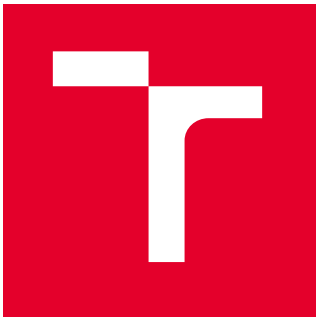


BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING

FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF MATHEMATICS

ÚSTAV MATEMATIKY

SEPARATION OF THE DYNAMIC IMAGE COMPONENT FOR THE SOLAR CORONA PLASMA RESEARCH

SEPARATION OF THE DYNAMIC IMAGE COMPONENT FOR THE SOLAR CORONA PLASMA RESEARCH

DOCTORAL THESIS

DIZERTAČNÍ PRÁCE

AUTHOR — **Ing. Petra Kalenská**

AUTOR PRÁCE

SUPERVISOR — **prof. RNDr. Miloslav Druckmüller, CSc.**

ŠKOLITEL

BRNO 2024

Abstrakt

Tato práce se zabývá rozkladem dynamické a téměř stabilní části sluneční koróny z videozáznamů pozorování Slunce. Nejdůležitějšími událostmi ve sluneční koróně jsou výrony koronální hmoty (CME). Abychom mohli studovat skutečnou povahu CME, je nutné je oddělit od ostatních chaotických pohybů ve vnitřní koróně. Tento proces je značně závislý na impulsním šumu, který je přítomen na všech snímcích z kosmických sond. Proto je prvním cílem této práce najít vhodnou metodu pro filtraci impulsního šumu v digitálních snímcích pořízených kosmickými koronagrafy a vyhodnotit její efektivitu pomocí testování na datech ze sondy SOHO. Dalším cílem je zjistit, zda jsou metody založené na řídké reprezentaci vhodné k odseparování výronů koronální hmoty z videí pořízených sondou SDO a zda jsou jejich výsledky lepší než výsledky z nyní používané metody zvané "running difference". V práci je zahrnuta potřebná teorie z oblasti astronomie, matematiky a registrace obrazu. Dále popisuje momentovou metodu a její použití pro filtraci impulsního šumu. V práci jsou také popsány metody mediánového filtru, analýzy hlavních komponent a metody dynamických modů, a výsledky jejich použití k odseparování výronů koronální hmoty. K práci jsou též přiloženy programy, které byly použity k filtrování impulsního šumu a k dekompozici videí slunečních pozorování.

Abstract

This thesis deals with the decomposition of the dynamic and nearly stable part of the solar corona from solar observation videos. The most important events in the solar corona are coronal mass ejections (CMEs). In order to study the true nature of CMEs, it is necessary to separate them from other chaotic motions in the inner corona. This process relies heavily on the impulse noise that is present in all spacecraft images. Thus, the primary objective of this thesis is to find an optimal method for filtering impulse noise in digital images captured by space-based coronagraphs and to evaluate its efficacy through testing on data obtained from the SOHO spacecraft. The second objective is to determine whether methods based on sparse representations can be employed to differentiate between coronal mass ejections and other phenomena observed in solar videos captured by SDO. Additionally, it is of interest to ascertain whether these methods offer a superior alternative to the current standard, namely running difference. This thesis encompasses astronomical, mathematical and image registration theory, a description of the moment method and its application to the filtration of impulse noise. Additionally, it presents a detailed account of methods based on sparse representations, including the median filter, the principal component pursuit method and dynamic mode decomposition, along with the results of their implementation. Furthermore, the thesis includes the software used for testing the filtration and decomposition of solar observation videos.

klíčová slova
sluneční koróna, CME, SOHO, SDO, impulsní šum, momentová metoda, mediánový filtr, analýza hlavních komponentů, metoda dynamických modů

keywords
solar corona, CME, SOHO, SDO, impulse noise, moment method, median filter, PCP, DMD

Kalenská, P.: *Separation of the dynamic image component for the solar corona plasma research*, Brno University of Technology, Faculty of Mechanical Engineering, 2024. 123 p. Doctoral thesis supervisor prof. RNDr. Miloslav Druckmüller, CSc.

I declare that I have written the doctoral thesis *Separation of the dynamic image component for the solar corona plasma research* on my own according to advice of my doctoral thesis supervisor prof. RNDr. Miloslav Druckmüller, CSc.,and using the sources listed in references.

August 20, 2024 Ing. Petra Kalenská

I would like to thank my supervisor prof. RNDr. Miloslav Druckmüller, CSc. for his guidance, assistance and especially for his big patience and goodwill to explain me some specific astronomical and mathematical problems.

I would also like to thank prof. Mgr. Pavel Rajmic Ph.D. for the idea of using methods based on sparse representations in this thesis.

I am very grateful to my colleagues in the Mathematics Department for their willingness to help me with any problem.

Finally, I would like to thank my family for their mental support.

Ing. Petra Kalenská

Contents

Introduction

Coronal mass ejections (CMEs) represent the most significant dynamic events occurring in the Sun's atmosphere. Such events have the potential to precipitate intense geomagnetic storms. In order to facilitate the study of CMEs, it is essential to distinguish them from other dynamical phenomena in the solar corona, particularly with regard to their velocity, direction, and structure. Although the method of running differences is the most commonly employed technique for the separation of components from the time series of solar observations, it does not offer a comprehensive representation of the actual structure of the CME. This is due to the unavoidable contamination of the results from other dynamics associated with the background solar wind. New methods that utilise images from the Solar and Heliospheric Observatory (SOHO) typically encounter issues with impulse noise in the image. The primary objective of this thesis is to identify a method for filtering out the noise in a manner that is observer-independent, while preserving the structural integrity of the CME.

The advancement of imaging technologies has outpaced the rate of evolution observed in SOHO. As a consequence, SOHO is currently positioned to lag behind in certain respects. A new spacecraft, the Solar Dynamics Observatory (SDO), has been designed with the objective of studying the influence of solar activity on Earth. In contrast to SOHO, which is a coronagraph, SDO is designed to image the entire Sun, not merely the outer corona. This presents a challenge in separating the CME from a time series of solar observations, as the images are subject to significant changes, not just in the CME, but in the overall appearance of the Sun. The second objective of this thesis is to ascertain whether methods based on sparse representations (median filter, Principal Component Pursuit, Dynamic Mode Decomposition) can be employed to differentiate between CMEs and other phenomena observed in solar videos captured by SDO. Additionally, it is of interest to determine whether these methods offer a superior alternative to the current standard.

This thesis is divided to three parts. Part I called Implemetal theory contains three chapters. Chapter 1 deals with the description of the Sun, the solar corona, coronal mass ejections and other neccesary terms from the astronomical background. The mathematical theory presented in Chapter 2 is fundamental to the implementation of methods for filtering impulse noise as well as to the separation of CMEs, and therefore the reader is reminded of the statistical background and the numerical linear algebra concepts that underpin these processes. The Chapter 3 begins with an overview of the theoretical background behind image registration, including the definition of image, image filters and the detailed process of calibration. It then goes on to present a number of image enhancement methods, including adaptive histogram equalization and Noise Adaptive Fuzzy Equalization (NAFE). Part II, Filtration of impulse noise, is divided to two chapters. The initial

chapter, Chapter 4, presents the established methodology for the filtration of impulse noise. The subsequent chapter, Chapter 5, provides a comprehensive theoretical analysis of the Moment method, illustrating its potential applications in filtration, implementation, and outcomes. Part III is entitled Decomposition of the dynamical and nearly stable part of the solar corona from videos. It is comprised of two chapters. The initial chapter, Chapter 6, presents the state-of-the-art methodology for decomposing videos and enhancing CMEs. The subsequent chapter, Chapter 7, provides a comprehensive overview of the Median filter, Principal Component Pursuit and Dynamic Mode Decomposition. Additionally, it outlines the implementation process, encompassing the pre- and post-processing of images by NAFE and adaptive histogram equalization, respectively. This chapter concludes with a presentation of the results of the described methods and an evaluation of their respective merits and shortcomings.

Part I

Implemental theory

Chapter 1

Astronomical background

This thesis is applied in astronomical field, therefore some useful informations and notations from this branch of science are introduced in this chapter. Some terms are taken as known, one can find them in [1]. In the following section is cited from [2], [3], [1b] and [2b], unless otherwise specified.

For sufficient insight to the main ideas of this work the reader should know some specific information about our closest star, which is giving us a heating and light, about *the Sun*.

1.1 The Sun

As said above, the Sun is a star, precisely it is G-type main-sequence star also called as white dwarf. Its astronomical symbol is $\odot$. The Sun's radius $R_\odot$ is approximately 695.5 thousand kilometers. Sun's rotation period is 25.05 Earth-days at its equator and 34.3 Earth-days at the poles.

The Sun is divided into *interior* and *atmosphere*. The solar interior consists of *core*, *radiative zone* and *convective zone*. The solar interior is not important for approach of this work, so its description is skipped. The solar atmosphere can be split into three main parts, which are called *photosphere*, *chromosphere* and *solar corona*. The described division is shown in Figure 1.1.

Photosphere The top part of convective zone is called photosphere sometimes referred to as sphere of light, because most of the visible photons originate there. This part is visible when one looks directly at the sun. It forms the almost sharp edge that can be observed. It is about 600 km thick (not even 0.1% of $R_\odot$), therefore it is considered as really thin layer. The photosphere is the brighest part of the solar atmosphere.

Chromosphere Just above photosphere lies chromosphere sometimes referred to as sphere of colour. It is possible to examine this part only during total solar eclipses or with sophisticated telescopes, because of brightness of the photosphere. This layer is also considered to be quite thin. It has about 1500 km, i.e. about 0.2% of $R_\odot$.

The transition zone is situated on the top of the chromosphere. This layer is not well recognisible and occurs to have chaotic motion. It performs a good separation of different temperatures between hot corona and much cooler chromosphere.

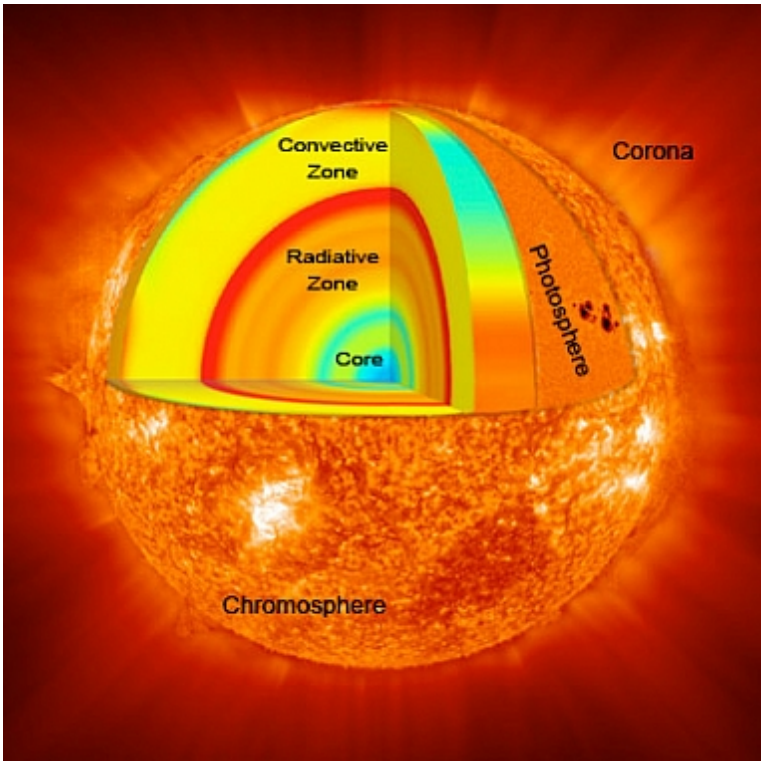


Figure 1.1: The Sun's division. The image was downloaded from [3b].

1.1.1 Solar corona

Solar corona, plasma atmosphere, is the most important part of the solar atmosphere for our approach. Unfortunately, same as the chromosphere, it can be observe only during the total solar eclipse. Because of that the *coronagraph* was invented (see section 1.2).

As said above the solar corona is located beyond the transition zone and continuously expands such that it fills the whole solar system, hence it can be said, that this layer has bigger volume than the Sun itself. In the solar corona atoms become ionized at very high temperature, it causes that the coronal gas is full of free electrons.

The solar corona can be divided into many components, each of them is produced by different light-emitting processes. We will describe only three of them. Other coronal components can be found for example in [4]. The description of following components is cited from [5].

E corona

The E corona is the least bright component of these three, that makes it almost non-observable. This component represents the light from ions in the corona that absorbs light from the photosphere and emits it between the energy levels of these ions. Each ion emits a unique pattern of spectral line, which makes the E corona observable by spectroscope. Due to it, astronomers may examine which elements are contained in coronal gas and get knowledges about their properties. The E corona extends almost until 700 000 kilometers above photosphere ($\sim 2R_{\odot}$).

K corona

It was already said earlier, that coronal gas is full of free electrons, due to this fact the photons of light emitted in photosphere can interact with them. This process causes that the photons are scattered in different directions, it is called *Thomson scattering*. These interactions can make electron-plasma cloud which can travel up to 2 millions kilometres above photosphere ($\sim 3R_{\odot}$). The K corona is also called continuum corona, because it emits continuum of light frequencies.

F corona

This component of the solar corona is the most mysterious one, because after $2R_{\odot}$ the hot ions from E corona disappear and some new gas appears. That gas has same spectral lines as the solar photosphere obtained by Fraunhofer[1], that is why this component is called F corona. During studies of F corona at infrared wavelengths, astrophysicists found out that it is made of interplanetary dust grains, which behave as mirrors of the photosphere. The F corona was detected up to 5 million kilometres above photosphere ($\sim 7R_{\odot}$) and its brightness declined nearly the inverse-square of the distance above the Sun. This coronal component is commonly observed by telescopes with infrared filters during total solar eclipses.

It is important to study phenomena on the solar corona, such as *coronal mass ejection* (CME), because they have impact on space weather. CME can cause geomagnetic storm, which may evoke really big problem on Earth[2]. In the following section is cited from [7] and [8].

1.1.2 Coronal mass ejection

CME is large expulsion of plasma and magnetic field from the solar corona, that propagates outward into interplanetary space and expands in size with distance from the Sun (see Figure 1.2). Its speed varies between 250 km/s and 3000 km/s. The fastest Earth directed CME can reach our planet in 15 hours, the slowest one travels several days. The Earth directed CME is also called *halo CME*, because as it reaches Earth, it appears to be larger than the Sun and it makes bright coronal emission 'halo' around it.

Important parameters for analysis of CME are size, speed and direction, which are nowdays obtain from coronagraphs (see section 1.2). Before invention of coronagraph, it was possible to observe CME only if it appeared during the total solar eclipse. The occurence rate of CME follows the 11-year solar cycle of solar activity. Around solar maximum CME occurs more frequently.

Taking photos of solar corona is the most complicated task in astrophotography, not only because it is possible to take the photos just during the total solar eclipse or through the coronagraph, but also because of the *brightness radial gradient*, which makes the postprocessing more difficult. More about this feature is written in the next section.

[1] The results of Fraunhofer experiment can be found in [6].

[2] Good ilustration what can happen is in [4b].

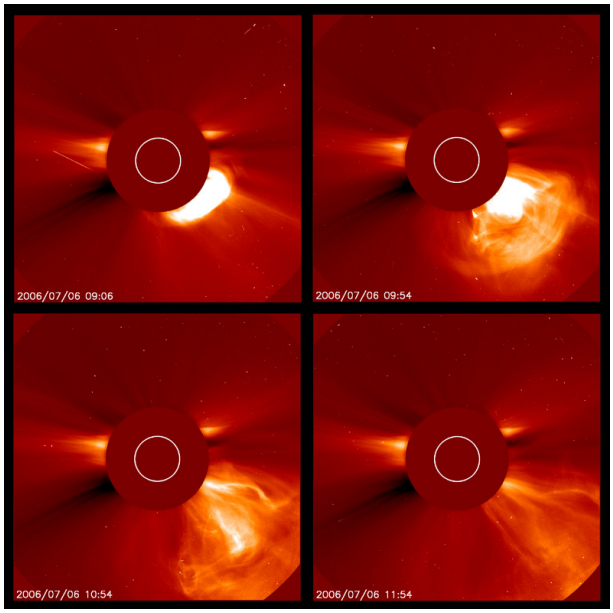


Figure 1.2: Sequence of coronagraph images with propagation of CME on July 7, 2006. The image was downloaded from [5b].

1.1.3 Brightness radial gradient

The brightness of the corona is approximately exponentially decreasing with height above the photosphere, this tendency can be seen on graphs in [9]. It is known that the light intensity is proportional to the *electron density* n_e in the corona. As is written in [10], the exponential decay of n_e can be approximated for the few solar radii $R_\odot$ by the Baumbach-Allen formula as

$$n_e(\zeta) = 10^8 \left(\frac{2.99}{(1+\zeta)^{16}} + \frac{1.55}{(1+\zeta)^6} + \frac{0.036}{(1+\zeta)^{1.5}} \right) \ cm^{-3},$$

where $\zeta = \dfrac{h}{R_\odot}$ is normalized height above the solar surface.

The main problem which is caused by brightness gradient is not capturing of the whole brightness range, but the technical difficulty in visualisation for the human eye. There is no visual device which could be able to display such high number of brightness levels. For that reason, scientists need methods which can lower the dynamic range of the image such that the transformed image will have maximum of 256 brightness levels (for definition of the image see section 3.1). The original high dynamic range image is still needed, because all the scientific measurements are made on it.

Methods of compensation

At time, when it was not possible to use computers for image processing, mechanical and optical methods for compensation of brightness gradient were invented.These historical methods are described in [3] and [11].

Mathematical methods of compensation of brigthness gradient are for example Adaptive Circular High-pass Filter, Fourier normalizing radial graded filter (more about these two methods can be found in [3]) and Normalizing-radial-graded filter (this method and

some of its improvements can be found in [2]).

Thereby the reader knows useful informations about the Sun and its components. Following sections talks about the spacecrafts used for imaging the solar corona. It will be cited from [6b] and [7b], unless otherwise specified.

1.2 Coronagraphs

A coronagraph blocks light from the center of the telescope beam and permits light from surrounding sources. It was invented by the French astronomer Bernard Lyot in 1939, for observing the solar corona also in time without total solar eclipse. The coronagraph is made up of mirrors, occulting spot, also called as occoulter (which absorbs most of the light from the photosphere) and Lyot spot (which block remaining light from the Sun). For better visualization see Figure 1.3.

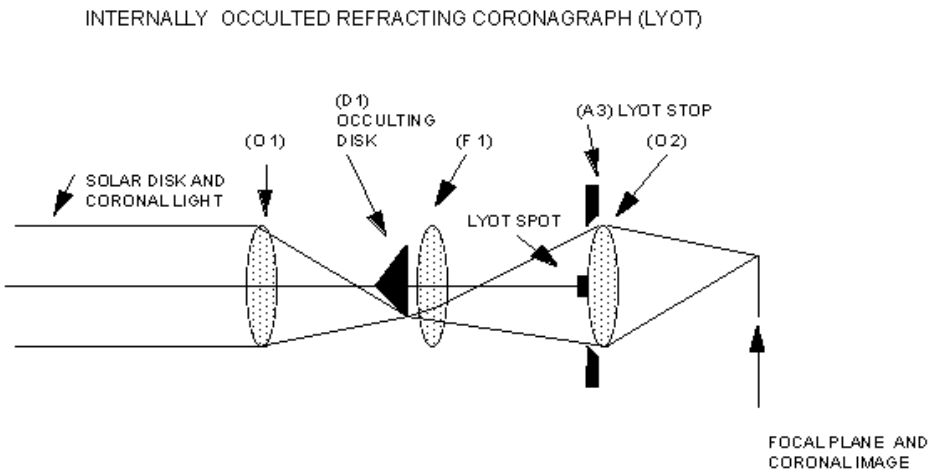


Figure 1.3: Model of Lyot coronagraph. The image was downloaded from [8b].

The coronagraphs can be divided to two groups - ground based coronagraphs and spacecrafts coronagraphs. The second one are more interensting for approach of this thesis.

1.2.1 The Solar and Heliospheric Observatory

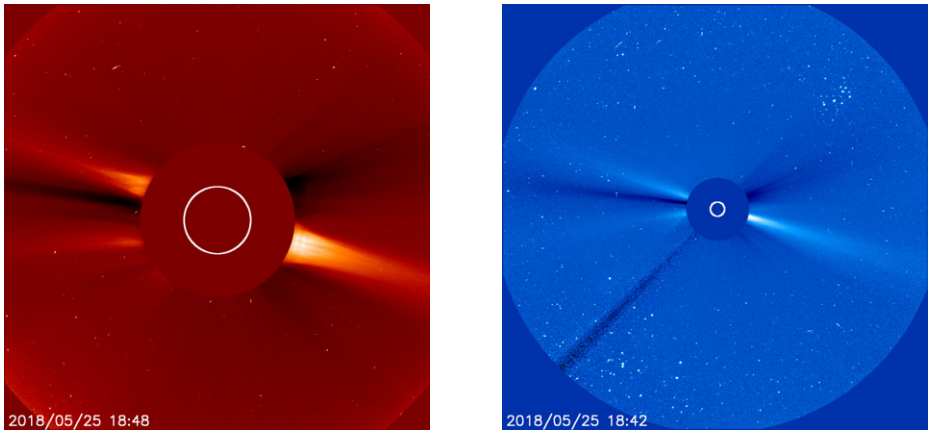
The Solar and Heliospheric Observatory (SOHO) is a project of international cooperation between ESA and NASA to study the Sun, from its deep core to the outer corona. It was launched by NASA from the Cape Canaveral Air Station on December 2, 1995. At the start of the program, it was planned that SOHO will be two years mission, but because it had big success, the mission was extended many times and it is still observing after 28 years. SOHO's key results are providing the most detailed and precise measurements of the temperature structure, improving of space weather forecast and discovering new dynamic solar phenomena such as coronal waves and solar tornadoes.

SOHO is a first three-axis stabilised spacecraft that constantly faces the Sun, which is operated without gyroscopes. It consists of two modules. The service module provides thrusters, power and communications. The payload module carries 12 instrument

packages, such as instruments for coronal spectrometry, solar oscillation measurement, radiation measurement etc. Among others, we can find there also Large Angle and Spectrometric Coronograph (LASCO).

LASCO

The LASCO instrument was a set of three coronagraphs that image the solar corona from 1.1 to $32R_{\odot}$. Now only two of them are working, namely C2 (covers distance range of 1.5 to 6 $R_{\odot}$) and C3 (range of 3 to 32 $R_{\odot}$). The LASCO instrument is currently primary used by forecasters to analyze and categorize CMEs. The images taken by individual coronagraphs are shown in Figure 1.4 (This paragraph was cited from [9b])



(a) (b)

Figure 1.4: Images of solar corona taken by LASCO C2 (a) and LASCO C3 (b). Downloaded from [10b]

Images from SOHO are stored in standard astronomic format called Flexible Image Transport System (FITS). Complete information about this format can be found in [12]. The paragraph below is extract from mentioned reference.

FITS

This media type was designed in order to ease the interchange of astronomical image data between observatories. Most of the technical details of the first basic FITS were developed by Don Wells and Eric Greise in March 1979. The first papers that defined the FITS format were published in 1981, after it FITS quickly became the standard for data interchange within the astronomical community.

FITS consists of header and data array. In the header, there is full description of data, such as used filters, date of observation, info about occoulter, name of telescope, intrument etc. Below this text, one can see a printout of the header of FITS image, which was used for testing created program (see Appendix A on page 121). The explanation of used variables is written in [12].

```
SIMPLE  =  T
BITPIX  =  16
NAXIS   =  2
```

```
NAXIS1  =  1024
NAXIS2  =  1024
BZERO   =  32768.0
BSCALE  =  1.0
DATAMIN =  0.0
DATAMAX =  65535.0
DATE-OBS= '2000-02-26T00:54:06.011'
TELESCOP= 'SOHO'
INSTRUME= 'LASCO'
DETECTOR= 'C2'
FILTER  = 'Orange'
POLAR   = 'Clear'
EXPTIME = '25.000000
CRPIX1  =  511.634000
CRPIX2  =  504.293000
CROTA2  =  0.000000
CDELT1  =  11.900000
CDELT2  =  11.900000
R_SUN   =  81.655461
OCC_X   =  511.500000
OCC_Y   =  515.500000
```

The images from SOHO are used in the Part II of this thesis. It is necessary to introduce one more spacecraft, namely *The Solar Dynamics Observatory* (SDO), since images taken by SDO are used in the Part III.

The following section is cited from [11b] and [12b].

1.3 The Solar Dynamics Observatory

SDO is designed to study the solar influence on Earth and Near-Earth space by studying the solar atmosphere on small scales of space and time and in many wavelengths simultaneously. It was launched on February 11, 2010, 10:23 am EST from Cape Canavera. Generally the goals of SDO are firstly studying the generation and structure of the Sun's magnetic field, secondly research on conversion and release of the magnetic energy into the heliosphere in the form of solar wind, energetic particles etc.

SDO contains a suite of instruments that provide different kinds of observations. Namely, Helioseismic and Magnetic Imager (HMI), Atmospheric Imaging Assembly (AIA) and Extreme Ultraviolet Variability Experiment (EVE). The setting of these instruments on SDO is depicted on Figure 1.5. All of these instruments are very compelling, but AIA is the only one, which is necessary for approach of this thesis.

The Atmospheric Imaging Asembly

The AIA is designed to provide a view of the solar corona, taking images that span at least $1.3R_{\odot}$ in multiple wavelengths, see Table 1.1, and at a cadence of 10 seconds. The values of the wavelength channels are expressed in Ångström (Å). The value of one Ångström corresponds to 100 pm i.e. 0.1 nm i.e. 10^{-10} m. The AIA produces data required for quantitative studies of the evolving coronal magnetic field, during flares and eruptions.

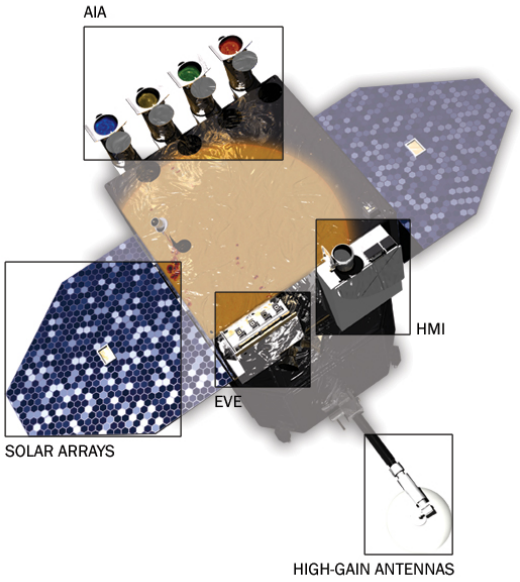


Figure 1.5: Illustration of SDO's intstruments. The image was downloaded from [11b].

The AIA science investigation aims to utilize these data in a comprehensive research program to provide new understanding of the observed processes. The images taken by AIA are stored in FITS format (see section 1.2.1)

Table 1.1: AIA wavelength bands. Information provided by [12b].

Wavelength channel	Primary ion(s)	Region of Solar atmosphere	Characteristic tempretature
white light	continuum	photosphere	5000 K
1700 Å	continuum	temperature minimum, photosphere	5000 K
304 Å	He II	chromosphere, transition region	500004 K
1600 Å	C IV + cont.	transition region + upper photosphere	10^5&5000 K
171 Å	Fe IX	quiet corona, upper transition region	$6.3 \cdot 10^5$ K
193 Å	Fe XII Fe XXIV	corona and hot flare plasma	$1.2 \cdot 10^6$ K & $2 \cdot 10^7$ K
211 Å	Fe XIV	active-region corona	$2 \cdot 10^6$ K
335 Å	Fe XVI	active-region corona	$2.5 \cdot 10^6$ K
94 Å	Fe XVII	flaring regions (partial readout possible)	$6.3 \cdot 10^6$ K
131 Å	Fe VIII Fe XX Fe XXIII	flaring regions (partial readout possible)	$4 \cdot 10^5$ K, 10^7 K & $1.6 \cdot 10^7$ K

Chapter 2

Mathematical background

This thesis deals with two problems, filtration of impulse noise (see Part II) and decomposition of dynamical and nearly stable part of the solar corona from videos (see Part III). Each part needs a different mathematical branch. Firstly, the useful statistical definitions are presented. Then, the definitions and theorems from numerical linear algebra are stated.

2.1 Statistical background

The following section introduces essential statistical theory for our approach such as normal distribution, tolerance intervals, hypothesis testing etc. Observation of this chapter are used mainly in filtration of impulse noise (see Part II).

It is assumed that the basics of statistics are well-known. If one needs to remind them, then it is recommended to read [13] and [14], which are cited in this section.

2.1.1 Normal distribution

The normal distribution, which is also called Gauss distribution is the most frequently used distribution in statistics. It is consider to be the most important one, because many common things in our lives are normally distributed such as height of population, inteligency quocient etc. Therefore, it is presented in this thesis.

Definition 2.1. (Normal "Gauss" distribution) We say a random variable X has a *normal distribution* if its probability density function (PDF) is

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \qquad for -\infty < x < \infty.$$

The parameter μ is called *mean* of X and σ^2 is called *variance* of X. We often write that X has a $N(\mu, \sigma^2)$ distribution.

2.1.2 Hypothesis testing

The filtration of impulse noise (see Part II) is mainly based on hypothesis testing. Therefore, the following text describes the basics of this mathematical field.

At the begining of this part, the used notions are described. Later on, the procedure of hypothesis testing is shown.

Basic notions

Definition 2.2. (A statistical hypothesis) *A statistical hypothesis* is a statement about the parameters of one or more populations.

Definition 2.3. (The null hypothesis, the alternative hypothesis) Suppose that a random variable X has density function $f(x;\theta)$, where $\theta \in \Omega$ for a specified set Ω. Suppose, due to theory or a preliminary experiment, that $\theta \in \omega_0$ or $\theta \in \omega_1$, where ω_0 and ω_1 are disjoint subsets of Ω and $\omega_0 \cup \omega_1 = \Omega$. These hypotheses are labeled as

$$H_0 : \theta \in \omega_0 \quad \text{versus} \quad H_1 : \theta \in \omega_1 \ .$$

The hypothesis H_0 is referred to as *the null hypothesis*, while H_1 is referred to as *the alternative hypothesis*.

Remark 2.4. Often the null hypothesis represents no change or difference from the past, while the alternative hypothesis represents the contrary. The decision rule to take H_0 or H_1 is based on a sample $X_1, \ldots, X_n$ from the distribution of X and, hence, the decision could be wrong.

Definition 2.5. (**The critical region**) Assume that $X_1, \ldots, X_n$ is a random sample from the distribution of a random variable X that has density $f(x;\theta)$, where $\theta \in \Omega$. Further assume that following hypotheses are tested

$$H_0 : \theta \in \omega_0 \quad \text{versus} \quad H_1 : \theta \in \omega_1, \quad \text{where} \quad \omega_0 \cap \omega_1 = \Omega.$$

Denote the space of the sample by $\mathcal{D}$, ergo $\mathcal{D} = \text{space}(\text{X}_1, \ldots, \text{X}_\text{n})$. Then *a test* of H_0 versus H_1 is based on a subset $\mathcal{C}$ of $\mathcal{D}$ and this set $\mathcal{C}$ is called the *critical region* and its corresponding test is

$$\begin{aligned} &\text{Reject } H_0 \quad (\text{Accept } H_1), \text{ if } (X_1, \ldots, X_n) \in \mathcal{C}. \\ &\text{Retain } H_0 \quad (\text{Reject } H_1), \text{ if } (X_1, \ldots, X_n) \in \mathcal{C}^c. \end{aligned}$$

The results of the hypothesis test in terms of the true state of nature, using the critical region from previous definition, are shown in the Table 2.1. It can be seen, that the test can lead to two wrong conclusions. Rejecting the null hypothesis H_0 when it is true, is defined as a *type I error*. Failing to reject the null hypothesis when it is false, is defined as a *type II error*.

	True State of Nature	
Decision	H_0 is true	H_1 is true
Reject H_0	Type I error	Correct decision
Accept H_0	Correct decision	Type II error

Table 2.1: Decision table for hypothesis test

Definition 2.6. (**The significance level**) The probability of making a type I error is denoted as

$$\alpha = P(\text{making type I error}).$$

The value of α is called *the significance level*, or the size of the test.

Definition 2.7. (**The power of the test**) The probability of making a type II error is denoted as

$$\beta = P(\text{making type II error}).$$

The value of β is sometimes called *β-error*. The probability of correctly rejecting a false null hypothesis is computed as $1 - \beta$ and it is called *the power of the test*.

Definition 2.8. (**One-sided/two-sided hypotheses tests**) A test of any hypothesis such as

$$H_0 : \mu = \mu_0 \quad \text{versus} \quad H_1 : \mu \neq \mu_0$$

is called *a two-sided test*. Such hypothesis is called *a two-sided hypothesis*. A test of any hypothesis such as

$$\begin{aligned} H_0 : \mu = \mu_0 \qquad &\text{versus} \qquad H_1 : \mu > \mu_0 \\ &\text{or} \\ H_0 : \mu = \mu_0 \qquad &\text{versus} \qquad H_1 : \mu < \mu_0 \end{aligned}$$

is called *a one-sided test* (sometimes it is called *one-tailed test*). Such hypotheses are called *one-sided hypotheses*.

Remark 2.9. In a two-sided test, the critical region is split into two parts, with (usually) equal probability placed in each tail of the distribution of the test statistic. If the alternative hypothesis is $H_1 : \mu > \mu_0$, the critical region should lie in the upper tail of the distribution of the test statistic, whereas if the alternative hypothesis is $H_1 : \mu < \mu_0$ the critical region should lie in the lower tail of the distribution.

The general procedure of hypothesis testing

Previous section introduced the basic notions of hypotheses testing. In the following text, the procedure for hypothesis testing is briefly described.

1. Identification of the parameter of interest, based on the problem context.
2. Assesment of the null hypothesis. (It is advisible to state the null hypothesis as an equality, because in this case the significance level can be controlled at a specific value.)
3. Assesment of an appropriate alternative hypothesis. (one-sided or two-sided)
4. Choice of desired significance level α.
5. Choice of an appropriate test statistic (t value, z value, F value etc. see [13] or [14])
6. Specification of the critical region.
7. Computation of all necessary sample quantities from the equation for the test statistic and computation of its value.
8. Decision on whether or not to reject the null hypothesis.

2.1.3 Tolerance interval

The training of the moment method (see Chapter 5) is based on calculating of the tolerance interval. Since the variables in that chapter are assumed to have Normal distribution, only the definition of tolerance interval for normal distribution is written below. It is cited from [15].

Definition 2.10. (**100p% tolerance interval with confidence level** $1-\alpha$) Let measurements $x_1, x_2, \ldots, x_n$ be values of a random sample $X_1, X_2, \ldots, X_n$ of size n from a Normal distribution with unknown mean μ and unknown variance σ^2 that is

$$X_i \sim N(\mu, \sigma^2), \quad i = 1, 2, \ldots, n; \quad \mu \text{ and } \sigma - \text{unknown}.$$

The $100p\%$ two-sided statistical tolerance interval with confidence level $1-\alpha$ is constructed by

$$(\bar{x} - ks, \bar{x} + ks)$$

for which the following equation is valid

$$P[P\left(\bar{x} - ks < X < \bar{x} + ks \geq p\right)] = 1 - \alpha$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is sample mean (estimate of the μ), $s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$ is sample standard deviation (estimate o f the σ) and $k = k(n, \nu = n-1, p, 1-\alpha)$ is tolerance factor.

Remark 2.11. The definition looks simple, but the computation of precise values of the tolerance factors k is fairly difficult. The commonly used factors can be found in [14] in Appendix A Table XI.

2.2 Numerical linear algebra concepts

This section deals with segment of theory around matrices. It is consider to be part of basic numerical linear algebra, however it is important to repeat these definitions, since the Part III is heavily using these knowledge. It is cited from [16], [17] and [18].

2.2.1 Matrix norms

First of all, the key terms and the specific matrix need to be defined.

Remark 2.12. Since it is not required to distinguish between $\mathbb{R}$ and $\mathbb{C}$ in following definitions, the symbol $\mathbb{K}$ is used as a generalized notation. The set of all matrices of size $m \times n$ (m rows and n columns) is denoted by $\mathcal{M}_{m,n}(\mathbb{K})$, the set of square matrices of size n is denoted by $\mathcal{M}_n(\mathbb{K})$.

Definition 2.13. (Rank of the matrix) Assume that matrix $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{K})$. The image of $\mathbf{A}$ (Im $\mathbf{A}$) is the set of vectors $\mathbf{y} \in \mathbb{K}^m$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$, with $\mathbf{x} \in \mathbb{K}^n$. The dimension of the linear space Im $\mathbf{A}$ is called *the rank* of $\mathbf{A}$.

Definition 2.14. (Identity matrix) A matrix $\mathbf{E} \in \mathcal{M}_n(\mathbb{K})$ with entries $(e_{i,j} = \delta_{i,j})_{1 \leq i,j \leq n}$ where $\delta_{i,j}$ is the Kronecker symbol taking the values $\delta_{i,i} = 1$ and $\delta_{i,j} = 0$ if $i = j$ is called *the identity matrix.*

Definition 2.15. (Adjoint matrix, Hermitian matrix) Let $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{K})$. The matrix $\mathbf{A}^* \in \mathcal{M}_{n,m}(\mathbb{K})$, defined by $\mathbf{A}^* = \bar{\mathbf{A}}^T = (\bar{a}_{j,i})_{1\leq i\leq m, 1\leq j\leq n}$, where $\bar{a}$ denotes complex conjugate of a is called *the adjoint matrix* of $\mathbf{A}$. If $\mathbf{A} = \mathbf{A}^*$ then $\mathbf{A}$ is called *Hermitian matrix.*

Remark 2.16. According to the definition of the adjoint matrix, the Hermitian matrix $\mathbf{A}$ must be a square matrix, moreover if the matrix $\mathbf{A}$ is real, then it is also said to be symmetric ($\mathbf{A} = \mathbf{A}^T$).

Definition 2.17. (Unitary matrix) A matrix $\mathbf{A} \in \mathcal{M}_n(\mathbb{K})$ is said to be *unitary* if

$$\mathbf{A}^*\mathbf{A} = \mathbf{E}.$$

Before introducing matrix norms, it is necessary to repeat the definition of vector norms. Vector norms are a way of measuring the length or magnitude of a vector, and are defined as follows.

Definition 2.18. (Norm on vector space) We call a mapping $\|\cdot\| : \mathbb{K}^n \to \mathbb{R}^+$ satisfying the following properties *a norm* on vector space $\mathbb{K}^n$

1. $\forall \mathbf{x} \in \mathbb{K}^n,\ \|\mathbf{x}\| = 0 \iff \mathbf{x} = \mathbf{0}$
2. $\forall \mathbf{x} \in \mathbb{K}^n,\ \forall \lambda \in \mathbb{K}, \|\lambda\mathbf{x}\| = |\lambda|\|\mathbf{x}\|$
3. $\forall \mathbf{x}, \mathbf{y} \in \mathbb{K}^n,\ \|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$

Remark 2.19. To enable the use of all vector norms for matrices, it is sufficient to convert a given matrix $\mathbf{A}$ into a vector (noted by vec $\mathbf{A}$). A matrix can be vectorised by stacking its columns, resulting in one long column vector. The norm of the matrix can be calculated as $\|\mathbf{A}\| = \|\text{vec } \mathbf{A}\|$.

Definition 2.20. (Matrix norm) A norm $\|\cdot\|$ defined on $\mathcal{M}_n(\mathbb{K})$ is a *matrix norm* if $\forall \mathbf{A}, \mathbf{B} \in \mathcal{M}_n(\mathbb{K})$, holds that

$$\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|. \tag{2.1}$$

Definition 2.21. (Subordinate matrix norm) Let $\|\cdot\|$ be a vector norm on $\mathbb{K}^n$. It induced a matrix norm defined by

$$\|\mathbf{A}\| = \sup_{\mathbf{x}\in\mathbb{K}^n,\ \mathbf{x}\neq 0} \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|}.$$

which is said to be *subordinate* to this vector norm.

Remark 2.22. It can be seen that the subordinate matrix norm is indeed a matrix norm on $\mathcal{M}_n(\mathbb{K})$ since the inequality (2.1) is satisfied. However, there are matrix norms that are not subordinate to any vector norm. A well-known example is the Frobenius norm.

Definition 2.23. (Frobenius norm) Assume that a matrix $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{K})$, then the *Frobenius norm* of matrix $\mathbf{A}$ is defined as

$$\|\mathbf{A}\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{i,j}|^2}.$$

A commonly used group of vector norms are the p-norms. Their subordinate matrix norms are defined below.

Definition 2.24. **(ℓ_p norm)** Assume that $p \in \langle 1, \infty)$ and matrix $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{K})$, then the *$\ell_p$ norm* of matrix $\mathbf{A}$ is as follows.

$$\|\mathbf{A}\|_p = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{i,j}|^p \right)^{\frac{1}{p}}$$

Definition 2.25. **(ℓ_∞ norm)** Assume that a matrix $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{K})$. *The ℓ_∞ norm* of matrix $\mathbf{A}$ is defined as

$$\|\mathbf{A}\| = \max_{1 \le i \le m,\ 1 \le j \le n} |a_{i,j}| \ .$$

Definition 2.26. **(ℓ_0 norm)** Assume that a matrix $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{K})$. The vectorization of matrix $\mathbf{A}$ is denoted by $\mathbf{a}$, thus $\mathbf{a} = \text{vec } \mathbf{A}$. Then the *$\ell_0$ norm* of matrix $\mathbf{A}$ is as follows

$$\|\mathbf{a}\|_0 = |\text{supp}(\mathbf{a})|,$$

where $\text{supp}(\mathbf{a})$ denotes a support of vector $\mathbf{a}$ (indexes of nonzero elements).

Remark 2.27. One can see, that the ℓ_0 norm is not norm according to Definition 2.18, since the condition 2 is not satisfied. However it is known under this name.

Part III is based on video separation using sparse representation methods, which decompose the video matrix into matrices that are low-rank and sparse. Thus, the definition of the sparse matrix is important for our research, it looks as follows.

Definition 2.28. **(Sparse matrix)** A matrix $\mathbf{A}$ is called *k-sparse matrix*, if it holds that

$$\|\text{vec } \mathbf{A}\|_0 \le k,$$

namely matrix $\mathbf{A}$ has maximally k nonzero values.

To transform optimization problem (7.1) from Part III into a convex problem, the definition of one extra matrix norm, namely the nuclear norm, is needed.

Definition 2.29. **(Nuclear norm)** Assume that a matrix $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{K})$. *The nuclear norm* of matrix $\mathbf{A}$ is defined as sum of its singular values, thus

$$\|\mathbf{A}\|_* = \|\sigma(\mathbf{A})\|_1 = \sum_{i=1}^{k} \sigma_i \ ,$$

where $\sigma(\mathbf{A})$ is vector of singular values of matrix $\mathbf{A}$ and $k = \min(m, n)$.

Remark 2.30. The singular values of matrix $\mathbf{A}$ are defined in the following section. Precisely it is Definition 2.33.

2.2.2 Singular value decomposition (SVD)

Two out of the three methods in Part III utilise SVD. A simple exhibition of this decomposition and the relevant theorems are outlined below. Prior to this, it is necessary to introduce the eigenvalues of the matrix.

Definition 2.31. (Eigenvalues of the matrix) Let $\mathbf{A} \in \mathcal{M}_n(\mathbb{K})$. The characteristic polynomial of $\mathbf{A}$ is the polynomial $P_A(\lambda)$ defined on $\mathbb{C}$ by

$$P_A(\lambda) = \det(\mathbf{A} - \lambda\mathbf{E}).$$

It is a polynomial of degree equal to n. It has thus n roots in $\mathbb{C}$, which we call *the eigenvalues* of $\mathbf{A}$. The algebraic multiplicity of an eigenvalue is its multiplicity as a root of $P_A(\lambda)$. An eigenvalue whose algebraic multiplicity is equal to one is said to be a simple eigenvalue, otherwise, it is called a multiple eigenvalue. We call a nonzero vector $\mathbf{x} \in \mathbb{K}^n$ such that $\mathbf{Ax} = \lambda\mathbf{x}$ *the eigenvector* of $\mathbf{A}$ associated with the eigenvalue λ.

Lemma 2.32. For any $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{C})$, the matrix $\mathbf{A}^*\mathbf{A}$ is Hermitian and has real, nonnegative eigenvalues.

Proof. Obviously $\mathbf{A}^*\mathbf{A}$ is a square Hermitian matrix of size n. If λ is an eigenvalue of a Hermitian matrix $\mathbf{B}$ and $\mathbf{u} \neq 0$ a corresponding eigenvector, we have

$$\lambda\|\mathbf{u}\|_2^2 = \langle\lambda\mathbf{u}, \mathbf{u}\rangle = \langle\mathbf{Bu}, \mathbf{u}\rangle = \langle\mathbf{u}, \mathbf{B}^*\mathbf{u}\rangle = \langle\mathbf{u}, \mathbf{Bu}\rangle = \langle\mathbf{u}, \lambda\mathbf{u}\rangle = \bar{\lambda}\|\mathbf{u}\|_2^2$$

which shows that $\lambda = \bar{\lambda}$, i.e. $\lambda \in \mathbb{R}$. It remains to show that they are nonnegative. Let λ be an eigenvalue of $\mathbf{A}^*\mathbf{A}$, and let $\mathbf{x} = \mathbf{0}$ be a corresponding eigenvector such that $\mathbf{A}^*\mathbf{Ax} = \lambda\mathbf{x}$. Taking the Hermitian product of this equality with $\mathbf{x}$, we obtain

$$\lambda = \frac{\langle\mathbf{A}^*\mathbf{Ax}, \mathbf{x}\rangle}{\langle\mathbf{x}, \mathbf{x}\rangle} = \frac{\langle\mathbf{Ax}, \mathbf{Ax}\rangle}{\langle\mathbf{x}, \mathbf{x}\rangle} = \frac{\|\mathbf{Ax}\|^2}{\|\mathbf{x}\|^2} \in \mathbb{R}^+$$

and the result is proved. □

Definition 2.33. *The singular values* σ of matrix $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{K})$ are nonnegative square roots of n eigenvalues of $\mathbf{A}^*\mathbf{A}$.

Remark 2.34. This definition makes sense thanks to Lemma 2.32, which proves that the eigenvalues of $\mathbf{A}^*\mathbf{A}$ are real nonnegative, so their square roots are also real.

Definition 2.35. (SVD factorization) Assume that a matrix $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{C})$ have r singular values. Further assume that $\mathbf{U} \in \mathcal{M}_m(\mathbb{C})$ and $\mathbf{V} \in \mathcal{M}_n(\mathbb{C})$ are unitary matrices and $\mathbf{\Sigma} \in \mathcal{M}_{m,n}(\mathbb{R})$ is a diagonal matrix. Then *a singular value decomposition* of matrix $\mathbf{A}$ is a factorization

$$\mathbf{A} = \mathbf{U\Sigma V}^*, \tag{2.2}$$

where $\mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p)$, $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_p \geq 0$ and $p = \min(m, n)$.

Theorem 2.36. Every matrix $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{C})$ has a singular value decomposition (2.2). Furthemore, the singular values σ_i are uniquely determined, and, if $\mathbf{A}$ is square and the σ_i are distinct, the left and right singular vectors ($\mathbf{u}_i$ respectively $\mathbf{v}_i$) are uniquely determined up to complex signs.

Proof. **Existence** Set $\sigma_1 = \|\mathbf{A}\|_2$. By compactness argument, there must be vectors $\mathbf{v}_1 \in \mathbb{C}^n$ and $\mathbf{u}_1 \in \mathbb{C}^m$ with $\|\mathbf{v}_1\|_2 = \|\mathbf{u}_1\|2 = 1$ and $\mathbf{A}\mathbf{v}_1 = \sigma_1\mathbf{u}_1$. Consider any extensions of $\mathbf{v}_1$ and $\mathbf{u}_1$ to an ortonormal basis $\mathbf{v}_i$ of $\mathbb{C}^n$ respectively to an ortonormal basis $\mathbf{u}_i$ of $\mathbb{C}^m$. Denote $\mathbf{U}_1$ the unitary matrix with columns $\mathbf{u}_1$, respectively $\mathbf{V}_1$ with columns $\mathbf{v}_1$. Then the following holds

$$\mathbf{U}_1^*\mathbf{A}\mathbf{V} = \mathbf{S} = \begin{bmatrix} \sigma_1 & \mathbf{w}^* \\ \mathbf{0} & \mathbf{B} \end{bmatrix},$$

where $\mathbf{0}$ is column vector of dimension $m-1$, $\mathbf{w}^*$ is a row vector of dimension $n-1$ and $\mathbf{B}$ has dimensions $(m-1)\times(n-1)$. Furthermore

$$\left\| \begin{bmatrix} \sigma_1 & \mathbf{w}^* \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \mathbf{w}^* \end{bmatrix} \right\|_2 \geq \sigma_1^2 + \mathbf{w}^*\mathbf{w} = (\sigma_1^2 + \mathbf{w}^*\mathbf{w})^{\frac{1}{2}} \left\| \begin{bmatrix} \sigma_1 \\ \mathbf{w}^* \end{bmatrix} \right\|_2,$$

implying $\|\mathbf{S}\|_2 \geq (\sigma_1^2 + \mathbf{w}^*\mathbf{w})^{\frac{1}{2}}$. Since $\mathbf{U}_1$ and $\mathbf{V}_1$ are unitary, it is known that $\|\mathbf{S}\|_2 = \|\mathbf{A}\|_2 = \sigma_1$, ergo $\mathbf{w} = \mathbf{0}$.

If $m = 1$ or $n = 1$, the proof is complete. Otherwise, by induction hypothesis, the matrix $\mathbf{B}$ has a SVD, ergo $\mathbf{B} = \mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2^*$. It is easily veirifed that

$$\mathbf{A} = \mathbf{U}_1 \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2 \end{bmatrix}^* \mathbf{V}_1^*$$

is SVD of $\mathbf{A}$, which completes the proof of existence.

Uniqueness Note that σ_1 is uniquely determined by the condition that it is equal to $\|\mathbf{A}\|_2$. Now suppose that in addition to $\mathbf{v}_1$, there is another linearly independent vector $\mathbf{q}$ with $\|\mathbf{q}\|_2 = 1$ and $\|\mathbf{A}\mathbf{q}\|_2 = \sigma_1$. Define a unit vector $\mathbf{v}_2$, orthogonal to $\mathbf{v}_1$, as a linear combination of $\mathbf{v}_1$ and $\mathbf{q}$, ergo

$$\mathbf{v}_2 = \frac{\mathbf{q} - (\mathbf{v}_1^*\mathbf{q})\mathbf{v}_1}{\|\mathbf{q} - (\mathbf{v}_1^*\mathbf{q})\mathbf{v}_1\|_2}.$$

This vector $\mathbf{v}_2$ is second right singular vector of $\mathbf{A}$ corresponding to the singular value σ_1. Conclude that, if the singular vector $\mathbf{v}_1$ is not unique, then the corresponding singular value σ_1 is not simple. Once $\sigma_1, \mathbf{v}_1$ and $\mathbf{u}_1$ are determined, the remainder of SVD is determined by the action of $\mathbf{A}$ on the space orthogonal to $\mathbf{v}_1$. Since $\mathbf{v}_1$ is unique up to a sign, this orthogonal space is uniquely defined, and the uniqueness of the remaining singular values and vectors now follow by induction. □

Theorem 2.37. The rank of $\mathbf{A}$ is equal to p, the number of nonzero singular values.

Proof. The rank of the diagonal matrix is equal to the number of its nonzero entries. Further the $\mathbf{U}$ and $\mathbf{V}$ are full rank in decomposition (2.2). Therefore

$$\text{rank } \mathbf{A} = \text{rank } \mathbf{\Sigma} = p.$$

□

Proposition 2.38. Let $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ be the SVD factorization of some matrix $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{C})$ having p nonzero singular values arranged in decreasing order. For $1 \leq k \leq p$, the matrix $\mathbf{A}_k = \sum_{i=1}^k \mathbf{u}_i\sigma_i\mathbf{v}_i^*$ is the best approximation of $\mathbf{A}$ by matrices of rank k, in the following sense: for all matrices $\mathbf{X} \in \mathcal{M}_{m,n}(\mathbb{C})$ of rank k, we have

$$\|\mathbf{A} - \mathbf{A}_k\|_2 \leq \|\mathbf{A} - \mathbf{X}\|_2.$$

Moreover, the error made in substituting $\mathbf{A}$ with $\mathbf{A}_k$ is $\|\mathbf{A} - \mathbf{A}_k\|_2 = \sigma_{k+1}$.

Proof. **Error** Compute

$$\mathbf{A} - \mathbf{A}_k = \sum_{i=k+1}^{p} \mathbf{u}_i \sigma_i \mathbf{v}_i^* = \begin{bmatrix} \mathbf{u}_{k+1} & \dots & \mathbf{u}_p \end{bmatrix} \operatorname{diag}(\sigma_{k+1}, \dots, \sigma_p) \begin{bmatrix} \mathbf{v}_{k+1}^* \\ \dots \\ \mathbf{v}_p^* \end{bmatrix}.$$

Denote the matrix $\operatorname{diag}(0, \dots, 0,\ \sigma_{k+1}, \dots, \sigma_p,\ 0, \dots, 0) \in \mathcal{M}_{m,n}(\mathbb{R})$ by $\mathbf{D}$. Then $\mathbf{A} - \mathbf{A}_k = \mathbf{U}\mathbf{D}\mathbf{V}^*$ and since ℓ_2-norm is invariant under unitary transformation it holds that

$$\|\mathbf{A} - \mathbf{A}_k\|_2 = \|\mathbf{D}\|_2 = \sigma_{k+1}.$$

Approximation property For all $\mathbf{x} \in \mathbb{C}^n$, it holds that

$$\|\mathbf{A}\mathbf{x}\|_2 = \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^*\mathbf{x}\|_2 = \|\mathbf{\Sigma}\mathbf{V}^*\mathbf{x}\|_2. \tag{2.3}$$

Let E be the subspace of $\mathbb{C}^n$, of dimension $k+1$, generated by the vectors $\mathbf{v}_1,\ \dots,\ \mathbf{v}_{k+1}$. If $\mathbf{x} \in E$, it holds that $\mathbf{x} = \sum_{i=1}^{k+1} x_i \mathbf{v}_i$ and

$$\mathbf{V}^*\mathbf{x} = \mathbf{V}^* \sum_{i=1}^{k+1} x_i \mathbf{v}_i = \sum_{i=1}^{k+1} x_i \mathbf{V}^* \mathbf{v}_i = \sum_{i=1}^{k+1} x_i \mathbf{e}_i\ ,$$

where $\mathbf{e}_i$ is the i-th vector of the canonical basis of $\mathbb{C}^n$. Thus it holds that

$$\mathbf{\Sigma}\mathbf{V}^*\mathbf{x} = (\sigma_1 x_1,\ \dots,\ \sigma_{k+1} x_{k+1},\ 0,\ \dots,\ 0)^t. \tag{2.4}$$

So by substituting (2.4) to (2.3),

$$\|\mathbf{A}\mathbf{x}\|_2 \geq \sigma_{k+1} \|\mathbf{x}\|_2, \quad \forall \mathbf{x} \in E. \tag{2.5}$$

If the matrix $\mathbf{X} \in \mathcal{M}_{m,n}(\mathbb{C})$ is of rank $k < r$, its kernel ($\operatorname{Kerr}(\mathbf{X})$ is set of vectors $\mathbf{y}$ that satisfy $\mathbf{X}\mathbf{y} = \mathbf{0}$) is of dimension $n - k \geq r - k \geq 1$, and $\forall \mathbf{x} \in \operatorname{Ker}(\mathbf{X})$, it holds that

$$\|\mathbf{A}\mathbf{x}\|_2 = \|(\mathbf{A} - \mathbf{X})\mathbf{x}\|_2 \leq \|\mathbf{A} - \mathbf{X}\|_2 \|\mathbf{x}\|_2.$$

Assume that $\mathbf{X}$ contradicts the inequality from the proposition:

$$\|\mathbf{A} - \mathbf{X}\|_2 < \|\mathbf{A} - \mathbf{A}_k\|_2.$$

Hence $\forall \mathbf{x} \in \operatorname{Ker}(\mathbf{X})$, it holds that

$$\|\mathbf{A}\mathbf{x}\|_2 < \|\mathbf{A} - \mathbf{A}_k\|_2 \|\mathbf{x}\|_2 = \sigma_{k+1} \|\mathbf{x}\|_2,$$

and therefore if $\mathbf{x} \in E \cap \operatorname{Ker}(\mathbf{X})$ with $\mathbf{x} \neq \mathbf{0}$, then it is a contradiction to (2.5). Indeed, the two spaces have a nonempty intersection since $\dim E + \dim \operatorname{Ker}(\mathbf{X}) > n$, so that the inequality from the proposition is satisfied. □

Chapter 3

Theory of image registration

The first part of this chapter outlines some necessary math definitions for imaging the solar corona. The second part explains the process of image calibration, which is important part of prepocessing of the images. The last part describes image enhancement methods, which are used in the Chapter 7.

In the following section, there is cited from [2],[3] and [19], unless otherwise specified.

3.1 Basics of image registration

Definitions from image registration, which are important for purpose of this thesis, are outlined below. For other definition and more informations from this topic see [20].

Definition 3.1. (Digital grey-scale image) Let $R = \{0, 1, \cdots, M-1\} \times \{0, 1, \cdots, N-1\}, M, N \in \mathbb{N}$ and let $W = \{0, 1, \cdots, w-1\}, w \in \mathbb{N}$. Function

$$f(x, y) : R \to W$$

is called a *digital grey-scale image* or *image* only if no confusion may be caused. M is called the *image width*, N the *image height*. Elements of R are called *pixels*, value of f in pixel (x, y) is called the *pixel value*. The value of w determines the image *dynamic range*. It is said that the *dynamic range is n bits per pixel* or it is an *n-bit image* if $w = 2^n$.

Remark 3.2. Only grey scale image are taken into account, since the colour components is not useful in this study. Usually, the image is represented by *image matrix*. Quasi, this matrix is table filled with pixel values. Operations with matrices, such as multiplication do not have any sense for image matrices. That is the reason, why the image is defined here as a function with discrete domain.

Nowdays, it is impossible to work with optical system in sterile settings faultlessly. Because of it, real images always contain at least one of the following: *additive noise, impulse noise*, *dust*, defects caused by our optical system, diffuse light etc.

Basics of statistic are needed for understanding to next definitions. Those can be find in [13] and [14].

Definition 3.3. (Additive noise) Let f be a digital grey-scale image representing an ideal image (containing no additive noise). Let n be a digital grey-scale image of the same size as f whose pixel values are rounded independent realization of a random variable X. Let $h = f + n$. Then it is said that image h contains additive noise. Image n is called *noise image* and the characteristics of X are called the *characteristics of additive noise*.

Remark 3.4. The previous definition does not say what happens to pixels where $f+n > w-1$ or $f+n < 0$, where w is dynamic range of the image (see definition 3.1). We will define it now followingly

$$h(x,y) = \begin{cases} 0 & \text{if } f(x,y)+n(x,y) < 0, \\ f(x,y)+n(x,y) & \text{if } 0 < f(x,y)+n(x,y) < w, \\ w-1 & \text{if } f(x,y)+n(x,y) \geq w. \end{cases}$$

Definition 3.5. (Impulse noise) Let f be a digital gray-scale image representing an ideal image. Let h be a digital gray-scale image with pixel values

$$h(x,y) = \begin{cases} f(x,y) & \text{if } Y(x,y) = 0, \\ m(x,y) & \text{if } Y(x,y) = 1, \end{cases}$$

where $Y(x,y)$ are realizations of a random variable X with Bernoulli distribution and $m(x,y)$ are independent realizations of a random variable Z (generally, its distribution is unknown). If $Y(x,y)$ are not independent realizations of X, it is said that image h contains *defects*. If $Y(x,y)$ are independent realizations of X, it is said that image h contains *impulse noise*. The impulse noise is called *constant* if for the same conditions (images taken with the same camera and same exposure setting shortly one after another), the same realizations of X are obtained. The impulse noise is called *variable* if the realizations of X are different for every image.

Remark 3.6. Similarly to additive noise, the definition does not talk about the case when $f+n > w-1$ or $f+n < 0$, where w is dynamic range of the image (see Definition 3.1). All values higher than $w-1$ are mapped to $w-1$ and values lower than 0 are mapped to 0 like it is done in remark 3.4.

Variable impulse noise in digital images is caused mostly by collisions of the chip with high-energy particles (cosmical radiation). The result of it is visible as saturated ($h(x,y) = w-1$) pixels called the hot pixels. The longer exposure time is used, the more hot pixels the image contains.

Definition 3.7. (**Dust**) Let f be a digital gray-scale image representing an ideal image containing no dust. Let s be a function $R^2 \to \langle 0,1 \rangle$ which is equal to 1 on R^2 apart from a finite number of bounded connected sets $A_1, \cdots A_n, \in \mathbb{N}$. Let image

$$h(x,y) = \mathsf{Round}(f(x,y)s(x,y))$$

Then it is said that image h contains dust. (Operation Round is the unary operation of rounding to the nearest integer).

Remark 3.8. Dust particles can be detected in the same way as impulse noise (see section 4.1.1), but the following process is different. Correction of it is not mentioned in this thesis, since the chips on SOHO (see section 1.2.1) and on SDO (see section 1.3) are dustproof.

From here on, it is assumed that the images in this thesis contain only impulse noise. Commonly used method for filtration of impulse noise, as well as our own newly implemented method, are closely described in Part II.

3.1.1 Image filters

Image filters have many different applications such as reducing of noise in images, sharpening, gamma transformation etc. The filters can work in frequency or spatial domain. Generally, the filters are applying some operation on the neighborhood of the processed pixel.

Definition 3.9. (Size-preserving image transformation) A *size-preserving image transformation* ϕ is a mapping $\phi : f \to g$ that maps gray-scale image to gray-scale image with the same width and height.

Definition 3.10. (Pixel-value transformation and image filter) A size-preserving image transformation $\phi : f \to g$ is called *pixel value transformation* if ϕ can be expressed for each pixel $(x, y) \in R$ as

$$g(x, y) = \phi\left(f(x, y)\right) = \left(\phi \circ f\right)(x, y).$$

An image transformation which is not a pixel-value transformation is called an *image filter*.

Remark 3.11. The previous definition exactly means that for pixel-value transformation, only the pixel value $f(x, y)$ is needed to evaluate g in pixel (x, y). Beyond the image filter usually need more pixel values in a neighborhood of processed pixel for evaluating g in that pixel.

One typical example of pixel-value transformation is called *gamma transformation* and it preserves the maximal and the minimal pixel value while brigthening or darkening the image. The transformation is defined as

$$\phi_\gamma(t) = t^{\frac{1}{\gamma}}$$

where γ is positive number and it is assumed that input and output values are in $\langle 0, 1\rangle$. For $\gamma > 1$ brighter image is obtained and for $\gamma < 1$ darker image is obtained.

Thereby the reader knows the necessary definitions for imaging the solar corona. Following section talks about process of image calibration. Text is cited from [13b] and [3].

3.2 Image calibration

Theoretically, an image taken by a CCD camera[1] gives accurate informations about the light flux distribution on a sky. Practically, it is not working, because imperfections of the instruments and the discrete nature of light itself insert errors in the measured data. The errors are the result of several factors, some of them are random consequence of nature phenomena, others are deterministic.

First goal of the calibration process is minimalization of the contribution of deterministic factors to the errors, it means that the calibration is trying to remove the signature of the instrument from the data.

A second goal is preservation of information about the noise sources, so that the random errors of the data can be evaluated.

[1] charge coupled device camera, for more information see [14b]

3.2.1 Implementation of basic terms

In the following text, defininitions of important terms used in the calibration are presented, namely *flat-fields, bias* and *dark frames.*

CCD camera response model

Raw pixel values for a CCD frame can be calculated as follows:

$$f(x,y) = B(x,y) + tD(x,y) + tG(x,y)I(x,y) + R(x,y), \tag{3.1}$$

where $B(x,y)$ is the bias value of each pixel, t is the integration (exposure) time, $D(x,y)$ is the dark current, $G(x,y)$ is the sensitivity, $I(x,y)$ is the light flux reaching the pixel and $R(x,y)$ is a noise (random) component.

The bias value represents a pre-set value that is adjusted in the sensor prior to the opening of the shutter. It is essential to ensure that no underflow values are generated, even in the presence of noise that may result in negative data values. The value of this parameter can exhibit slight alterations at the pixel level. Additionally, its value may vary depending on the specific instrument utilized.

The dark current, too, is influenced by the instrument employed, but its behavior is further affected by temperature. Higher temperatures lead to increased dark current values. This phenomenon is a primary reason why it is common practice to cool cameras used for scientific purposes.

As previously stated, the function $G(x,y)$ characterises the sensitivity of each sensor on the chip. However, it also describes the vignetting effect [2] of the optical system and the impact of dust particles on the chip.

$R(x,y)$ is employed in a similar way to that of a generic constant in ODEs. Therefore, following the division by or addition of a constant, it will remain $R(x,y)$.

In order to estimate the flux values reaching the sensor, it is necessary to estimate the values of B, D and G. Once this has been done, the value of I can be obtained from equation (3.1). The estimation of B, D and G is achieved through the use of bias, dark and flat frames.

Bias frame

Assume, that very short exposures ($t = 0$) are taken without opening the camera's shutter, thus the light flux is absent $I(x,y) = 0$. These images are called bias frames. The equation (3.1) becomes

$$b(x,y) := f(x,y) = B(x,y) + R(x,y).$$

This bias frame b can be used to estimate the searched bias value B, since the noise component R is random. The estimation $\tilde{B}$ is then as follows

$$\widetilde{B}(x,y) := b(x,y).$$

This estimation is not very precise. Better estimation can be obtained if the average of more bias frames is computed. One can get arbitrarily close to the B if Chebyshev's

[2] Vignetting is a centrally symmetric darkening of the image with distance from the axis of the optical system caused by its optical properties.

inequality is used (read [21]). Assume, that N is number of images used for the computation. The estimation then looks like follows

$$\widetilde{B}(x,y) := \frac{1}{N}\sum_{i=0}^{N-1} b_i(x,y).$$

This value is called *the master bias frame.*

Dark frame

Now assume, that longer exposures ($t = t_0$) with the closed shutter ($I(x,y) = 0$) are taken. Like this, the dark frames are obtained. The equation (3.1) becomes

$$d(x,y) := f(x,y) = B(x,y) + t_0 D(x,y) + R(x,y)\ .$$

Similarly like in bias frame, noise R can be neglected for obtaining the estimation of dark current D. Further the estimation $\widetilde{B}$ is substracted from dark frame d and it is divided by the exposure time t_0. Estimation of dark current is

$$\widetilde{D}(x,y) := \frac{d(x,y) - \widetilde{B}(x,y)}{t_0}\ .$$

For reduction of the noise contribution, the average of several (M) dark frames is done:

$$\widetilde{D}(x,y) = \frac{1}{t_0}\frac{1}{M}\sum_{i=0}^{M-1} d_i(x,y) - \widetilde{B}(x,y)$$

It is better to work with a different form of the dark current frame, e.g. the one below

$$\widetilde{D}^*(x,y) := t_0\widetilde{D}(x,y) = \frac{1}{M}\sum_{i=0}^{M-1} d_i(x,y) - \widetilde{B}(x,y),$$

which is called *the bias subtracted master dark frame.*

Assume that the data frame (observed image) is taken with an integration time of t_1, then the first two terms in equation (3.1) are estimated by *the master dark frame* $\widetilde{D}_M$, thus

$$\widetilde{D}_M(x,y) = \widetilde{B}(x,y) + t_1\widetilde{D}(x,y) = \widetilde{B}(x,y) + \frac{t_1}{t_0}\widetilde{D}^*(x,y). \tag{3.2}$$

Assume that the data frame and the dark frames have same integration (exposure) time ($t_1 = t_0$), then the master dark frame can be computed as an average of M dark frames,

$$\widetilde{D}_M(x,y) = \frac{1}{M}\sum_{i=0}^{M-1} d_i(x,y). \tag{3.3}$$

Dark current $D(x,y)$ depends strongly on temperature. In order to ensure the accuracy of the dark frame images, it is necessary to take them shortly before or after the observed image.

Flat-field frame

The estimations of $\widetilde{B}$ and $\widetilde{D}$ are obtained, now the way how to estimate G in equation (3.1) has to be find. To do this, flat-field (even) illumination is applied to the camera (exposure time $t = t_f$) and flat-field frame is then obtained as

$$ff(x,y) := f(x,y) = B(x,y) + t_f D(x,y) + t_f G(x,y)L + R(x,y), \tag{3.4}$$

where L is the light flux reaching each pixel (assumed equal across the frame). Flat-field illumination images can be obtained by taking a photo of blue sky through a tracing paper.

Then a master dark frame for the flat fields $D_M^F(x,y)$ can be calculated (computed from (3.2) and (3.3)). It is subtracted from the flat-fields frame and the following equation is obtained

$$ff'(x,y) = ff(x,y) - D_M^F(x,y) = t_f G(x,y)L + R(x,y). \tag{3.5}$$

Equally as before, to reduce the noise contribution, average of K flat-field frames is computed. Thus *a master flat-field frame* is obtained

$$\widetilde{F}_M(x,y) = \frac{1}{K}\sum_{i=0}^{K-1} ff_i'(x,y). \tag{3.6}$$

If L is known, the equation (3.1) could be solved for $G(x,y)$. But the absolute value of L is not known and in many cases it varies between different sets of images. So, instead of finding the absolute value of $G(x,y)$, its variation across the frame is tried to be removed. Assume, that

$$G(x,y) = \bar{G}g(x,y),$$

thus $\bar{G}$ is the average of G over whole image, where the average of $g(x,y)$ across the frame is 1. The variability of G over an image is caused by many factors, e.g. different sensitivity of sensors, dust particles on the chip, optics vignetting etc. Then the equation (3.6), by using equation (3.5) becomes

$$\widetilde{F}_M'(x,y) = t_f \bar{G} g(x,y)L + R(x,y) \tag{3.7}$$

Assume that the average of $R(x,y)$ is computed and $\bar{R}$ is obtained. Then the average of $\widetilde{F}_M(x,y)$ across the frame is taken and following is obtained

$$\begin{aligned} \bar{F} &= \sum_x \sum_y t_f \bar{G} g(x,y)L + \bar{R} \\ &= t_f \bar{G} L \sum_x \sum_y g(x,y) + \bar{R} \\ &= t_f \bar{G} L + \bar{R} \end{aligned}$$

When equation (3.7) is divided by $\bar{F}$, the estimation of $g(x,y)$ is obtained,

$$\widetilde{g}(x,y) = \frac{\widetilde{F}_M(x,y)}{\bar{F}},$$

which is the final result that can be learnt from flat-fields.

3.2.2 Process of calibration

At this moment the way how to obtain bias, dark and flat-field frames is known. The following text shows the ideas how the proper calibration works.

It is cited from lecture notes for Numerical methods of image analysis taught by prof. RNDr. Miloslav Druckmüller, CSc.

Calibration using dark frame

Let us denote the observed image by $f(x, y)$ and the calibrated result by $f^*(x, y)$. Assume that dark frame $d(x, y)$ is obtained by process written on page 37. Then the $f^*(x, y)$ is obtained as

$$f^*(x, y) = f(x, y) - d(x, y)$$

This process has two presumptions. The observed image has to have same exposure time as dark frame and they had to be taken at almost identical temperatures. By experiments, it is proven that the acceptable change of the temperature is up to $0.1K$.

This calibration causes that the zero pixel value means zero light sensitivity. It is absolute necessity to do at least this type of calibration. However, it has one disadvantage. It increases the noise in the final image f^*. To minimalize this effect, master dark frame $\widetilde{D}_M$ (average of several dark frames) has to be used, ergo the calibrated image is computed as

$$f^*(x, y) = f(x, y) - \widetilde{D}_M(x, y),$$

where $\widetilde{D}_M(x, y)$ was computed using formula (3.3).

Calibration using dark frame and flat-field frame

Let us denote the observed image by $f(x, y)$, its exposure time by t_1. Further, let us denote dark frame, obtained by process written on page 37 with exposure time $t_0 = t_1$, by $d^1(x, y)$. Beyond let us have flat-field frame $ff(x, y)$ with exposure time t_f (see formula (3.4)). Let us denote dark frame with exposure time t_f by $d^2(x, y)$. The calibrated image f^* is obtained as

$$f^* = l \cdot \frac{f(x, y) - d^1(x, y)}{ff(x, y) - d^2(x, y)},$$

where l is supplemental variable, which is setting final dynamic range of f^*, since result of the fraction is in interval $\langle 0, 1 \rangle$.

This calibration removes also vignetting effects and problem of dust particles on the chip. But again, it is increasing the noise in the final image. Similarly as before, master dark frame (see formulas (3.2) and (3.3)) and master flat-field frame (see formula (3.6)) are used to minimize the noise deviation. Then the calibrated image f^* is obtained as

$$f^* = l \cdot \frac{f(x, y) - \widetilde{D}^1_M(x, y)}{\widetilde{F}_M(x, y) - \widetilde{D}^2_M(x, y)}.$$

If exposures are longer then $20s$ it is recommended to use median of several dark frames or flat-field frames instead of average of them.

3.3 Image enhancement

There is currently no unifying theory for enhancing images due to the lack of a general definition of image quality that can be used as a design criterion for an image enhancement processor. This section is cited from [20], unless otherwise specified.

Techniques for enhancing images aim to improve visual appearance or to convert them into a more suitable format for human or machine analysis. For the purposes of image analysis, the definition of image enhancement does not include information extraction. For instance, an image enhancement system could highlight an object's edge outline in an image through high-frequency filtering. Subsequently, this edge-enhanced image would function as input for a machine to trace the edges' outline. In this scenario, the image enhancement processor would highlight significant features of the initial image and simplify the processing task of a data-extraction machine. Techniques for enhancing images include contrast manipulation, modifying histograms, cleaning noise, and sharpening edges. In this paper, only modifications to the histogram will be closely described as they are utilized in Part III. Other techniques are described in [20].

3.3.1 Histogram modification

The images of a typical natural scene tend to have less perceptible detail in the darker regions. The histogram of such images is usually heavily skewed towards the darker levels, with a majority of pixels having a luminance lower than the average. One way to improve these types of images is to use a technique called histogram modification, in which the original image is rescaled so that the histogram of the improved image follows a desired shape.

The histogram modification process can be viewed as a monotonic point transformation $g_d = T\{f_c\}, c = \{1, 2, \cdots, C\}, d = \{1, 2, \cdots, D\}$, where the input amplitude variable $f_1 \leq f_c \leq f_C$ is mapped to an output variable $g_1 \leq g_d \leq g_D$ such that the output probability distribution $P_R\{f_c = b_d\}$ follows some desired form for a given input probability distribution $P_R\{g_c = a_c\}$, where a_c and b_d are the reconstruction values of the cth and dth level, respectively. It is clear that the input and output probability distributions must each sum to one. Thus,

$$\begin{aligned} \sum_{c=1}^{C} P_R\{f_c = a_c\} &= 1 \\ \sum_{d=1}^{D} P_R\{g_d = b_d\} &= 1 \end{aligned}$$

Furthermore, the cumulative distributions must equate for any input index c. That is, the probability that pixels in the input image have an amplitude less than or equal to a_c must be equal to the probability that pixels in the output image have amplitude less than or equal to b_d, where $b_d = T\{a_c\}$ because the transformation is monotonic. Hence

$$\sum_{n=1}^{d} P_R\{g_n = b_n\} = \sum_{m=1}^{c} P_R\{f_m = a_m\}. \tag{3.8}$$

The summation on the right is the cumulative probability distribution of the input image. For a given image, the cumulative distribution is replaced by the cumulative histogram to yield the relationship

$$\sum_{n=1}^{d} P_R\{g_n = b_n\} = \sum_{m=1}^{c} H_F(m), \tag{3.9}$$

where $H_F(m)$ represents the fractional number of pixels in an input image whose amplitude is quantized to the c-th reconstruction level. Equation (3.9) must be inverted to obtain a solution for g_d in terms of f_c.

The histogram transformation can be approximated by replacing the discrete probability distributions of equation (3.8) by continuous probability densities. The resulting approximation is

$$\int_{g_{\min}}^{g} p_g(g)\mathrm{d}g = \int_{f_{\min}}^{f} p_f(f)\mathrm{d}f,$$

where $p_f(f)$ and $p_g(g)$ are the probability densities of f and g, respectively. The integral on the right is the cumulative distribution function $P_f(f)$ of the input variable f. Hence

$$\int_{g_{\min}}^{g} p_g(g)\mathrm{d}g = P_f(f). \tag{3.10}$$

The table of several output image histograms and their corresponding transfer functions can be found in [20].

A very common method of histogram modification is *histogram equalization*, which forces the histogram of the image to be uniform.

Nonadaptive histogram equalization

Recall that the histogram of an image shows the sorting of pixels into individual quantized bins according to their value. Histogram equalisation attempts to create an output image by point rescaling so that its histogram has approximately the same number of pixels in all quantized bins.

The scaling algorithm is developed as follows. The average of the histogram is calculated. Then, starting from the lowest grey level of the original, the pixels in the quantization bins are combined until the sum is closest to the average. All these pixels are then rescaled to the new first reconstruction level at the midpoint of the first quantization bin of the enhanced image. The process is repeated for higher value grey levels. If the number of reconstruction levels of the original image is large, it is possible to rescale the grey levels so that the histogram of the enhanced image is almost constant. It should be noted that the number of reconstruction levels of the enhanced image must be less than the number of levels of the original image in order to achieve proper grey level redistribution.

Mathematicaly speaking, the probability density of the output image is

$$p_g(g) = \frac{1}{g_{\max} - g_{\min}} .$$

Thus from equation (3.10) the histogram equalization transfer function becomes

$$g = (g_{\max} - g_{\min})P_f(f) + g_{\min}.$$

Images corrected by histogram equalization often look like painted images, as subtle gradients between shades are lost. This method is unsuitable for use in the night sky as it erases most of the celestial formations.

Adaptive histogram equalization

The histogram equalization discussed earlier involves applying the same transformation or mapping function to each pixel in an image. The mapping function is based on the histogram of the whole image. This process can be made spatially adaptive by applying a histogram modification to each pixel based on the histogram of pixels within a moving window neighbourhood. This technique is obviously computationally intensive, as it requires histogram generation, mapping function calculation, and mapping function application to each pixel.

Adaptive histogram equalisation is a technique in which histograms are generated only at a rectangular grid of points, and the mappings at each pixel are generated by interpolating the mappings of the four nearest grid points. A histogram is computed at each grid point in a window around the grid point. The window dimension can be smaller or larger than the grid spacing. Let $M_{00}, M_{01}, M_{10}, M_{11}$ denote the histogram modification mappings generated at four neighbouring grid points. The mapping to be applied at pixel $f(j,k)$ is determined by a bilinear interpolation of the mappings of the four nearest grid points as given by

$$M = a[bM_{00} + (1-b)M_{10}] + (1-a)[bM_{01} + (1-b)M_{11}],$$

where

$$\begin{aligned} a &= \frac{k-k_0}{k_1-k_0}, \\ b &= \frac{j-j_0}{j_1-j_0}. \end{aligned}$$

Adaptive histogram equalization is unsuitable for images with high global contrast, such as a river/road running across the image, half in shadow and half in sunlight. This object often disappears from the final image or is severely deformed.

Typically, several image enhancement methods are used to achieve the best results. One such method is *Noise adaptive fuzzy equalization method*, which uses contrast and histogram modifications.

3.3.2 Noise adaptive fuzzy equalization method (NAFE)

NAFE generates an image filter, as defined in Definition 3.10. Unlike the gamma transform detailed on page 35, the filter does not maintain a constant transformation of pixel values across the entire image. Instead it differs for each pixel, relying on its specific neighbourhood. The method derivation presented below is cited from [22] and [23].

Let f_A denote the original input image. In this image the pixel values have a linear dependence on the intensity of the coronal emission. Let f_B denote the resulting image. This image is a linear combination of two images: $\phi_\gamma(f_A)$ and $E_{N,\sigma}(f_A)$, such that

$$f_B = (1-w)\phi_\gamma(f_A) + wE_{N,\sigma}(f_A). \tag{3.11}$$

$\phi_\gamma(f_A)$ denotes the gamma transform of image f_A, $E_{N,\sigma}(f_A)$ is an image created by the NAFE method, which is described in following text. The constant w is called the NAFE weight. Typical values for w lie in the interval $\langle 0.05, 0.3\rangle$. Constant $w = 0$ gives an image f_B without any enhancement.

Let us denote by $B = [b_{kl}]$ the matrix of size $n \times n$, whose elements are in the interval $\langle 0, 1\rangle$ and

$$k, l \in \left\{ \frac{1-n}{2}, \ldots, 0, \ldots, \frac{n-1}{2} \right\}.$$

The fuzzy neigborhood $\tilde{B}^n_{x,y}$ is a fuzzy set [3] with support $N^n_{x,y}$ and membership function $\mu^n_{x,y} : N^n_{x,y} \to \langle 0, 1\rangle$ where the membership grade of pixel $f_A(x + k, y + l)$ to the fuzzy neighborhood is defined as

$$\mu^n_{x,y}(x + k, y + l) = b_{k,l}.$$

Now let us define the fuzzy histogram of $\tilde{B}^n_{x,y}$ as

$$h^n_{x,y}(t) = \sum_{k=\frac{1-n}{2}}^{\frac{n-1}{2}} \sum_{l=\frac{1-n}{2}}^{\frac{n-1}{2}} b_{k,l}\delta_{t,f(x+k,y+l)} \tag{3.12}$$

where δ denotes the Kronecker delta. Then we define the cumulative fuzzy histogram

$$H^n_{x,y}(t) = \sum_{s=f^0_A}^{t} h^n_{x,y}(s)$$

and the normalized cumulative fuzzy histogram

$$C^n_{x,y}(t) = \frac{H^n_{x,y}(t)}{H^n_{x,y}(f^1_A)} \ ,$$

where f^0_A, f^1_A are minimum and maximum pixel values in $N^n_{x,y}$. Finally, we define the fuzzy equalizing function

$$g^n_{x,y}(t) = f^0_B + (f^1_B - f^0_B)C^n_{x,y}(t) \ ,$$

where f^0_B, f^1_B are minimum and maximum output pixel values. This function is different for every pixel $f_A(x, y)$ and the output pixel $f_B(x, y)$ is computed according to the formula

$$f_B(x, y) = g^n_{x,y}(f_A(x, y)).$$

The use of the fuzzy equalizing function solves the majority of the problems of the classical histogram equalization. However, one serious problem persists, namely the extreme amplification of additive noise in areas with very low contrast. This issue can result in compromised image quality. In cases where faint and low contrast details are lost, the entire dynamic range of the output image is utilised for noise display if the neighbourhood $N^n_{x,y}$ contains solely noise. The solution is to add more artificial additive noise to the image that is used for histogram computing. This will decrease the contrast of the original additive noise in the resulting image. However, this approach is impractical as

[3] The theory about fuzzy sets can be found in [24]

it requires additional work, since two copies of the input image must be used, with the noise generator adding additive noise to one of them.

Since the original noise in the image and the artificial noise are stochastically independent, the appropriate approach is to calculate the convolution of $C^n_{x,y}(t)$ with the PDF of noise that we wish to include in the image. Let us suppose that the added noise has a Gaussian distribution $N(\mu = 0, \sigma^2)$ (see definition 2.1) and PDF $G_\sigma(t)$. Then we define the noise adaptive fuzzy equalizing function as

$$g^n_{x,y}(t) = f^0_B + (f^1_B - f^0_B)C^n_{x,y}(t) * G_\sigma(t).$$

This function is used for the creation of image $E_{N,\sigma}(f_A)$ in formula (3.11). The convolution of the normalized cumulative fuzzy histogram with a Gaussian kernel has a significant influence only in the pixel neighborhood dominated by noise in which the image has very low contrast. On the other hand, the influence is negligible in the contrasty parts of the image.

There exist one more problem, namely loss of contrast on boundaries between areas with significantly different brightness. It is caused by computing the fuzzy histogram (3.12) from all pixels in the processed pixel neighborhood regardless of its values. If the position of the processed pixel is near the border of areas with significantly different pixel values, the neighborhood contains pixels from both areas. The solution is to use for fuzzy histogram computing only pixels that belong to area which the processed pixel belongs to. Therefore, we replace (3.12) with formula

$$h^{n,\varepsilon}_{x,y}(t) = \sum_{k=\frac{1-n}{2}}^{\frac{n-1}{2}} \sum_{l=\frac{1-n}{2}}^{\frac{n-1}{2}} b_{k,l}\delta_{t,f_A(x+k,y+l)}\Delta^{\varepsilon}_{t,f_A(x+k,y+l)} \ ,$$

where

$$\Delta^{\varepsilon}_{t,f_A(x+k,y+l)} = \begin{cases} 1 & \text{if } |t - f_A(x+k, y+l)| < \varepsilon \\ 0 & \text{else .} \end{cases}$$

The optimal value for ε needs to be determined via experimentation. Extremely small ε values result in image fragmentation into small areas with very high contrast features that fail to represent significant boundaries within the image. Conversely, high ε values produce images that are either identical or nearly identical to those obtained when equation (3.12) is applied. Let us denote the corresponding normalized cumulative fuzzy histogram by $C^{n,\varepsilon}_{i,j}(x)$ and the corresponding noise adaptive fuzzy equalizing function by $g^{n,\varepsilon}_{i,j}(x)$. The use of this function in formula (3.11) solves both problems meintoned before.

Parameter setting for NAFE can be found in [22] and [23]. Comparison of the original image and the NAFE proccesed image can be seen in Figure 3.1.

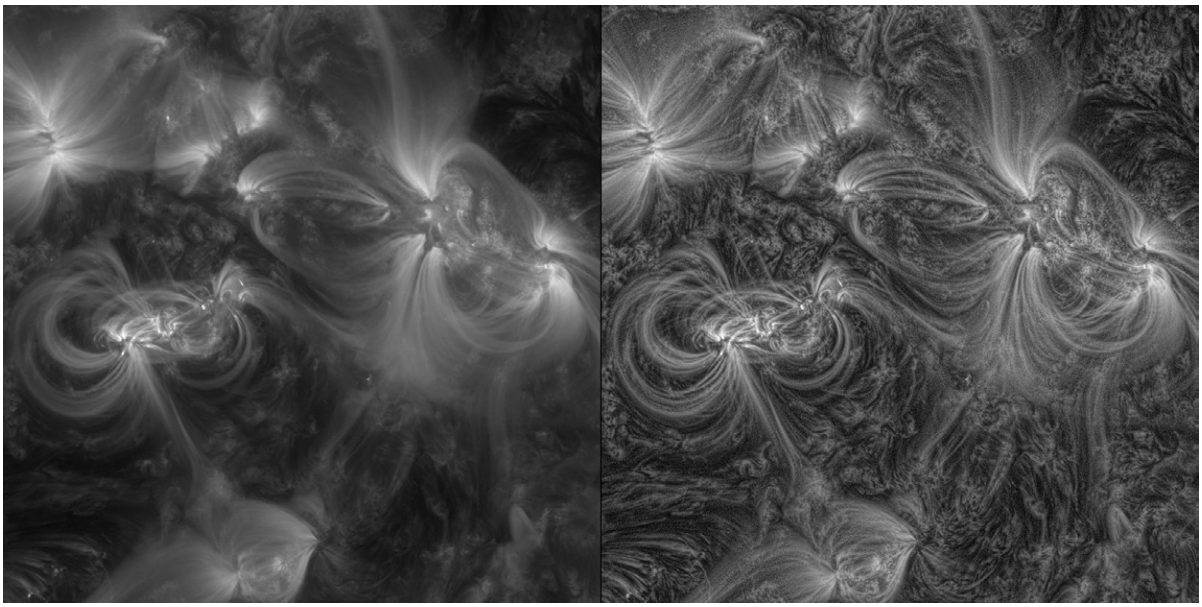


Figure 3.1: A comparison between an image that has undergone NAFE method processing (right) and the original, unprocessed image (left). This is a segment of a 171 Å image taken by the SDO-AIA, downloaded from [15b]

Part II

Filtration of impulse noise

Chapter 4

Basic procedure

The filtration of impulse noise (see Definition 3.5) is very important in image analysis. However, nowadays it is mainly performed by method that is highly dependent on the person doing the filtering. The goal of this paper is to find a new way how to do it.

The usual procedure is described here, the possible new way of filtration can be found in Chapter 5.

4.1 Theory

This section is written accordingly with the lecture notes from Numerical methods of image analysis taught by prof. RNDr. Miloslav Druckmüller, CSc. and some parts are cited from [2]

Filtration of impulse noise consists of two steps:

- **Detection**... determination of the coordinates of defective pixels
- **Correction**... modification of defective pixels by knowledge of their neighborhood

We will start with detail description of the detection. Afterwards we will move to the second step, namely to the correction.

4.1.1 Detection

In the past, the real detection was often replaced by a hardware method. For more informations about it, look into [2]. Nowadays it is rarely used.

It is recommended to read Chapter 7 in [13] for better understanding of next part.

Detection process

It is based on testing the statistical hypothesis (see Definition 2.2) that the pixel is defective, thus noise. As mentioned on page 25, it is possible that the statistical error is made during rejecting/accepting the hypothesis. There exist two types of these errors:

- **type I error**: the hypothesis is true, but we reject it

- **type II error**: the hypothesis is false, but we do not reject it

Type I error is not a big problem for the final image, as it only causes that the correction is not ideal, ergo some noise pixels remain in the image. The type II error is much worse, since pixels that do not contain noise are also corrected, which means that the new defects are imported into the image.

The probability of making a type I error is called *the level of significance*. The probability of making a type II error is called *power of the test*. For their mathematical definitions see page 26. The goal is to make them as small as possible. It is not feasible to minimize them simultaneously, it is needed to do the compromise and minimize the one that is more important for the case. Thus, according to the paragraph above, the type II error has to be minimized.

For assessment of the hypothesis, the statistical sample is needed. Therefore, an explicit number of pixels in the neighbourhood of a testing pixel is used as the sample. The size of this neighbourhood is not precisely specified. If a small one is chosen, there is a possibility that the necessary information will not be obtained. On the other hand, if a large neighbourhood is chosen, there is no homogeneous information, which may affect the accuracy of the test. Frequently the meshes 5×5, 7×7 or 9×9 are used (they are odd to have the central pixel).

Testing criterion

This section shows how to determine whether a pixel is defective or not. The first criterion is as follows:

$$|x - \bar{x}| > \varepsilon,$$

x – the pixel value
$\bar{x}$ – any statistical measure of position of statistical sample
ε – constant affecting the level of significance and power of the test

The $\bar{x}$ is chosen as one of these three statistical measures of position: median, arithmetic mean, or their combination. The problem with the median is its long computational time.

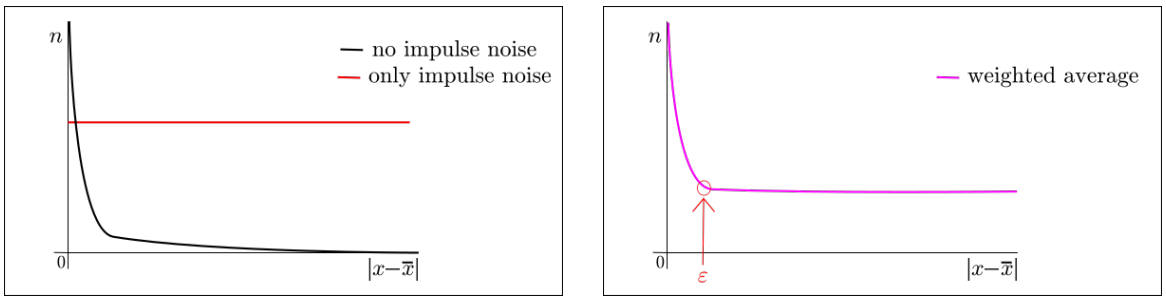
The choice of ε is a bit tricky. If a large ε is chosen, the probability of making a type I error is high, thus only a small number of defective pixels will be corrected. On the contrary, if a small ε is chosen, the probability of making a type II error is high, thus the image is spoiled.

The optimal ε is obtained by making an histogram of the dependence of n, percent of occurence, on $|x - \bar{x}|$ (see Figure 4.1). ε is not universal for the whole image.

In some cases we need other criterion then the one we described above. Second criterion which is commonly used is *rank criterion*.

Rank criterion

In general, this criterion works with a statistical sample of size n. This sample is sorted by the values, from the smallest to the largest one. The ordinal number of the element x



(a) (b)

Figure 4.1: Histogram of dependence of n (percent of occurance) on $|x - \bar{x}|$. Image (b) is weighted average from image (a).

is called *rank* and it is denoted by $r(x)$. To accept the hypothesis, the following inequality must hold:

$$\left| r(x) - \frac{n}{2} \right| > \varepsilon$$

When this criterion is used for hypothesis that the tested pixel is defective, the pixels in the neighbourhood of the tested pixel has to be sorted according to their pixel values. Thus the rank of the pixel with the lowest pixel value has $r(x) = 1$ and the pixel with the highest pixel value has $r(x) = n$. The choice of ε leads to the same problem as was described in the previous section. In this case, the ε is the same for the entire image.

This completes the process of detection and the second part (the process of correction) can be described.

4.1.2 Correction

It is fully dependent on detection. Only those defective pixels found during detection can be corrected.

Methods of correction

There are a variety of methods for correcting the defective pixels. The following three methods are the most commonly used.

One-pass method

As the name suggests, this method works without iterations, thus everything is done in one pass. More precisely, the defective pixel is corrected directly after detection. The correction consists of replacing the value x with the value $\bar{x}$. The value $\bar{x}$ is also calculated from the defective pixels, which means that the correction is inaccurate.

Double-pass method

Similar to before, the correction consists of replacing the value x with the value $\bar{x}$. The difference is that the detection is done in the first pass and the correction is done in the second pass. Since the correction is done after the whole process of detection, $\bar{x}$ is computed only from nondefective pixels. For this reason, the method is more accurate

than the one-pass method. Even though this method is quite slow, it is abundantly used in practise.

Interpolation method

The last method shown in this section is quite different from the previous ones. Firstly, the whole process of detection is done. Then, the defective pixel is corrected by linear regression (see Chapter 11 in [14]).To build the linear regression model, only the pixels in the neighborhood of the defective pixel that have been evaluated as nondefective are used. Mainly first or second degree polynomials are used as regression functions.

In a case, the filtration of additive noise (see Definition 3.3) has to be done, the filtration of impulse noise has to be done before. In other case the filtration of impulse noise can not be done anymore.

The script for this procedure was written in Embacardero Delphi XE6 (see Appendix A on page 121). It provides a starting point for our next approach. In the following section, there will be brief description of the selected steps.

4.2 Implementation

4.2.1 Routine

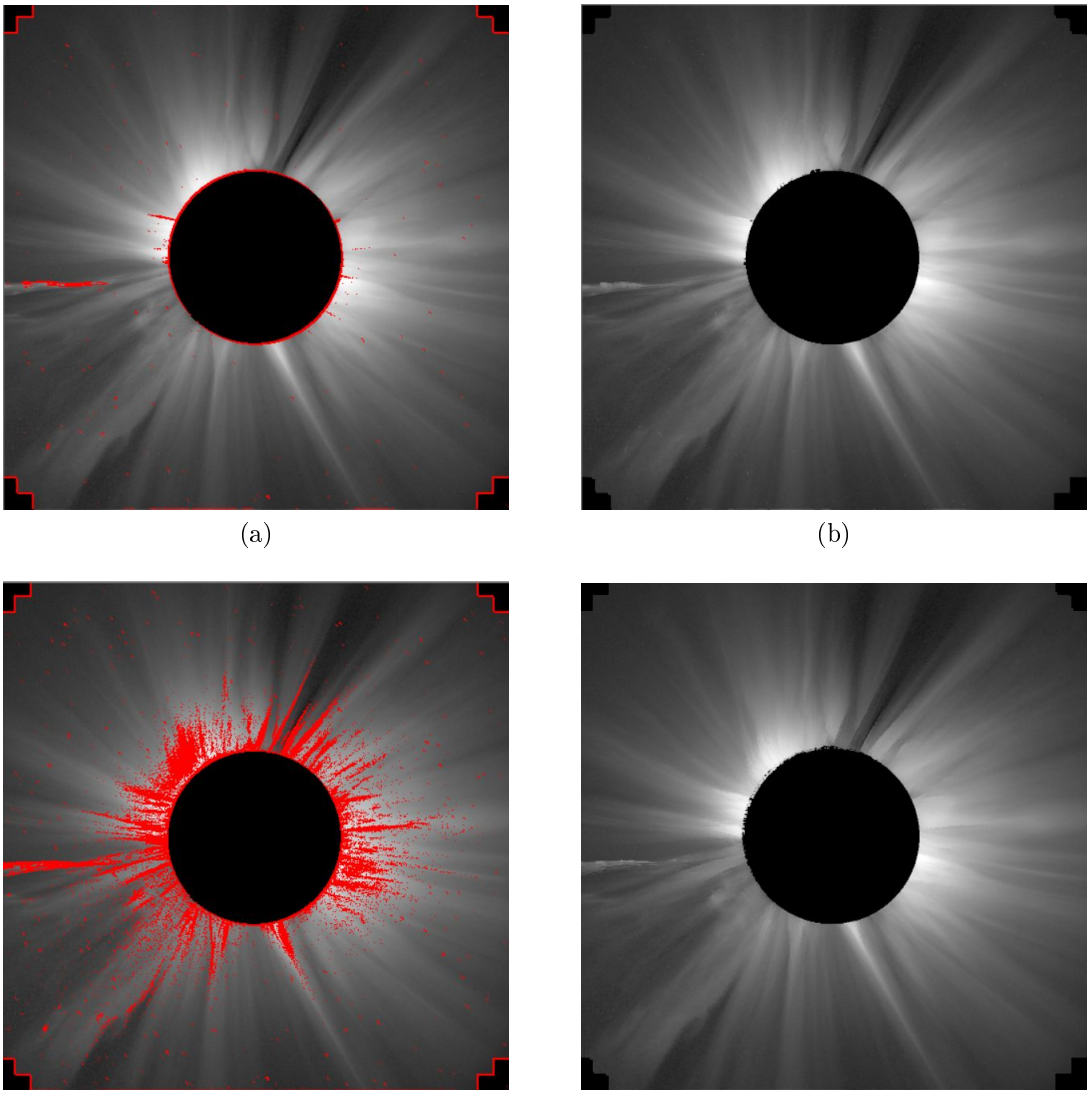
The neighbourhood for detection was chosen as 7×7 mesh. The classical criterion was used, i.e. the criterion with pixel values. The median was taken as the statistical measure of position because it is the most accurate and its time cost is not important for the approach of this thesis.

The iterative method was used for correction. It is a combination of the one-pass method and the double-pass method. In the first iteration, the defective pixels are corrected by the median $\bar{x}_1$, which was already calculated during the detection, ergo the median also contains the values of the defective pixels. In the second iteration, the new median $\bar{x}_2$ is computed in a slightly larger neighbourhood (11×11 mesh) from the nondefective pixels and the corrected pixels.

4.2.2 Results

The result of applying this filter is shown in Figure 4.2. Detected impulse noise is represented by red pixels. One can see that the choice of ε is really crucial. There is a difference not only on images from detection, but also on the filtered (corrected) images. Image (d) looks like it was painted with watercolors, which is an undesirable effect.

This procedure also determines normal solar rays, CMEs, and other things that normally occur in the solar atmosphere (see Chapter 1) as impulse noise because the shape of a structure is not included in the consideration. This should be improved by the moment method, which is described in the next chapter.



(a) (b)

(c) (d)

Figure 4.2: The comparison of images with different ε in the testing criterion. The test image was taken from SOHO LASCO C2 at 13:54:05 on March 20, 2000. Image (a) is the result of detection of impulse noise with $\varepsilon = 300$. Image (b) is the result of filtration of detected impulse noise from image (a). Image (c) is the result of filtration of impulse noise with $\varepsilon = 100$. On image (d) one can see that the statistical type II error was made, i.e. the image is worse than before filtration, ergo the hypothesis was accepted even for nondefective pixels.

Chapter 5

Moment method

5.1 Theory

This section is written according to the lecture notes from Numerical methods of image analysis taught by prof. RNDr. Miloslav Druckmüller, CSc. and some parts are cited from [25].

The moment method is mainly used for pattern recognition. There are two ways to compute it. In the first one, pixel values are not considered, in the second one they are used. Both methods give slightly different results, so the goal is to find out which method is more suitable for our approach.

5.1.1 Introduction to geometric moment theory

As mentioned earlier, the moment method is mainly used for pattern recognition. Pattern recognition is highly dependent on the coordinate system. For this reason, a special coordinate system has to be defined. It must be invariant to scale changes, translation, and rotation. Before a proper definition of this system, the basic theory is required.

Definition 5.1. (General geometric moment) *The general geometric moment* $M_{k,l}$ of object G relative to axes x and y is defined as

$$M_{k,l}(G) = \iint_G x^k \cdot y^l \, \mathrm{d}x\mathrm{d}y \ ,$$

where $k + l$ is called *the order of the method.*

Since the image stored in the computer has discrete character, the surface integral can be converted into a double sum. Then the formula looks like

$$M_{k,l}(G) = \sum_i \sum_j x_i^k \cdot y_j^l \cdot S \ , \tag{5.1}$$

where x_i, y_j are taken as the center of the pixels and S is the area of a pixel. The value of S can be chosen to be equal to 1. This means that the units are pixels, which is better for our approach.

The moment $M_{0,0}(G)$ is equal to the area of the object G. It holds that the coordinates of the centroid T of the area G are obtained as follows

$$T_G : \left[\frac{M_{1,0}(G)}{M_{0,0}(G)}, \frac{M_{0,1}(G)}{M_{0,0}(G)}\right] .$$

The general geometric moments change with the scale change, translation and rotation. It is working in the general system of coordinates (see Figure 5.1).

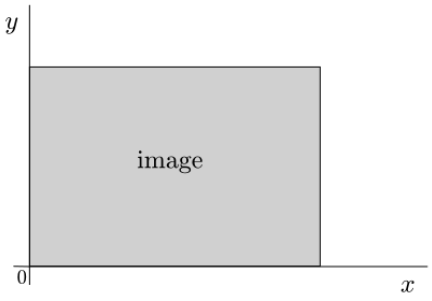


Figure 5.1: Illustration of the general system of coordinates.

Principal coordinate system

The problem with the dependency on scale change, translation and rotation is solved by means of the following conditions:

1. Invariance for scale change: the condition (5.2) defines the scale on both axes

$$M_{0,0}(G) = 1 \tag{5.2}$$

2. Invariance for translation: the conditions (5.3) and (5.4) set the origin of the coordinate system to the centroid T_G

$$M_{1,0}(G) = 0 \tag{5.3}$$
$$M_{0,1}(G) = 0 \tag{5.4}$$

   At this moment the coordinate system has a known origin and the scale is set on both axes. The unknowns of the system are the axis labels, the trend of the axes, and the orientation of the axes, ergo the rotation by angle α with respect to the general axes (see Figure 5.2).

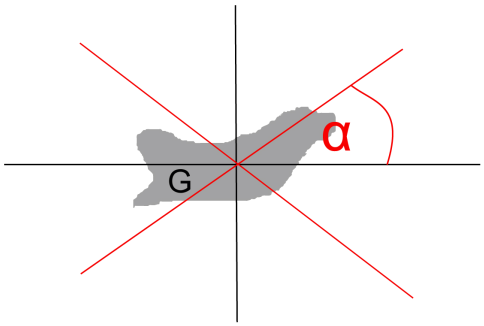


Figure 5.2: Illustration of the system of coordinates with origin at the center of object G. The red axes represent the system rotated by the angle α.

3. Invariance for rotation:

   - Orientation of the axes is given by condition (5.5)

$$M_{1,1}(G) = 0 \tag{5.5}$$

     This fact is illustrated in Figure 5.3. The moment $M_{1,1}(G)_\alpha$, which is a second order moment with dependence on the angle of rotation α, is plotted there. Since the result of $M_{1,1}(G)_\alpha$ can also be negative, the size $|M_{1,1}(G)_\alpha|$ is plotted in polar coordinates (as distance from the origin) and the sign is added to individual parts of the curve. It is quite obvious that the axes are placed at the position where $M_{1,1}(G)_\alpha$ changes sign, that is, at the places where it is equal to zero.

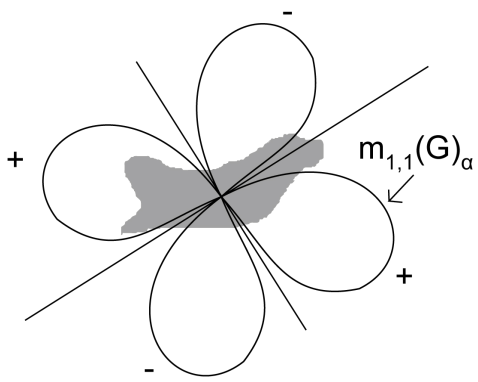


Figure 5.3: Plot of $M_{1,1}(G)_\alpha$ in polar coordinates. This is also a coordinate system satisfying conditions (5.2)-(5.5).

   - Axis labels is done by the help of unmixt second order moments $M_{2,0}(G)$ and $M_{0,2}(G)$. According to the condition (5.6), the larger moment corresponds to the axis x. If they are equal, the labeling does not change the results.

$$M_{2,0}(G) \;\geq\; M_{0,2}(G) \tag{5.6}$$

     Similarly as before, the moments $M_{2,0}(G)_\alpha$ and $M_{0,2}(G)_\alpha$ are calculated, thus moments that depend on the angle. Then they are plotted in polar coordinates (see Figure 5.4). In this case, no negative values occur.

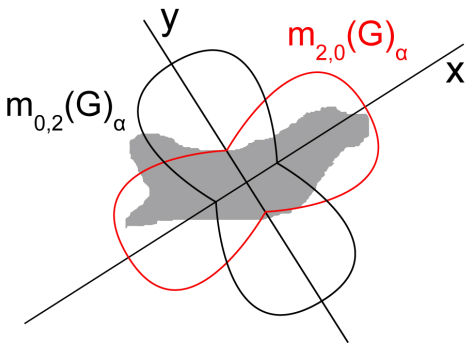


Figure 5.4: Plot of $M_{2,0}(G)_\alpha$ and $M_{2,0}(G)_\alpha$ in polar coordinates. Moreover, it is a coordinate system satisfying conditions (5.2)-(5.6).

- Trend of the axes can be obtained by calculating the third order moment $M_{3,0}(G)$. The condition (5.7) indicates the trend on the x axis. Together with the condition (5.2), the trend on both axes is set. If the inequality becomes equality, the trend can be chosen in either direction.

$$M_{3,0}(G) \geq 0 \tag{5.7}$$
$$M_{0,3}(G) \geq 0 \tag{5.8}$$

  Once again, the moment $M_{3,0}(G)_\alpha$ is calculated. The result may be negative, so the size $|M_{3,0}(G)_\alpha|$ is plotted as distance from the origin and signs are added to corresponding curves (see Figure 5.5).

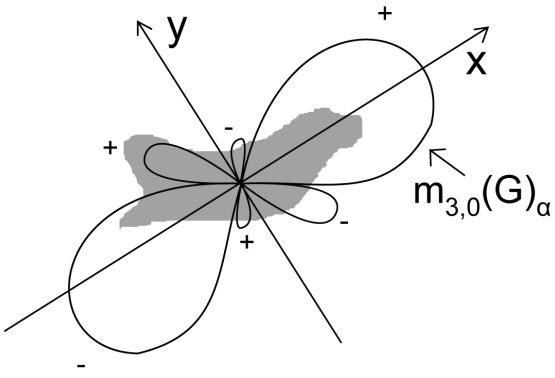


Figure 5.5: Plot of $M_{3,0}(G)_\alpha$ in polar coordinates. This is also a coordinate system satisfying conditions (5.2)-(5.7).

Condition (5.8) is additional. Together with (5.5)-(5.7), it gives us invariance for flipping. There might be a problem with the system defined by (5.2)-(5.8). This system is not applicable when it is used in text recognition, since b and d are not the same letters (as are p and q), but this system is taking them as the same objects. To obtain an unproblematic

system of coordinates that is invariant to scale change, rotation, and translation, the condition (5.8) has to be replaced by the following rule

$$\textit{System of coordinates is positively (clockwise) oriented.} \tag{5.9}$$

Definition 5.2. The coordinate system that satisfies conditions (5.2)-(5.7) and condition (5.9) is called *the principal system of coordinates* of the object G. The moments calculated in this system are called *the principal moments*.

Prior the computation of the principal moments, the central moments and the normalized central moments have to be defined.

Central geometric moments

The central geometric moments of an object G are computed relatively to a coordinate system whose origin is at the centroid of the object and whose axes are parallel to the axes of the general system (see Figure 5.1). Both axes have the same scale.

Definition 5.3. Assume that centroid T_G of object G has coordinates $[x_c,\ y_c]$, then the *central geometric moment* is defined by following formula

$$CM_{k,l}(G) = \iint_G (x - x_c)^k \cdot (y - y_c)^l \, \mathrm{d}x\mathrm{d}y \ , \text{ respectively} \tag{5.10}$$

$$CM_{k,l}(G) = \sum_i \sum_j (x_i - x_c)^k \cdot (y_j - y_c)^l \cdot S, \tag{5.11}$$

where $x_i,\ y_j$ are taken as the center of the pixels and S is the area of a pixel.

The central geometric moments are invariant to translation.

Normalized central geometric moments

The normalized central geometric moments are attached to the same system of coordinate as the central moments. The only difference is that the scale of this system is chosen to be equal to 1, which means that the moment of zero order is equal to 1.

Definition 5.4. Assume that $CM_{k,l}(G)$ was computed by (5.11) and $M_{0,0}(G)$ by (5.1), then *the normalized central geometric moment* is defined by following formula

$$NCM_{k,l}(G) = \frac{CM_{k,l}(G)}{M_{0,0}(G)^{\frac{k+l+2}{2}}} \ . \tag{5.12}$$

The normalized central geometric moments are invariant to translation and scale change.

Principal moments

First of all, the angle of rotation of the axes with the respect to the general system of coordinates has to be computed. Assuming that the $NCM_{k,l}(G)$ was obtained by the formula (5.12), the computation of α_j is done as follows

$$\alpha_j = \frac{1}{2} \arctan \frac{2 \cdot NCM_{1,1}(G)}{NCM_{2,0}(G) - NCM_{0,2}(G)} + \frac{(j-1) \cdot \pi}{2} \ , \quad j = 1, 2, 3, 4 \ . \tag{5.13}$$

If α_j is not uniquely determined by the conditions (5.2) - (5.7) and (5.9), then the α_j with the smallest j satisfying these conditions is chosen. This α_j is denoted as f.

Assume that $NCM_{k,l}(G)$ was obtained by the formula (5.12). Let us denote

$$\begin{aligned} S_f &= \sin f \ , \\ C_f &= \cos f \ , \\ NCM_{k,l} &= NCM_{k,l}(G) \ , \end{aligned}$$

then it holds that

$$PM_{2,0}(G) = C_f^2 \ NCM_{2,0} + 2S_f C_f \ NCM_{1,1} + S_f^2 \ NCM_{0,2} \tag{5.14}$$

$$PM_{0,2}(G) = S_f^2 \ NCM_{2,0} - 2S_f C_f \ NCM_{1,1} + C_f^2 \ NCM_{0,2} \tag{5.15}$$

$$\begin{aligned} PM_{3,0}(G) &= C_f^3 \ NCM_{3,0} + 3S_f C_f^2 \ NCM_{2,1} + 3S_f^2 C_f \ NCM_{1,2} \\ &\quad + S_f^3 \ NCM_{0,3} \\ PM_{2,1}(G) &= -S_f C_f^2 \ NCM_{3,0} + (C_f^3 - 2S_f^2 C_f) \ NCM_{2,1} \\ &\quad - (S_f^3 - 2S_f C_f^2) \ NCM_{1,2} + S_f^2 C_f \ NCM_{0,3} \\ PM_{1,2}(G) &= S_f^2 C_f \ NCM_{3,0} + (S_f^3 - 2S_f C_f^2) \ NCM_{2,1} \\ &\quad + (C_f^3 - 2S_f^2 C_f) \ NCM_{1,2} + S_f C_f^2 \ NCM_{0,3} \\ PM_{0,3}(G) &= -S_f^3 \ NCM_{3,0} + 3S_f^2 C_f \ NCM_{2,1} - 3S_f C_f^2 \ NCM_{1,2} \\ &\quad + C_f^3 \ NCM_{0,3} \end{aligned}$$

If one need also principal moments of fourth order, read [25].

The principal moments depend only on the shape of the object G. Their geometric interpretation is vivid nicely on moments of second order, which define *Legendre ellipse*.

Equimomental (Legendre) ellipse

Assume that the object is displayed in the principal system of coordinates. The problem of finding an ellipse E that has the same geometric moments as the object, is in general, intractable. It becomes solvable if the ellipse is taken as a homogeneous thin plate with homogeneous density. The equimomental ellipse is then defined as follows.

Definition 5.5. Assume that the object G has density equal to 1. Equimomental ellipse to the object G is such an ellipse which has density less than or equal to 1 and which approximates the object so that they have the same physical moments (see section 5.1.2) up to the second order, inclusive. The size of semi-axes is calculated as

$$A = 2\sqrt{PM_{2,0}(G)} \ , \tag{5.16}$$

$$B = 2\sqrt{PM_{0,2}(G)} \ , \tag{5.17}$$

where A is semi-major axis, B is semi-minor axis and moments $PM_{2,0}(G)$, $PM_{0,2}(G)$ are obtained from the formulas (5.14), (5.15).

Equimomental ellipse is depicted on Figure 5.6. Since the density of E is less than or equal to 1, the area of it is always greater than or equal to the area of the object G.

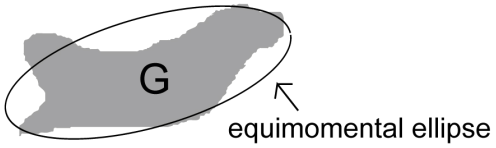


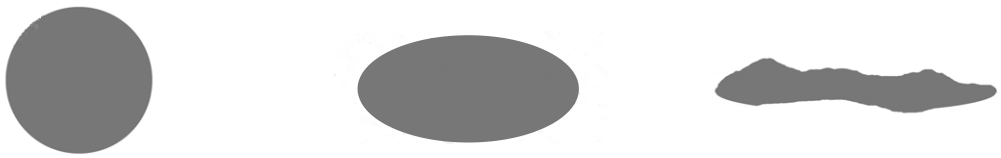
Figure 5.6: Equimomental ellipse of the object G.

The equimomental ellipse has specific features that can help in recognising of the object, namely elongation and dispersion.

Definition 5.6. Assume that the A, B are obtained from (5.16) and (5.17). Then *the elongation* is defined as

$$EL = \log_2 \left(\frac{A}{B} \right) \ .$$

Some results for EL are displayed in Figure 5.7.



(a) (b) (c)

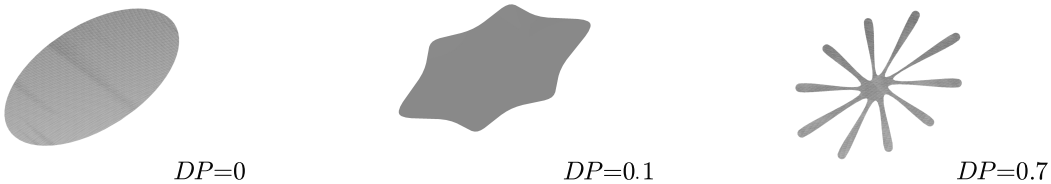
Figure 5.7: The comparison of objects with different elongations. On image a) a classical circle is shown. On image b) an ellipse with axes ratio $A : B = 1 : 2$ can be seen and on image c) a random object which can be approximated by equimomental ellipse is shown.

Elongation is a suitable characteristic when one has to describe the object just by one number.

Definition 5.7. Assume that the area of the equimomental ellipse is computed as $\pi \cdot A \cdot B$, where A, B are obtained from (5.16) and (5.17). Then *the dispersion* is defined as

$$DP = \log_2 (\pi \cdot A \cdot B) \ .$$

The more the object differs from the ellipse, the more the dispersion increases. Some results for DP are shown in Figure 5.8.



DP=0 DP=0.1 DP=0.7

(a) (b) (c)

Figure 5.8: The comparison of objects with different dispersion. On image a), there is ellipse with axis ratio $A : B = 1 : 2$. On image b), there is the same ellipse but with rounded boundary. On image c), there is random object constructed from the ellipse on image a).

5.1.2 Introduction to physical moment theory

It has already been mentioned that there are other moments, called physical moments. Like the geometric moments, the physical moments are also strongly dependent on the coordinate system. Therefore, a system invariant to scale change, translation, and rotation has to be introduced.

General physical moment

Definition 5.8. *General physical moment* of the object H with known density $f(x, y)$ is defined as

$$m_{k,l}(H) = \iint_H x^k \cdot y^l \cdot f(x, y)\ \mathrm{d}x\mathrm{d}y\ ,$$

where $k + l$ is called *order of the moment.*

As mentioned in the previous subsection, the image stored in the computer has discrete character, so the surface integral is converted into double sum. Then the formula is as follows

$$m_{k,l}(H) = \sum_i \sum_j x_i^k \cdot y_j^l \cdot f(x_i, y_j) \cdot S\ , \tag{5.18}$$

where $x_i,\ y_j$ are taken as the center of pixels, $f(x_i, y_j)$ are the pixel values of object H and S is the area of a pixel.

These moments depend on scale change, translation and rotation of the object H. They are suitable only as characteristics, which are taken relatively to the general coordinate system (see Figure 5.1).

In the same way as in the previous section, the *central physical moment* and *normalized central physical moment* will be defined.

Central physical moment

The central physical moments of the object H are related to a coordinate system which has its origin in the centroid T_H of the object and whose axes are parallel to the axes of the general system (see Figure 5.1). Both axes have the same scale.

Definition 5.9. Assuming that the centroid T_H of the object H has coordinates $[x_c,\ y_c]$ and is computed using (5.18) as follows

$$T_H = \left[\frac{m_{1,0}(H)}{m_{0,0}(H)}, \frac{m_{0,1}(H)}{m_{0,0}(H)}\right] \ .$$

Then *the central physical moment* is defined as

$$cm_{k,l}(G) = \iint_H (x - x_c)^k \cdot (y - y_c)^l \cdot f(x, y)\ \mathrm{d}x\mathrm{d}y \ ,$$

where $f(x, y)$ is density of the object H in position $[x, y]$.

Again, the surface integral can be converted to a double sum since the computer image has discrete character. The formula is as follows

$$cm_{k,l}(G) = \sum_i \sum_j (x_i - x_c)^k \cdot (y_j - y_c)^l \cdot f(x_i, y_j) \cdot S, \tag{5.19}$$

where $x_i,\ y_j$ are taken as the center of pixels, $f(x_i, y_j)$ are the pixel values of the object H and S is the area of a pixel.

The central physical moments are invariant to translation of the object. They change with scale change and rotation of the object.

Normalized central physical moment

The normalized central geometric moments are attached to the same system of coordinate as the central moments. The only difference is that the scale of this system is chosen to be equal to 1.

Definition 5.10. Assume that $cm_{k,l}(H)$ was computed by (5.19) and $m_{k,l}(H)$ by (5.18). Then *the normalized central physical moment* is defined as

$$ncm_{k,l}(H) = \frac{cm_{k,l}(H)}{m_{k,l}(H)^{\frac{k+l+2}{2}}} \ . \tag{5.20}$$

The normalized central geometric moments are invariant to translation and scale change.

However, invariance for rotation of the object is also required for good pattern recognition. For this, *the principal physical moments* has to be introduced.

Principal physical moment

The principal physical moments are attached to the modified principal coordinate system desribed on pages 53 - 56. The modification consists of replacing $M_{k,l}(G)$ by $m_{k,l}(H)$. The calculation of the principal physical moments then follows the same steps as the calculation of the principal moments. First, the computation of the angle α_j of rotation of our axes from the general system of coordinates has to be done. Then, the α_j satisfying the condition of principal coordinate system is found and denoted as g (see formula (5.13)).

Assume that $ncm_{k,l}(H)$ was obtained by the formula (5.20). Let us denote

$$\begin{aligned} S_g &= \sin g \ , \\ C_g &= \cos g \ , \\ ncm_{k,l} &= ncm_{k,l}(H) \ , \end{aligned}$$

then it holds that

$$pm_{2,0}(H) = C_g^2 \ ncm_{2,0} + 2S_gC_g \ ncm_{1,1} + S_g^2 \ ncm_{0,2} \tag{5.21}$$

$$pm_{0,2}(H) = S_g^2 \ ncm_{2,0} - 2S_gC_g \ ncm_{1,1} + C_g^2 \ ncm_{0,2} \tag{5.22}$$

$$\begin{aligned} pm_{3,0}(H) &= C_g^3 \ ncm_{3,0} + 3S_gC_g^2 \ ncm_{2,1} + 3S_g^2C_g \ ncm_{1,2} \\ &\quad + S_g^3 \ ncm_{0,3} \\ pm_{2,1}(H) &= -S_gC_g^2 \ ncm_{3,0} + (C_g^3 - 2S_g^2C_g) \ ncm_{2,1} \\ &\quad - (S_g^3 - 2S_fgC_g^2) \ ncm_{1,2} + S_g^2C_g \ ncm_{0,3} \\ pm_{1,2}(H) &= S_g^2C_g \ ncm_{3,0} + (S_g^3 - 2S_gC_g^2) \ ncm_{2,1} \\ &\quad + (C_g^3 - 2S_g^2C_g) \ ncm_{1,2} + S_g^3 \ ncm_{0,3} \\ pm_{0,3}(H) &= -S_g^3 \ ncm_{3,0} + 3S_g^2C_g \ ncm_{2,1} - 3S_gC_g^2 \ ncm_{1,2} \\ &\quad + C_g^3 \ ncm_{0,3} \end{aligned}$$

The principal physical moments of fourth order are written in [25].

Although principal physical moments are not depending only on the shape but also on the density of the object (thus the pixel values), their geometrical interpretations can still be described.

Equimomental ellipse

Definition 5.11. Equimomental ellipse of the object H is such an ellipse which has constant density less than or equal to m (see formula (5.23)) which approximates the object in the way, that they have the same physical moments up to the second order, inclusive.

$$m = \frac{1}{n} \sum_i \sum_j f(x_i, y_j) \ , \tag{5.23}$$

where (i, j) goes through each pixel of the object, $f(x_i, y_j)$ is pixel value and n is number of pixels in the object. The semi-axes of this ellipse in principal coordinate system are defined as

$$\begin{aligned} a &= 2\sqrt{pm_{2,0}(H) \cdot m} \ , \\ b &= 2\sqrt{pm_{0,2}(H) \cdot m} \ , \end{aligned}$$

where $pm_{2,0}(H)$ and $pm_{0,2}(H)$ are computed according to the formulas (5.21) and (5.22).

Equimomental ellipse of the object H is shown in Figure 5.9

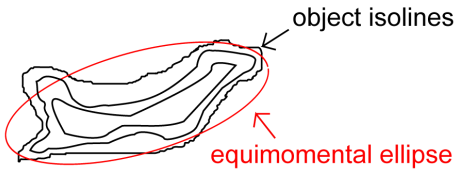


Figure 5.9: Equimomental ellipse of object H with depicted isolines, thus density level curves.

Elongation and dispersion have similar results as in geometric moments theory, therefore only different formulas are shown

$$\begin{aligned} el &= \log_2\left(\frac{a}{b}\right) , \\ dp &= \log_2(\pi \cdot a \cdot b \cdot m) , \end{aligned}$$

where m is computed by formula (5.23).

Graphical outputs can be seen in Figures 5.7 and 5.8.

Thereby the introduction to moment's theory ended. In the next subsections, the sketch of pattern recognition with geometric or physical moments is shown.

5.1.3 Pattern recognition using physical and geometric moments

In the following part, unless stated, no distinction is made between principal moments and principal physical moments.

Pattern recognition consists of two steps:

1. **Training**: calculation of tolerance intervals (see Definition 2.10) for principal moments
2. **Testing**: finding out if the principal moments of the object lie in those tolerance intervals

Training

At start, a training set of objects is needed. A sufficient set is 30 objects, but it is better to have up to 100 objects. Then, the appropriate principal moments for the approach have to be selected. Further, the tolerance intervals are calculated for all selected principle moments. Since the principal moments are assumed to have normal distribution (see Definition 2.1), the tolerance intervals are calculated according to the Definition 2.10.

Testing

The test of the hypothesis, that the unknown object fits to the set defined by our training set, is performed. It means, that the hypotheses is accepted if all the principal moments (of the orders selected in the training set) are within the tolerance intervals. If any of them lies outside the relevant interval, then the result is uncertain.

Pattern recognition using geometric moments is independent on change of light, camera, focus, etc. When the physical moments are used, one has to be aware of the dependence on any variance that causes a change of pixel values. This is reason why this method is not used very often in practise (since it is not easy to always have same conditions).

For our purpose, this should not be a problem as the photos are taken from SOHO (se section 1.2.1). This means that the photos are always taken with the same camera, focus, exposition etc. The only change that is different on each photo is luminosity of the sun, which could be a problem.

5.2 Usage in the filtration

At the beginning of this chapter it was said that the moment method is used mainly as a pattern recognition tool. The impulse noise (see Definition 3.5) can be considered as a pattern since it has its own specific features like shape, size, lightness etc. Therefore, when filtering impulse noise, the detection part (see subsection 4.1.1) can be replaced by moment method, and the correction part (see subsection 4.1.2) remains unchanged.

The section below shows, if it is suitable replacement or not. The procedure shown there is coded in our program (see Appendix A on page 121).

5.3 Implementation

In this section, we do not distinguish between principal moments and principal physical moments, if it is not specified.

5.3.1 Routine

Firstly the training set of objects has to be chosen. For that, the detection of defective pixels is done as it is described in section 4.2. Then each continous area, which was found by that method, is taken as separate object.

Since the shape of impulse noise in the images is well known, the usable moments for our approach are general moment of zero order and principal moments of second and third order. The second order moments are used for computation of the elongation of the equimomental ellipse. All of these moments are computed and the tolerance intervals should be calculated. Our objects however are so specific and small, that the intervals degenerate to one point, ergo the intervals are only the exact value of the moments.

At this moment the objects are tested if they are impulse noise or not. As said above the tolerance intervals degenerated to one point. Consequently the test only checks the equivalence of the values.

Finaly, the correction of defective pixels is done by iterative method, which was described in section 4.2.

5.3.2 Results

Firstly, the geometric moments were used for the filtration. $M_{0,0}, PM_{3,0}, PM_{2,1}$ and EL were computed for each object in the training set. By the basic knowledges about the

impulse noise, the decision tree was created for its detection. It is depicted on Figure 5.10.

The problem of the basic procedure for filtration of impulse noise (see section 4.2) is that it is filtering also commonly occuring phenomena in the solar atmosphere like solar rays and CMEs (see section 1.1.2). The filtration improved by Moment method solves this problem and is now far more accurate then the basic procedure. Their comparison is depicted on Figure 5.11.

Secondly, the physical moments were computed, ergo $m_{0,0}, pm_{3,0}, pm_{2,1}$ and el of each object in our training set. Unfortunately it was found out, that the physical moments are not suitable for our approach since the variance of pixel values among the images is higher than initial expectation.

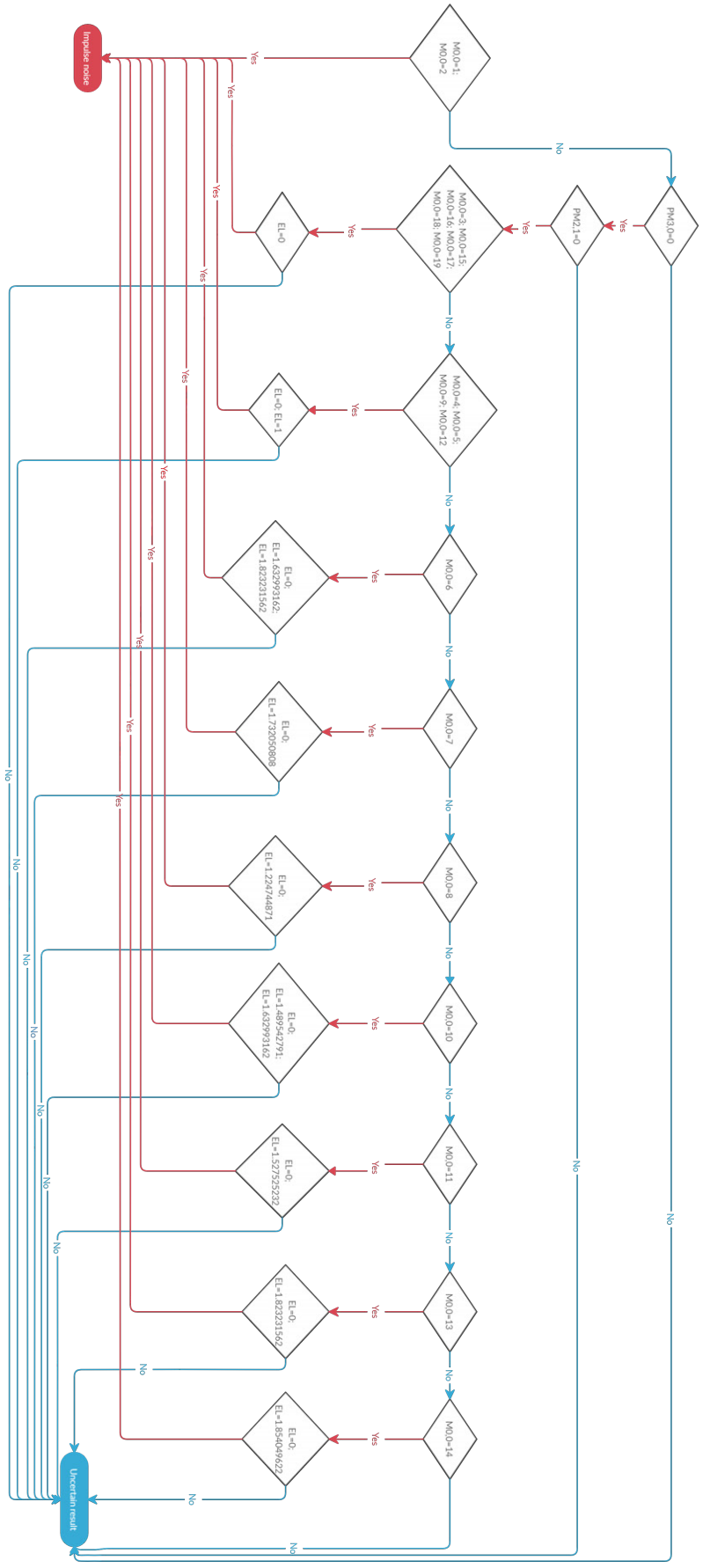
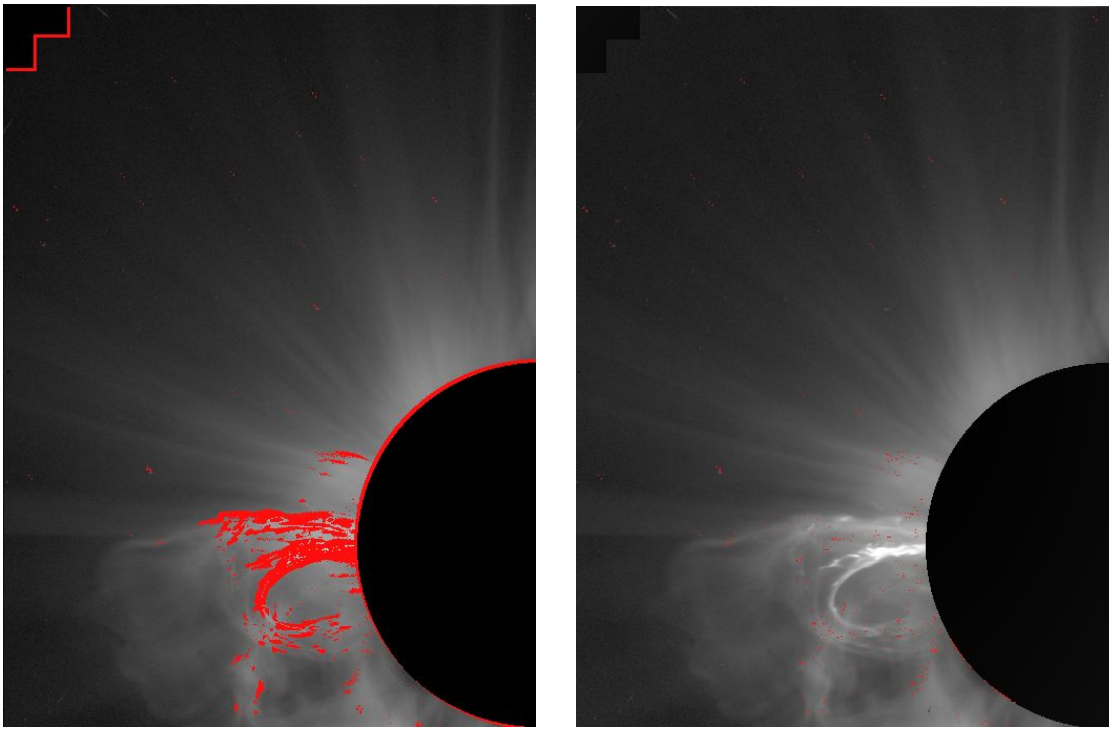


Figure 5.10: Decision tree for detection of impulse noise using geometric moments



(a) (b)

Figure 5.11: Comparison of the results from the detection of impulse noise. Image (a) was obtained by implementing the basic procedure. Image (b) was obtained by implementing the improved procedure, which is using moment's method. Testing image was captured on March 20, 2000 at 9:06:05 by SOHO LASCO C2.

Part III

Decomposition of the dynamical and nearly stable part of the solar corona from videos

Chapter 6

The actual state of knowledge

At the moment, there are a variety of decomposition techniques of video components. The phase correlation is one of the simpler methods used for this purpose. It is based on finding the shift between two frames using Fourier transform [27].

An improvement of this method is shown in [28]. The individual frames of the video are segmented into squares of different sizes according to the estimated movement. The phase correlation is then applied only to these segments. Another innovation, called phase amplified correlation, can be found in [29]. This method integrates the magnification of the motion, resulting in sub-pixel accuracy of the shift vector.

The weakness of all methods based on phase correlation is inability to work with videos that have different motion dynamics in foreground and background since the dominant peaks in the phase spectrum do not have to be the relevant ones. Consequently, the results of phase correlation are not applicable without a perfect knowledge of the phenomenon in the video and their effect on the shape of the phase correlation. Therefore, these methods are limited to videos with static backgrounds only.

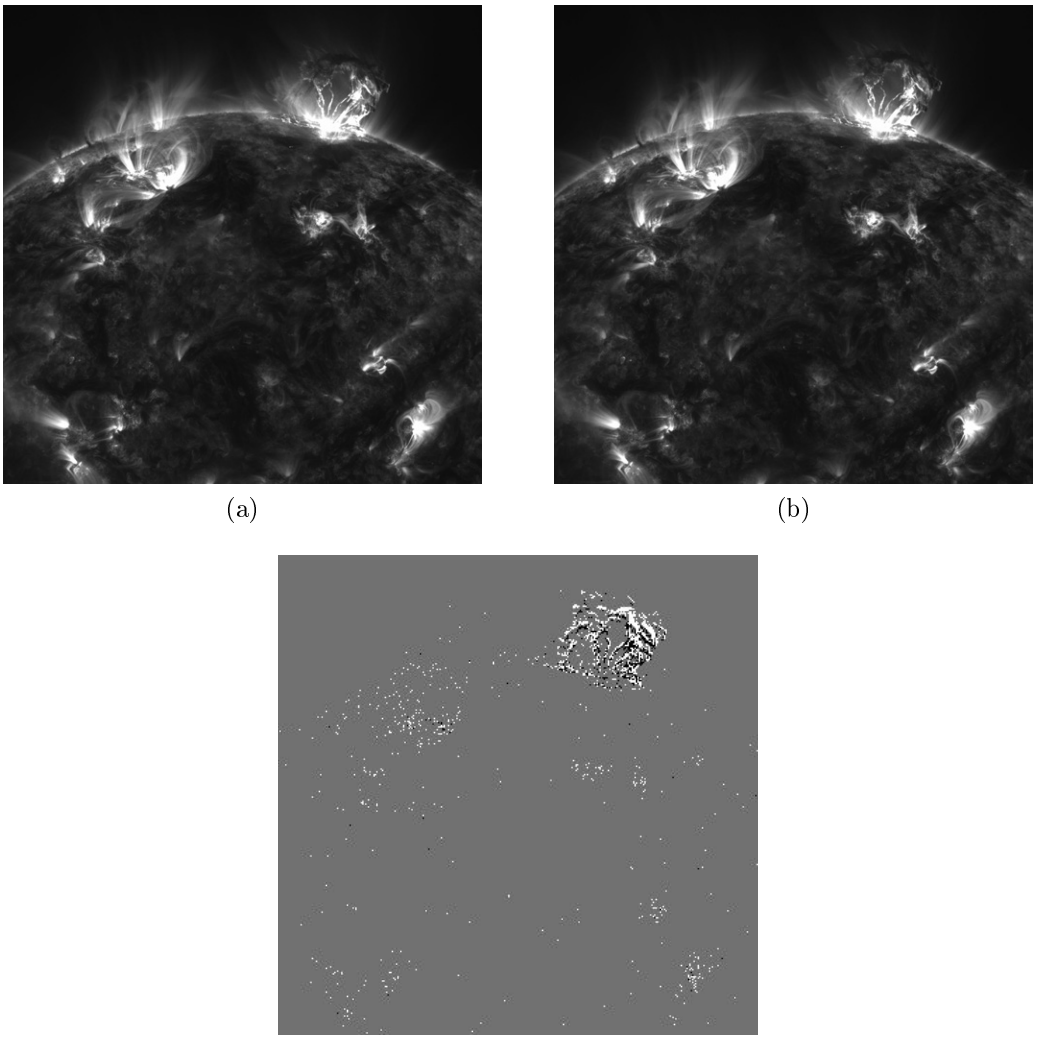
The concept of optical flow is another possible way of dealing with the movement of the objects in series of images. In [30] it is described as the distribution of the velocity of certain elements in the series of images based on their brightness. Some of the methods estimating optical flow are given below

- Lucas–Kanade method based on an affine model of the flow [31],
- Horn–Schunck method based on preserving the brightness and smoothness of the flow [31],
- Buxton–Buxton method based on the detection of edges and the motion of these edges [32].

The disadvantage of methods based on optical flow is the assumption of brigthness consintency through the series of images. This assumption is not fulfilled in the solar observation videos.

The running difference method is a well-established technique for separating components of solar observation video [33], [34]. It is based on the subtraction of subsequent video frames. This method has the advantage of being very easy to implement with a clear mathematical definition. However, it has two disadvantages. Firstly, this method is

sensitive to a wide variety of signals, making it difficult to interpret the results [35]. Secondly, it shows a time derivative rather than the true CME structure, making it difficult to use for structural interpretation of CMEs [36]. The results are shown in Figure 6.1. It can be seen, that the decomposed image also shows other chaotic motions in the inner corona than the desired CMEs (see section 1.1.2).



(a) (b)

(c)

Figure 6.1: Decomposition of the solar corona from the image sequence using the running difference method. The test images were taken by SDO - AIA 171 Å on April 16, 2012 a) 17:35:36 UTC and b) 17:35:50 UTC. The image c) shows the result of the running difference method, the grey color represents the background, and the white and black pixels represent the different brightness levels between test images. It can be seen that those pixels occur in the whole image, which means that this method separate all chaotic motions in the images, not only the CME

The method known as base difference images addresses the issue of running differences to a certain extent, as evidenced by the findings of [37]. The method involves selecting a frame from the time series and using it as a reference image. The differences between this reference image and each of the other frames are then computed. This process allows for the identification of whether the image returns to the background level or not. However,

this method is limited by its sensitivity to any drift of the image with respect to the reference. Consequently, after a certain period of time, the ability to identify smaller and more rapid variations is reduced.

The method based on the previous two is called the running center-median filter [35]. It calculates the difference between the current image and the per-pixel median of all images falling within a given time interval of the current image. Using the median avoids the limitations of the running difference and base difference methods described above. The principal limitation of this method is its relative insensitivity in comparison to the running difference. Indeed, this method is sensitive to transient variation from timescales of approximately half the window width down to the frame-to-frame scale, whereas the running difference is most effective for detecting variation at the half window width scale.

An alternative approach, multiscale wavelet decomposition, is also effective in revealing CME structure [38]. It is a more sophisticated method than running differences, but unfortunately it also enhances other high-frequency components, such as the edges of streamers.

Another possibility is presented in the paper [36]. It aims to improve the automated detection and cataloging of CMEs from coronagraphs images. The described process is based on an iterative deconvolution in time and space that aims to isolate the CME from the background in a white light image, resulting in separate images of the dynamic and quiescent components of the corona. According to the authors of paper [36], one of the limitations of this technique is that any feature in the coronagraphic data that is not smooth in the radial and temporal dimensions, e.g. noise, will be seen in the final separated CME component. Even very faint dynamical events are therefore present in the CME component images.

The goal of this thesis is to find out whether the methods based on sparse representations are suitable for decomposing dynamic and nearly stable part of the solar corona from videos. These methods are tested on images from SDO - AIA (see section 1.3). The proposed methods are explained in the next sections.

Chapter 7

Decomposition Techniques

7.1 Preprocessing of the data

The first step is to process the input data. Assume, that the input is composed of $m \in \mathbb{N}$ consecutive images. Each image can be interpreted as the matrix of pixel values $\mathbf{M}_i,\ i \in \langle 1, m \rangle$. The final matrix $\mathbf{M}$, which is used in the decomposition techniques is obtained as follows

$$\mathbf{M} = [\text{vec } \mathbf{M}_1, \text{vec } \mathbf{M}_2, \cdots, \text{vec } \mathbf{M}_m],$$

where vec $\mathbf{M}_i, i \in \langle 1,\ m \rangle$ stands for the vectorization of the matrix. Now each row of the matrix $\mathbf{M}$ represents values of one pixel in time.

7.2 Basic idea

Assuming that there is not any rapid movement in our video, then the columns of matrix $\mathbf{M}$ are alike, so the matrix is recognized as a low-rank matrix (see Definition 2.13). On the contrary, if there is any movement, a few pixels per row differ from the surroundings. If only those pixels are kept and the rest are changed to zero, then the generated matrix is called a sparse matrix (see Definition 2.28). Based on these ideas the matrix $\mathbf{M}$ can be decomposed to low-rank matrix $\mathbf{L}$ (static component) and sparse matrix $\mathbf{S}$ (dynamic component) such that $\mathbf{M} = \mathbf{L} + \mathbf{S}$. Thus the suitable methods for our purpose could be the ones based on sparse representations. The different approaches to obtain the matrices $\mathbf{L}$ and $\mathbf{S}$ are shown in the following sections.

7.3 Median filter method

The easiest way to decompose matrix $\mathbf{M}$ into the low-rank and sparse matrix is an application of the median filter. The classical model works with the assumption that the columns of the matrix $\mathbf{M}$ are alike and the only differences between them are elements of matrix $\mathbf{S}$. Also assume that individual pixels have the same value in more than half of the frames. Then, since the median is the value that halves a series of ascending data, the background matrix $\mathbf{L}$ is obtained as the median over rows of $\mathbf{M}$, and the matrix $\mathbf{S}$ is a difference of $\mathbf{M}$ and $\mathbf{L}$, i.e. $\mathbf{S} = \mathbf{M} - \mathbf{L}$.

The primary benefit of applying a median filter is its ability to withstand extreme values, which is a result of the median's inherent properties. A straightforward implemen-

tation requiring only a few lines in nearly all programming languages further strengthens its appeal.

One significant restriction of the median filter is that if the input data involves a large number of moving objects, there could be more images that contain a dynamic component than those that show the background. In such a scenario, the median acquired for each row no longer matches the background values. Thus, the calculated background estimation is heavily prejudiced.

7.4 Principal Component Pursuit (PCP)

7.4.1 Basics of the PCP Technique

The second method simultaneously minimizes the number of different columns in the matrix $\mathbf{L}$, and also minimizes the number of nonzero elements in the matrix $\mathbf{S}$. For our dataset, this method calculates the matrices $\mathbf{L}$ (background) and $\mathbf{S}$ (dynamic component) in parallel. In other words, it tries to minimise both the almost static motion associated with the background and the fast motion associated with the dynamic component.

Mathematicaly speaking, the problem of decomposition $\mathbf{M}$ into low rank matrix $\mathbf{L}$ and sparse matrix $\mathbf{S}$ can be written as following optimization problem

$$\min_{\mathbf{L},\mathbf{S}} \operatorname{rank}(\mathbf{L}) + \|\mathbf{S}\|_0 \quad \text{subject to } \mathbf{L}+\mathbf{S}=\mathbf{M}, \tag{7.1}$$

since the ℓ_0 implies the number of nonzero elements of the matrix (see Definition 2.26) and rank corresponds to the maximal number of linearly independent columns (see Definition 2.13). Unfortunately, both rank and ℓ_0 norm make this problem non-convex. Therefore, it cannot be solved in feasible time.

According to [39], the low rank plus sparse decomposition (7.1) can be approximated as a convex optimization problem (PCP)

$$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \quad \text{subject to } \mathbf{L}+\mathbf{S}=\mathbf{M}, \tag{7.2}$$

where $\|\mathbf{L}\|_*$ denotes nuclear norm of the matrix $\mathbf{L}$ (see Definition 2.29), $\|\mathbf{S}\|_1$ denotes ℓ_1 norm of the matrix $\mathbf{S}$ (see Definition 2.24)and λ is a balancing parameter. Based on [39], the equation (7.2) has an exact solution under the assumption that $\mathbf{M}$ is not both low rank and sparse. Thus, first, it is necessary to ensure that the low-rank element $\mathbf{L}$ is not sparse at the same time. This can be accomplished by using the so-called *incoherence conditions*. Assume that the SVD of $\mathbf{L} \in \mathcal{M}_{m,n}(\mathbb{R})$ (see Definition 2.35) is $\mathbf{L}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$, with $\sigma_1 \geq \cdots \geq \sigma_p \geq 0$. Then, the incoherence condition with parameter μ states that

$$\max_i \|\mathbf{U}^*\mathbf{e}_i\|_2 \leq \frac{\mu p}{m}, \quad \max_i \|\mathbf{V}^*\mathbf{e}_i\|_2 \leq \frac{\mu p}{n} \quad \forall i \in \langle 1,p\rangle$$

and

$$\|\mathbf{U}\mathbf{V}^*\|_\infty \leq \sqrt{\frac{\mu p}{mn}},$$

where $p=\operatorname{rank}(\mathbf{L})$, $\mathbf{e}_i$ is unit vector, $\|\cdot\|_2$ is ℓ_2 norm (see Definition 2.24) and $\|\cdot\|_\infty$ is ℓ_∞ norm (see Definition 2.25). These conditions ensure that the singular vectors are reasonably spread out for small values of μ. Another issue may arise if the matrix $\mathbf{S}$,

which is sparse, is also low rank. Hence, we will insist on the sparse matrix having its non-zero elements distributed uniformly.

According to [39], matrices $\mathbf{L}$ whose singular vectors are reasonably spread can be recovered with probability nearly one from arbitrary and completely unknown moving patterns (as long as these are randomly distributed). Moreover it is found that there exists an universal λ, precisely $\lambda = \frac{1}{\sqrt{\tilde{n}}}$, where $\tilde{n} = \max(m, n)$ for the matrix $\mathbf{L}_{m,n}$. The presence of this universal tuning parameter λ, represents a significant advantage of the PCP algorithm. This is because the parameter guarantees the convergence of the problem, regardless of the matrices $\mathbf{L}$ and $\mathbf{S}$. One disadvantage of the PCP algorithm is the high time consumption caused by the computation of the SVD at each iteration.

7.4.2 Algorithm

The PCP algorithm is derived from Alternating Direction Method of Multipliers (ADMM) applied to (7.2). The principle of ADMM is described for example in [40]. This method exploits augmented Lagrangian, which for the problem (7.2) takes the form

$$L_\mu(\mathbf{L}, \mathbf{S}, \mathbf{Y}) = \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \langle \mathbf{Y}, \mathbf{M} - \mathbf{L} - \mathbf{S}\rangle + \frac{\mu}{2}\|\mathbf{M} - \mathbf{L} - \mathbf{S}\|_{\mathrm{F}}^2, \tag{7.3}$$

where $\mathbf{Y}$ is a matrix of the Lagrange multiplier, μ is penalty paramater and $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm (see Definition 2.23).

The equation (7.3) can be reformulated to represent Augmented Lagrangian in its scaled form, which is more appropriate for the following calculations.

$$L_\mu(\mathbf{L}, \mathbf{S}, \mathbf{Y}) = \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \frac{\mu}{2}\|\mathbf{M} - \mathbf{L} - \mathbf{S} + \mathbf{Y}/\mu\|_2^2 - \frac{\mu}{2}\|\mathbf{Y}/\mu\|_2^2 \tag{7.4}$$

Since (7.4) is a convex problem, the ADMM is eligible to find the global minimum. The ADMM design for equation (7.4) is

$$\begin{aligned}
\mathbf{L}^{(k+1)} &= \arg\min_{\mathbf{L}} \left(\frac{\mu}{2}\|\mathbf{M} - \mathbf{L} - \mathbf{S}^{(k)} + \mathbf{Y}^{(k)}/\mu\|_2^2 + \|\mathbf{L}\|_*\right) \\
\mathbf{S}^{(k+1)} &= \arg\min_{\mathbf{S}} \left(\frac{\mu}{2}\|\mathbf{M} - \mathbf{L}^{(k+1)} - \mathbf{S} + \mathbf{Y}^{(k)}/\mu\|_2^2 + \lambda\|\mathbf{S}\|_1\right) \\
\mathbf{Y}^{(k+1)} &= \mathbf{Y}^{(k)} + \mu\left(\mathbf{M} - \mathbf{L}^{(k+1)} - \mathbf{S}^{(k+1)}\right)
\end{aligned}$$

Proximal operators can be utilized for numerically solving convex optimization problems [41]. In the case of problem (7.2), soft tresholding and singular value tresholding are utilized, which correspond to $\|\cdot\|_1$ and $\|\cdot\|_*$, respectively. According to [17], the soft tresholding scheme is as follows

$$\mathrm{soft}_\tau(x_i) = \frac{x_i}{|x_i|}\max(|x_i| - \tau, 0);\ \tau > 0\ . \tag{7.5}$$

Assume that singular value decomposition of $\mathbf{X}_{m,n}$ is $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$. The elements of $\mathbf{\Sigma}$ are denoted as σ_i and the vectors of $\mathbf{U}, \mathbf{V}^*$ are denoted as $\mathbf{u}_i, \mathbf{v}_i^*$. Then, according to [17], the singular value tresholding scheme is as follows

$$\mathrm{svt}_\tau(\mathbf{X}) = \sum_{i=1}^{n} \mathrm{soft}_\tau(\sigma_i)\mathbf{u}_i\mathbf{v}_i^*\ , \tag{7.6}$$

where τ is a balancing parameter. More on proximal operators can be found in [42].

Now, by using the proximal operators (7.5) and (7.6) the PCP algorithm is obtained. (Based on [39])

Algorithm 1: Principal Component Pursuit

Result: $\mathbf{L}$, $\mathbf{S}$

initialization: $\mathbf{S}^{(0)} = \mathbf{Y}^{(0)} = 0,\ \mu > 0$;

while $\|\mathbf{M} - \mathbf{L}^{(k+1)} - \mathbf{S}^{(k+1)}\|_{\mathrm{F}} > \delta\|\mathbf{M}\|_{\mathrm{F}}$ **do**

$\mathbf{L}^{(k+1)} = \mathrm{svt}_{\frac{1}{\mu}}\left(\mathbf{M} - \mathbf{S}^{(k)} + \mathbf{Y}^{(k)}/\mu\right)$;

$\mathbf{S}^{(k+1)} = \mathrm{soft}_{\frac{\lambda}{\mu}}\left(\mathbf{M} - \mathbf{L}^{(k+1)} + \mathbf{Y}^{(k)}/\mu\right)$;

$\mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + \mu\left(\mathbf{M} - \mathbf{L}^{(k+1)} - \mathbf{S}^{(k+1)}\right)$;

end

7.5 Dynamic Mode Decomposition (DMD)

The final method proposed in this paper is the DMD method. It is a recent, data-driven method, first used in hydro-mechanics. Since then, the popularity of this method has grown, mainly because it is a great tool for understanding and controlling nonlinear dynamical systems without full knowledge of them.

The main concept of DMD is to simplify the intricate workings of a system using a linear model. This allows us to extract the key modes that are responsible for its behaviour. DMD derives these modes and their respective frequencies by taking measurements that depict the system's behaviour across time. By analysing these dominant modes, we gain insight into the dynamics of the system and can even predict its future behaviour. But we're just discussing the immediate future because in most cases the imprecision gained from DMD decomposition rises exponentially over time.

The following sections are based on [43] and [44].

7.5.1 Basics of the DMD technique

In this section, we present the method for the general base case, which involves predicting the next step of an arbitrary system. This theoretical section is essential for the correct application of the method to video component separation. Subsequently, in the algorithm section, we demonstrate the adaptation to our dataset.

Assume, that m observations were obtained in evenly spaced time instants $t_i, i \in \langle 1, m\rangle$ and each observation consists of n data points. Let us denote these observations as vectors $\mathbf{x}_i, \in \langle 1, m\rangle$, then the following matrices can be set

$$\begin{aligned}\mathbf{X}_1 &= [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{m-1}],\\ \mathbf{X}_2 &= [\mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_m].\end{aligned}$$

Now, the linear operator $\mathbf{A}$ has to be found, such that the data from time t_i are mapped to the data from time t_{i+1}, i.e. $\mathbf{x}_{i+1} = \mathbf{A}\mathbf{x}_i$. Operator $\mathbf{A}$ is aproximation of the Koopman operator ([45]).

The DMD algorithm uses the operator $\mathbf{A}$ in such a way that it is the best representation of $\mathbf{X}_1$ while the columns of

$$\mathbf{X}_1 = [\mathbf{x}_1, \mathbf{A}\mathbf{x}_1, \ldots, \mathbf{A}^{m-2}\mathbf{x}_1]$$

form a Krylov subspace. Thus $\mathbf{X}_2 = \mathbf{A}\mathbf{X}_1$. The operator $\mathbf{A}$, which has the form of a square matrix, can then be computed in a least-squares sense as follows

$$\mathbf{A} = \mathbf{X}_2\mathbf{X}_1^+ \ . \tag{7.7}$$

Such a calculation would be time-consuming so the SVD (see Definition 2.35) of $\mathbf{X}_1$ is used for the reduction of its dimension.

The level of reduction of the dimension is determined by the parameter r, it is chosen to reduce the dimension as much as possible while keeping the basic structure of the system described by $\mathbf{X}_1$. The singular value decomposition is calculated as follows: $\mathbf{X}_1 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$, where $\mathbf{U} \in \mathcal{M}_{n,r}(\mathbb{C})$, $\mathbf{\Sigma} \in \mathcal{M}_{r,r}(\mathbb{C})$ and $\mathbf{V} \in \mathcal{M}_{(m-1),r}(\mathbb{C})$. The equation (7.7) can be rewritten as

$$\mathbf{A} = \mathbf{X}_2\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^*.$$

The DMD algorithm uses eigenvalues of matrix $\mathbf{A}$ for estimation of the future behavior of the system. This would be a long process, so the Arnoldi algorithm [46] is used to compute the eigenvalues of the similar matrix $\tilde{\mathbf{A}}$, since they can be taken as an approximation of the eigenvalues of the matrix $\mathbf{A}$. The similar matrix is computed with $\mathbf{U}$, i.e.

$$\tilde{\mathbf{A}} = \mathbf{U}^*\mathbf{A}\mathbf{U}. \tag{7.8}$$

The matrix of eigenvalues of $\tilde{\mathbf{A}}$ is denoted as $\mathbf{\Lambda}$ and the matrix of corresponding eigenvectors is denoted as $\mathbf{W}$. Then the equation (7.8) can be rewritten as

$$\begin{aligned} \mathbf{A}\mathbf{U} &= \mathbf{U}\tilde{\mathbf{A}} = \mathbf{U}\mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1}, \\ \mathbf{A}\mathbf{U}\mathbf{W} &= \mathbf{U}\mathbf{W}\mathbf{\Lambda}, \\ \mathbf{A}\mathbf{\Phi} &= \mathbf{\Phi}\mathbf{\Lambda}, \end{aligned}$$

where $\mathbf{\Phi} = \mathbf{U}\mathbf{W}$ is the matrix of so called projected DMD modes. Thus the j-th DMD mode is computed as $\boldsymbol{\varphi}_j = \mathbf{U}\mathbf{w}_j$.

For having the option to predict the future behavior of the dynamical system, the Fourier mods ω_j have to be introduced. They are computed as

$$\omega_j = \frac{\ln(\lambda_j)}{\Delta t},$$

where λ_j are elements of $\mathbf{\Lambda}$ and $\Delta t = t_{j+1} - t_j$. Since DMD eigenvalues λ_j tend to be complex, we calculate the logarithm of the complex number in polar coordinates. Then the real part of ω_j defines the growth and descent of DMD modes, meanwhile, the imaginary part defines their oscillation.

Therefore, the DMD prediction for time $t > t_1 = 0$ is

$$\mathbf{x}_{\mathrm{DMD}}(t) = \sum_{j=1}^{r} b_j\boldsymbol{\varphi}_j e^{\omega_j t} = \mathbf{\Phi}e^{\mathbf{\Omega}t}\mathbf{b}, \tag{7.9}$$

where $\mathbf{\Omega}$ is diagonal matrix containing the Fourier descriptors ω_j and $\mathbf{b}$ is obtained from the initial condition $\mathbf{x}_{\mathrm{DMD}}(0) = \mathbf{\Phi}\mathbf{b}$, i.e. $\mathbf{b} = \mathbf{\Phi}^+\mathbf{x}_1$.

7.5.2 Algorithm

Since the video is a sequence of images captured evenly in time, and each image consists of the same amount of data points, it is straightforward to rewrite it as matrices $\mathbf{X}_1, \mathbf{X}_2$. The first $m-1$ columns of $\mathbf{M}$ are the matrix $\mathbf{X}_1$ and last $m-1$ columns form matrix $\mathbf{X}_2$.

The formula (7.9) predicts only one step, i.e. one image ahead. To obtain the whole video, the time vector $\mathbf{t} = [t_1, t_2, \ldots, t_m]$ has to be implemented. The reconstruction of the video using DMD mode is done as follows

$$\mathbf{X}_{\mathrm{DMD}} = \sum_{j=1}^{r} b_j \boldsymbol{\varphi}_j e^{\omega_j \mathbf{t}} = \boldsymbol{\Phi} e^{\boldsymbol{\Omega} \mathbf{t}} \mathbf{b}. \tag{7.10}$$

Again, the vector $\mathbf{b}$ is obtained from $\mathbf{x}_1 = \boldsymbol{\Phi}\mathbf{b}$. Thus, the diagonal matrix $\boldsymbol{\Omega}$ defines the changes of the first image in time. Any component of a video that does not change over time, or changes very little, has a corresponding Fourier mode (ω_j) near the origin in the complex space, i.e. $|\omega_j| \approx 0$. This property is the basic idea behind using DMD for background/foreground separation.

Assume that for $\omega_p,\ p \in P \subset R = \{1, 2, \ldots, r\}$ holds $|\omega_p| \approx 0$ and for $\forall j \neq p$ the $|\omega_j|$ is not close to zero, then

$$\mathbf{X}_{DMD} = \underbrace{b_p \boldsymbol{\varphi}_p e^{\omega_p \mathbf{t}}}_{\text{background}} + \underbrace{\sum_{j \neq p} b_j \boldsymbol{\varphi}_j e^{\omega_j \mathbf{t}}}_{\text{dynamical part}}.$$

The individual elements $b_j \boldsymbol{\varphi}_j e^{\omega_j \mathbf{t}}$ are complex, only the sum of all elements forms real matrix. This is quite a problem, because only real values are usable for separation. The problem can be solved with the property $\mathbf{M} = \mathbf{L} + \mathbf{S}$. Since the low rank matrix (background) looks as follows

$$\mathbf{L} = b_p \boldsymbol{\varphi}_p e^{\omega_p \mathbf{t}},$$

the sparse matrix (dynamical part) can be obtained as follows

$$\mathbf{S} = \mathbf{M} - |\mathbf{L}|\ , \tag{7.11}$$

where $|\cdot|$ denotes the absolute value of each element of the matrix $\mathbf{L}$, since $\mathbf{L}$ computed in this way may contain negative values.

The computation (7.11) may result in $\mathbf{S}$ containing negative elements, which is inappropriate since the pixel values can only be positive. Consequently, the residual matrix $\mathbf{R} \in \mathbb{C}^{n \times m}$ containing the negative values of $\mathbf{S}$, has to be implemented. The matrix $\mathbf{R}$ is then added to $|\mathbf{L}|$ and subtracted from $\mathbf{S}$ to satisfy the property $\mathbf{M} = \mathbf{L} + \mathbf{S}$, thus mathematically speaking

$$\begin{aligned} \tilde{\mathbf{L}} &= |\mathbf{L}| + \mathbf{R}\ , \\ \tilde{\mathbf{S}} &= \mathbf{S} - \mathbf{R}\ , \\ \mathbf{M} &= \tilde{\mathbf{L}} + \tilde{\mathbf{S}}\ . \end{aligned}$$

The DMD algorithm was taken and modified from book [47].

Algorithm 2: Dynamic Mode Decomposition

Result: $\tilde{\mathbf{L}}$, $\tilde{\mathbf{S}}$
initialization: $\mathbf{M}$, $\mathbf{X}_1$, $\mathbf{X}_2$, $\delta > 0$, $r > 0$, $\mathbf{\Lambda}$, $\mathbf{\Phi}$, $\mathbf{L} = \mathbf{0}$;
for $j = 1 : r$ **do**
  if $\omega_j < \varepsilon$ **then**
    $\mathbf{L} = \mathbf{L} + b_j \boldsymbol{\varphi}_j e^{\omega_j \mathbf{t}}$;
  else
    $\mathbf{L} = \mathbf{L}$;
  end
end
$\mathbf{S} = \mathbf{M} - |\mathbf{L}|$;
$\mathbf{S} \to \mathbf{R}$;
$\tilde{\mathbf{L}} = |\mathbf{L}| + \mathbf{R}$;
$\tilde{\mathbf{S}} = \mathbf{S} - \mathbf{R}$;

One of the key advantages of DMD over PCP is its relatively low computational complexity. In both techniques, the most challenging step is the computation of the SVD. However, in DMD, this step is only performed once, which allows the DMD algorithm to be used for real-time video separation if implemented efficiently. The disadvantage of DMD is high noise sensitivity. This problem is discussed in [48].

7.6 Implementation

The algorithms were implemented in MATLAB R2019b. Input data are expected to be a series of images. These images have to be grayscale, which is fulfilled by using images from SDO - AIA. Original images have 4096×4096 pixels. We are using 500 consecutive images, thus one video takes approximately 67 GB. Because of the limited size of RAM on our computer (see Table 2 in Appendix B, page 123), only one-quarter of each image is used, i.e. the methods are computed on segments of size 2048×2048 pixels.

According to [22], the suitable frequency for our approach is 171 Å, since the CMEs are most visible there. Second tested freqency is 304 Å. More informations about these frequencies can be found in Table 1.1 in section 1.3. The methods were tested on three videos containing CMEs, precisely videos from June 7, 2011, April 16, 2012 and October 2, 2014.

7.6.1 NAFE enhancement

According to [22], the SDO-AIA images are typically characterised by extreme contrast between bright features in active regions and dark backgrounds containing low-contrast structures. In order to fully exploit the information contained in these images, we pre-processed them using NAFE (see section 3.3.2).

The output of this method can be influenced by three parameters, γ, ω and σ. The γ parameter affects the brightness of the image. Lower values for γ result in darker images and higher values result in brighter images. The constant ω is called the NAFE weight and its specification allows you to control the enhancement of the structures. The σ is the standard deviation of the Gaussian distribution of the added noise. The higher the

value of σ, the lower the amount of noise in the low contrast parts of the image. It is not correct to suppress the noise too much because the low contrast details will be lost. The values we have chosen for these parameters are written in the Table 7.1.

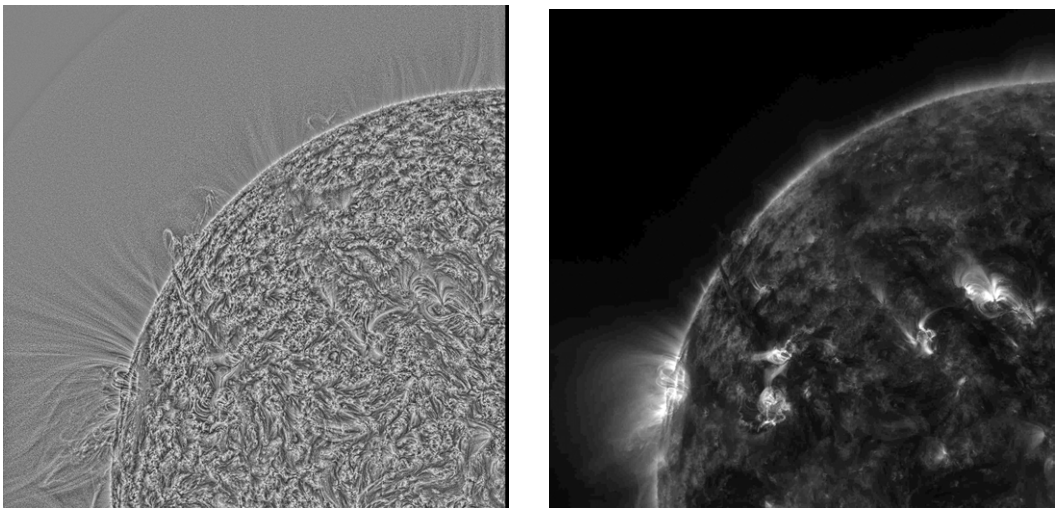
Table 7.1: NAFE parameter selection

Parameter	**Value**
γ	2.6
ω	0.2
σ	15

The preprocessing of the data was done as a linear combination of images from our dataset and images enhanced by NAFE. Thus, in the following way

$$\mathbf{M}_{\text{new}} = w \cdot \mathbf{M}_{\text{NAFE}} + (1 - w) \cdot \mathbf{M},$$

where the value of w was chosen as 1, 0.8, 0.6, 0.4, 0.2 and 0. The images $\mathbf{M}_{\text{NAFE}}$ and $\mathbf{M}$ are shown in Figure 7.1.



(a) (b)

Figure 7.1: The difference between a) NAFE processed image and b) unprocessed image. The images were taken by SDO-AIA 171 Å on April 16, 2012.

7.6.2 Depiction of the results

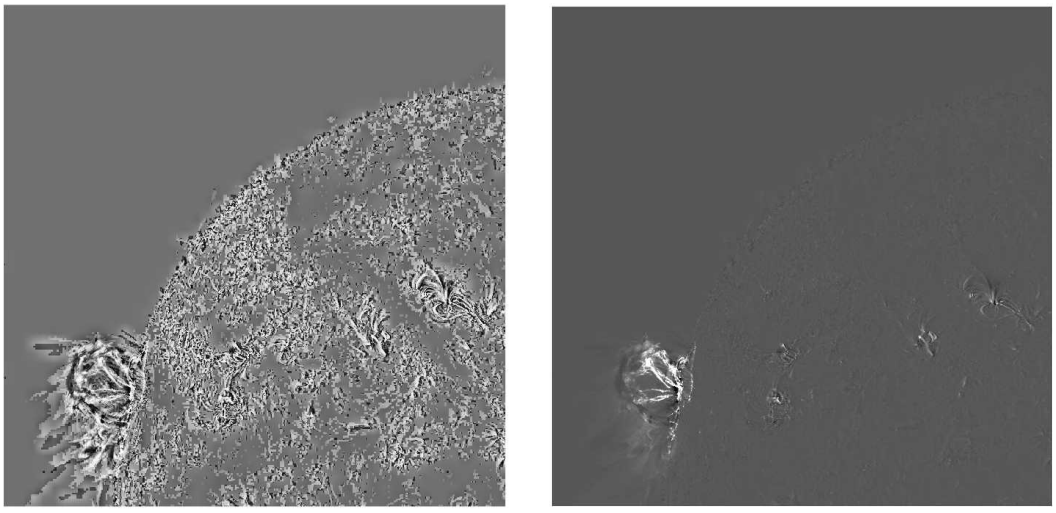
The images in MATLAB are stored in format `uint8`, unfortunately during the computation, the data have to be converted into format `double`, since the algorithms contain subtraction and division. Therefore, the output data needs to be transformed back into interval $\langle 0, 255 \rangle$.

The transformation is done as follows:

$$F_{i,j} = \begin{cases} B & \text{if } F_{i,j} < B, \\ \dfrac{F_{i,j} - B}{W - B} \cdot 255 & \text{if } B < F_{i,j} < W, \\ W & \text{if } F_{i,j} > W, \end{cases}$$

where $\mathbf{F}$ stands for matrices $\mathbf{M}$, $\mathbf{L}$, and $\mathbf{S}$. B is 0.001-quantile from $\mathbf{F}$ and W is 0.999-quantile from $\mathbf{F}$.

The resulting videos lack depth, thus the accuracy of the methods cannot be sufficiently evaluated. It is necessary to apply an image enhancement method (see section 3.3). Firstly we tried to use the method described in [49], which is based on enhancing the images by using the arctan function. This method was found to be unsuitable as it resulted in the suppression of those areas of the image where significant changes were occurring. It can be reasonably assumed, therefore, that the optimal method is the one that makes optimal use of all available brightness levels. Therefore, we have chosen to implement the adaptive histogram equalization (see section 3.3.1). The process of histogram equalization changes the histogram of the image to be as uniform as possible. The adaptivity of this process is based on the progressive changing of the histogram on segments of a given size, in our case square of size 50 pixels. Comparison between the resulting image with and without using an adaptive histogram is shown in Figure 7.2.



(a) (b)

Figure 7.2: Comparison between a resulting image a) with and b) without using adaptive histogram equalization. The images were taken by SDO-AIA 171 Å on April 16, 2012.

The following sections describe the particular routines and results of all proposed methods. The evaluation of the accuracy and quality of each method is conducted exclusively through visual analysis. The use of objective methods is precluded due to the inability to precisely define the correctly extracted object.

Firstly, the median filter method is presented (the theory is described in section 7.3).

7.6.3 Median filter method

Routine

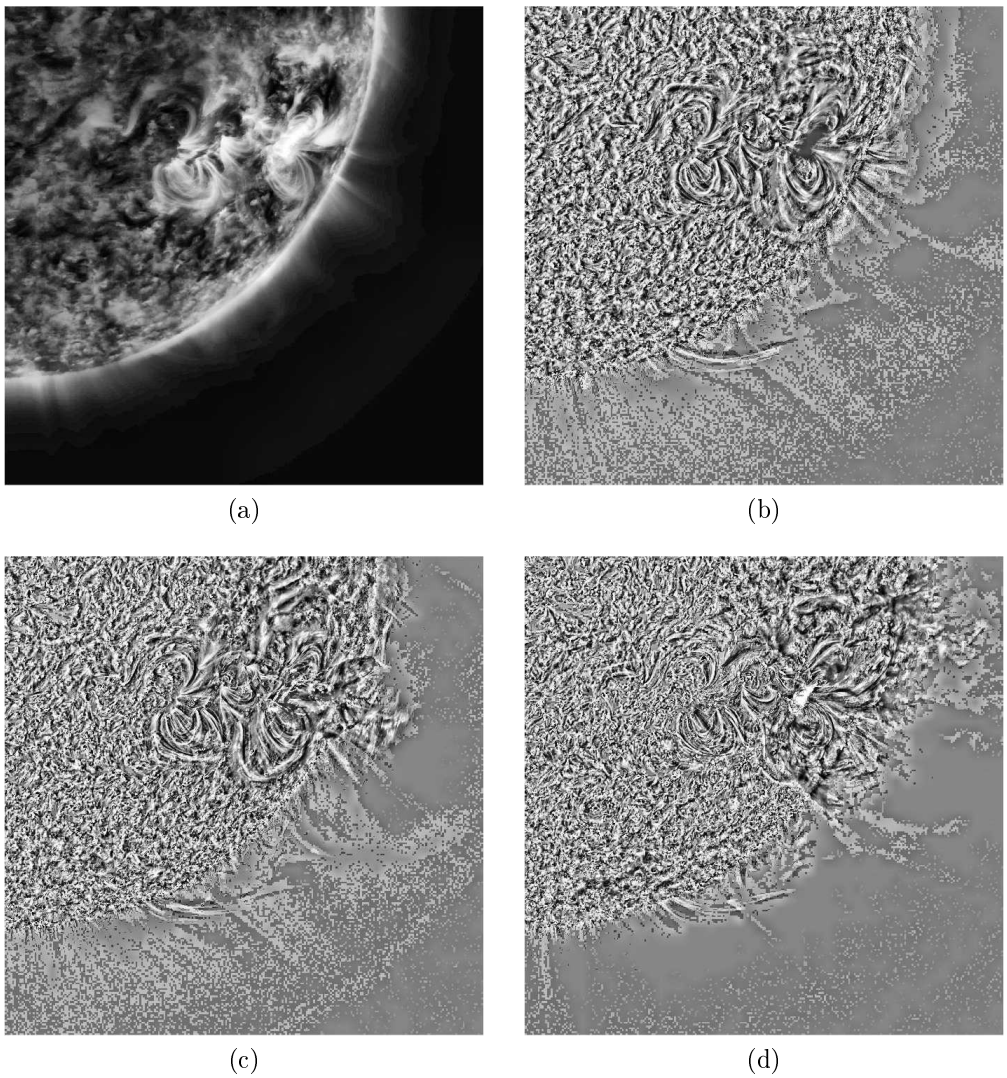
As said in theoretical section, the implementation of this method is very straightforward. Further, there is no need to specify the routine.

Results

The output of the Median filter method is one image of the background (i.e. a constant video) and a video of the dynamic component. First, we describe the results of applying the median filter to videos from SDO-AIA 171 Å.

SDO-AIA 171 Å

It can be seen in Figures 7.3, 7.4 and 7.5 that this method detects also additional motions in the inner corona besides desired CMEs, so it does not bring any improvement compared to the running difference method used nowadays. Further in Figure 7.3, it can be seen that if the vast majority of the CME motion in the images occurs directly in the solar space and not in the solar atmosphere the method does not work at all.



(a) (b) (c) (d)

Figure 7.3: Outcome of Median filter method without NAFE enhancement. It shows a) the computed image of the background (matrix $\mathbf{L}$) and the b) 50th, c) 150th, and d) 250th images from the video of the dynamic component (matrix $\mathbf{S}$). The images were taken by SDO-AIA 171 Å on June 7, 2011.

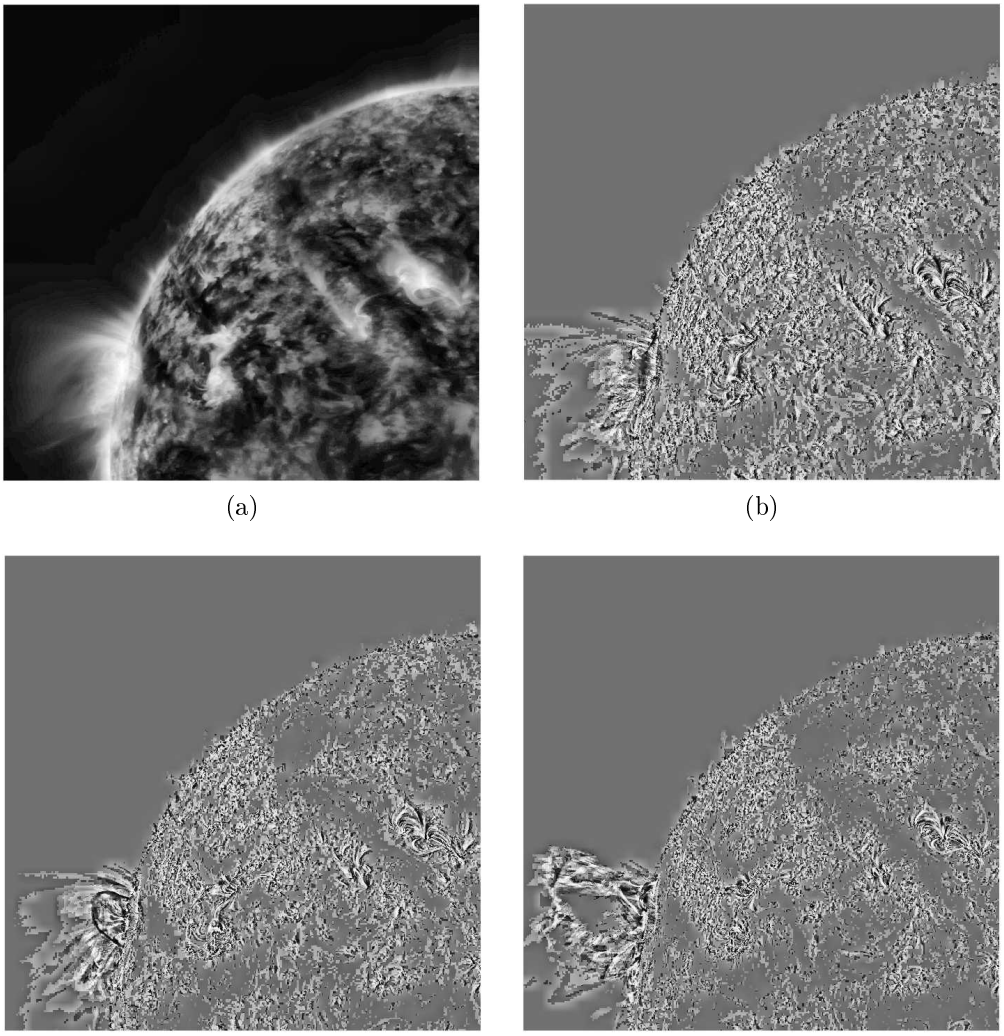


Figure 7.4: Outcome of Median filter method without NAFE enhancement. It shows a) the computed image of the background (matrix $\mathbf{L}$) and the b) 150th, c) 200th, and d) 250th images from the video of the dynamic component (matrix $\mathbf{S}$). The images were taken by SDO-AIA 171 Å on April 16, 2012.

NAFE enhancement does not improve the results. As can be seen in Figure 7.6 the outcome is the same for $w = 0$ and $w = 0.2$. The other weights cause deterioration of the results since the noise is now more contrast and the method evaluates it as a movement.

The computational time of this method is $18 - 20$ seconds.

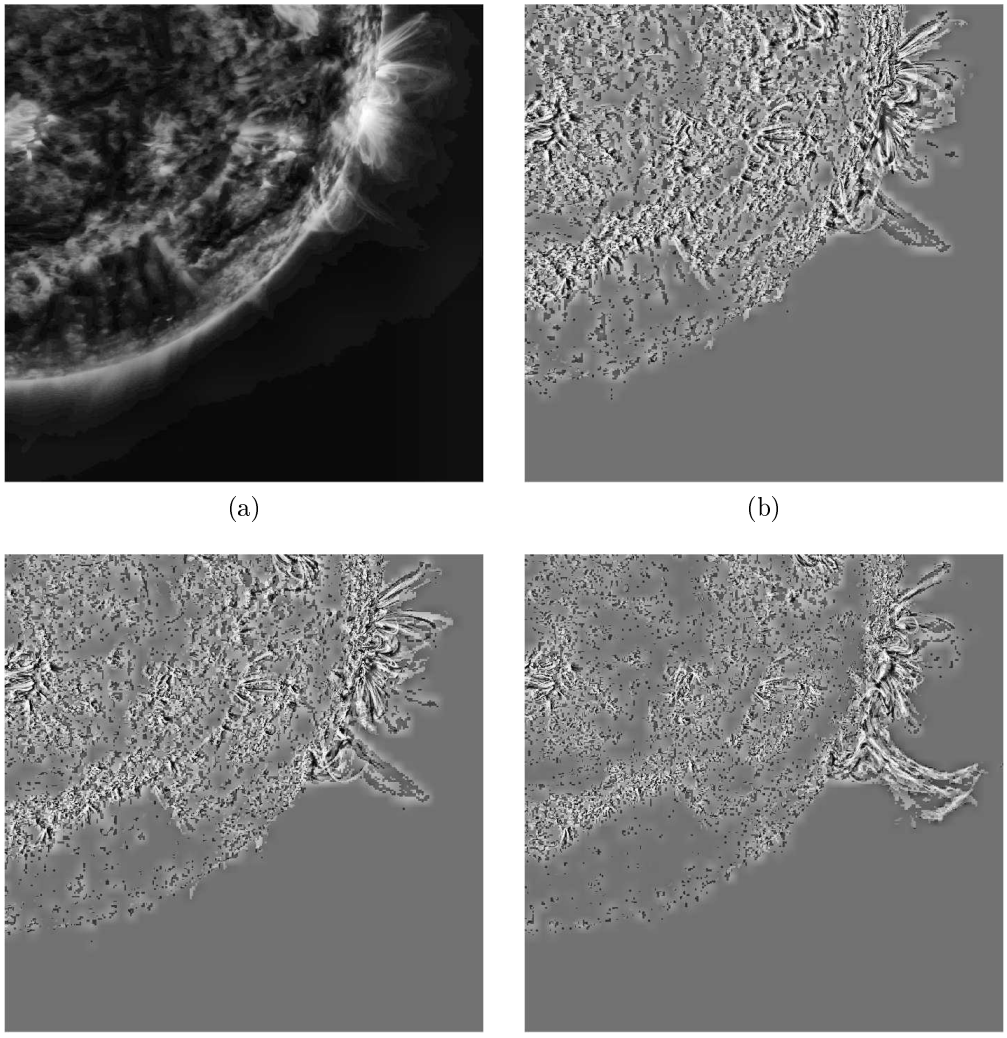
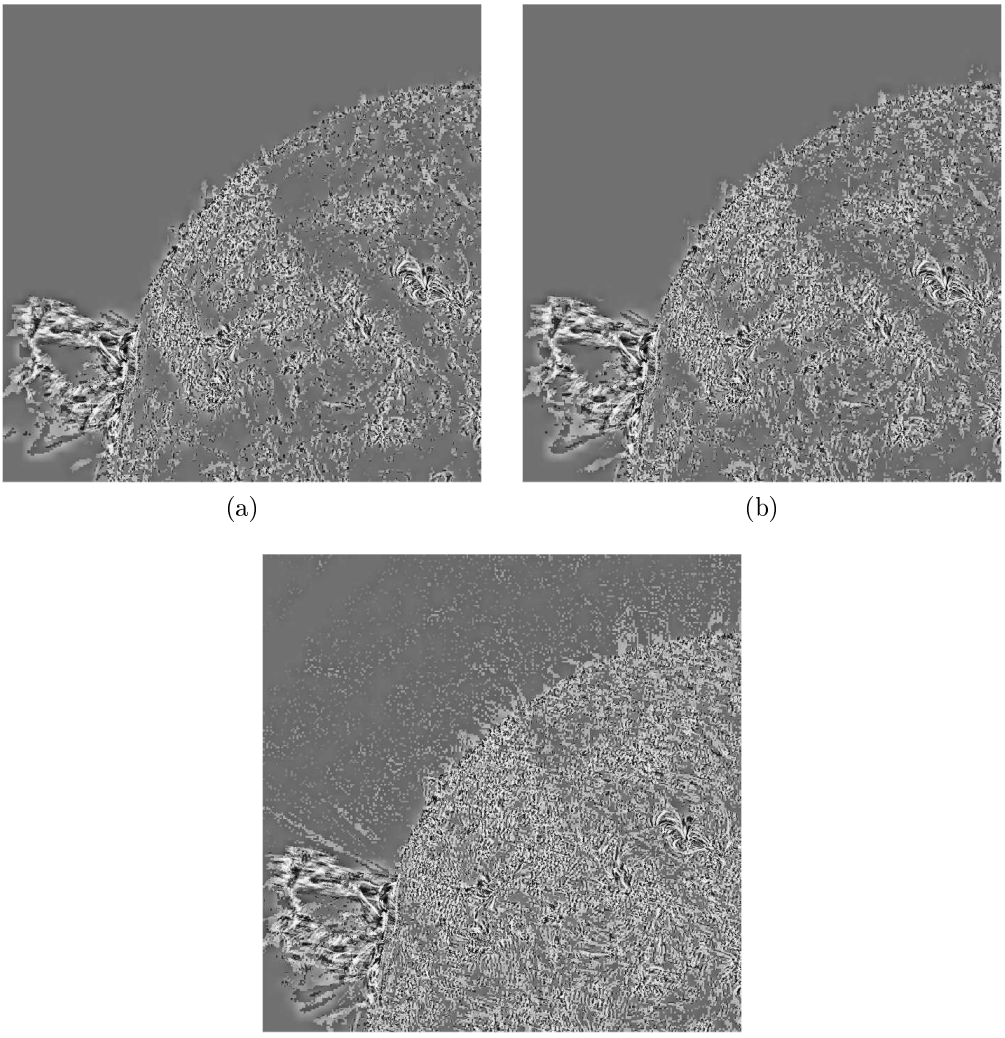


Figure 7.5: Outcome of Median filter method without NAFE enhancement. It shows a) the computed image of the background (matrix **L**) and the b) 47th, c) 127th, and d) 207th images from the video of the dynamic component (matrix **S**). The images were taken by SDO-AIA 171 Å on October 2, 2014.



(a) (b)

(c)

Figure 7.6: Comparison of median filter results with different NAFE enhancement weights. It shows the 250th frame from the video of the dynamic component with the used weights a) $w = 0$, b) $w = 0.2$ and c) $w = 0.4$. The images were taken by SDO-AIA 171 Å on April 16, 2012.

Now, we describe the results of applying the median filter to videos from SDO-AIA 304 Å.

SDO-AIA 304 Å

It can be seen in Figures 7.7, 7.8 and 7.9 that, as before, with the 171 Å frequency this method detects additional motions in the inner corona besides the desired CMEs. The results shown in Figure 7.8 are practically the same as those in Figure 7.4. Unfortunately, the other results are even worse than before.

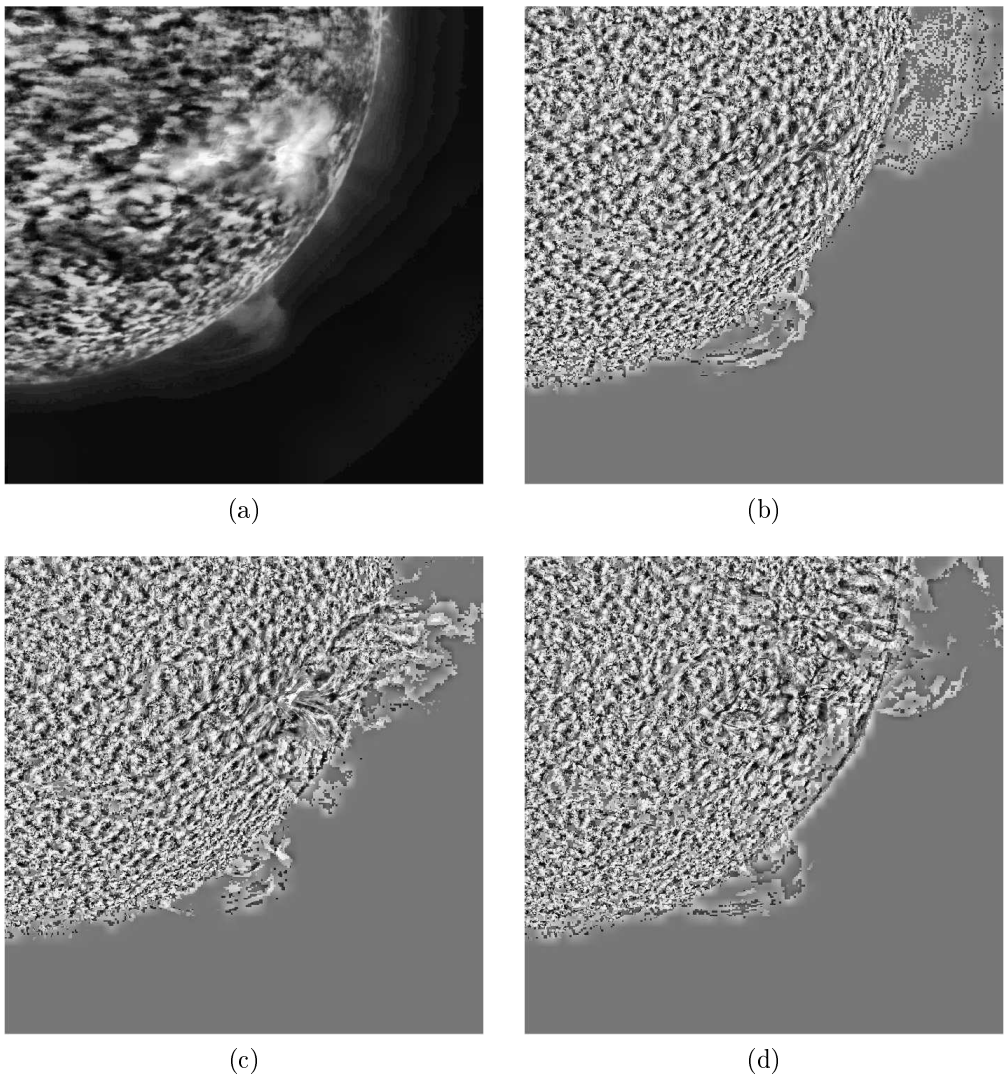
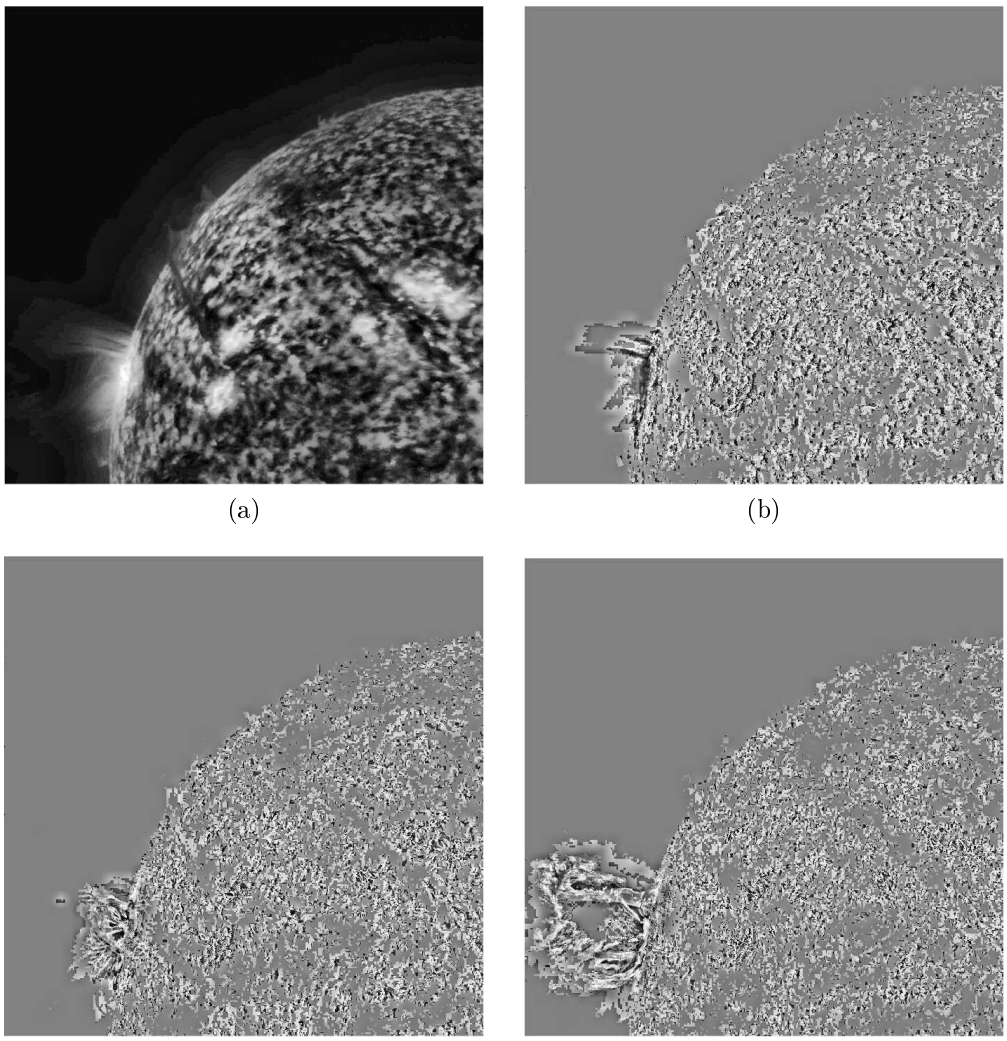


Figure 7.7: Outcome of Median filter. It shows a) the computed image of the background (matrix **L**) and the b) 10th, c) 50th, and d) 90th images from from the video of the dynamic component (matrix **S**). The images were taken by SDO-AIA 304 Å on June 7, 2011.

The computation time of this method is about 18 seconds. The sequence from 2011 contains only 303 images, so the computation time is less than 11 seconds, but the time for one image is comparable.



(a) (b) (c) (d)

Figure 7.8: Outcome of Median filter method. It shows a) the computed image of the background (matrix **L**) and the b) 150th, c) 200th, and d) 250th images from the video of the dynamic component (matrix **S**). The images were taken by SDO-AIA 304 Å on April 16, 2012.

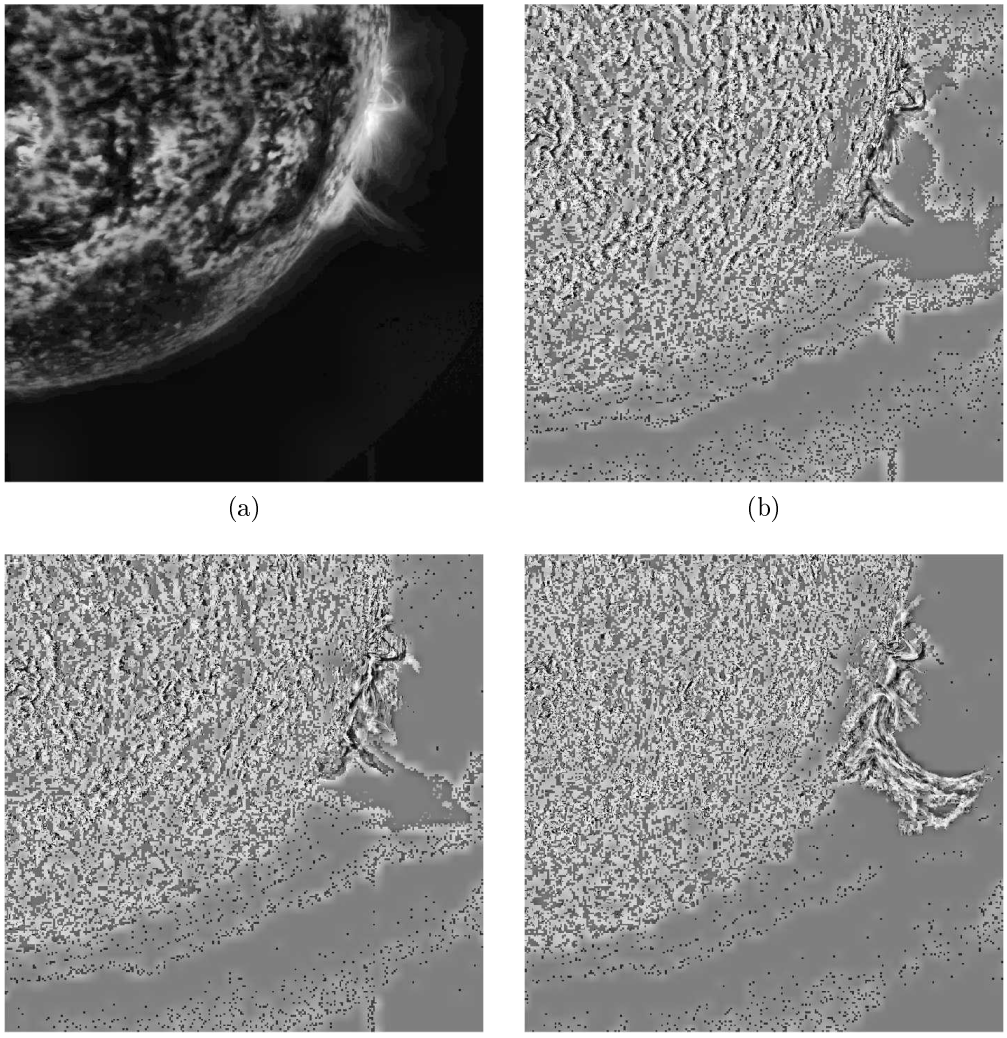


Figure 7.9: Outcome of Median filter method. It shows a) the computed image of the background (matrix **L**) and the b) 47th, c) 127th, and d) 207th images from the video of the dynamic component (matrix **S**). The images were taken by SDO-AIA 304 Å on October 2, 2014.

The routine and results of the PCP method are given below (the theory is described in section 7.4).

7.6.4 PCP

Routine

The first step of the algorithm is a computation of the singular value thresholding. According to formula (7.6), it is needed to compute SVD of the $(\mathbf{M} - \mathbf{S}_k + \mathbf{Y}_k/\mu)$. The full decomposition is time-consuming. The faster method is to compute the reduced SVD. The reduced singular value decomposition is suitable for our purposes as we are solely concerned with non-zero singular numbers and the singular vectors that correspond to these numbers.

Further, since the singular values are always positive, it is feasible to omit the component assigning the number sign in (7.5), i.e.

$$\begin{aligned} \mathbf{M} - \mathbf{S}_k + \mathbf{Y}_k/\mu &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \\ \mathrm{svt}_{\frac{1}{\mu}}(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^*) &= \sum_{i=1}^{n} \max(\sigma_i - \frac{1}{\mu}, 0)\mathbf{u}_i\mathbf{v}_i^* \end{aligned}$$

Since the matrix $\mathbf{S} = \{s_{i,j}\}$ has also negative numbers, the soft thresholding of its values is computed (in Algorithm 1) as follows

$$\begin{aligned} \text{set} \quad \mathbf{T} &= \mathbf{M} - \mathbf{L}^{(k+1)} + \frac{\mathbf{Y}^{(k)}}{\mu}\,, \\ \text{then} \quad s_{i,j}^{(k+1)} &= \frac{t_{i,j}}{|t_{i,j}|}\max(|t_{i,j}| - \frac{\lambda}{\mu}, 0). \end{aligned}$$

The PCP algorithm contains two optional parameters, namely the accuracy parameter δ and step size μ. According to [39], the suitable choices are

$$\begin{aligned} \delta &= 10^{-5} \\ \mu &= \frac{mn}{4\|\mathbf{M}\|_1}, \quad \text{for } \mathbf{M}_{m,n}. \end{aligned}$$

Results

First, we describe the results of applying the PCP method to videos from SDO-AIA 171 Å.

SDO-AIA 171 Å

The outcome of the PCP algorithm seems to be a great improvement. It can be seen in Figures 7.10, 7.11 and 7.12 that before the occurrence of CME, the algorithm looks for the fastest movement so it highlights the noise in the image. During it, the noise almost disappears and the resulting images contain only CME and few fast movements in the inner corona. The results do not depend on the position of the CME in the video, qualitatively we get the same results whether the course of the CME is entirely on the Sun or extends into the solar atmosphere.

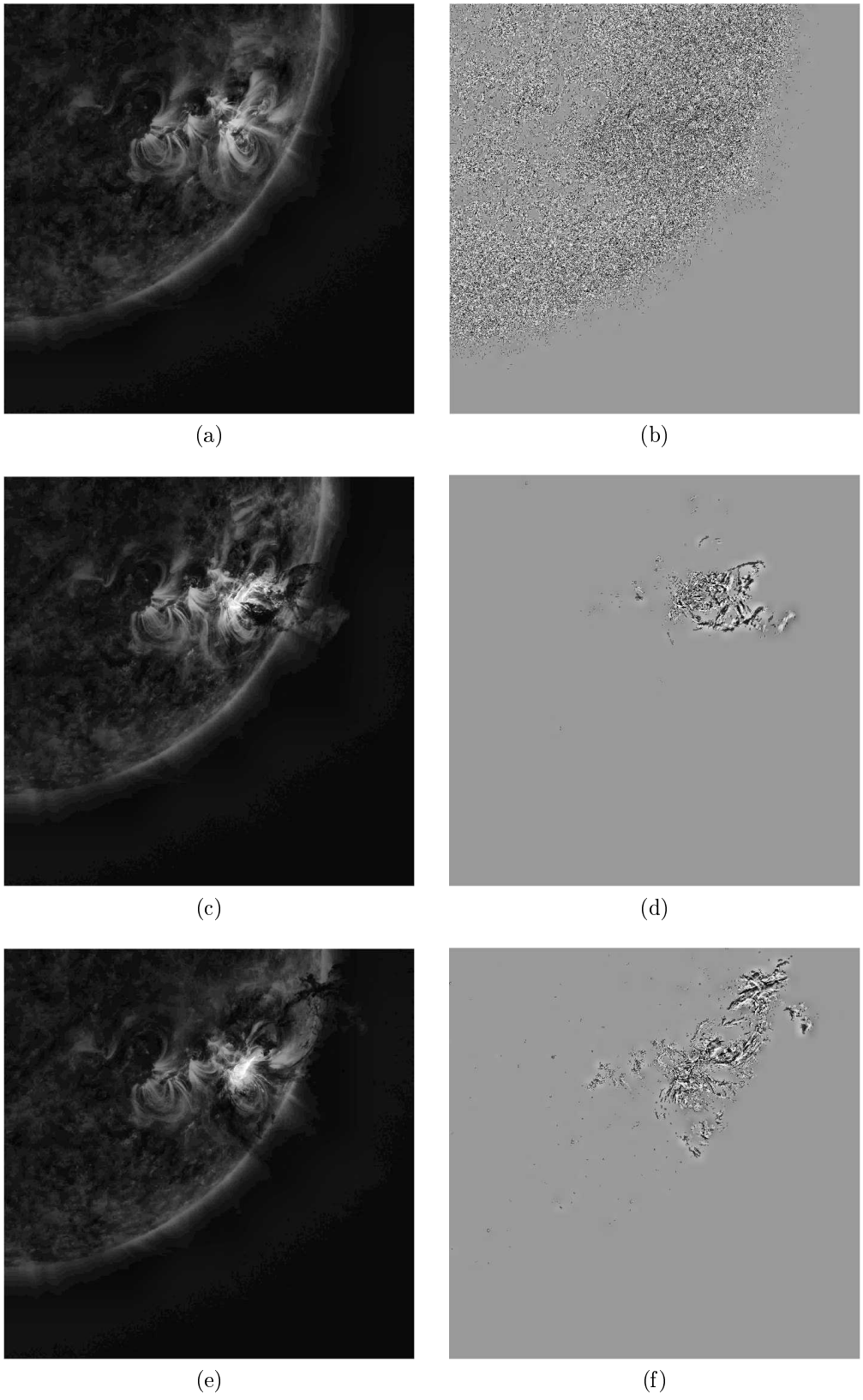


Figure 7.10: Outcome of PCP algorithm with NAFE enhancement ($w = 0.2$). It shows the a) 50th, c) 150th, and e) 250th frames from the matrix **L**. The images b), d) and f) show the same frames of matrix **S**. The images were taken by SDO-AIA 171 Å on June 7, 2011.

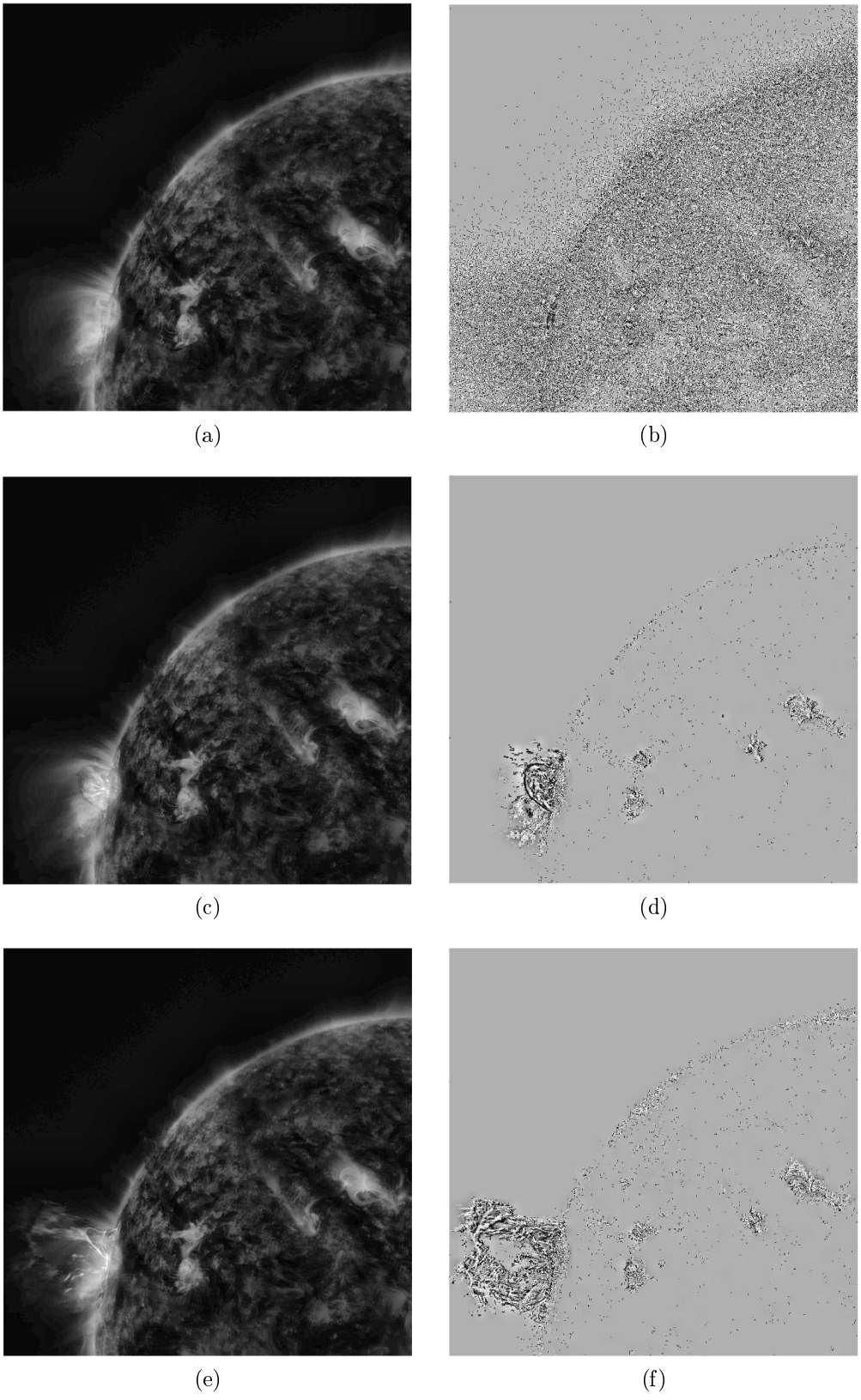


Figure 7.11: Outcome of PCP algorithm with NAFE enhancement ($w = 0.2$). It shows the a) 150th, c) 200th, and e) 250th frames from the matrix $\mathbf{L}$. The images b), d) and f) show the same frames of matrix $\mathbf{S}$. The images were taken by SDO-AIA 171 Å on April 16, 2012.

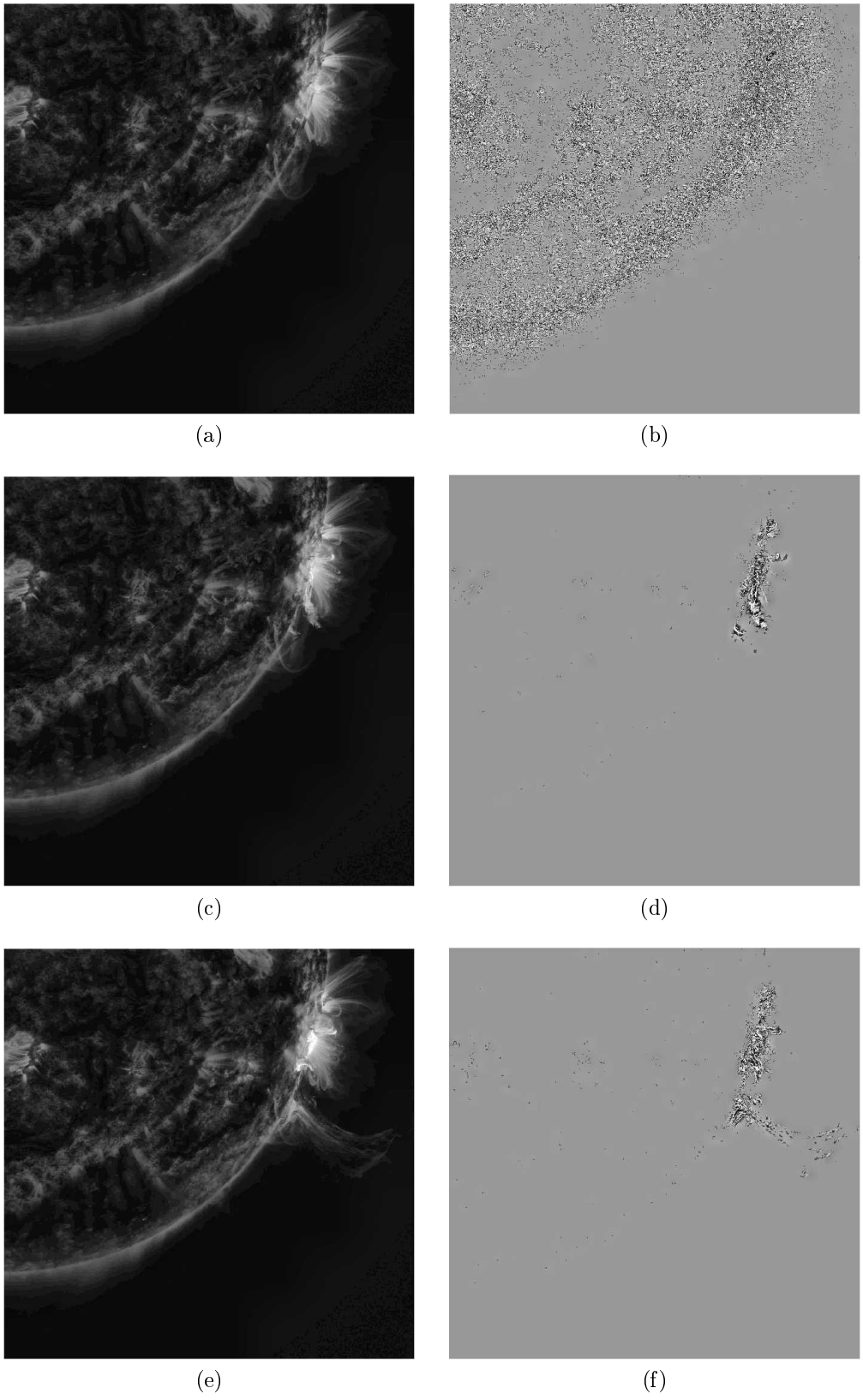


Figure 7.12: Outcome of PCP algorithm with NAFE enhancement ($w = 0.2$). It shows the a) 47th, c) 127th, and e) 207th images from the matrix **L** The images b), d) and f) show the same frames of matrix **S**. The images were taken by SDO-AIA 171 Å on October 2, 2014.

For this method, NAFE enhancement improves the results. As can be seen in Figure 7.13 the best outcome is for $w = 0.2$. The movements in the inner corona are better neglected for $w = 0.4$, but the CME in this case is not correctly separated. The other weights cause deterioration of the results since the noise is now more contrast and the method evaluates it as a movement.

The computational time of this method is heavily dependent on number of iterations. One iteration takes $80 - 100$ seconds. Mostly, the PCP algorithm ended after four iterations, so the computational time is around 350 seconds (approx. 6 minutes). The deviation from this time occurs for the video from April 16, 2012 with $w = 0.2$, its computational time is 10542 seconds (approx. 3 hours) and the algorithm ended after 104 iterations. The reason for this increase in iterations is the complexity of the CME on a given video. The NAFE enhancement increased the noise in the image, thus destroying the complexity of the CME and therefore the time was not as long for the other weights.

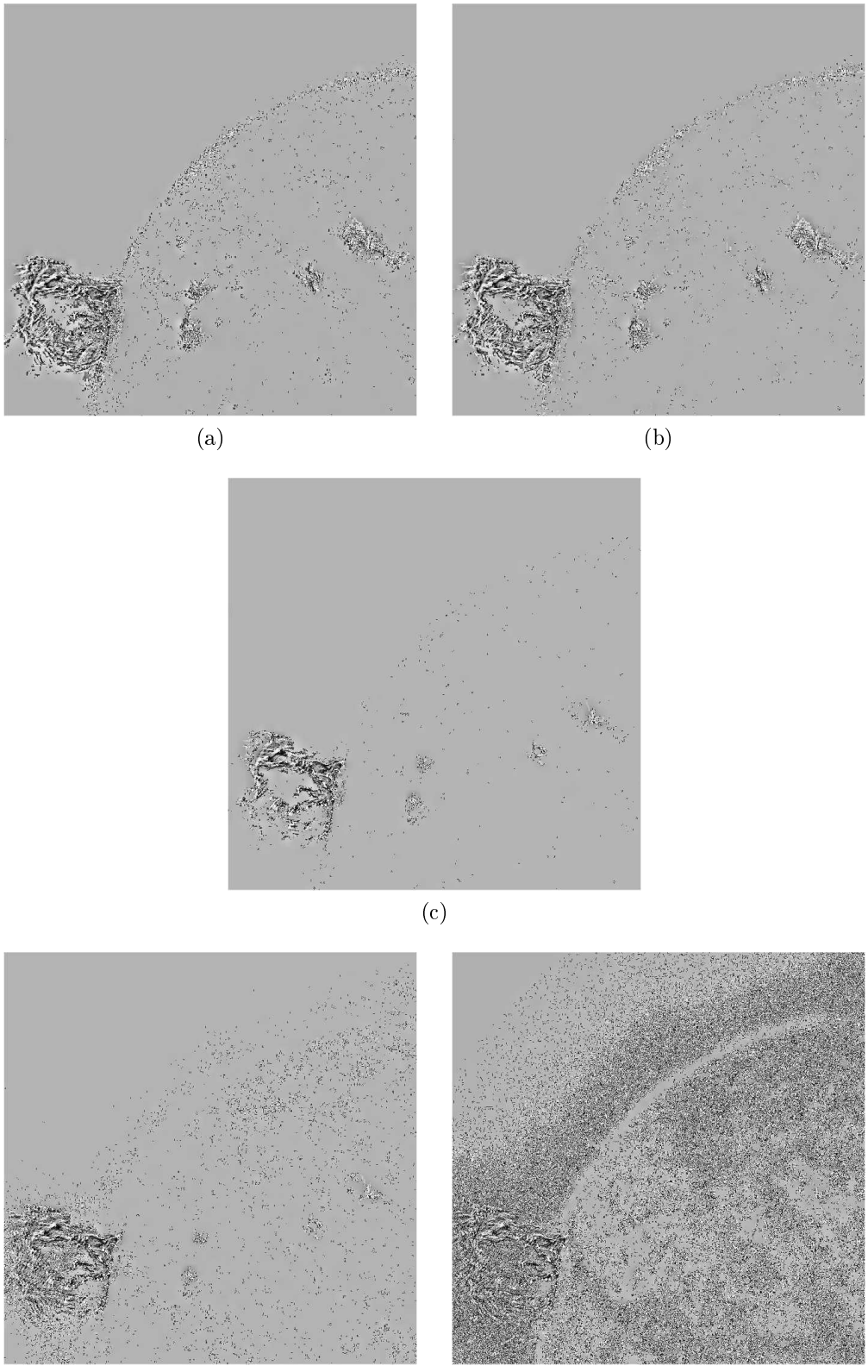


Figure 7.13: Comparison of outcomes of PCP algorithm with different NAFE enhancement weights. It shows the 250th frame from the video of the dynamic component with the used weights a) $w = 0$, b) $w = 0.2$, c) $w = 0.4$, d) $w = 0.6$ and e) $w = 0.8$. The images were taken by SDO-AIA 171 Å on April 16, 2012.

Now, we describe the results of applying the PCP method to videos from SDO-AIA 304 Å.

SDO-AIA 304 Å

It can be seen in Figures 7.14, 7.15 and 7.16 that, as before, with the 171 Å frequency the algorithm highlights the noise in the image prior to occurence of the CME. Afterwards the resulting images contain only CME and few fast movements in the inner corona. The outcomes of both frequencies are therefore comparable. The white line visible in Figure 7.16 is caused by the fact that the chip on the SDO - AIA 304 Å does not have built-in antiblooming. The method has therefore only highlighted the defect caused by the high saturation.

One iteration takes about 100 seconds to complete. The algorithm finished after 166 and 209 iterations for the 2012 and 2014 videos, respectively. The computation time is thus between 4 and 6 hours. Since the 2011 video contains only 303 frames, one iteration takes only 52 seconds, the whole algorithm finished after 5 iterations, so the computation time is $4 - 5$ minutes.

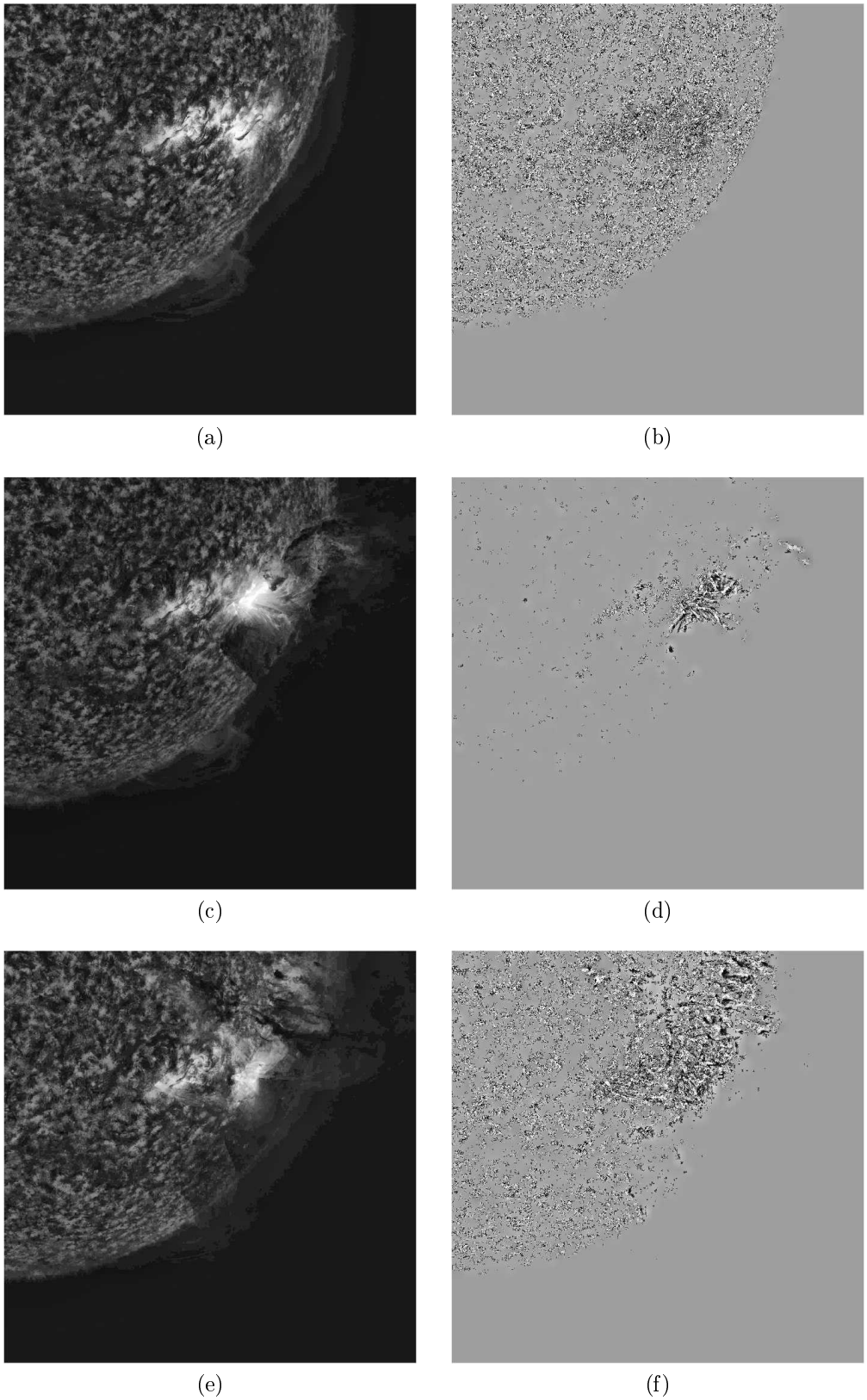


Figure 7.14: Outcome of PCP algorithm. It shows the a) 10th, c) 50th, and e) 90th frames from the matrix **L**. The images b), d) and f) show the same frames of matrix **S**. The images were taken by SDO-AIA 304 Å on June 7, 2011.

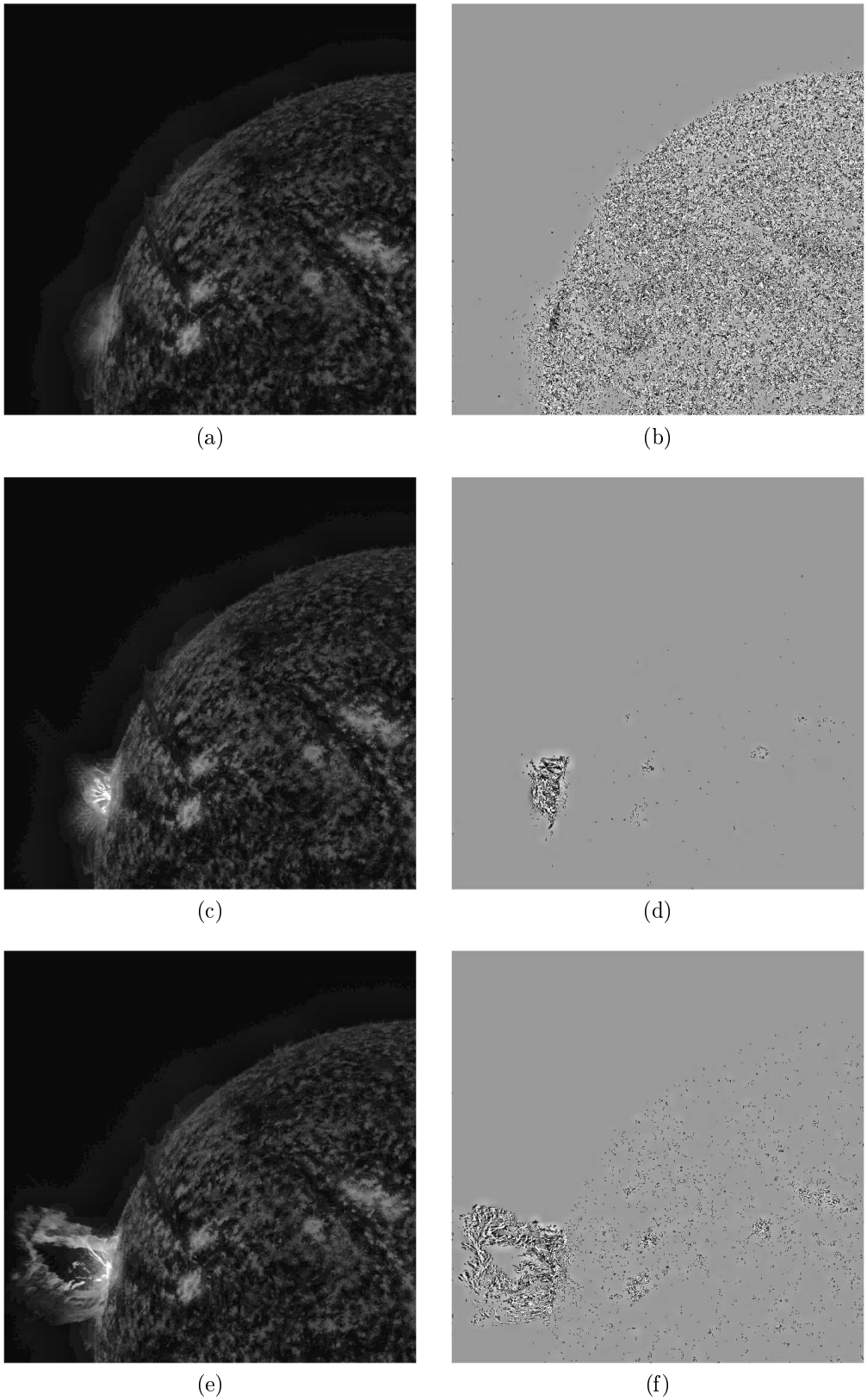


Figure 7.15: Outcome of PCP algorithm. It shows the a) 150th, c) 200th, and e) 250th frames from the matrix **L**. The images b), d) and f) show the same frames of matrix **S**. The images were taken by SDO-AIA 304 Å on April 16, 2012.

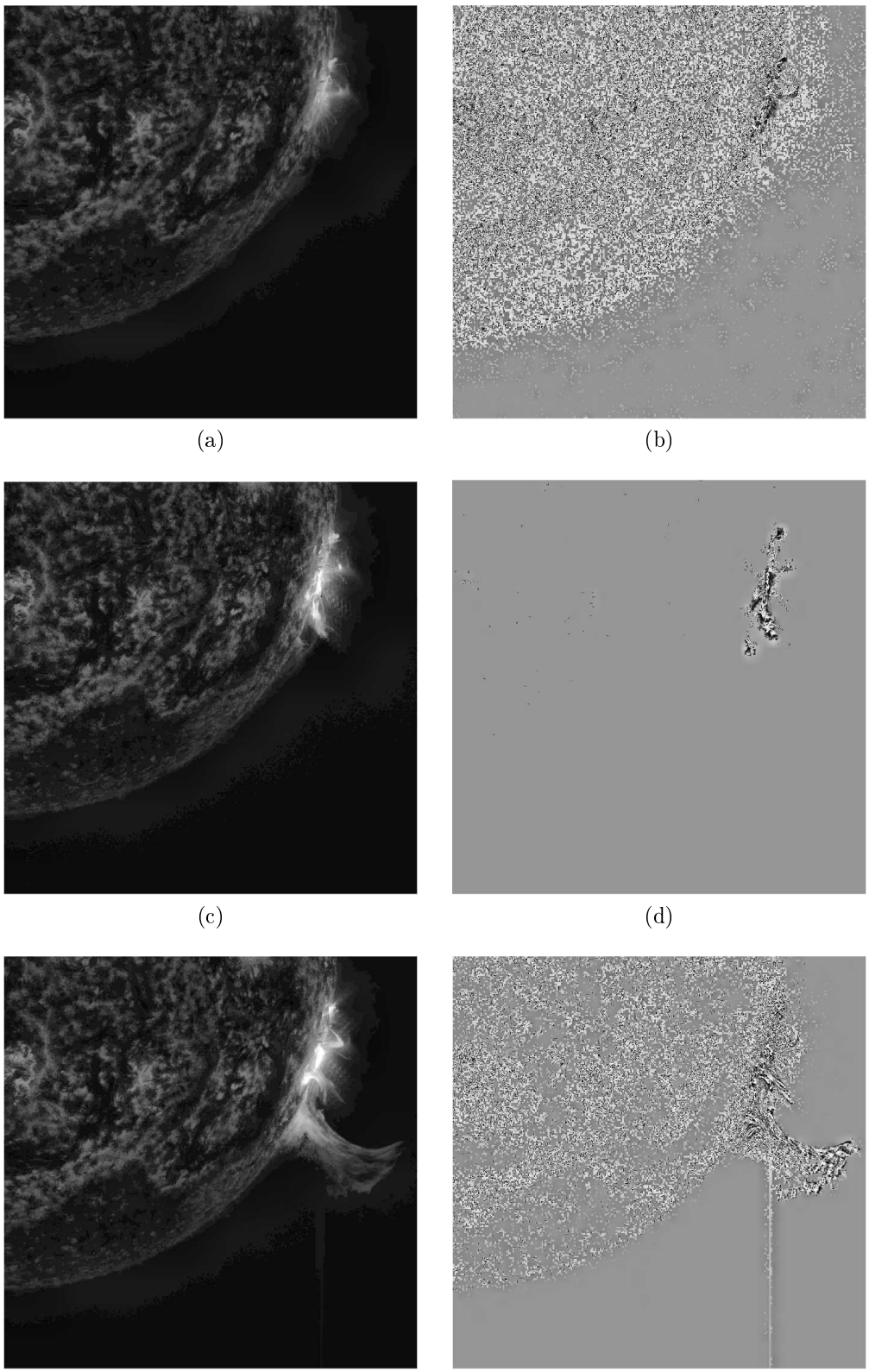


Figure 7.16: Outcome of PCP algorithm. It shows the a) 47th, c) 127th, and e) 207th images from the matrix **L**. The images b), d) and f) show the same frames of matrix **S**. The images were taken by SDO-AIA 304 Å on October 2, 2014.

The routine and results of the DMD method are given below (the theory is described in section 7.5).

7.6.5 DMD

Routine

As with the PCP method, it is sufficient to compute the reduced SVD decomposition. The DMD method uses two optional parameters. The input parameter ε determines the threshold at which a given Fourier mode is still considered to be near the origin in complex space. Based on [50], the value of ε is chosen to be

$$\varepsilon = 10^{-2} .$$

The next parameter to set is r, which determines the dimension reduction. It should be chosen to capture all dominant modes. For videos with a stationary background, the rank of the background matrix $\mathbf{L}$ should be 1, so the ideal choice is $r = 1$. The larger r is chosen, the more error is introduced into the dynamic component. For real video, the rank of the matrix $\mathbf{L}$ is unknown, so the number of dominant modes needed to capture the behaviour of a given system is unclear. Therefore, for solar observation videos, the maximum possible r is always chosen, i.e.

$$r = m - 1,$$

where m is the number of frames in a given video.

Results

First, we describe the results of applying the DMD method to videos from SDO-AIA 171 Å.

SDO-AIA 171 Å

The outcome of the DMD algorithm seems similar to the result of the median filter method. As can be seen in Figures 7.17, 7.18 and 7.19 the DMD algorithm captures also the movements in the inner corona. It can also be seen that the background video changes very little, in contrast to the results of the PCP method. The outcomes of this method is the same wherever the CME is located.

Just like for the Median filter method, NAFE enhancement does not improve the results. As can be seen in Figure 7.20 the best outcome is from unprocessed images. NAFE processing causes deterioration of the results because of emphasized noise.

The computational time of this method is $200 - 300$ seconds (approx. $3 - 5$ minutes).

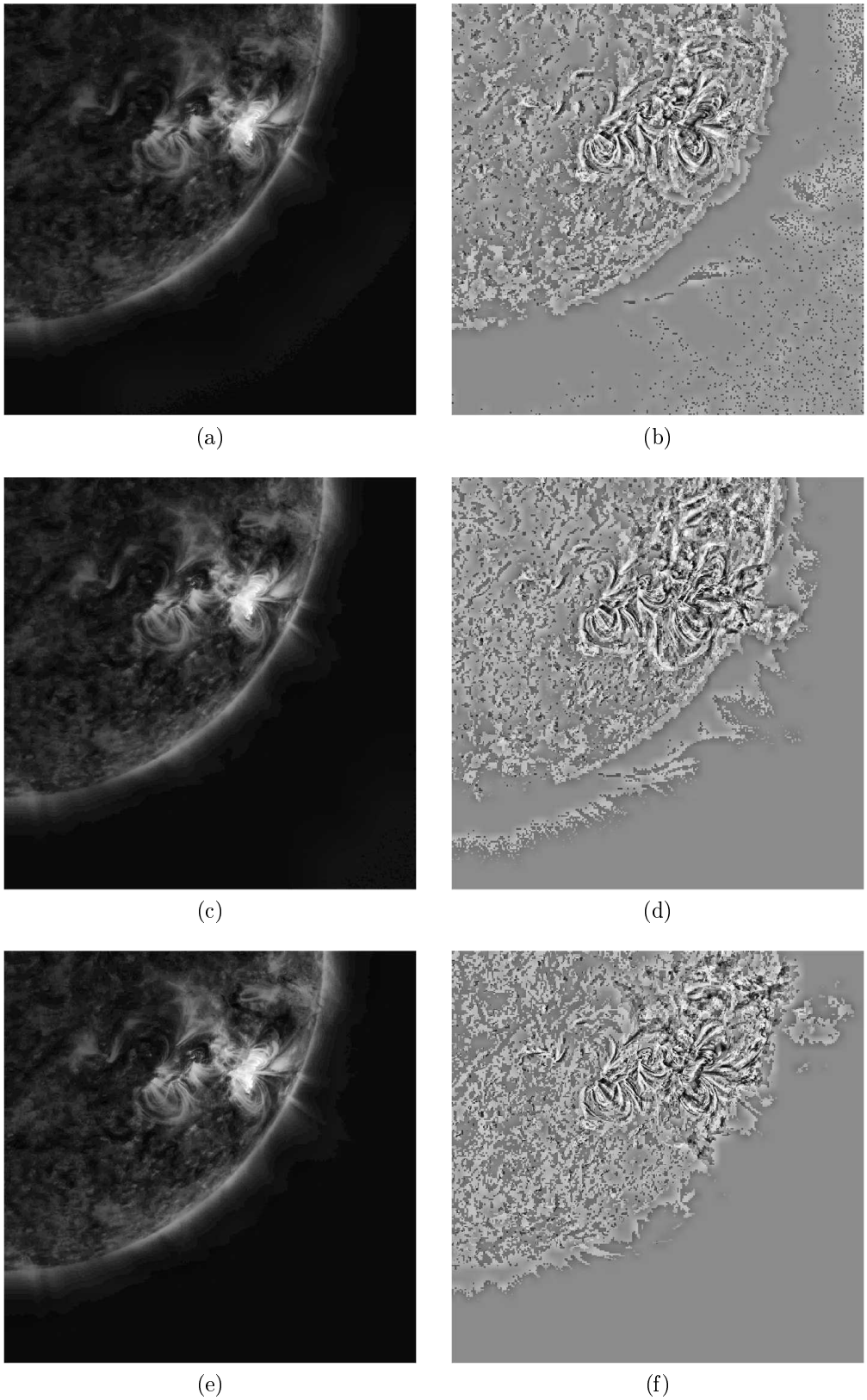


Figure 7.17: Outcome of DMD algorithm without NAFE enhancement. It shows the a) 50th, c) 150th, and e) 250th frames from the matrix **L**. The images b), d) and f) show the same frames of matrix **S**. The images were taken by SDO-AIA 171 Å on June 7, 2011.

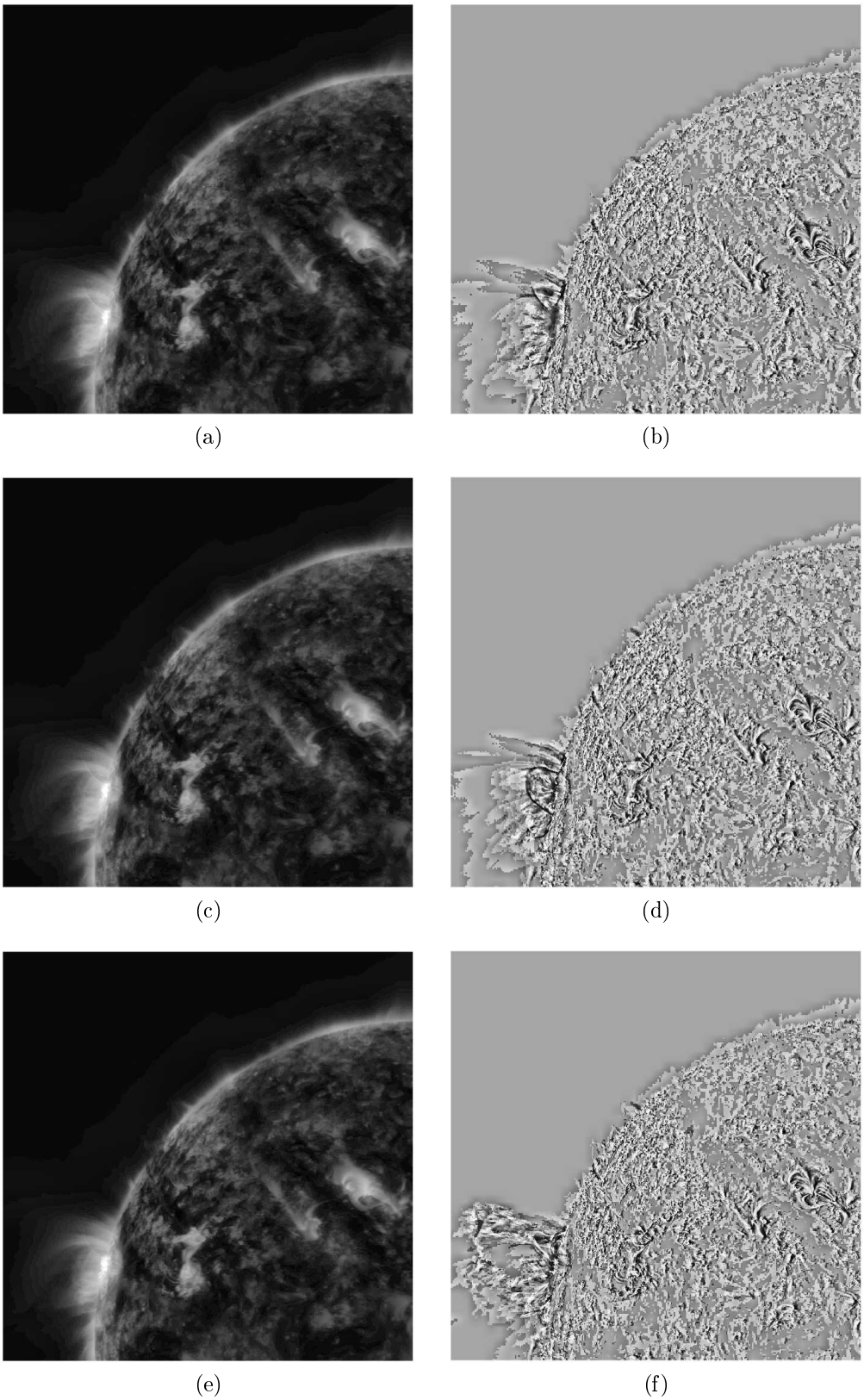


Figure 7.18: Outcome of DMD algorithm without NAFE enhancement. It shows the a) 150th, c) 200th, and e) 250th frames from the matrix **L**. The images b), d) and f) show the same frames of matrix **S**. The images were taken by SDO-AIA 171 Å on April 16, 2012.

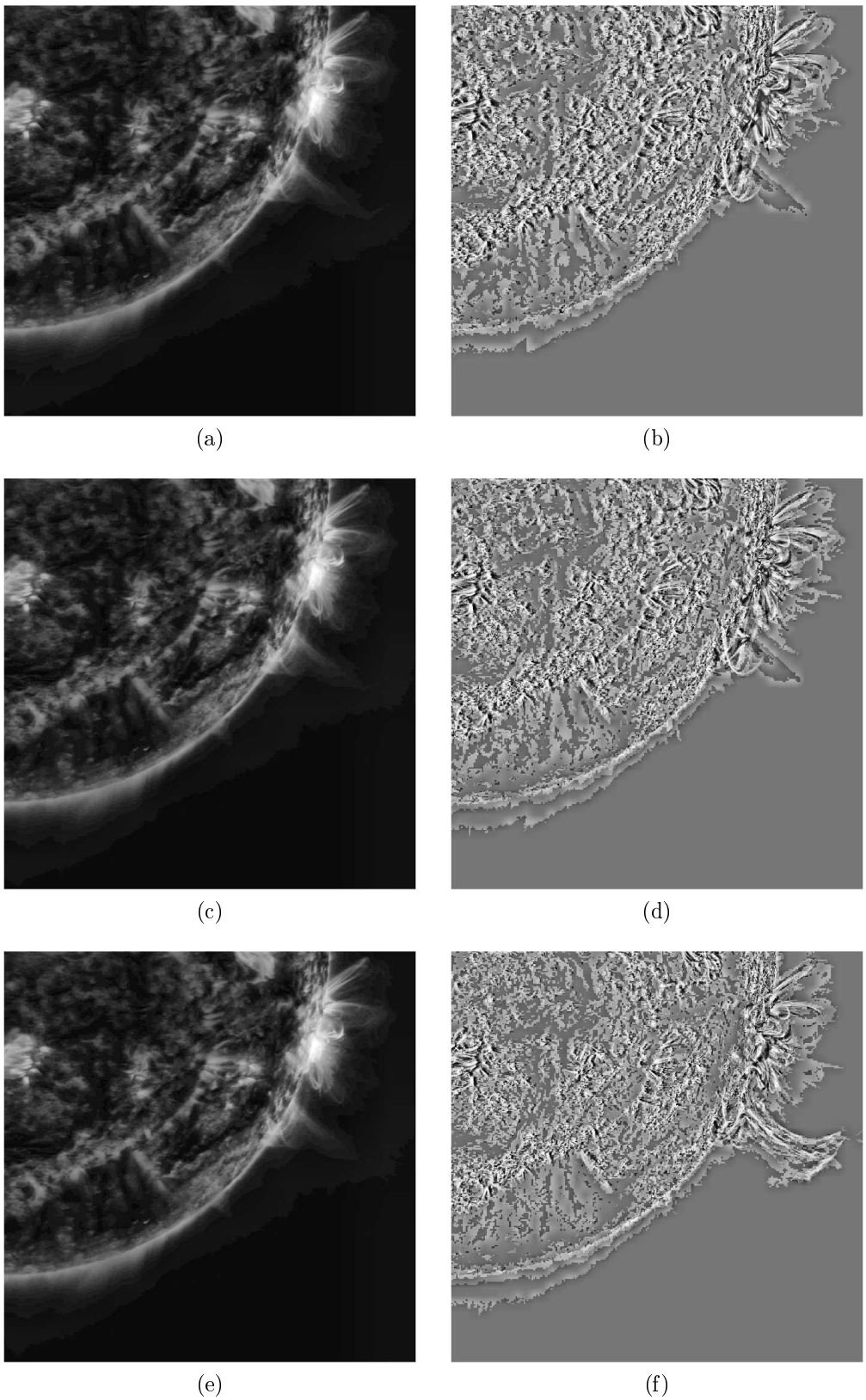


Figure 7.19: Outcome of DMD algorithm without NAFE enhancement. It shows the a) 47th, c) 127th, and e) 207th images from the matrix **L**. The images b), d) and f) show the same frames of matrix **S**. The images were taken by SDO-AIA 171 Å on October 2, 2014.

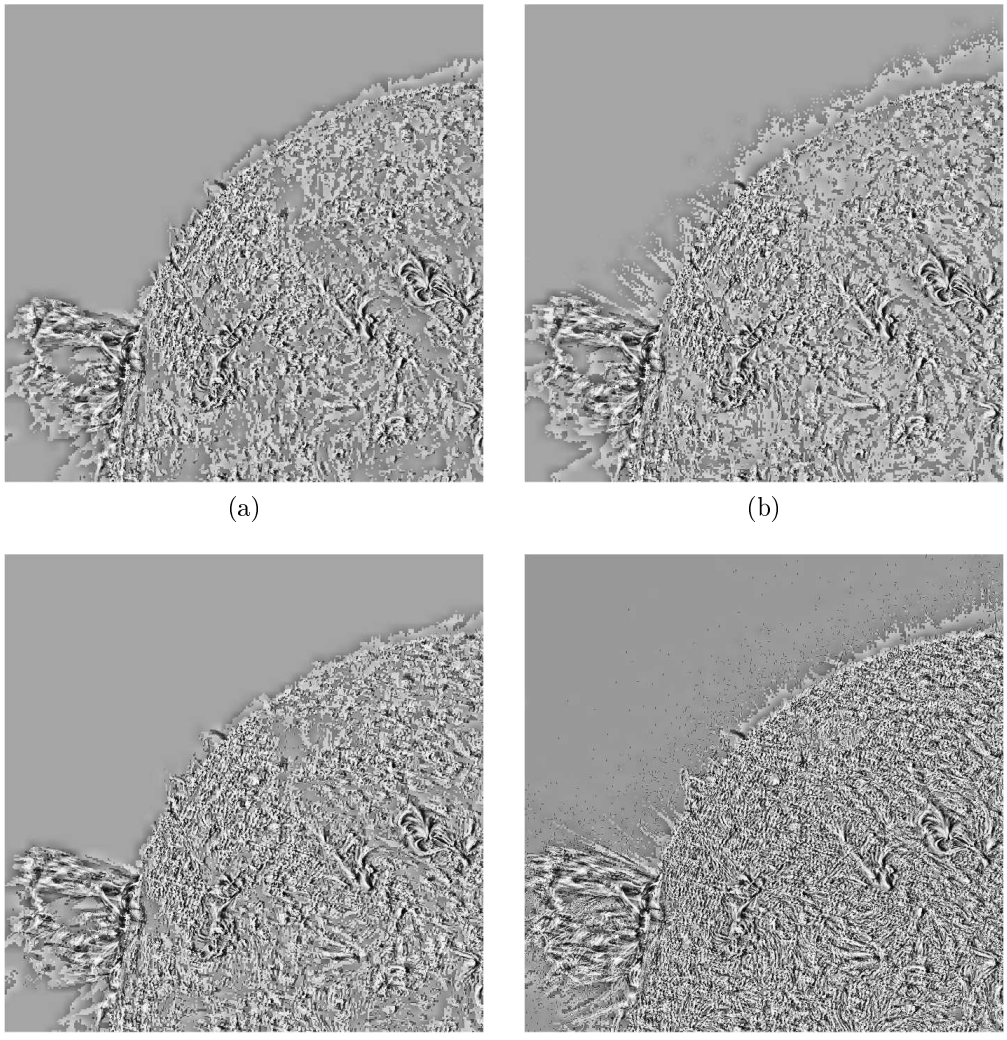


Figure 7.20: Comparison of outcomes of DMD algorithm with different NAFE enhancement weights. It shows the 250th frame from the video of the dynamic component with the used weights a) $w = 0$, b) $w = 0.2$, c) $w = 0.4$ and d) $w = 0.6$. The images were taken by SDO-AIA 171 Å on April 16, 2012.

Now, we describe the results of applying the DMD method to videos from SDO-AIA 304 Å.

SDO-AIA 304 Å

It can be seen in Figures 7.21,7.22 and 7.23 that, as before, with the 171 Å frequency this method detects additional motions in the inner corona besides the desired CMEs. Furthermore, it is evident that the method has isolated CME, but it has not captured the entire phenomenon. Therefore, this method is unsuitable for this frequency.

The computational time of this method is around 210 seconds (approx. 4 minutes). Since the video from 2011 contains only 303 frames, the computation time is 122 seconds (approx. 2 minutes). However, the time per frame is the same.

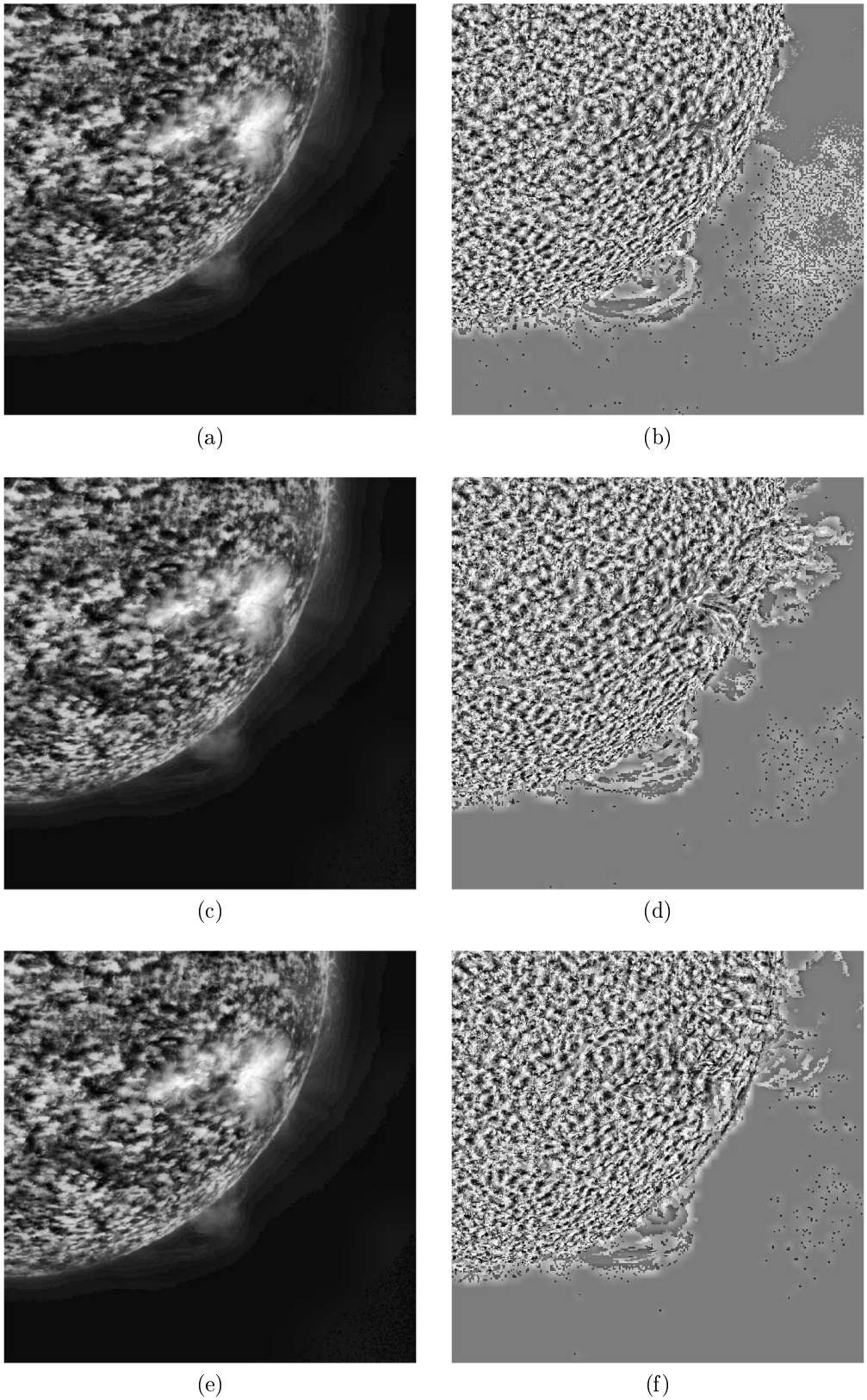


Figure 7.21: Outcome of DMD algorithm. It shows the a) 10th, c) 50th, and e) 90th images from the matrix **L**. The images b), d) and f) show the same frames of matrix **S**. The images were taken by SDO-AIA 304 Å on June 7, 2011.

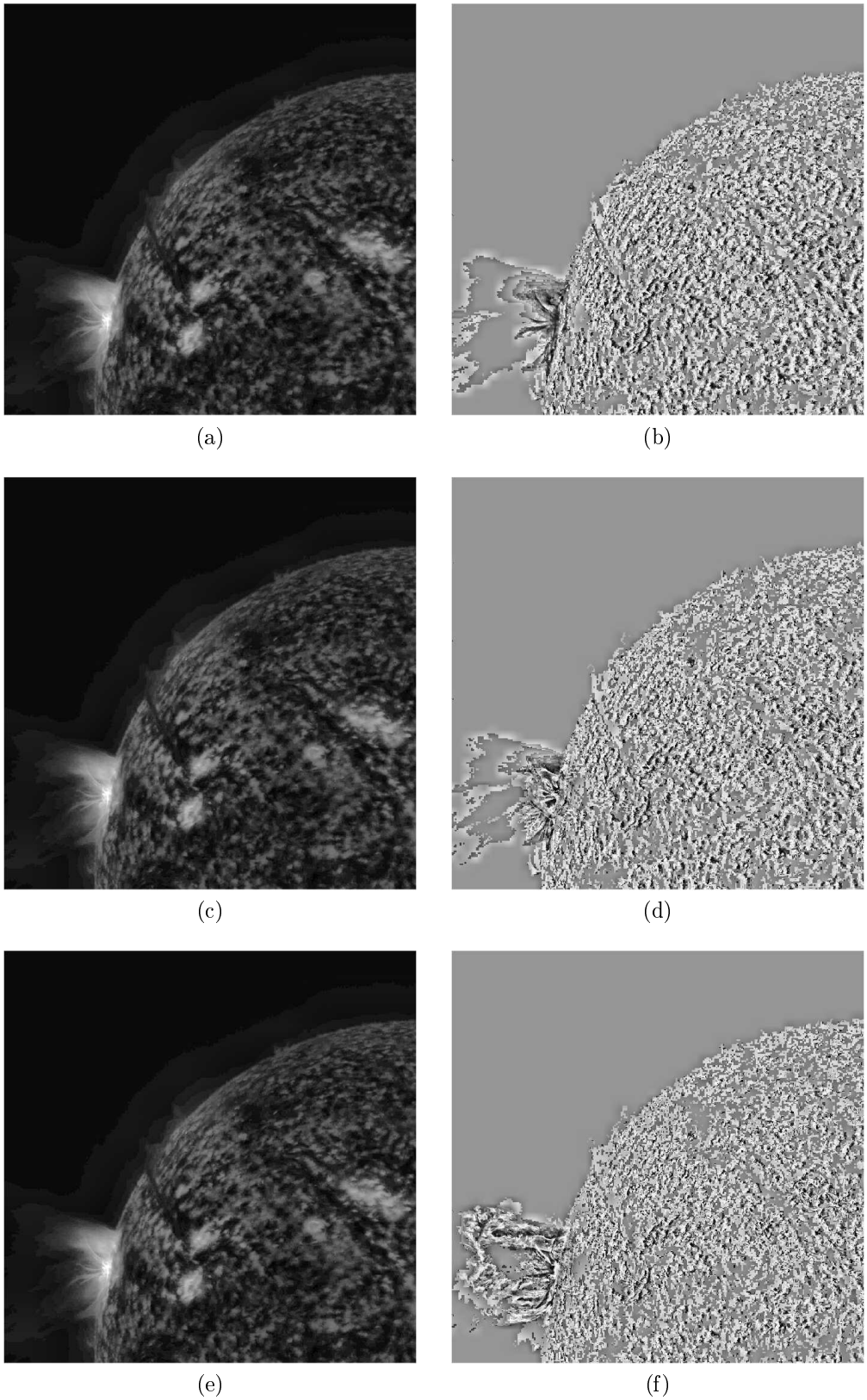


Figure 7.22: Outcome of DMD algorithm. It shows the a) 150th, c) 200th, and e) 250th images from the matrix **L**. The images b), d) and f) show the same frames of matrix **S**. The images were taken by SDO-AIA 304 Å on April 16, 2012.

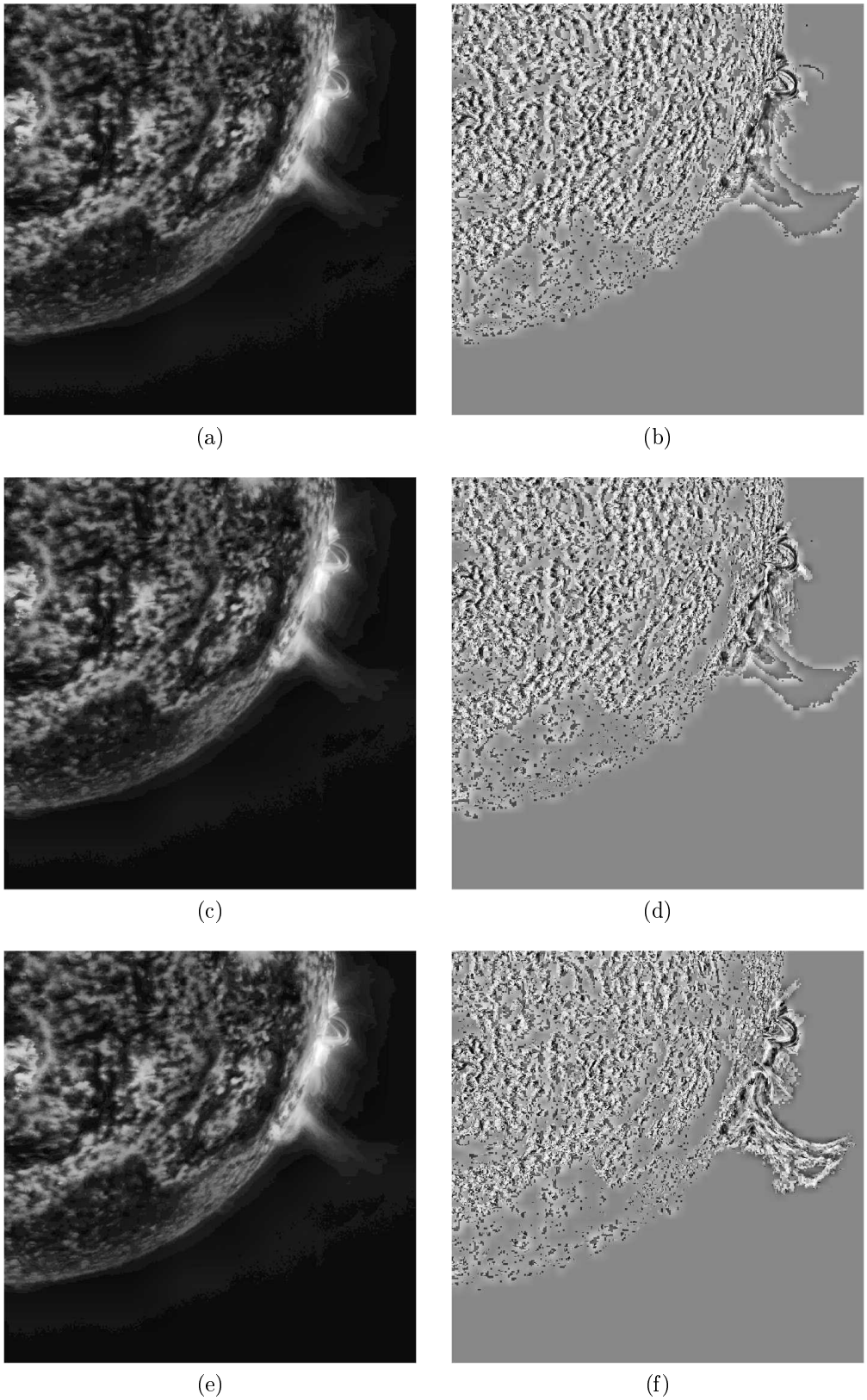


Figure 7.23: Outcome of DMD algorithm. It shows the a) 47th, c) 127th, and e) 207th images from the matrix **L**. The images b), d) and f) show the same frames of matrix **S**. The images were taken by SDO-AIA 304 Å on October 2, 2014.

Conclusion

This doctoral thesis provides an overview of the astronomical, mathematical and image registration background required for the decomposition of the dynamical and nearly stable part of the solar corona from videos. Furthermore, it demonstrates the implementation of the Moment method for the filtration of impulse noise. The concluding section of this thesis is dedicated to the methods based on sparse representations, namely the median filter method, principal component pursuit and dynamic mode decomposition.

The issue with the fundamental methodology for the filtration of impulse noise is that it also filters commonly occurring phenomena in the solar atmosphere, such as solar rays and CMEs. The filtration improved by Moment method addresses this issue and is now a significantly more accurate approach than the basic procedure.

The median filter is the fastest proposed method. Unfortunately, it does not perform well for videos with different movement dynamics. It successfully decomposes only videos with static backgrounds. In other words, this method did not lead to an improvement.

The DMD algorithm gives similar results as the median filter method, the CME is decomposed, but the slower movements in the inner corona remain in the decomposition. Same as before, this method did not bring any improvement either.

The PCP algorithm decomposes the video almost excellently. It separates the fastest dynamic image component. Unfortunately, the resulting video includes motion in the inner corona, which has the same velocity as the CME, because the velocity of the CME is not always unique at the time the image is taken. This technique resolves the issue of the running difference method primarily during the fastest CME movement. The method yields superior outcomes compared to the state-of-the-art method. The quality of this algorithm is redeemed by a high computational time.

The doctoral thesis fulfills the goals set in the treatise:

- The moment method has been demonstrated to be an effective approach for filtering impulse noise. As illustrated in Figure 5.10, this method is independent of the observer. Additionally, as shown in Figure 5.11, it minimises type II error, preserving the CME structure.

- It has been shown that methods based on sparse representations can be used to decompose solar observation videos. However, only the PCP method is an improvement on the method currently used. This method applied to the NAFE preprocessed data enables the clear differentiation of CMEs, from other phenomena within the examined video, see Figures 7.10-7.16.

Further investigation into both noise filtering and CME separation is worthy of consideration. Potential ways for further research include:

- Test whether the values derived from the decision tree for impulse noise filtering (see Figure 5.10) are applicable to probes other than SOHO.
- Determine whether additional parameter variation in NAFE will result in enhanced outcomes for the PCP method.
- Test whether the PCP method is also applicable to other probes, including TRACE, STEREO, Parker Solar Probe and Solar Orbiter.
- Employ machine learning techniques for the separation of CME from solar observation videos and to assess the accuracy of the results by comparing them with the outcomes of the PCP algorithm.

Bibliography

[1] NORTON, Andrew. AN INTRODUCTION TO ASTROPHYSICS AND COSMOLOGY. The Open University, 2002. ISBN-10: 0749297638, ISBN-13: 978-0749297633. Available from: http://www.heptapolis.com/wp-content/uploads/pdf/A.%20NORTON%20Introd.to%20Astrol%20_Cosmology%20.pdf

[2] Kosová, P.: Numerical methods of space-based coronagraph image processing, Brno: Brno University of Technology, Faculty of Mechanical Engineering, 2019. 71 p. Diploma thesis supervisor prof. RNDr. Miloslav Druckmüller, CSc. Available from: https://www.vutbr.cz/studenti/zav-prace/detail/115505

[3] DRUCKMÜLLEROVÁ, Hana. Application of adaptive filters in processing of solar corona images. Brno, 2014. Doctoral thesis. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Supervisor: Doc.PaedDr. Dalibor Martišek, Ph.D.

[4] SHOPOV, Y.Y., D.A. STOYKOVA, K. STOITCHKOVA, et al. Structure of the solar dust corona and its interaction with the other coronal components. Journal of Atmospheric and Solar-Terrestrial Physics [online]. Elsevier, 2008, 70(2), 356-364, DOI: 10.1016/j.jastp.2007.08.058. ISSN 1364-6826. Available from: https://arxiv.org/ftp/arxiv/papers/0909/0909.1722.pdf

[5] The Origin of the Corona's Light. Total eclipse 2017 - Science [online]. National Aeronautics and Space Administration. Available from: https://eclipse2017.nasa.gov/origin-corona's-light

[6] ACKSON, Myles W. Fraunhofer and his spectral lines. Annalen der Physik [online]. 2014, 526(7-8), A65-A69. DOI: 10.1002/andp.201400807. ISSN 0003-3804. Available from: https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.201400807

[7] Coronal mass ejection. Space weather prediction center: National oceanic and atmospheric administration [online]. Broadway. Available from: https://www.swpc.noaa.gov/phenomena/coronal-mass-ejections

[8] MOLDWIN, Mark. Coronal mass ejection. In: Encyclopædia Britannica [online]. 2010. Available from: https://www.britannica.com/science/coronal-mass-ejection

[9] KOUTCHMY, S., G. STELLMACHER, O. KOUTCHMY, N. I. DZUBENKO, V. I. IVANCHUK, O. S. POPOV, G. A. RUBO, S. K. VSEKHSVIATSKII, and P. LAMY. Photometrical analysis of the June 30, 1973 solar corona. Astronomy and Astrophysics. 1978, Vol. 69, pp. 35-42. Available from: http://articles.adsabs.harvard.edu/cgi-bin/nph-iarticle_query?1978A%26A....69...35K&amp

[10] ZAQARASHVILI, T. V., V. N. MELNIK, A. I. BRAZHENKO, M. PANCHENKO, A. A. KONOVALENKO, A. V. FRANZUZENKO, V. V. DOROVSKYY a H. O. RUCKER. Radio seismology of the outer solar corona [online]. 2013, 555. DOI: 10.1051/0004-6361/201321548. ISSN 0004-6361. Available from: https://www.aanda.org/articles/aa/pdf/2013/07/aa21548-13.pdf

[11] OWAKI, Naoaki and Kuniji SAITO. Photometry of the Solar Corona at the 1962 February Eclipse. Publications of the Astronomical Society of Japan. 1967, Vol. 19, pp. 279-289. Available from: http://adsabs.harvard.edu/abs/1967PASJ...19..279O

[12] The FITS Support Office at NASA/GSFC [online]. Responsible NASA representative: Dr. Thomas A. McGlynn: High Energy Astrophysics Science Archive Research Center, 2017 [cit. 2018-05-26]. Available from: https://fits.gsfc.nasa.gov/fits_home.html

[13] HOGG, Robert V, J. MCKEAN, and A. T. CRAIG, Introduction to Mathematical Statistics: Seventh Edition, Boston, MA: Pearson, 2013, ISBN 978-0-321-79543-4, x +694 pp.

[14] MONTGOMERY, Douglas C, Douglas MONTGOMERY a George C RUNGER. Applied statistics and probability for engineers. 3th ed. New York: Wiley, 2003, xiv +706 pp. : il. 1 CD-ROM. ISBN 0-471-38181-0.

[15] JANIGA, Ivan a Ivan GARAJ. On exact two-sided statistical tolerance intervals for normal distributions with unknown means and unknown common variability. Slovak University of Technology in Bratislava, 2009.Available from: http://asa-qprc.org/2009/papers/QPRC_Contributed_Session_4/Janiga_Paper_konecny.pdf

[16] ALLAIRE, Grégoire, Sidi Mahmoud KABER a Karim TRABELSI. Numerical Linear Algebra. New York: Springer Science+Business Media, 2008. ISBN 978-0-387-34159-0. DOI: 10.1007/978-0-387-68918-0. Available from: http://mitran-lab.amath.unc.edu/courses/MATH662/biblio/AllaireKaber_2008_Book_NumericalLinearAlgebra.pdf

[17] RAJMIC, Pavel. Řídké a nízkohodnostní reprezentace signálů s aplikacemi. Brno, 2014, 154 s. Habillitační práce. Vysoké učení technické v Brně.

[18] TREFETHEN, Lloyd N. a David BAU. Numerical linear algebra. Philadelphia: Society for Industrial and Applied Mathematics, 1997. ISBN 978-0-898713-61-9.

[19] DRUCKMÜLLEROVÁ, Hana. Phase-correlation based image registration. Brno, 2010. Diploma thesis, Fakulta strojního inženýrství Vysoké učení technické v Brně. Supervisor: Doc.PaedDr. Dalibor Martišek, Ph.D.

[20] PRATT, William K. Digital Image Processing, WILEY 2007, xix +735pp, ISBN-13: 978-0471767770, ISBN-10: 0471767778

[21] WILLIAMS, David. Weighing the odds: a course in probability and statistics. New York: Cambridge University Press, 2001, 547 pp. ISBN 05-210-0618-X.

[22] DRUCKMÜLLER, M. A NOISE ADAPTIVE FUZZY EQUALIZATION METHOD FOR PROCESSING SOLAR EXTREME ULTRAVIOLET IMAGES. Online. The Astrophysical journal. Supplement series. 2013, roč. 207, č. 2, s. 1-5. ISSN 0067-0049. [cit. 2023-10-30]. Available from: https://doi.org/10.1088/0067-0049/207/2/25.

[23] DRUCKMÜLLER, Miloslav a DRUCKMÜLLEROVÁ, Hana. A Noise Adaptive Fuzzy Equalization Method with Variable Neighborhood for Processing of High Dynamic Range Images in Solar Corona Research. Online. In: Combinatorial Image Analysis. Cham: Springer International Publishing, s. 262-271. ISBN 9783319071473. ISSN 0302-9743. Available from: https://doi.org/10.1007/978-3-319-07148-0_23 [cit. 2023-10-30].

[24] L.A. Zadeh, Fuzzy sets, Information and Control, Volume 8, Issue 3, 1965, Pages 338-353, ISSN 0019-9958, Available from:https://doi.org/10.1016/S0019-9958(65)90241-X

[25] DRUCKMÜLLER, Miloslav a Pavel HERIBAN. Digital Image Processing System for Windows version 5.0. Brno: SOFO, 1996.

[26] AGGARWAL, Medhavi, Ranjit KAUR a Beant KAUR. Design of Efficient Adaptive Image Filters to suppress Salt and Pepper Noise. Journal of Emerging Trends in Computing and Information Sciences. 2014, No. 5, pp. 676-681. ISSN 2079-8407. Available from: http://www.cisjournal.org/journalofcomputing/Download_Sep_14_pdf_3.aspx

[27] PEETERS, M.H.G. Implementation of the phase correlation algorithm: motion estimation in the frequency domain. Technische Universiteit Eindhoven, 2003. Available from: https://pure.tue.nl/ws/portalfiles/portal/2318202/567039.pdf

[28] ARGYRIOU, V a T VLACHOS. Motion estimation using quad-tree phase correlation [online]. In: IEEE International Conference on Image Processing 2005. IEEE, 2005, [cit. 2021-7-30]. ISBN 9780780391345. ISSN 1522-4880.

[29] KONSTANTINIDIS, Dimitrios, Tania STATHAKI a Vasileios ARGYRIOU. Phase Amplified Correlation for Improved Sub-Pixel Motion Estimation. IEEE transactions on image processing [online]. United States: IEEE, 2019, 28(6), 3089-3101 [cit. 2021-7-30]. ISSN 1057-7149.

[30] HORN, Berthold K.P a Brian G SCHUNCK. Determining optical flow. Artificial intelligence [online]. Elsevier B.V, 1981, 17(1), 185-203 [cit. 2021-7-30]. ISSN 0004-3702.

[31] ZHANG, Gangfu a Hubert CHANSON. Application of local optical flow methods to high-velocity free-surface flows: Validation and application to stepped chutes. Experimental thermal and fluid science [online]. Elsevier, 2018, 90, 186-199 [cit. 2021-7-30]. ISSN 0894-1777.

[32] HUMPHREYS, Glyn W. a Vicki BRUCE. Visual Cognition: Computational, Experimental and Neuropsychological Perspectives. Psychology Press, December 25, 1989. ISBN 978-0-86377-124-8.

[33] SHEELEY, JR., N. R.; WARREN, H. P.; LEE, J.; CHUNG, S.; KATZ, J. et al. USING RUNNING DIFFERENCE IMAGES TO TRACK PROPER MOTIONS OF XUV CORONAL INTENSITY ON THE SUN. Online. The Astrophysical Journal. 2014, i. 797, n. 2. ISSN 1538-4357. Available from: https://doi.org/10.1088/0004-637X/797/2/131.

[34] THOMPSON, B. J.; GURMAN, J. B.; NEUPERT, W. M.; NEWMARK, J. S.; DELABOUDINIÈRE, J.-P. et al. SOHO/EIT Observations of the 1997 April 7 Coronal Transient: Possible Evidence of Coronal Moreton Waves. Online. The Astrophysical Journal. 1999, i. 517, n. 2, s. L151-L154. ISSN 0004637X. Available from: https://doi.org/10.1086/312030.

[35] PLOWMAN, Joseph. A center-median filtering method for detection of temporal variation in coronal images. Online. Journal of space weather and space climate. 2016, i. 6, s. A8. ISSN 2115-7251. Available from: https://doi.org/10.1051/swsc/2016002.

[36] MORGAN, Huw; BYRNE, Jason P. a HABBAL, Shadia Rifai. AUTOMATICALLY DETECTING AND TRACKING CORONAL MASS EJECTIONS. I. SEPARATION OF DYNAMIC AND QUIESCENT COMPONENTS IN CORONAGRAPH IMAGES. Online. The Astrophysical Journal. 2012, i. 752, n. 2. ISSN 0004-637X. Available from: https://doi.org/10.1088/0004-637X/752/2/144.

[37] LONG, D. M.; BLOOMFIELD, D. S.; GALLAGHER, P. T. a PÉREZ-SUÁREZ, D. CorPITA: An Automated Algorithm for the Identification and Analysis of Coronal "EIT Waves." Online. Solar physics. 2014, i. 289, n. 9, p. 3279-3295. ISSN 0038-0938. Available from: https://doi.org/10.1007/s11207-014-0527-5.

[38] STENBORG, G. a COBELLI, P. J. A wavelet packets equalization technique to reveal the multiple spatial-scale nature of coronal structures. Online. Astronomy and astrophysics (Berlin). 2003, i. 398, n. 3, p. 1185-1193. ISSN 0004-6361. Available from: https://doi.org/10.1051/0004-6361:20021687

[39] CANDES, Emmanuel, Xiandong LI a Yi MA. Robust Principal Component Analysis? Journal of ACM. ACM, 2011, , 11:1–11:37. ISSN 0004-5411. http://doi.acm.org/10.1145/1970392.1970395

[40] ZÁVIŠKA, Pavel, Ondřej MOKRÝ a Pavel RAJMIC. S-SPADE Done Right: Detailed Study of the Sparse Audio Declipper Algorithms. 2020. https://arxiv.org/pdf/1809.09847.pdf

[41] COMBETTES, Patrick L a PESQUET, Jean-christophe. Proximal Splitting Methods in Signal Processing. Online. ArXiv.org. 2010. Available from: https://doi.org/10.48550/arxiv.0912.3522 [cit. 2023-11-19].

[42] PARIKH, Neal. Proximal Algorithms. Foundations and Trends® in Optimization. 2014, 1(3), 127-239. ISSN 2167-3888. DOI:10.1561/2400000003

[43] BRUNTON, Steven L. (Steven Lee) a Jose Nathan KUTZ. Data-driven science and engineering: machine learning, dynamical systems. Cambridge: Cambridge University Press, 2019, xxii, 472 p. ISBN 978-1-108-42209-3.

[44] GROSEK, Jacob a J. Nathan KUTZ. Dynamic Mode Decomposition for Real-Time Background/Foreground Separation in Video. Computing Research Repository. (abs/1404.7592). https://arxiv.org/pdf/1404.7592.pdf

[45] ARBABI, Hassan. Introduction to Koopman operator theory of dynamical systems. MIT, 2018. https://www.mit.edu/~arbabi/research/KoopmanIntro.pdf

[46] GOLUB, Gene H. (Gene Howard) a Charles F VAN LOAN. Matrix computations. 3rd ed. Baltimore: Johns Hopkins University Press, 1996, xxvii, 694 s. ISBN 0-8018-5414-8.

[47] KUTZ, Jose Nathan, Steven L BRUNTON, Bingni W BRUNTON a Joshua L. PROCTOR. Dynamic mode decomposition: data-driven modeling of complex systems. Philadelphia: SIAM, Society for Industrial and Applied Mathematics, 2016, xvi, 234 p. ISBN 978-1-61197-449-2.

[48] WU, Ziyou; BRUNTON, Steven L a REVZEN, Shai. Challenges in dynamic mode decomposition. Online. Journal of the Royal Society interface. 2021, i. 18, n. 185, p. 20210686-20210686. ISSN 1742-5662. Available from: https://doi.org/10.1098/rsif.2021.0686.

[49] MORGAN, Huw a DRUCKMÜLLER, Miloslav. Multi-Scale Gaussian Normalization for Solar Image Processing. Online. Solar physics. 2014, i. 289, n. 8, p. 2945-2955. ISSN 0038-0938. Available from: https://doi.org/10.1007/s11207-014-0523-9.

[50] GEBRTOVÁ, Karolína. Rozklad videosekvencí na více složek s různou dynamikou. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav matematiky, 2023, 109 s. Diplomová práce. Vedoucí práce: prof. Mgr. Pavel Rajmic, Ph.D.

Scientifically less trustable references

[1b] MCGOODWIN, Michael C. Stars, Galaxies, and Beyond: Summary of notes and materials related to University of Washington astronomy courses. In: McGoodwin website [online]. 2012. Available from: https://www.mcgoodwin.net/pages/astrophysics_2012.pdf

[2b] General Astronomy [online]. Wikibooks, 2018 [cit. 2018-05-23]. Available from: https://upload.wikimedia.org/wikipedia/commons/f/f9/Astronomy.pdf

[3b] "Parts" of the Sun. In: UCAR Center for science education [online]. Boulder: University corporation for atmospheric research, 2012. Available from: https://scied.ucar.edu/sun-regions

[4b] GREEKGEEK, Ellen. If the Massive Solar Flare of 1859 (the "Carrington Flare") Happened Today... In: Owlcation [online]. 2011 [cit. 2018-05-24]. Available from: https://owlcation.com/stem/massive-solar-flare-1859

[5b] HILL, Steele. Coronal mass ejection. In: NASA Goddard Space Flight Center [online]. 2006 [cit. 2018-05-24]. Available from: https://www.nasa.gov/vision/universe/watchtheskies/july7_cme.html

[6b] CORONAGRAPHY. In: The Lyot project [online]. American museum of natural history [cit. 2018-05-25]. Available from: http://lyot.org/background/coronagraphy.html

[7b] FLECK, Bernhard. Fact sheet: SOHO. SOHO: Solar and Heliospheric Observatory [online]. 2003. Available from: http://sohowww.nascom.nasa.gov/about/docs/SOHO_Fact_Sheet.pdf

[8b] FISHER, Richard R. Coronagraphs. National Aeronautics and Space Administration: NASA [online]. 27 May, 1995 [cit. 2018-11-28]. Available from: https://umbra.nascom.nasa.gov/spartan/coronagraphs.html

[9b] LASCO CORONAGRAPH. In: Space weather prediction center: National oceanic and atmospheric administration [online]. Available from: https://www.swpc.noaa.gov/products/lasco-coronagraph

[10b] SOHO LASCO Latest images. In: Solar and Heliospheric Observatory [online]. Available from: https://sohowww.nascom.nasa.gov/data/realtime/c3/ or ttps://sohowww.nascom.nasa.gov/data/realtime/c2/

[11b] SDO| Solar Dynamics Observatory [online]. NASA official [cit. 2023-02-26]. Available from: https://sdo.gsfc.nasa.gov

[12b] CHEUNG, Mark. Atmospheric Imaging Assembly [online]. NASA official [cit. 2023-02-26]. Available from: https://aia.lmsal.com/index.htm

[13b] CORLAN, Radu. CCD Reduction. GCX User's Manual: Version 0.7.6 December 7, 2004 [cit. 2019-10-14]. Available from: http://astro.corlan.net/gcx/html/node7.html

[14b] Karim Nice, Tracy V. Wilson, Gerald Gurevich. How Digital Cameras Work. November 29, 2006 [cit. 2019-10-14]. Available from: https://electronics.howstuffworks.com/cameras-photography/digital/digital-camera.htm

[15b] DRUCKMÜLLER, Miloslav. Input data and examples of NAFE processing. NAFE Image Analyzer 1.0 [online]. 2013 [cit. 2023-11-19]. Available from: http://www.zam.fme.vutbr.cz/~druck/nafe/Data/0-info.htm

Used symbols and abbreviations

Abbreviations

CME	Coronal mass ejection, see section 1.1.2 or read [7] and [8]
SOHO	Solar and Heliospheric Observatory, see section 1.2.1 or read [7b]
LASCO	Large Angle and Spectrometric Coronograph, see section 1.2.1 or read [7b]
FITS	Flexible Image Transport System, see section 1.2.1 or read [12]
SDO	Solar Dynamics Observatory, see section 1.3 or read [11b]
AIA	Atmospheric Imaging Assembly, see section 1.3 or read [12b]
PDF	probability density function, see Definition 2.1 or read [13] or [14]
SVD	singular value decomposition, see Definition 2.35 or read [18]
CCD	charge coupled device, read [14b]
NAFE	Noise Adaptive Fuzzy Equalization, see section 3.3.2 or read [22]
PCP	principal component pursuit, see section 7.4or read [39]
ADMM	Alternating Direction Method of Multipliers, see section 7.4 or read [40]
DMD	dynamical mode decomposition, see section 7.5 or read [47]

Symbols

$N(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2, see Definition 2.1
H_0	null hypothesis, see Definition 2.2
H_1	alternative hypothesis, see Definition 2.2
$\mathbb{K}$	generalized notion for $\mathbb{R}$ and $\mathbb{C}$, see Remark 2.12
$\mathcal{M}_{m,n}(\mathbb{K})$	set of all matrices of size $m \times n$, see Remark 2.12
$\mathcal{M}_n(\mathbb{K})$	set of all square matrices of size n, see Remark 2.12
$f(x, y)$	digital grey scale image, see Definition 3.1
$B(x, y)$	bias value of each pixel, see section 3.2.1
$D(x, y)$	dark current, see section 3.2.1
$G(x, y)$	sensitivity of each sensor on the chip, see section 3.2.1
$I(x, y)$	light flux reaching the pixel, see section 3.2.1
$R(x, y)$	noise component of image, see section 3.2.1
$b(x, y)$	bias frame, see section 3.2.1
$\widetilde{B}$	estimation of bias value of each pixel, or master bias frame, see section 3.2.1
$d(x, y)$	dark frame, see section 3.2.1
$\widetilde{D}(x, y)$	estimation of dark current, see section 3.2.1
$\widetilde{D}^*(x, y)$	bias subtracted master dark frame, see section 3.2.1
$D_M(x, y)$	master dark frame, see formulas (3.2) and (3.3)

$ff(x,y)$	flat-field frame, see formula (3.4)
$\widetilde{F}_M(x,y)$	master flat-field drame, see formulas (3.6) and (3.7)
$\bar{G}$	average of sensitivity G over whole image, see section 3.2.1
$\bar{R}$	average of random element (noise) over whole image, see section 3.2.1
$\tilde{B}^n_{x,y}$	fuzzy neighborhood, see section 3.3.2
$b_{k,l}$	membership grade, see section 3.3.2
$h^n_{x,y}(t)$	fuzzy histogram, see section 3.3.2
$H^n_{x,y}(t)$	cumulative fuzzy histogram, see section 3.3.2
$C^n_{x,y}(t)$	normalized cumulative fuzzy histogram, see section 3.3.2
$g^n_{x,y}(t)$	fuzzy equalizing function, see section 3.3.2
x	the pixel value, see section 4.1.1
$\bar{x}$	any statistical measure of position of statistical sample, see section 4.1.1
ε	constant affecting the level of significance and power of the test, see section 4.1.1
$M_{k,l}(G)$	general geometric moment of object G with order $k+l$, see section 5.1.1
$M_{k,l}(G)_\alpha$	moment of order $k+l$ with dependence on rotating angle α, see section 5.1.1
$CM_{k,l}(G)$	central geometric moment of order $k+l$, see section 5.1.1
$NCM_{k,l}(G)$	normalized central geometric moment of order $k+l$, see section 5.1.1
$PM_{k,l}(G)$	principal moment of order $k+l$, see section 5.1.1
A	semi-major axis of the ellipse, see section 5.1.1
B	semi-minor axis of the ellipse, see section 5.1.1
EL	elongation of the ellipse, see section 5.1.1
$m_{k,l}(G)$	physical moment of object G with order $k+l$, see section 5.1.2
$cm_{k,l}(G)$	central physical moment of order $k+l$, see section 5.1.2
$ncm_{k,l}(G)$	normalized central physical moment of order $k+l$, see section 5.1.2
$pm_{k,l}(G)$	principal physical moment of order $k+l$, see section 5.1.2
a	semi-major axis of the ellipse, see section 5.1.2
b	semi-minor axis of the ellipse, see section 5.1.2
el	elongation of the ellipse, see section 5.1.2

List of Figures

List of Tables

Appendix A

Enclosed CD

The CD contains following files:

- the doctoral thesis with active cross references
- testing images from SOHO LASCO (more of them can be obtained from SOHO)
- source code for filtration of impulse noise
- testing images from SDO-AIA (more of them can be obtained from SDO)
- source code for separation of CMEs
- shortened versions of resulting videos (the whole videos can be obtained from author of this thesis)

Appendix B

Computer specifications

Table 2: The computer specifications used for the calculations.

Memory (RAM)	8×16 GB 2666 MHz RDIMM
Processor (CPU)	2× Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz, 18 cores, 36 threads
Graphics processor (GPU)	NVIDIA Corporation GP100GL [Quadro GP100] (rev a1) NVIDIA Corporation GP104GL [Quadro P4000] (rev a1)
Storage	2× HDD 2 TB: ATA ST2000DM001-1ER1, 2 TB SSD 500 GB: SAMSUNG MZVLB512HAJQ-000H1