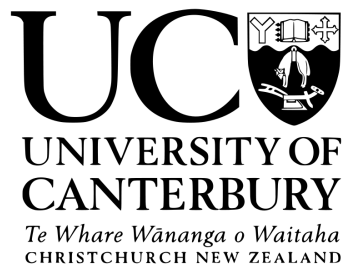


Department of Physics and Astronomy, University of Canterbury,
Private Bag 4800, Christchurch, New Zealand

The Detection of Faint Asteroids by the Shifting and Stacking of Difference Images

Edward Ashton



Masters of Science in Astronomy

2015

Supervisor: Dr. Michael Albrow

Contents

1	Introduction	1
1.1	What Are Asteroids?	1
1.1.1	Near Earth Asteroids (NEA)	1
1.2	The Importance of Finding NEAs	2
1.2.1	The Threat of Asteroid Impacts	2
1.2.2	Understanding the Early Solar System	3
1.2.3	Asteroid Mining	3
1.3	A History of NEA Discoveries	4
1.3.1	Early On	4
1.3.2	Spaceguard Survey	4
1.3.3	George E. Brown, Jr. Near-Earth Object Survey Act	5
1.3.4	Current Numbers	5
1.4	Surveys	6
1.4.1	Lincoln Near-Earth Asteroid Research (LINEAR)	6
1.4.2	Catalina Sky Survey (CSS)	6
1.4.3	Panoramic Survey Telescope and Rapid Response System (Pan-STARRS)	7
1.4.4	Asteroid Terrestrial-impact Last Alert System (ATLAS)	7
1.5	How to Detect Moving Objects	7
1.5.1	Detection of Faint Moving Objects	7
1.5.2	The Shift and Stack Method	8
1.5.3	Detection of Asteroids	9
1.6	Stacking Images	9
1.6.1	Stacking Difference Images	10
1.6.2	Weighted Averages	11
1.6.3	Optimally Stacking Difference Images	11
1.7	Angular Separation	12
1.8	Thesis Outline	13
2	Software	14
2.1	Difference Image Analysis	14
2.1.1	How to Produce a Difference Image	14
2.1.2	PYSIS3.1.2	15

2.1.3	Creating the Difference Images used by the Algorithm	16
2.1.4	Asteroids in Difference Images	17
2.2	Star Identification	17
2.3	Astrometry.net	18
3	Equipment	19
3.1	Telescope	19
3.2	Camera	19
4	The Algorithm	21
4.1	Overview	21
4.2	Files Needed to Run the Algorithm	23
4.3	Variables	23
4.3.1	Changeable	23
4.3.2	Fixed	24
4.4	Reusing Masks	24
4.5	Velocity Range Limits	25
4.5.1	Astrometry	25
4.5.2	Obtaining Velocity Range Limits	26
4.6	Loading the Difference Images	27
4.6.1	Creating the List	27
4.6.2	Loading the Difference Images and Removing the Infinite Border	29
4.7	Masking	29
4.7.1	Bright Star Mask	34
4.7.2	Saturated Star Mask	35
4.7.3	Poor Quality Section Mask	35
4.7.4	High Values Mask	39
4.8	Search for Asteroids	41
4.8.1	Search	43
4.8.2	Decreasing Velocity Range Limit	45
4.8.3	Increasing Velocity Range Limit	45
4.8.4	Asteroid Found	45
4.8.5	Resetting Velocity Range Limit	46
4.8.6	Signal Decrease of Possible Asteroid	46
4.8.7	Difference Image Cleaning	47
4.8.8	More Difference Image Cleaning	47
4.8.9	Function: Creating the Velocity Phase Space Matrix	47
4.8.10	Function: Origin Removal	49
4.8.11	Function: Asteroid Detection	50
4.8.12	Function: Asteroid Masking	51

4.8.13	Function: Asteroid Positions	51
4.8.14	Function: Saving Images	52
4.9	Output Files	54
4.9.1	Asteroid Positions	54
4.9.2	Masks	54
4.9.3	Masked Images	54
4.9.4	Stacked Images	54
4.9.5	Velocity Phase Space	55
4.10	Miscellaneous Functions	55
4.10.1	Blurring	55
4.10.2	Getting Observation Times	55
4.10.3	Removing Poor Quality Images	56
4.10.4	Seeing	56
4.10.5	Shifting and Stacking Images	56
5	Results and Analysis	58
5.1	Testing the Algorithm	58
5.1.1	26 February 2014	59
5.1.2	30 May 2014	60
5.1.3	31 May 2014	61
5.1.4	1 June 2014	63
5.1.5	2 June 2014	64
5.1.6	3 June 2014	66
5.1.7	21 June 2014	66
5.1.8	22 June 2014	68
5.1.9	26 June 2014	69
5.1.10	28 June 2014	70
5.2	Statistics of the Testing	72
5.2.1	False Positives	75
5.2.2	Additional Asteroids	75
5.2.3	Unused Data	75
6	Summary	77
6.1	Future Work	78
	References	79

List of Figures

1.1	A graph of the number of asteroids found in each half year interval and which survey discovered them.	6
1.2	A diagram that explains the shift and stack method.	9
2.1	An image of a star field that contains asteroid 2014 CD13 taken on 26/02/2014.	15
2.2	The difference image of the image in Figure 2.1 produced by <i>PYSIS3.1.2</i> . . .	15
2.3	A difference image containing two images of the same asteroid.	16
3.1	A diagram of a Cassegrain telescope.	19
4.1	A diagram of the points on the reference image that were used to calculate the arcsecond per pixel scale.	26
4.2	A flow chart of how the algorithm deals with the length of the List.	28
4.3	A bright star from a section of an original image.	30
4.4	The difference image of the image in Figure 4.3.	30
4.5	The mask created by the algorithm to mask the star residuals in the difference image in Figure 4.4.	30
4.6	The difference image from Figure 4.4 with the mask from Figure 4.5 applied to it.	30
4.7	A saturated star in a section of an original image.	31
4.8	The difference image of the original image in Figure 4.7 that has been masked by the bright star mask.	31
4.9	The saturated star mask created by the algorithm for the middle saturated star in Figure 4.7.	31
4.10	The image from Figure 4.8 with the mask from Figure 4.9 applied to it. . . .	31
4.11	A ‘flare’ from an original image.	32
4.12	The difference image of Figure 4.11 that has been masked by the bright and saturated star masks.	32
4.13	The mask created by the algorithm for the Poor Quality Section in Figure 4.12.	32
4.14	The image from Figure 4.12 with the mask from Figure 4.13 applied to it. . .	32
4.15	Part of an image that contains an imperfection.	33
4.16	A difference image, where Figure 4.15 is the reference image, that has been masked by the star and poor quality section masks.	33

4.17	The mask created by the algorithm for the High Values in the image in Figure 4.16.	34
4.18	The image in Figure 4.16 with the mask from Figure 4.17 applied to it. The black pixels seen in Figure 4.16 have disappeared.	34
4.19	Three examples of the diamond-shaped mask used to remove the residuals of stars in difference images.	34
4.20	The algorithm follows this flow chart for the sections that might contain a poor quality section.	37
4.21	A flow chart of how the algorithm searches for asteroids for a particular velocity range.	42
4.22	A diagram of the velocity phase space of a group of images that contain the asteroid 2014 CJ13 on 26 February 2014.	44
4.23	The diagram in Figure 4.22 with the peak at the origin removed.	44
4.24	A stacked image that is comprised of the original images that are shifted relative to one another so that the images of the asteroid are aligned.	53
4.25	A stacked image that is comprised of the masked difference images that are shifted relative to one another so that the images of the asteroid are aligned.	53
5.1	A plot of the magnitude versus velocity of the test asteroids.	73
5.2	A plot of the visual magnitude versus velocity of the slow moving, faint test asteroids.	74

List of Tables

1.1	The number each type of NEO discovered up until 27 February 2015.	5
5.1	The default changeable variable values for testing the algorithm on data. . .	58
5.2	A table of the asteroids that were observed on 26 February 2014.	59
5.3	A table of the asteroids that were observed on 30 May 2014.	60
5.4	A table of the asteroids that were observed on 31 May 2014.	62
5.5	A table of the asteroids that were observed on 1 June 2014.	63
5.6	A table of the asteroids that were observed on 2 June 2014.	65
5.7	A table of the asteroids that were observed on 3 June 2014.	66
5.8	A table of the asteroids that were observed on 21 June 2014.	67
5.9	A table of the asteroids that were observed on 22 June 2014.	68
5.10	A table of the asteroids that were observed on 26 June 2014.	69
5.11	A table of the asteroids that were observed on 28 June 2014.	71

Abstract

The amount of interest in asteroids, particularly near-Earth asteroids, has been slowly increasing the last few years. Early on, it was the fear of the potential hazards of an asteroid impact (which still remains); now, science and exploration are driving the motivation for detecting asteroids. By incorporating a process that attempts to remove stars from images, known as difference image analysis, into the shift and stack method for finding moving objects, an algorithm that searches for asteroids in series of images was created.

When tested on previously discovered asteroids, the algorithm was able to detect 73% (104 out of 142) of the asteroids, with a false positive occurring in roughly one in six runs. Two of the test asteroids found by the algorithm were not found using a different shift and stack technique. Almost all of the asteroids that the algorithm could not detect were either slow moving, faint or both. The detection capability of the algorithm can easily be improved to be more successful at detecting slow and/or faint asteroids. This can be achieved by using a better method for masking the star residuals or by increasing the time between successive images when searching for asteroids with velocities of $2.4''/\text{min}$ or less. A better mask would also decrease the likelihood of a false positive and could increase the chance of an asteroid in a crowded field being detected. These improvements, if made, could give this method an advantage over methods currently in asteroid detected.

Acknowledgements

I dedicate this thesis to my oma, Francis Ouwerling, who turned 81 on the day this thesis was submitted.

Professional

I would like to thank my supervisor, Michael Albrow, for all the help he has given me throughout my thesis. I would also like to thank Alan Gilmore and Pam Kilmartin for supplying the images that were used in the testing of the algorithm, and for their advice on detecting asteroids. I would also like to thank the creators of PYSIS3.1.2, IRAF, and Astrometry.net for the use of their software.

Personal

I would like to thank all my friends and family, especially my fellow office mates, for all the happy times and cake when the thesis writing got tough. I would also like to thank Christoph Bergmann, for the thesis page count race, and the Blackcaps, for their exceptional cricketing abilities.

1 Introduction

1.1 What Are Asteroids?

Asteroids, from the Greek word *asteroeidēs*, meaning starlike, are astronomical objects that orbit the sun directly, not a planet or a comet, and reside in the inner part of the Solar System (out to the orbit of Jupiter). Asteroids that are smaller than 1 m are usually labeled as meteoroids.

The two main areas where the majority of asteroids are located are the main asteroid belt and Jupiter trojans. Main belt asteroids have orbits that lie between Mars and Jupiter. Asteroids that belong in the Jupiter trojan group have similar orbits to Jupiter and are about 60° in front and behind Jupiter.

1.1.1 Near Earth Asteroids (NEA)

The definition of a Near Earth Asteroid is an asteroid that, at some point in its orbit, is closer than 1.3 AU^1 to the Sun. NEAs are believed to originate from the asteroid belt [Wetherill, 1979] and had their orbits altered due to gravitational interactions with Jupiter and the Yarkovsky effect (thermal drift force) [Morbidelli and Vokrouhlický, 2003]. They tend to have a ‘lifetime’ of a few million years and, due to perturbations by the planets, are either flung out of the Solar System or impact with a planet or the Sun [Morbidelli et al., 2002].

There are four different types of NEAs depending on their orbit:

Atira These Asteroids have orbits that are completely inside the orbit of the Earth.

Aten Asteroids that belong in the Aten group have orbits that cross the orbit of the Earth and have a semi-major axis² less than 1AU.

Apollo Just like the Atens, their orbits cross Earth’s orbit but they have a semi-major axis greater than 1AU.

¹An Astronomical Unit (AU) is the average distance between the Earth and the Sun.

²The semi-major axis of an orbit is half the longest diameter of the ellipse created by the orbit.

Amor NEAs that have orbits that are completely outside the Earth's orbit are classed as Amors.

One should note that these classifications are only concerning the plane of the Earth's orbit and do not take into account the asteroids orbital inclination (the angle between the two orbital planes). Therefore even though Atens and Apollos intersect Earth's orbit in the two-dimensional Earth orbit plane, they likely will not in three-dimensional space, making a collision between a NEA and Earth very unlikely but still possible. One should also note that orbits are not static, so an orbit that has no chance of collision can change into one where that has.

NEAs that have a predicted path that brings them to within 0.05 AU of Earth's orbit are labeled as potentially hazardous asteroids (PHAs).

1.2 The Importance of Finding NEAs

Asteroids have always been important in terms of the damage one can do if it collided with the Earth, even though humans have just comprehended it in the near past. Asteroids can increase our understanding of the universe, for their composition can give insight into how the Solar System formed. Recently, the advancement of technology has allowed companies to consider the possibilities of mining asteroids for resources.

1.2.1 The Threat of Asteroid Impacts

The effects of asteroids colliding with larger bodies in the Solar System are evident by the number of craters on the Moon, Mars and Mercury. Even on the Earth, with its ever changing landscape, there are scars left by falling space rock, the prime example being Meteor Crater in Arizona. The damage to life is far more significant. In 1908 the explosion due to a roughly 50 m in diameter asteroid entering Earth's atmosphere destroyed thousands of square kilometres of forest in Siberia. If this asteroid had landed in a different location, a whole city would have been turned into rubble.

Asteroid impacts can cause catastrophe on a global scale. It was an asteroid that is widely believed to be the major contributor to the extinction of dinosaurs. In Mexico the Chicxulub impact structure is the site of where the asteroid unleashed billions of megatons of energy onto the Earth that caused the mass extinction at the end of the Cretaceous period, about 65 million years ago [Hildebrand and Boynton, 1990]. If an asteroid roughly

100 times less energetic hit Earth during modern times, a firestorm the size of India and worldwide loss of agriculture for a season, among other effects, could result in billions of deaths [Chapman, 2004]. Luckily impacts like this occur very rarely, every tens to hundreds of million years.

If any future asteroid impact were known years before they were to occur, measures can be made to minimise the amount of destruction the asteroid would inflict. The path of the asteroid could be altered to avoid a collision with Earth or evacuate the area where the asteroid is likely to hit. Therefore discovering all NEA could save a lot of lives.

1.2.2 Understanding the Early Solar System

The belief in the scientific community is that asteroids and comets were created in the early phases of the solar system. Therefore they can provide a glimpse into how the protoplanetary nebula became the Solar System, in terms of environmental conditions and chemical composition [Xu et al., 2007].

In order to learn more about the asteroids and comets, a number of space agencies have sent, and are planning to send, probes to bring back samples of these objects. The National Aeronautics and Space Administration (NASA) plans to go a step further and capture a small, 2-10 m diameter, NEA [Lightfoot, 2013]. After being captured, the asteroid will be brought into a lunar orbit so astronauts can travel to it and bring back samples to Earth. This mission is known as the Asteroid Redirect Mission (ARM) and is likely to occur sometime in the 2020s.

Before ARM can go ahead, NASA need to find the right asteroid to capture. Since NASA are looking for such a small NEAs, detection equipment and methods have to vastly improve.

1.2.3 Asteroid Mining

Asteroids, and extinct comets³, contain materials that can be of use to humankind. Precious metals⁴ are rare in the Earth's crust, making them of high monetary value. Since some types of asteroids contain large quantities of precious metals [Kargel, 1994], a profit could be made by mining these types of asteroids and returning the precious metals to Earth.

³Extinct comets are comets that have lost most of their volatile ice.

⁴The precious metals are gold, platinum, iridium, osmium, palladium, rhodium and ruthenium.

1 Introduction

In the last few years some companies have expressed interest in mining asteroids for profit in the near future. One of these companies is Planetary Resources. According to their website⁵ they are looking to develop three different kinds of spacecraft. One that studies NEAs and finds suitable candidates. Another one will get close to the NEAs to get accurate data about them. The third will study the shape, rotation, density, and surface and sub-surface composition of asteroids that are not close by.

To save money on transporting costs of getting materials into a low Earth orbit, missions on their way to other locations in the Solar System could mine asteroids for water. Carbonaceous asteroids, which make up about 10% of all the Near Earth Objects (NEOs) [Stuart and Binzel, 2004], contain water. This water could be mined and used for life support or turned into oxygen and hydrogen, and used for rocket fuel.

For companies like Planetary Resources, or future space missions, to successfully mine asteroids, a sufficient number of asteroids with the desired characteristics have to be discovered. Just like with ARM, companies would want to mine small NEAs, so detection equipment and methods have to vastly improve.

1.3 A History of NEA Discoveries

1.3.1 Early On

Eros was the first NEA discovered, by Carl Gustav Wittin in 1898. It is also the second largest NEA, after Ganymed (which was found in 1924), and belongs to the Amor group. The first NEA that crosses Earth's orbit was found 34 years later, in 1932. This asteroid was given the name Apollo, and is an Apollo type NEAs, which is where the name type comes from. Due to limitations in the technology and techniques, new NEA were hard to come by for most of the 20th century. The number of NEAs discovered did not pass the 100 mark until after the late 1980s, which is about the time the number of new asteroid discoveries took off.

1.3.2 Spaceguard Survey

The increase in NEA discoveries in the late 1980s led to the realisation of how close some of these asteroids were getting to Earth, which caused the U.S. Congress to make a man-

⁵<http://www.astronomysource.com/tag/m-type-asteroids/>

date that NASA needs to find 90% of all NEAs larger than 1km within the next 10 years [Harris, 2008]. This was known as the Spaceguard Survey, which NASA got behind in 1998. In 2011 the spaceguard goal was found to be met [Mainzer et al., 2011], which the LINEAR survey, see Section 1.4.1, contributed the most number of 1 km NEA discoveries.

1.3.3 George E. Brown, Jr. Near-Earth Object Survey Act

Even before the Spaceguard Survey was completed, the U.S. Congress produced another mandate to NASA, which is known as the George E. Brown, Jr. NEO Survey Act. This act was formed in 2005, and it requires NASA to discover, track and identify the physical characteristics of 90% of all NEOs that have a diameter that is 140m or larger before the end of 2020. More on the act can be found here <http://neo.jpl.nasa.gov/neo/report2007.html>.

1.3.4 Current Numbers

As of 27 February 2015, the number of Near Earth Objects found is 12,330, where the vast majority are NEAs, as can be seen in Table 1.1. Over half of these NEOs are of Apollo type, which are the ones that cross Earth's orbit, but only a minority of them are classed as PHAs. Of all the NEAs found so far, only 1559 are PHAs. 151 of them are larger than 1 km. The surveys that found most of these asteroids are described in Section 1.4, along with one future survey to find smaller NEAs.

The number of NEAs that have a diameter larger than 140m that have been found since 12 January 2015 is very roughly 6500⁶. The total number of these NEAs is thought to be $13,200 \pm 1900$ [Mainzer et al., 2011]. Therefore 43-58% of NEAs larger than 140m have been found, which is some way off the goal of the George E. Brown, Jr. NEO Survey Act of 90%.

NEO Type	Number Discovered
Near Earth Comet	96
Atira	15
Aten	905
Apollo	6615
Amor	4699
Total	12330

Table 1.1: The number each type of Near Earth Objects (NEO) discovered up until 27 February 2015. The data in this table was obtained from <http://neo.jpl.nasa.gov/stats/>.

⁶<http://neo.jpl.nasa.gov/stats/>

1.4 Surveys

The three surveys that are most prolific in finding NEAs are the LINEAR, Catalina Sky Survey and Pan-STARRS, as can be seen from the graph in Figure 1.1. These surveys mostly find the big NEAs, which are the ones that can cause catastrophic damage to life. If any of these NEAs that will impact Earth are discovered, the time frame to prepare for the collision will be in the years. A new survey that is planned will deal with the opposite situation. ATLAS will search for small NEAs, around 100 m and less in size, for which only a couple of weeks notice will be given.

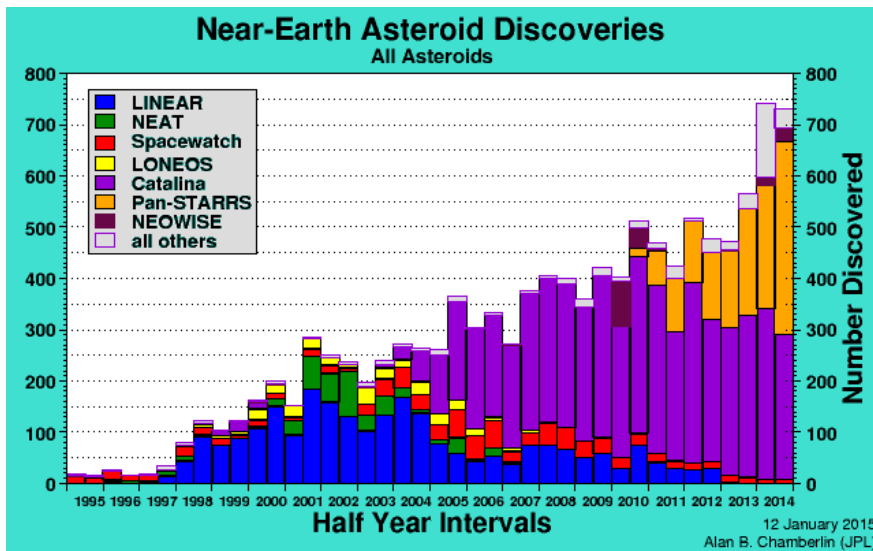


Figure 1.1: A graph of the number of asteroids found in each half year interval and which survey discovered them. This image is from <http://neo.jpl.nasa.gov/stats/>.

1.4.1 Lincoln Near-Earth Asteroid Research (LINEAR)

The LINEAR project, from the Massachusetts Institute of Technology's Lincoln Laboratory, has been searching for NEAs since 1998. LINEAR uses two 1 m telescopes located in New Mexico to search for asteroids that are larger than 1 km in diameter [Elowitz et al., 1999]. Since it is believed that more than 90% of all NEAs over 1 km in diameter have been found [Mainzer, 2011], the two telescopes were discontinued in 2013. A 3.5 m telescope is planned to take over the search.

1.4.2 Catalina Sky Survey (CSS)

The main goal of CSS is to find PHAs. Two telescopes, one 1.5 m and the other 0.68 m, in the Tucson area of Arizona are used for CSS [Larson et al., 2003]. Currently CSS has found the most NEOs out of all the surveys and is still going strong.

1.4.3 Panoramic Survey Telescope and Rapid Response System (Pan-STARRS)

Starting in 2010, Pan-STARRS has been searching for NEAs in the whole area of the sky that has a declination greater than -47.5° . Just like CSS, the primary mission of Pan-STARRS is to search for PHAs. This is achieved with the use of a 1.8 m telescope in Hawaii, with more telescopes coming [Kaiser et al., 2010].

1.4.4 Asteroid Terrestrial-impact Last Alert System (ATLAS)

ATLAS, which is planned to be completed later this year (2015), will use two telescopes 100 miles apart to search the entire sky a few times a night for moving objects [Tonry, 2011]. According to their website <http://www.fallingstar.com/home.php>: “*ATLAS can provide one day’s warning for a 30-kiloton ‘town killer’, a week for a 5-megaton ‘city killer’, and three weeks for a 100-megaton ‘county killer’.*”.

1.5 How to Detect Moving Objects

By taking multiple images of the same star field one would see a moving object, like an asteroid or comet, change positions relative to the stars. Assuming the moving object is in at least two images, the velocity of the object can be calculated, which can be used to find the orbital elements of the object. The more images with the object in it, the more accurate the object’s orbital parameters are, which leads to a higher chance of finding the object again at a later date.

1.5.1 Detection of Faint Moving Objects

If the moving object is not bright enough to be seen in a single image, then the object’s signal-to-noise ratio (SNR) has to be increased. The SNR of an object is a measure of the quality of the data. A faint object would produce a low SNR, meaning that there is high inaccuracy in the objects data. If it is low enough then there is doubt as to whether there is an object there at all. Whereas a bright object would have a high SNR, meaning that the data is excellent and can be used to get high precision results.

1 Introduction

A common way to increase the SNR of a faint star is to increase the exposure time of the image. If the same approach is attempted on objects moving on the celestial sphere⁷, a streak will occur on the image and the results will differ. The ratio of the SNR of an object if it is moving to the SNR of the same object if it was stationary is approximately [Shao et al., 2014]:

$$\frac{SNR_v}{SNR_0} = \frac{w}{w + s} \quad , \quad (1.1)$$

where w is the width of the object image and s is the length of the streak of the moving object.

As can be seen from Equation 1.1 the longer the exposure, the longer the streak, then the smaller the SNR of the streaked image compared with a non-streaked image. Therefore increasing the exposure time is not an efficient way of increasing the SNR of an asteroid (unless it is a slow moving asteroid).

1.5.2 The Shift and Stack Method

A common method for finding moving objects is the shift and stack technique, also known as “shift and add” and “stack and track”. This involves taking multiple short exposure images, short enough that the image appears as a point source and not a streak, then shifting the images relative to one another so the object in each image is aligned with each other, and then combining the images into one by adding the pixel values, as can be seen in Figure 1.2. Thereby the SNR of the moving object is increased.

Using the time of acquisition of the stacking images, the amount the images are shifted by can be turned into a shifting velocity. If the moving object in each image is stacked on-top of one another then the shifting velocity will be equal and opposite to the objects velocity.

The shift and stack method was used in 1992 to find Kuiper Belt Objects (KBOs) [Tyson et al., 1992], even though a description of the method used was severely lacking. In 1995 the shift and stack method was used again, but this time with a detailed description, to find very faint Trans-Neptunian Objects (TNOs) [Cochran et al., 1995]. Since then the shift and stack method has been successfully used many times to find TNOs

⁷Stars are in fact not stationary and also move on the celestial sphere, but their movement in the time frame of an image exposure is negligible.

and KBOs ([Gladman and Kavelaars, 1997], [Gladman et al., 1998], [Gladman et al., 2001], [Luu and Jewitt, 1998], [Chiang and Brown, 1999], [Allen et al., 2001], [Allen et al., 2002], [Fraser et al., 2008], [Fraser and Kavelaars, 2009], [Fuentes et al., 2009]).

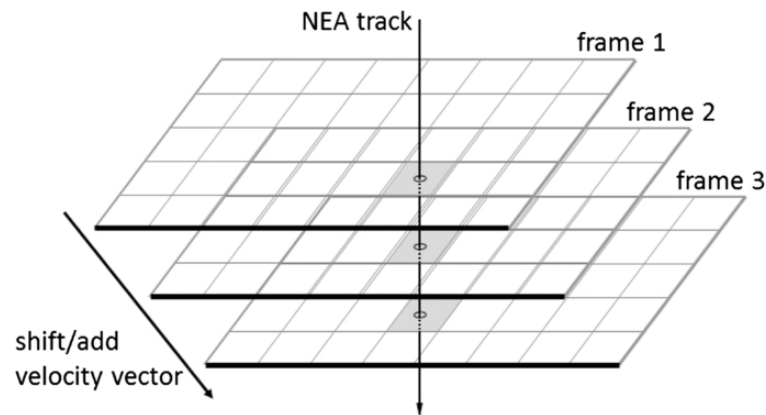


Figure 1.2: A diagram that explains the shift and stack method. Three images (frames) are shifted relative to one another using a shift velocity that corresponds to the asteroid being aligned with itself in all three frames. [Zhai et al., 2014]

1.5.3 Detection of Asteroids

The shift and stack technique is very computationally intensive, which is why none of the major NEA surveys use it. Some surveys, like LINEAR, use moving target indicator algorithms (MTI) to find asteroids, due to the fact that they are implemented easily and do not require much computational power. MTI algorithms usually work by finding all objects in a few images and comparing these objects across the images to see if any of them correspond to a moving asteroid. Even though the shift and stack method will find more asteroids [Gural et al., 2005], some of the faster moving asteroids have to be reimaged on the same night to increase the precision of the orbital elements so that these asteroids can be recovered during the next night. The shift and stack method is usually too slow in to achieve same-night reimaging.

KBOs, TNOs and asteroids are the only space objects the shift and stack method has helped to find. The technique has also been used to locate space debris in geosynchronous orbits [Yanagisawa et al., 2005].

1.6 Stacking Images

If a series of images were shifted relative to one another with a constant shifting velocity of $V = (v_x, v_y)$ and then stacked, the value of a single pixel in the stacked image would equal

$$S_{ij} = \sum_k I_{(i-v_x t_k)(j-v_y t_k)}^k \quad , \quad (1.2)$$

where I^k is the k^{th} image in the series, and t_k is the time between acquisition of the k^{th} image and an arbitrary image in the series.

If, for the sake of simplicity, random noise is approximated to be the only source of noise, then the number of counts in a pixel follows Poisson distribution and the noise of the pixel I_{ij} is equal to $\sigma^2 = I_{ij}/g$ [Newberry, 1991]. g is the gain of the image, which is the conversion factor of the number of electrons that get excited by a photon in the CCD to the number of counts in the final image or ADU (analogue to digital unit). Using this equation the variance of S_{ij} can be calculated as

$$\sigma_{s_{ij}}^2 = \frac{1}{g} \sum_k I_{(i-v_x t_k)(j-v_y t_k)}^k \quad . \quad (1.3)$$

The SNR of a pixel in any image can be calculated by dividing the value of the pixel by the standard deviation of the pixel. Therefore, by combining both Equation 1.2 & 1.3, the SNR of a pixel in a shifted and stacked image is

$$\frac{S_{ij}}{\sigma_{s_{ij}}} = \frac{\sqrt{g} \sum_k I_{(i-v_x t_k)(j-v_y t_k)}^k}{\sqrt{\sum_k I_{(i-v_x t_k)(j-v_y t_k)}^k}} \quad , \quad (1.4)$$

1.6.1 Stacking Difference Images

A difference image is, in simplistic terms, the difference in two images of the same star field. Difference images only contain objects that move or have variable brightness from the original images. Details on what difference images are and how they are made are given in Section 2.1. When stacking difference images, Equation 1.4 becomes

$$\frac{S_{ij}}{\sigma_{s_{ij}}} = \frac{\sqrt{g} \sum_k D_{(i-v_x t_k)(j-v_y t_k)}^k}{\sqrt{\sum_k O_{(i-v_x t_k)(j-v_y t_k)}^k}} \quad , \quad (1.5)$$

where D are the difference images and O are the original images.

1.6.2 Weighted Averages

When dealing with multiple measurements of a single quantity, a simple way of combining them is to find the average value. If the measurements have uncertainties, then these uncertainties can produce a more accurate way of combining the measurements in the form of weighted averages. The weighted average, x_{wav} , of N measurements of a quantity x is [Taylor, 1997]

$$x_{wav} = \frac{\sum_{i=1}^N w_i x_i}{\sum_{i=1}^N w_i} \quad , \quad (1.6)$$

with a variance of

$$\sigma_{wav}^2 = \frac{1}{\sum_{i=1}^N w_i} \quad , \quad (1.7)$$

where $w_i = \frac{1}{\sigma_i^2}$ and σ_i is the uncertainty of the x_i measurement.

1.6.3 Optimally Stacking Difference Images

An optimally stacked difference image can be obtained with the use of weighted averages. Applying Equation 1.6 to Equation 1.5, a pixel in the optimally stacked difference image equals

$$S_{ij} = \frac{\sum_k D_{(i-v_x t_k)(j-v_y t_k)}^k / [\frac{1}{g} T_{(i-v_x t_k)(j-v_y t_k)}^k]}{\sum_k 1 / [\frac{1}{g} T_{(i-v_x t_k)(j-v_y t_k)}^k]} \quad . \quad (1.8)$$

Using Equation 1.7 the variance of a pixel in the optimally stacked difference image equals

$$\sigma_{s_{ij}}^2 = \frac{1}{g \sum_k 1 / T_{(i-v_x t_k)(j-v_y t_k)}^k} \quad . \quad (1.9)$$

Combining Equations 1.8 & 1.9, the SNR of a pixel in the optimally stacked difference image is

$$\frac{S_{ij}}{\sigma_{ij}} = \frac{\sqrt{g} \sum_k D_{(i-v_x t_k)(j-v_y t_k)}^k / T_{(i-v_x t_k)(j-v_y t_k)}^k}{\sqrt{\sum_k 1 / T_{(i-v_x t_k)(j-v_y t_k)}^k}} \quad . \quad (1.10)$$

1.7 Angular Separation

The motion of an asteroid, or any moving object, on the celestial sphere can be calculated by finding the angular separation between the asteroid in two points in time. The angular separation of two points on the celestial sphere, A , can be found using the Pythagorean formula

$$A = \sqrt{(\Delta\delta)^2 + (\Delta\alpha)^2} \quad , \quad (1.11)$$

where $\Delta\delta$ is the declination arclength and $\Delta\alpha$ is the right ascension arclength. $\Delta\delta$ is just the difference in declination of the two points $\delta_b - \delta_a$. Since sphere geometry is different to planar geometry, $\Delta\alpha$ depends on the declination of the two points. If the two points are close together (separated by less than a degree) then this approximation can be used

$$\Delta\alpha = (\alpha_b - \alpha_a) \cos\left(\frac{\delta_b + \delta_a}{2}\right) \quad . \quad (1.12)$$

If the right ascension is in hours then the right hand side of Equation 1.12 needs to be multiplied by 15 so that both the arclengths are in degrees⁸. Therefore Equation 1.13 becomes

$$A = \sqrt{(\delta_b - \delta_a)^2 + \left((\alpha_b - \alpha_a) \cos\left(\frac{\delta_b + \delta_a}{2}\right)\right)^2} \quad . \quad (1.13)$$

1.8 Thesis Outline

Since the presences of stars are lesser on difference images compared with the original images, as can be seen in Figure 2.2, the incorporation of difference images into the shift and stack method could lead to better detection rate, especially in crowded star fields. In the rest of this thesis I will show how I created an algorithm in the programming language `python` that detects asteroids by shifting and stacking difference images. A breif introduction into how difference images are created, along with other software needed to run the algorithm, is in Chapter 2. Chapter 4 describes in detail how the algorithm works, including the removal of imperfections in difference images, the way the difference images are stacked and the criteria for the asteroid to be classed as detected.

The algorithm is tested on multiple series of images, that contain an asteroid that have already been discovered, to determine whether this is an effective method of asteroid detection. The equipment used to take the test images are described in Chapter 3. Chapter 5 contains the results of the test and statistics on how well the algorithm performed. Finally, Chapter 6 summarises the thesis and suggests possible follow up work.

⁸since $\frac{(\text{hour})}{(\text{degree})} = 15$

2 Software

A couple of computer programs were used in this thesis. This section provides details on these programs, including the software that produces the difference images, the functions in IRAF to produce the pixel positions and magnitudes of the stars in the same star field as the asteroid, and the program used to calculate the coordinates of the asteroids.

2.1 Difference Image Analysis

Difference Image Analysis (DIA) effectively removes stationary objects of constant brightness from images, as it is the difference between two images. The main purpose of DIA is to produce high quality photometric data of stars in crowded fields, with errors close to the theoretical Poisson limits [Bramich, 2008]. This method was first used successfully in 1998 [Alard and Lupton, 1998]. Since the high quality photometry from DIA can be used to create accurate lightcurves¹ of stars, some of the uses of DIA over the years have included searching for planets through the transit method [Bramich et al., 2005] and finding microlensing events [Bond et al., 2001].

2.1.1 How to Produce a Difference Image

A difference image is defined as the target image subtracted from a convoluted reference image

$$D = R \otimes K - O \quad , \quad (2.1)$$

where D is the difference image, R is the reference image, K is the convolution kernel, and O is the original image. The convolution kernel of the DIA used in this thesis is a discrete pixel array, where the pixel values are calculated by minimising χ^2 in a least squares

¹A star's lightcurve is a function of the star's brightness over time.

fit [Bramich, 2008]

$$\chi^2 = \sum_{ij} \left(\frac{D_{ij}}{\sigma_{ij}} \right)^2 . \quad (2.2)$$

2.1.2 PYSIS3.1.2

The program *PYSIS3.1.2* [Albrow et al., 2009] was used to create the difference images used in this thesis. *PYSIS3.1.2* is run by typing the following into the command terminal:

```
PATH/pySIS3.1.2/bin/reduce3 -s SITE -e FILE -i ds9 -n 3
```

where **PATH** is the *PYSIS3.1.2* directory path, **SITE** is the code name of the site used to create the images² and **FILE** are the fits files that DIA is to be performed on. An example of what a difference image produced by *PYSIS3.1.2* looks like is in Figure 2.2.

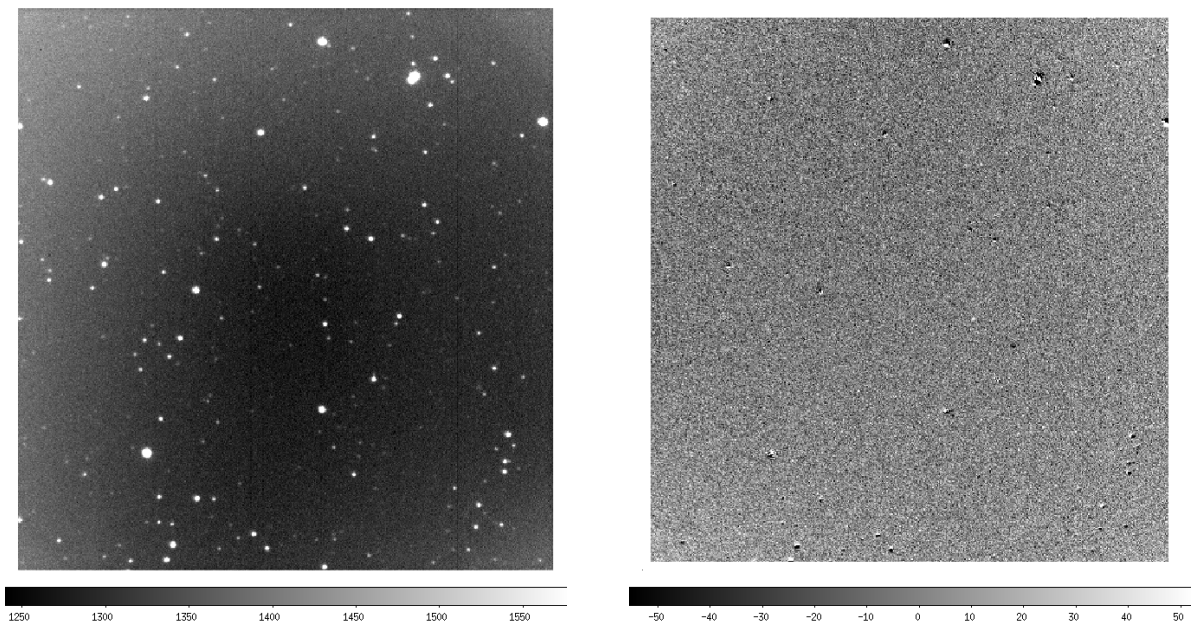


Figure 2.1: An image of a star field that contains asteroid 2014 CD13 taken on 26/02/2014.

Figure 2.2: The difference image of the image in Figure 2.1 produced by *PYSIS3.1.2*.

²The code name has to be in the list of sites in *PYSIS3.py*, with details about the images, like the gain, read out noise, how the image file names are structured, et cetera.

2.1.3 Creating the Difference Images used by the Algorithm

Usually *PYSIS3.1.2* creates the reference image by combining multiple images with comparatively good seeing. For this thesis only one image was used as the reference image. This was achieved by creating a text file *Ref.include*, which only contains the file name of the image with the best seeing, before running the program.

When running *PYSIS3.1.2* with the text file *Ref.include* in the same directory, after a while the program opens the reference image in *ds9*³ and asks for the coordinates of the star on which photometry is to be performed. Since photometry is not needed, the coordinates for the centre of the image were entered. The program then finds the nearest star to the coordinates given and asks if this was the desired star. After inputting no, an error occurred but not before the desired difference images were created.

A difference image has a file name of *conv_* followed by the file name of the original image used to create it. The program still creates a difference image of the image that is used as the reference image. This difference image is not used by the algorithm, so it was deleted. The program also creates, before the error occurs, a text file with the seeing of all the images, labeled *seeing*, and original images that have been aligned so the stars have the same pixel coordinates as the ones in the reference image, labeled *interp_* followed by the original image file name. Both the *seeing* file and the *interp* images are used by the asteroid finding algorithm.

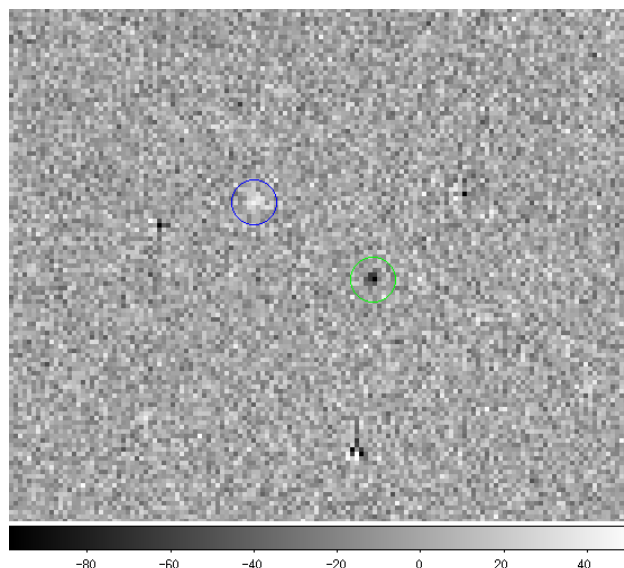


Figure 2.3: A difference image containing a black dot caused by an asteroid in the original image (green circle) and a white dot caused by the same asteroid in the reference image (blue circle).

³*ds9* is an astronomical imaging and data visualisation application.

2.1.4 Asteroids in Difference Images

If there is an asteroid in the original image, then, since the original image is subtracted from the reference image, the asteroid will appear as a negative image on the difference image. If the asteroid is also in the reference image, then there will be a second image of the asteroid on the difference image, but this time a positive image of the asteroid. Figure 2.3 shows both the images of an asteroid in a difference image.

2.2 Star Identification

To run the algorithm on a series of images, text files containing the positions and magnitudes of all the non-faint stars in the reference image need to be in the same directory as the difference images, see Section 4.2. This section explains how the software IRAF (Image Reduction and Analysis Facility) is used to create the star text files. IRAF is a common tool used by astronomers to reduce and analyse astronomical data⁴.

The difference images were created before the stars were identified, in order to know which image was to be used as the reference image and what the seeing of the reference image was. The reference image was loaded into IRAF and the function *daofind* was used to search for local density maxima in the image. Any maxima found which had a full-width half-maximum (FWHM) equal to that of the reference images seeing, which has to be divided by 2.355⁵ to convert from standard deviations to FWHM, and a peak amplitude greater than 2 above the background was saved along with the coordinates of the centre of maxima. The seeing of the reference image was taken from the *seeing* text file created during DIA, see Section 2.1.3. The standard deviation of the sky was calculated from roughly averaging the standard deviation obtained from about 10-20 areas of the reference image that do not have any stars in them using the *imexam* function in IRAF. All the maxima found by *daofind* were assumed to be all the stars in the reference image.

The IRAF function *phot* uses the coordinates saved from *daofind* to perform photometry on all the stars in the reference image to produce a magnitude for each star. *pstselect* reads the data on 25 stars calculated in *phot*, and the function *psf* builds the point spread function (PSF) of the reference image using these 25 stars. *nstar* calculates more accurate values for the positions and magnitudes of the 25 stars by fitting the PSF to the data in the reference image. *substar* creates a new image by removing any stars neighbouring the 25 stars from the reference image. A better PSF of the 25 stars was created using the *psf* func-

⁴For more on, IRAF visit <http://iraf.noao.edu/>.

⁵Since $\text{FWHM} = 2\sqrt{2\ln 2}\sigma$.

tion on the newly created image by *substar*. This better PSF was used, in conjunction with the function *allstar*, to create better positions and magnitudes of all the stars in the reference image. If the function could not successfully fit the PSF to a star, then that star was rejected.

The output star data from the *allstar* function were sorted from brightest to faintest using the *psort* function. The column that contains the magnitudes were copied into a newly created text file with the file name *Mag.txt*. Similarly, the columns that contain the x and y positions were copied into newly created text files with the file names *Xvalues.txt* and *Yvalues.txt*, respectively.

2.3 Astrometry.net

Astrometry is the science of precisely measuring the positions of objects on the celestial sphere. Astrometry is important for asteroid discoveries, since accurate positions of the asteroid are needed to calculate its orbital parameters. Even though this thesis was more concerned with detecting the asteroid than cataloging it, astrometry can be used to calculate the asteroid's velocity. This is useful for deciding what range of asteroid velocities to search for, and for comparing the asteroid's velocity calculated by the algorithm with the actual velocity to confirm that the asteroid has been correctly detected.

To find the world coordinates (RA, Dec) of an object in an image, the conversion from pixel to world coordinates needs to be known. The conversion can be obtained using the program Astrometry.net. When Astrometry.net is run on an image, the program finds the brightest stars in the image and compares the pattern these stars make in the image with patterns of stars in images of known areas of the sky. These images of known areas of the sky are called index files. If the program finds a match then it produces, amongst other files, an image that is the same as the input image but also has a RA and Dec value for each pixel.

Astrometry.net can either be used through a command line or through a web service at <http://nova.astrometry.net/>. The latter was used for this thesis. To run Astrometry.net through a command line the Astrometry.net software package had to be downloaded, along with the appropriate index files. The software package, index files, and information on how to install it can be found at <http://astrometry.net/use.html>. For this thesis the 4200, 4201, 4202, and 4203 index files were used.

3 Equipment

The data used to test the algorithm was taken by Alan Gilmore and Pam Kilmartin. The telescope and camera they used are detailed below.

3.1 Telescope

The data used in testing the algorithm was taken using the 1 m McLellan telescope at Mount John University Observatory. It is a Cassegrain telescope, which consists of a primary mirror that reflects light onto a secondary mirror in front of it. Then the secondary mirror reflects light down the centre of the primary mirror, as shown in the diagram in Figure 3.1. The McLellan telescope is a Dall-Kirkham type of Cassegrain telescope, which has a concave elliptical primary mirror and a convex spherical secondary.

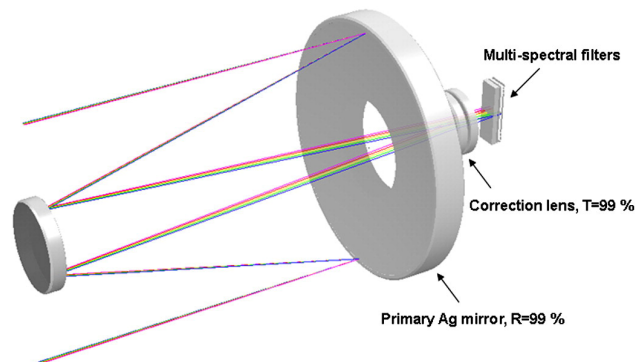


Figure 3.1: A diagram of a Cassegrain telescope with the path of two light rays coming in from the left and reflecting off the primary mirror, to the secondary mirror, then through the hole in the primary mirror to the detector.[Hsiao, 2014]

3.2 Camera

Apogee Alta F6

The Alta F6 is the camera used to produce the images used to test the algorithm on. It is a front-illuminated CCD (Charged Couple Device). The size of the camera is 1024 by 1024 pixels, where each pixel is 24 microns in length and width. The pixels are binned with a bin

3 Equipment

size of 2 by 2 pixels, which changes the camera to an effective size of 512 by 512 pixels and the pixels to an effective size of 48 microns. When used with the McLellan telescope's 7.7 m focal length, the pixels have a scale of about 1.2 arcseconds.

The Alta F6 is thermoelectrically-cooled to around 50°C below ambient. When used with the McLellan telescope, on most nights the camera is cooled down to between -30°C and -35°C, rarely to -40°C. For more information on the Alta F6 please go to the website www.ccd.com/alta_f6.html.

4 The Algorithm

4.1 Overview

This chapter describes in detail how the algorithm works, including all the files that are inputted or outputted, what is done to the difference images before stacking, how the difference images are stacked, and what is required for a successful asteroid detection.

Before running the algorithm, certain types of files with the correct file name should be in the current working directory. If not the algorithm will not successfully run. A list of these files can be found in Section 4.2.

Throughout the algorithm there are variables that affect how the algorithm performs. There are two types, variables that may have to be altered to be able to detect some asteroids, and variables that depend on the specifications of the camera used to take the images and should not be changed. Section 4.3 contains the lists of both the types of variables, what they are used for, and where in the algorithm they occur.

One of the problems with trying to find an undiscovered asteroid is the fact that the asteroid's velocity will be unknown. Therefore many different shifting velocities have to be used to be able to find the correct velocity that stacks the images of the asteroid on top of one another. Since it would be impractical to trial every velocity, a velocity range is needed. Section 4.5 details how one is created with the use of astrometry.

To be able to load the difference images into python, the algorithm needs to know their file name. This is presented in Section 4.6. Not all the difference images have to be used, the algorithm may choose not to use some of the images depending on how many images there are and what asteroid velocities are being searched.

Unfortunately the difference images contain imperfections that could interfere with the detection of an asteroid. These imperfections are mainly due to stars but they also include, but are not exclusive to, cosmic rays that were missed in the cosmic ray removal process and faulty pixels in the CCD. Masks are used to remove the imperfections. The algorithm will

4 The Algorithm

either load (if they exist) or make the masks. Details on how and when the algorithm loads the masks are given in Section 4.4.

If the algorithm has to make the masks, then it performs the masking in four different ways. Firstly removing bright star residuals, followed by saturated star residuals, after that poor quality sections, and lastly individual high value pixels. Section 4.7 contains the method used for masking. If masking was not successful in removing a majority of the imperfections from one or more of the difference images, then those images are not used in the asteroid search process.

The masked difference images are stacked using a range of different shifting velocities. By taking the minimum pixel value in each stacked image a velocity phase space matrix is created, where the position of the matrix elements correspond to a componential shifting velocity. Due to the signals of faint stars, the origin of the velocity phase space matrix is removed. By comparing the minimum value of the origin-removed velocity phase space matrix with the values at the same distance from the origin, the algorithm decides whether an asteroid is causing the minimum value or not.

If an asteroid is found, the algorithm saves the asteroids position in each image to a text file. The asteroid is masked from the difference images and the search is performed again. If no asteroid is found then the algorithm either changes the velocity range limit or masks the difference image more. The search for asteroids is then repeated. If too much masking is done, the algorithm stops searching. Section 4.8 has the details on the search for asteroids.

The files that the algorithm outputs depend on whether an asteroid is found. If no asteroid is found, fits files of the masks are created along with difference images that have been masked by just the Star Mask. If an asteroid is found, then along with the previously stated files, the algorithm also saves the masked difference images that were used to find the asteroid and two stacked images, one of the masked difference images and the other of the original images. Both the stacked images were shifted so that the asteroid aligned with itself in each image. Section 4.9 has more details on the files that the algorithm outputs.

Section 4.10 contains functions that are used in the algorithm that do not belong to a specific section. Some of these functions are used more than once. They include finding the acquisition time and seeing of each image, blurring images, and shifting and stacking the images.

4.2 Files Needed to Run the Algorithm

To be able to run properly the algorithm needs these files in the current working directory:

- The original images that start with the name of the current working directory and end with `.fits`.
- The reference image that has the file name `Ref.fits`.
- The difference images. They have the same file names as the original images but with a `conv_` at the beginning.*
- The original images that have been shifted to be aligned with the reference image. They have the same file names as the original images but with a `interp_` at the beginning.*
- Three text files. One that contains a column of the magnitude of the stars in the field of the reference image, and the other two containing the a column of the x and y positions of those stars. The text files are labeled `Mag.txt`, `Xvalue.txt` and `Yvalue.txt`, respectively.
- The raw image of the reference image that has the file name `Saturate.fits`.
- A text file containing four columns. The first column contains the names of the original images and the second column contains the image's seeing. The other two columns are not needed. Labeled as `seeing`.*

*These file are created by the program PySIS3.1.2.

4.3 Variables

At the top of the algorithm's script is a group of variables that affect certain aspects of the algorithm. Below are two lists of the variables. The first list contains variables that, for a particular set of images, may have to be changed so that asteroids can be detected. The second list contains variables that are camera dependent and, for a particular set of images, should not be altered. For each variable the section in which it was used is given.

4.3.1 Changeable

Half Tells the algorithm whether the number of difference images needs to be halved. Used in Section 4.6.1.

Star Mask Scaler Used to scale the size of the bright stars' masks. Used in Section 4.7.1.

Saturated Star Mask Scaler Used to scale the size of the saturated stars' masks. Used in Section 4.7.1.

High Value Number The minimum number of standard deviations below the mean a pixel value in a difference image has to be in order to be classed as a High Value. Used in Section 4.7.3.

Threshold Value The minimum ratio of the number of High Values in one section to the other, for the section with lesser number of high values to be discarded. Used in Section 4.7.3.

Poor Quality Section Mask Modifier Determines how much the poor quality section masks are increased by. Used in Section 4.7.3.

First High Value Mask Limit When the algorithm masks more than this number of “high negative values from the stacked image” in the initial high value masking, it stops the masking. Used in Section 4.7.

Second High Value Mask Limit When the algorithm masks more than this number of “high negative values from the stacked image” in the second or greater high value masking, it stops the masking. Used in Section 4.8.

Velocity Vector Length The number of different velocity values used in shifting and stacking for a particular direction. Used in Section 4.8.1.

4.3.2 Fixed

Saturation Count The value of a pixel that is saturated, i.e. the maximum value for a pixel. Used in Section 4.7.1.

Gain The conversion factor of the number of electrons that get excited by a photon in the CCD to the number count in the final image. Used in Section 4.8.9.

4.4 Reusing Masks

The algorithm creates masks for the difference images before they are shifted and stacked. Information on masking is given in Section 4.7. After creating the masks the algorithm saves them, so if the algorithm is rerun on the images, time can be saved by loading the masks instead of creating them again.

The first task the algorithm performs is to locate the directory **Masks** in the current working directory. If **Masks** is found, the algorithm will then search for the directory **Poor_Images**. If **Poor_Images** is also found, the algorithm will output “**Do you want to use the poor quality images? (y/n)** ”. Information about poor quality images can be found in Section 4.7.3. Answering yes will move the images from the directory **Poor_Images**

back into the current working directory and remove the directory `Poor_Images`¹. If the answer is no or the directory `Poor_Images` is not found, then the algorithm will output “**Do you want to reuse the masks?**”. If the answer is yes the algorithm will load all the masks into python and Section 4.7 will be skipped. If `Masks` is not found but `Poor_Images` is, then the algorithm will move the images from the directory `Poor_Images` back into the current working directory and remove the directory. If the masks are not reused, the algorithm will delete them and the `Masks` directory.

4.5 Velocity Range Limits

Since the velocity of an undiscovered asteroid is unknown, a range of shifting velocities has to be trialled to try to find the velocity that correctly stacks the images. Therefore velocity range limits have to be decided.

Most asteroids have velocities that can be assumed to be constant over the time frame of a series of images, which simplifies the search for the asteroids’ velocities.

4.5.1 Astrometry

To be able to convert from velocities in arcseconds per minute into the number of pixels each image has to be shifted by the arcseconds per pixel scale has to be known. The scale can be obtained by using the program `astrometry.net` to perform astrometry on the reference image.

The algorithm starts off by loading the reference image into python by using the `fits.getdata()` command on the `Ref.fits` file. The algorithm then searches the current working directory for `Ref.new`, to see if astrometry has been successfully performed on the reference image before². If it is found, the algorithm will load the file into python using the `WCS(’ ’)` command and continues onto the next section. If it is not found, the algorithm will ask “**Do you want to perform astrometry on the reference image? (y/n)**”. If the answer is yes then `astrometry.net` will be run. If the program is successful, then the newly created `Ref.new` file will be loaded into python and used as the new reference image.

If astrometry is not successful or not performed, then the algorithm will still run using velocity range limits in pixels per second instead. The problem is that if an asteroid is found, its proper position (in RA and Dec) and velocity (in arcseconds per minute) cannot be calculated.

¹Using the poor quality images means that the masks cannot be reused, since every image has its own mask and the poor quality images will not have complete masks in the `Masks` directory.

²`Ref.new` is one of the files created by `astrometry.net` if astrometry is successfully performed.

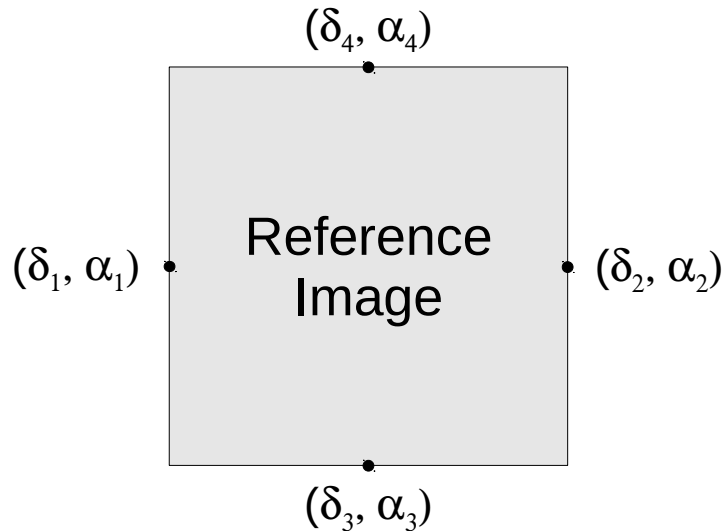


Figure 4.1: A diagram of the points on the reference image that were used to calculate the arcsecond per pixel scale.

4.5.2 Obtaining Velocity Range Limits

There are two ways the algorithm obtains velocity range limits depending on whether astrometry was successfully performed:

Astrometry WAS Successfully (or Already Has Been) Performed

The algorithm calculates the rough angular scale of each pixel by using the `wcs_pix2world` command to find the declination and right ascension of the middle pixel in each edge of the reference image. This corresponds to (δ_1, α_1) , (δ_2, α_2) , (δ_3, α_3) , (δ_4, α_4) in Figure 4.1. Next the angular separation between points 1 and 2, and points 3 and 4 are calculated using Equation 1.13. The inputs for this equation are α/δ_1 as α/δ_a and α/δ_2 as α/δ_b for the angular separation in the x-direction, A_x , and α/δ_3 as α/δ_a and α/δ_4 as α/δ_b for the angular separation in the y-direction, A_y . The angular separations are converted from degrees to arcseconds by multiplying the values by 3600^3 . The arcsecond to pixel scales for each direction are calculated using these equations

$$P_x = \frac{A_x}{l_x - 1} \quad \text{and} \quad P_y = \frac{A_y}{l_y - 1} \quad , \quad (4.1)$$

where l_x is the length of the reference image in the x-direction in terms of pixels, and similarly for l_y . The algorithm takes the average of P_x and P_y and assigns that as the arc-

³Since $\frac{(\text{degree})}{(\text{arcmin})} = 60$ and $\frac{(\text{arcmin})}{(\text{arcsec})} = 60$

second per pixel scale, P .

The algorithm will then ask **“Do you want to search for slow asteroids? (<3”/min) (Y/N)”**. If the answer is yes, then the algorithm will convert the highest velocity in the range (3”/min) into a pixels/s velocity using the arcsecond to pixel scale calculated just prior. A new list called “Limit” is created and the pixels/s velocity is added to it. The same process is repeated for **“medium-speed”** (3”-12”/min) and **“fast”** (12-48”/min)⁴. Then the algorithm asks **“If you want to use your own velocity limit, input it here (else put a zero): ”**. If the input value is greater than zero, the input value will be converted to a pixels/s velocity and added to “Limit”. If “Limit” is empty an error will come up.

Astrometry WAS NOT Successfully (or Not at All) Performed

The algorithm outputs: **“Please enter the limiting velocity range (pixels/s): ”**, and the input value will be added to the newly created list called “Limit”.

4.6 Loading the Difference Images

Before the algorithm can shift and stack the difference images, they have to be loaded into python. To load an image into python, the file of the image has to be known. Since the difference image file names are the same as the original image file names with `conv_` at the front, the difference images can easily be loaded using a list of original file names.

4.6.1 Creating the List

To create a list of all the original images in the current working directory, the algorithm finds the name of the current working directory and calls it *ObjectName*. The file names of all the files in the current working directory that have a file name starting with *ObjectName* and ending with `.fits` are added to a list that will be referred to in the rest of the thesis as the List.

Whenever the phrase ‘the difference image in the List’ is mentioned from now on, what is actually meant is ‘the difference image of an original image in the List’.

⁴These velocity ranges were chosen due to the fact that to most of the asteroids from the data supplied by Alan Gilmore and Pam Kilmartin had a speed of a few arcseconds per minute.

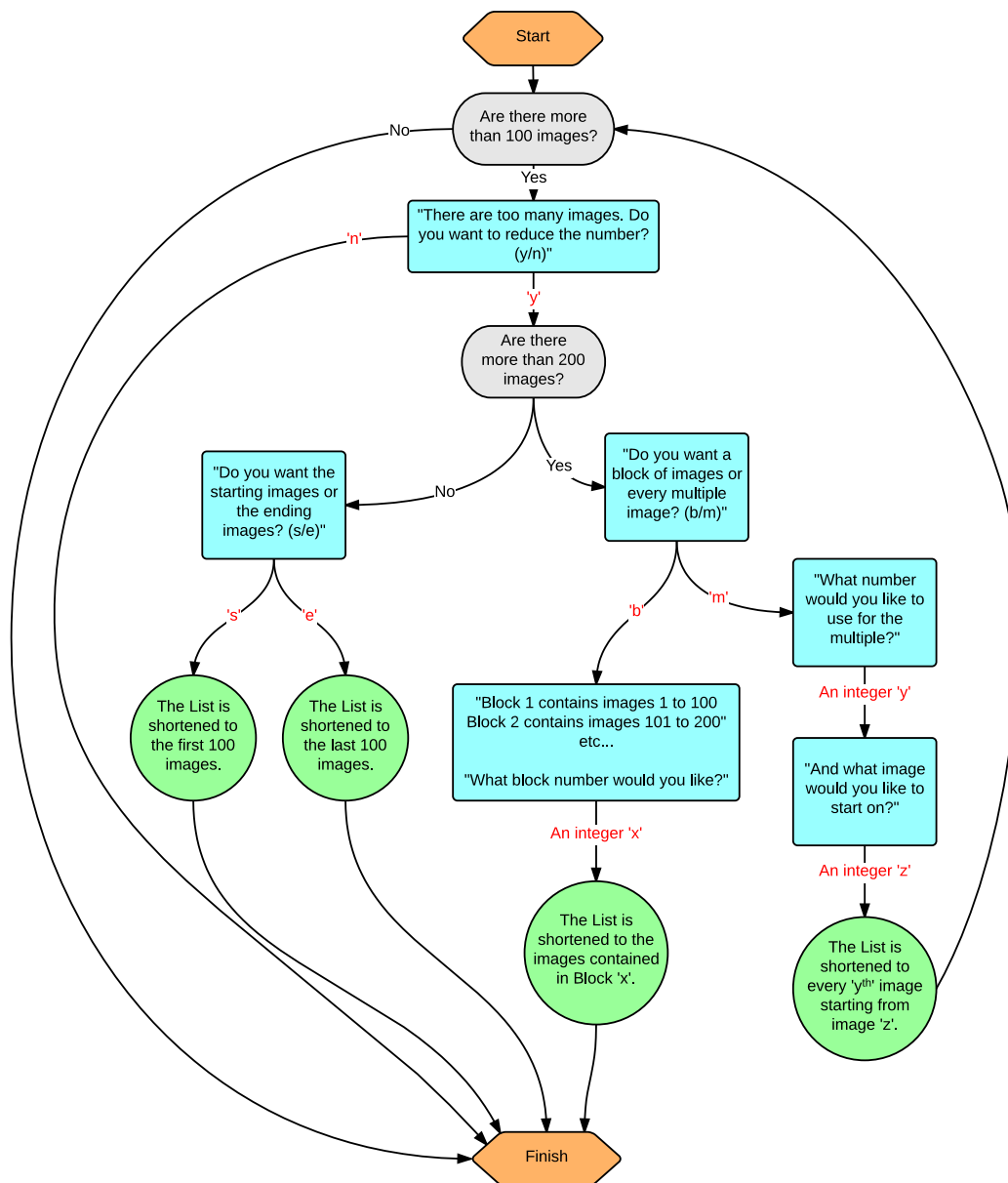


Figure 4.2: A flow chart of how the algorithm deals with the length of the List. Blue boxes contain algorithm outputs. Red letters indicate algorithm inputs. Green circles contain algorithm tasks. Grey boxes are decision points that the algorithm faces.

Some of the slower search velocities are in the area of the velocity phase space that is

removed, see Section 4.8.10. Asteroids with these velocities cannot be detected. This can be remedied by setting the variable *Half*, see Section 4.3, to *True*, which makes the algorithm remove every second image from the List⁵.

The increase in signal-to-noise drops off exponentially as the number of images being stacked increases. Therefore the increase in signal-to-noise for stacking in excess of 100 images is negligible. The algorithm follows the flow chart in Figure 4.2 to try to reduce the number of images in the List if there are more than a hundred of them.

4.6.2 Loading the Difference Images and Removing the Infinite Border

The algorithm now loads the difference images into python. This is achieved by using the `fits.getdata()` command on images that have `conv_` added to the original image file names that are in the List. The images are represented as a two-dimensional matrix of pixel values, and the algorithm combines all the difference image matrices to create a three-dimensional matrix where the image number is the third dimension.

Along the edge of the difference images there is an border of infinite values. The number of how many pixels deep the border is varies from difference image to difference image. The algorithm finds the smallest infinite border of the difference images on the List and removes a border that is the same size as the smallest from all the difference images on the list, i.e. decreases the first two dimensions of the difference image matrix. Then, for the difference images that have an infinite border that is larger than the smallest infinite border, the algorithm replaces the rest of the infinite border with a border of zero values.

4.7 Masking

Ideally, a difference image would only contain moving objects (like asteroids) and noise. Unfortunately this is not the case; difference imagery, and imagery in general, is not perfect. As can be seen in Figure 2.2, difference images of star fields can contain a large number of imperfections. A mask for the difference images has to be created so that any imperfections that could potentially overshadow the signal of an asteroid are removed.

Once the algorithm creates the List, it then produces a mask for each difference image in the List. The masks are matrices that are the same size as the difference images where each element is either a one or a zero. A zero means the same element on the difference image is masked and a one means it is not. To perform the masking the difference image

⁵This only decreases the size of the removed velocity phase space, so some velocities will still not be able to be tested

4 The Algorithm

matrix is multiplied by the mask matrix (element multiplication not matrix multiplication).

Most stars, like the star in Figure 4.3, are not completely removed from their difference images. Residuals of the stars still remain, as can be seen in Figure 4.4. The brighter the star the larger the residuals. Therefore bright stars are masked, see Section 4.7.1.

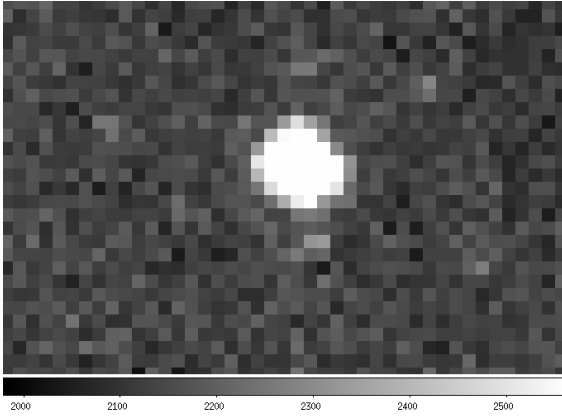


Figure 4.3: A bright star from a section of an original image that contains asteroid 2014 CJ13 on 26/02/2014.

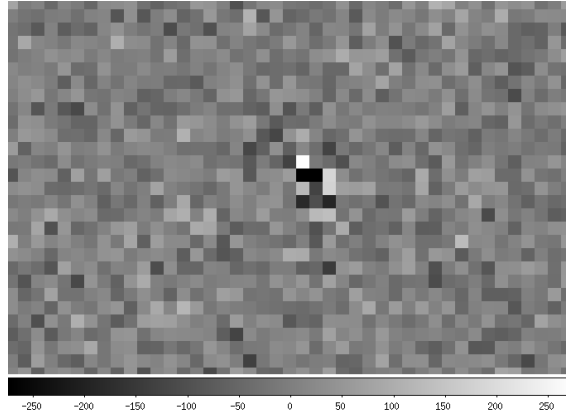


Figure 4.4: The difference image of the image in Figure 4.3.

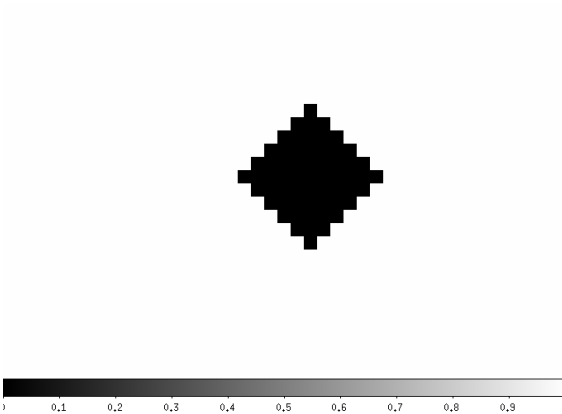


Figure 4.5: The mask created by the algorithm to mask the star residuals in the difference image in Figure 4.4, where black pixels indicate which areas are to be masked. The Star Mask Scaler was set to 2.

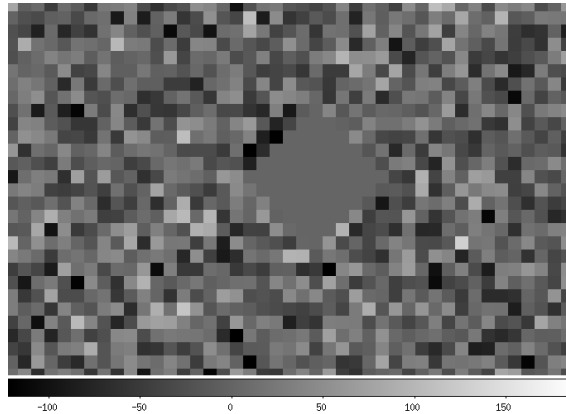


Figure 4.6: The difference image from Figure 4.4 with the mask from Figure 4.5 applied to it.

Saturated stars, like the star in the middle of Figure 4.7, do not have a well defined point spread function. Therefore they cannot be correctly identified, so there is a high chance that they will not be properly masked, as can be seen in Figure 4.8. Hence a different method is used to mask saturated stars, which is explained in Section 4.7.2. The mask created by

this method for the saturated star in Figure 4.7 is shown in Figure 4.9. When this mask is applied to the difference image, the saturated star residuals are properly masked, as shown in Figure 4.10.

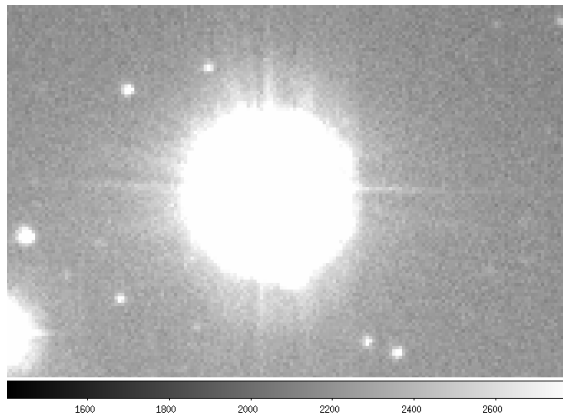


Figure 4.7: A saturated star in a section of an original image that contains asteroid 2014 JK25 on 30/05/2014.

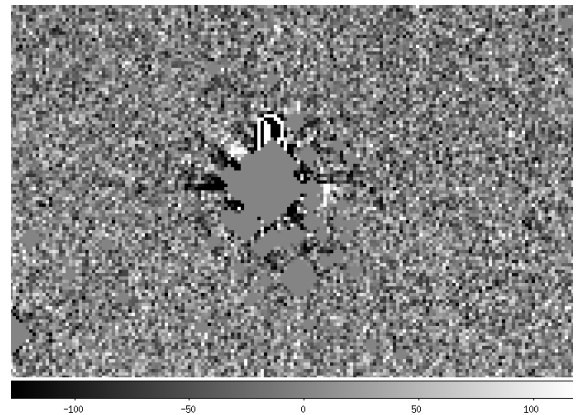


Figure 4.8: The difference image of the original image in Figure 4.7 that has been masked by the bright star mask.

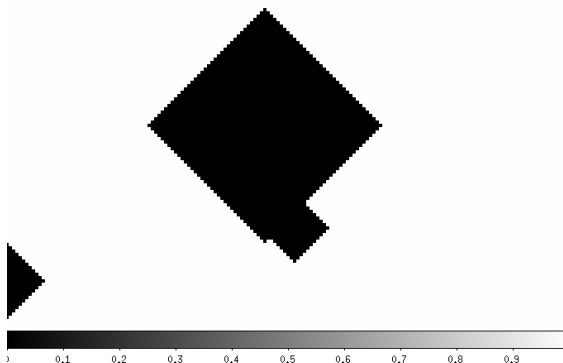


Figure 4.9: The saturated star mask created by the algorithm for the middle saturated star in Figure 4.7, where black pixels indicate which areas are to be masked. The Saturated Star Mask Scaler was set to 3.

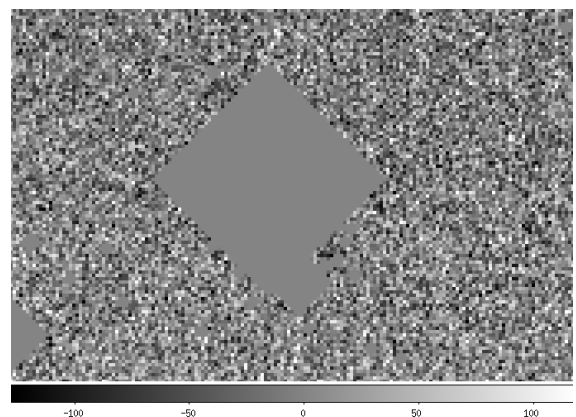


Figure 4.10: The image from Figure 4.8 with the mask from Figure 4.9 applied to it.

The bright star mask and the saturated star mask are combined, by multiplying (element by element) the two mask matrices, to produce a single star mask. Since the star mask matrix was produced from the reference and saturated image matrices, it does not have the same dimensions as the difference image matrices produced in Section 4.6.2. To make them have the same dimensions, a border the same size as the border removed from the difference images is removed from the star mask matrix.

4 The Algorithm

A very bright star that is just outside the field of the image can affect the image by producing what looks like a “flare”, as can be seen in Figure 4.11. On the difference images, these “flares” cause a number of high negative values that are concentrated in one area. These kind of areas, and others like it, will be referred to as a poor quality sections. The bright and saturated star masks cannot usually cover poor quality sections, as can be seen in Figure 4.12, so a separate method is used to mask them.

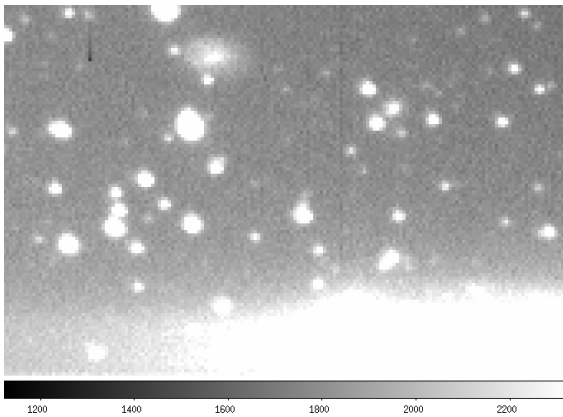


Figure 4.11: A ‘flare’ from an original image. This is a section of an image that contains asteroid 2014 HN178 on 30/05/2014.

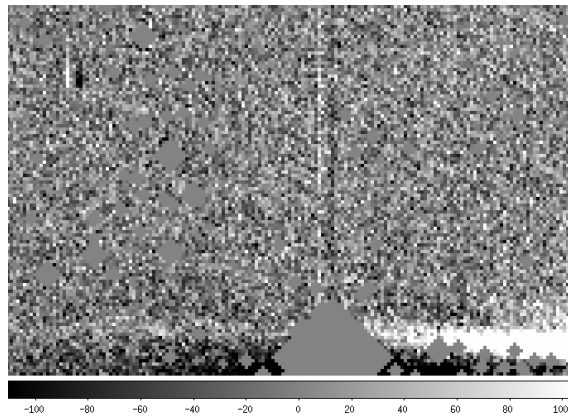


Figure 4.12: The difference image of Figure 4.11 that has been masked by the bright and saturated star masks. The poor quality section at the bottom is caused by the ‘flare’.

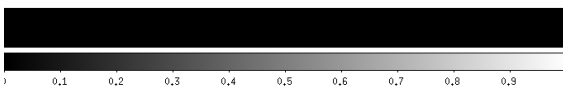


Figure 4.13: The mask created by the algorithm for the Poor Quality Section in Figure 4.12, where black pixels indicate which areas are to be masked. The Poor Quality Section Mask Modifier was set to 1.5.

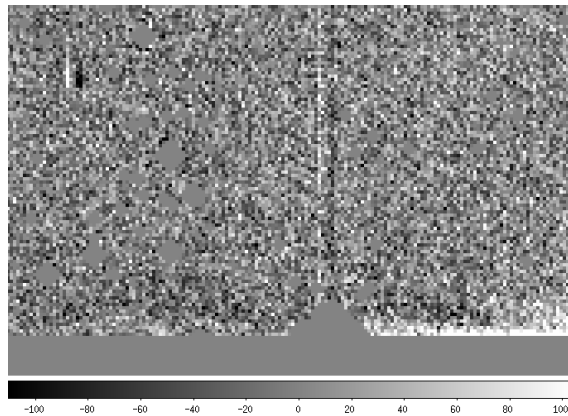


Figure 4.14: The image from Figure 4.12 with the mask from Figure 4.13 applied to it.

If one or more of the difference images in the List have a poor quality section, then the image is expected to have a number of highly negative values after the star mask is applied.

To decide whether to search for poor quality sections the algorithm calculates the number of pixels in each star masked difference image in the List that is more than High Value Number, see Section 4.3, of standard deviations away from the mean pixel value. These highly negative values will from now on be referred to as High Values ⁶. If the image with the highest number of High Values has more than 50 High Values, then that image is searched for Poor Quality Sections, see Section 4.7.3.

The poor quality section mask created after the algorithm was run on the image in Figure 4.12 is shown in Figure 4.13. When this mask is applied to the difference image, the poor quality section is properly masked, as can be seen in Figure 4.14.

The algorithm then identifies any poor quality images using the function described in Section 4.10.3, and removes any of these images from the List.

Some imperfections only affect a few pixels. These include, but are not limited to, cosmic rays that were missed in the cosmic ray removal process, faulty pixels in the CCD, and very faint stars that only appear in the stacked image with zero shifting. The algorithm tries to remove these imperfections using the High Value Mask, see Section 4.7.4, using the First High Value Mask Limit (see Section 4.3).

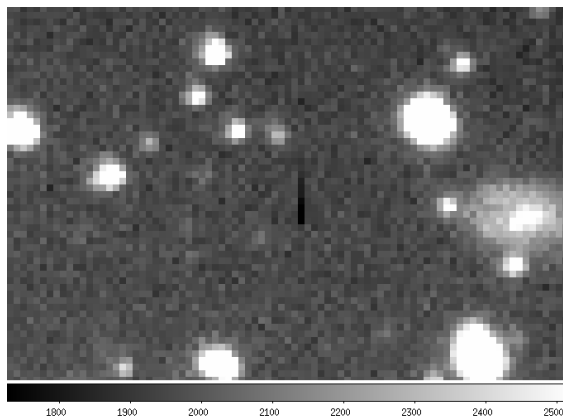


Figure 4.15: Part of an image that contains an imperfection (the black line in the centre). This is a section of an image that contains asteroid 2014 HN178 on 30/05/2014.

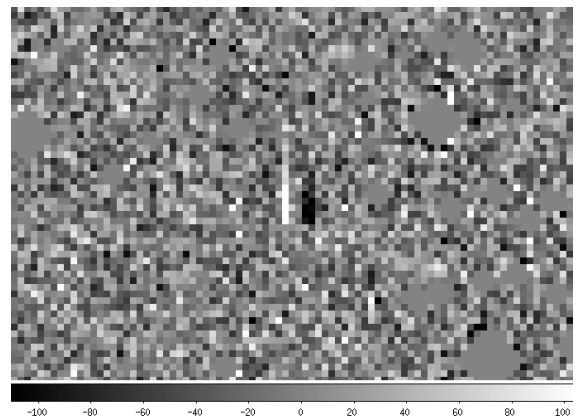


Figure 4.16: A difference image, where Figure 4.15 is the reference image, that has been masked by the star and poor quality section masks. There is a group of High Values (black) in the middle.

⁶These High Values are slightly different to the ones in Section 4.7.4.

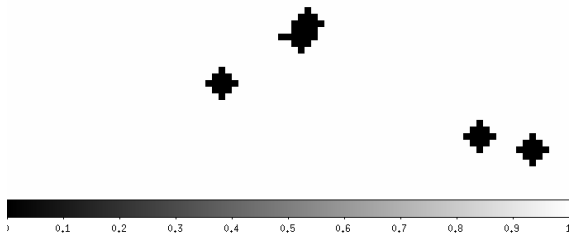


Figure 4.17: The mask created by the algorithm for the High Values in the image in Figure 4.16.

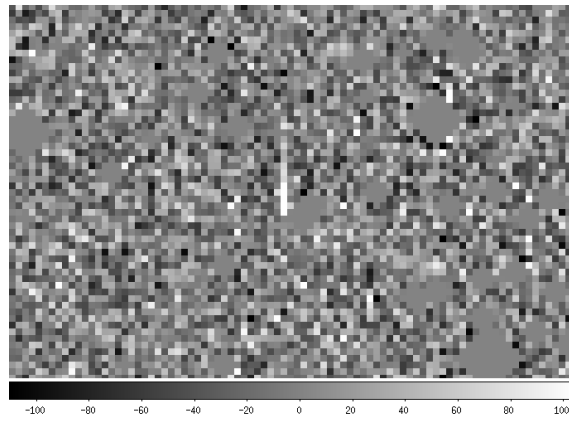


Figure 4.18: The image in Figure 4.16 with the mask from Figure 4.17 applied to it. The black pixels seen in Figure 4.16 have disappeared.

To be able to search for a second asteroid if one is found, the found asteroid has to be masked. The algorithm creates the Asteroid Mask by making a matrix that is the same size as the other mask matrices and all the elements are one. Nothing is added to the Asteroid Mask until the searching phase finds an asteroid. For details on how asteroids are masked see Section 4.8.12.

4.7.1 Bright Star Mask

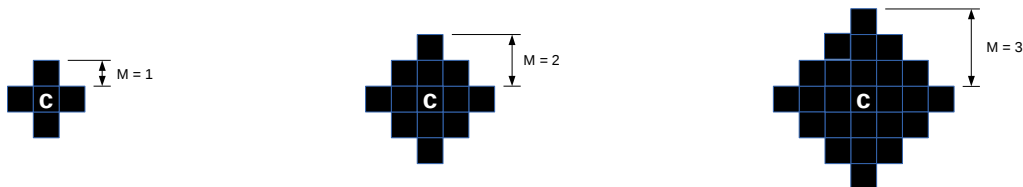


Figure 4.19: Three examples of the diamond-shaped mask used to remove the residuals of stars in difference images. The mask size, M , is the number of pixels the mask extends to from the centre straight up (or down, left or right, since the mask shape is symmetrical). The central pixel has a white C in it.

The magnitude and pixel position of all the stars in the field of view of the reference image that are brighter than 16^{th} magnitude are loaded into python from the `Mag.txt`, `Xvalue.txt`, and `Yvalue.txt` files, respectively. For each star a diamond-shaped mask, see Figure 4.19, is produced that is centred on the star's pixel position. The size of each mask depends on how bright the star is and on the seeing of the image. The mask size for a star is the rounded

value of 18 minus the stars magnitude. If the star is brighter than 13th magnitude, then the mask size is increased by two. If the mask size is less than two, then the mask size is increased to two. For each image the mask size of each star is then multiplied by the seeing of the image divided by the Star Mask Scaler, see Section 4.3, and then rounded.

To produce the bright star mask, first the algorithm creates a matrix that has the same dimensions as the reference image matrix and each element is one. Then all the corresponding pixels on the star mask matrix that the masks would cover on the images are replaced by zeroes.

4.7.2 Saturated Star Mask

To mask the saturated stars properly, first the `Saturate.fits` image is loaded into python. Then the algorithm finds the positions of all the pixels in the saturated image that have value greater than or equal to the Saturation Count, see Section 4.3. It then puts these pixels into groups depending on if they are “connected” to one another. If a pixel is adjacent to one of the pixels in a group (one above, below, left or right of it), then that pixel is put into that group. If a pixel is not next to any of the pixels in any of the groups, then that pixel is put into a newly created group. Each group is assumed to be a single saturated star and the number of pixels in each group is proportional to the size of the saturated star. Therefore the size of the mask for each saturated star should depend on the size of its saturated pixel group, N_{Sat} :

If $N_{Sat} > 200$: Mask size = $\sqrt{N_{Sat}} + 50$

If $200 \geq N_{Sat} > 50$: Mask size = $0.3N_{Sat} + 5$

If $50 \geq N_{Sat} > 20$: Mask size = $0.2N_{Sat} + 12$

If $20 \geq N_{Sat} > 10$: Mask size = $0.1N_{Sat} + 12$

If $10 \geq N_{Sat}$: Mask size = 13

These values were initially chosen at random, then altered as they were tested on saturated stars. For each saturated star in each image the mask size is then multiplied by the seeing of the image divided by the Saturated Star Mask Scaler, see Section 4.3, and then rounded. The saturated star mask is created in the same way as the bright star mask.

4.7.3 Poor Quality Section Mask

The algorithm prints “**Trying to locate the poor quality section in image:** L_{PQ} ”, where L_{PQ} is the file name of the original image whose difference image is being searched

4 The Algorithm

for a poor quality section. This difference image has the most High Values out of all the difference images. A trial mask that is the same size as the star mask is created. Using a *while* loop, while the difference image of L_{PQ} multiplied by its star mask and the trail mask has more than 50 High Values, the algorithm executes the following next two paragraphs.

The algorithm creates a list of the sections that might be a bad section. Initially the list only contains the whole difference image. Going through the whole section list 25 times, or until all the sections in the section list do not change over an iteration, the algorithm follows the flow chart in Figure 4.20. Once that is completed, if a section on the section list is more than 25 pixels and less than 1000 pixels then it is said to be a poor quality section, and all the elements in that section on the trail mask are turned into zeros. The edges of the poor quality section are put into a list and the algorithm outputs:

“Poor quality section located:

size = S

x = $x_{bottom}:x_{top}$

y = $y_{bottom}:y_{top}$ ”

Here S is the number of pixels of the poor quality section, x_{bottom} and x_{top} are the columns of the difference image that are the top and bottom columns of the poor quality section, respectively. Similarly for y_{bottom} and y_{top} and the rows of the difference image. If the trial mask does not change over an iteration, which is to say no new poor quality sections were located, then the algorithm outputs **“Unable to locate any poor quality section”** and the while loop is broken.

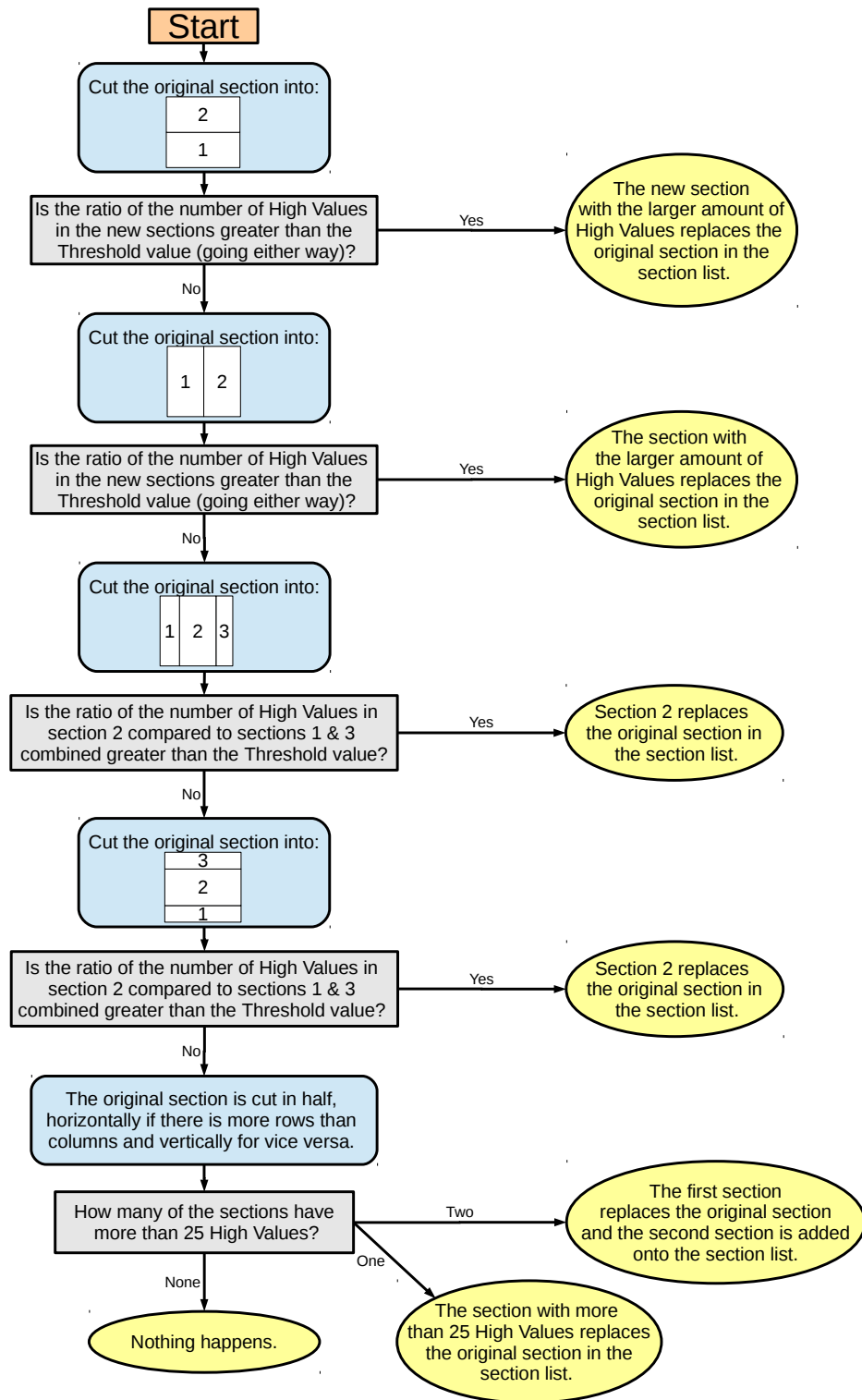


Figure 4.20: The algorithm follows this flow chart for the sections that might contain a poor quality section. Blue boxes are tasks the algorithm performs. Grey boxes are decision points that the algorithm faces. Yellow ovals state how the section list changes.

Once the *while* loop has finished the algorithm makes the poor quality section mask.

4 The Algorithm

Using the lists of the poor quality section edges, the size of the poor quality sections is increased. If the number of rows is larger than the number of columns, then the poor quality section edges are increased to

$$x_{bottom}^* = x_{bottom} - \left\lfloor \frac{0.75(x_{top} - x_{bottom})}{P} \right\rfloor, \quad (4.2)$$

$$x_{top}^* = x_{top} + \left\lfloor \frac{0.75(x_{top} - x_{bottom})}{P} \right\rfloor, \quad (4.3)$$

$$y_{bottom}^* = y_{bottom} - \left\lfloor \frac{(y_{top} - y_{bottom})}{P} \right\rfloor, \quad (4.4)$$

$$y_{top}^* = y_{top} + \left\lfloor \frac{(y_{top} - y_{bottom})}{P} \right\rfloor, \quad (4.5)$$

where an asterisk indicates the new value and P is the Poor Quality Section Mask Modifier, see Section 4.3. If the number of rows is less than the number of columns, then the poor quality section edges are changed by

$$x_{bottom}^* = x_{bottom} - \left\lfloor \frac{(x_{top} - x_{bottom})}{P} \right\rfloor, \quad (4.6)$$

$$x_{top}^* = x_{top} + \left\lfloor \frac{(x_{top} - x_{bottom})}{P} \right\rfloor, \quad (4.7)$$

$$y_{bottom}^* = y_{bottom} - \left\lfloor \frac{0.75(y_{top} - y_{bottom})}{P} \right\rfloor, \quad (4.8)$$

$$y_{top}^* = y_{top} + \left\lfloor \frac{0.75(y_{top} - y_{bottom})}{P} \right\rfloor. \quad (4.9)$$

If any of the edges lie outside the image, either below zero or above the maximum column/row number, then they are set as the image limit, either to zero or the maximum column/row number.

To make the poor quality section, a matrix that has every element as one and is the same size as one of the difference images is created. Using the increased poor quality section edges, all elements that are in areas of the poor quality section mask that the poor quality sections occupy on the difference image are given a value of zero.

4.7.4 High Values Mask

There are five different types of high values that the algorithm masks:

High negative values from the stacked image

If the minimum pixel value, μ_{stack} , of an image consisting of all the masked difference images, which have been blurred then stacked on top of one another with zero shifting, satisfies the equation

$$\mu_{stack} \leq 0.33N \sum_{i=1}^N \nu_i \quad , \quad (4.10)$$

where N is the number of difference images and ν_i is the minimum pixel value of the i^{th} blurred image, then the pixel in each image on the List that has the same coordinates as the minimum pixel value of the stacked image is the centre for a diamond mask, see Figure 4.19, with a mask size of 2.

High values from the absolute stacked image

If the maximum pixel value, $\mu_{abstack}$, of an image consisting of all the masked difference images, for which the absolute value of all the pixels has been taken and which then been stacked on top of one another with no shifting, satisfies the equation

$$\mu_{abstack} \geq 0.5N \sum_{i=1}^N n_i \quad , \quad (4.11)$$

where n_i is the maximum pixel value of the i^{th} absolute image, then the pixel in each image on the List that has the same coordinates as the maximum pixel value of the absolute stacked image is the centre for a diamond mask, see Figure 4.19, with a mask size of 2.

High negative values from a single image

If the minimum pixel value of a masked difference image, μ_i , satisfies the equation

$$\mu_i \leq \bar{\mu} - 1.5\sigma_{mu} \quad , \quad (4.12)$$

4 The Algorithm

where $\bar{\mu}$ is the mean of the minimum pixel values of all the masked difference images, and σ_{mu} is the standard deviation of the minimum pixel values of all the masked difference images, then the minimum-value pixel gets masked.

High positive values from a single image

If the maximum pixel value of a masked difference image, m_i , satisfies the equation

$$m_i \geq \bar{P} + 10\sigma_P \quad , \quad (4.13)$$

where \bar{P} is the mean of all the pixel values of that image, and σ_P is the standard deviation of all the pixel values of that image, then the maximum-value pixel gets masked.

Super high negative values from a single image

If the minimum pixel value of a masked difference image satisfies the equation

$$\mu_i \leq \bar{P} - 25\sigma_P \quad , \quad (4.14)$$

then the minimum-value pixel gets masked.

To create a High Value mask, two mask matrices are produced the same way the other mask matrices are made. Mask matrix one is a two-dimensional matrix and mask matrix two is a three dimensional matrix with the third dimension being the number of images in the List. Essentially mask matrix one is one mask for all the difference images and mask matrix two is a group of masks with each difference image having its own individual mask.

The algorithm masks the difference images with the Star, Poor Quality Section and the two newly created High Value masks. For each image, if any of the pixels are “Super High Negative Values From a Single Image” or “High Positive Values From a Single Image” then the masks for those pixels are added to mask matrix two. The algorithm then searches for “High negative values from a single image” for all the images, and the masks for those pixels are added to mask matrix two if any are found. If one or more are found, the algorithm remasks the difference images using the same masks and searches again for those three High Value types. This process is repeated until no more “High Negative Values From a Single Image” are found.

The algorithm produces two stacked images, one with the masked images that have been blurred using Section 4.10.1 with *Sigma* values from Section 4.10.4, and another with the masked images where each pixel is replaced with its absolute value. Both stacking have zero shifting. The algorithm then searches for a “High Negative Value From the Stacked Image” and a “High Value From the Absolute Stacked Image” in the blurred stacked image and absolute stacked image, respectively. If either one is found, they are masked in mask matrix one.

The algorithm repeats the process described in the two paragraphs above until no more “High Negative Value From the Stacked Image” or “High Value From the Absolute Stacked Image” are found, or the number of “High Negative Values From the Stacked Image” or “High Values From the Absolute Stacked Image” becomes greater than the high value mask limit.

Once all the high values have been found, the algorithm combines the two mask matrices by multiplying mask matrix one with every two-dimensional slice of mask matrix two which represents the mask for a single difference image. This new mask will be known as the High Value Mask. The algorithm then outputs the numbers of each type of high values it has found.

4.8 Search for Asteroids

Now that the masks have been created for the difference images, the search for asteroids can commence. For every velocity range limit in the list created in Section 4.5.2, the algorithm produces a single mask, which will be known as the Total Mask, by multiplying all the masks produced in Section 4.7; Star Mask, Poor Quality Section Mask, first High Value Mask, and the Asteroid Mask⁷. Then it calls the current velocity range limit L , creates a new string *MoreCleaning*, which is equal to zero, and follows the flow chart in Figure 4.21.

⁷Since the Asteroid Mask will change if an asteroid is found, the algorithm will always use the up to date Asteroid Mask when creating the Total Mask.

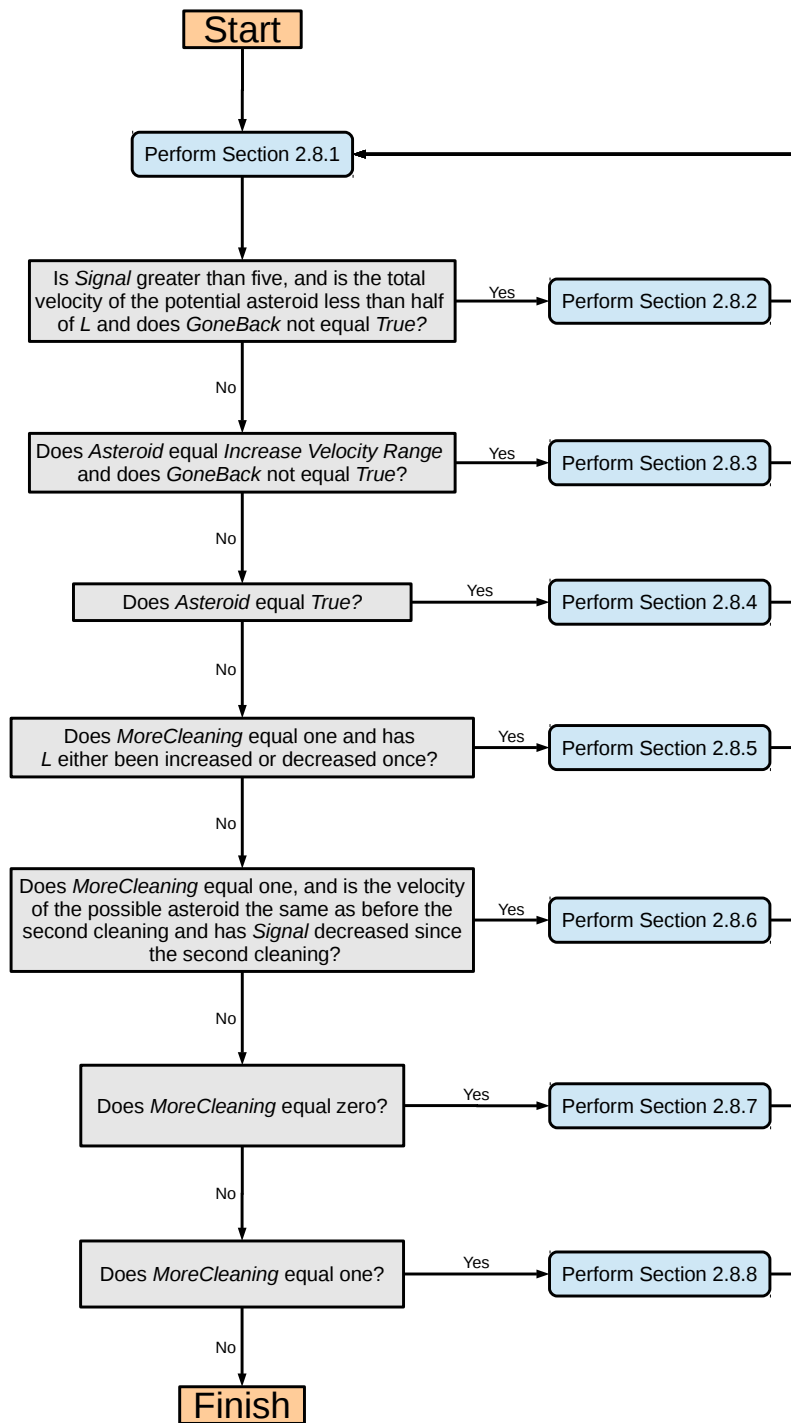


Figure 4.21: A flow chart of how the algorithm searches for asteroids for a particular velocity range. Blue boxes are tasks the algorithm performs. Grey boxes are decision points that the algorithm faces.

4.8.1 Search

To increase the chances of an asteroid being detected, the masked difference images are stacked in such a way that an optimally stacked difference image is produced. Details on how to make an optimally stacked image are given in Section 1.6.

Since a large number of stacked images is produced⁸, doing a detailed analysis of each stacked image would be too time consuming. Instead, the minimum pixel value of each stacked image is used to create a two-dimensional matrix, where the elements of the matrix correspond to a certain shifting velocity. This matrix will be known as the velocity phase space matrix.

The velocity phase space matrix is created as described in Section 4.8.9 with both V_{max}^x and V_{max}^y set to L , and V_{min}^x and V_{min}^y set to $-L$, Len set to the Velocity Vector Length (see Section 4.3), and the input mask is the Total Mask. The function outputs are not renamed but the last S_{op} is not used.

If the difference images are masked properly then they should only contain moving objects and noise. The minimum pixel value of the stacked images should be fairly similar for different shifting velocities, unless the shifting velocities match up with the velocity of an asteroid that is in the images. This should cause a large spike in the minimum pixel value at this velocity due to the asteroid in each image aligning with itself. Unfortunately, there is always a spike at the origin due to very faint stars that are hard to mask. To make sure this spike at the origin does not interfere with the detection of the asteroid's spike, the origin of the velocity phase space diagram is removed.

The removal of the origin in the velocity phase space diagram is achieved by replacing all the elements of the velocity phase space matrix that are within a certain distance from the origin with an average of the values at that distance. This distance will be the shift velocity that corresponds to a shift between two consecutive images of one pixel on average.

In the velocity phase space diagram shown in Figure 4.22, the signal spike due to the asteroid is overshadowed by the signal spike at the origin, so the detection method used by the algorithm will not be able to detect the asteroid. If the spike at the origin is removed, as shown in Figure 4.23, the signal spike due to the asteroid is now the largest negative value in the diagram and now has a chance of being detected.

⁸The number of stacked images produced depends on the size of the Velocity Vector Length (see Section 4.3), but a small Velocity Vector Length would decrease the chance of an asteroid being detected compared to a large Velocity Vector Length.

4 The Algorithm

The algorithm uses the Origin Removal function described in Section 4.8.10 to remove the origin of the newly created velocity phase space matrix. The input to this function is M_{VS} as M , V^x and V^y as V^1 and V^2 , respectively, and X_{marker} as R . The function output M_{OR} is renamed to M_{ORVS} .

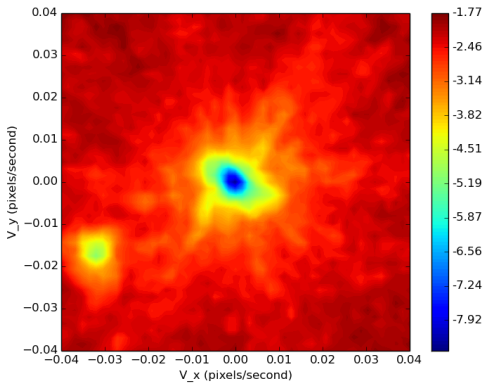


Figure 4.22: A diagram of the velocity phase space of a group of images that contain the asteroid 2014 CJ13 on 26 February 2014.

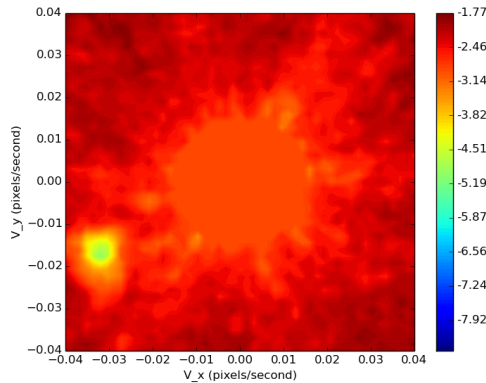


Figure 4.23: The diagram in Figure 4.22 with the peak at the origin removed.

With the origin removed from the velocity phase space matrix the only signal spike should be due to an asteroid and the spike should be the most negative value of the matrix. Therefore to tell if there is an asteroid in the images, the minimum value of the velocity phase space matrix should be compared with other values in the matrix. As can be seen in Figure 4.22, there is a trend of the minimum pixel value increasing radially going out from the origin. Therefore the minimum value in the matrix should be compared with values that are the same radius from the centre, but not too close to the minimum value in case the asteroid has an effect on that value. If the minimum value is eight standard deviations from the mean of the comparing values, then the minimum value is said to be caused by an asteroid, so an asteroid is detected.

The Asteroid Detection function in Section 4.8.11 is used to find out whether the minimum value of M_{ORVS} passes the criteria to be called an asteroid. The input to this function is M_{ORVS} as M_{Det} , V^x and V^y as V^1 and V^2 , respectively. The function outputs are not renamed.

4.8.2 Decreasing Velocity Range Limit

The velocity of the possible asteroid relative to L is low such that the resolution around the velocity is not high enough to be able to get an accurate enough result from the detection method described in Section 4.8.11. Therefore L needs to be lower in order to be able to successfully determine whether the possible asteroid meets the criteria of being an asteroid.

The algorithm outputs **“Possible asteroid’s velocity is too low.”** and the L is decreased so that it is equal to one and a fifth of the total velocity

$$L = 1.2\sqrt{(V_{PA}^x)^2 + (V_{PA}^y)^2} \quad . \quad (4.15)$$

The algorithm then outputs **“Search velocity range decrease to: L ”**.

4.8.3 Increasing Velocity Range Limit

The velocity of the possible asteroid is too high, relative to L , such that the detection method described in Section 4.8.11 can either not be used or there are not enough points to produce an accurate interpolation of the test arc. Therefore L needs to be raised in order to be able to successfully determine whether the possible asteroid meets the criteria of being an asteroid.

L is increased by a factor of one and a fifth and **“Search Velocity range increased to: L ”** is printed in the terminal.

4.8.4 Asteroid Found

An asteroid has been identified! The positions and velocity of the asteroid should be saved and, in case there is another, the algorithm should be able to search for more asteroids in the images.

To be able to search for another asteroid when one has been detected, the detected asteroid must be masked from the difference images. To do this first the algorithm finds the position of the minimum pixel value of the correctly stacked image by using the Creation of the Velocity Phase Space Matrix described function in Section 4.8.9 to create the correctly stacked image (the last S_{op} produced). The input for the function for both V_{min}^x and V_{max}^x is V_{PA}^x , similarly both V_{min}^y and V_{max}^y is V_{PA}^y , Len is set to 1, and the mask used is the Total Mask. Using the Asteroid Masking function in Section 4.8.12, with V_{PA}^x and V_{PA}^y set to V_A^x

and V_A^y , respectively, and I_{cs} set to S_{op} , the mask for the asteroid is added to the Asteroid Mask. The total mask is remade by multiplying the Star Mask, Poor Quality Section Mask, the first High Value Mask and the Asteroid Mask together. *MoreCleaning* is reset to zero.

Asteroid_Count is increased by one and the algorithm runs the Asteroid Positions Function, see Section 4.8.13, with P^x and P^y as A^x and A^y , respectively and A_{Num} set to *Asteroid_Count*. This function outputs in the terminal and writes to a file the asteroid's position in each image.

4.8.5 Resetting Velocity Range Limit

If the velocity range limit was changed, by Section 4.8.2 or Section 4.8.3, and no asteroid was found then L is reset back to the original velocity range limit and the algorithm outputs **“Changing the velocity limit did not work, changing it back.”**. *GoneBack* is set to *True* so that the algorithm knows that the velocity range limit was changed but now it is back to its original limit.

4.8.6 Signal Decrease of Possible Asteroid

If the velocity of the possible asteroid has not changed since the first set of cleaning, described in Section 4.8.7, and *Signal* has decreased, then it is possible that the first set of cleaning could have, partially or fully, masked some of the images of the asteroid on the difference images. That could mean that the High Value mask, described in Section 4.7, also masked some of the images of the asteroid. Therefore redoing the High Value Mask with a smaller High Value Mask Limit could reveal the presence of the asteroid.

The algorithm prints **“Too many high values removed. Redoing high value mask.”** and a new High Value Mask is made as described in Section 4.7.4 where the Star Mask multiplied by the Poor Quality Section Mask is used as the input mask and the High Value Mask Limit is set to 10.

The Total Mask is remade by multiplying the Star Mask, Poor Quality Section Mask, the new High Value Mask and the Asteroid Mask together. **“Average fraction of pixels masked: $\overline{M_T}$ ”** is printed in the terminal, where $\overline{M_T}$ is the mean of the Total Mask matrix. *MoreCleaning* is increased by 1, so if no asteroids are found in the next run, then the algorithm will stop looking. *GoneBack* is set to *False*.

4.8.7 Difference Image Cleaning

If no asteroids could be detected it is possible that there is an asteroid in the images but its signal is hidden. To reveal the signal the algorithm masks more High Values (Section 4.7.4) from the difference images, which will be termed as ‘cleaning’. If the first set of cleaning is unable to mask anything, then there is no point redoing the search for an asteroid. A new initial High Value Mask could reveal an asteroid.

If the first set of cleaning is yet to be performed and there is no second High Value mask, then one is created using Section 4.7.4 with the Total Mask being used as the input mask and the Second High Value Mask Limit, see Section 4.3, being the High Value Mask Limit. If nothing new is masked, then a new initial High Value Mask is made as described in Section 4.7.4 using the Star Mask multiplied by the Poor Quality Section Mask as the input mask and the High Value Mask Limit is set to 10. If the second High Value Mask masks at least one pixel, whether it has just been created or made earlier, then the Total Mask is multiplied by the second High Value Mask.

The algorithm outputs “**Average fraction of pixels masked: $\overline{M_T}$** ”, where $\overline{M_T}$ is the mean of the Total Mask matrix, and *MoreCleaning* is increased by one.

4.8.8 More Difference Image Cleaning

If no asteroids could be detected after the first set of cleaning, then maybe a second set of cleaning will reveal the asteroid if it is there.

Just like the case with the first set of cleaning, if there is no third High Value mask then one is created using Section 4.7.4 with the Total Mask being used as the input mask and the Second High Value Mask Limit, see Section 4.3, being the High Value Mask Limit. If nothing was masked then the algorithm stops the loop shown in Figure 4.21. If the second High Value Mask masks at least one pixel, whether it has just been created or made earlier, the Total Mask is multiplied by the third High Value Mask.

The algorithm outputs “**Average fraction of pixels masked: $\overline{M_T}$** ”, where $\overline{M_T}$ is the mean of the Total Mask matrix, and *MoreCleaning* is increased by one.

4.8.9 Function: Creating the Velocity Phase Space Matrix

The input to this function is the minimum x and y velocity, V_{min}^x and V_{min}^y , respectively, the maximum x and y velocity, V_{max}^x and V_{max}^y , respectively, the length of each velocity vector,

4 The Algorithm

l , and the input mask.

For each image on the List, the algorithm loads the file that has `interp_` followed by the images filename. These are the original images that have been shifted to align with the reference image and they will be known as the interp images. If the interp image contains a pixel that has a value less than zero, then the value is increased to zero. The algorithm then removes the image border so that it is the same size as the difference images, see Section 4.6.2. These images are combined to create a single three-dimensional matrix.

The difference image matrix is multiplied by the total mask matrix and then divided by the interp image matrix. This new matrix is called the A matrix. Another matrix, B matrix, is created by dividing the total mask matrix by the interp image matrix. If there are any NaN or infinite elements in either A or B , they are replaced by zeros. Both A and B matrices are blurred using the blurring function described in Section 4.10.1 with $Sigma$ values from Section 4.10.4.

If l is equal to one then V_{step}^x and V_{step}^y are set to zero. If l is greater than one the algorithm prints **“Shifting and stacking difference images. I may be some time...”** and creates the x and y velocity steps, V_{step}^x and V_{step}^y , respectively, using the equations

$$V_{step}^x = \frac{V_{max}^x - V_{min}^x}{l - 1} \quad , \quad (4.16)$$

$$V_{step}^y = \frac{V_{max}^y - V_{min}^y}{l - 1} \quad . \quad (4.17)$$

Two velocity vectors, V^x and V^y , are created where the i^{th} element of the velocity vectors are

$$V_i^x = V_{min}^x + (i - 1)V_{step}^x \quad , \quad (4.18)$$

$$V_i^y = V_{min}^y + (i - 1)V_{step}^y \quad , \quad (4.19)$$

where i ranges from 1 to l .

For all the combinations of elements of V^x and V^y as the stacking velocities, starting from the first element of both, the blurred A and B matrices are turned into stacked matrices, S_A and S_B , respectively, using Section 4.10.5. From S_A and S_B and the gain, g , the optimal

stacked image matrix, from Equation 1.10, for a particular pair of stacking velocities, S_{op} , is created as

$$S_{op} = \frac{\sqrt{g}S_A}{\sqrt{S_B}} \quad . \quad (4.20)$$

The algorithm removes the elements of S_{op} that are not a combination of all the images. All the values contained in the border of S_{op} that has a length equal to the second highest absolute value of Δx_i value for the horizontal borders, and similarly Δy_i for the vertical borders, are changed to zero. Also, any NaN elements of S_{op} are replaced by a zero.

The lowest element value of each S_{op} matrix is put into a two-dimensional matrix, where the element in which the lowest S_{op} value is inserted into depends on the componential shifting velocities. This matrix is the velocity phase space and will be known as the *Stack_Min* matrix.

While going through each combination of V^x and V^y , if the maximum absolute of value Δx_i is lower than the length of the Δx vector plus one, then the string *xmarker* is set to the element of V^x being used in the stacking.

The function outputs the *Stack_Min* matrix, the V^x and V^y vectors, *xmarker*, and the last S_{op} produced.

4.8.10 Function: Origin Removal

The inputs to this function are a two-dimensional matrix, M , the two vectors that represent the matrix coordinates, V^1 and V^2 , and the distance from the origin, R , which is to be removed, R .

To remove the origin the algorithm first creates a circle of 36 equally spaced points centred on the origin, the element of the matrix where V^1 and V^2 are both zero. The radius of the circle is R . Using the interpolation function `scipy.interpolate.interp2d()` the matrix values at all the points are calculated. Any element in the velocity phase space matrix that its inside the circle of points has its value replaced by the average value of the 36 points. The new matrix is called M_{OR} .

The output of this function is the M_{OR} matrix.

4.8.11 Function: Asteroid Detection

The inputs to this function are a velocity phase space matrix, M_{Det} , the two vectors that represent the velocity coordinates, V^1 and V^2

The algorithm finds the minimum value of M_{Det} , labels it as μ_O , and the x and y velocity that corresponds to it, V_{PA}^x and V_{PA}^y , respectively. It then calculates the total velocity of that point and the angle between the positive x velocity axis and the total velocity vector going clockwise. If the total velocity, V_{Tot} , is close enough to the Velocity Range Limit, Lim , that is if it satisfies this equation

$$V_{Tot} > Lim - \frac{2Lim}{Len - 1}, \quad (4.21)$$

where Len is the Velocity Vector Length, then the algorithm outputs **“WARNING: possible asteroid’s velocity is too high. Increasing the velocity range.”**. Then $Signal$ is set to zero and $Asteroid$ is set to to ‘*Increase Velocity Range*’. If Equation 4.21 is not satisfied then the algorithm creates an arc of 25 equally spaced points centred on the origin that has a radius of the total velocity and starts at an angle of $\phi + \frac{\pi}{3}$ and ends at $\phi + \frac{5\pi}{3}$. Using the same interpolation function that was used in Section 4.8.10, the minimum pixel value at all the points in the arc is calculated. Using the arc values and μ_0 , $Signal$ is calculated as

$$Signal = \frac{\overline{Arc} - \mu_O}{\sigma_{Arc}}, \quad (4.22)$$

where \overline{Arc} is mean of the arc values and σ_{Arc} is the standard deviation of the arc values. If $Signal$ is greater than 8 then the algorithm outputs **“We have an asteroid at $V_x = V_{PA}^x$, $V_y = V_{PA}^y$ ”** and assigns *True* to the variable $Asteroid$. If $Signal$ is between 5 and 8 then the algorithm outputs **“We might have an asteroid at $V_x = V_{PA}^x$, $V_y = V_{PA}^y$ ”** and assigns *False* to the variable $Asteroid$. If $Signal$ is less than 5 then the algorithm outputs **“No asteroid detected. $V_x = V_{PA}^x$, $V_y = V_{PA}^y$ ”** and assigns *False* to the variable $Asteroid$.

The outputs of this function are V_{PA}^x , V_{PA}^y , $Asteroid$ and $Signal$.

4.8.12 Function: Asteroid Masking

The inputs to this function are the asteroids componential velocity, V_A^x and V_A^y , the asteroid's correctly stacked image, I_{cs} .

The algorithm finds how many pixels each image was shifted by to get the correct stacked image. This is achieved by multiplying the asteroids velocities, V_A^x and V_A^y , with the time between acquisition of the image in focus and the first image on the List. If the velocity in one of the directions is negative, then the smallest shift amount (the shift amount of the last image on the list) will be subtracted from the shift amount of each image.

The position of the asteroid in correctly stacked image is calculated by finding the minimum value of I_{cs} . By subtracting the number of pixels an image was shifted by off the position of the asteroid in its correctly stacked image, the position of the asteroid in that image is found. For each difference image the asteroid position is the centre for a diamond mask, see Figure 4.19, with a mask size of 4.

The function outputs the positions of the asteroid in each difference image on the List as vectors A_D^x and A_D^y .

4.8.13 Function: Asteroid Positions

The inputs to this function are *Asteroid_Count* and the positions of the asteroid in each difference image on the List as vectors A_D^x and A_D^y .

The file *Asteroid_Positions.txt* is opened⁹ and if *Asteroid_Count* is equal to 1, then the algorithm will write “**Image Name:** *ObjectName*”, for *ObjectName* see Section 4.6.1. The algorithm writes “**Asteroid Number:** *Asteroid_Count*” two lines below what was last written in the file. On the next line the algorithm writes a table with four columns into the text file. The first two columns are “**Image**” and “**Time**”. Each filename in the List is added under the column “**Image**” and its corresponding time of acquisition, from the list *time*, is added under the column “**Time**”.

The next two columns depend on if astrometry has been successfully performed:

Astrometry HAS been successfully performed: The last two columns are “**RA (degrees)**” and “**Dec (degrees)**”. The algorithm uses the function `wcs_pix2world()` to turn the position of the asteroid in each difference image from pixel coordinates to Right Ascension and Declination coordinates and writes them under the “**RA (degrees)**”

⁹If there is no file *Asteroid_Positions.txt* in the current working directory, then one is created.

and “**Dec (degree)**” columns, respectively.

Astrometry HAS NOT been successfully performed: The last two columns are “**X Position (pixel)**” and “**Y Position (pixel)**” and the algorithm adds the x and y positions of the asteroid in each difference image under those two columns, respectively.

At the same time as the algorithm writes the asteroid’s position to the text file, it also outputs the positions to the terminal. First the algorithm outputs “**Coordinates of the Asteroid**” and then outputs the same table that was written in the `Asteroid_Positions.txt` file.

The componential velocities of the asteroid are calculated from the distance between the asteroid in the first and last difference image in the List and the difference in time between the first and last time of acquisition in the list *time*. The distance will be in arcseconds if astrometry was successfully performed and pixels if it was not. The total velocity, V_{Tot} is calculated by adding the componential velocities in quadrature. Using trigonometry, the angle between the positive Declination vector and the asteroid’s vector, θ , is found. The algorithm prints “**Asteroid Velocity: V_{Tot} ”/min @ θ° ” if astrometry was successfully performed and “**Asteroid Velocity: V_{Tot} pixels/min @ θ° ” if it was not.****

4.8.14 Function: Saving Images

The inputs for this function are the componential stacking velocities V^x and V^y , an input mask, and the correctly stacked image. This stacked image will be referred to as the difference stacked image.

For each image on the List, the algorithm loads the file that has `interp_` followed by the original image’s file name. These are the original images that have been shifted to align with the reference image. The algorithm then removes the border from these images so that it is the same size as the difference images, see Section 4.6.2. The algorithm combines the images into a single three-dimensional matrix and uses it as the input data for the Shifting and Stacking function described in Section 4.10.5 along with V^x and V^y as the velocities of the stacking. The output stacked image will be referred to as the original stacked image.

The algorithm adds a circle to both stacked image matrices that is centred on the indices of the asteroid. The circle is achieved by changing the pixel value of all the pixels that satisfy the two equations

$$\sqrt{x^2 + y^2} < r + 1 \quad , \quad (4.23)$$

$$\sqrt{x^2 + y^2} > r - 1 \quad , \quad (4.24)$$

where x is the number of pixels the pixel is away from the indices of the asteroid in the x direction, similarly for y , and r is the intended radius of the circle, 10 pixels. For the original stacked image the pixels that satisfy both Equations 4.23 and 4.24 are given a value of the Saturation Count, see Section 4.3. For the difference stacked image those pixels are given a value that is the negative of the lower pixel value of the difference stacked image.

The algorithm searches the current working directory and counts the number of files that start with `SnS_NormIm_A` and `SnS_DifIm_A`. If both counts are not the same, then the algorithm outputs **“WARNING: Saved stacked images do not match up.”** It then adds one to this count of files that start with `SnS_NormIm_A` and labels it A . The algorithm then saves the original stacked image with a file name of `SnS_NormIm_AA.fits` and the difference stacked image with a file name of `SnS_DifIm_AA.fits`. An example of an original and a difference stacked image is shown in Figure 4.24 and 4.25, respectively.

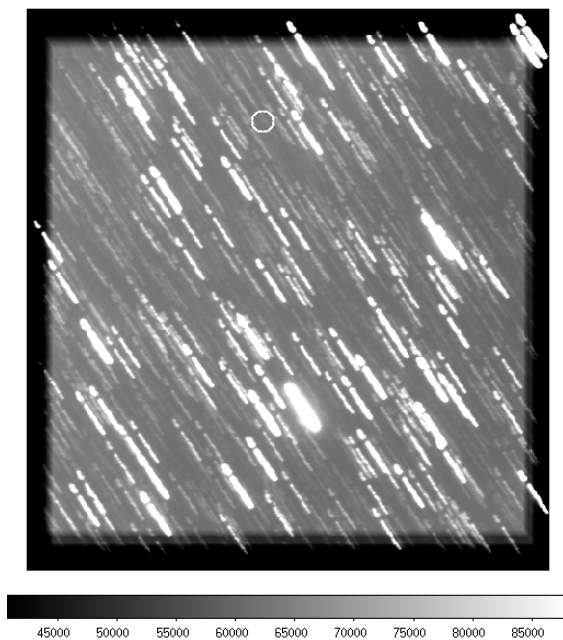


Figure 4.24: A stacked image that is comprised of the original images that are shifted relative to one another so that the images of the asteroid are aligned. The asteroid is circled in white.

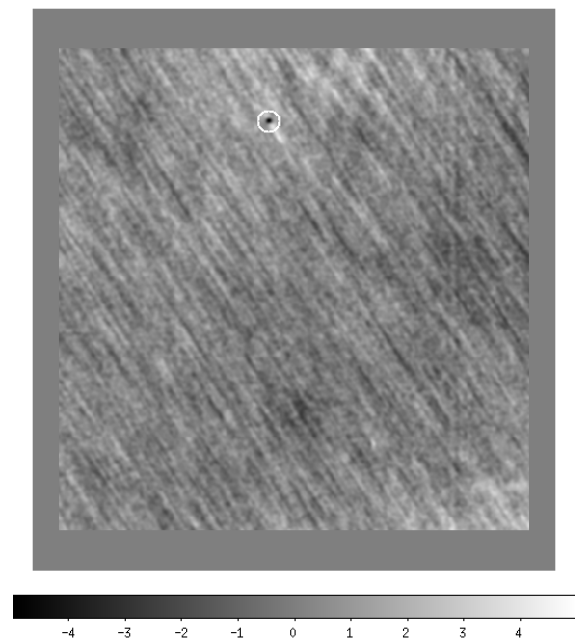


Figure 4.25: A stacked image that is comprised of the masked difference images that are shifted relative to one another so that the images of the asteroid are aligned. The asteroid is circled in white.

For each image on the List, the algorithm masks its difference image with its corresponding input mask, and saves the image as `AA_Masked_Original`. *Original* is the file name of the original image.

4.9 Output Files

4.9.1 Asteroid Positions

If one or more asteroids are detected, the algorithm writes the asteroids' positions to a text file with the file name `Asteroid_Positions.txt`. See Section 4.8.13 for details.

4.9.2 Masks

Saving and reusing the masks would decrease the run time of the algorithm if it is run again. Saving the masks is the last thing that the algorithm does.

The algorithm deletes the directory `Masks` from the current working directory and makes a new one. Then it saves the Star and Poor Quality Section Masks for each image on the list as `Sta_mask_` and `BS_mask_`, respectively followed by the image's file name. It also saves the three High Value Masks (with a prefix of `HV1_mask_`, `HV2_mask_` and `HV3mask_`, respectively).

4.9.3 Masked Images

Saves difference images that have been masked only by the Star Mask with file names that start with `initial_` followed by the original image file name. See Section 4.7.3 for details.

If an asteroid has been detected, the algorithm saves the masked difference images that were used in the search for that asteroid. The file name of these images is `AA_Masked_` followed by the file name of the original image, where *A* is the number of asteroids found at the point of saving. See Section 4.8.14 for details.

4.9.4 Stacked Images

If an asteroid has been detected, the algorithm saves two of the asteroid's correctly stacked images. One using the original images and the other one using the difference images. The file names of the two images are `SnS_NormIm_AA.fits` and `SnS_DiffIm_AA.fits`, respectively, where *A* is the number of asteroids found at the point of saving. Both stacked images have a

circle around the asteroid. For details when these images are created see Section 4.8.4. For details how these images are created see Section 4.8.14.

4.9.5 Velocity Phase Space

Each time an asteroid is found the algorithm hangs on to the velocity phase space matrix that was used to identify the asteroid. Once the search for asteroids has been completed, all the velocity phase space matrices that were kept plus the last one produced are turned into contour plots.

The plots are saved as `Velocity_Space_speed.png` where *speed* depends on the velocity range limit.

4.10 Miscellaneous Functions

4.10.1 Blurring

Using the function `ndimage.gaussian_filter()` the algorithm blurs the input images with a Gaussian kernel. It assumes the input images are derivatives of, and in the same order as, the images in the List. *Sigma* is used as the standard deviation of the Gaussian kernel. If *Sigma* is one value, all the images are blurred by the same amount. If *Sigma* is a list that is the same size as the List, each input image is blurred by the *Sigma* value that has the same position on the *Sigma* list as the image has on the List. If *Sigma* is a list but not the same size as the List, the algorithm outputs **“WARNING: Incorrect Sigma, no blurring done”** and the images are not blurred. The function outputs the blurred images as single three dimensional matrix.

4.10.2 Getting Observation Times

For every image in the List the algorithm loads the original image¹⁰ and its header. The algorithm then tries to find a value labeled *“JD”* for Julian Date. If that does not work, then it tries to find a value labeled *“PHJDMID”*. If that does not work, then it tries to find a value labeled *“DATE-OBS”* and converts the date to the number of seconds since the start of the calendar. As soon as one works the algorithm will put the value into a list of acquisition times. If none of these work then an error message will pop up. If the times are in Julian Dates then they are converted into seconds.

¹⁰The original image is not actually needed, but it needs to be loaded to also load the header.

The function outputs *time*, whether it got the time from “*JD*”, “*PHJDMID*” or “*DATE-OBS*” labeled as *whichtime*, and a list containing the headers of all the original images on the List.

4.10.3 Removing Poor Quality Images

The input to this function is the Star Mask, Section Mask, the List of number of High Values and the index of the images with the first and second highest number of High Values.

If any of the difference images had more than 100 High Values after the star and poor quality section masks had been applied, then these images were classed as poor images and were removed from the list.

If any of the images are in the *ImRe* list, then the algorithm prints “**Removed Images: *ImRe***”. In the current working directory a new directory labeled *Poor_Images* is created and all of the original image files in *ImRe* are moved into that directory. If there are no images in the *ImRe* list then the algorithm outputs “**No images removed**”.

The function outputs *ImRe* and Poor Quality Section Mask.

4.10.4 Seeing

The Algorithm opens the text file *seeing* and reads it into python. Then it extracts the image names and the seeing value from the first and second column, respectively. If an image name matches with one found in the List, the image’s corresponding seeing is added to a newly created list called *Sigma*. All the seeing values in the list are converted from full width half maximums to standard deviations by dividing each value by 2.355. The function outputs *Sigma*.

4.10.5 Shifting and Stacking Images

The inputs to this function are the velocity of the stack, in component form, and the images, in the form of a three-dimensional matrix, too be stacked.

The algorithm calculates how much each image has to be shifted by in both x and y directions relative to the first image, Δx_i and Δy_i , respectively, by using the velocity of the shift and the time that each image was taken, described in Section 4.10.2. Then it calculates the size of the shifted and stacked matrix by finding the maximum absolute shifts, Δx_{max} and Δy_{max} , and then adding it to the size of the difference images. One at a time the

algorithm adds each of the images on the List to the shifted and stacked matrix. If the shifts in a particular direction are positive, say $\Delta x_{max} > 0$ and $\Delta y_{max} > 0$, then each element of the i^{th} image, $I_{x,y}^i$, is added to the shifted and stacked image, S , by the following equation

$$S_{x+\Delta x_i, y+\Delta y_i} = S_{x+\Delta x_i, y+\Delta y_i} + I_{x,y}^i \quad . \quad (4.25)$$

If $\Delta x_{max} > 0$ and $\Delta y_{max} \not> 0$, then $I_{x,y}^i$ is added to S by

$$S_{x+\Delta x_i, y+y_E+\Delta y_i} = S_{x+\Delta x_i, y+y_E+\Delta y_i} + I_{x,y}^i \quad . \quad (4.26)$$

If $\Delta x_{max} \not> 0$ and $\Delta y_{max} > 0$, then $I_{x,y}^i$ is added to S by

$$S_{x+x_E+\Delta x_i, y+\Delta y_i} = S_{x+x_E+\Delta x_i, y+\Delta y_i} + I_{x,y}^i \quad . \quad (4.27)$$

If $\Delta x_{max} \not> 0$ and $\Delta y_{max} \not> 0$, then $I_{x,y}^i$ is added to S by

$$S_{x+x_E+\Delta x_i, y+y_E+\Delta y_i} = S_{x+x_E+\Delta x_i, y+y_E+\Delta y_i} + I_{x,y}^i \quad . \quad (4.28)$$

After all the images are added to S , the function outputs S , Δx_i and Δy_i for all the images.

5 Results and Analysis

5.1 Testing the Algorithm

The data used to test the algorithm was collected by Alan Gilmore and Pam Kilmartin. They perform follow up measurements to possible near-Earth objects using a form of the shift and stack method, but their method requires the asteroids velocity to be known beforehand. Some of the object in the test data are near-Earth comets, but for simplicity they will all be referred to as asteroids. They acquired the data using the Apogee Alta F6 camera, see Section 3.2, mounted on the McLellan telescope, see Section 3.1.

For all the test data the changeable variables used by algorithm, see Section 4.3.1, initially had the values shown in Table 5.1. If the asteroid was not found in the initial run, then one or more of the changeable variables were altered to try to find the asteroid in following runs. The fixed variables, see Section 4.3.2, were the same for all data. Saturation Count was set to 65535 and the gain was set to 4. For each series of images the velocity range that contained the asteroids actual velocity were selected for searching, see Section 4.5.2, unless specified otherwise.

Variable	Value
Half	False
Star Mask Scaler	2
Saturated Star Mask Scaler	2
High Value Number	4
Threshold Value	4
Poor Quality Section Mask Modifier	10
First High Value Mask Limit	100
Second High Value Mask Limit	150
Velocity Vector Length	51

Table 5.1: The default changeable variable values, from Section 4.3.1, that were used when testing the algorithm on the data.

The data was broken up into the night that the images were taken. Each subsection in this section contains the results of the algorithm tested on the data from one night of observing, including the velocity of the asteroid calculated by the algorithm compared with

the actual velocity, the Signal of the asteroid, see Section 4.8.11, and any special notes on the testing. The fraction of asteroids the algorithm was able to detect that night along with any additional discoveries were also given.

The asteroids velocities were given as total magnitude and angle between the positive declination vector and the asteroids velocity vector going clockwise.

5.1.1 26 February 2014

Of the 16 asteroids that were observed on the night of 26 February 2014 the algorithm was able to detect 15 of them, see Table 5.2. Of the 15 asteroids that were detected, only two were found not using the initial changeable variables or velocity range. One false positive was detected.

Asteroid Name	Actual Velocity	Algorithm Velocity	Signal	Detected?
2014 CJ13	2.78"/min @ 240°	2.80"/min @ 240°	19.2	YES
2014 DA (1)	0.92"/min @ 106°	COULD NOT FIND		no
2014 CF14	2.55"/min @ 281°	2.92"/min @ 280°	23.8	YES
2014 CD13	26.34"/min @ 122°	26.60"/min @ 119°	74.2	YES
2014 DG	8.29"/min @ 218°	8.21"/min @ 217°	13.5	YES
2014 CG13	5.53"/min @ 201°	5.61"/min @ 202°	66.4	YES
2014 CM13	2.49"/min @ 269°	2.46"/min @ 269°	31.5	YES
2014 DP7 (2)	1.89"/min @ 342°	1.85"/min @ 343°	9.2	YES
2014 BR43	1.14"/min @ 209°	1.42"/min @ 205°	17.9	YES
2014 CU13	1.51"/min @ 89°	1.39"/min @ 89°	12.6	YES
2014 DB11	2.91"/min @ 169°	2.99"/min @ 171°	17.2	YES
2014 CG14	2.59"/min @ 239°	2.63"/min @ 240°	18.9	YES
2014 CP13	6.23"/min @ 145°	6.11"/min @ 146°	53.1	YES
2014 DC (3)	3.29"/min @ 103°	3.18"/min @ 103°	8.4	YES
2014 BF63 (4)	1.37"/min @ 144°	1.38"/min @ 145°	9.0	YES
2014 BG60	2.20"/min @ 108°	2.21"/min @ 107°	10.1	YES

Table 5.2: A table of the asteroids that were observed on 26 February 2014. For each asteroid, this table contains their actual velocity at the time the images, the asteroid's velocity and Signal (see Section 4.8.11) calculated by the algorithm, and whether the asteroid was detected.

Notes about the detection of asteroids from Table 5.2:

- (1)** A faint, trident looking poor quality section in the images were not masked by the algorithm, causing the asteroid to not be detected.
- (2)** The Poor Quality Section Mask Modifier had to be changed to 0.75 in order to detect the asteroid.

- (3) The asteroid was found when the custom Velocity Range limit was used and set to 4"/min.
- (4) A faint poor quality section in some of the images was not able to mask, and caused a false positive asteroid detection. The false positive had a signal of 10.2 and a velocity of 1.12"/min @ 11° .

5.1.2 30 May 2014

Of the 23 asteroids that were observed on the night of 30 May 2014 the algorithm was able to detect 18 of them, see Table 5.3. Of the 18 asteroids that were detected, only four were found not using the initial changeable variables. There were three possible false positives. Possibly two additional asteroids were found, i.e. asteroids that were not being searched for. One of the 18 asteroids detected by the algorithm, P10bF5f, was not detected by Alan Gilmore and Pam Kilmartin.

Asteroid Name	Actual Velocity	Calculated Velocity	Best Signal	Detected?
2014 JA31	4.66"/min @ 286°	COULD NOT FIND		no
2014 JW24	13.27"/min @ 252°	13.15"/min @ 254°	53.8	YES
VKB2FF1	8.80"/min @ 102°	8.92"/min @ 101°	44.1	YES
P10bJ7q (1)	1.54"/min @ 235°	1.52"/min @ 235°	9.0	YES
P10bF5f	1.24"/min @ 246°	1.55"/min @ 242°	10.3	YES
2014 HN178	4.08"/min @ 184°	4.09"/min @ 185°	17.8	YES
2014 JK25	0.90"/min @ 16°	COULD NOT FIND		no
2014 GC45	1.76"/min @ 169°	COULD NOT FIND		no
2014 BF63 (2)	0.68"/min @ 346°	0.70"/min @ 340°	12.1	YES
2014 KE45	2.08"/min @ 267°	2.19"/min @ 269°	15.4	YES
2014 HW177	3.97"/min @ 164°	COULD NOT FIND		no
2014 KN4	1.73"/min @ 73°	1.65"/min @ 72°	9.7	YES
2014 KQ75 (3)	29.75"/min @ 180°	30.74"/min @ 183°	89.5	YES
2014 KC46 (4)	4.02"/min @ 187°	4.12"/min @ 187°	21.8	YES
2014 KM22	6.78"/min @ 75°	6.75"/min @ 73°	6.5	no
2014 HD3 (5)	8.00"/min @ 139°	7.94"/min @ 138°	18.2	YES
2014 KA46	5.35"/min @ 120°	5.11"/min @ 120°	14.9	YES
2014 JV55 (6)	3.37"/min @ 238°	3.34"/min @ 237°	14.0	YES
2014 JH57 (7)	1.31"/min @ 107°	1.56"/min @ 113°	11.5	YES
P10bJan	6.05"/min @ 170°	6.16"/min @ 171°	37.6	YES
2014 JN57	4.39"/min @ 140°	4.59"/min @ 142°	10.6	YES
2014 HQ124	1.90"/min @ 80°	2.12"/min @ 67°	10.2	YES
2014 GH35 (8)	5.77"/min @ 36°	5.85"/min @ 25°	55.0	YES

Table 5.3: A table of the asteroids that were observed on 30 May 2014. For each asteroid, this table contains their actual velocity at the time the images, the asteroid's velocity and Signal (see Section 4.8.11) calculated by the algorithm, and whether the asteroid was detected. Asteroids highlighted in grey were not found by Alan Gilmore and Pam Kilmartin.

Of the five asteroids that were not detected by the algorithm, one was not detected by Alan Gilmore and Pam Kilmartin either, and one was found by the algorithm but did not have a high enough signal to be classed as a detection. Ignoring the asteroid that Alan Gilmore and Pam Kilmartin did not detect, 19/21 asteroids were found and 18/21 asteroids were detected.

Notes about the detection of asteroids from Table 5.3:

- (1) On the initial run the algorithm found an asteroid with a velocity of $2.00''/\text{min}$ @ 274° and a signal of 8.8. On the stacked image the asteroid does not have a circular shape, which suggests this is a false positive. The asteroid was found when the First High Value Mask Limit was set to 0, Half set to *True* (starting with the second image), and the Saturated Star Mask Scaler was set to 3.
- (2) The asteroid was found when Half set to *True*.
- (3) The asteroid was found when the Star Mask Scaler and the Saturated Star Mask Scaler were set to 1.
- (4) Another asteroid was found with a velocity of $0.85''/\text{min}$ @ 277° and a signal of 17.0.
- (5) When the asteroid was found it was not properly masked, which led to the asteroid being detected again (increasing the asteroid mask size to 6 fixed this).
- (6) Another possible asteroid was found with a velocity of $1.89''/\text{min}$ @ 89° and a signal of 8.9. On the stacked difference image the asteroid is right on the edge of the image, which suggests this is a false positive.
- (7) The asteroid was found when the First High Value Mask Limit was set to 5. Another two asteroids were found. One that could be confirmed (it could be seen in the images), which had a velocity of $2.09''/\text{min}$ @ 98° and a signal of 14.2. The other one, which had a velocity of $2.02''/\text{min}$ @ 224° and a signal of 8.4, could not be confirmed.
- (8) Another possible asteroid was found with a velocity of $4.98''/\text{min}$ @ 50° and a signal of 8.8. On the stacked image the asteroid does not have a circular shape, which suggests this is a false positive.

5.1.3 31 May 2014

Of the 20 asteroids that were observed on the night of 31 May 2014 the algorithm was able to detect 14 of them, see Table 5.4. Of the 14 asteroids that were detected, only five were found not using the initial changeable variables. There was one possible false positive and one additional asteroid.

Of the six asteroids that were not detected by the algorithm one was not detected by Alan Gilmore and Pam Kilmartin either, and another one had moving stars in their images, which was subsequently not used. Two of those six that were not detected were found by the algorithm but did not have a high enough signal to be classed as a detection. Ignoring the asteroid that Alan Gilmore and Pam Kilmartin did not detect and the one that wasn't used, 16/18 asteroids were found and 14/18 asteroids were detected.

Asteroid Name	Actual Velocity	Calculated Velocity	Best Signal	Detected?
2014 JA31	4.18"/min @ 293°	COULD NOT FIND		no
2014 HJ184	1.17"/min @ 271°	COULD NOT FIND		no
2014 JW24	11.84"/min @ 269°	11.68"/min @ 268°	84.6	YES
2014 JQ24 (1)	1.66"/min @ 202°	COULD NOT FIND		no
2014 KQ75 (2)	90.72"/min @ 180°	90.97"/min @ 181°	26.4	YES
2014 KO76	6.35"/min @ 131°	6.49"/min @ 134°	9.8	YES
VKB2FF1	9.57"/min @ 100°	9.93"/min @ 100°	55.6	YES
2014 KE45	2.13"/min @ 277°	1.90"/min @ 269°	9.1	YES
2014 KB39 (3)		DID NOT USE		
2014 JY24	1.32"/min @ 259°	1.61"/min @ 258°	9.8	YES
2014 KN4	1.81"/min @ 73°	1.81"/min @ 73°	4.6	no
2014 KC46 (4)	4.94"/min @ 187°	4.97"/min @ 187°	11.2	YES
2014 KW76	1.44"/min @ 160°	1.40"/min @ 159°	8.7	YES
2014 KT76	1.30"/min @ 274°	1.33"/min @ 269°	5.6	no
2014 KQ76	2.30"/min @ 268°	2.32"/min @ 269°	9.7	YES
2014 KX76	5.41"/min @ 170°	5.07"/min @ 171°	42.5	YES
P10bJau (5)	1.77"/min @ 308°	1.67"/min @ 299°	9.6	YES
P10bJaw (6)	1.61"/min @ 77°	1.65"/min @ 79°	8.9	YES
P10bJax	4.14"/min @ 137°	4.14"/min @ 137°	32.1	YES
2014 JN57 (7)	4.29"/min @ 140°	4.40"/min @ 143°	10.9	YES

Table 5.4: A table of the asteroids that were observed on 31 May 2014. For each asteroid, this table contains their actual velocity at the time the images, the asteroid's velocity and Signal (see Section 4.8.11) calculated by the algorithm, and whether the asteroid was detected. Asteroids highlighted in grey were not found by Alan Gilmore and Pam Kilmartin.

Notes about the detection of asteroids from Table 5.4:

- (1) A bit less than half way through the number of images the seeing deteriorated and the stars were not able to be masked properly.
- (2) The asteroid was found when Star Mask Scaler and Saturated Star Mask were both set to 1 and the cut off mark for labelling an image as a poor image was raised to 160 High Values, see Section 4.10.3.
- (3) The stars were moving too much in the images, so these images were not tested on the algorithm.
- (4) The asteroid was found when the image rejection was raised to 150 High Values, see

Section 4.10.3. Another asteroid was found with a velocity of $10.2''/\text{min}$ @ 218° and a signal of 8.4. The non-circular shape of the possible asteroid in the stacked image suggests that this is a false positive.

- (5) The asteroid was found when the First High Value Mask Limit was set to 10 and the Second was set to 15. Another asteroid was found with a velocity of $1.27''/\text{min}$ @ 183° and a signal of 12.2.
- (6) The asteroid was found when the First High Value Mask Limit was set to 150.
- (7) The asteroid was found when the First High Value Mask Limit was set to 10.

5.1.4 1 June 2014

Of the 14 asteroids that were observed on the night of 1 June 2014 the algorithm was able to detect six of them, see Table 5.5. Of the six asteroids that were detected, three were found not using the initial changeable variables. There were two false positive and one additional asteroid was found.

Asteroid Name	Actual Velocity	Calculated Velocity	Best Signal	Detected?
2014 JA31	$3.73''/\text{min}$ @ 299°	COULD NOT FIND		no
c2014 J1	$3.17''/\text{min}$ @ 281°	$2.86''/\text{min}$ @ 282°	17.9	YES
2014 JQ24	$1.57''/\text{min}$ @ 201°	COULD NOT FIND		no
2014 KO76	$5.98''/\text{min}$ @ 130°	COULD NOT FIND		no
2014 KQ84	$10.46''/\text{min}$ @ 99°	$10.87''/\text{min}$ @ 99°	57.6	YES
2014 HW177	$3.84''/\text{min}$ @ 162°	COULD NOT FIND		no
2014 KB39 (1)	$3.54''/\text{min}$ @ 84°	COULD NOT FIND		no
2014 KW76 (2)	$1.09''/\text{min}$ @ 165°	$1.16''/\text{min}$ @ 152°	6.0	no
2014 KT76 (3)	$1.25''/\text{min}$ @ 274°	$1.27''/\text{min}$ @ 274°	8.5	YES
2014 KQ76	$2.30''/\text{min}$ @ 268°	$2.36''/\text{min}$ @ 268°	16.4	YES
2014 KR84 (4)	$1.81''/\text{min}$ @ 307°	$1.93''/\text{min}$ @ 304°	12.5	YES
2014 KV84 (5)	$1.62''/\text{min}$ @ 77°	$1.46''/\text{min}$ @ 82°	8.9	YES
2014 JH78	$1.02''/\text{min}$ @ 135°	COULD NOT FIND		no
2014 HQ124 (6)	$3.24''/\text{min}$ @ 73°	COULD NOT FIND		no

Table 5.5: A table of the asteroids that were observed on 1 June 2014. For each asteroid, this table contains their actual velocity at the time the images, the asteroid's velocity and Signal (see Section 4.8.11) calculated by the algorithm, and whether the asteroid was detected. Asteroids highlighted in grey were not found by Alan Gilmore and Pam Kilmartin.

Of the eight asteroids that were not detected by the algorithm four were not detected by Alan Gilmore and Pam Kilmartin either, and another one was so bright it was classed as a poor quality section. Another one of the eight that was found by the algorithm but did not have a high enough signal to be classed as a detection. Ignoring the asteroids that Alan Gilmore and Pam Kilmartin did not detect and the one that was classed as a poor

quality section, since detecting asteroids of that brightness is not the goal of the algorithm, 7/9 asteroids were found and 6/9 asteroids were detected.

Notes about the detection of asteroids from Table 5.5:

- (1) A dark line going through the images resulted in a false positive with a velocity of $1.72''/\text{min}$ @ 350° and a signal of 8.2.
- (2) Another asteroid was found with a velocity of $0.71''/\text{min}$ @ 263° and a signal of 18.7.
- (3) The asteroid was found when the First High Value Mask Limit was set to 200.
- (4) The asteroid was found when the First High Value Mask Limit was set to 10.
- (5) The asteroid was found when the First High Value Mask Limit was set to 10 and the Second High Value Mask Limit was set to 15.
- (6) Could not detect the asteroid since it was bright enough to be classed as a poor quality section.

5.1.5 2 June 2014

Of the 15 asteroids that were observed on the night of 2 June 2014 the algorithm was able to detect six of them, see Table 5.6. Of the six asteroids that were detected, only one was found not using the initial changeable variables. There was one possible false positive.

Of the nine asteroids that were not detected by the algorithm two were not detected by Alan Gilmore and Pam Kilmartin either, and two had moving stars in their images, which were subsequently not used. Another two were in star fields that were too crowded for the star position software to work properly, therefore they were not used. Ignoring the asteroids that Alan Gilmore and Pam Kilmartin did not detect and the ones that were not used, 6/9 asteroids were detected.

Asteroid Name	Actual Velocity	Calculated Velocity	Best Signal	Detected?
2014 KE46 (1)		DID NOT USE		
c2014 H1	1.43"/min @ 246°	1.50"/min @ 237°	10.4	YES
c2014 G7 (2)	0.54"/min @ 317°	COULD NOT FIND		no
2014 HP178	1.33"/min @ 144°	COULD NOT FIND		no
2014 KG39	9.36"/min @ 301°	COULD NOT FIND		no
2014 KD45	2.04"/min @ 87°	COULD NOT FIND		no
2014 KQ84 (3)	11.16"/min @ 95°	10.75"/min @ 96°	44.4	YES
P10bF5f	1.24"/min @ 248°	COULD NOT FIND		no
2014 JU24 (4)		DID NOT USE		
2009 NJ	1.23"/min @ 286°	1.34"/min @ 287°	21.5	YES
2014 KT76	1.20"/min @ 274°	1.27"/min @ 269°	10.6	YES
2014 KQ76	2.29"/min @ 268°	2.20"/min @ 269°	12.5	YES
2011 WW4 (5)		DID NOT USE		
2011 BH10 (6)		DID NOT USE		
2014 QH124 (7)	4.29"/min @ 67°	4.50"/min @ 69°	29.8	YES

Table 5.6: A table of the asteroids that were observed on 2 June 2014. For each asteroid, this table contains their actual velocity at the time the images, the asteroid's velocity and Signal (see Section 4.8.11) calculated by the algorithm, and whether the asteroid was detected. Asteroids highlighted in grey were not found by Alan Gilmore and Pam Kilmartin.

Notes about the detection of asteroids from Table 5.6:

- (1) The algorithm was not tested on this asteroid since stars were moving in some of the images.
- (2) There were two poor quality sections in the difference images. In the difference image that the algorithm tried to locate the poor quality sections, the second poor quality section was not as prominent. This resulted in the second poor quality section not being masked, which in turn caused the asteroid to not be detected.
- (3) The asteroid was found when Star Mask Scaler and Saturated Star Mask were both set to 1.25.
- (4) The algorithm was not tested on this asteroid since stars were moving in some of the images.
- (5) The star finding software did not work properly due to the star field was being too crowded. Therefore this asteroid was not used.
- (6) The star finding software did not work properly due to the star field was being too crowded. Therefore this asteroid was not used.
- (7) When the asteroid was found it was not properly masked, which led to the asteroid being detected again (increasing the asteroid mask size to 6 fixed this). Another asteroid was found with a velocity of 3.81"/min @ 164° and a signal of 11.4. On the stacked image the possible asteroid does not have a circular shape, which suggests this is a false positive.

5.1.6 3 June 2014

Of the six asteroids that were observed on the night of 3 June 2014 the algorithm was able to detect 3 of them, see Table 5.7. Of the three asteroids that were detected, only one was found not using the initial changeable variables. There were no false positives or additional asteroids found.

Of the three asteroids that were not detected by the algorithm one was not detected by Alan Gilmore and Pam Kilmartin either. Ignoring this asteroid, 3/5 were detected.

Asteroid Name	Actual	Calculated Velocity	Best Signal	Detected?
2014 JA31	3.03"/min @ 307°	COULD NOT FIND		no
2014 HN178 (1)	5.74"/min @ 180°	COULD NOT FIND		no
2014 KE46 (2)	0.94"/min @ 354°	0.90"/min @ 348°	15.2	YES
2014 KO76	5.00"/min @ 128°	5.02"/min @ 129°	10.5	YES
2014 KE45	2.26"/min @ 270°	2.21"/min @ 269°	18.2	YES
P10bssl	1.03"/min @ 51°	COULD NOT FIND		no

Table 5.7: A table of the asteroids that were observed on 3 June 2014. For each asteroid, this table contains their actual velocity at the time the images, the asteroid's velocity and Signal (see Section 4.8.11) calculated by the algorithm, and whether the asteroid was detected. Asteroids highlighted in grey were not found by Alan Gilmore and Pam Kilmartin.

Notes about the detection of asteroids from Table 5.7:

- (1) There were two poor quality sections in the difference images. In the difference image that the algorithm tried to locate the poor quality sections, the second poor quality section was not as prominent. This resulted in the second poor quality section not being masked, which in turn caused the asteroid to not be detected.
- (2) The asteroid was found when the First High Value Mask Limit was set to 500 and Half was set to *True* (starting with the second image in the List).

5.1.7 21 June 2014

Of the 19 asteroids that were observed on the night of 21 June 2014 the algorithm was able to detect 10 of them, see Table 5.8. Of the 10 asteroids that were detected, only four were found not using the initial changeable variables. There were two possible false positives.

Of the nine asteroids that were not detected by the algorithm two were not detected by Alan Gilmore and Pam Kilmartin either, two were moving too slow to be detected, and one was found by the algorithm but did not have a high enough signal to be classed as a detection. Ignoring the asteroids that Alan Gilmore and Pam Kilmartin did not detect and

the ones that were too slow to be detected, 11/15 asteroids were found and 10/15 asteroids were detected.

Asteroid Name	Actual Velocity	Calculated Velocity	Best Signal	Detected?
2014 LN14 (1)	0.19"/min @ 332°	COULD NOT FIND		no
2014 LO14 (2)	0.069"/min @ 1°	COULD NOT FIND		no
2014 BF63 (3)	DID NOT USE			
2014 MP	17.93"/min @ 193°	18.31"/min @ 192°	28.0	YES
2014 LR26	0.85"/min @ 262°	COULD NOT FIND		no
P10cbZu	6.84"/min @ 203°	6.59"/min @ 204°	14.2	YES
VMBA720 (4)	14.38"/min @ 195°	14.69"/min @ 195°	33.8	YES
VMBA819 (5)	2.77"/min @ 344°	2.69"/min @ 343°	10.9	YES
P10ccZQ	2.12"/min @ 251°	COULD NOT FIND		no
2014 LQ26 (6)	0.72"/min @ 272°	0.72"/min @ 291°	8.6	YES
2014 HW177	1.72"/min @ 121°	COULD NOT FIND		no
P10ccrp (7)	0.69"/min @ 269°	0.70"/min @ 269°	10.5	YES
VMBAB72	8.15"/min @ 128°	7.87"/min @ 129°	13.9	YES
2014 LU14	1.42"/min @ 298°	1.41"/min @ 300°	4.6	no
P2014 L3	0.51"/min @ 103°	COULD NOT FIND		no
2014 LM9	4.73"/min @ 81°	4.75"/min @ 80°	16.1	YES
2014 LK26	1.40"/min @ 42°	COULD NOT FIND		no
2014 KA46	6.43"/min @ 101°	6.65"/min @ 101°	22.8	YES
2014 LQ25 (8)	1.84"/min @ 296°	1.92"/min @ 296°	8.2	YES

Table 5.8: A table of the asteroids that were observed on 21 June 2014. For each asteroid, this table contains their actual velocity at the time the images, the asteroid's velocity and Signal (see Section 4.8.11) calculated by the algorithm, and whether the asteroid was detected. Asteroids highlighted in grey were not found by Alan Gilmore and Pam Kilmartin.

Notes about the detection of asteroids from Table 5.8:

- (1)** The asteroid was moving too slow to be able to detect.
- (2)** The asteroid was moving too slow to be able to detect.
- (3)** The star finding software did not work properly due to the star field was being too crowded. Therefore this asteroid was not used.
- (4)** The asteroid was found when the Star Mask Scaler and the Saturated Star Mask Scaler were both set to 1.5.
- (5)** A false positive was also found which has a velocity of 2.32"/min @ 300° and a signal of 11.1.
- (6)** The asteroid was found when Half was set to *True* and the Star Mask Scaler was set to 1.5.
- (7)** The asteroid was found when Half was set to *True*. A false positive was also found, which has a velocity of 1.49"/min @ 224° and a signal of 8.4.

(8) The asteroid was found when the Second High Value Mask Limit was set to 250.

5.1.8 22 June 2014

Of the 27 asteroids that were observed on the night of 22 June 2014 the algorithm was able to detect 13 of them, see Table 5.9. Of the 13 asteroids that were detected, only two were found not using the initial changeable variables. There was one possible false positive and one additional asteroid found.

Asteroid Name	Actual Velocity	Calculated Velocity	Best Signal	Detected?
2014 KM76	2.08"/min @ 182°	COULD NOT FIND		no
2014 KL22	6.47"/min @ 244°	6.59"/min @ 242°	10.2	YES
2014 HH177	0.72"/min @ 128°	COULD NOT FIND		no
2007 HR	4.47"/min @ 294°	4.35"/min @ 293°	22.8	YES
2014 KE45	2.66"/min @ 286°	2.81"/min @ 284°	8.5	YES
2014 JK25	0.62"/min @ 55°	COULD NOT FIND		no
2014 HR4 (1)	0.80"/min @ 211°	COULD NOT FIND		no
P10bFbf	0.83"/min @ 251°	COULD NOT FIND		no
2014 MP	18.11"/min @ 193°	18.32"/min @ 192°	38.2	YES
2014 JC31 (2)		DID NOT USE		
2014 LX14	1.34"/min @ 354°	1.27"/min @ 354°	15.7	YES
2014 LX21	3.30"/min @ 28°	3.27"/min @ 29°	37.3	YES
2014 HW177	1.66"/min @ 119°	COULD NOT FIND		no
P10cedm (2)		DID NOT USE		
VMBAEEF	0.72"/min @ 201°	0.77"/min @ 195°	5.7	no
VMBAEC6	0.92"/min @ 304°	COULD NOT FIND		no
VMBAE59 (3)	0.68"/min @ 270°	0.92"/min @ 269°	15.5	YES
VMBAE35	2.09"/min @ 328°	COULD NOT FIND		no
VMBA720	14.48"/min @ 195°	14.47"/min @ 195°	83.3	YES
P10ccQZ	2.39"/min @ 252°	2.41"/min @ 256°	27.2	YES
P10ccrp	0.68"/min @ 269°	COULD NOT FIND		no
VMBAB72	5.76"/min @ 129°	COULD NOT FIND		no
2014 KX76 (4)	0.68"/min @ 211°	COULD NOT FIND		no
2012 XD17	2.09"/min @ 119°	2.23"/min @ 123°	13.1	YES
2014 KG2 (5)	1.15"/min @ 30°	1.10"/min @ 49°	14.5	YES
2012 TU	1.39"/min @ 317°	1.59"/min @ 321°	12.7	YES
2011 WL15	1.79"/min @ 102°	1.85"/min @ 100°	14.6	YES

Table 5.9: A table of the asteroids that were observed on 22 June 2014. For each asteroid, this table contains their actual velocity at the time the images, the asteroid's velocity and Signal (see Section 4.8.11) calculated by the algorithm, and whether the asteroid was detected. Asteroids highlighted in grey were not found by Alan Gilmore and Pam Kilmartin.

Of the 14 asteroids that were not detected by the algorithm three were not detected by Alan Gilmore and Pam Kilmartin either, and one was in a star field that was too crowded for the star position software to work. Another one of the 14 was found by the algorithm

but did not have a high enough signal to be classed as a detection. Ignoring the asteroids that Alan Gilmore and Pam Kilmartin did not detect and the one that was not used, 14/23 asteroids were found and 13/23 asteroids were detected.

Notes about the detection of asteroids from Table 5.9:

- (1) There was a false positive, with a signal of 10.8, and an additional asteroid, with a signal of 8.4, found. Both had a velocity of $0.92''/\text{min}$ @ 89° .
- (2) There were not enough images (only 2).
- (3) The star finding software did not work properly due to the star field was being too crowded. Therefore this asteroid was not used.
- (4) The asteroid was found when Half was set to *True*.
- (5) The images contained a poor quality section that could not be masked properly.
- (6) The asteroid was found when Half was set to *True* and the First High Value Mask Limit was set to two and the Second High Value Mask Limit was set to three.

5.1.9 26 June 2014

Of the nine asteroids that were observed on the night of 26 June 2014 the algorithm was able to detect six of them, see Table 5.10. All of the six asteroids that were detected were found using the initial changeable variables. There was one possible false positive.

Of the three asteroids that were not detected by the algorithm two were not detected by Alan Gilmore and Pam Kilmartin either. Ignoring the asteroids that Alan Gilmore and Pam Kilmartin did not detect, 6/7 asteroids were detected.

Asteroid Name	Actual Velocity	Calculated Velocity	Best Signal	Detected?
2014 MP (1)	$13.83''/\text{min}$ @ 194°	$14.03''/\text{min}$ @ 196°	8.1	YES
2014 LX9 (2)	$2.11''/\text{min}$ @ 161°	COULD NOT FIND		no
P10cg30	$3.31''/\text{min}$ @ 119°	$3.23''/\text{min}$ @ 119°	41.9	YES
P10ch1X (3)	$5.08''/\text{min}$ @ 278°	$5.07''/\text{min}$ @ 277°	63.1	YES
2014 MG6	$17.11''/\text{min}$ @ 36°	$16.86''/\text{min}$ @ 36°	91.9	YES
2014 MH6	$2.20''/\text{min}$ @ 244°	$2.20''/\text{min}$ @ 242°	17.4	YES
2014 KN4	$3.78''/\text{min}$ @ 74°	$3.80''/\text{min}$ @ 75°	22.0	YES
2014 MB6	$2.15''/\text{min}$ @ 135°	COULD NOT FIND		no
NM06qr9	$3.64''/\text{min}$ @ 258°	COULD NOT FIND		no

Table 5.10: A table of the asteroids that were observed on 26 June 2014. For each asteroid, this table contains their actual velocity at the time the images, the asteroid's velocity and Signal (see Section 4.8.11) calculated by the algorithm, and whether the asteroid was detected. Asteroids highlighted in grey were not found by Alan Gilmore and Pam Kilmartin.

Notes about the detection of asteroids from Table 5.10:

- (1) Astrometry was not successful. Using the rough scale of 1.28 pixels per arcsecond calculated from other asteroids, an input velocity range limit of 0.2 pixels per second was used. The algorithm velocity was converted to arcsecond per minute using the same scale and the assumption that north in the images was in the negative y direction.

- (2) Astrometry was not successful. Using the rough scale of 1.28 pixels per arcsecond calculated from other asteroids, an input velocity range limit of 0.03 pixels per second was used.

- (3) A false positive was found with a velocity of $2.94''/\text{min}$ @ 176° and a signal of 8.8. This false positive was due to a star being improperly mask.

5.1.10 28 June 2014

Of the 21 asteroids that were observed on the night of 28 June 2014 the algorithm was able to detect 12 of them, see Table 5.11. Of the 12 asteroids that were detected, only four were found not using the initial changeable variables or velocity range. There were eight possible false positives and one additional asteroid found.

Of the nine asteroids that were not detected by the algorithm two were not detected by Alan Gilmore and Pam Kilmartin either, and another two were moving too slow to be detected. Ignoring the asteroids that Alan Gilmore and Pam Kilmartin did not detect and the two that were too slow to be detected, 14/17 asteroids were found and 12/17 asteroids were detected.

Asteroid Name	Actual Velocity	Calculated Velocity	Best Signal	Detected?
2014 MV2	1.79"/min @ 142°	1.78"/min @ 145°	7.5	no
2014 JO25	0.85"/min @ 114°	COULD NOT FIND		no
2014 MP (1)	11.67"/min @ 195°	COULD NOT FIND		no
2014 MA18 (2)	2.28"/min @ 126°	COULD NOT FIND		no
2014 MF18 (3)	8.62"/min @ 238°	8.59"/min @ 239°	27.8	YES
2014 KU76	1.21"/min @ 239°	1.31"/min @ 229°	5.5	no
P10cihc (4)	0.91"/min @ 3°	1.06"/min @ 6°	9.2	YES
2014 KP4 (5)	0.25"/min @ 70°	COULD NOT FIND		no
2014 JY24(6)	0.25"/min @ 70°	COULD NOT FIND		no
2014 MT5 (7)	1.58"/min @ 230°	1.58"/min @ 229°	13.1	YES
2014 MQ5	2.75"/min @ 9°	2.80"/min @ 10°	10.7	YES
2014 KQ76	1.19"/min @ 256°	1.25"/min @ 232°	12.5	YES
VMBB91D (8)	3.50"/min @ 186°	3.53"/min @ 186°	10.5	YES
2014 MF6 (9)	3.75"/min @ 206°	3.54"/min @ 210°	8.9	YES
P10cgT2	7.22"/min @ 210°	7.06"/min @ 211°	10.2	YES
P10cgT0	1.27"/min @ 247°	1.30"/min @ 249°	13.1	YES
P10cigz (10)	1.23"/min @ 223°	1.46"/min @ 231°	8.9	YES
VMBCEC9	1.16"/min @ 229°	1.46"/min @ 231°	8.9	YES
P10ahHC (11)	1.05"/min @ 298°	0.99"/min @ 297°	9.4	YES
P10cmdF	8.05"/min @ 193°	COULD NOT FIND		no
NMO6qr9	3.39"/min @ 262°	COULD NOT FIND		no

Table 5.11: A table of the asteroids that were observed on 28 June 2014. For each asteroid, this table contains their actual velocity at the time the images, the asteroid’s velocity and Signal (see Section 4.8.11) calculated by the algorithm, and whether the asteroid was detected. Asteroids highlighted in grey were not found by Alan Gilmore and Pam Kilmartin.

- (1) In some of the images the residuals of the faint stars were unusually large, causing the asteroid to not be detected.
- (2) Astrometry was not successful. Using the rough scale of 1.28 pixels per arcsecond calculated from other asteroids, an input velocity range limit of 0.04 pixels per second was used. There were two poor quality sections in the difference images. In the difference image that the algorithm tried to locate the poor quality sections, the second poor quality section was not as prominent. This resulted in the second poor quality section not being masked, which in turn caused the asteroid to not be detected. Four false positives due to the non-masked poor quality section were detected.
- (3) A false positive was found with a velocity of 3.82"/min @ 191° and a signal of 10.2. This false positive is due to vertical black lines in the images. Another possible asteroid was found with a velocity of 3.70"/min @ 209° and a signal of 8.5 but an irregular shape in stacked difference image suggests that it is a false positive.
- (4) The asteroid was found when Half was set to *True*. Another possible asteroid was found with a velocity of 0.85"/min @ 62° and a signal of 8.8 but an irregular shape in stacked difference image suggests that it is a false positive.

- (5) The asteroid was moving too slow to be able to detect.
- (6) The asteroid was moving too slow to be able to detect.
- (7) The asteroid was found when the Second High Value Mask Limit was set to 250.
- (8) Astrometry was not successful. Using the rough scale of 1.28 pixels per arcsecond calculated from other asteroids, an input velocity range limit of 0.05 pixels per second was used. The algorithm velocity was converted to arcsecond per minute using the same scale and the assumption that north on the images was in the negative y direction.
- (9) When the asteroid was found it was not properly masked, which led to the asteroid being detected again (increasing the asteroid mask size to 6 fixed this).
- (10) Another asteroid was found with a velocity of $1.53''/\text{min}$ @ 64° and a signal of 8.1. Another possible asteroid was found with a velocity of $1.27''/\text{min}$ @ 359° and a signal of 8.1. On the stacked difference image the asteroid is right on the edge of the image, which suggests this is a false positive.
- (11) The asteroid was found when Half was set to *True*. Another asteroid was found with a velocity of $0.71''/\text{min}$ @ 242° and a signal of 9.7, along with another possible asteroid with a velocity of $0.72''/\text{min}$ @ 353° and a signal of 9.1.

5.2 Statistics of the Testing

Looking at all the test data that was not ignored, there were a total of 142 test asteroids. Of the 142 asteroids, 112 were found and 104 were detected. Percentage wise the amount found was 80% and detected was 73%. 77 of the 104 asteroids detected were found using the initial changeable variable values.

When looking at the magnitudes¹ and the velocities of the test asteroids, as can be seen in Figures 5.1 & 5.2, a majority of the asteroids that were not detected by the algorithm were faint, slow moving asteroids. The algorithm found 60% (47/78) of the asteroids that had velocities slower than $2.4''/\text{min}$, compared to 89% (57/64) of asteroids that had velocities of $2.4''/\text{min}$ or faster. The algorithm also found 55% (33/58) of the asteroids that had a visual magnitude fainter than 20.5, compared to 86% (72/84) of asteroids of 20.5th magnitude or brighter.

A possible reason for slower asteroid's being harder to detect by the algorithm could be due to the proximity of their correctly stacked image to the origin in the velocity phase space. Very faint stars produce a spike at the origin in the velocity phase space, which diminishes a slow moving asteroids spike. If the images were taken at longer time intervals, then the

¹One should note that fainter objects have higher magnitudes.

influence of the spike at the origin on the chances of a slow moving asteroid being detected would decrease. This, in turn, would increase the chance of the asteroid being detected.

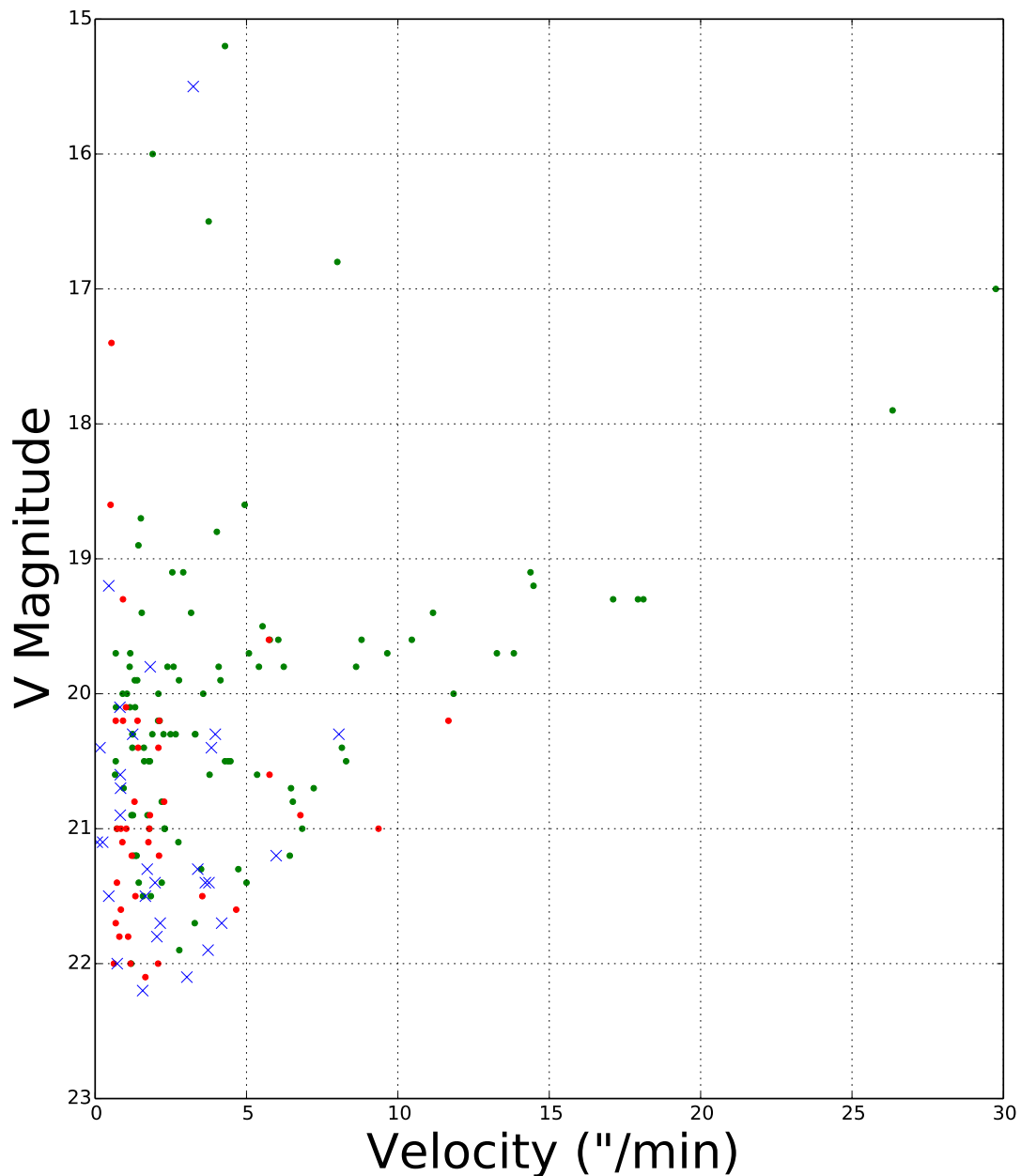


Figure 5.1: A plot of the magnitude versus velocity of the test asteroids. The asteroids detected by the algorithm are represented by green dots, where as the asteroids that were not detected are represented by red dots. Blue crosses are used for the asteroids that were ignored. This plot does not include 2014 KQ75, which has a velocity of $90.75''/\text{min}$.

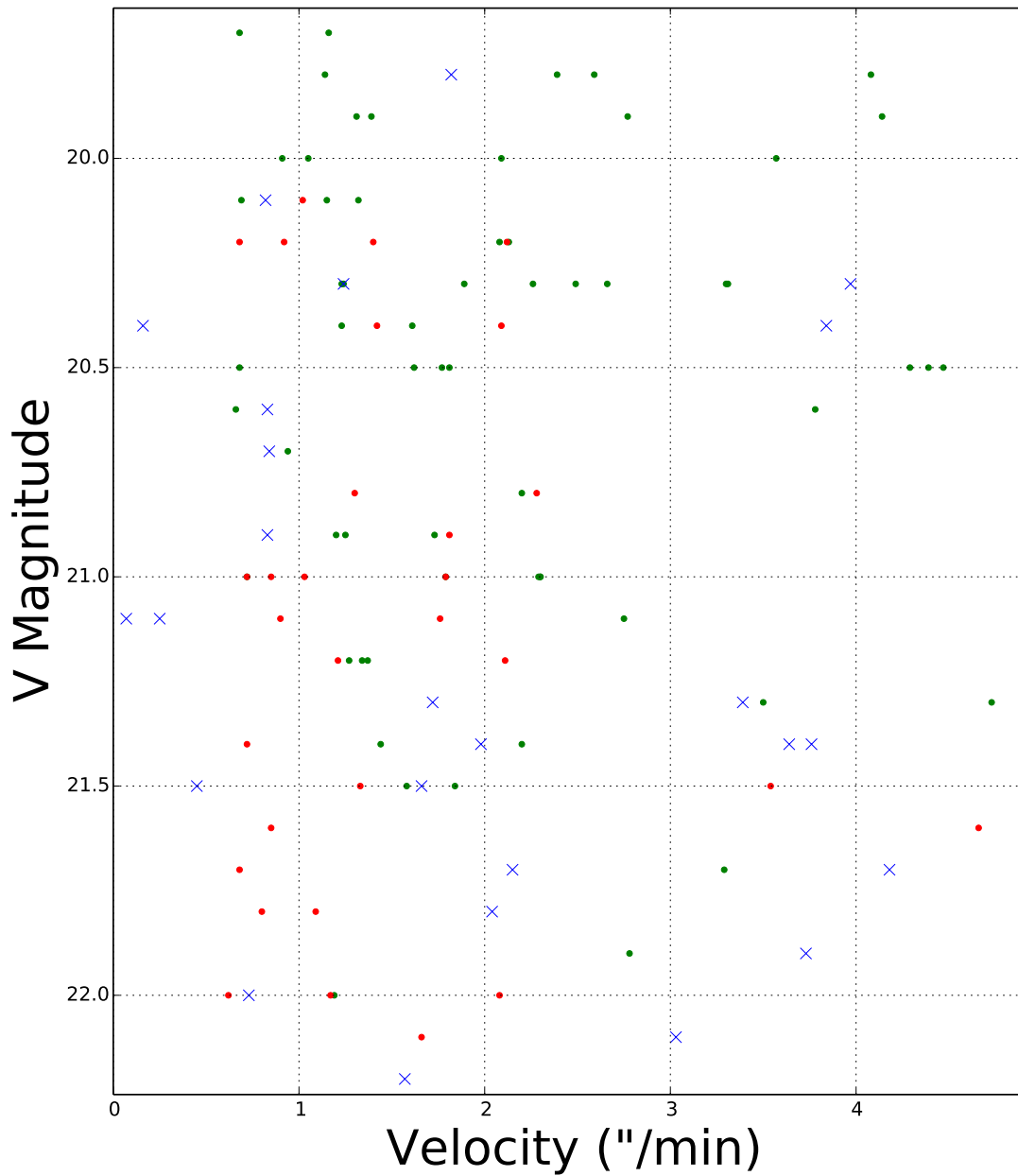


Figure 5.2: A plot of the visual (V) magnitude versus velocity of the slow moving, faint test asteroids. The asteroids detected by the algorithm are represented by green dots, whereas the asteroids that were not detected are represented by red dots. Blue crosses are used for the asteroids that were ignored.

A possible reason for faint asteroids being harder to detect by the algorithm might be due to the fact that more images are needed to find a faint asteroid. The more images taken of a particular asteroid, the more likely that the asteroid would be masked by a star mask in one or more of the difference images, due to the size of the star residuals being over estimated

a majority of the time. If the asteroid is masked in any of the difference images, then the asteroid's spike in the velocity phase space would be decreased, which would decrease the asteroid's chance of being detected. A more efficient star mask could lead to an increase of asteroid detection likelihood.

5.2.1 False Positives

There were a total of 21 false positives found in the test. This means that there is a 16% chance that a false positive would occur in a series of images, or roughly one in six. This value is somewhat inflated by the series of images that contained the asteroid 2014 MA18, which contained four false positives.

A significant majority of the false positives found were caused by improper masking or occurred right on the edge of the stacked difference image. If the masking was improved and detections that were right on the edge of the difference image were disallowed, then the number of false positives would greatly decrease.

5.2.2 Additional Asteroids

In addition to the 104 test asteroids detected by the algorithm, at least another seven asteroids that the algorithm was not being tested on were also found.

There were two asteroids Alan Gilmore and Pam Kilmartin were not able to find that were detected by the algorithm. This suggests there might be an advantage to using the algorithm over the method used by Alan Gilmore and Pam Kilmartin.

5.2.3 Unused Data

A total of 29 asteroids were ignored while looking at the statistics. Seventeen of these asteroids were ignored since they could not be found either by the algorithm or Alan Gilmore and Pam Kilmartin. The images of three of the asteroids that were ignored had some images where the stars were moving in the images, which was caused by the telescope not tracking the stars properly.

Three asteroids had star fields containing too many stars for the star position software to calculate positions accurately, resulting in the star residuals not being masked properly. This led to these three asteroids not being used. This was unfortunate since finding as-

teroids in crowded fields was one of the motivations for this thesis. It is likely that these asteroids would not have been found even if the stars' positions were correctly identified, due to the size of the star residuals being overestimated a majority of the time.

If the asteroid's velocity is slow enough, even after halving the number of images, that the spike due to the asteroid in the velocity phase space was removed in the origin removal process, then it cannot be detected. This is the case for four of the asteroids that were tested. If images taken further apart were used, then there would be a chance of detecting the asteroids.

Of the two remaining asteroids that were not used, one only contained two images and the other one was bright enough to be classed as a poor quality section.

6 Summary

Asteroids, near-Earth asteroids in particular, have importance due to the threat of collisions with Earth, the information they contain about the early Solar System, and the possibility of being mined for resources. As a result there are a number of astronomical surveys with the main goal of finding these asteroids of interest.

A popular technique for finding moving celestial objects, but not so much for asteroids, is the shift and stack method. This method increases the signal-to-noise of asteroids by shifting a series of images relative to one another so that an asteroid is aligned in all the images, then the images are combined into one.

This thesis attempted to improve the shift and stack method by including difference image analysis, since difference images remove stationary objects of constant brightness from images. An algorithm was created that masks any imperfections from difference images, like star residuals, and performs multiple shifts and stacks of these images with different shift velocities. Ignoring the stacked images that have a low shift velocity, the algorithm then compares the lowest minimum value of all the stacked images with the minimum value of stacked images with a similar shift velocity. If the most negative minimum value was significantly smaller than these other minimum values, then the algorithm would signal an asteroid was detected.

The algorithm was tested using 171 series of images containing asteroids taken by Alan Gilmore and Pam Kilmartin. A total of 29 of these asteroids were ignored mainly due to Alan Gilmore and Pam Kilmartin not finding them either, the star fields being too crowded, the asteroid moving too slow, or the stars moving in the images. The algorithm managed to find 112 and detect 104 of these asteroids, which corresponds to 80% and 73% of the asteroids being tested respectively. An additional seven asteroids were found, along with 21 false positives.

A vast majority of the test asteroids that were not detected by the algorithm were faint and/or slow moving. Only looking at the test asteroids that have a velocity of 2.4"/min or faster and a visual magnitude of 20.5 or brighter, the algorithm detection rate was 96% (44

out of 46). Whereas with asteroids slower than $2.4''/\text{min}$ and fainter than magnitude 20.5, the percentage was down to 48% (19 out of 40). The detection of slow moving asteroids could be improved if the time between consecutive images was increased. More faint asteroids could be detected if the star mask was more efficient. An improved masking method would also decrease the number of false positives found.

6.1 Future Work

The way the poor quality section mask was made could be improved. There were three instances of the series of images producing two poor quality sections in the difference images. In the difference image in which the algorithm tried to locate the poor quality sections, the second poor quality section is not as prominent, which resulted in the second poor quality section not being masked. This caused the three asteroids to not be detected and four false positives to be.

Three of the test asteroids that were ignored were in star fields that were too crowded for the star position software. By finding a new method for the identification of stars that works well on crowded fields, these asteroids could be detected.

For a better test of the algorithm on slow moving asteroids, a series of images of slow moving asteroids could be taken where the time between the acquisition of consecutive images is longer. A better option might be to take both short and long inter-acquisition images of the same asteroid and see if the algorithm can detect the asteroid in either set of images.

Since the purpose of the algorithm was to find the asteroid, most of the asteroid velocities were inaccurate, as can be seen in some of the tables in Section 5.1. To get a more precise value for the velocity, a high resolution velocity phase space around the suspected velocity could be created.

References

- [Alard and Lupton, 1998] Alard, C. and Lupton, R. H. (1998). A Method for Optimal Image Subtraction. *Astrophysical Journal*, 503:325–331.
- [Albrow et al., 2009] Albrow, M. D., Horne, K., Bramich, D. M., et al. (2009). Difference imaging photometry of blended gravitational microlensing events with a numerical kernel. *Monthly Notices of the Royal Astronomical Society*, 397(4):2099–2105.
- [Allen et al., 2001] Allen, R. L., Bernstein, G. M., and Malhotra, R. (2001). The Edge of the Solar System. *Astrophysical Journal*, 549:L241–L244.
- [Allen et al., 2002] Allen, R. L., Bernstein, G. M., and Malhotra, R. (2002). Observational Limits on a Distant Cold Kuiper Belt. *Astronomical Journal*, 124:2949–2954.
- [Bond et al., 2001] Bond, I. A., Abe, F., Dodd, R. J., et al. (2001). Real-time difference imaging analysis of MOA Galactic bulge observations during 2000. *Monthly Notices of the Royal Astronomical Society*, 327:868–880.
- [Bramich, 2008] Bramich, D. M. (2008). A new algorithm for difference image analysis. *Monthly Notices of the Royal Astronomical Society*, 386:L77–L81.
- [Bramich et al., 2005] Bramich, D. M., Horne, K., Bond, I. A., et al. (2005). A survey for planetary transits in the field of NGC 7789. *Monthly Notices of the Royal Astronomical Society*, 359:1096–1116.
- [Chapman, 2004] Chapman, C. (2004). The hazard of near-Earth asteroid impacts on earth. *Earth and Planetary Science Letters*, 222:1–15.
- [Chiang and Brown, 1999] Chiang, E. I. and Brown, M. E. (1999). Keck Pencil-Beam Survey for Faint Kuiper Belt Objects. *Astronomical Journal*, 118:1411–1422.
- [Cochran et al., 1995] Cochran, A. L., Levison, H. F., Stern, S. A., and Duncan, M. J. (1995). The Discovery of Halley-sized Kuiper Belt Objects Using the Hubble Space Telescope. *Astrophysical Journal*, 455:342.
- [Elowitz et al., 1999] Elowitz, R. M., Stokes, G. H., Bezpalko, M., et al. (1999). A Progress Report on the Lincoln Near Earth Asteroid Research Project. In *American Astronomical Society Meeting Abstracts*, volume 31 of *Bulletin of the American Astronomical Society*, page 108.
- [Fraser and Kavelaars, 2009] Fraser, W. C. and Kavelaars, J. J. (2009). The Size Distribution of Kuiper Belt Objects for $D \gtrsim 10$ km. *Astronomical Journal*, 137:72–82.

References

- [Fraser et al., 2008] Fraser, W. C., Kavelaars, J. J., Holman, M. J., et al. (2008). The Kuiper belt luminosity function from $m=21$ to 26. *Icarus*, 195:827–843.
- [Fuentes et al., 2009] Fuentes, C. I., George, M. R., and Holman, M. J. (2009). A Subaru Pencil-Beam Search for $m_R \sim 27$ Trans-Neptunian Bodies. *Astrophysical Journal*, 696:91–95.
- [Gladman and Kavelaars, 1997] Gladman, B. and Kavelaars, J. J. (1997). Kuiper Belt searches from the Palomar 5-m telescope. *Astronomy and Astrophysics*, 317:L35–L38.
- [Gladman et al., 1998] Gladman, B., Kavelaars, J. J., Nicholson, P. D., Loredó, T. J., and Burns, J. A. (1998). Pencil-Beam Surveys for Faint Trans-Neptunian Objects. *Astronomical Journal*, 116:2042–2054.
- [Gladman et al., 2001] Gladman, B., Kavelaars, J. J., Petit, J.-M., et al. (2001). The Structure of the Kuiper Belt: Size Distribution and Radial Extent. *Astronomical Journal*, 122:1051–1066.
- [Gural et al., 2005] Gural, P. S., Larsen, J. A., and Gleason, A. E. (2005). Matched Filter Processing for Asteroid Detection. *Astrophysical Journal*, 130:1951–1960.
- [Harris, 2008] Harris, A. (2008). What spaceguard did. *Nature*, 453:1178–1179.
- [Hildebrand and Boynton, 1990] Hildebrand, A. and Boynton, W. (1990). Proximal creta-ceoustertiary boundary impact deposits in the caribbean. *Science*, 248:843–847.
- [Hsiao, 2014] Hsiao, C. (2014). On the thin film optics of a cassegrain telescope. *Thin solid films*, 570:471–474.
- [Kaiser et al., 2010] Kaiser, N., Burgett, W., Chambers, K., et al. (2010). The Pan-STARRS wide-field optical/NIR imaging survey. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7733 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 0.
- [Kargel, 1994] Kargel, J. (1994). Metalliferous asteroids as potential sources of precious metals. *Journal of Geophysical Research*, 99:21129 – 21141.
- [Larson et al., 2003] Larson, S., Beshore, E., Hill, R., et al. (2003). The css and sss neo surveys. *Bulletin of the American Astronomical Society*, 35:982.
- [Lightfoot, 2013] Lightfoot, R. (2013). Presentation on asteroid strategy.
- [Luu and Jewitt, 1998] Luu, J. X. and Jewitt, D. C. (1998). Deep Imaging of the Kuiper Belt with the Keck 10 Meter Telescope. *Astrophysical Journal*, 502:L91–L94.
- [Mainzer, 2011] Mainzer, A. (2011). Neowise observations of near-earth objects: Preliminary results. *Astrophysical Journal*, 743.
- [Mainzer et al., 2011] Mainzer, A., Grav, T., Bauer, J., et al. (2011). NEOWISE Observations of Near-Earth Objects: Preliminary Results. *Astrophysical Journal*, 743:156.

- [Morbidelli et al., 2002] Morbidelli, A., Bottke, Jr., W. F., Froeschlé, C., and Michel, P. (2002). Origin and Evolution of Near-Earth Objects. *Asteroids III*, pages 409–422.
- [Morbidelli and Vokrouhlický, 2003] Morbidelli, A. and Vokrouhlický, D. (2003). The Yarkovsky-driven origin of near-Earth asteroids. *Icarus*, 163:120–134.
- [Newberry, 1991] Newberry, M. V. (1991). Signal-to-noise considerations for sky-subtracted ccd data. *Publications of the Astronomical Society of the Pacific*, 103:122–130.
- [Shao et al., 2014] Shao, M., Nemati, B., Zhai, C., et al. (2014). Finding Very Small Near-Earth Asteroids using Synthetic Tracking. *Astrophysical Journal*, 782:1.
- [Stuart and Binzel, 2004] Stuart, J. S. and Binzel, R. P. (2004). Bias-corrected population, size distribution, and impact hazard for the near-earth objects. *Icarus*, 170(2):295 – 311.
- [Taylor, 1997] Taylor, J. R. (1997). *Error Analysis: The Study of Uncertainties in Physical Measurements*. University Science Books, 55D Gate Five Road, Sausalito CA.
- [Tonry, 2011] Tonry, J. L. (2011). An Early Warning System for Asteroid Impact. *Publications of the Astronomical Society of the Pacific*, 123:58–73.
- [Tyson et al., 1992] Tyson, J. A., Guhathakurta, P., Bernstein, G. M., and Hut, P. (1992). Limits on the Surface Density of Faint Kuiper Belt Objects. In *American Astronomical Society Meeting Abstracts*, volume 24 of *Bulletin of the American Astronomical Society*, page 1127.
- [Unknown, 2007] Unknown (2007). Near-earth object survey and deflection analysis of alternatives. Technical report, NASA.
- [Wetherill, 1979] Wetherill, G. (1979). Steady-state populations of Apollo-Amor objects. *Icarus*, 37:96–112.
- [Xu et al., 2007] Xu, R., Cui, P., Qiao, D., and Luan, E. (2007). Design and optimization of trajectory to Near-Earth asteroid for sample return mission using gravity assists. *Advances in Space Research*, 40:220–225.
- [Yanagisawa et al., 2005] Yanagisawa, T., Nakajima, A., Kimura, T., et al. (2005). Detection of Small GEO Debris by Use of the Stacking Method. *Japan Society of Aeronautical Space Sciences Transactions*, 44:190–199.
- [Zhai et al., 2014] Zhai, C., Shao, M., Nemati, B., et al. (2014). Detection of a faint fast-moving near-earth asteroid using the synthetic tracking technique. *The Astrophysical Journal*, 792(1):60.