsity structure, nor the amount of sparsity each solution has. There exist a lot of equivalent solutions (that obey the same constrains) since the problem is underconstrained, even sparser than the ground truth. In order to improve on this reconstruction, one needs to make more assumptions on the solution (smoothness, structure, etc).

6.1.2 The torus dataset

The next synthetic data experiment is on a synthetically generated half torus against a flat background, as illustrated in Fig. 6.5. This is interesting as it tests the possibility of reconstructing discontinuities, holes inside the semi-transparent object. The minimization problem for this dataset will have the following form:

$$\text{minimize } \alpha\|t\| + \beta f(x)$$
$$\text{subject to } Ax = B + t,\ x > 0$$

The addition of t makes the problem much easier to minimize, as constraints are not always satisfiable. If the parameter α is much bigger than β the final effect is the same without using t, but convergence is faster. The non-convex minimization problem using directly the sparsity measure is prohibitively slow making it useless for any image larger than 20 pixels.

We test the results using the l_2 norm and iterated minimization with weight l_1 norm (the non-convex optimization using the analytical sparsity measure will be omitted).

The results of the l_2 cost function can be seen in Fig. 6.6. As one might expect the solution has extremely low sparsity. There is no need to impose any smoothing conditions as the solution is extremely smooth to start with. Notice that the "hole" inside the half torus cannot be reconstructed as it introduces an strong discontinuity.

Next we will try to iteratively minimize with the l_1 norm. Since this produces extremely sparse solutions and highly discontinuous at the same time, we need to

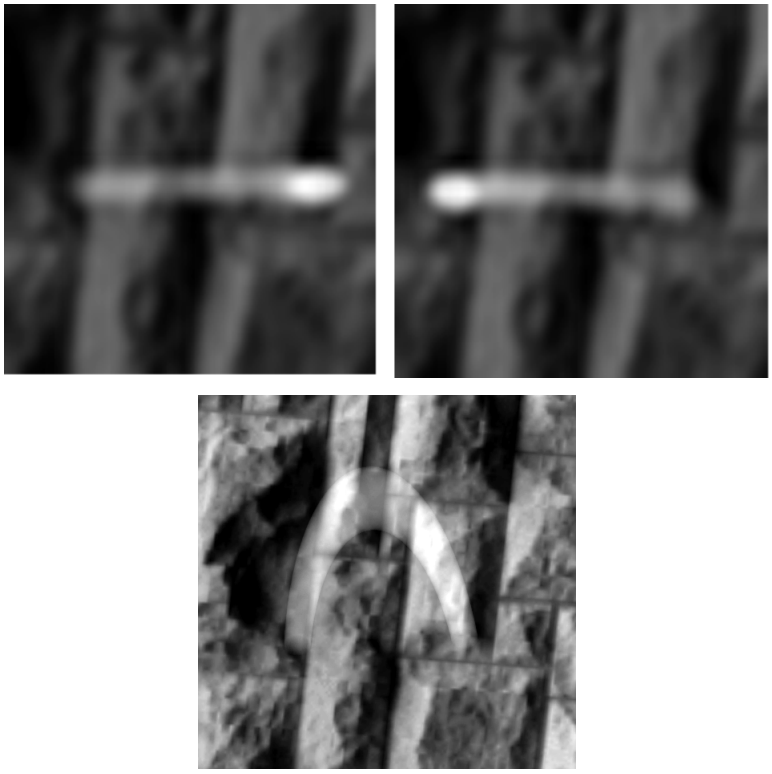

Figure 6.5. Torus dataset: top - the two input views, bottom - sideways view

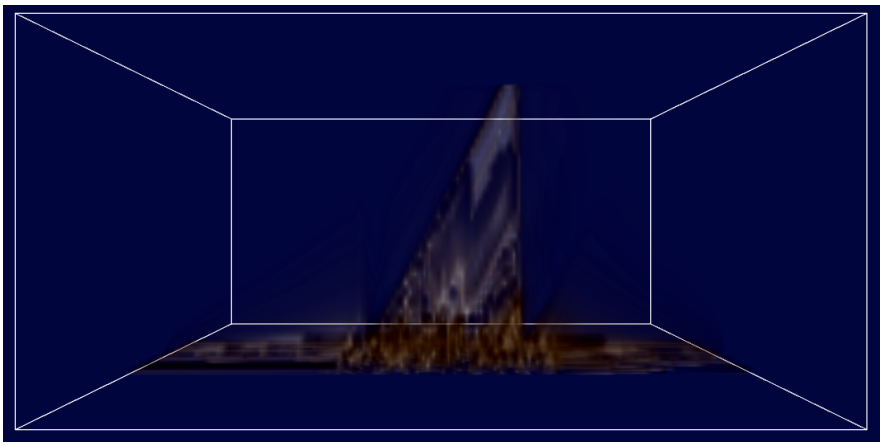

Figure 6.6. Torus reconstruction with l_2 norm

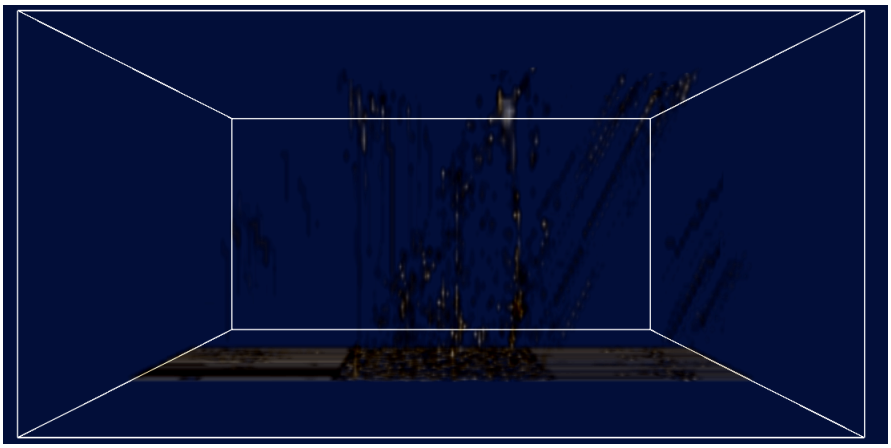

Figure 6.7. Torus reconstruction with iterative minimization

impose as well some smoothing constraints. The lack of smoothing will produce solutions that look very much alike clouds of points. With smoothing, the individual points will be joined together as much as possible. A simple smoothing criteria includes the sum of difference between neighbors along the rows and columns inside our 2D grid. The results can be seen in Fig. 6.7. It is obvious that this solution is much sparser than the previous one. The algorithm converges towards the filiform structures due to the combination of smoothness/sparseness. The vertical and oblique streaks one can observe outline the rays casted from the cameras. In both Fig. 6.6 and 6.7 the reconstruction yields a triangular structure rather than a full torus since it is generally more compact (higher sparsity). The combination of smoothness and sparsity leads to a hysteresis behavior.

6.2 Solar images

In order to test our reconstruction methods on real solar data we will choose a pair of images from STEREO from the 28^{th} July 2007 taken in the $171\mathring{A}$ band. Since the images are fairly big, we will reconstruct just a 200 pixel wide window that contains activity. At this time, the separation was about 22°. This dataset produces the best

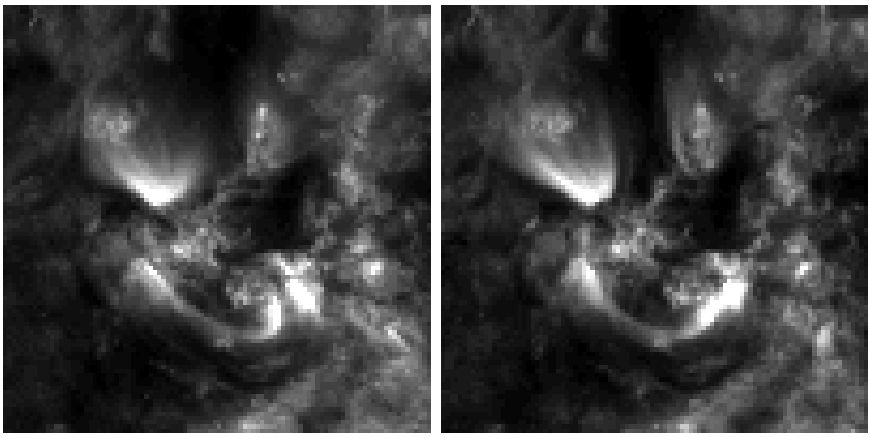

Figure 6.8. Left: STEREO A, Right: STEREO B

results we have obtained to date. We will also try another dataset from the 10^{th} October 2007 to test the effect of bigger separations between satellites. The original images from the $STEREO$ pass through the following stages to yield a reconstruction:

1. compute the camera and correction matrices for internal parameters
2. rectify images
3. reconstruct in 3D a region of the rectified image

6.2.1 Reconstruction

The rectified images are shown in Fig. 6.8. You can notice in the lower part of the image a solar filament that joins two regions that are close together. For solar data we will use the cost function based on the l_1 norm. The iterative minimization is not very suitable as it produces a solution that is too sparse considering the smooth nature of the real data and so far we have not been able to control the degree of sparseness or a certain sparsity pattern of the resulting solution.

In Fig. 6.9 you can see the reconstruction using the l_1 norm. While the top view looks remarkably good, you can notice that the loops are reconstructed as columns.

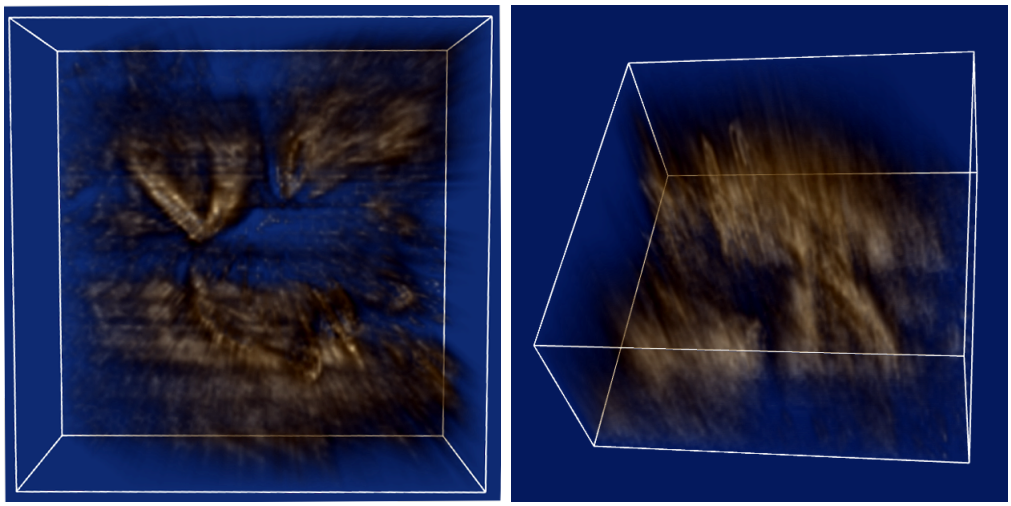

Figure 6.9. Top and oblique views of the reconstruction

Rather than having loops that are connected to the solar surface only at the endpoints, we obtain a solution which is comprised of vertical structures that span the whole hight of the loop. The "hole" inside the torus-like loops represents a discontinuity which is impossible to obtain unless the shape of the structures gets modeled.

We also tried to reconstruct a dataset from the 10^{th} October 2007. At this point the separation between the satellites is almost 40°. For this dataset we also tried a reconstruction using at the same time all three wavelengths that get formed high enough inside the solar atmosphere to show some motion parallax. To do this we just stacked the three epipolar lines on the right hand side of our system and replicated three times the matrix A from equation 5.5. From a physics standpoint this does not make a lot of sense since the same feature might have different apparent displacements between the two views depending on the temperature of the feature. We do this in hope to enforce three times more constraints on the problem. In Fig. 6.10 you can see the rectified images in the three wavelengths.

In Fig. 6.11 you can see the results from the newer dataset using the l_1 norm. It is apparent that this reconstruction is not as good as the previous one. One of the

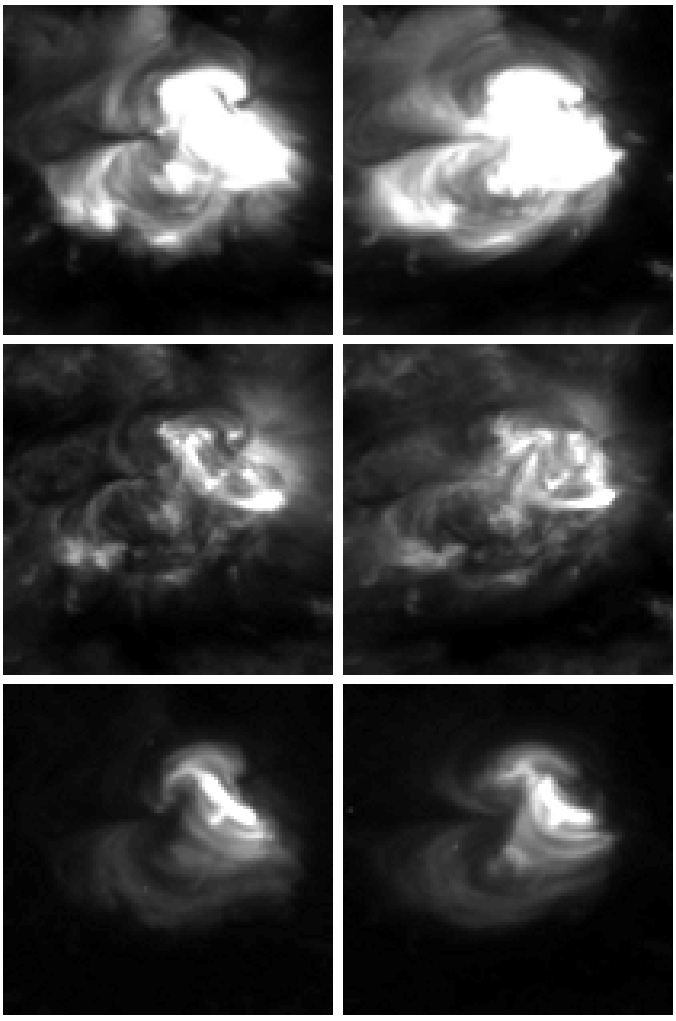

Figure 6.10. Left: STEREO A, Right: STEREO B

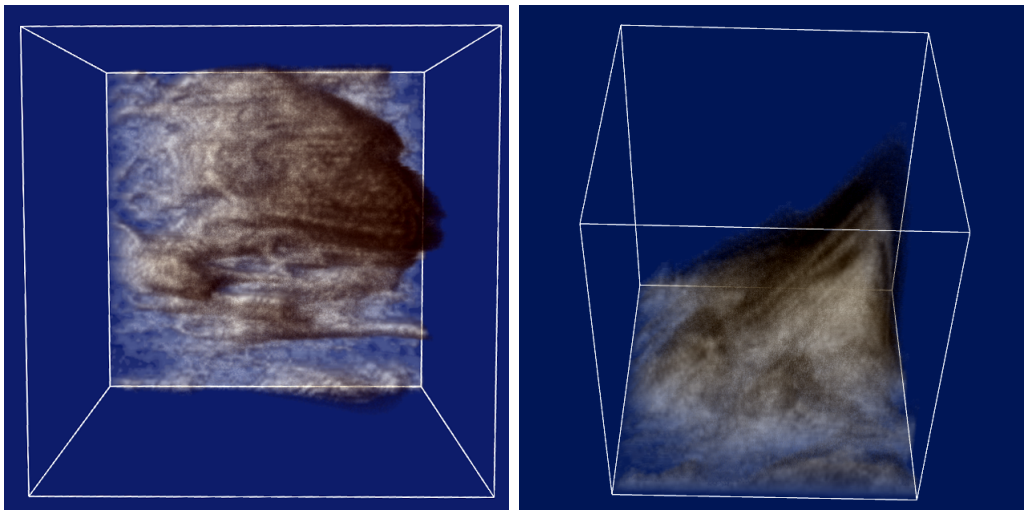

Figure 6.11. Top and oblique view of the reconstruction

reasons is that the bigger separation made it much harder to match regions. Actually one of the reasons for which the satellites were placed on different orbits is to try out different separation angles to determine the optimal angles for reconstruction. This dataset is more complicated than the previous one since there are a lot of features under the loops. The reconstruction barely resembles the input images even when regarded from the most favorable angle.

6.2.2 Motion segmentation

Another interesting application of this project is to segment the featured from the background so that another more specialized reconstruction method can be applied as a second stage. If you remember our reconstruction grid from Fig. 5.3, the diagonal represents the matter on the surface of the Sun. Due to our rectification assumptions, feature on the surface of the Sun should not exhibit any parallax. If we substract from images of the satellites A and B, the image of the stationary background, we should be able to have just the feature from the two views. This is shown in Fig. 6.12

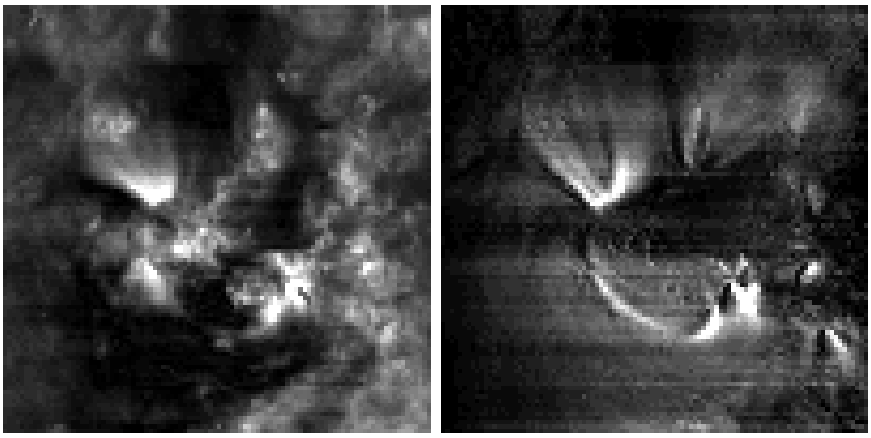

Figure 6.12. Left: Reconstruction of the surface of the Sun, Right: STEREO A minus the surface showing just the moving parts

There is no simple measure of "goodness of fit". One possible visual estimate of the quality of the reconstruction can be obtained by comparing the two rectified views to identify the moving parts and compare it with our segmention results.

Chapter 7

DISCUSSIONS AND CONCLUSIONS

The main contributions of this project included a rectification/high precision alignment method for satellite imagery and a stereoscopic reconstruction method designed for transparent objects.

The reconstruction method is guaranteed to rectify any camera configuration and always giving finite output images. Since the rectification surface is a sphere, the straight lines do not get preserved. The reconstruction grid induced by the rectification method has two important properties:

1. the zero disparity surface is a sphere with radius R_{Sun}.

2. the reconstruction grid is given by the intersection of rays cast from each view.

A consequence of the first property is that objects on the surface of the Sun do not exhibit any motion parallax in the rectified images. The second property guarantees that all grid points inside the reconstruction volume always project onto image pixels, without the need to interpolate. This makes density computations very easy to handle.

The reconstruction method proposed can work with arbitrary transparent images that have been rectified in a similar fashion (objects with motion parallax above an immobile zero disparity surface). The reconstruction volume is limited to one side of the zero disparity surface (with a modified reconstruction grid one can reconstruct both sides), since the zero disparity surface is assured to be opaque.

The method gives a dense reconstruction and makes no assumptions on the shape of reconstructed objects. All a priori information can be introduced in a natural way

as constraints or cost function. Images that are taken at different times can be used to provide extra constraints on the solution.

The method is fairly fast. The Matlab implementation using 100x100 images takes less then 3 minutes to reconstruct using the l_1 norm.

7.1 Further developments

The generality of the algorithm hinders the quality of the solutions. One of the ways to improve the solutions would be to fit some physical model to the data such as magnetic field extrapolation. This would however most likely not produce dense reconstructions as just the modeled phenomenon would be reconstructed.

To provide extra constraints to the problem one could add more images of the two satellites taken at different times. For this the algorithm would require two small modifications:

- due to the rotation of the Sun around its own axis the features will change position over time. We could estimate the amount of rotation for a region and update the translation of the satellites such that the surface of the Sun remains static.

- one could replace the 4 input raw images by reconstructed electron and temperature densities. While intensity values in original images change quickly as there are bursts of activity, the temperature and electron density does not change as fast. This would greatly improve the robustness of the reconstruction.

An extra constraint could be provided by a third view given by SOHO. Unfortunately it is impossible to rectify triplets of images at the same time unless they share the same baseline. This is unfortunately not the case for SOHO, but the method could be modified to accept two rectified pairs (SOHO/STEREO A and SOHO/STEREO B).

All smoothing so far is applied only along epipolar lines. In order to smooth between epipolar lines one would need to iterate between a stage of constrained minimization and inter epipolar line smoothing.

BIBLIOGRAPHY

[1] http://photocreations.ca/radial_distortion/rd_org.jpg.

[2] http://science.jrank.org/pages/6603/Sun-brief-history-solar-observations.html.

[3] M. Stix. *The sun : an introduction*. Berlin ; Springer-Verlag, New York, 1989.

[4] Z. Svestka. *Solar Flares*. Reidl Pusblishing, 1976.

[5] R. Noyes. *The sun, Our Star*. Harvard University Press, 1982.

[6] L. Harra. *Space Science*. Imperial College Press, London, 2004.

[7] V. Bothmer D. Rust, J. Davila. The sun and heliosphere in 3 dimensions, 1997.

[8] W. Thomson. Stereo science operations plan, 2004.

[9] W. T. Thompson. Coordinate systems for solar image data. *Astronomy and Astrophysics*, dec 2005.

[10] Markus Aschwanden. *Physics of the solar corona: An Introduction*. Springer.

[11] Alan M. Title Carolus J. Schrijver, Marc L. DeRosa. The nonpotentiality of active-region coronae and the dynamics of the photospheric magnetic field. *The Astrophysical Journal*, 2005.

[12] S Cuperman. Extrapolation of oblique boundary values. *Astron. Astrophy*, 1990.

[13] N. SEEHAFER. Determination of constant a force-free solar magnetic fields from magnetograph data. *Astron. Astrophy*, 1978.

[14] T Wiegelman. An idl widget program to compute and display linear force free coronal magnetic fields.

[15] R. Hartley. *Multiple View Geometry*. Cambridge Press, Cambridge, 2000.

[16] R. Horaud. *Vision par ordinateur:outil fondamentaux*. 1995.

[17] Takeo Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920 – 932, September 1994.

[18] S. M. Seitz K. N. Kutulakos. A theory of shape by space carving. *International Journal of Computer Vision*, 38:199–218, 2000.

[19] Janne Heikkila. A four-step camera calibration procedure with implicit image correction. *Computer Vision and Pattern Recognition*, 1997.

[20] E. Trucco. *Introductory techniques for 3-D Computer vision*. Prentice Hall, 1998.

[21] S. Roy, J. Meunier, and I. Cox. Cylindrical rectification to minimize epipolar distortion. *In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.393-399*, 1997.

[22] M. Pollefeys. A simple and efficient rectification method for general motion. *Proc. International Conference on Computer Vision*, pages 496–501, 1999.

[23] D Dingrui Wan Jie Zhou Zhang. A spherical rectification for dual-ptz-camera system. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, pages 777–780, 2007.

[24] Akihiko Torii Jun Fujiki and Shotaro Akaho1. Epipolar geometry via rectification of spherical images. *Lecture Notes in Computer Science*, pages 461–471, 2007.

[25] A. Mohammad-Djafari. Image reconstruction of a compact object from a few number of projections. *Proceedings International Conference of Signal and Image Processing*, pages 325–329, 1996.

[26] Vidya Elangovan and Ross T. Whitaker. From sinograms to surfaces: A direct approach to the segmentation of tomographic data. *Lecture Notes in Computer Science*, 2208:213, 2001.

[27] G.T. Herman. *Discrete Tomography: Foundations,Algorithms, and Applications*. Birkhauser, 1999.

[28] Naveen Santhanam, Timothy S. Newman, and Dennis L. Gallagher. Exploiting known latitudinal variations for improved limited-angle tomographic reconstruction of the plasmasphere. In *ITCC '04: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 2*, page 602, Washington, DC, USA, 2004. IEEE Computer Society.

[29] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. In *ICCV*, pages 517–526, 1998.

[30] Photo-consistent reconstruction of semitransparent scenes by density-sheet decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(5):870–885, 2007. Student Member-Samuel W. Hasinoff and Member-Kiriakos N. Kutulakos.

[31] Masahiko Shizawa. Direct estimation of multiple disparitiesl for transparent multiple surfaces in binocular stereo, 1997.

[32] M. Borga and H. Knutsson. Estimating multiple depths in semi-transparent stereo images. *In Proceedings of the Scandinavian Conference on Image analysis, Greenland, June 1999. SCIA. Also as Technical Report LiTH-ISY-R-2248.*, 1999.

[33] `http://www.lmsal.com/~aschwanden/stereo/2000easton/cdaw.html`.

[34] Jong Kwan Lee, Timothy S. Newman, and G. Allen Gary. Automated detection of solar loops by the oriented connectivity method. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4*, pages 315–318, Washington, DC, USA, 2004. IEEE Computer Society.

[35] Markus Aschwanden. 2d feature recognition and 3d reconstruction in solar euv images. *Solar Physics*, 228:339–358(20), May 2005.

[36] T. Wiegelmann. How to use magnetic field information for coronal loop identification. *Solar Physics*, 228:67–78(12), May 2005.

[37] T. Wiegelmann and B. Inhester. Magnetic Stereoscopy. *solphys*, 236:25–40, June 2006.

[38] Wiegelmann T. Including stereoscopic information in the reconstruction of coronal magnetic fields. *Solar Physics*, 208:233–251(19), August 2002.

[39] M. J. Aschwanden, J. S. Newmark, J.-P. Delaboudinière, W. M. Neupert, J. A. Klimchuk, G. A. Gary, F. Portier-Fozzani, and A. Zucker. Three-dimensional Stereoscopic Analysis of Solar Active Region Loops. I. SOHO/EIT Observations at Temperatures of (1.0-1.5) X 10ˆ6 K. *apj*, 515:842–867, April 1999.

[40] B. Inhester. Stereoscopy basics for the STEREO mission. *ArXiv Astrophysics e-prints*, dec 2006.

[41] Martin. Volumetric description of objects from multiple views. *Trans. Pattern Analysis and Machine Intelligence*, page 2, 1991.

[42] E. Landi G. Del Zanna, B. J. I. Bromage. Solar euv spectroscopic observations with soho/cds. *Astron. Astrophys*, 2001.

[43] U. Feldman E. Landi. Chianti-an atomic database for emission lines. v. comparison with an isothermal spectrum observed with sumer. *The Astrophysical Journal Supplement Series*, 2002.

[44] E. Landi and M. Landini. Simultaneous temperature and density diagnostics of optically thin plasmas. *Astron. Astrophys*, 1997.

[45] Lieven Vandenberghe Stephen Boyd. *Convex Optimization*. Cambridge University Press, 2004.

[46] L. Liberti and N. Maculan. *Global Optimization: From Theory to Implementation*. 2006.

[47] M. Heiler. Controlling sparseness in non-negative tensor factorization. *ECCV*, May 2004.

[48] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning*, 5:1457–1469, May 2004.

[49] B. D. Rao. Deriving algorithms for computing sparse solutions to linear inverse problems. *Signals, Systems and Computers, Conference Record of the Thirty-First Asilomar Conference on*, 1997.

[50] B. D. Rao. An affine scaling methodology for best basis selection. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 47, 1999.