Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Elin Thomfohrde-Dammann

An Algorithm for the Discovery of Variable Stars and Transients in the Transiting Exoplanet Survey Satellite data archive

Fakultät Technik und Informatik
Studiendepartment Informations- und Elektrotechnik

Faculty of Engineering and Computer Science
Department of Information and electrical engineering

Elin Thomfohrde-Dammann

An Algorithm for the Discovery of Variable Stars and Transients in the Transiting Exoplanet Survey Satellite data archive

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Elektrotechnik und Informationstechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Klaus Jünemann
Zweitgutachter: Dr. Volker Perdelwitz

Eingereicht am: 3. März 2023

Elin Thomfohrde-Dammann

Thema der Arbeit
Ein Algorithmus zur Entdeckung von Variablen Sternen und Transienten im Datenarchiv des Transiting Exoplanet Survey Satellite

Stichworte
Bildverarbeitung, Photometrie, Bildsubtraktion

Kurzzusammenfassung
In dieser Bachelorarbeit wird ein Python-Algorithmus zur Entdeckung von variablen Sternen und Transienten in den Full-frame Images (FFIs) des Transiting Exoplanet Survey Satellite (TESS) entwickelt. Der vorgeschlagene Ansatz beruht auf der Implementierung von Bildsubtraktion, um nicht-variable Objekte zu entfernen und das Rauschen in den Lichtkurven zu reduzieren. Die Wirksamkeit dieser Methode wird durch die erfolgreiche Identifikation bekannter bedeckungsveränderlicher Sterne demonstriert. Mit dem Code ist es möglich, ein Transientenarchiv zu erstellen, welches der größte Katalog seiner Art sein wird und der astrophysikalischen Gemeinschaft zur Verfügung gestellt werden kann.

Title of the paper
An Algorithm for the Discovery of Variable Stars and Transients in the Transiting Exoplanet Survey Satellite data archive

Keywords
image processing, photometry, image subtraction

Abstract
The goal of this thesis is the development of a python algorithm for the discovery of variable stars and transients within Transiting Exoplanet Survey Satellite (TESS) Full-frame Images (FFIs). The proposed method is based on the implementation of image subtraction to remove non-variable objects and reduce noise in the light curves. The effectiveness of this approach is evaluated by showing the recoverability of known eclipsing binaries. The code will enable the creation of a transient archive which will constitute the largest catalogue of its kind and will be made available to the astrophysical community.

Contents

List of Tables

List of Figures

Acronyms

CCD charge-coupled device

FFI Full-frame Image

FITS Flexible Image Transport System

TESS Transiting Exoplanet Survey Satellite

TPF Targeted Pixel Files

WCS World Coordinate System

1 Introduction

In 2018, NASA's Transiting Exoplanet Survey Satellite (TESS) began its science operations, conducting a near all-sky survey to observe millions of stars and enable a wide range of astrophysical investigations. The TESS mission has the primary goal of discovering exoplanets orbiting bright stars via the transit method. These planets can later be studied in detail using follow-up spectroscopic observations to determine the planets' masses and atmospheric compositions[1]. For a large number of pre-selected stars, NASA provides pipeline-produced light curves that can be used to identify transients based on changes in the observed brightness during transit. However, observations for the majority of stars in TESS' field of view are only available in the form of Full-frame Images (FFIs), which are expected to contain a wealth of previously unidentified variable stars and transients. By studying variable stars, astronomers can learn about stellar properties like mass, radius, luminosity, temperature, internal and external structure, composition and evolution[2]. The study of transients can provide valuable insights into the dynamics and evolution of the universe[3].

1.1 Objectives

The objective of this thesis is to develop an algorithm to automatically identify candidates for transients and variable stars in the TESS FFIs and the extraction of light curves. A light curve is a graph that shows how the brightness of an object changes over time and is obtained by measuring the flux of a source at several points in time. The algorithm uses image subtraction to identify and measure the brightness of variable stars, as well as transient events which are not associated with stellar sources.
The program takes the FFIs provided as data products on the website of the Mikulski Archive for Space Telescopes[4] as its input and produces a set of light curves. The light curves are created for positions in the sky that show variability, indicating transient events or variable stars. To evaluate the performance of the algorithm it is shown that it is possible to recover known eclipsing binaries.

1.2 Structure of the thesis

The thesis is structured as follows. In Chapter 2, the scientific context of the research is presented, including an introduction to transients, TESS, image subtraction and aperture photometry. This chapter lays the foundation for the subsequent chapters and provides a comprehensive overview of the astrophysical and technical concepts used in the thesis. In addition, it discusses other relevant works in the field.
Chapter 3 analyses the user requirements and constraints that need to be taken into account when developing the software. This analysis results in a list of requirements that the program needs to meet. These requirements are the basis for the overall software design in Chapter 4, where the validation process is also discussed. That chapter provides a detailed overview of the architecture of the software, the algorithms used and the data formats employed. Additionally, it discusses the programming language and coding style that was used in the implementation of the software.
Chapter 5 delves into the implementation details of the software design, including the algorithms and techniques employed. The chapter discusses how the software was developed to meet the requirements laid out in Chapter 3 and provides a detailed explanation of the methods used to extract light curves from TESS Full-frame Images.
In Chapter 6, the algorithm's performance and accuracy are evaluated. The ability of the algorithm to detect known eclipsing binaries is tested, and the results are compared to the results of other works to determine the algorithm's precision.
Finally, Chapter 7 provides a summary of the findings, discusses the limitations and potential applications of the algorithm and suggests areas for further research.

2 Scientific context

This chapter presents the scientific context that is essential for comprehending the work presented in this thesis. It introduces significant key terms, concepts and methods and provides an overview of the TESS mission. Additionally, this chapter describes the Full-frame Images (FFIs) that the program processes and reviews previous work in the field that influenced the approach to this task.

2.1 Variable stars and transients

This thesis aims to discover variability in the TESS FFIs. This section provides a general overview of the different classes of variability in the sky and why they are of interest to astronomers.

Variable stars

Variable stars exhibit periodic or stochastic fluctuations in their observed brightness, which can result from a variety of physical processes occurring within the star (intrinsic variables) or its surrounding environment (extrinsic variables)[2]. One type of variable stars that are commonly observed are eclipsing binaries, which are comprised of two stars orbiting around each other. The orientation of their orbital plane causes one star to periodically pass in front of the other, resulting in recurrent dips in the observed brightness of the system[5].
Astronomers study variable stars as they offer valuable insights into the internal structure and evolution of stars, as well as their physical properties, such as mass, radius and temperature[2]. In this thesis, variable stars, particularly eclipsing binaries, are utilised to validate the results of the program since a substantial number of such systems with known properties are present across the entire sky.

Transients

Transients are objects or events that show sudden, non-periodic changes in brightness over relatively short periods of time. There are several types of transients, including supernovae, gravitational microlensing events and stellar flares[3].

- **Supernovae**: These are massive stellar explosions that mark the end of a star's lifetime. They are among the brightest events in the universe. Depending on the properties of the star that exploded and the presence or absence of certain elements in the supernovae's spectrum, they can be classified into different types.
- **Microlensing events**: These are changes in the apparent brightness of a distant star due to the gravitational field of an object passing in front of it, such as a star or a planet. The light from the source is bent and focused by the foreground object's gravitational field, resulting in a temporary increase in the star's brightness. Microlensing events can be used to identify objects that are otherwise to dark to observe.
- **Stellar flares**: These are sudden increases in brightness of a star which are caused by magnetic processes on the stellar surface.

Transients are important objects of study because they can provide insights into some of the most variable or explosive phenomena in the universe, helping astronomers to understand various aspects of the dynamics and evolution of the universe[3].

2.2 Aperture photometry

Aperture photometry is a technique used to measure the brightness of a celestial object. The brightness of an object is defined as its measured energy per time[6]. It involves using an aperture, or region of the sky, around the object of interest. The brightness is then measured by summing the flux within the aperture[1], with the flux being the brightness divided by the area over which it was measured[6]. The size of the aperture is typically chosen to be larger than the effective size of the object, but small enough to not include any nearby objects that might bias the measurement.

2.3 Computer vision pipeline

The computer vision pipeline is a series of steps that are commonly used by computer vision systems to process and analyse visual data[7]. The general structure of a computer vision pipeline is shown in Figure 2.1.

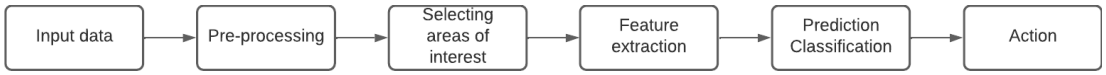

Figure 2.1: General structure of the computer vision pipeline

The pipeline starts with the acquisition of visual data, which can come from sources such as cameras. In the pre-processing stage, the data is standardised. The next two stages identify and extract relevant features from the data, which are needed to define certain objects. The prediction or classification stage can use a machine learning classifier to make decisions or trigger actions[8].

2.4 TESS mission

This section provides an overview of the TESS mission, including its main objective, observing strategy, science instruments and data products.

2.4.1 Mission objectives

The Transiting Exoplanet Survey Satellite mission is a NASA mission led by the Massachusetts Institute of Technology (MIT) with the primary goal of discovering exoplanets orbiting nearby bright stars[9]. TESS was launched on April 18, 2018, aboard a SpaceX Falcon 9 rocket and reached its final orbit 60 days later. Regular science operations began on July 25, 2018[1]. Since then TESS has observed nearly the entire celestial sphere and observations will continue until some critical system failure occurs, which may be years or decades from now[1].
TESS is designed to identify exoplanets using the transit method. The transit method involves measuring periodic dips in the brightness of a star to detect the presence of a planet. These dips occur when the planet passes in front of the star from the observer's perspective, causing a small fraction of the star's light (usually around 1%) to be blocked. Figure 2.2 schematically shows how this method works.

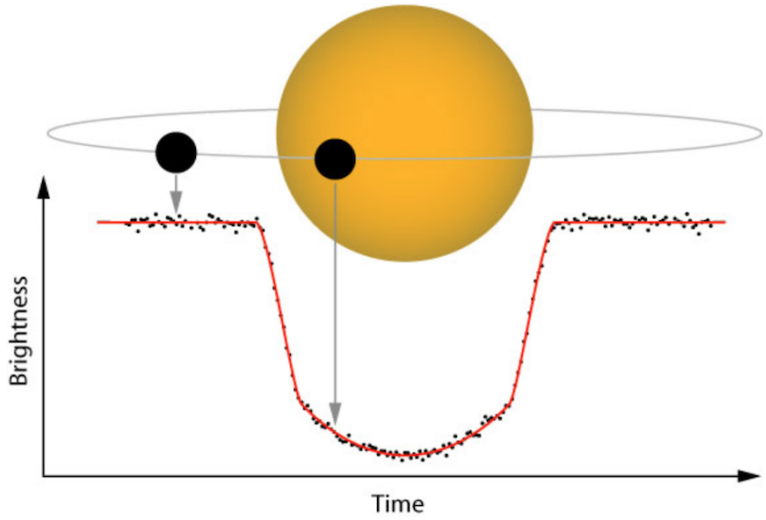

Figure 2.2: Schematic of the light curve of a transiting exoplanet. The transiting planet blocks a fraction of the stars light as viewed by the spacecraft. Image credit: NASA[1]

By focusing on bright stars that are located within about 200 light-years of Earth, TESS aims to find a diverse range of exoplanets. The ultimate goal of the TESS mission is to contribute to the understanding of the prevalence and diversity of exoplanets in our galaxy. Up till today, TESS has discovered 285 confirmed planets, with another 6137 candidates in the vetting process[10].

Observing strategy

TESS uses a set of four wide-field cameras mounted on the satellite to observe the sky, each with a field of view of 24x24 degrees. When combined, the four cameras cover an area in the sky of 96x24 degrees known as an observing sector[9]. TESS observes each sector for two orbits of the satellite, with a combined duration of about 27 days.
The total search space of TESS includes the entire celestial sphere, out to a distance of about 200 light-years[11].

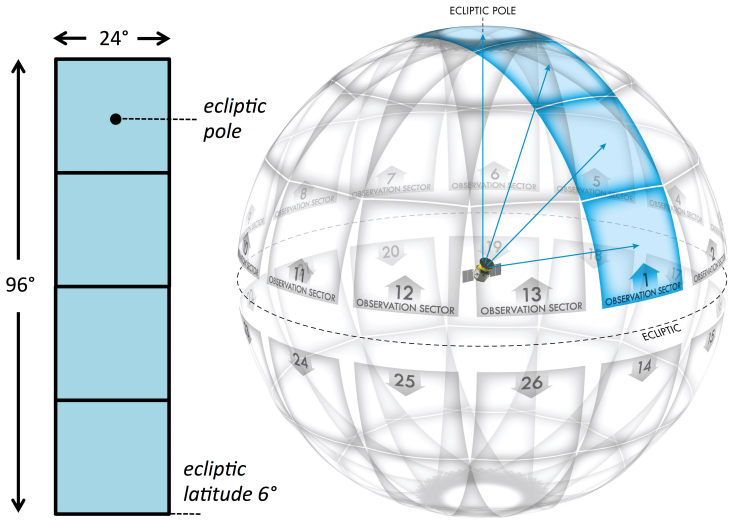

Figure 2.3: Field of view of one observing sector and the sectors observed during the primary mission. Image credit: NASA[1]

Figure 2.3 shows the 26 sectors observed during the primary mission, which lasted from July 2018 to July 2020 and covered both the northern and southern celestial hemispheres. The sectors overlap at the ecliptic poles, meaning these areas were continuously monitored for a full year. TESS observed 15 000 target stars per sector with a 2-minute cadence and collected FFIs every 30 minutes[1].
After the primary mission, TESS entered its first extended mission, which lasted from July 2020 to September 2022. During this mission, TESS re-observed parts of both northern and southern celestial hemispheres, with a total of 26 sectors being observed. The FFIs were collected at a 10-minute cadence instead of the 30-minutes cadence used during the primary mission. The targeted stars during this mission were mainly suggested through the guest investigator program.
The second extended mission, which began in September 2022 and will continue until 2024, will provide FFIs collected at a cadence of 200 seconds[1].

Orbit

To enable continuous observation of large portions of the sky in both celestial hemispheres, TESS is placed in a highly stable, elliptical high-Earth orbit. This orbit allows TESS to maintain a clear view of the sky, free from obstruction by Earth, Sun or Moon. The low-radiation and near-constant temperature environment in which its cameras operate helps to minimise noise

and improve the quality of the data collected by the spacecraft.
This highly elliptical orbit, proposed by Gangestad et al.[12], has a period of 13.7 days and is therefore in a 2:1 resonance with the moon, helping to keep the orbit stable. The distance between TESS and Earth varies between 108 000 km at its closest point (perigee) and 375 000 km at its farthest point (apogee). At perigee, TESS is able to exchange data with ground stations on Earth[1]. Figure 2.4 shows the TESS orbit.

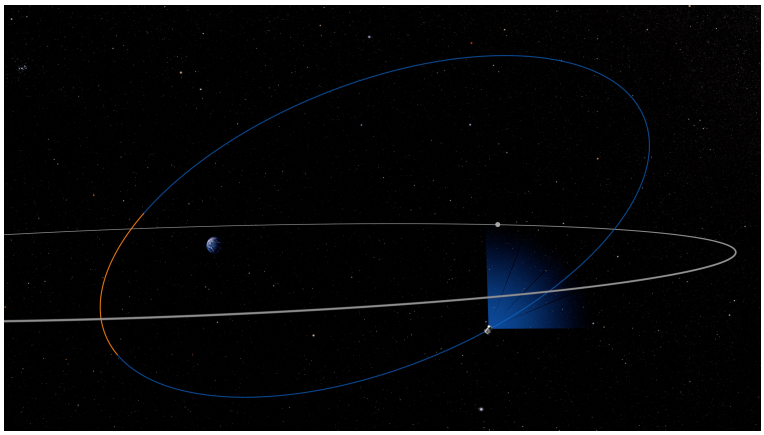

Figure 2.4: During the blue part of the orbit TESS observes the sky, while the orange part is used to transmit data to Earth. Image credit: NASA[13]

2.4.2 TESS spacecraft

The TESS telescope consists of a spacecraft and a payload, with the payload being a single science instrument - a camera system consisting of four wide-field-of-view optical cameras and their associated data handling unit. The payload is housed on the spacecraft, which is equipped with a range of subsystems to support its scientific mission and maintain its position in orbit[1]. Figure 2.5 shows the satellite.

Figure 2.5: The TESS spacecraft. Image credit: MIT[1]

TESS cameras

Each of the four identical cameras has seven lenses and four charge-coupled device (CCD) detectors[14]. A CCD is a highly sensitive light detector composed of a multitude of light-sensitive semiconductor elements, that convert the energy of photons into electrical charge, which can then be read out and converted into digital values[15]. In astronomy, these digital values then usually represent units of count or electrons per second[6]. In the case of calibrated TESS FFIs the pixel values are in the unit of electrons per second [16].
To prevent scattered light from Earth and Moon from entering the cameras, each camera has a lens hood[14]. The cameras are placed on a camera plate which is again enclosed by a sunshade to block stray light.
The CCDs are sensitive to light in the visible and near-infrared spectrum (600 - 1000 nm)[14]. This spectrum was chosen because it corresponds to the wavelength of peak emission of M dwarfs, which are the most common type of star in the galaxy[17]. M dwarfs are also of particular interest because they are relatively small. The depth of transit can be estimated as follows[18]:

$$\frac{\Delta F}{F} = \left(\frac{R_P}{R_S}\right)^2 \tag{2.1}$$

With F being the flux of the star, R_P the planet's radius and R_S the star's radius. Therefore the transit signals for smaller stars are larger and hence easier to detect.

2.4.3 Data products

The TESS mission provides several data products to the public, which are available for download on the website of the Mikulski Archive for Space Telescopes[4].

- **Targeted Pixel Files (TPF)**: These are files that contain a time series of a small region of pixels around a targeted object. They are useful to study individual stars in more detail.
- **Light curves**: These are derived from TPFs and contain a time series of the flux for a particular object.
- **Secondary data products**: There a several other data products of TESS, a full list of which can be found on the web page of the Mikulski Archive for Space Telescopes.

However, the focus of this work is on the Full-frame Images (FFIs), which are images of the entire field of view of one CCD. The data acquired by each CCD is published individually, therefore there are 16 FFIs for each time cadence of each sector. The FFIs are available in both calibrated and uncalibrated formats. Calibrated data has undergone basic image reduction procedures, such as bias, dark current, and flat field corrections, thus removing instrumental artifacts[1]. Therefore, the calibrated images were used throughout the work described in this thesis.
The TESS FFIs can be interpreted as grey-scale images, with the value of each pixel representing the measured flux in electrons/sec[16]. It is important to note that FFIs have regions at the edge of the CCDs which do not contain useful information due to the way the CCDs are constructed and read out. They are visible in figure 2.6 on the left, top and right edges of the image in the form of a darker edge.

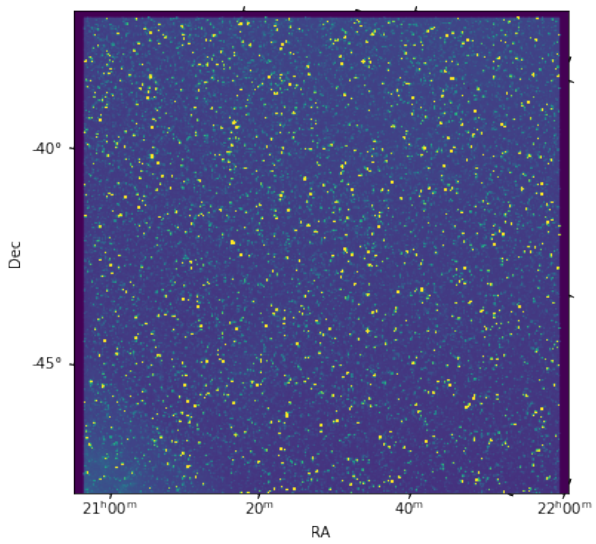

Figure 2.6: An example FFI from Sector 1, Camera 1, CCD 1, as provided as a data product in the Mikulski Archive.

FITS file format

The TESS FFIs are stored in the Flexible Image Transport System (FITS) format, which is a file format commonly used for storing astronomical data, such as images or data tables. Its development started in the 1970s, due to the need of transferring astronomical images from one institute to another. The quick adoption of this format for exchanging data among astronomers resulted in it being officially recognised by the International Astronomical Union in 1982 as the standard for the field[19].
A FITS file is a binary file that consists of one or more ”headers” followed by the actual data, which is stored as an array of numbers in a simple binary format. The headers consist of a series of ASCII keyword/value pairs that provide meta-information about the data such as the size of an image, the data type or the coordinate system. In the case of TESS, it also contains a data quality flag, which indicates anomalies in the data[16].
A detailed description of the FITS file format can be found in the FITS data format description document[19]. There are several available for accessing FITS files, like libraries in many programming languages. The one used for this thesis is the astropy.io.fits package for python[20].

Celestial coordinates

The location of objects in the sky can be described by using a construct called the celestial sphere. It is an imaginary sphere of an infinite radius with Earth at its centre. Any point on it can be defined using two coordinates: right ascension and declination, which work in the same way as latitude and longitude do to describe a location on Earth[21]. Figure 2.7 shows an illustration of the celestial sphere.

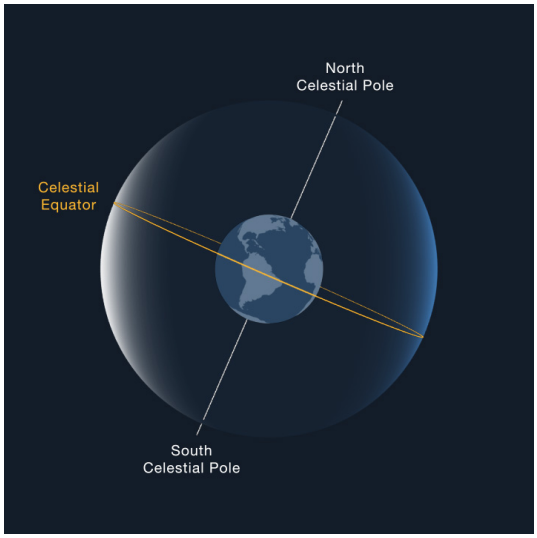

Figure 2.7: The Celestial Sphere. Image credit: NASA[21]

World Coordinate System

The World Coordinate System (WCS) is a standard for representing coordinates of celestial objects in astronomical images. It is used to map pixel coordinates of an image to celestial coordinates, such as right ascension and declination.
The WCS is defined by a set of mathematical transformations to convert from pixel to celestial coordinates and vice versa. The transformation parameters are listed in the header of FITS images.
Using the WCS allows astronomers to accurately determine the position of objects in the sky and compare different observations from individual images to each other. It has been adopted as a standard by the International Astronomical Union and is supported by a variety of software tools, for example, the astropy.wcs[20] package for python, which was used in this

thesis. A detailed description of the representation of the WCS in FITS files can be found in the publication by Greisen and Calbretta in 2002[22].

Image artefacts

The quality of the data produced by TESS can be affected by various factors, including saturation, crowding and scattered light. These effects should be taken into account when working with the data.
Saturation occurs when the intensity of the light reaching a CCD's pixel within the exposure time is too high, causing the charge of the pixel to spread into other pixels of the same row, an effect known as blooming[23]. Figure 2.8 shows this effect in one of the TESS FFIs. Accurate photometry for saturated stars would require including the pixels showing blooming.

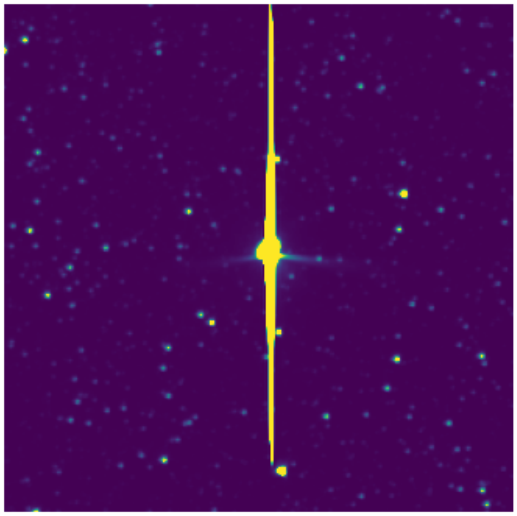

Figure 2.8: Blooming caused by a bright star affecting other pixels within the same row. Cutout from an FFI from Sector 18, Camera 1, CCD 2.

Crowding refers to the case where two or more objects in the sky are located so close together that the telescope can not resolve them individually, prohibiting photometry of the individual components. Due to the design requirement of a large field of view and a resulting pixel size of 21 arcseconds, TESS is particularly susceptible to this effect[1].
Scattered light from Earth and Moon can also be visible in the FFIs. Many of the images show some contamination by scattered light that the sunshades were unable to block. This effect is visible in figure 2.9. Typically, the brightness in these areas is between 2 and 6 times higher than the nominal sky background[14].

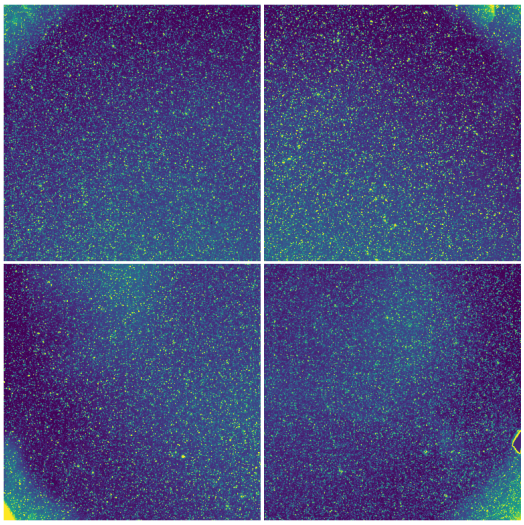

Figure 2.9: The brighter patches in the images are caused by scattered light from Earth and Moon. The image shows all four CCDs of Sector 1, Camera 1.

2.5 Previous work

The search for transients and variable stars in crowded fields has been an active area of research for some time, particularly with the rise of CCD cameras with large fields of view that allow for longer observation periods of large densely crowded fields of the sky. This section reviews relevant work in the field, followed by a motivation for the improvement carried out in the course of this thesis.

DAOphot: A computer program for crowded-field stellar photometry

In 1987, Stetson published a paper introducing a program for automatically detecting stars in digital images by searching for local maxima and performing synthetic aperture photometry[24]. However, he noted that one of the major drawbacks of his approach was the lack of a method to cross-identify stars in multiple frames or account for different seeing conditions when performing photometry on multiple frames. Stetson proposed using positional coincidence to cross-identifies stars between data frames.

Image subtraction

Motivated by the search for microlensing events, Alard and Lupton presented a method for image subtraction to detect variable objects in crowded fields[25]. They proposed that the

photometry and detection of variable objects should be based on the difference between frames. This process involves several steps. First, multiple images of the same area of the sky are taken at different times. Then, the images are scaled to match the seeing conditions of a reference image using 2D convolution. Finally, the reference image is subtracted from each of the images. Ideally, this leaves only objects that changed in brightness or that do not appear in both images. Alard and Lupton used the image with the best seeing conditions as the reference image and employed a set of Gaussian functions multiplied by a polynomial as the convolutional kernel. They assumed that most pixels in the two images should be identical, allowing the kernel to be determined by minimising the difference between the images through least squares optimisation.

Precision Light Curves from TESS Full-frame Images: A Different Imaging Approach

Oelkers et al.[26] developed a method for extracting light curves from TESS FFIs using image subtraction based mainly on the work of Alard and Lupton[25], as well as Alard[27], Bramich[28], Miller[29], and Oelkers et al.[30]. This has already been shown to be effective in mitigating the effects of crowding and variable point spread functions in the images. Their approach involves three steps:

1. Background subtraction and image alignment: To estimate the background, which is then subtracted, the sky background is sampled every 32x32 pixels and interpolated between these boxes. For aligning the images they used the WCS embedded in the header of each image.

2. Master frame creation: To create the master frame, all available images are first split into smaller subsets and median-combined. The resulting images are then in turn median-combined to create the final reference image.

3. Image subtraction, aperture photometry and trend removal: The image subtraction method is based on the one presented by Alard and Lupton[25], but uses a different convolutional kernel that is better suited for TESS FFIs.

According to Oelkers et al., their pipeline has a run-time of 3 minutes per FFI and is able to achieve the expected level of precision. They also demonstrate that it is possible to recover variable stars using their pipeline.

Quick-look Pipeline

The TESS Quick-look Pipeline by Huang et al.[31], presents a program to extract light curves from TESS FFIs. They published light curves for all bright sources in the TESS Input Catalogue observed during the primary mission. The process for extracting the light curves involves several steps, including background correction, image subtraction and aperture photometry. The measured flux is then converted from difference flux to absolute flux using the expected flux for each source based on a magnitude estimate. The pipeline also includes post-processing steps such as detrending and outlier removal to improve the quality of the light curves.

TESS Backdrop

Since the launch of TESS, multiple software tools for the mitigation of background effects within TESS FFIs have been developed, including TESS Backdrop, which was developed by Christina Hedges and is available on GitHub[32]. The package uses a linear model of polynomials and splines to model the background. To account for straps caused by the CCD, the model allows for deviation as pixel column offsets [32]. The different parts of the model, as shown in Figure 2.10, are combined to create the final model.

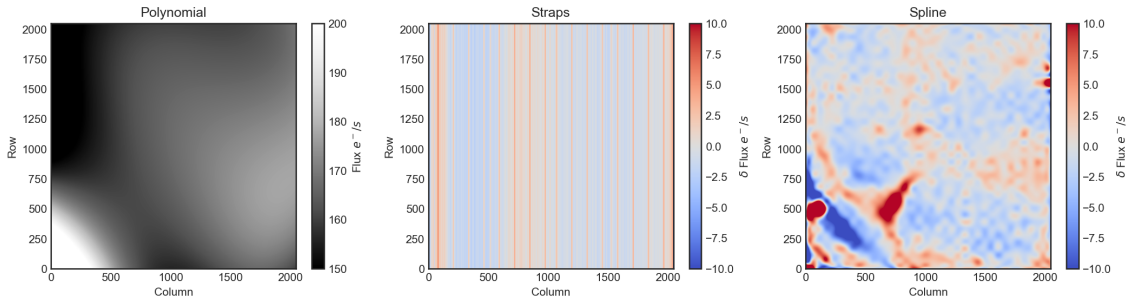

Figure 2.10: Model components of the backdrop model. Image Credit: Christina Hedges[32].

Target improvements carried out in this work

Overall, the image subtraction approach has proven to be effective in the past for extracting light curves from TESS FFIs, as demonstrated by the work of Oelkers et al.[26]. However, their method requires the user to provide the coordinates of the targeted stars. In this thesis the star finding routine is applied to each individual frame after image subtraction to automatically identify variable stars and transients. Additionally, the issue of cross-matching the coordinates of the same star in different images, as pointed out by Stetson in his work on crowded-field

photometry, is addressed.
Another issue with the method of Oelkers et al. is the lengthy run-time needed to process a data set. The program presented in this thesis aims to significantly reduce the computation time, while still maintaining the precision of the light curve results. One of the slowest parts of the program by Oelkers et al.[26] is the background correction. Therefore, TESS backdrop was tried to determine whether it produces similar results while reducing the required computation time. This will enable the processing of the entire TESS archive, resulting in an unprecedented catalogue of transient events for the astronomical community.
The light curves from the TESS Quick-look Pipeline were used for comparison to the light curves extracted using the program presented in this thesis.

3 Requirements

This chapter analyses the user requirements and constraints that have been identified for the program. The analysis results in functional and non-functional requirements the program needs to meet.

3.1 User requirements and constraints

Scope

The scope of the program is to process the TESS FFIs to reduce them to light curves of possible variable stars and transients. To achieve this 3 main steps can be defined.

- **Pre-processing**: The program should perform pre-processing steps on the TESS FFIs, including correcting for background effects. It should also correct inaccuracies in the image alignment.
- **Image subtraction**: The program should perform image subtraction on the pre-processed FFIs to enable the identification of differences between the images. As explained in section 2.2, this requires a reference image, which has to be constructed.
- **Feature extraction**: The program should automatically identify variable sources in the FFIs and extract the light curves. The light curves should be saved to a user-friendly format, to be easily accessible for future work.

The classification of the resulting light curves and removal of false positives is beyond the scope of this work, and will be carried out by the astronomical community in the near future. The program is also not able to extract precise photometric measurements for saturated pixels that bloomed over larger parts of the image since these areas only constitute a minor fraction of the overall observed area and would require a different aperture than most sources.

User requirements

The program is intended for use by researchers, mainly astrophysicists, who may have specific needs or fields of interest and may require only certain parts of the program's functionality. In order to meet the needs of these users, the program must have the following capabilities:

- **Flexibility and customisation**: The program must be modular and parameterisable to allow for customisation and enable researchers to adapt it to their specific needs. This will also allow researchers to potentially distribute the workload across multiple machines.
- **Ease of use**: Researchers in this field have different levels of programming skills, so the program must be easy to use. It does not require a graphical user interface, since programs that are operated from the command line or by directly writing the parameters into the code are fairly common in the community.
- **Data compatibility**: The program must be able to handle the specific data format of the TESS FFIs without requiring any additional preparation from the user. The output data from the program must be compatible with common tools or software used in the field in order to facilitate its use by researchers.
- **Performance**: Researchers will need to process data in a timely manner, and may have limited access to computer resources or deadlines to meet. Therefore, the program must be efficient and able to handle large volumes of data.
- **Accuracy**: The main objective of the algorithm is to maximise the true positives in the detection of variable stars and transients, while the number of false positives is of secondary importance. The accuracy of the light curves should be sufficient to enable classification. The positional accuracy should be sufficient to identify known objects.

Overall, the program must be easy to use, flexible and efficient in order to meet the needs of researchers in this field.

Constraints

The following constraints were taken into consideration:

- **Hardware**: The program is developed on a high-performance computer (AMD Ryzen 9 3950X, 64 GB RAM and RTX 2080 Ti).

- **Maintainability**: As TESS is still in operation, the program must be maintainable in the face of potential changes to TESS data, so it stays applicable. This also enables users to adapt the program to existing or future telescope data.

3.2 Resulting requirements

This chapter outlined the requirements and constraints which the program must meet. The resulting requirements can be summarised as follows:

3.2.1 Functional requirements

1. The program corrects for systematic errors, mainly scattered light
2. The program corrects for inaccuracies in the image alignment
3. The program constructs a reference frame for the image subtraction
4. The program performs the image subtraction
5. The program identifies variable sources
6. The program extracts light curves for a given set of coordinates

3.2.2 Non-functional requirements

a) The program should be modular to allow for customisation and flexibility

b) The program should be easy to use and be accessible for people with basic programming skills

c) The program should process the images in a timely manner

d) The output data should have a format that is compatible with common tools in the field

e) The program should be sufficiently accurate to produce reliable results, enabling classification as well as identification of known objects.

4 Design

This chapter describes the program design. The design is based on the requirements outlined in section 3.2 and takes into consideration factors such as the programming language and coding style, the program architecture, the format and storage of the generated data and the validation processes that is employed.

4.1 Programming language and coding style

Python was chosen as the programming language for a number of factors. Python is a popular language that is well-suited for use in scientific and data-driven applications such as this project.
One of the main advantages is that it is relatively easy to write and read, thanks to the simple and clean syntax. This can make it easier for developers to understand and maintain the code, as well as for users to use the software and is therefore in line with requirement b).
Another advantage of Python is the extensive availability of libraries and frameworks that can be used to extend the capabilities of the language. These libraries cover a wide range of domains, including data analysis, astronomy, scientific computing and more, which cuts down on the development time.
In addition, Python has several tools and features that can make testing and debugging easier, such as Jupyter notebooks, which allow developers to run individual sections of code.
Despite these advantages, Python has some drawbacks. One of the main limitations of Python is that it is relatively slow compared to other programming languages. Also, Python is prone to run time errors, since it is an interpreted language rather than a compiled one. Finally, conflicting dependencies of Python packages can make it tedious to find a working configuration of package versions.
Despite these disadvantages, Python was ultimately chosen because of the high support for astronomical and data-driven applications, as well as the high productivity it offers. The extensive libraries and frameworks that are available for Python made it particularly appealing for this project since for a lot of previous works, like the ones presented in section 2.5, there are python implementations available.

In addition to choosing a suitable programming language, it is also important to establish a consistent coding style throughout the program. This makes the code easier to understand and work with for developers and users alike. To this end, the following guidelines for the coding style were established:

- variable names are written in snake_case
- function names are written in camelCase
- variable and functions names are descriptive
- At the beginning of each function, there is a short description of the function and its input and output parameters, as shown in Figure 4.1

```
1  def exampleFunction(input_a, input_b):
2      """
3      The function shows what a function in this program should look like
4      Inputs:
5      input_a[data_type]: short despricption
6      input_b[data_type]: short despricption
7      Returns:
8      output[data_type]: short description
9      """
10     #do something
11     out = input_a + input_b
12     return out
```

Figure 4.1: Template of how a function is written in this program

One important thing to note when using functions in Python is, that the parameter's are passed by assignment. This means that, depending on whether a passed object is mutable (e.g. a list) or immutable (e.g. an integer), changes to the parameter inside the function may affect the parameters value in the calling function as well[33]. Therefore, parameters that are modified within a function are returned and reassigned. This way the code is clearer and easier to understand and unintentional modification of variables is prevented.
Following these guidelines ensures that the code for this project is easy to understand and maintain, as well as consistent and organised, complying with requirements a) and b).

4.2 Program architecture

A well-structured program architecture is crucial in achieving a code that is easy to read, maintain and adapt. In order to meet the processing requirements of the TESS FFIs, as outlined in section 3.2, the program logic was divided into several sequential steps. The program

steps were encapsulated into functions to comply with the requirements for modularity and flexibility (requirement a)). Those functions do not depend on any global variables and are therefore easy to modify without the need to look at the whole program. Using functions to structure the code makes it also more readable and easier to use, thus fulfilling requirement b). The overarching structure of the program was based on the computer vision pipeline design pattern introduced in section 2.3. The data acquisition in this case can be done by using the download scripts for the calibrated FFIs from the Mikulski Archive for Space Telescopes. The pre-processing steps consist of background correction (requirement 1) and image alignment (2). The creation of a reference image (3) and the image subtraction (4) could also be interpreted as part of the pre-processing. However, since the execution order is important for these steps and they can be seen as independent tasks, those were defined as additional steps in the pipeline. The next step in the computer vision pipeline is selecting the areas of interest, which in this case are the variable sources (5), for which light curves are then extracted(6). After that, the next step would normally be the prediction or in this case the classification of the light curves. However, the classification is outside the scope of this work. Figure 4.2 shows the adapted pipeline structure, including the in- and outputs of each step. Between the steps the data is written to and read from the computer storage, making it possible to execute each step independently. The work of Oelkers et al.[26] presented in section 2.5 followed a similar structure, further supporting the validity of the pipeline approach in achieving the goals of this thesis.

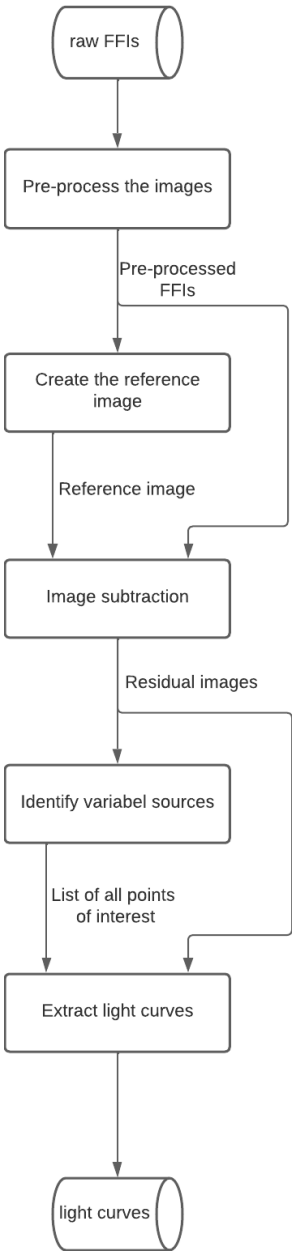

Figure 4.2: Overview of the program structure

The program steps and how they are implemented is explained in detail in chapter 5.

4.3 Data format

Resulting from the program architecture, the program has several data products, all of which are stored in the FITS file format, which is described in more detail in section 2.4.3. This format was chosen because it is the format in which FFIs are available for download, as well as it being the standard data format in modern astrophysics. Additionally, it is memory efficient and allows for easy processing due to the available Python libraries. Since it is the standard format in the field, it is also compatible with commonly used tools, thus fulfilling requirement d). The different data products and their size per file are listed in table 4.1.

Data product	**Size per file**
raw FFIs	33.9 MB
pre-processed FFIs	32.0 MB
reference image	32.0 MB
residual images	32.0 MB
light curves	30.9 kB

Table 4.1: Data products in the FITS file format

Although all files have the same format, they vary in their composition. The structure of the files can be seen in Figure 4.3. The pre-processed FFIs, the reference image and the residual images have a similar structure, with only minor differences in their header information. Therefore the figure only displays the structure of the pre-processed images.

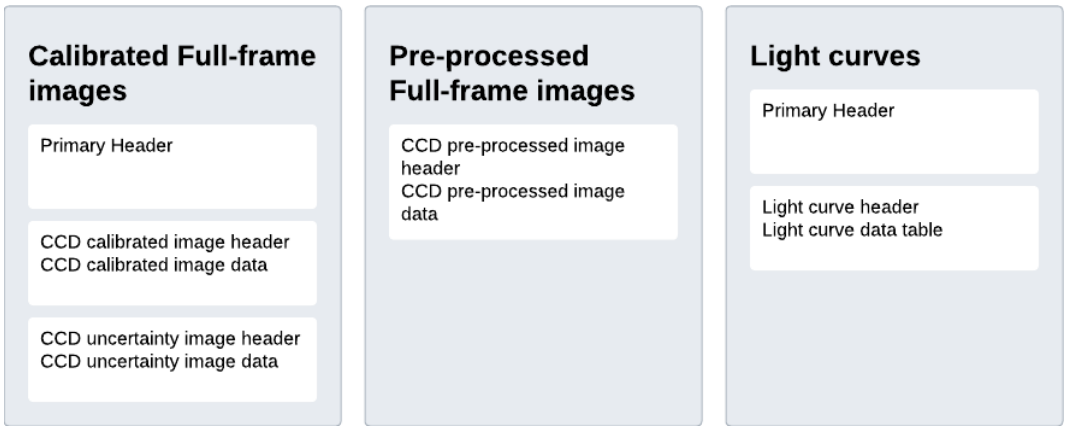

Figure 4.3: FITS file composition of the different data products

From the calibrated FFIs only the calibrated image header and data are needed. Those are modified in the pre-processing step and saved to a new FITS file, resulting in the file composition visible in the middle tile of figure 4.3. The data saved in those files looks very similar to the calibrated image data and consists of a 2-dimensional array of floating point values, the only difference being that the data array of the pre-processed images is slightly smaller. As described in section 2.4.3, the calibrated images contain some pixels on the image's edges that do not hold valuable information for this application. Therefore, those are removed during the pre-processing, resulting in an array size of 2048x2048 pixels for all other image data.
The light curves are stored as tables, with columns for the time of measurement and the measured brightness in electrons per second. This data can be plotted to show changes in brightness over time. An example light curve plot is shown in figure 4.4, with the x-axis showing the time and the y-axis showing the observed brightness.

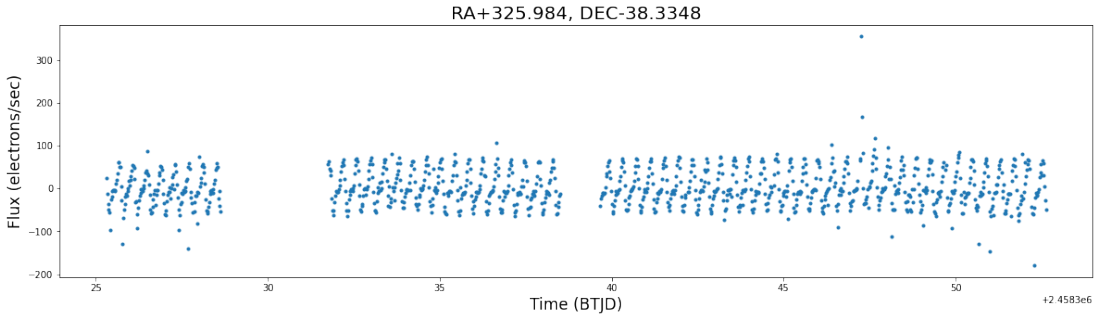

Figure 4.4: Example light curve extracted from the TESSs FFIs. The target is CRTS J214355.9-382010, a rotationally variable star.

The program also has the option to generate a list of on-sky coordinates of the extracted light curves in the form of a CSV file. This list can be used for cross-matching the sources identified by the program with other catalogues.

4.4 Validation

The aim of the validation is to assess whether the program satisfies the requirements defined in section 3.2. To evaluate the background correction, a comparison between the Oelkers et al. method[26] and the TESS Backdrop Python package[32], based on their run time and their ability to remove scattered light and other systematic errors, is made. The results of both methods are visually compared for an image with visible contamination by scattered light. In addition, the standard deviation of light curves obtained using each method is analysed

as visible patterns in the standard deviation plotted against the on-sky location can indicate systematic errors.

To verify image alignment accuracy, the on-sky location variation of a selected pixel over time is compared before and after aligning the images. The program's ability to identify variable sources is evaluated using known eclipsing binaries from the TESS eclipsing binaries catalogue[34] within a data set's field of view to confirm the program's capability of detecting them. The light curves extracted by the program for these known objects are compared with the light curves extracted for the same objects by the TESS Quick-look pipeline[31] to determine the accuracy of the program's light curves.

The run time of the program to process the data of a single CCD in one observing sector is measured to assess whether it is sufficiently fast to process large amounts of data, such as the entire TESS data catalogue.

One potential limitation of this validation is that it only examines the performance of the program on a single data set. It is possible that other data sets behave differently, producing different results. It is important to acknowledge the potential impact of data set specific factors on the performance of the program.

5 Implementation

This chapter presents the implementation details of the program architecture outlined in section 4.2. The objective of this chapter is to provide a comprehensive explanation of the techniques and algorithms used to achieve the desired results.

5.1 Pre-processing

This step aims to extract relevant information from the raw data and remove image artifacts, as described in Section 2.4.3. This step also filters out files that miss essential information regarding the WCS or have data quality issues flagged in the image header.
The pre-processing step has to iterate over all FFIs of a CCD. The program includes a function to check if parts of the files have already been processed, which enables the user to interrupt and restart the program if needed, without losing progress. When the program starts, it generates a list of all FITS files in the input folder and removes files that have already been pre-processed by checking the contents of the output folder. It removes the file names that are in both folders from the list of files to process.
As defined in the program design in section 4.2, the pre-processing step also includes aligning the image and removing the background and other image artifacts. After those steps have been applied to an image, it is indicated by the "ALIGN" and "BKSUB" fields in the image header.

5.1.1 Using the World Coordinate System for image alignment

Each TESS FFI has a World Coordinate System (WCS) defined in the header, which can be used to align the images by shifting them until the pixel coordinates, converted to celestial coordinates, match as closely as possible. This is done by the *hcongrid* function from the python *FITS_tools* package. To align the images, one image is chosen as a reference and the other images are aligned to it. Since the pointing of the TESS satellite varies very little, it is not critically important which image is chosen as the reference. For simplicity, the first image of each data set is used as the reference.

5.1.2 Background removal

To remove the background from the images, including scattered light and other sources of contamination, two different Python routines that have been developed by other researchers and have already proven to be successful, were used. The first routine is the one by Oelkers et al.[26], the second one is the TESS Backdrop package by Hedges[32], both of which were outlined in Section 2.5. Both routines were implemented in the software and their performance compared in terms of speed and effectiveness. The user is able to select which routine is used by setting a parameter.

5.2 Reference image

The reference image is an essential component of image subtraction, as it serves as a baseline for comparison to each observation. The goal of image subtraction is to isolate transients and other variable objects in the residual images by subtracting a reference image that does not contain these objects.
To generate the reference image, a median combination technique similar to the one applied by Oelkers et al.[26], which was described in section 2.5, was employed. This method involves calculating the median value of each pixel across the input images and using these values to create a new image. One of the key benefits of median combination is its ability to effectively remove outliers from the data.
Most stars emit a relatively constant flux of light, making their pixel values consistent across all input images. Subtracting the median value effectively removes these constant sources, leaving only noise at the same pixels, which is of the same order as that of the background. On the other hand, transients and variable stars, whose flux differs from the median in most images, remain in the residuals and thus are detected by the algorithm.
To conserve computational resources, this implementation includes the option to only use a subset of the images for calculating the median. For example, the user can choose to use every second image.

5.3 Image subtraction

The image subtraction algorithm developed by Oelkers et al.[26] was employed. Their implementation, written in C, was called from the python program

5.4 Identifying variable sources

The aim of this step is to compile a comprehensive list of coordinates where variability has been detected. This involves two sub-steps.
Firstly, the residual images are iterated over and a source detection algorithm is applied to identify any sources present in each image. This generates a separate list of sources for each residual.
Secondly, the individual lists are merged to create a final list of sources. Since most sources are likely to appear in multiple residuals, the merging process cross-references all lists to identify common sources and eliminates duplicates, resulting in a final, comprehensive list of variable sources.

5.4.1 Source detection

To identify sources that persist in the residual images and may indicate the presence of transients or variable stars, the star-finding routine described in the DAOphot paper[24], discussed in Section 2.5, was used. This routine is based on identifying local maxima in the image.
It is important to note that stars in normal images have higher pixel values than the background. However, if a source in one of the FFIs is fainter than in the reference image, it will appear as a darker spot in the residual image. Nonetheless, the source should still maintain its shape. The algorithm detects it by applying the star-finding routine to the inverted residual image.
The source detection algorithm is applied to each residual image to identify all persistent sources.

5.4.2 Cross-referencing of sources between images

To determine whether two observations from different images correspond to the same object, an algorithm based on proximity was implemented. The algorithm utilises a user-defined threshold distance below which two observations are considered to belong to the same object. To optimise the efficiency of the algorithm and avoid the need to compare every source in one image against every source in every other image, Python's built-in list sorting capabilities were utilised. By sorting the lists of sources in advance, the algorithm only needs to consider a small section of the already identified sources. Specifically, the list of sources for the first image is used as a starting point, and then the sources from every subsequent image are compared against this list. If a source is not already in the reference list, it is inserted. This process is

repeated for each image, discarding sources that have a match in the reference list. This results in a final, comprehensive list of variable sources.

5.5 Extracting the light curves

To extract the light curves of the variable sources, the program measures the flux of each source in each residual image and links this information with the corresponding image timestamp. This produces a table with columns for time, flux, and flux error for each source, which can be saved as a FITS file for further analysis.
Aperture photometry was utilised to measure the flux of the sources in the residual images. This method involves summing the flux within a fixed-size circular aperture centred on the source, as explained in Section 2.2. Several factors motivated the selection of this method. Firstly, it is relatively insensitive to variations in the point spread function of the image, which can change substantially over the camera's field-of-view, as detailed in the TESS instrument Handbook[14]. The point spread function is the impulse response of an optical instrument[35]. Secondly, it is a simple and easy-to-implement method, well-suited for measuring the flux of a large number of sources. Furthermore, it is computationally efficient, making it an appropriate choice for this program, which requires measuring the flux of numerous sources.
The error in the measured flux is calculated as the square root of the measured flux, reflecting the uncertainty in the measurement due to counting statistics arising from the Poisson distribution of the measured photons for the CCD[23].
The extracted light curves are saved as FITS files using the Lightkurve Python package [36]. This package has the benefit of producing standardised FITS files with headers consistent with those created by other researchers, thus fulfilling requirement d).

6 Evaluation

This chapter aims to assess the performance of the developed algorithm in terms of run time, image quality and its ability to detect known eclipsing binaries. Additionally, the program's potential for detecting moving objects in the data is demonstrated.

6.1 Run time

In this section, an evaluation of the different program sections in terms of their run time is made. Table 6.1 presents the measured times for each of the program's parts.

Program section	**Run time**
Pre-processing (TESS Backdrop)	01:08:46
Pre-processing (Oelkers et al.)	07:02:16
Creating the reference image	00:06:20
Image subtraction	02:33:51
Source detection	00:23:12
Cross-matching	00:01:05
Light curve extraction	00:38:46

Table 6.1: Run time of the program sections for processing 1095 FFIs from sector 1, camera 1, CCD 1, with a detection threshold of 50 and 24 202 extracted light curves.

It is evident that the Tess Backdrop package is considerably faster in pre-processing the FFIs, making it the preferred method to comply with requirement c).
The most time-consuming steps are the pre-processing and image subtraction, both of which could benefit from parallelisation in the future, given that they process each image individually. While the program's defined run-time requirement c) is generally met, there is room for improvement in terms of optimising the time-intensive steps.
For this data set, the program's overall run time amounts to 04h 54m 16sec. Assuming a similar run time for other data sets, it would take approximately three days to process an entire sector

on a single computer.
It is worth noting that this program is a significant improvement in terms of run time compared to the Oelkers et al. approach, as its overall run time is faster than the time required for the pre-processing step of that program alone. Nonetheless, further optimisation of the program's time-intensive steps would make the program even more efficient. However, it is also noteworthy that most users of the code will have access to clusters with typically tens of thousands of cores, so a simple parallelisation will reduce run time of the entire TESS data set to less than a day. Except for the pre-processing, all other program parts run on a single CPU-core.

6.2 Pre-processed images

In this section, the results of the evaluation of the image alignment and background removal process are presented and analysed.

6.2.1 Image alignment

The accuracy of the program's image alignment was evaluated by selecting a single pixel and converting its pixel coordinates to sky coordinates in right ascension and declination using the WCS of both the raw and aligned images. The on-sky separation between the selected pixel in each image and the corresponding pixel in the first image of the data set was calculated and can be seen in Figure 6.1.

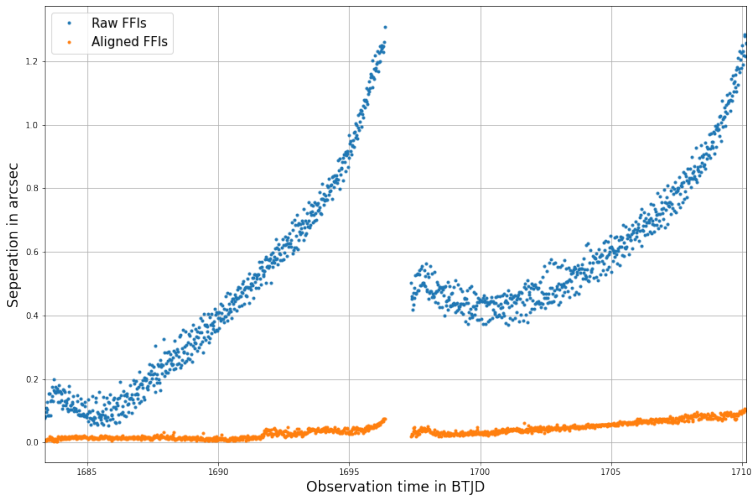

Figure 6.1: On-sky separation of pixel (170,420) to the first image for the raw and the aligned FFIs in sector 14, camera 2, CCD 1

The results presented in figure 6.1 show a time of approximately one month. As seen in the graph, the on-sky separation of the raw images varies between 0.1 and 1.3 arcseconds, which corresponds to around $\frac{1}{200}$ to $\frac{1}{15}$ of a pixel. The on-sky separation is slightly better than the assumed offset of up to 2 arcseconds reported by Fausnaugh et al.[37].
The graph also shows the variation of the telescope pointing over the course of an orbit. The data shows two orbits with a gap in between, which can be explained by the down-link to Earth which is done once every orbit as described in section 2.4.1.
The on-sky separation for the aligned images varies between 0 and 0.1 arcseconds, demonstrating a significant improvement in image alignment. The small offset between images remaining is only about $\frac{1}{400}$ of a pixel, which meets requirement 2 for precise image alignment.
In conclusion, the results suggest that the program's image alignment is very accurate and meets the defined requirement. The program's ability to improve the alignment of raw images is a critical step in processing the data to ensure a clean image subtraction later on.
The light curves produced by the standard TESS pipeline for pre-defined targets are usually prone to quasi-periodic systematics corresponding to the orbital period of the satellite. These results show, that, even for the pre-defined TESS targets, this pipeline yields more reliable results.

6.2.2 Background correction

The goal of the background correction process was to eliminate scattered light and other systematic effects from the images. To evaluate the effectiveness of the applied methods, a comparison is made between the raw and the background-corrected images using the two different methods that were implemented. Figure 6.2 shows the raw image and the background corrected ones.

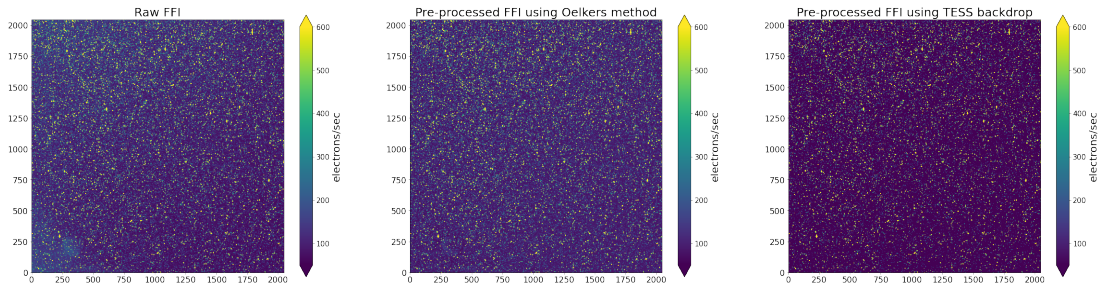

Figure 6.2: From left to right: FFI without background correction, FFI with background correction using the method by Oelkers et al.[26], FFI with background correction using TESS backdrop[32]

The raw FFI on the left in figure 6.2 shows some bright spots caused by scattered light, particularly in the bottom-left corner. Both applied background correction methods effectively remove those spots to a degree where they are no longer visible by naked eye. The raw image also features a slightly brighter background in the top-left corner, whereas the background-corrected images appear to have a more uniform background throughout. However, the correction method by Oelkers et al.[26] evens out the background without notably changing its overall brightness, while the correction using Tess Backdrop results in a notably darker background.

This observation is confirmed by examining the histograms in figure 6.3. The histograms demonstrate that the raw image and the image processed with the Oelkers method have their highest peak at around 80 electrons/sec, whereas the peak for the TESS Backdrop pre-processed image is at zero. Both pre-processing methods result in a sharper peak, showing that both have made the background more uniform.

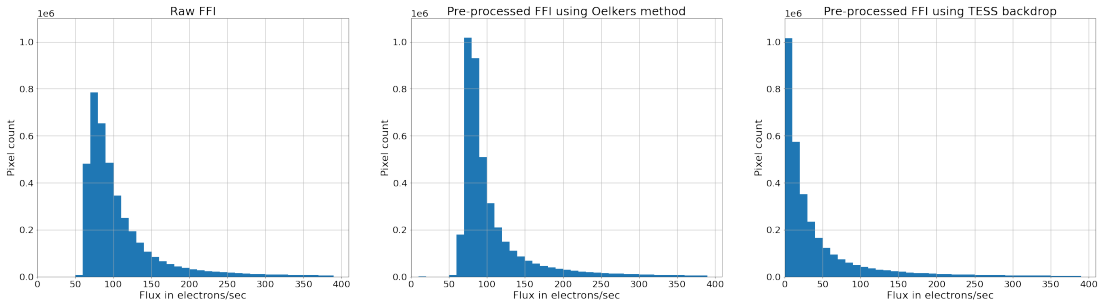

Figure 6.3: Histograms of the flux per pixel for the FFIs from figure 6.2

For the image subtraction to work properly, it is important that the background remains approximately constant throughout all images of a data set, regardless of its brightness. To assess, if this is the case, the resulting residuals were analysed. The following figure 6.4 shows the residual images resulting from the two background-corrected images from figure 6.2.

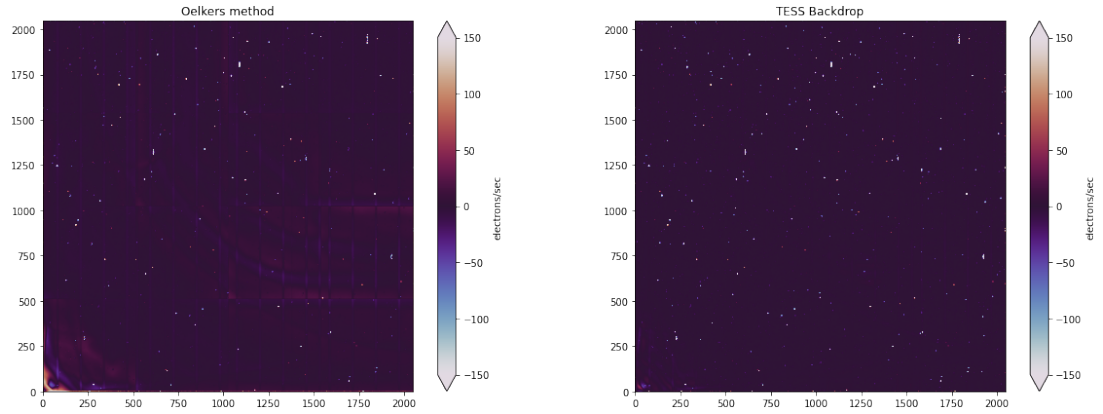

Figure 6.4: Residual images

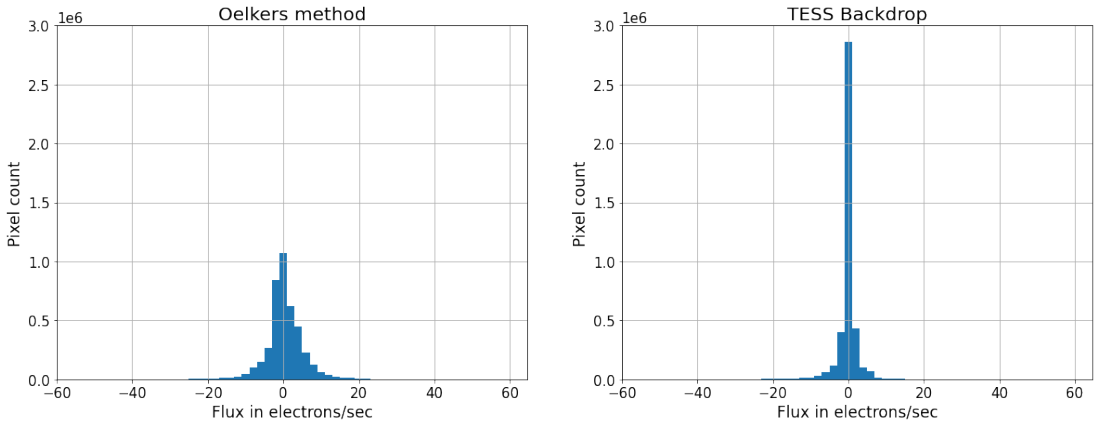

Figure 6.5: Histogram of the flux in the residual images

The majority of pixels in both images shown in figure 6.4 are close to zero, indicating that both correction methods have maintained a constant background over time. The histograms in figure 6.5 also show that most pixels have a flux close to zero. However, the residual image from the Oelkers correction method shows some contamination, such as remnants of brighter sky patches in the bottom left corner and stripes caused by the CCD readout. The faint outlines of the boxes used for background correction, as mentioned in section 2.5, are also noticeable, especially in the centre-right part of the image. This also shows in the histogram in form of the lower peak at zero for the image obtained by the Oelkers method.
In contrast, the residual image produced by the Tess Backdrop correction method shows almost no visible defects, with only a small remnant of scattered light visible in the bottom left corner. Overall, the correction method using Tess Backdrop results in a cleaner residual image with less visible contamination by unwanted effects. However, the Oelkers method still provides acceptable results with a consistent background close to zero.
The residual images also show that requirement 4 was fulfilled since a working image subtraction was applied.
To assess the impact of these imperfections on the resulting light curves, the standard deviation of each light curve is evaluated. Ideally, the standard deviation of neighbouring light curves would be uncorrelated. The two plots in figure 6.6 show the standard deviation and on-sky position of each light curve. The plot on the left has fewer data points, as fewer sources were identified in the images processed using the Oelkers et al. method.

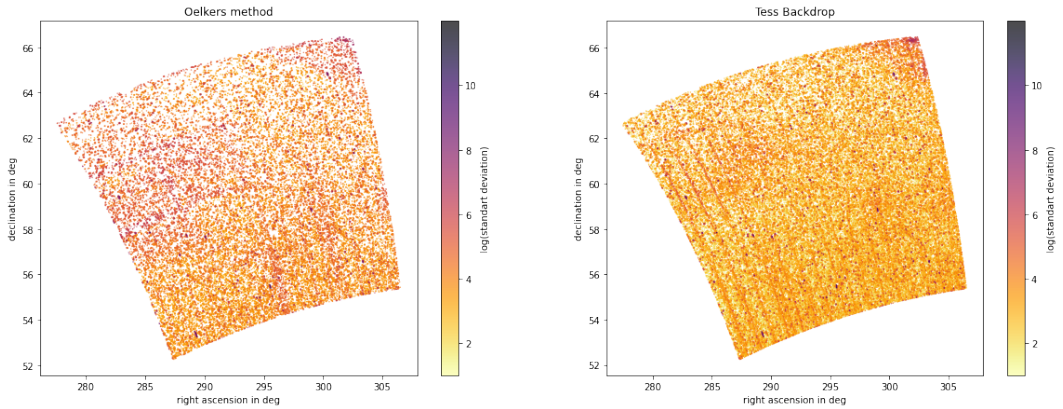

Figure 6.6: Standard deviation of the light curves

The plots indicate that background effects in residual images do indeed affect the light curves. Both plots show a higher standard deviation of the light curves for the area most strongly affected by scattered light. In the left plot, the outlines of the boxes that were already visible in figure 6.4 are still recognisable. The vertical lines, which in figure 6.4 were only visible in the left plot, are only faintly visible here for the light curves from the images that were pre-processed using the Oelkers method. In the plot on the right for the light curves extracted from images that were pre-processed using Tess Backdrop, these vertical lines are more distinctive, indicating that those light curves are stronger affected by this effect.
Overall the light curves for both methods show some level of contamination by various effects. However, for the images pre-processed using the Oelkers method, larger areas of the images are affected and the most prominent effect is introduced by the background correction itself due to the division of the images into smaller sections. For the images that were pre-processed using Tess Backdrop a much smaller area is affected.
Overall both methods mostly satisfy requirement 1 to remove background effects. However, the results of the TESS Backdrop package are notably better. Considering that TESS backdrop is also faster, it constitutes the overall better option for this application.

6.3 Recovering known eclipsing binaries

In this section, the program's ability to recover known eclipsing binaries from the TESS FFIs is demonstrated. For this evaluation, the FFIs from the first sector, camera 1, CCD 1 were used. The eclipsing binaries were published in the TESS Eclipsing Binaries Catalogue[34].

6.3.1 Identifying the sources

The program identified a total of 24 202 sources in the selected FFIs, using a threshold of 50 for the absolute image value above which a source is detected. The cross-matching between the sources identified in the individual images was performed using a matching radius of 3 pixels. Table 6.2 shows the coordinates of the eclipsing binaries from the catalogue and the closest source identified by the program.

Known objects		Found sources		
RA in deg	DEC in deg	RA in deg	DEC in deg	Separation in arcsec
314.50875	-46.066771	314.50883	-46.06775	3.54285
312.97905	-40.784704	312.97879	-40.78343	4.61464
318.73236	-43.395447	318.73161	-43.39539	1.95100
316.79610	-39.766323	316.79507	-39.76729	4.52045
316.21450	-46.326206	316.21481	-46.32670	1.96344
318.58049	-42.798657	318.59235	-42.80141	32.8689
325.99416	-38.017427	326.00656	-38.02134	37.8944
322.95322	-45.045093	322.95378	-45.04595	3.42482
320.91898	-39.778801	320.92278	-39.78246	16.8758
323.03897	-34.714271	323.03873	-34.70916	18.4131
312.56828	-42.217943	312.56885	-42.21746	2.29662
313.67157	-45.730706	313.67143	-45.72850	7.92903
316.49578	-36.259440	316.49432	-36.25978	4.42028
314.69595	-42.617883	314.69631	-42.61777	1.04369
327.97270	-42.373202	327.97403	-42.37200	5.57981
322.07546	-44.154830	322.07535	-44.15538	1.99962
320.63958	-46.460585	320.64044	-46.45778	10.3209
313.99231	-36.448604	313.99129	-36.44886	3.11072
321.07188	-44.917582	321.07344	-44.91736	4.05567
317.64281	-40.523895	317.64315	-40.52233	5.69030
317.75599	-40.539302	317.75362	-40.54418	18.7208
313.26148	-38.594876	313.26524	-38.59516	10.6338

Table 6.2: Coordinates of known eclipsing binaries and the closest sources identified in the residual images and the on-sky separation between them

Table 6.2 shows that it was possible to recover all 22 known eclipsing binaries. The largest separation between the position given in the catalogue and the position identified by the program is about 37.9 arcsec, which is less than two pixels in the images (1 pixel spans about 21 arcsec). Therefore, this difference can most likely be explained by the resolution of the TESS images. However, all but two binaries are detected with offsets well below the TESS pixel size. The fact that all eclipsing binaries were identified with reasonable accuracy in position confirms the effectiveness of the identification process applied in this thesis, satisfying the requirements 5 and e).
However, it is important to note that the program also identified a lot of false positives, thus producing many light curves that seem to show only noise. It was decided that a high number of false positives is preferable over potentially missing true positives, since a post-processing step to remove false positives could be added in the future and the number of extracted light curves is still manageable.

6.3.2 Light curves

The ability of the algorithm to extract light curves for the identified sources was evaluated by comparing them with those obtained from the TESS Quick Look Pipeline, presented in section 2.5, for known eclipsing binaries. Since these objects have a Tess Input Catalogue ID, their light curves are readily available in the TESS Quick Look archive.
For this comparison, two of the eclipsing binaries from the TESS Eclipsing Binaries Catalogue were selected. Figure 6.7 shows the light curves extracted using this program and the TESS Quick-look Pipeline.

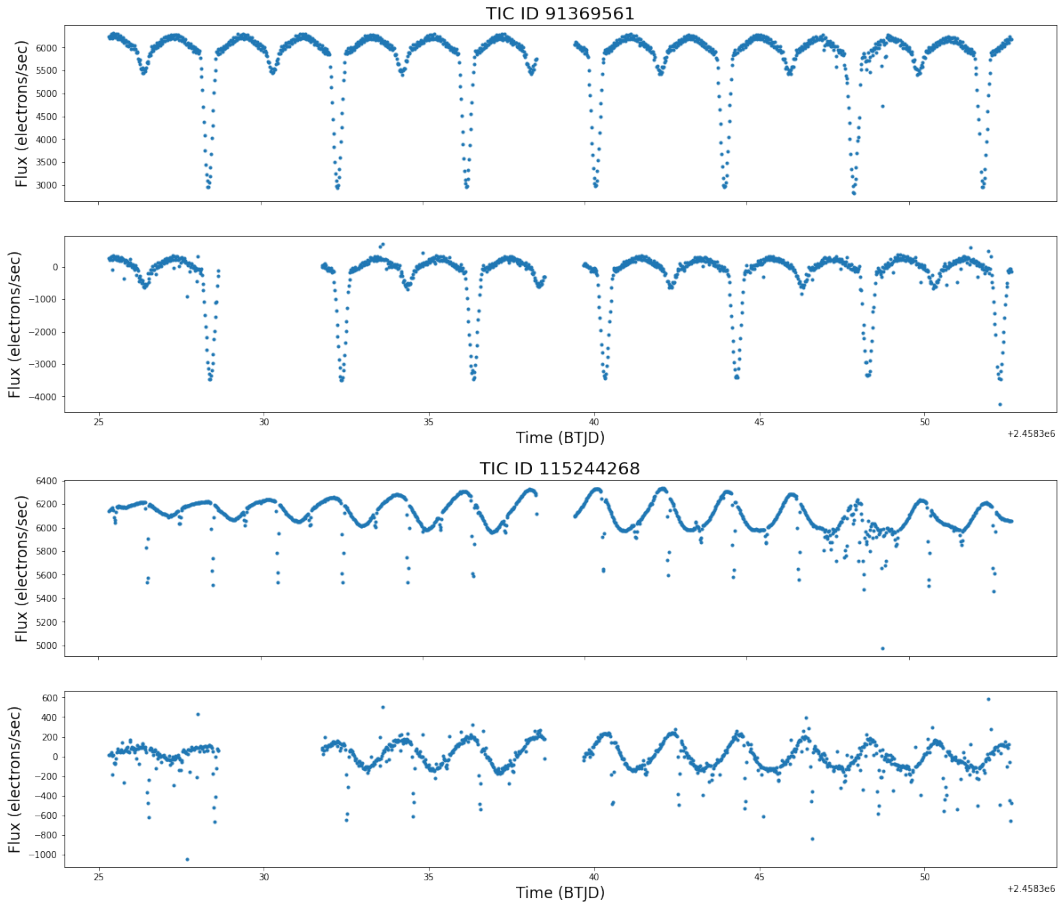

Figure 6.7: Comparison of the light curves obtained for the eclipsing binaries TIC 91369561 and TIC 115244268 with the TESS Quick Look Pipeline (top) and my program(bottom).

As can be seen from figure 6.7, the light curves obtained from both methods exhibit the same shape, confirming the program's ability to extract accurate light curves from the TESS FFIs. However, there are differences in the flux values, because this program measures the differenced flux while the Quick Look Pipeline measures the absolute flux. Additionally, the light curves from this program show some outliers that are not present in the Quick Look light curves because the Quick Look Pipeline includes a post-processing step that removes outliers and performs detrending. To address this issue, a post-processing step, similar to the one implemented in the Quick-look Pipeline, could be added in the future. For now, it is more important to keep outliers in the data products, since some of them may actually be caused by real physical phenomena.
Furthermore, the light curves extracted using this program exhibit a larger gap in the data.

This gap results from a data quality flag set in multiple images of the data set, indicating that the data contained in these frames are unreliable. Those images were excluded in this program. The gap in the middle of all light curves results from the time when TESS was down-linking data to Earth and therefore was not observing.
Overall, the comparison with the TESS Quick Look Pipeline confirms the program's ability to extract accurate light curves for variable sources in the TESS FFIs, satisfying requirements 6 and e).

6.4 Identification of Moving Objects

When inspecting the on-sky locations of the sources identified by the program, it became apparent that several sources were aligned along almost straight lines, as can be seen in Figure 6.8.

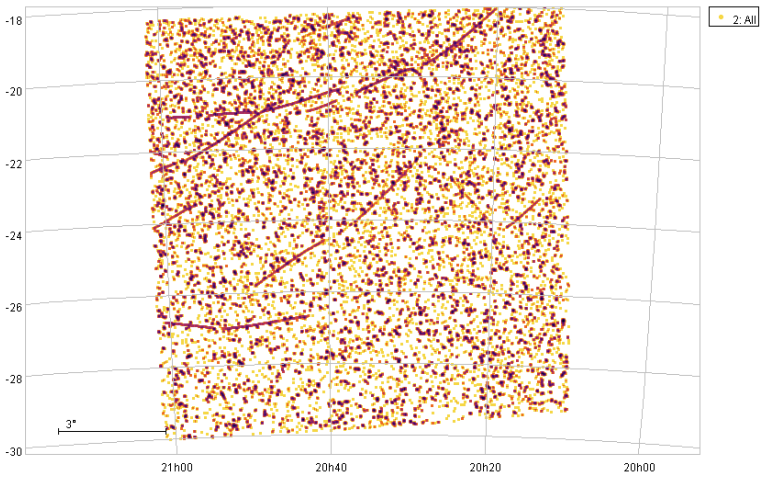

Figure 6.8: Location of the identified sources in the sky. A darker colour indicates a higher density of identified sources in the area

Further examination of the light curves along these lines revealed that they all share the same shape of a flux of zero throughout most of the observation time except for a single high peak. The time of the peak is different in each light curve. Upon examination of several light curves along the same line, it was observed that the peaks in brightness appeared to move through the image, with the width and height of the peaks remaining mostly constant. This behaviour indicates that the light curves corresponded to moving objects passing through the field of view.

By cross-referencing one set of those observations with the position of known Solar System objects, it was identified as the asteroid Urhixidur, confirming that those observations belong to moving objects. Figure 6.9 shows three exemplary light curves of such an object.

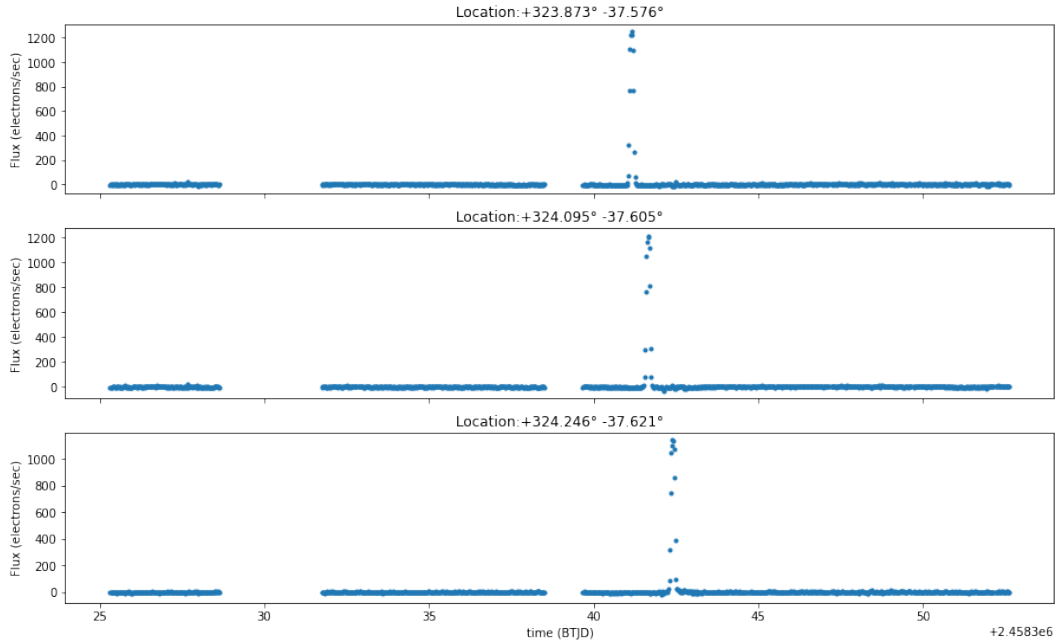

Figure 6.9: Light curves caused by a moving object passing through TESS' field of view

Overall, the identification of moving objects demonstrates an additional capability of the program beyond its primary purpose of identifying variable stars and transients. This is an important result for the broader scientific community, as it provides a means of discovering and studying previously unknown moving objects in the TESS data.
The fact that the positions identified by the program are sufficiently accurate to relate the observations to known objects such as Urhixidur again shows the accuracy of the program (requirement e)).

7 Conclusion

7.1 Summary

In this thesis, a program for the automated identification of candidates for variable stars and transients in the TESS FFIs was presented. The program includes routines for image pre-processing, reference frame construction, image subtraction, variable source identification, and light curve extraction.

One significant finding of this study is that the TESS backdrop python package [32] outperforms the method implemented by Oelkers et al. [26] for applying background correction to the FFIs in terms of speed and quality. However, residual images and light curves were still affected by scattered light and other effects to some extent.

It was shown that the star finding routine by Stetson [24] can be used for source detection on residual images, enabling the program to identify candidates for variable stars and transients in the data. Using this approach, it was possible to recover a set of known eclipsing binaries from the data, validating the effectiveness of the program. It is worth noting that the program identifies a significant number of false positives. However, prioritising the identification of all true variable sources was preferred over reducing the number of false positives.

Comparing the light curves of the identified objects to the TESS Quick-Look Pipeline data confirmed the reliability of the photometry used in this thesis, as the characteristics and shape of the extracted light curves matched the known behaviour of the objects.

Furthermore, the analysis of the resulting light curves showed that the program can also be used to find moving objects in the data, which was not originally intended. This finding opens up new possibilities for using the program to identify other types of objects in the TESS data.

Overall, the program presented in this thesis represents a significant step forward in the automated identification of variable stars and transients in the TESS data. The program's ability to identify a large number of variable sources and its potential for uncovering new discoveries make it a valuable tool for astronomers.

7.2 Future work

While the program presented in this thesis is a step forward, there are still several avenues for future work that could further improve its capabilities and broaden its impact.
One major area for future work is the classification of the light curves. Proper labelling of the data and removing false positives from the light curves are essential steps to creating a comprehensive catalogue that can be made available to the astrophysical community. This work has the potential to create a lot of scientific value, enabling a lot more research on all kinds of variable objects.
Another important direction for future work is applying the program to analyse the entire TESS data set. To process the data efficiently, future work may involve optimising for speed or parallelising the computation. Additionally, implementing post-processing techniques such as outlier rejection to the light curves may improve their quality and facilitate the classification of the objects or events they represent.

Bibliography

[1] P. T. Boyd, *Tess science support center*, Last accessed 17 February 2023, 2022. [Online]. Available: https://heasarc.gsfc.nasa.gov/docs/tess/.

[2] A. H. EOW, *Variables: What are they and why observe them?*, Last accessed 22. February 2023, 2022. [Online]. Available: https://www.aavso.org/variables-what-are-they-why-observe-them.

[3] J. T. Jason Hinkle, *Guide to transient astronomy*, Last accessed 22. February 2023, 2022. [Online]. Available: https://astrobites.org/2022/10/30/guide-to-transient-astronomy/.

[4] *Tess bulk downloads*, Last accessed 28. February 2023, Feb. 2023. [Online]. Available: https://archive.stsci.edu/tess/bulk_downloads/bulk_downloads_ffi-tp-lc-dv.html.

[5] S. staff, *Types of variable stars: Cepheid, pulsating and cataclysmic*, Last accessed 15. February 2023, 2015. [Online]. Available: https://www.space.com/15396-variable-stars.html.

[6] *Gnu astronomy utilities - 8.1.3 flux brightness and magnitude*, Last accessed 7. February 2023, 2020. [Online]. Available: https://www.gnu.org/software/gnuastro/manual/html_node/Flux-Brightness-and-magnitude.html.

[7] L. Kakanuru, *Computer vision: A key concept to solve many problems related to image data*, Last accessed 8. February 2023, 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/11/computer-vision-a-key-concept-to-solve-many-problems-related-to-image-data/.

[8] M. Elgendy, *Computer vision pipeline, part 1: The big picture*, Last accessed 8. February 2023, 2019. [Online]. Available: https://freecontent.manning.com/computer-vision-pipeline-part-1-the-big-picture/.

[9] *Tess mit website*, Last accessed 17 February 2023, 2022. [Online]. Available: https://tess.mit.edu/.

[10] N. E. S. Institute, *Nasa exoplanet archive*, Last accessed 16. February 2023, 2023. [Online]. Available: https://exoplanetarchive.ipac.caltech.edu/.

[11] G. R. Ricker, J. N. Winn, R. Vanderspek, *et al.*, "Transiting Exoplanet Survey Satellite (TESS)", in *Space Telescopes and Instrumentation 2014: Optical, Infrared, and Millimeter Wave*, J. Oschmann Jacobus M., M. Clampin, G. G. Fazio, and H. A. MacEwen, Eds., ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 9143, Aug. 2014, 914320, p. 914 320. DOI: 10.1117/12.2063489. arXiv: 1406.0151 [astro-ph.EP].

[12] J. W. Gangestad, G. A. Henning, R. R. Persinger, and G. R. Ricker, "A high earth, lunar resonant orbit for lower cost space science missions", *arXiv preprint arXiv:1306.5333*, 2013.

[13] A. E. L. Scott Wiessinger, *Goddard media studios, tess social media products*, Last accessed 20. January 2022, 2018. [Online]. Available: https://svs.gsfc.nasa.gov/12886.

[14] R. Vanderspek, J. Doty, M. Fausnaugh, *et al.*, *Tess instrument handbook*, 2018.

[15] Wikipedia, *Ccd-sensor — wikipedia, die freie enzyklopädie*, [Online; Stand 28. Januar 2023], 2023. [Online]. Available: https://de.wikipedia.org/w/index.php?title=CCD-Sensor&oldid=229485165.

[16] J. D. Twicken, D. A. Caldwell, J. M. Jenkins, *et al.*, "Tess science data products description document: Exp-tess-arc-icd-0014 rev f", 2020.

[17] A. Rajpurohit, F. Allard, S. Rajpurohit, *et al.*, "Exploring the stellar properties of m dwarfs with high-resolution spectroscopy from the optical to the near-infrared", *Astronomy & Astrophysics*, vol. 620, A180, 2018.

[18] W. Schmidt and M. Völschow, "Numerical python in astronomy and astrophysics: A practical guide to astrophysical problem solving", in Springer, 2021, p. 188.

[19] L. CHIAPPETTI, M. J. Currie, S. Allen, *et al.*, "Definition of the flexible image transport system (fits) the fits standard version 4.0: Updated 2016 july 22 by the iaufwg original document publication date: 2016 july 22 language-edited document publication date: 2018 august 13", 2018.

[20] Astropy Collaboration, A. M. Price-Whelan, P. L. Lim, *et al.*, "The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package", *apj*, vol. 935, no. 2, 167, p. 167, Aug. 2022. DOI: 10.3847/1538-4357/ac7c74. arXiv: 2206.14220 [astro-ph.IM].

[21] P. Davis, *Nasa solar system exploration: The celestial sphere*, Last accessed 25. January 2023, 2023. [Online]. Available: https://solarsystem.nasa.gov/basics/chapter2-2/.

[22] E. W. Greisen and M. R. Calabretta, "Representations of world coordinates in fits", *Astronomy & Astrophysics*, vol. 395, no. 3, pp. 1061–1075, 2002.

[23] S. B. Howell, *Handbook of CCD astronomy*. Cambridge University Press, 2006, vol. 5, pp. 73–74.

[24] P. B. Stetson, "Daophot: A computer program for crowded-field stellar photometry", *Publications of the Astronomical Society of the Pacific*, vol. 99, no. 613, p. 191, Mar. 1987. DOI: 10.1086/131977. [Online]. Available: https://dx.doi.org/10.1086/131977.

[25] C. Alard and R. H. Lupton, "A method for optimal image subtraction", *The Astrophysical Journal*, vol. 503, no. 1, p. 325, 1998.

[26] R. J. Oelkers and K. G. Stassun, "Precision light curves from tess full-frame images: A different imaging approach", *The Astronomical Journal*, vol. 156, no. 3, p. 132, 2018.

[27] C. Alard, "Image subtraction using a space-varying kernel", *Astronomy and Astrophysics Supplement Series*, vol. 144, no. 2, pp. 363–370, 2000.

[28] D. Bramich, "A new algorithm for difference image analysis", *Monthly Notices of the Royal Astronomical Society: Letters*, vol. 386, no. 1, pp. L77–L81, 2008.

[29] J. P. Miller, C. Pennypacker, and G. L. White, "Optimal image subtraction method: Summary derivations, applications, and publicly shared application using idl", *Publications of the Astronomical Society of the Pacific*, vol. 120, no. 866, p. 449, 2008.

[30] R. J. Oelkers, L. M. Macri, L. Wang, *et al.*, "Difference image analysis of defocused observations with cstar", *The Astronomical Journal*, vol. 149, no. 2, p. 50, 2015.

[31] C. X. Huang, A. Vanderburg, A. Pál, *et al.*, "Photometry of 10 million stars from the first two years of tess full frame images", *arXiv preprint arXiv:2011.06459*, 2020.

[32] C. Hedges, *Tess-backdrop*, Last accessed 24 January 2023, 2021. [Online]. Available: https://ssdatalab.github.io/tess-backdrop/.

[33] M. Mogyorosi, *Pass by reference in python: Background and best practices*, Last accessed February 2023. [Online]. Available: https://realpython.com/python-pass-by-reference/.

[34] A. Prša, A. Kochoska, K. E. Conroy, *et al.*, "Tess eclipsing binary stars. i. short-cadence observations of 4584 eclipsing binaries in sectors 1-26", *ApJs*, vol. 258, no. 1, 16, p. 16, Jan. 2022. DOI: 10.3847/1538-4365/ac324a. arXiv: 2110.13382 [astro-ph.SR].

[35] cosmostat.org staff, *Psf estimation and images restoration*, Last accessed 28. February 2023. [Online]. Available: http://www.cosmostat.org/wl/psf-estimation-and-images-restoration/psf-modeling.

[36] J. V. d. M. Cardoso, C. Hedges, M. Gully-Santiago, *et al.*, "Lightkurve: Kepler and tess time series analysis in python", *Astrophysics Source Code Library*, ascl–1812, 2018.

[37] M. M. Fausnaugh, C. J. Burke, G. R. Ricker, and R. Vanderspek, "Calibrated full-frame images for the tess quick look pipeline", *Research Notes of the AAS*, vol. 4, no. 12, p. 251, Dec. 2020. DOI: 10.3847/2515-5172/abd63a. [Online]. Available: https://dx.doi.org/10.3847/2515-5172/abd63a.

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 3. März 2023 Elin Thomfohrde-Dammann