

Library Declaration and Deposit Agreement

1. STUDENT DETAILS

Richard Ashley
1360981

2. THESIS DEPOSIT

- 2.1 I understand that under my registration at the University, I am required to deposit my thesis with the University in BOTH hard copy and in digital format. The digital version should normally be saved as a single pdf file.
- 2.2 The hard copy will be housed in the University Library. The digital version will be deposited in the University's Institutional Repository (WRAP). Unless otherwise indicated (see 2.3 below) this will be made openly accessible on the Internet and will be supplied to the British Library to be made available online via its Electronic Theses Online Service (EThOS) service. [At present, theses submitted for a Masters degree by Research (MA, MSc, LLM, MS or MMedSci) are not being deposited in WRAP and not being made available via EThOS. This may change in future.]
- 2.3 In exceptional circumstances, the Chair of the Board of Graduate Studies may grant permission for an embargo to be placed on public access to the hard copy thesis for a limited period. It is also possible to apply separately for an embargo on the digital version. (Further information is available in the Guide to Examinations for Higher Degrees by Research.)
- 2.4 (a) Hard Copy I hereby deposit a hard copy of my thesis in the University Library to be made publicly available to readers immediately.
I agree that my thesis may be photocopied.
- (b) Digital Copy I hereby deposit a digital copy of my thesis to be held in WRAP and made available via EThOS.
My thesis can be made publicly available online.

3. GRANTING OF NON-EXCLUSIVE RIGHTS

Whether I deposit my Work personally or through an assistant or other agent, I agree to the following: Rights granted to the University of Warwick and the British Library and the user of the thesis through this agreement are non-exclusive. I retain all rights in the thesis in its present version or future versions. I agree that the institutional repository administrators and the British Library or their agents may, without changing content, digitise and migrate the thesis to any medium or format for the purpose of future preservation and accessibility.

4. DECLARATIONS

(a) I DECLARE THAT:

- I am the author and owner of the copyright in the thesis and/or I have the authority of the authors and owners of the copyright in the thesis to make this agreement. Reproduction of any part of this thesis for teaching or in academic or other forms of publication is subject to the normal limitations on the use of copyrighted materials and to the proper and full acknowledgement of its source.
- The digital version of the thesis I am supplying is the same version as the final, hardbound copy submitted in completion of my degree, once any minor corrections have been completed.
- I have exercised reasonable care to ensure that the thesis is original, and does not to the best of my knowledge break any UK law or other Intellectual Property Right, or contain any confidential material.
- I understand that, through the medium of the Internet, files will be available to automated agents, and may be searched and copied by, for example, text mining and plagiarism detection software.

(b) IF I HAVE AGREED (in Section 2 above) TO MAKE MY THESIS PUBLICLY AVAILABLE DIGITALLY, I ALSO DECLARE THAT:

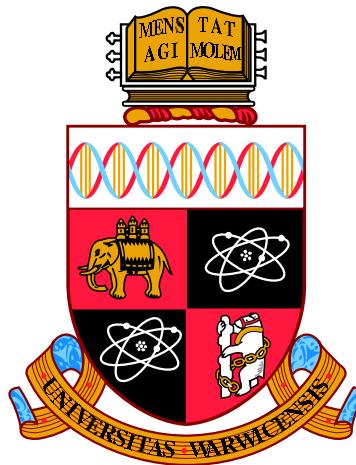
- I grant the University of Warwick and the British Library a licence to make available on the Internet the thesis in digitised format through the Institutional Repository and through the British Library via the EThOS service.
- If my thesis does include any substantial subsidiary material owned by third-party copyright holders, I have sought and obtained permission to include it in any version of my thesis available in digital format and that this permission encompasses the rights that I have granted to the University of Warwick and to the British Library.

5. LEGAL INFRINGEMENTS

I understand that neither the University of Warwick nor the British Library have any obligation to take legal action on behalf of myself, or other rights holders, in the event of infringement of intellectual property rights, breach of contract or of any other right, in the thesis.

Please sign this agreement and return it to the Graduate School Office when you submit your thesis.

Student's signature: Date:



Automated reduction of the Ultracam data archive

by

Richard Ashley

Thesis

Submitted to the University of Warwick

for the degree of

Master of Science by Research

Department of Physics

December 2014

THE UNIVERSITY OF
WARWICK

Contents

Acknowledgments	iv
Declarations	v
Abstract	vi
Abbreviations	vii
Chapter 1 Introduction	1
1.1 Background	1
1.2 Candidate objects	3
1.2.1 Eclipsing binaries	4
1.2.2 W UMa systems	4
1.2.3 Ellipsoidal variables	5
1.2.4 Cataclysmic variables	5
1.2.5 Intrinsic variables	6
1.2.6 Flare stars	7
1.2.7 Asteroids	7
1.3 The ULTRACAM instrument	8
1.3.1 Camera Optics	9
1.3.2 Filter sets	9
1.3.3 Field size and pixel scale	9
1.3.4 High speed operation	10
1.4 ULTRACAM data	13
1.4.1 Data capture	13
1.4.2 Typical run length	17
1.4.3 Run cadences	18

1.4.4	The data archive	20
Chapter 2	Automating the reduction of the imaging photometry	22
2.1	Data reduction for CCD images in Astronomy	22
2.2	Current method of ULTRACAM data reduction	26
2.3	Automating the pipeline	28
2.3.1	Motivation for automating the pipeline	28
2.3.2	Algorithm of the automated pipeline	28
2.3.3	Object matching	36
2.3.4	WCS solutions	45
Chapter 3	Creating a browse-able front-end for the data	51
3.1	Web browsers	51
3.2	Web technologies	52
3.3	The Web site	55
3.4	Accessing the data	55
Chapter 4	Results from the automated pipeline	57
4.1	Introduction	57
4.2	Quality of the photometry	58
4.3	Object matching	63
4.4	Finding variable objects	68
4.5	Reducing the entire ULTRACAM archive	74
Chapter 5	Objects identified by the automated pipeline	76
5.1	Object identification	76
5.2	Discovered objects	77
5.2.1	W UMa: 2005-05-10-run012-73	79
5.2.2	W UMa: 2013-07-21-run010-48	80
5.2.3	W UMa: 2013-07-21-run010-163	83
5.2.4	Eclipsing binary: 2013-07-21-run011-162	83
5.2.5	δ Scuti: 2013-07-21-run010-23	85
5.2.6	Asteroid: 1998 SU139	87
5.2.7	Asteroid: 910 Toruyusa	90

Chapter 6 Conclusion	91
6.1 Current status of the pipeline	92
6.1.1 Areas for immediate improvement	92
6.2 Future enhancements	94
6.3 Recommendations for ULTRACAM users	95
Appendix A Automated software user manual	97
A.1 Running the pipeline	97
A.1.1 Prerequisites	97
A.1.2 Installing the Python code	99
A.1.3 Config file	99
A.1.4 ucambuilder.conf	100
A.1.5 Producing the output for a particular run	102
A.1.6 Producing the output for a full night's observing	108
A.2 Using the archive	109
A.2.1 Night summary page	109
A.2.2 Run page	109

Acknowledgments

I would like to thank the following people who made this project possible.

Professor Tom Marsh for having the idea for this project and helping and guiding me through the development of the automated pipeline. Prof. Marsh also created the Python and C libraries which are extensively used by this project to read and extract the raw ULTRACAM data.

Matthew Green who worked through the light-curves of thousands of objects and found our most interesting variables.

Doctor Elmé Breedt for teaching me to use the current ULTRACAM data reduction pipeline and for assistance with many diverse queries.

This project makes use of software provided by Astrometry.net and Astromatic.net

Declarations

I declare that this thesis is my own, original work except where it has been explicitly stated within the work. In instances of collaborative research my own contribution has been indicated. Use of other published material is clearly acknowledged or referenced. I declare that my own original work has not been published prior to submission of this document and has not been submitted to another University for a degree at any level.

Abstract

Since May 2002, the Ultracam high-speed CCD photometric camera (ULTRACAM) has been taking observations of a variety of astronomical objects using the William Herschel Telescope (WHT), Very Large Telescope (VLT) and New Technology Telescope (NTT). Over this period it has produced approximately 10 Terabytes of raw data (CCD frames plus meta-data) taken on 406 observing nights and covering approximately 566 target objects. In this data there may be objects that hold scientific interest but have not been investigated since they were not the intended target object of the observer. Objects in the data need to be identified, listed and have reductions performed to determine their light-curves. In this project we have built a suite of software that is able to automatically reduce the full set of data residing in the ULTRACAM archive and produce light-curves for all of the objects identified in each run. This reduced data has been made available as a set of interactive web pages allowing a user to browse and review the archive. So far we have visually inspected the light-curves of about 20% of the objects and have found several variables that are not listed in any catalog. These are W Uma stars, δ Scuti stars and eclipsing binaries. We have also found and produced a light-curve for an asteroid passing through a field.

Abbreviations

WHT The *William Herschel Telescope* located at the Roque de los Muchachos observatory on the island of La Palma, Spain.

VLT The *Very Large Telescope* located at Cerro Paranal, Chile.

NTT The *New Technology Telescope* located at La Silla, Chile.

run Each data file generated by the ULTRACAM is called a 'run'. On every night that the camera is in use, a unique run identifier is generated from the combination of the date and an ordered sequence of run numbers. This means that any ULTRACAM run is uniquely identified by a run date and a run number. The format we will use to specify a unique run in this document is *YYYY-MM-DD/runXXX* where 'YYYY' is the year, 'MM' the month and 'DD' the date of the night on which the run was produced. 'XXX' is a unique run number on that date. For example: Run number 111 on the 13th of July 2013 is referenced as *2013-07-13/run111*.

ADU Analogue Digital Unit. The value returned for each pixel from the CCD chip after a readout. This is a measure of the flux arriving at the pixel.

Chapter 1

Introduction

1.1 Background

The ULTRACAM high speed photometry camera (hereafter, ULTRACAM) had its ‘first-light’ at the William Herschel Telescope (WHT) on the 16 May 2002. Since then it has been used on many occasions at three telescopes, namely the William Herschel Telescope (WHT), La Palma, Islas Canarias, the Very Large Telescope (VLT), Cerro Paranal, Chile and New Technology Telescope (NTT), La Silla, Chile. The camera is portable and can travel between these three telescopes. It was designed specifically to perform high speed photometry of faint objects in three colours. These design attributes are fundamental to understanding the scientific value of ULTRACAM and also set important context around the reason for doing this project.

Observations in astronomy very often rely on long exposures to capture enough light in order to perform the measurements. The general assumption is that the objects we are looking at do not change on very short timescales (< seconds). This is not true for all objects. Indeed, some objects are variable on short timescales of much less than 1 second. At the extreme end of this scale are pulsars, which spin at a rate of 100s of times per second, (Phinney and Kulkarni, 1994). When compact objects such as white dwarfs are eclipsed in a binary system, the light from the eclipsed object usually disappears within 2-3 minutes although some have ingress times as short as 30 seconds, making a steep eclipse shape. In fact, a sub-class of these stars, cataclysmic variables (CVs) with strong magnetic fields have a very compact bright region that can

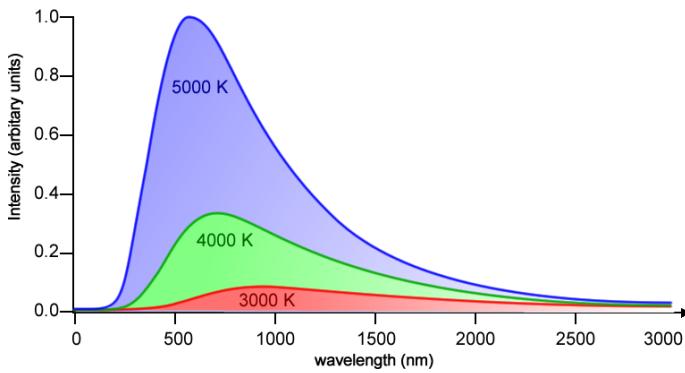


Figure 1.1: The distribution of energy as a function of wavelength for an ideal black body². Stars can be approximated as black bodies, radiating their energy according to this relationship. Hotter stars emit their flux at shorter wavelengths than cooler stars. By measuring an object's flux at various wavelengths, we can deduce information about the temperature of the object.

disappear in less than a few seconds, (Warner, 2003). The ingress and egress of exoplanets transiting the disc of their parent star can last for a few minutes or so, (Winn, 2010). Pulsating white dwarfs stars show variations in their light output on timescales of a few minutes, (Winget and Kepler, 2008).

ULTRACAM was designed to be installed on some of the larger telescopes in the world. The WHT has a mirror diameter of 4.2 metres, NTT 3.5 metres and the VLT 8.2 metres. Using large telescopes enables the high speed operation as the increased light gathering power of the large mirrors allows for shorter exposure times compared to what we would need at smaller telescopes. The combination of ULTRACAM with large telescopes like the VLT has enabled researchers to take high speed photometric measurements of targets such as GU Mus, which is a 20th magnitude (now in quiescence) X-ray nova with a time resolution of ~ 5 seconds, (Shahbaz et al., 2010). It can also take even higher speed data of cataclysmic variables, such as V834Cen, a 17th magnitude polar CV at a time resolution of ~ 0.05 seconds.

Since stellar objects can be treated (in the first approximation) as black-bodies, we can use the colour of the object to give us an indication of their temperature. This is because a black body will radiate at different wavelengths depending on its temperature according to Planck's Law, with the maximum

²Diagram taken from: <http://www.spaceflight.esa.int>

flux at a wavelength predicted by Wien’s Law, $\lambda_{max}T = 2.898 \times 10^{-3}$ mK. By using pre-defined filters we can measure the flux at different wavelengths and thereby derive colour and hence an indication of the object’s temperature. If the object is an eclipsing binary, the colour changes during the eclipse allows us to infer the temperature of each component in the system. During an exoplanet transit, subtle changes in colour during primary transit and also just before and after secondary transit, allows to determine the colour of the planet and therefore give us a clue to the composition of its surface and/or atmosphere, (Burton et al., 2012). For an intrinsic variable, such as a δ Scuti star, the colour changes during a pulsation cycle allow us to monitor the temperature changes on the surface during this cycle, (Aerts et al., 2010).

In nearly all observing runs there is a specific target object defined and the camera and telescope are set up to optimise the observations for this kind of object. Exposure times, filters and field sizes are chosen that are appropriate to the science data that is required. The ULTRACAM archive contains data that consists of measurements taken with a diverse range of these settings. Since the camera has is constantly taking images of a region that is several arc minutes in size (as shown in table 1.1), it will also capture data for any other objects that just happen to be in the field (or ‘passing-through’ the field) during the run. One of the main reasons for automating the reduction pipeline is that we might serendipitously, discover objects that, although not the intended targets of the run, are nevertheless, displaying some kind of variability. Depending on the size and orientation of the windows that the observer has selected and how crowded the sky in the field of view is, we could have many objects recorded in the raw data that have not had light-curves produced. Producing and analysing these new light-curves will hopefully allow us to discover new variable objects.

1.2 Candidate objects

In this section we list some of the object classes that could appear by chance in the ULTRACAM data.

1.2.1 Eclipsing binaries

Many of the stars in the galaxy are not isolated, solitary stars, but reside in a multiple star systems. It is estimated that the field star population in the solar neighbourhood consists of 50% binary systems, (Eggleton and Tokovinin, 2008). A fraction of these systems will be viewed edge-on from Earth ($i \sim 90^\circ$) and therefore eclipses will occur at least once per orbit. Eclipses are seen as a drop in flux. The eclipse shape indicates the relative sizes of the two objects in the system. If the two objects differ in temperature, then the eclipse will show a change in colour which can be estimated using the relative fluxes measured in the 3 channels. Since the typical run length for ULTRACAM is about 1-3 hours, we are biased towards detecting binaries with short orbital periods (<1 day).

1.2.2 W UMa systems

W UMa systems are binary systems in which both stars have filled their Roche lobes and their atmospheres are effectively merged, although their cores are separate and in orbit about each other, (Lucy, 1968). Their outer layers appear to be at similar temperature over the full extent of both their shared common envelope. Since each of the stars are not necessarily of equal mass, their cores will be at different temperatures according to the mass-temperature relationship for main sequence stars, $T \propto M^{0.65}$. As these objects rotate, we see a change in flux due to the non-spherical shape of each star as they present different surface areas towards the Earth. Since the stars are in contact with each other, eclipses are visible across a wide range of inclinations ($90^\circ > i > 30^\circ$). The depth of this flux change will depend on the inclination of the orbit. If we are seeing the orbit pole-on, with inclination, $i \sim 0$ then we will see very little variability. At inclinations of $i \sim 90$ we will see the maximum drop in flux of approximately $\sim 50\%$. Since the temperatures of these bodies are similar there is little change in colour during the orbital cycle. Their periods fall in the 6 to 20 hour range and should therefore be obvious in the ULTRACAM data for runs of duration 30 minutes or more.

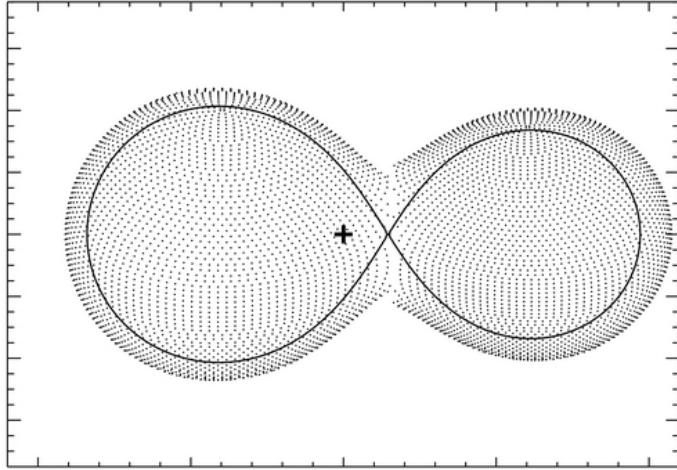


Figure 1.2: Diagram showing the layout of a W UMa contact binary. The surface of the outer envelope is defined by a Roche equipotential. The + symbol denotes the centre of gravity of the system. From Davenport et al. (2013).

1.2.3 Ellipsoidal variables

Stars in binaries that are not in contact but still sufficiently close to each other to cause tidal distortion of one or more of the components will show variability for the same reasons as the W UMa stars. The projected area of the star presented towards the Earth varies over the orbital cycle. The typical change in brightness of an ellipsoidal variable is about 0.1 magnitudes or a 10% change in flux. Orbital periods should be similar to those of the W UMa category, about 6-20 hours and should be visible in the ULTRACAM data.

1.2.4 Cataclysmic variables

Cataclysmic variables are objects containing two stars in a semi-detached state with one of the stars having filled its Roche lobe and streaming material onto the companion. The companion in this case is a white dwarf. The white dwarf has a mass between $0.4M_{\odot}$ and $1.4M_{\odot}$, yet a radius that is similar to the radius of the Earth. The white dwarf's surface is, therefore, a long way from the gravitational equilibrium point where this flowing material leaves the donor star and starts to fall inwards. Since the material has its own angular momentum, it cannot fall directly toward the white dwarf, but spirals inward,

usually forming an accretion disc through which it eventually migrates onto the white dwarf's surface.

Cataclysmic variables are highly variable on many different timescales. On the timescale of centuries, they can undergo nova explosions where they explode a shell of hydrogen that has been built up on their surface, increasing their brightness by 6-19 magnitudes. Over a period of weeks to months, their accretion disks can brighten dramatically in 'outbursts', increasing by 3-8 magnitudes. On the timescale that is most relevant to a typical ULTRACAM run, we can expect to see eclipses of the white dwarf, bright-spot and disc (assuming this is an eclipsing system) and flickering caused by the accretion stream flowing onto the bright-spot.

For a complete overview of the field of study of cataclysmic variables, refer to Warner (2003).

1.2.5 Intrinsic variables

RR Lyrae stars

RR Lyrae stars are horizontal branch stars that have evolved away from the main sequence and are in the instability strip. Pulsations in RR Lyraes are driven primarily by helium ionisation zones in their interiors. The mechanism by which opacity drives pulsations is known as the κ mechanism, (Handler, 2013). They exhibit periods of several hours to a few days and their light curves are usually non-sinusoidal (with harmonics) and often have a 'sawtooth' shape. The amplitude of the brightness variation can be between 0.3 and 1.2 magnitudes (corresponding to a change in flux by a factor of 1.3 to 3 times). The colours change significantly during the cycle as the surface temperature rises and falls and this should be evident in the ULTRACAM data.

δ Scuti stars

δ Scuti stars are driven by similar mechanisms to the RR Lyraes but these are generally more massive stars and are still on the main sequence. They exhibit non-radial pulsation modes and therefore have shorter characteristic periods and smaller amplitudes. Typical periods for the oscillations in δ Scutis range from 18 minutes to 8 hours, with amplitudes up to ~ 0.1 mag, (Aerts

et al., 2010). Like the RR Lyraes, the surface temperature of the star changes during the pulsation cycle and we expect to see a colour modulation in the light-curve.

Pulsating White Dwarfs

White dwarf stars pass through a region of the HR diagram that can be seen as an extension of the instability strip. In this region, the white dwarf will experience a similar driving mechanism to that which drives pulsations for the main sequence and horizontal branch stars, namely the κ mechanism. The zone where this mechanism is active is the white dwarf's thin hydrogen or helium atmosphere. Pulsating hydrogen white dwarfs are known as DAVs or ZZ Ceti stars and helium white dwarfs as DBVs. Usually they have a fairly complex oscillation pattern, with many frequencies, as they oscillate with many modes excited. Nevertheless, a pulsating white dwarf should be fairly obvious when observed with ULTRACAM. The oscillations will have periods of a few minutes and amplitudes on the order of 0.1 mag.

Pulsations in white dwarfs stars are reviewed in Winget and Kepler (2008).

1.2.6 Flare stars

Flare stars are usually red dwarf stars that undergo flares in their atmospheres resulting in rapid changes in brightness on timescales of minutes to hours. Typical flare rates for the flare star UV Cet are about every 2.5 to 6 hours. In the optical region we expect the flares to appear as impulse rises with an exponential decay lasting minutes to hours. During the flare, we can expect the intensity to double, (Paganano, 2013). Flares have also been shown to exhibit colour changes and there is evidence for an anti-correlated time-evolution between the relative flux and the flare colour. ULTRACAM has been used to measure colour changes during flare events, (Kowalski et al., 2011).

1.2.7 Asteroids

Solar system objects such as asteroids should be visible in the ULTRACAM archive. Main asteroid belt and near-Earth objects are likely to move across

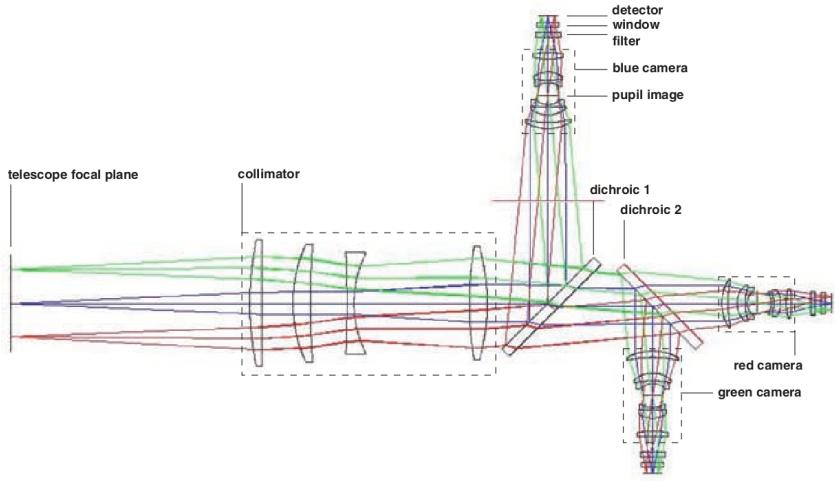


Figure 1.3: A ray-trace through the ULTRACAM optics, showing the major optical components: the collimator, dichroics, cameras, filters and detector windows. The diagram is to scale the largest lens is in the collimator and has a diameter of 120 mm. The diagram is taken from Dhillon et al. (2007).

the field at a rate of a few arc seconds per minute, meaning that they would cover a fair fraction of the exposed CCD during the course of a 1-2 hour run of the ULTRACAM. We can expect a ~ 300 m diameter near-Earth object to have a magnitude in V of around 15, (Vaduvescu, 2005) which is an appropriate brightness for most ULTRACAM observations.

Kuiper belt objects will have apparent magnitudes of around $V \sim 20$ and would move only a few arcseconds per hour. It is fairly unlikely (although not impossible) that we might record one of these objects in an ULTRACAM run.

1.3 The ULTRACAM instrument

ULTRACAM is an ultrafast, triple-beam, cascade dichroic CCD camera, and has helped open up simultaneous multiband, sub-second time domain astronomy, (Dhillon et al., 2007).

1.3.1 Camera Optics

The camera has three CCD detectors enabling it to capture data in three colour bands simultaneously. Two dichroic beamsplitters divide the light from the collimator into three beams, which shall hereafter be referred to as the ‘red’, ‘green’ and ‘blue’ channels. The three CCD detectors are mounted at right angles to each other on the camera. Therefore, each detector is at the end of a slightly different optical path. The images produced on each of the three CCDs chips are therefore of the same field but with very slightly different orientations, distortions and offsets. Towards the edges of the chips, these differences can be on the order of 10 pixels from channel to channel.

In general, the exposures are synchronised across all three detectors, meaning that all three CCDs start their exposure, stop their exposure and read-out at the same time. It is, however, possible to have the detector in the blue channel remain exposed and not read-out while the other two are going through multiple exposures and read-outs. This is to allow for longer exposures where there might be less flux in the blue CCD. Reduced flux in the blue CCD is caused by several factors, including lower transmission of the optics and atmosphere for blue light, the reduced sensitivity of the CCD detector to blue light and the low intrinsic flux of most astronomical objects in this region of the spectrum.

1.3.2 Filter sets

The filters for each channel can be altered by the observer. In usual configurations, the SDSS filters (u , g , r , i , z) are used, but there are a selection of narrow-band filters that can be substituted. Depending on the scientific measurements that the observer is trying to perform, these filters can be changed. Figure 1.5 shows the response curves of the ULTRACAM camera with the SDSS filter set.

1.3.3 Field size and pixel scale

ULTRACAM is usually mounted on one of the three telescopes mentioned in the introduction of this chapter (VLT, NTT, WHT). Field sizes, pixel scales

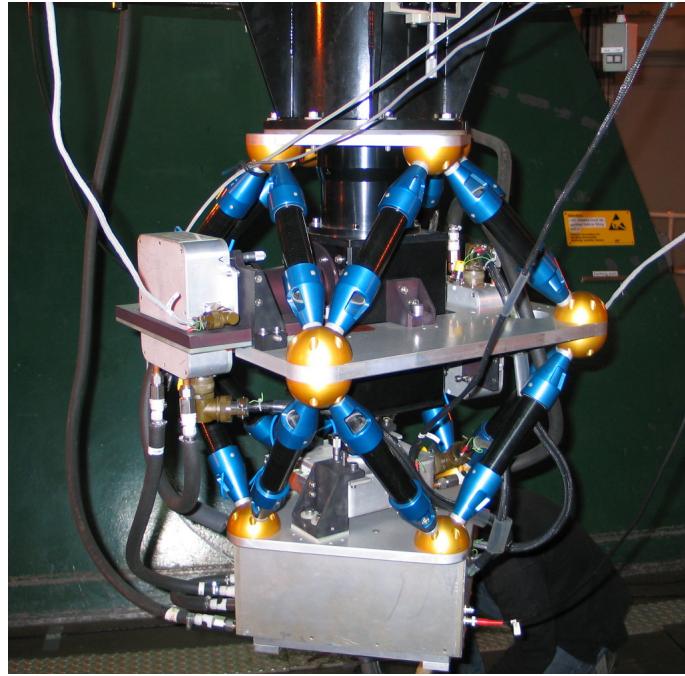


Figure 1.4: The Ultracam being commissioned in May 2002.

and orientations are summarised in table 1.1. The orientations quoted refer to when the camera is not rotated.

Telescope	Field size (arc minutes)	Pixel scale (arc seconds/pixel)	Orientation
WHT	5.0x5.0	0.30	N(up), E(left)
NTT	6.0x6.0	0.35	N(up), W(left)
VLT	2.5x2.5	0.15	N(up), W(left)

Table 1.1: The field sizes and pixel scales of ULTRACAM on each of the three telescopes.

1.3.4 High speed operation

A key aspect of the design of the camera is its ability to perform at high cadence, or frames per second. It is possible to have the camera read-out at up to 500Hz (frames per channel per second), (Dhillon et al., 2007). This makes the camera useful for observations of rapid transient events with accurate timing. Although the camera is not often used in this very high speed mode, there are a few observing runs where the camera has been used for measurements with

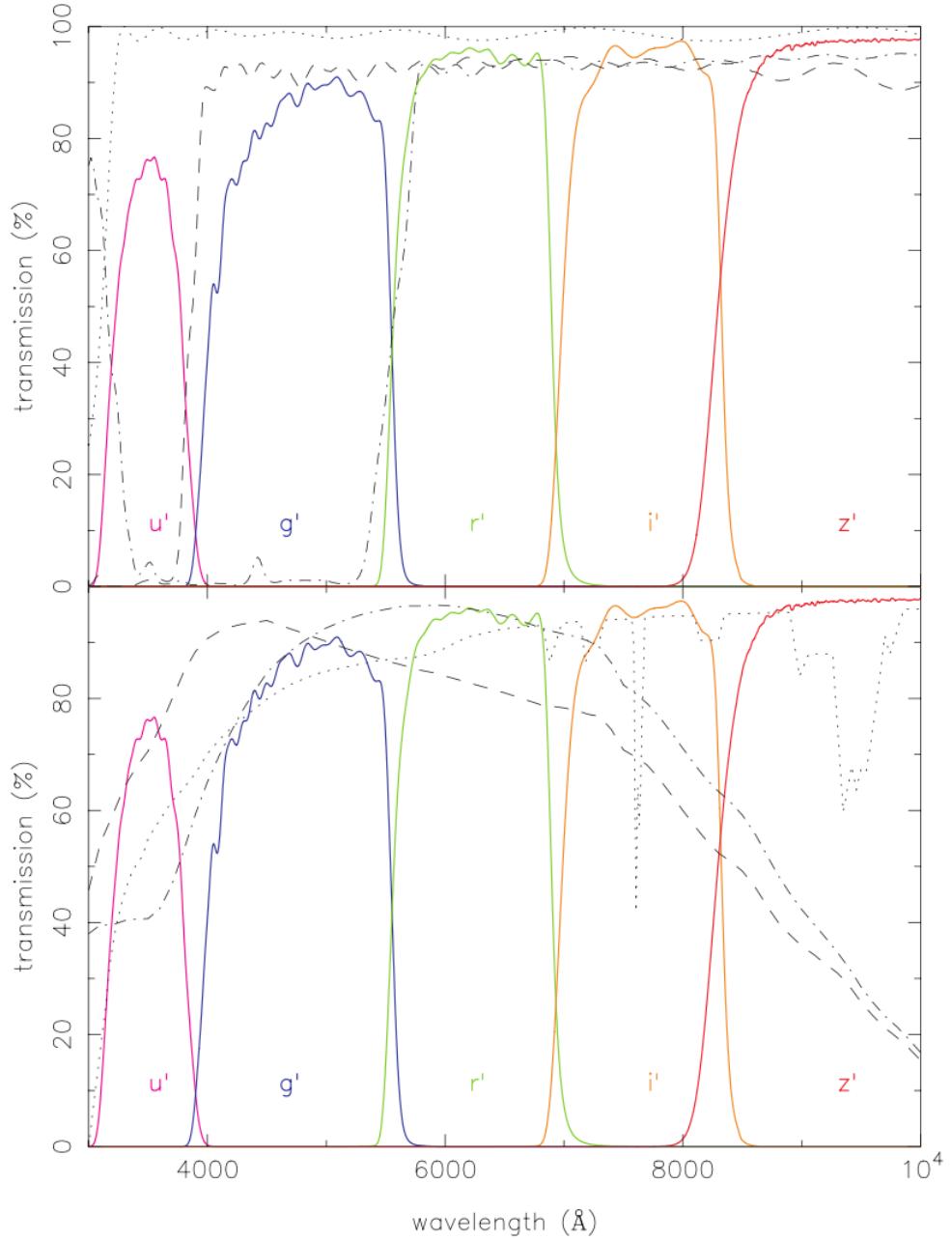


Figure 1.5: The response curves of ULTRACAM with the SDSS filter set. The upper plot shows the SDSS filter set, the response of the antireflection coating used on the lenses (dotted line) and the two dichroics (dashed line and dash-dotted line). The lower plot shows response of the SDSS filter set and the atmosphere (per unit airmass). The figure is taken from Dhillon et al. (2007).



Figure 1.6: One of the three CCD detectors. The masked-off area is visible in the lower half of the chip surface.

exposure times of approximately 0.005 seconds. These runs are described in chapter 4.

Each CCD has a total pixel area of 2057x1024 pixels. Half of these pixels are masked and never exposed to light. They are used as a temporary buffer for reading out the information capture on the CCD. CCD detectors are normally read-out serially, but in order to decrease the time between exposures, the full image can be moved from the imaging area of the CCD to the storage area and this can then be read-out while the imaging area is once again exposed to light.

ULTRACAM gives the observer the option to reduce the area of the detector that is used for the recording of data. This reduces the chip read-out time and enables the rapid operation of the camera. Reducing the number of pixels recorded also decreases the amount of data storage needed for the run. The observer can define pairs of windows that are centred on the target objects. By making the windows suitably small, the observer can use the camera in extremely high cadence mode.

The highest cadence mode is called *Drift mode*. This mode uses the storage area of the CCD to store several exposures simultaneously. Only a portion of the imaging area of the CCD is shifted into the storage area. The fact that the camera is not shifting the whole of the imaging area means that

it is ready to be re-exposed sooner than if it was transferring the full-frame. This mode requires that only the lower portion of the imaging area, close to the boundary of the masked and un-masked areas, is selected used for the exposures. This means that the camera has to be rotated so that the target object (and a suitable comparison star) are positioned correctly. ULTRACAM is therefore designed to be rotated about the optical axis of the telescope in order to allow for this specific object placement. For any particular run, it is possible that we might have any orientation ($0 - 360^\circ$) of the camera relative to the sky coordinates. The ULTRACAM logs do not record this rotation angle. This is an important point to remember when we try to find astrometric solutions for the runs.

More details on the camera design and operation can be found in Dhillon et al. (2007).

1.4 ULTRACAM data

1.4.1 Data capture

Usually the camera remains installed on the telescope for a week or so and is used for observations on consecutive nights. Each separate recording of data is called a run. On most nights, many runs are recorded. A run can be defined as a period when the camera is active and gathering data. Not all runs are used for gathering scientific data. Some runs are used for target acquisition and camera calibration.

The types of runs are:

- *Science runs*: These are the runs that contain the valuable scientific data. They usually comprise the longest portions of the observations during the night, unless the camera is having difficulties or adverse weather conditions are preventing useful astronomical observations.
- *Acquisition runs*: These are runs, usually of short duration (ie a few minutes) during which the telescope is being moved in order to place the candidate object(s) in the field of view. The camera may also be rotated in order to align the CCD such that the targets avoid ‘bad’ pixels or are

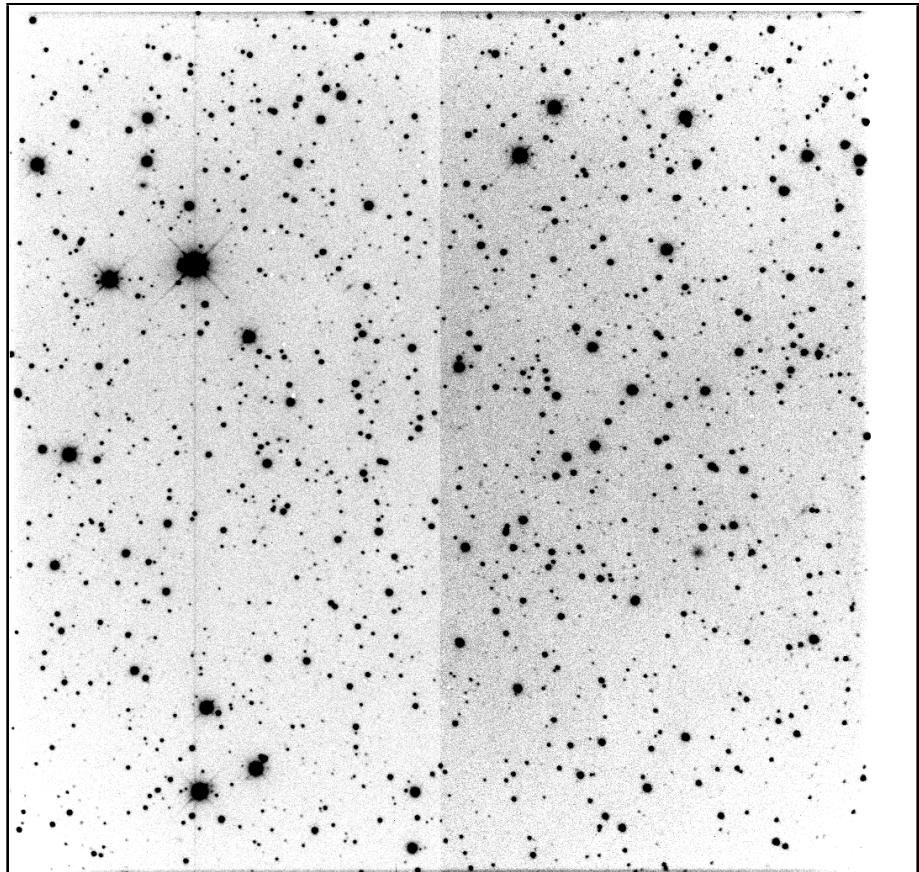


Figure 1.7: A 'fully' exposed CCD with 1 pair of windows (512x1024 pixels each). This mode reads out the full area of the CCD chip. The field of view in this image is approximately 5x5 arc minutes since it was taken with the ULTRACAM mounted on the WHT. The field of view is 6x6 arc minutes and 2.5x2.5 arc minutes when ULTRACAM is mounted on NTT and VLT respectively.

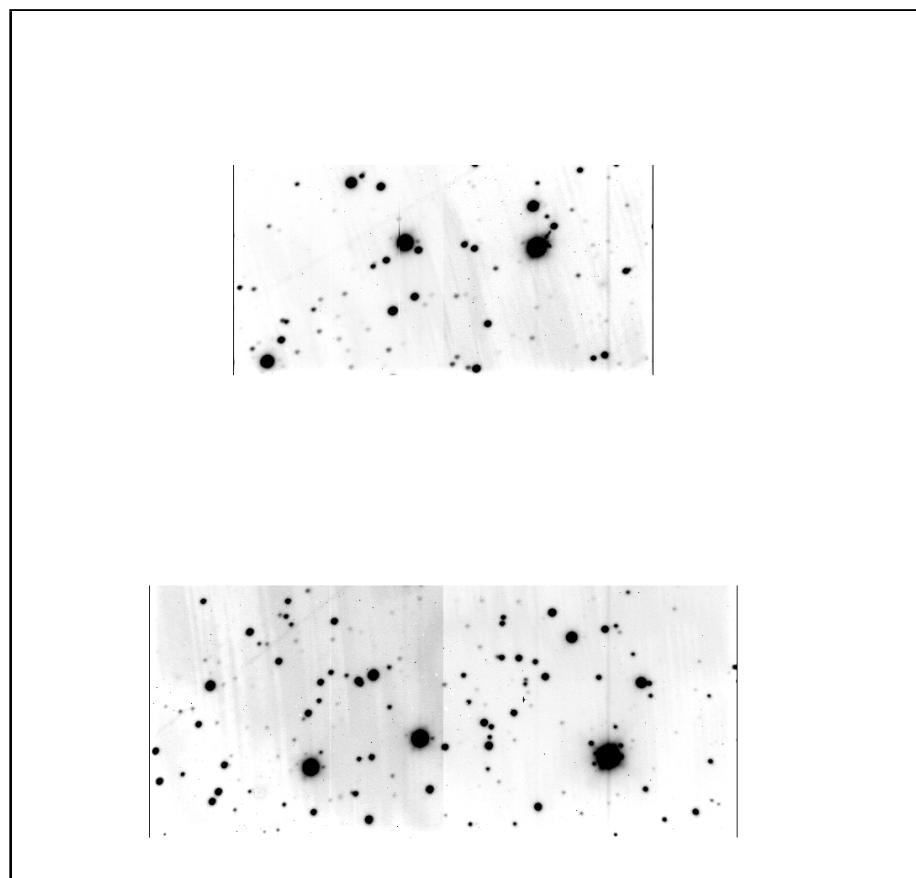


Figure 1.8: A 'masked' exposure with 2 pairs of windows (350x300 and 250x250 pixels each, respectively)

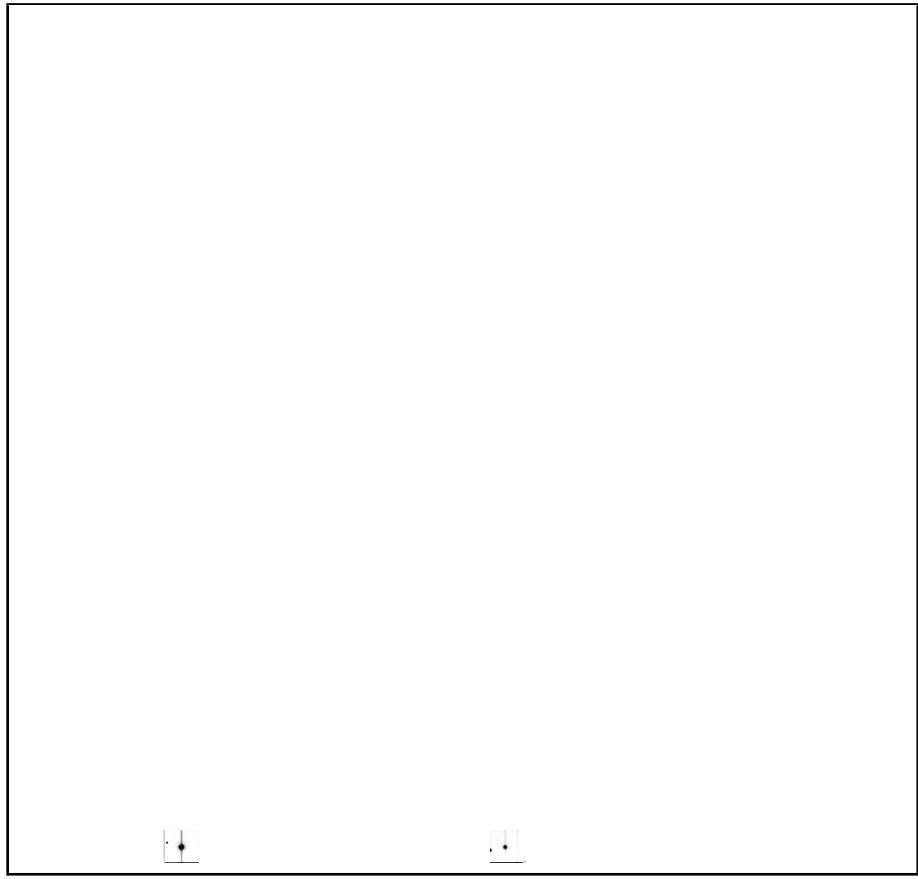


Figure 1.9: The camera operating in *drift-mode*. Note the very small windows (172x156 pixels) located at the bottom of the imaging area.

near to the lower boundary of the detector (eg for high speed readout in drift mode).

- *Flat fields*: At the start and the end of the night (usually during twilight) the observer will take a few runs to create *sky-flats* that will be used later for calibrating the variations in pixel sensitivity across each of the detectors. Sky-flats are generated by exposing the camera to patches of sky during the twilight.
- *Biases*: Are a set of short exposures with the CCD not exposed to light to build calibration readings for measuring the bias of the detector. This bias will be subtracted from these data during the reduction.
- *Timing calibration runs*: One way to check the timing calibration of the camera is to take frames of a well-known, rapidly oscillating source, for example, the Crab Pulsar (PSR B0531+21). The timing of the optical pulses as measured by the camera can then be compared to the expected times for the pulsar. This is used as a standard clock for timing calibration.
- *Darks*: Dark frames are taken with the camera exposed to no light at all, or as close to no light as is physically possible. They are taken over a range of exposure times similar to the exposure times that are used during the science runs and with the detector at a similar temperature. The purpose of these dark frames is to correct for the gradual accumulation of electrons in the pixels of the detector due to thermal noise. The three ULTRACAM sensors are Peltier cooled to $\sim 233\text{ K}$ and at this temperature are expected to deliver a dark current of $\sim 0.05\text{ e}^-\text{ pixel}^{-1}\text{ s}^{-1}$. This is significantly lower than the expected sky background of $\sim 0.3\text{ e}^-\text{ pixel}^{-1}\text{ s}^{-1}$ for this camera at a typical site, (Dhillon et al., 2007).

1.4.2 Typical run length

Since ULTRACAM is designed for high-speed photometry, observers using the instrument are typically looking for variations that are clearly noticeable on timescales from a few minutes to a few hours. Most science runs last for a

few hours at the most. The longest runs are usually observations of exoplanet transits which can last from about 4-7 hours. Sometimes these have a break at about the middle of the run if the telescope goes through the zenith. All three of the telescopes on which ULTRACAM is usually mounted employ the ‘Alt-az’ design, rather than an ‘Equatorial’ mount. Alt-az designs cannot observe directly at the zenith and the run is interrupted for 3 minutes or so, while the telescope is repositioned after the zenith ‘blind-spot’.

A look at the distribution of run length shows a large bulk of the runs are shorter than 5 minutes. This is because there are far more acquisition runs, flat-fields and bias runs than there are science runs. There are also many runs that are nominally science runs (and can be used as such) but are short because the observer has noticed something that they would like to change. This could be an adjustment of the focus, binning factor or integration time. The run is then cut short and a new run is started. The longest run length in the 10 year data archive is 566 minutes or 9.5 hours, taken on 25th of April 2010 at the NTT. This was for the observation of a transit of the exoplanet Wasp-15b. The output for this run can be seen at <http://deneb.astro.warwick.ac.uk/phrnaw/sitedev/2010-04-25/run020.html>.

1.4.3 Run cadences

As mentioned in the section describing the camera, ULTRACAM was specifically designed for high-speed photometry. Certain phenomena in astrophysics have observable variability in the optical region that is apparent over relatively short periods. Ultra-compact binaries, cataclysmic variables, neutron stars, etc all have optical variability that can be resolved at even sub-second intervals. ULTRACAM is mounted on some relatively large telescopes, with mirrors ranging from 4.2m - 8.2m in diameter. This means that it is able to perform high-speed photometry of relatively faint objects.

Only certain objects require the highest cadences (<1 second). These are X-ray binaries, polars, pulsars and flare stars. Many other science runs can use the camera with a >1 second exposure time. Long runs for exoplanet transit observations often use longer exposures of 2-3 seconds. The longest exposure times are around 20-25 seconds. This is only required when the

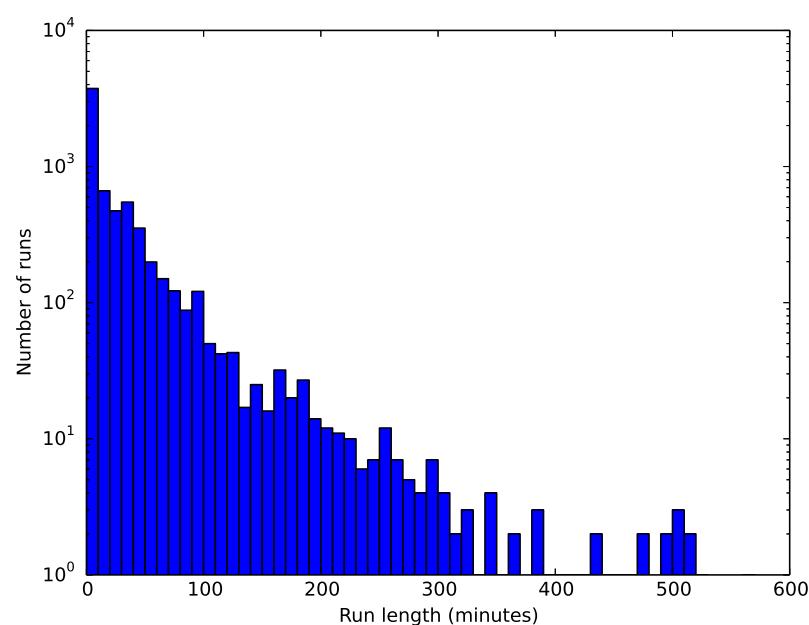


Figure 1.10: Distribution of run length. Many runs are shorter than 5 minutes, but these are not usually *science runs*. Note that this is plotted on a logarithmic scale.

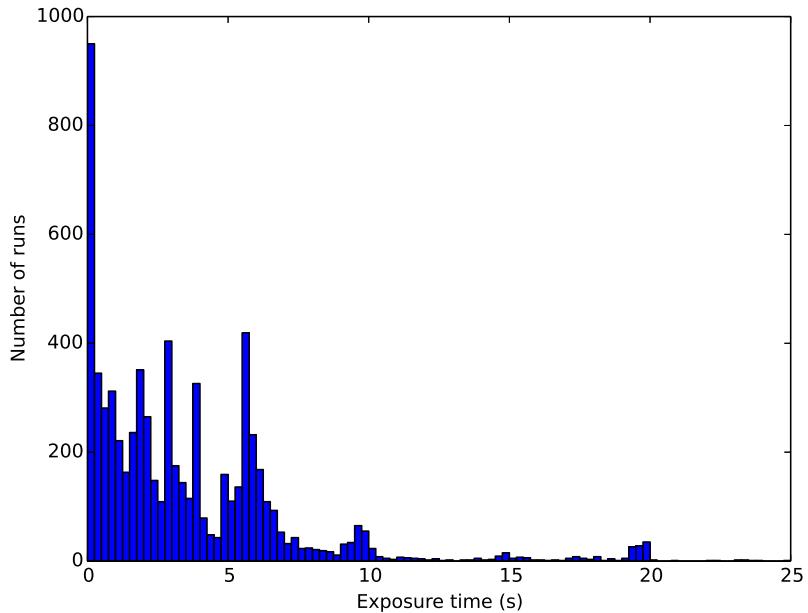


Figure 1.11: Distribution of exposure times used in the *science* runs.

object being observed is extremely faint (>19 magnitude).

Figure 1.11 shows a distribution of exposure times by run, for all of the *science* runs in the ULTRACAM data archive. There are several groupings apparent in the histogram. Firstly, the very short exposures for the rapidly variable objects, such as polars, pulsars and X-ray binaries. Then there is a cluster of runs with exposure times of 2, 3, 4 and 5.5 seconds. These are typically observations of eclipsing white dwarf binaries. The next cluster occurs at about 10 seconds, which are usually runs for exoplanet transits. The final cluster of run lengths occurs at around 20 seconds and is usually for very faint objects of magnitude >19 .

1.4.4 The data archive

At the time of writing, the ULTRACAM data archive is >10 terabytes. This can be broken down as:

- 406 nights on which ULTRACAM was operational at a telescope.
- 12 649 runs, including science runs, acquisition runs, flat fields and

biases.

- *119 817 742* frames in total. This total includes all the frames for each channel: red, green and blue.
- *10.54* terabytes of raw image data.

The data set is relatively large and is housed on a network-mounted storage device that is only available through the internal university computer network. This means that it is not possible to access these data from remote locations (for example, by research collaborators in different institutions). If a researcher needs to access the output of an observing run, then they need to contact a member of department at Warwick or Sheffield and request a data reduction. There is no means of accessing or exploring the ULTRACAM data set from a remote location. Providing a simple means of accessing these data from remote locations would benefit all of the research collaborators and is one of the aims of this project.

Chapter 2

Automating the reduction of the imaging photometry

2.1 Data reduction for CCD images in Astronomy

Since the late 1980's, Charge Coupled Devices (CCDs) have risen to prominence in astronomy. Before CCDs, high-speed photometry in the optical was accomplished using photomultiplier tubes. Photomultipliers are high tension (high voltage) devices that detect faint light by using the photoelectric effect followed by amplification. When a photon impinges on the detector, an electron is ejected, this electron is then 'amplified' by a series of voltage steps until the resulting signal is read out by an analog to digital convertor. The main problem with this configuration is that the photomultiplier only has one detection element and the starlight needs to pass through a physical aperture in the camera so that only light from the target object is measured. Using a single photomultiplier, it is not possible to measure multiple objects simultaneously.

CCDs are 2-dimensional detectors covered with light sensitive semiconductors (each one defining a 'pixel') that are able to measure and count photons. The structure and construction of CCDs depend on their intended purpose. Consumer electronic devices such as the cameras in mobile phones or digital cameras use a different arrangement and read-out process to that used in astronomical CCDs. In order to decrease read-out times, consumer

devices use an architecture called 'interline' where the read-out pixels and electronics are located in alternate rows of the CCD. This means that the every alternate row of the CCD is masked from light and used for readout only. The disadvantage of this architecture is that the 'fill-factor' or area that is sensitive to light of the CCD is now reduced to 50%. Astronomy purpose CCDs usually read out the entire chip line-by-line before the next exposure can begin. For higher frames rates in astronomy full-frame transfer devices are used. These devices are divided into 2 equal areas. One half of the device is exposed to light and the other half is masked. After each exposure, the exposed (or imaging) area is shifted completely into the masked (or storage) area. The imaging area is now ready to be exposed again, while the image can be read out from the storage area. This architecture has the advantage that it allows more time for the read-out process to take place. A slower read out process introduces less noise into the output. The fill factor of the imaging area is closer to 100% as none of the pixels are masked. There is still a small gap between pixels though so we don't quite reach a fill-factor of 100%.

CCDs can be sensitive to optical, infra-red and ultraviolet light depending on the materials used and their manufacture. In ULTRACAM, all three of the CCDs are E2V 47-20 CCDs. They are frame-transfer chips with imaging areas of 1024×1024 pixels and storage areas of 1024×1033 pixels. To improve quantum efficiency, the ULTRACAM chips are thinned, back-illuminated and antireflection coated with E2V's enhanced broad-band astronomy coating (in the case of the blue and green chips) and standard mid-band coating (in the case of the red chip), (Dhillon et al., 2007).

Since the CCD is recording a 2-dimensional image of the sky, it will capture multiple objects simultaneously. In the 2-dimensional image it is possible to create a virtual aperture as part of the reduction process. Physical apertures (used with photomultipliers) needed to be large enough to account for changes in seeing and drifts in the target object's position as the telescope tracks the sky. Large apertures are a problem since the sky starts to contribute a significant amount of the total flux in the aperture. Nights with poor seeing and variable sky conditions were often not photometric in the days of photomultipliers. Nowadays, thanks to the nature of the CCD imaging, we can still get photometric data from nights like this. Although early

CCDs lacked the quantum efficiency of photomultipliers, this limitation has since been surpassed. Photomultipliers are rarely used nowadays, at least for optical astronomy. They are still used to detect photons from scintillations in Cerenkov detectors and neutrino detectors.

In order to extract photometry from a CCD image, we need to perform several steps to convert the 2-dimensional image into a 1-dimensional flux measurement for each object in the image. This is known as ‘reduction’. Reduction in CCD photometry can be summarised as follows:

Bias subtraction: CCDs have an intrinsic noise known as readout noise. In order to ensure that this noise does not fluctuate around the value zero with the potential therefore of having a negative number in the output we set an arbitrary bias for the CCD to ensure that each pixel always reads out a positive value. Since we are storing our pixel ADU values as 16-bit unsigned integers, negative numbers would appear as large ADU values ($65535 - \text{value}$). During the night, bias frames are recorded by taking CCD readings without exposing the CCD to any light. These bias frames are then subtracted from each science CCD frame.

Flat fields: We cannot be certain that each pixel in the CCD array has the same sensitivity to light as its neighbours. This is a problem if we are trying to measure the flux of an object very accurately. If the object’s light falls on different pixels in each subsequent exposure, then the pixel sensitivity will impact our measurement of the object’s true flux. To measure each pixel’s relative sensitivity, we expose the entire imaging area to a uniform source. The source is either a white screen mounted on the inside of the dome, or, more commonly, an exposure of the sky during twilight. This flat-field is then ‘normalised’ by dividing each pixel’s count by the average of all the pixels. Each pixel’s sensitivity can then be factored into our reduction of a science run.

Source extraction & aperture creation: Our exposed 2-dimensional image can have many sources on it and we need to pick out the objects of scientific interest. We also need to ignore spurious image artifacts such as those caused by cosmic rays. Source detection can be automatic, as we will use in this project, or manual, as performed with the current ULTRACAM reduction pipeline. The source extraction stage will produce a list of apertures. The

apertures are defined by having a position (usually the centroid of the object) and a 2-dimensional area, which is usually a circle centred on the object's position that captures all the light from the object but excludes light from any other nearby objects. These apertures are virtual; defined in $(x, y, radius)$ parameter space, rather than physical; as in the earlier photomultiplier devices. In this project, we are using SExtractor, a popular, third party software package to perform the source extraction and aperture definition, (Bertin and Arnouts (2006)).

Sky subtraction & Flux measurement: The Earth's atmosphere is not completely dark at night and still glows blue with light scattered from the stars, reflected from the Earth or scattered from the Moon. This sky background needs to be taken into account and subtracted from the total flux measured in the aperture in order to leave us with the flux that is contributed by our target object. There are two main methods of dealing with the sky background. The first is to derive an overall sky-background for the image, which is then subtracted from the flux measured in the aperture. The flux remaining is then the flux contributed by the target object. This overall sky background is not a single value, but a polynomial fit in order to allow for smooth variations across the field.

The second approach is to measure the flux in an annular aperture centered around the target object that is close enough to the object to have a similar sky reading, but also far enough away to ensure that no significant flux is contributed by the object itself. The flux per unit pixel in the annulus is subtracted from the flux per unit pixel in the object's aperture in order to remove the sky background. Calculating the size of the aperture and the inner radius of the annulus is aided by fitting a Moffat profile to the star's point-spread-function (PSF). The PSF is the two dimensional profile of the image as recorded on the CCD. We could choose to fit either a Moffat profile or a Gaussian profile to match the PSF, but the Moffat profile is favoured as its shape matches the real PSF of the star's image as it was derived by convolving atmospheric seeing profiles with telescope diffraction profiles, (Moffat, 1969).

The formula for the Moffat profile is

$$I_r = I_0 \left[1 + \left(\frac{r}{\theta} \right)^2 \right]^{-\beta}$$

where I_0 is the intensity at image center, θ is the half width at half

maximum (FWHM) of the image in the absence of atmospheric scattering and β is the atmospheric scattering coefficient. By fitting a Moffat profile to the star's PSF we can derive a value for the FWHM of the star's image. This value gives us a radius to use to set the radius of the aperture and the inner and outer radii of the sky annular aperture. These values are generally some pre-defined constant multiplied by the FWHM.

2.2 Current method of ULTRACAM data reduction

Tom Marsh at the University of Warwick has developed a set of software tools for reduction of these ULTRACAM data. For the rest of this document we will refer to this pipeline as the *traditional* pipeline and the new pipeline created in this project as the *automated* pipeline.

It is possible to run the traditional pipeline at the telescope during the observation. This allows observers to review these data in ‘real-time’. It also acts as a ‘preview’ for the observer and allows adjustments to be made during the run. After the run, the raw data are copied to the archive and this can be used for reduction later. This can happen the following day, or much later, when the observer has returned from the observatory. Any of these data in the data archive can be ‘re-reduced’ at any time since all the raw data are stored.

The current data reduction process for ULTRACAM is designed to produce three colour light curves from the raw image data. The pipeline consists of the following stages:

1. Producing *bias* frames that are used to calibrate the CCD detector's thermal noise characteristics.
2. Producing *flat-fields* to calibrate the pixel sensitivity of each of the 3 CCD detectors.
3. Defining *apertures* for the objects of interest in the run. This step involves manually choosing the objects of interest in the frames and defining the aperture sizes and positions for each object. Apertures are set

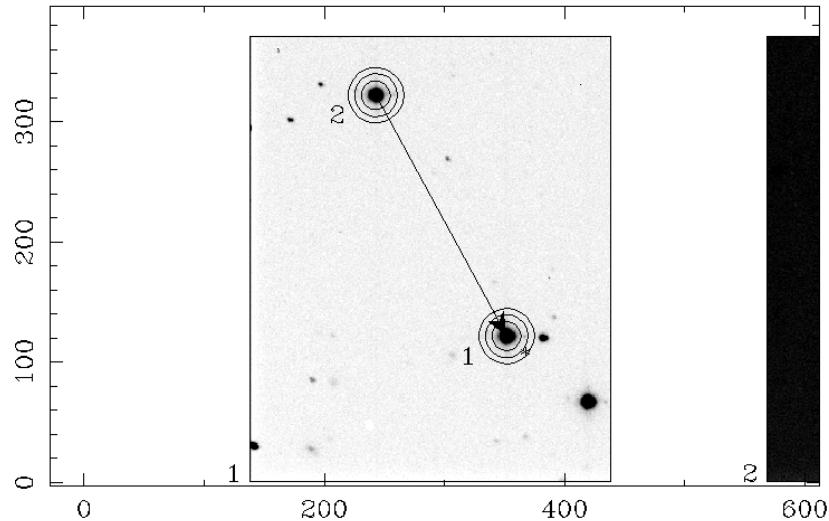


Figure 2.1: Defining the apertures for the reduction using the *traditional* pipeline. Note that the two apertures can be linked. This instructs the pipeline to maintain the pixel separation of the apertures even if there is a small amount of movement from frame to frame. This is useful if our target object is likely to alter in brightness during the run.

independently for each channel (r, g, b). An example of this can be seen in figure 2.1.

4. Running the *reduction* software. The reduction code uses the apertures defined in the previous step and measures the flux of each object in each colour. The software is able to track changes in the object's size and shape due to changes in the point spread function (PSF) by scaling the virtual aperture. It is also able to track small changes in the positions of the objects.

Although this process is not particularly cumbersome, it includes some manual steps and it does not scale well when there are a large number of runs to be processed or if there are many target objects in a run. For example, the run shown in figure 1.7 contains more than 1000 objects. Manually defining apertures for each of these objects in each channel is not practical. An automated method enables data reduction for all of the objects captured in each run without the need for manual intervention.

2.3 Automating the pipeline

The outcome of this MSc project is a system that is able to process the raw image data from ULTRACAM and, without any manual intervention, produce a set of light curves for all of the objects in the data. It produces a set of web pages that can be viewed from anywhere with an internet connection.

2.3.1 Motivation for automating the pipeline

ULTRACAM has recorded a large amount of photometric data at some of the world's best sites and largest telescopes. Since the CCD captures more objects than the observer is strictly interested in, there is a good chance that some objects of astronomical interest have been captured by ULTRACAM, but not been noticed or analysed during the reduction process. We could have some highly valuable and scientifically useful data hidden in the archive. By automating the reduction pipeline we can produce photometry for all of the objects captured by ULTRACAM and then search through these light-curves to find new objects of interest.

By making these data accessible through a standard web-browser, the automated pipeline will allow collaborators who are not physically at the observatory to browse these data as it is being gathered. Although not strictly 'real-time', the automated pipeline can be invoked immediately after the run is completed. This enables a closer collaboration with on-site and off-site teams during an observing run as the off-site team can review the reduced data very easily.

2.3.2 Algorithm of the automated pipeline

The following section describes the key steps in the automated pipeline at a high level. The subsequent sections will describe each in more detail:

Key stages for automation

The stages of the reduction process are as follows:

1. Stage 1: Extract all of the 'detectable' objects in all of the frames.

- (a) Read the raw image file, containing all frames for a particular run.
 - (b) Initialise an empty list of objects
 - (c) For each frame in the run
 - i. For each colour channel in the frame.
 - ii. Extract each window from the frame.
 - iii. Send the window bitmap data to the SExtractor software.
 - iv. Use SExtractor to process these data and produce a catalog of all sources, with their positions and flux measurements.
 - v. Read the results of the source extraction process, including pixel position and flux measurements for each object.
 - vi. For each object returned:
 - A. Try to match this object with one already in the list, based on nearest distance.
 - B. If the object is not already in a list, add this object to the list as a *new* object.
 - (d) Store the list of objects for each of the three channels.
2. Stage 2: Filter this list, removing objects that are likely to be artifacts. This is done by looking for objects that do not persist across more than a pre-defined percentage of frames; and objects that have a size equal to one pixel.
 3. Stage 3: Produce catalogs of the objects ordered by ‘brightness’ as measured by the average flux. Pass these catalogs to the *Astrometry.net* library to resolve the WCS solution for the fields. Perform this task separately for each of the three channels (r, g, b). Since each channel has a very slightly different view of the field and different distortions in the image, their respective WCS solutions will differ by a small amount.
 4. Stage 4: Using the WCS solutions, merge the three catalogs to ‘cross-identify’ each object across the three channels. This may seem to be a trivial step for many ULTRACAM runs because the differences in the fields from channel to channel are minor, however in crowded fields such

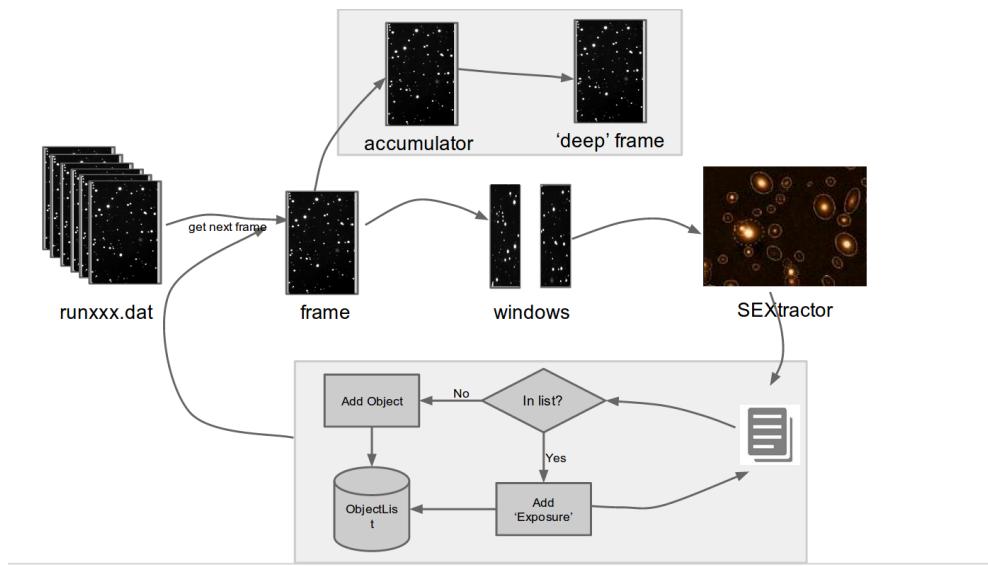


Figure 2.2: Schematic of Stage 1 of the pipeline.

as the one shown in figure 1.7, simply matching objects based on their pixel coordinates is not enough to disambiguate them.

5. Stage 5: Produce deep images for each channel and export to a web-viewable format, such as the Portable Network Graphics (PNG) format.
6. Stage 6: Create Javascript Object Notation (JSON) files and HTML to enable the reduced data to be loaded into a web-browser.
7. Stage 7: Publish this ‘web-enabled’ version of the information to a web site that is accessible outside of the university network.

Source extraction

A popular software tool used for source extraction is *SExtractor*, (Bertin and Arnouts, 2006). *SExtractor* is able to process a 2-dimensional image and produce a catalog of sources in that image, along with a measurement of the flux of each object.

For each frame in the data run (which could consist of a few frames up to several thousand frames), we extract the image data, which consists of a 2D pixel map with the CCD counts (or ADU) for each pixel, and pass this to *SExtractor* for source extraction.

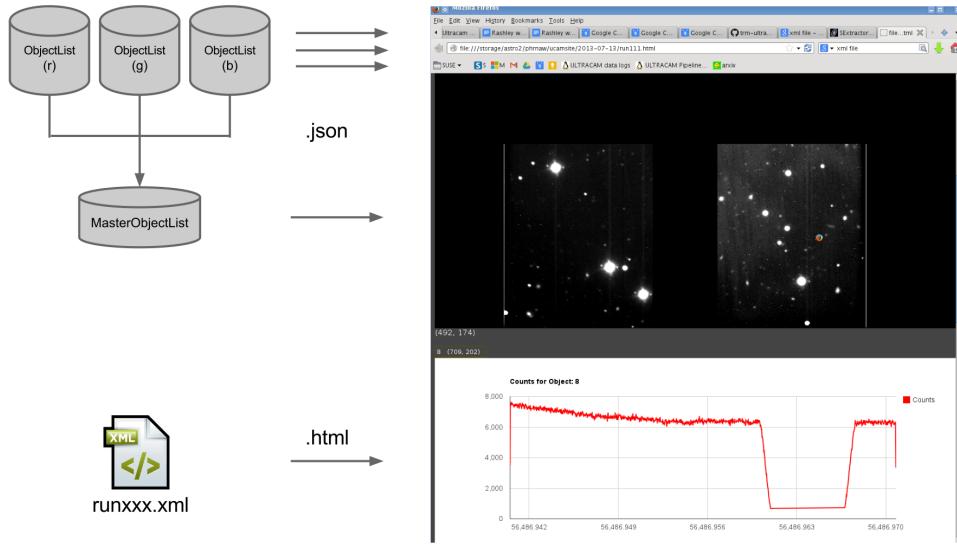


Figure 2.3: Schematic of Stage 6 and 7 of the pipeline.

SExtractor performs two passes on this image. On the first pass it makes an estimate of the ‘background signal’ of the entire image. We assume that the ‘background-signal’ contains both the sky-background and the bias of the CCDs as we have not subtracted the bias from each frame before passing it to SExtractor. It estimates the background by creating a mesh-grid of background readings for the whole image, applying a median clipping algorithm and then fitting a bicubic spline to interpolate between the mesh points. It then subtracts this background from the image. The size of the mesh grid is configurable and the pipeline allows the user to tweak this parameter if necessary. In the current version of the pipeline, this parameter is not configured automatically. If small scale changes in the background are expected then the mesh size should be reduced. The usual value that is used in the pipeline is a mesh grid size of 64 pixels. We have not noticed any reductions where this choice of mesh size has caused any obvious problems.

On the second pass, SExtractor applies a convolution filter to the image. This step is intended to increase the detectability (enhance the signal-to-noise ratio) for target objects on the frame. The default filter used in the pipeline is a simple ‘circular’ PSF defined by the 3x3 mask¹:

¹This filter is normalised before being applied to the image.

```
1 2 1  
2 4 2  
1 2 1
```

SExtractor then applies thresholding to the background subtracted and filtered image. The threshold for detection is defined as the pixel's ADU value above the background (in units of the background's standard deviation). This threshold is configurable and can be modified before running the automated pipeline. The default value we have used for most of the pipeline processing so far is 3σ . Decreasing this parameter will have the effect of increasing the number of objects detected for any particular run but reducing it too much will cause the source extraction to produce too many spurious object detections. For example, setting this value to 1σ results in the source extraction identifying spurious sources that are really just noise in the data background. This leads to the automated pipeline being overloaded with too many new sources to process, causing it to grind to a halt as the number of tracked objects climbs rapidly. At the moment, the automated pipeline is not able to automatically tune this parameter for each run, although this is something that will be considered for future iterations of the software. Figure 2.4 shows how modifying this threshold parameter can increase the number of objects detected by the pipeline.

The image in figure 2.4 has been created by stacking all of the images in the sequence, consisting of 726 frames. Stacking reveals objects that are not visible above the noise when viewing each individual frame. Since the pipeline extracts sources for each frame in turn, it is not able to detect these fainter objects that are so clearly revealed on the deep, stacked images. This suggests that, for future iterations of the pipeline, a different approach might be taken for source extraction and aperture definition. One idea is to perform a first pass to produce a stacked image and then use this image for source detection and aperture creation. These pre-defined apertures would then be used as an input for the flux measurement of each frame. This approach would need some flexibility in order to be able to deal with small movements of the image from frame to frame and to enable aperture tracking of objects that move through the field (such as asteroids).

SExtractor then determines the pixel position of each object by calcu-

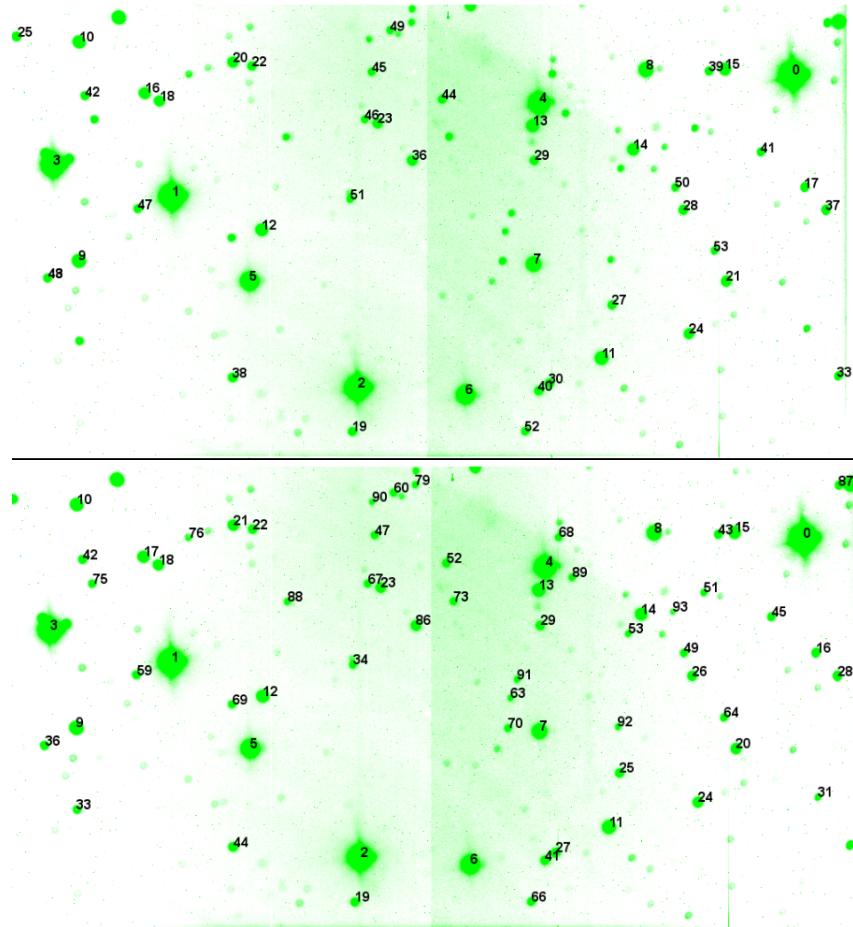


Figure 2.4: The effect of tweaking the DETECT_THRESH parameter in SExtractor. The upper plot shows the objects detected in a run with the threshold set at $3 \times \sigma_{background}$ [56 objects detected], while the lower plot was produced with DETECT_THRESH set at $1.5 \times \sigma_{background}$ [101 objects detected]. Note: setting it to 1.2 detected 149 objects, but slowed the processing of the entire run by a factor of 3x.

lating a weighted mean in the x and y dimensions. The weighting applied to each pixel is the pixel intensity (after background subtraction and filtering). If I_i is the pixel intensity for pixel i in the segmented collection of pixels, S then:

$$X = \frac{\sum I_i x_i}{\sum I_i}, \text{ and}$$

$$Y = \frac{\sum I_i y_i}{\sum I_i}$$

Flux measurement

For each object that has been selected through the threshold criterion, the flux is then calculated. This is computed by one of 2 methods.

1. *AUTO: Automatic aperture mode* SExtractor uses a routine inspired by Kron's 'first moment' algorithm, (Kron, 1980). It defines an elliptical aperture that has an elongation ϵ and position angle θ defined by the second order moments of the object's light distribution. Across this ellipse, a first moment is computed $r_1 = \frac{\sum r I_r}{\sum I_r}$. The aperture is then scaled by user defined parameters to 2.5-3.5 times r_1 . This scale factor is configurable by the user before the run, but does not require modification for this project.
2. *APER: Fixed aperture mode* In this mode, the measurement of flux for the object is calculated by summing pixel intensity, $I_{x,y}$, of all of the pixels that are within a pre-defined aperture radius centred on the x,y position as calculated above. SExtractor does not perform a Moffat fit to the PSF.

$$FLUX = \sum_{x,y \in \text{aperture}} I_{x,y}$$

When using the fixed aperture mode, the aperture radius is defined before the pipeline is run and is specified manually. Due to the diversity of data in the ULTRACAM archive, there is no obvious single choice for this value. For example, some of the runs containing bright sources (ie magnitude 12 and brighter) that are follow-up observations of sources found in the SuperWASP², (Pollacco et al., 2006) and HAT³, (Bakos

²<http://www.superwasp.org/>

³<http://hatnet.org/>

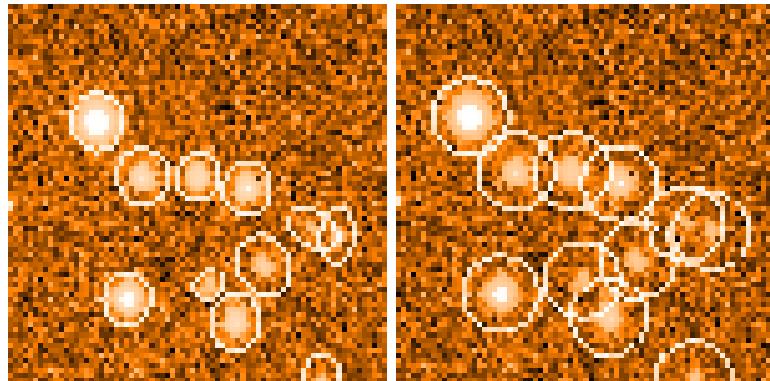


Figure 2.5: AUTO vs FIXED apertures in SExtractor. The image on the left is the result of using the AUTO aperture setting in SExtractor. Note that apertures can be elliptical and are allowed to overlap. On the right is a FIXED aperture setting with the aperture diameter set (manually) to 15 pixels. The FIXED aperture is always circular.

et al., 2002), surveys use the telescope in a deliberately de-focused state. De-focussing the telescope to this extent transforms the Moffat-like PSF into something that resembles a flat disc, or even, in extreme cases, a disc with a hole in the centre (which is an image of the secondary mirror). In this case, the fact that SExtractor does not attempt a Moffat fit is an advantage. Aperture sizes for these runs need to be between 50 and 80 pixels. When the telescope is in focus, for ‘normal’ observations and with crowded fields, we need to use far smaller apertures of between 8 and 15 pixels. It is conceivable that the decision on which aperture size could be automated by in the pipeline. A simple way to do this would be to perform a 2-pass approach. First use AUTO aperture mode to allow SExtractor to return a list of aperture sizes, then choose a fixed aperture size that represents the best choice for this run, most likely the median of the aperture sizes returned after the first-pass. This is something that will be considered for future iterations of the pipeline.

In order to facilitate the automatic running of the pipeline across the entire ULTRACAM data archive, the default setting for the flux measurement is the ‘AUTO’ aperture mode. This means that SExtractor uses the algorithm (described above) to determine the most appropriate aperture size for each object. After inspection of the output, runs that would benefit from fixed

apertures can then be ‘re-computed’ with a manual setting for the aperture size.

The output from SExtractor is a FITS file that contains a catalog of all of the detected objects with measurements of their flux. Each individual window of each frame in each channel (r, g, b) are treated separately by SExtractor and there is no tracking of objects. The catalog returned by SExtractor doesn’t maintain a list of object ids that are consistent from frame to frame. Since the source extraction is performed for each frame individually, the ranking of the objects may change and some fainter object may not appear in each returned catalog. The task of building and maintaining a consistent list of the objects is undertaken by the automated pipeline software built for this project.

2.3.3 Object matching

The automated pipeline creates light curves for all of the objects and, in order to do so, it needs to keep track of these objects throughout the run (across frames and channels). The approach is to use the pixel (x, y) coordinates of the objects returned by SExtractor as the key attribute for identifying an object that recurs from frame to frame.

For each window sent to SExtractor, the pipeline performs the following steps:

1. *Read* the catalog file, containing the (x, y) coordinates and fluxes for all objects in the window.
2. *Rank* all of the objects from brightest to faintest.
3. *Transform* the pixel coordinates from the reference frame of the individual window, to an overall reference frame, matching the size of the full CCD.
4. *Match* all objects in the catalog. For each object in the list, check its proximity to any objects that have been identified on previous frames. The distance threshold for a match is configurable and the default is 10 pixels. If there is more than one match, the closest match wins. If there

is no match, then this is a ‘new’ object and it is added it to the list of detected objects.

The algorithm begins with an empty list of objects. In the first frame, this list grows to include most of the objects we expect to detect in the run. However, the number of objects being tracked will slowly increase as the run is being processed. At this stage of the pipeline we do not remove any objects from our list.

This approach will not cope with situations where the telescope may have been moved or disturbed during a run and there is a sudden ‘step-jump’ in the positions of all of the objects on the frame. This is sometimes referred to as a ‘glitch’. If the ‘glitch’ results in a movement that is greater than the distance threshold (default 10 pixels) then the pipeline will claim to have detected many new objects. In order to deal with glitches, we use a simple technique to determine the pixel shift ($\Delta x, \Delta y$) for each window, compared to the previous window.

We create 2 dimensional map centred at zero, with all values set to zero. For each source detected (object) in the current window we calculate the pixel displacement vector to every object in the previous window $(\delta x, \delta y)_i$. We then increment our map by ‘1’ at the corresponding position. When we have done this for all objects detected in the window, the resulting image ‘histogram’ will have a peak at a value of $(\delta x, \delta y)_{max}$ which corresponds to the overall offset of the second window from the first. ie $(\Delta x, \Delta y) = (\delta x, \delta y)_{max}$. We find this peak by first smoothing the image map (with a gaussian blur) and then fitting a quadratic around the maximum value and determining its value when the derivatives are zero in the x and y directions. We then apply this offset to our current window before looking for matches to objects in the previous window.

For each detected object in the window, we store:

- *ID* A unique ID for this object that will persist across all frames.
- For each frame in the run:
 - *Flux* The flux measurement for this object, as determined by SExtractor.

- *Position* The pixel position (x, y) for this object, as determined by SExtractor, adjusted to the reference frame of the CCD. Note we do not include the small offset measured from frame-to-frame as determined by the histogram map procedure described above.
- *Flux radius* As measured by SExtractor, this is defined as the radius of the circle centred on the barycenter that encloses about half of the total flux. For a Gaussian profile, this is equal to $1/2$ the FWHM.

Before the pipeline attempts to match objects across each of the three channels (r, g, b), it performs a ‘clean-up’ of these data. It makes three passes of the object list performing the following filtering:

- *Cosmic ray filtering*. This step filters out any object that appears on only one frame in the run.
- *Low coverage filtering*. This step removes any objects that appear on fewer than a pre-defined percentage of frames. This value is configurable. The default is 20%. This value should be set to ‘0’ if we are looking for any kind of transient object in the run.
- *Single pixel filtering*. This step removes any object that has a Flux radius, as measured by SExtractor, that is less than or equal to 1 pixel.

At this stage, the pipeline has produced three distinct object lists. One for each of the red, green and blue channels. These lists are referred to as ‘catalogs’ as they now contain position and photometric information for each object detected in the run. Now the pipeline attempts to cross-identify objects across the three catalogs. This is done based on the object’s average position (\bar{x}, \bar{y}) in each channel and the minimum distance between it and its corresponding location in the other channel. If there is an astrometric solution for each channel then this is used in favour of the pixel position as the astrometric solutions are likely to be far more accurate than the pixel positions. Unfortunately, for most of our runs, we do not have an astrometric solution and we have to revert to matching based on pixel distance. Astrometric solutions, and our difficulties with finding solutions, are discussed later in this chapter in section 2.3.4.

The position we use is the *mean* pixel position of the object through the duration of the run. So we have three values to match $(\bar{x}, \bar{y})_{red}$, $(\bar{x}, \bar{y})_{green}$ and $(\bar{x}, \bar{y})_{blue}$. Since the red catalog (which is derived from the optical channel that is usually configured to use the Sloan ‘i’ or ‘r’ filter) has the most number of objects detected, it is used to seed the master catalog. In other words, the master catalog is initialised with all of the objects in the red catalog. For each object in the ‘master’ catalog, the pipeline consults the catalogs from the other two channels looking for a nearest match in distance within a pre-defined threshold.

For each object in the green catalog indexed by the letter i , we calculate its distance from each object in the master catalog $D_{i,j}$.

$$D_{i,j} = \sqrt{(\bar{x}_{red,j} - \bar{x}_{green,i})^2 + (\bar{y}_{red,j} - \bar{y}_{green,i})^2}$$

The matched master object for this green object is the one with the closest pixel distance $(D_{i,j})_{min}$. We merge these two objects together in the catalog, which now contains one unique identifier (number) and the red and green photometry. If there is no match within the minimum distance threshold, then the object is treated as a ‘new’ object and added to the master catalog. It is possible that an object can be identified in the green channel, but not in the red. In this case, it is still added to the master catalog as a new object. This means that we have the capability of dealing with objects that have photometry in one or two colours, but not all three.

The process is then repeated for the blue catalog. Again we match the blue coordinates to each object in the master catalog. First trying for a match of the blue coordinates to the red coordinates and then, if no match is found, we repeat looking for a match between the blue coordinates and the green coordinates. Once again, if no match is found, then the object is added to the master catalog as a ‘new’ object.

It is obvious that this is a very crude method of object matching. It is, however, surprisingly robust for the majority of the ULTRACAM runs in the data archive. Many of the runs do not have crowded fields so there is little ambiguity in the object’s positions. It can fail in a few situations. Since the red, green and blue channels do not have identical optical configurations, the images are not exactly aligned geometrically. The images in the r, g, b channels can differ from each other in terms of translation, rotation and

distortion (across the image). This becomes particularly obvious for a full-frame image (using the full area of the CCD) and is most visible towards the edges of the field. An example of this difference in the pixel locations from channel-to-channel can be seen in figures 2.6 and 2.7.

Yet another complication is caused by the time-variation in this non-overlap of the three channels during the course of a particular observing run. When the airmass of the target field undergoes a significant change the image distortion due to the atmosphere varies in each channel. Another factor is that the camera's physical orientation changes and the optical paths will undergo changes due to flexure in the instrument's chassis. The change in the offset position from channel to channel can be as much as 4 pixels as the airmass goes from 1.0 to 1.2 and for even larger airmass variation the object will move by as much as 15 pixels in the blue channel relative to the red and green channels. See figures 2.8 and 2.9 for examples of this drifting effect.

The gradual movement of an object's position is dealt with in the first stage of the pipeline by allowing the object to drift gradually from frame-to-frame. It compensates by constantly updating the object's position in each frame and using the new value as the comparison position when looking for matches in the next frame. It can deal with a general 'slow' migration in the object's position. Therefore, in the first pass of the automated pipeline, when we are building the catalogs for each channel independently of each other, we do not have problems with object matching.

The process can fail when cross-matching across the different channels when mean position of the object is displaced by a large amount from channel to channel, or the field is crowded and there is more than one match for any particular set of objects. Fortunately, for many of the crowded fields, we can find an astrometric solution. In this case, we are no longer relying on pixel coordinates but world coordinates to perform the match across the channels and these image distortions have been taken into account.

Once the object cross-matching stage is complete, the pipeline writes the new 'three-colour' master catalog to a folder on the web server, ready to be loaded in the web browser.

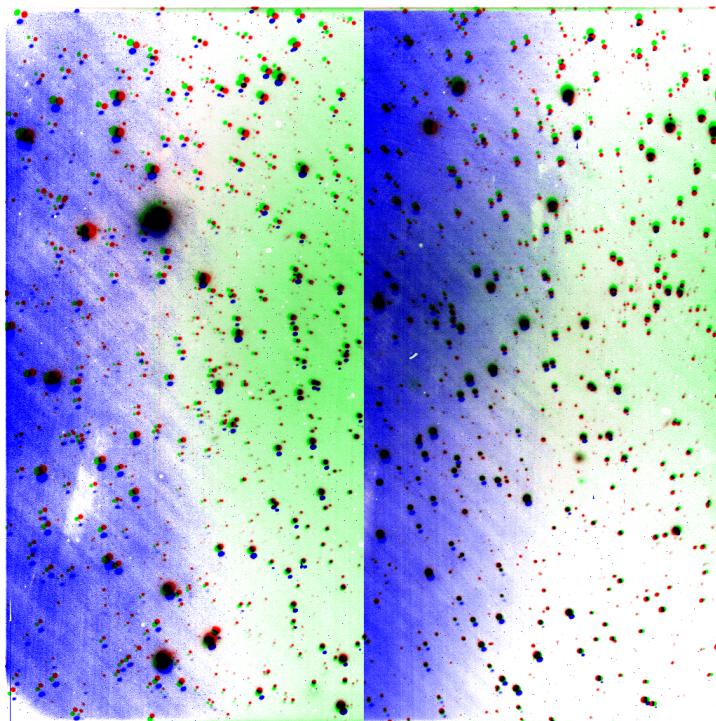


Figure 2.6: Images of the three channels overlaid (without any distortion correction applied). It is clear that the three channels do not have identical images. Translation, rotation and differential distortion are all visible. This makes matching of objects across the three channels difficult in runs with crowded fields.

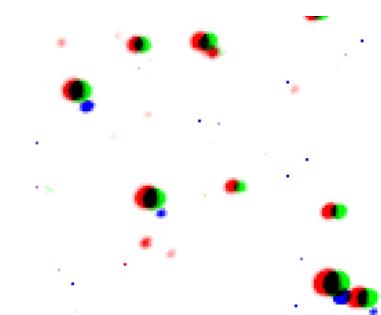


Figure 2.7: A close up of figure 2.6 showing the bottom right hand corner. Note that the blue image is significantly translated with respect to the red and green image. To make matters worse, this distortion is not constant throughout the duration of a single run and changes as the airmass of the target field changes.

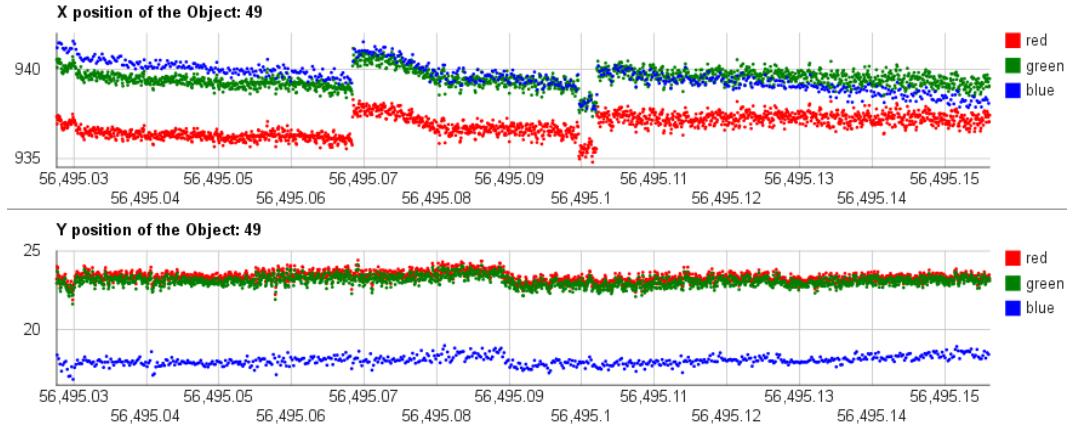


Figure 2.8: Plot of the (x, y) position of the object in the lower left corner of figure 2.7 showing how the position varies over the course of the run - 3 hours. During this time the airmass, sec z , of the target field varies from 1.02 to 1.21. Note how the x position of the object in the blue channel drifts with respect to the x position of the object in the red and green channels. The step changes in the object's position are caused by the observer making manual adjustments to the guiding at the telescope.

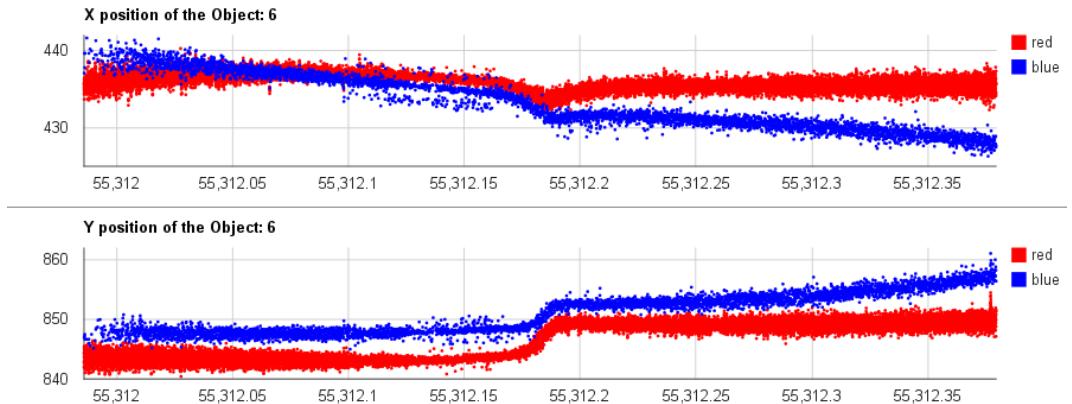


Figure 2.9: Another plot showing the change in relative positions of a single object in different channels over the course of a long run. These data is taken from the longest run in the ULTRACAM data archive, the 9.5 hour long run *2010-04-25/run020*. During this run, the airmass varies from a minimum of 1.0 (corresponding with the center of the plot), to a maximum of 1.99 (at the extreme left and right ends of the plot). The change in offset from the red to the blue channels is most noticeable on the x position, where the blue channel's offset moves by about 13 pixels.

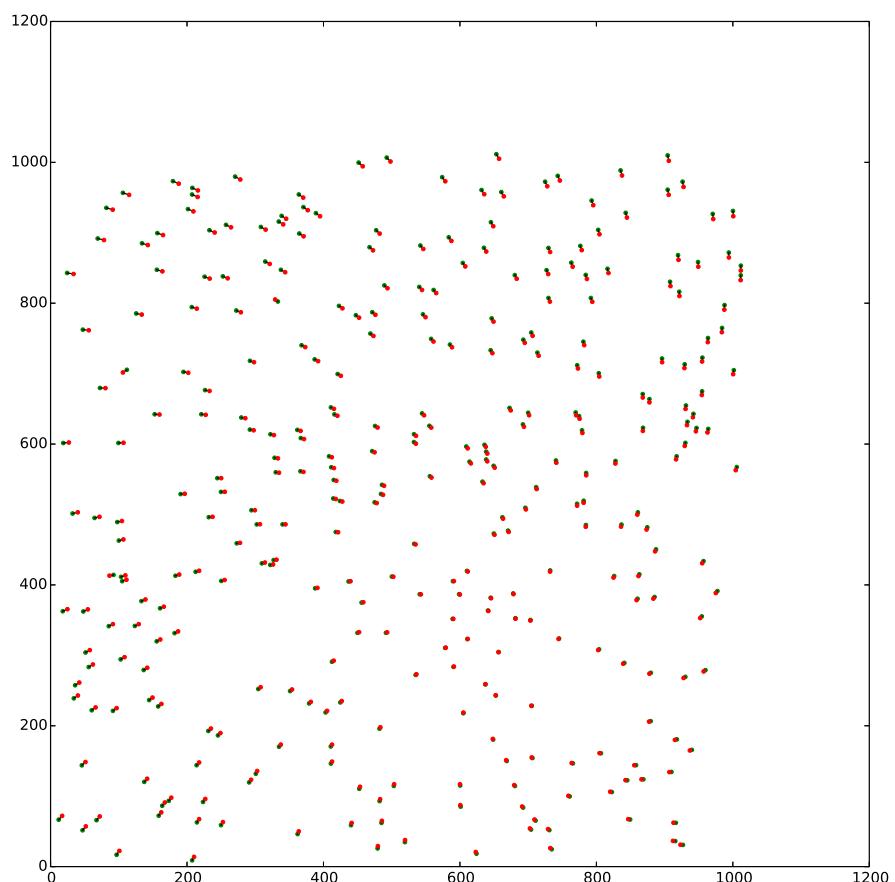


Figure 2.10: The difference in object mean positions between the ‘red’ channel and the ‘green’ channel for run: *2013-07-21/run010*

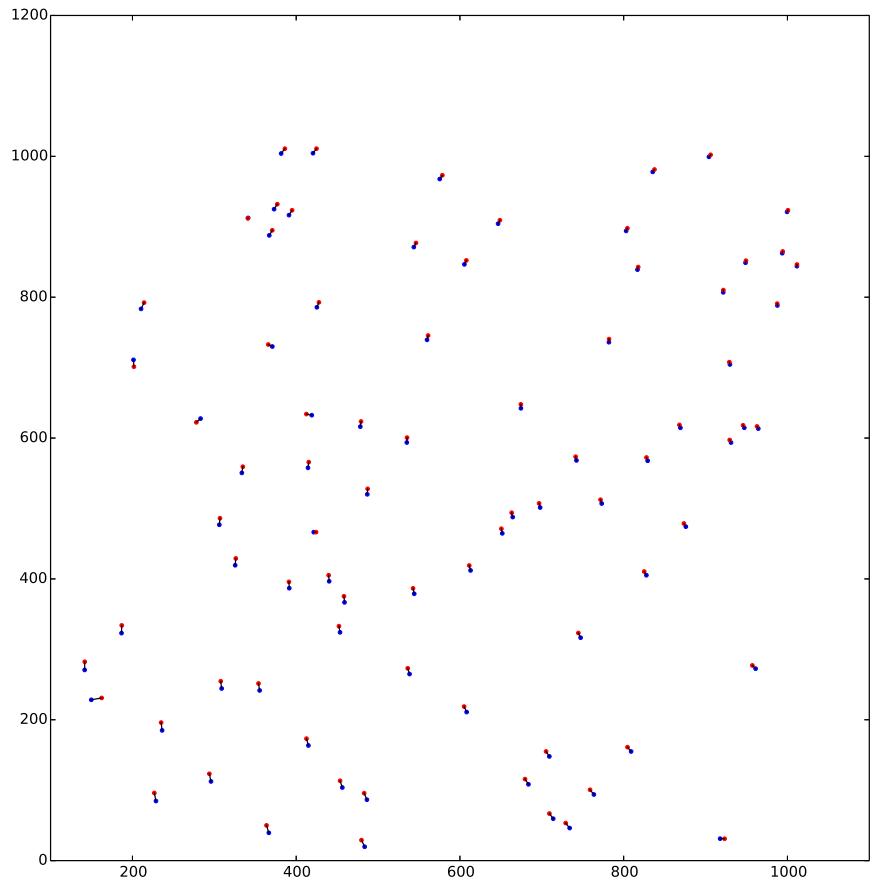


Figure 2.11: The difference in object mean positions between the ‘red’ channel and the ‘blue’ channel for run: *2013-07-21/run010*

2.3.4 WCS solutions

After tests using SCAMP (Bertin, 2006) and Astrometry.net (Hogg and Lang, 2012) it was clear that the Astrometry.net software was more reliable at finding good WCS solutions to the fields. The software was downloaded to a local machine (including the extensive index files) and compiled. Despite being the solution that yields the most positive results so far, it still does not consistently find WCS solutions for all of the fields.

There are several challenges to finding a WCS solution for the fields. These are:

- *Lack of telescope pointing information:* ULTRACAM does not integrate with the pointing software of any of the telescopes and does not get pointing information automatically. We rely on the observer entering a name of the candidate object for each run and then, when the data are archived, a *SIMBAD* lookup is used to turn the object identifier into a right ascension and declination. This gives us a world coordinate that is somewhere in the field, but it is not known which object (or pixel location) this applies to.
- *Field rotation:* Since ULTRACAM can be rotated about the optical axis to allow for optimal alignment of the objects, the field of view can be at any arbitrary angle of rotation, giving an extra degree of freedom when attempting to match the field to a known catalog.
- *Windows:* Many ULTRACAM runs are configured to use only portions of the CCD area. An example of this is shown in figure 1.9. This means that there is an incomplete view of the sky for that field. When trying to match to existing indexes, there could be important, bright objects that are in the index file, but do not appear in the ULTRACAM field due to masking by the window choice.
- *Sparse fields:* On uncrowded fields, we might have fewer than 4-5 objects to be used for field identification.
- *Very small windows:* Some runs, particularly ones in high cadence mode, use very small windows (eg 172x156 pixels) in order to decrease readout

time. This means that our images (and input catalogs) might only contain two objects. This makes matching to a reference catalog impossible.

- *Choice of reference index by colour:* The Astrometry.net software uses USNO-B and Tycho-2 reference catalogs by default. These are based on infra-red and V magnitudes. This means that the blue channel (which is often using the SDSS u filter) might not match the reference indexes. Indeed, current tests often result in a match in red, a match in green but no match in blue.

After the first stage of the automated pipeline we have three catalogs of objects for each of the channels (red, green and blue). These catalogs contain pixel coordinates and flux measurements for each frame in the run that the object has been identified. We produce a simplified catalog based on the mean pixel positions and mean flux for each object ($\bar{x}_i, \bar{y}_i, \bar{F}_i$). This catalog is then sorted in order of decreasing mean flux, F . The Astrometry.net package is given this input catalog and asked to find an astrometric solution for the field.

Astrometry.net compares objects in its reference catalog to the catalog and pixel positions in the input. The matching algorithm is based on comparing the relative positions of quadruples of stars. The indexes are ‘pre-built’ and derived from the USNO-B survey, which contains $\sim 10^9$ stars and Tycho-2 which has $\sim 2.5 \times 10^6$ stars.

In addition to providing Astrometry.net a catalog of objects to match, we also give it the known location of the field that has been provided via a SIMBAD lookup of the coordinates of the target object as specified by the observer at the telescope. This gives us the world coordinates that are guaranteed⁴ to be somewhere within our field. We provide a limit to the coordinates of the solution as a maximum distance of 1 degree from our input location. We also provide upper and lower limits to the expected field scale of the solution. Providing these parameters saves computation time as it restricts Astrometry.net to a small region of the potential solution space and removes the need of doing a comprehensive search. Considering that our field sizes are only a few arc minutes wide, specifying 1 degree as the search radius is probably

⁴Provided that the telescope operator has correctly entered the target name, and the SIMBAD lookup has been successful.

overkill. A future task for this automated pipeline project will be to find the optimal value for this parameter.

If Astrometry.net can find a solution for our field, it generates a FITS format file containing the parameters defining the solution. These parameters consist of the position in right ascension and declination ($\alpha_{ref}, \delta_{ref}$) of a particular reference pixel in the image (x_{ref}, y_{ref}), plus 4 parameters that define a transformation matrix to move from pixel coordinates (x, y) to world coordinates, (α, δ). These values are labeled $CD1_1$, $CD1_2$, $CD2_1$ and $CD2_2$. The transformation from pixel coordinates to world coordinates is then given by:

$$\begin{pmatrix} \alpha \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha_{ref} \\ \delta_{ref} \end{pmatrix} + \begin{pmatrix} CD1_1 & CD1_2 \\ CD2_1 & CD2_2 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Where (x', y') are the pixel offsets to the reference pixel (x_{ref}, y_{ref}) .

These values are saved to the web repository as a JSON⁵ object, ready to be loaded when the web browser accesses the page.

These 4 values define a scale transformation and a rotation from pixel to world coordinates. They do not express distortion across the image. In order to encapsulate this distortion, Astrometry.net also provides Simple Imaging Polynomial (SIP) correction parameters, (Shupe and Hook, 2008). In this project, we use a SIP polynomial of the 3rd order to account for distortion across the ULTRACAM field. Figures 2.12, 2.13 and 2.14 show the SIP polynomial corrections applied to the positions of each object identified in the run *2013-07-21/run010*. The correcting factors provided by this polynomial are generally very small, providing a corrections of a few thousandths of a pixel in most cases.

At this stage of the pipeline, we have completed the reduction of these data and are ready to render our results in a web-browser.

⁵Javascript Object Notation (JSON) is a plain text file format that encapsulates object structure and is becoming increasingly common as a convenient way to store data to be displayed on web pages. It has some similarities to XML, but with a reduced syntax. It is described in more detail in the following chapter 3.

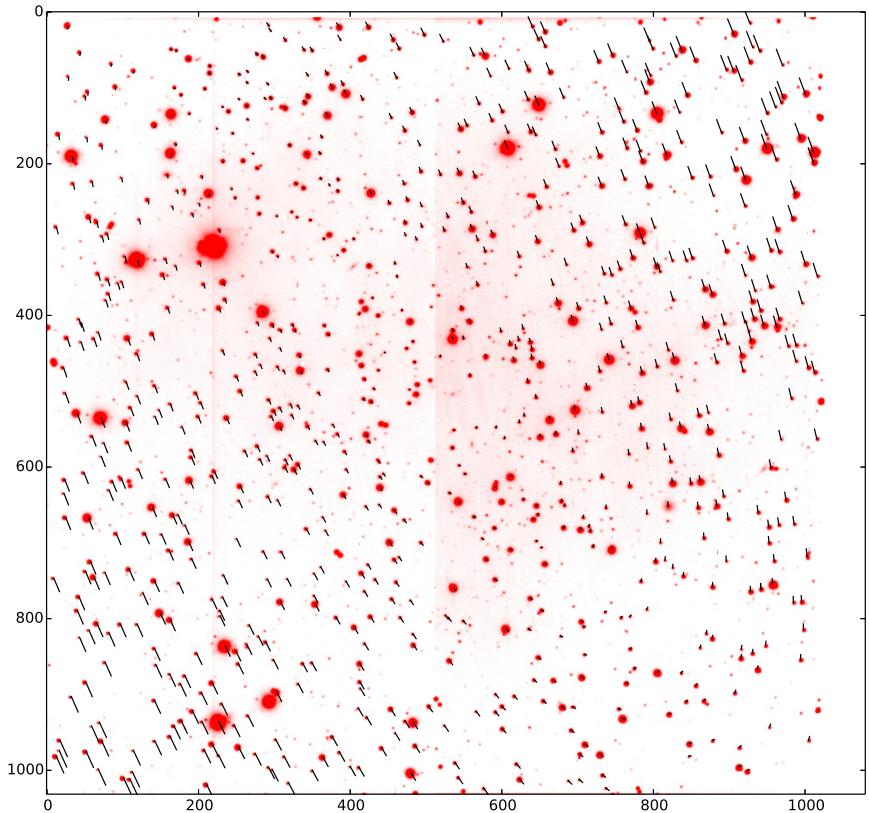


Figure 2.12: An indication of the distortion present in the field for the ‘red’ channel. The background image shows the deep image of the field. The vectors leading away from the objects were generated by calculating the ‘World’ coordinates (WCS) for the object’s position (without the SIP corrections) and the reverting those WCS coordinates back to pixel coordinates (using the SIP corrections). The displacements have been exaggerated by a factor of 10^7 .

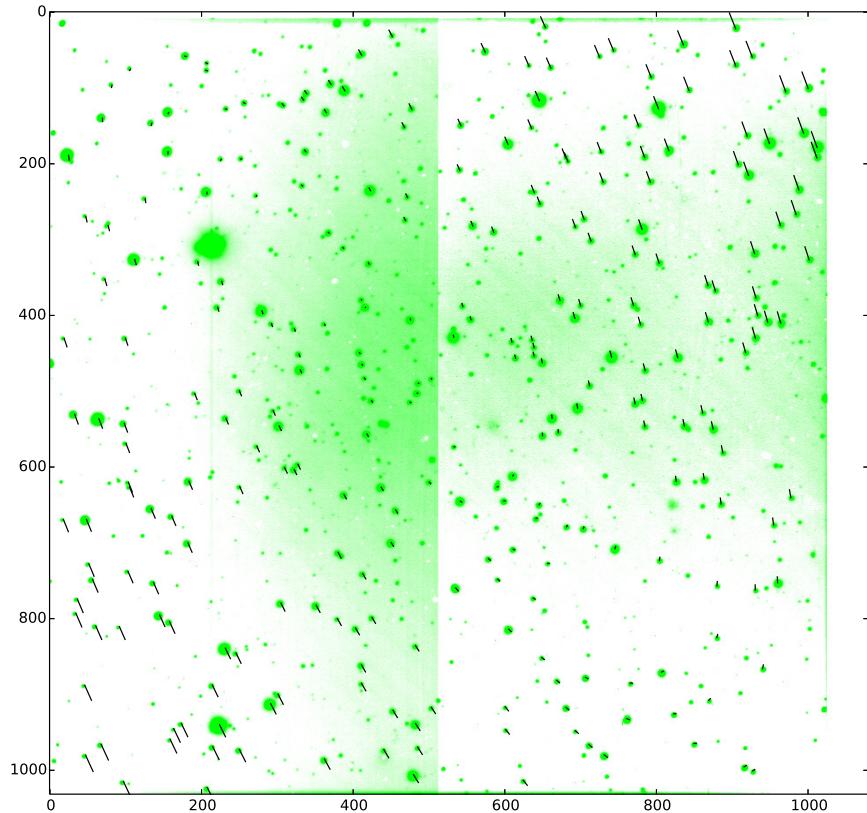


Figure 2.13: An indication of the distortion present in the field for the ‘green’ channel. The background bitmap shows the deep image of the field. The vectors leading away from the objects were generated by calculating the ‘World’ coordinates (WCS) for the object’s position (without the SIP corrections) and the reverting those WCS coordinates back to pixel coordinates (using the SIP corrections). The displacements have been exaggerated by a factor of 10^7 .

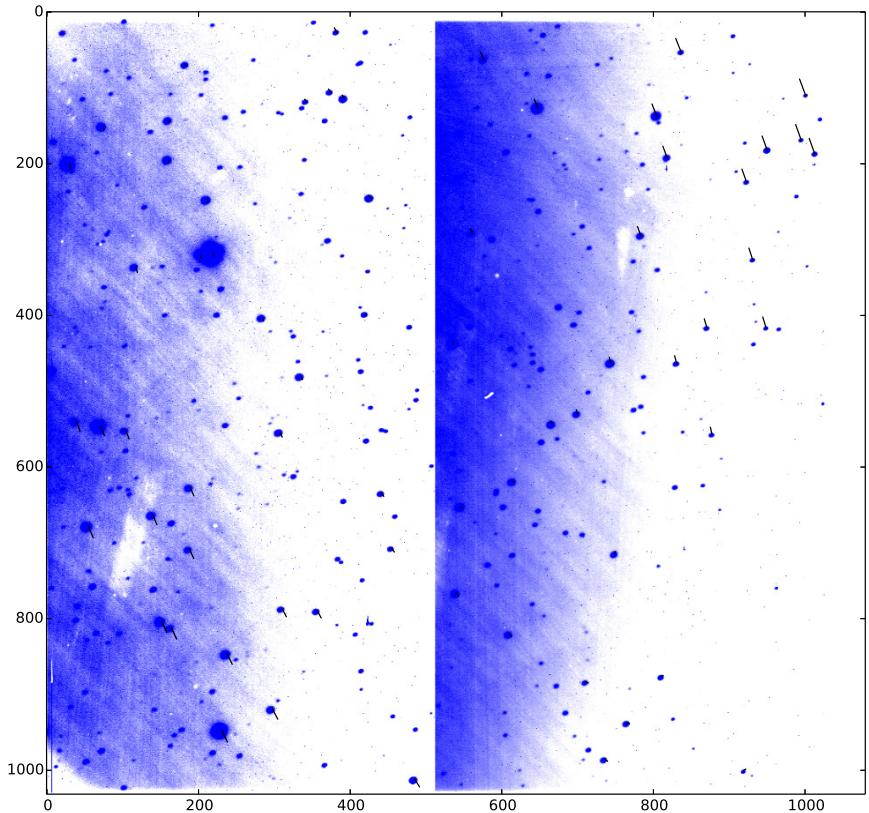


Figure 2.14: An indication of the distortion present in the field for the ‘blue’ channels. The background bitmap shows the deep image of the field. The vectors leading away from the objects were generated by calculating the ‘World’ coordinates (WCS) for the object’s position (without the SIP corrections) and the reverting those WCS coordinates back to pixel coordinates (using the SIP corrections). The displacements have been exaggerated by a factor of 10^7 .

Chapter 3

Creating a browse-able front-end for the data

The goal of this project was not only to automate the processing of the ULTRACAM raw data, but also to provide a mechanism whereby the entire data archive could be made available for easy access to researchers worldwide. The obvious approach is to make the output available through a web enabled interface. From the outset of the project, all efforts have been focused on ensuring that the resulting data is accessible through an easy-to-use web interface.

3.1 Web browsers

In order to enable universal access to the output of the automated pipeline it is important to have a solution that does not require the user to install any additional software on their own computer. We can reasonably safely assume that everyone who has an interest in accessing the ULTRACAM data has a standard web browser installed, a web version of this data archive is the best solution.

A more specific definition of our assumption is that we expect that the user accessing the archive will have a browser that has the capability of rendering HTML5 markup, Javascript and CSS. These technologies are standard in all popular web browsers in mid-2014. The only notable exception is that Internet Explorer, by Microsoft, is not supported in this project. Although Internet Explorer is a modern browser and does have support for the required

technologies, its implementation of these technologies is significantly different to that of all of the other browsers and would have required extra coding for support. We felt that, since the majority of astronomers don't use the Microsoft operating system, we were justified in making this omission.

3.2 Web technologies

HTML, CSS and JavaScript are three core technologies driving the development of the dynamic, interactive and flexible applications we are becoming accustomed to on the web these days. We chose these three technologies to present the ULTRACAM archive. The result of this is that this archive is immediately available to anyone with a modern web browser and working on any type of computer (desktop, laptop or tablet). There are some high demands on memory, so it is not recommended that the archive is browsed using a mobile phone. There is no functionality that restricts the use of the ULTRACAM site on such a device (just the memory constraints).

HTML provides the underlying structure of a modern webpage. It is a semantic markup language, meaning that its purpose is to inform the browser of the document's structure. Despite the habit of many people who dabble at making web pages, HTML is *not* meant to be used to alter the presentation of content. *CSS (or Cascading Style Sheets)* is the language that is design to inform the browser on how the presentation of each element on the page should look. For example, CSS can define the fonts or colours for each particular element (or set of elements), like headings, paragraphs, etc. *Javascript* provides the technology to enable the interactive portion of the page, allowing the user to trigger actions when a mouse is clicked or a new object is loaded. It can be used to manipulate the structure of the existing page. It also provides the mechanism for mathematical computation.

Another way of stating this is to say that HTML provides the semantic structure, CSS the presentation layer and JavaScript the programmatic environment. This structure echoes the classic "Model-View-Controller" approach used in many development paradigms in the field of computer science.

The final stage in the ULTRACAM automated pipeline produces a set of files that are available to a web browser. These files are hosted on a web

server that is operated by the University of Warwick CSC team. The pipeline prepares the files and then writes them to the appropriate location in the university's local storage. As soon as the pipeline has finished running, the web pages can be viewed globally.

Web applications like this, are often referred to as 'client-server' applications, meaning that the application consists of two parts, one running on the client (web browser) and the other running on the server of the institution hosting the application. Obviously, there is a one-to-many relationship between clients and server. There is usually only one server involved, but many clients can connect to that server and interact individually with the application. When writing the web interface for this project we had to make a decision on how much of the functionality we should place on the server versus the client. There were two main competing factors to consider:

- *Complexity of the application:* Writing an application that has complex components on both the server-side and the client-side, increases the difficulty of writing and maintaining the application. We need to install and configure a web server that is able to run code locally and that also needs access to local data sources, such as databases. If we structure our application such that all of the complexity is on the client then the web server only needs to host and serve static files. This makes the management of the server-side portion trivial. If, on the other hand, we decide to split the application code to run on both the client and the server, then we need to write code for both components. The connection between client and server can add some latency (time-lag) to the interactions. This would be noticeable if, say, every time the user clicks on a new object in the field, we need to make a request to the server to fetch a new batch of data to render.
- *Browser memory constraints:* Loading all of the data required to display the results of one of the ULTRACAM runs can be quite demanding on the browser. For some runs there are several hundred objects each with several hundred exposures. This can result in a JSON file for the object data that is >300 MByte in size. All of this has to be loaded into the browser's memory. If the user is working on a tablet or an older

desktop PC or laptop, then this can cause memory issues. Some long runs with extremely high cadences have very few objects, but hundreds of thousands of exposures and these will tax the memory management of the browser. That said, it is true that for the vast majority of the runs, the memory load on the browser, although significant, is not a problem.

In order to aid rapid development of this project, we decided to opt for a purely client-side code implementation, leaving the web server to serve only static files. By making this choice, we are placing the burden on the web browser to perform the rendering, temporary storage and manipulation of the data for the run. This is working adequately in terms of meeting the needs and scope of the project, but it is clear that, for future iterations of this pipeline, we should carefully consider moving to an application model that relies more heavily on the server to manipulate, store and serve data. We cannot place any more load on the client.

As our data storage tool, we chose JSON (JavaScript Object Notation)¹ as the format as this meant that it could be easily loaded by the Javascript code running on the browser. JavaScript has several built-in methods to load and parse a JSON object. JSON is a flexible, open format that allows a hierarchical structure to be defined for each object stored. It is also designed to be human-readable, meaning that it is possible to open in a text editor and check the contents. The problem with this format is that it is stored as 'plain text' and uncompressed. The text itself defines the structure of each object it contains, leading to some amount of redundancy in the file (eg the repeating of labels, etc). While it is true that JSON is inefficient in many ways, it is a useful format to use thanks to its flexibility and the ease with which the developer can check and debug the data.

Many client-server applications use a relational database to store their data, often something like *MySQL*. Since we were not writing any code to run on the server-side and purely relying on the web server for static files, this did not seem appropriate. It is a topic that will be re-considered when we look at implementing server-side code in future iterations of this automated pipeline.

¹<http://json.org/>

3.3 The Web site

The core ‘visible product’ of the project is a website that allows a user to browse all of the data in the ULTRACAM archive. The key features of this website are:

- A catalog of runs organised by calendar date, containing *thumbnail images* of the fields.
- For each run, a web page that shows the user:
 - *deep images* of the field in each of the three channels (r, g, b).
 - *light-curves* of each object as the user clicks on the object with the mouse.
 - plots of the *pixel position* of each object over the course of the run.
 - *world coordinates* of each object, provided that a correct astrometric solution has been found for the run.
 - a *light-curve* for the object that is currently being used as the ‘comparison’ object.
- Light-curves can be plotted as absolute measured flux or a *relative flux* compared to the comparison object in the field.
- The web page allows the user to *export* the data in a CSV format.

See Fig. 3.1 for an example of the web-page.

3.4 Accessing the data

The pipeline deposits the output HTML, Javascript, image (PNG) and data files (JSON) to a folder that is configured to be served by the University of Warwick’s CSC web server at <http://deneb.astro.warwick.ac.uk>. Please refer to the User manual to find out how to access and browse the data A.

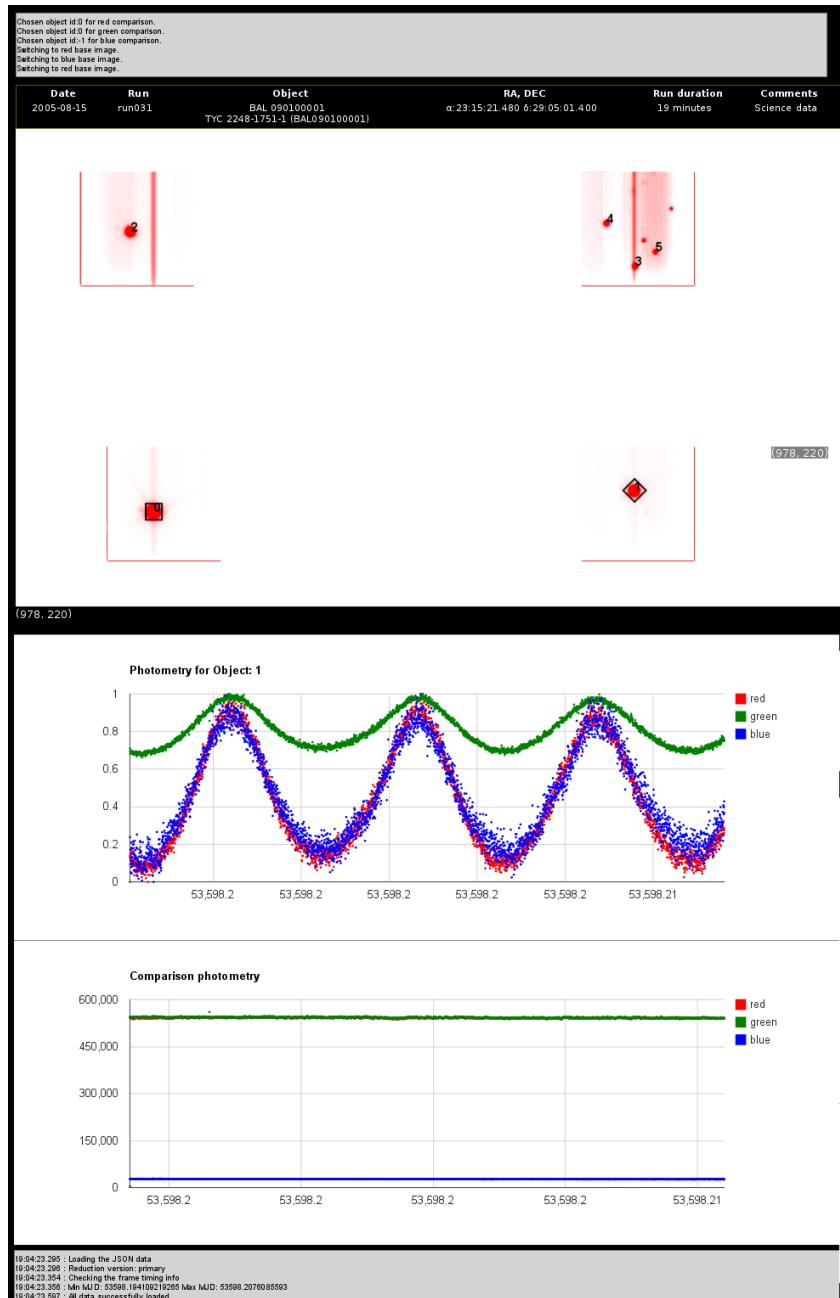


Figure 3.1: Example of the webpage for browsing the light-curves of a particular run. The object shown is the pulsating sub-dwarf known as TYC 2248-1751-1 or Balloon 090100001.

Chapter 4

Results from the automated pipeline

4.1 Introduction

Once the first version of the automated pipeline was built, we started using it to reduce the ULTRACAM archive. We evaluated some key aspects of the automated pipeline. The first aspect to investigate was whether the photometry the pipeline produces is of sufficient quality to allow researchers to view light-curves that clearly demonstrate the astronomical events occurring in the data. We looked at some well known objects and compared the photometry produced by the traditional pipeline with this new automated pipeline. We also wanted to evaluate the usability of the web interface as a method of discovering new variable objects. We visually inspected the output of the automated pipeline looking for variability in the light-curves. We also expected to be able to easily identify the intended target of each run even if the field contained many more objects since it should give itself away by its variability. Finally, we attempted to use the pipeline to process the entire ULTRACAM archive to verify if it was robust enough to reduce the full set of the ULTRACAM data despite the diversity of the input.

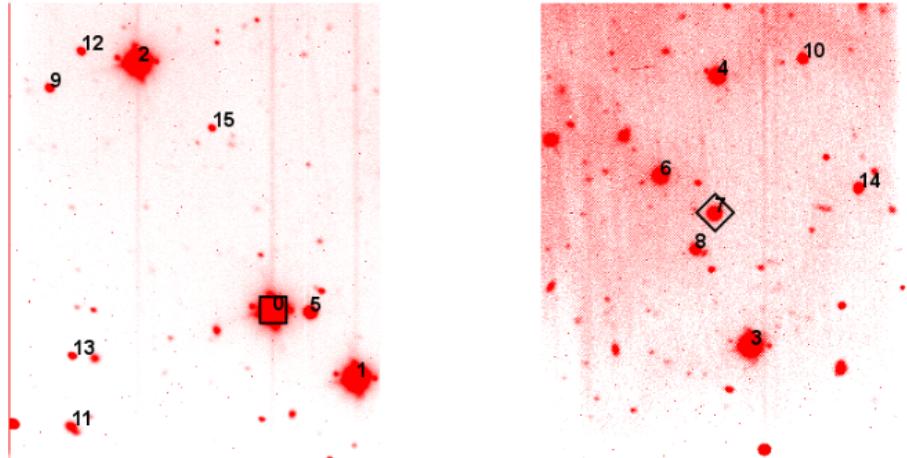


Figure 4.1: Snapshot taken from the automated pipeline browser for 2013-07-13/run111. The target, NN Ser is labeled object ‘7’ and the object we have used as the comparison is labeled ‘0’ in the image. This is a stacked image from the ‘red’ CCD with the Sloan ‘i’ filter.

4.2 Quality of the photometry

The main purpose of this MSc project was to establish a process for automatically reducing the light-curves for all objects in the data archive rather than trying to produce accurate and well-calibrated measurements. The diverse nature of the dataset means that it is not trivial to write an automated algorithm that can perform fully calibrated measurements. The automated pipeline lacks the ability to correctly identify the appropriate bias readings, flat-fields and standard stars that should be used for photometric calibration. Therefore, this step is skipped altogether. The magnitudes and flux counts produced by the automated pipeline are not calibrated and will differ from their true values by a certain offset.

Since the ULTRACAM already has a well-established data reduction pipeline, it is useful to compare the results of this pipeline with the new, automated one built in this project. As mentioned above, the automated pipeline does not perform calibrated photometry, but we can still compare the non-calibrated photometry to get an estimate of how well our new pipeline performs.

In order to do this, we chose a run of a target object that has often been observed with the ULTRACAM. The object is NN Ser, a white-

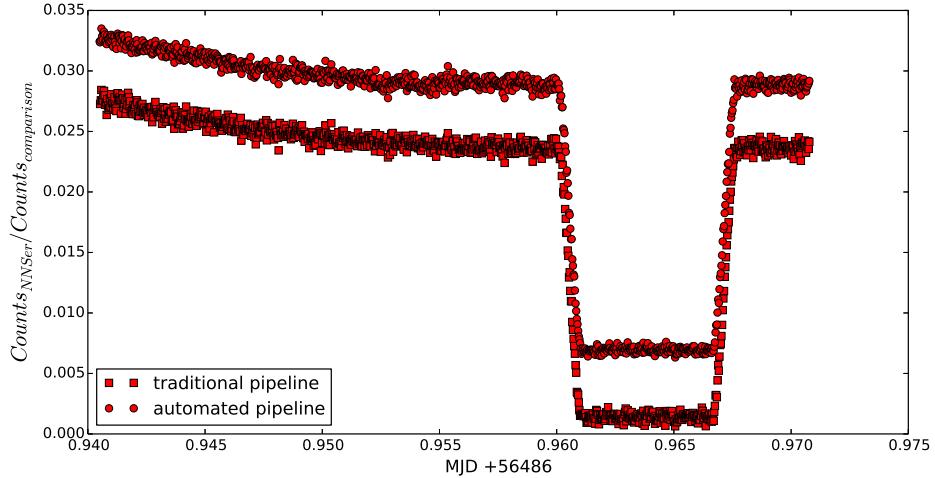


Figure 4.2: Comparison of the light-curves for NN Ser in the Sloan i filter. Square data points were generated by the traditional pipeline and circles by the automated pipeline. The vertical offset applied to the circles is 0.0054.

dwarf, M-dwarf eclipsing binary. The specific run in question is `2013-07-13/run111`. Producing the photometry using the automated pipeline on this run is achieved by simply typing: `runbuilder.py 2013-07-13/run111` on the command line. Please refer to the user manual in appendix A for instructions on how to install and run the pipeline. The reduction takes about 5 minutes to process when running on a standard desktop machine in the University of Warwick Astronomy department. The output of this reduction can be seen at <http://deneb.astro.warwick.ac.uk/phrnaw/sitedev/2013-07-13/run111.html>. We also reduced the same run with the traditional pipeline. In both cases we produced light-curves by plotting the relative flux of the target relative to the comparison. In figures 4.2 to 4.7 we have plotted the results of the automated pipeline together with the traditional pipeline (with a small offset applied to separate the data points). Inspections of these plots shows that the pipelines produce consistent results. The pipelines seem to have similar RMS scatter and show the same trends. Closer inspection reveals that outlying data points usually occur concurrently demonstrating that the systematic differences between the two approaches are smaller than the intrinsic errors in the measurements.

Since the automated pipeline relies on the third party software, SEx-

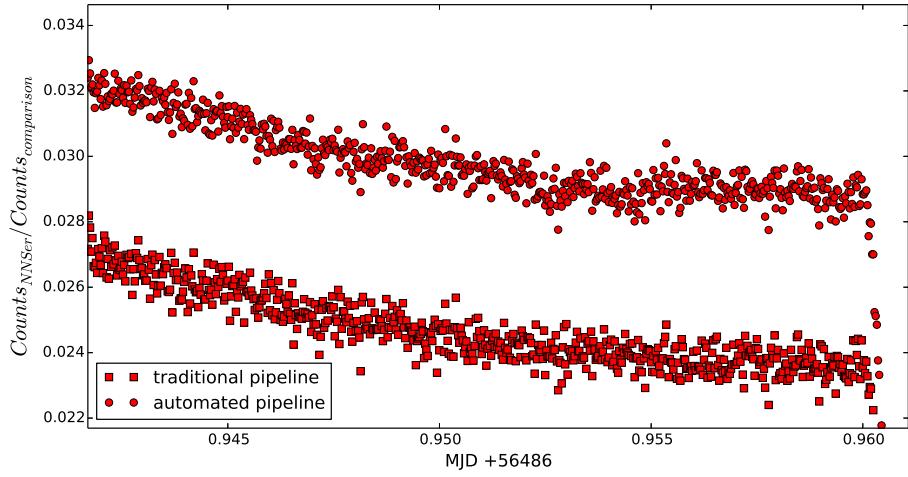


Figure 4.3: A closer look at the comparison of the light-curves for NN Ser in the Sloan i filter, from the start of the run to the beginning of the eclipse ingress. Square data points were generated by the traditional pipeline and circles by the automated pipeline. The vertical offset applied to the circles is 0.0054.

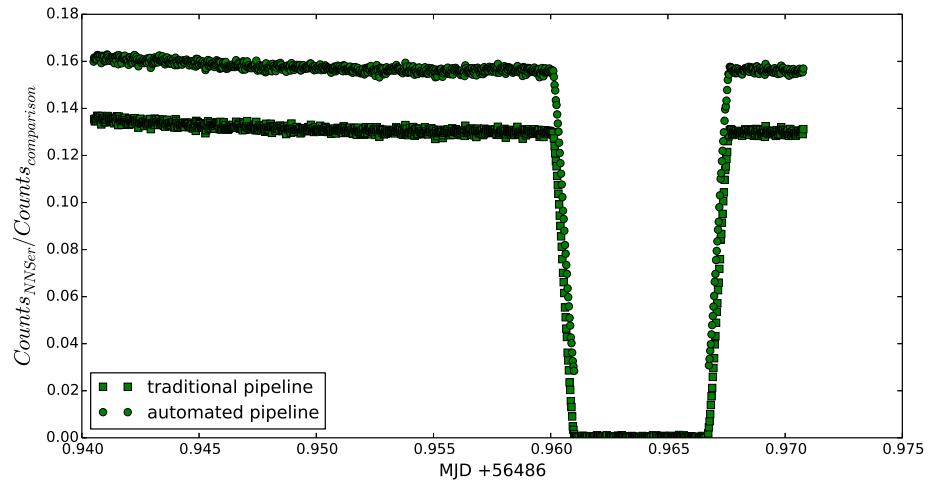


Figure 4.4: Comparison of the light-curves for NN Ser in the Sloan g filter. Square data points were generated by the traditional pipeline and circles by the automated pipeline. The vertical offset applied to the circles is 0.027. Note that the automated pipeline has no data for the duration of eclipse totality. We discuss the reason for this in the text.

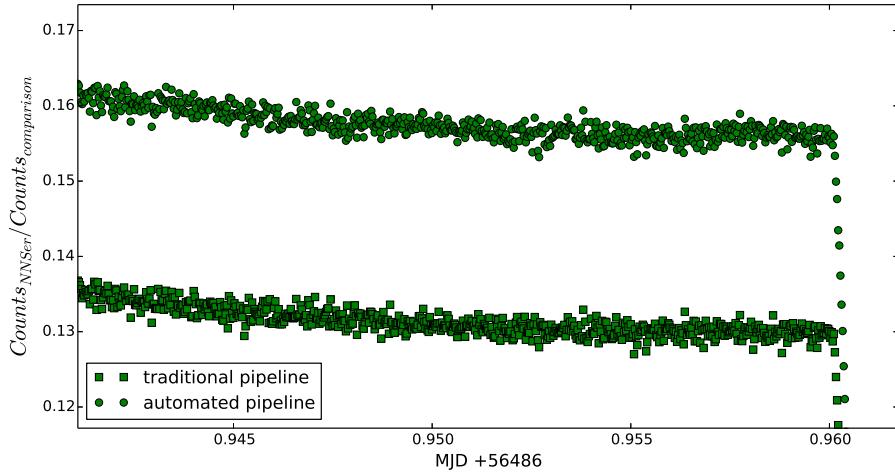


Figure 4.5: A closer look at the comparison of the light-curves for NN Ser in the Sloan g filter, from the start of the run to the beginning of the eclipse ingress. Square data points were generated by the traditional pipeline and circles by the automated pipeline. The vertical offset applied to the circles is 0.027.

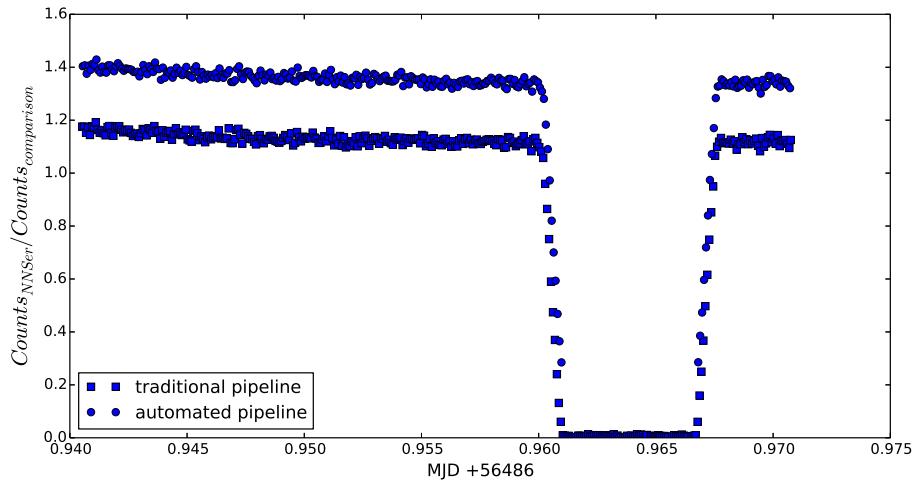


Figure 4.6: Comparison of the light-curves for NN Ser in the Sloan u filter. Square data points were generated by the traditional pipeline and circles by the automated pipeline. The vertical offset applied to the circles is 0.23. Note that the automated pipeline has no data for the duration of eclipse totality. We discuss the reason for this in the text.

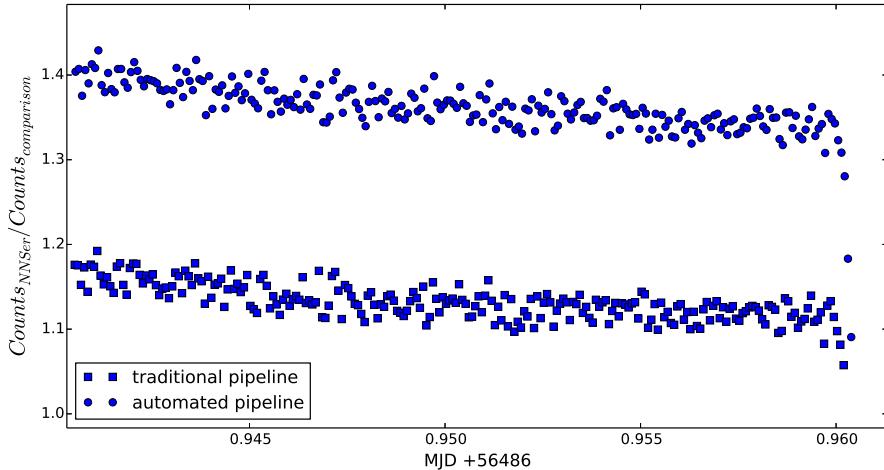


Figure 4.7: A closer look at the comparison of the light-curves for NN Ser in the Sloan u filter, from the start of the run to the beginning of the eclipse ingress. Square data points were generated by the traditional pipeline and circles by the automated pipeline. The vertical offset applied to the circles is 0.23.

tractor, to determine the apertures on each frame, objects that do not meet the required signal to noise ratio on any particular frame will not be detected and therefore have no aperture defined for that frame. No aperture means that we will have no photometry. This means that objects that fade or are generally quite faint, might disappear on some frames and then re-appear on subsequent frames. The tracking algorithm allows a re-appearing object to be identified with an object that had previously disappeared on earlier frames provided that the pixel location is roughly similar. An illustration of this can be seen in figures 4.4 and 4.6 where the automated pipeline loses the target object in the ‘g’ and ‘u’ bands after the ingress of the primary eclipse, but picks it up again at the start of egress. In contrast, the traditional reduction pipeline can have apertures that are linked to other objects in the field and can therefore continue to measure flux in the aperture for the target even if the target is barely detectable above the sky background.

As an alternative to comparing the light-curves side-by-side we can check the systematic differences between the pipelines by comparing their measured flux values as a ratio of each other. We used both pipelines to

Table 4.1: Table showing the statistics of the relative photometry produced by dividing the flux counts from the automated pipeline by the flux counts from the traditional pipeline for objects ‘0’ and ‘2’ in figure 4.1.

Filter	Relative flux for object ‘0’	Relative flux for object ‘2’
	<i>mean</i> [<i>std.dev</i>]	<i>mean</i> [<i>std.dev</i>]
‘i’	0.991[0.003]	0.988[0.002]
‘g’	0.994[0.003]	0.995[0.003]
‘u’	0.988[0.007]	0.987[0.005]

produce light-curves for the comparison star labeled ‘0’ in figure 4.1. Rather than plotting each light curve separately, we have plotted the counts measured by the automated pipeline divided by the counts measured by the traditional pipeline. This is shown in figure 4.8. The statistics of this data set is shown in table 4.1. The traditional pipeline gives a slightly higher reading for the overall flux than the new automated pipeline, resulting in a mean that is less than unity. The likely cause of this is the different size of aperture used by each pipeline. This will be investigated as the pipeline is enhanced to give calibrated photometry in future versions.

As a deeper inspection of the systematics between the two pipelines we plotted relative flux counts measured in each pipeline as a ratio of each other. Firstly we computed the relative flux of object ‘2’ to object ‘0’ in the automated pipeline, F_{auto} . Then we computed the same ratio for the same two objects in the traditional pipeline, F_{trad} . These two sets of values were combined to produce the plot in figure 4.10 by dividing these two ratios, $\frac{F_{auto}}{F_{trad}}$. The amplitude of the scatter in these plots is on the order of 1% in the blue channel and 0.1% for the red and green channels.

4.3 Object matching

As discussed in chapter 2, the automated pipeline can struggle to cross-identify the same object across the channels, r, g, b. This is usually only a problem in crowded fields where the average pixel position of the object does not clearly distinguish it from nearby objects. In other words, an object might have been identified as the same object in the blue channel (due to its proximity on the image) but is actually a near neighbour to this object in the red and green

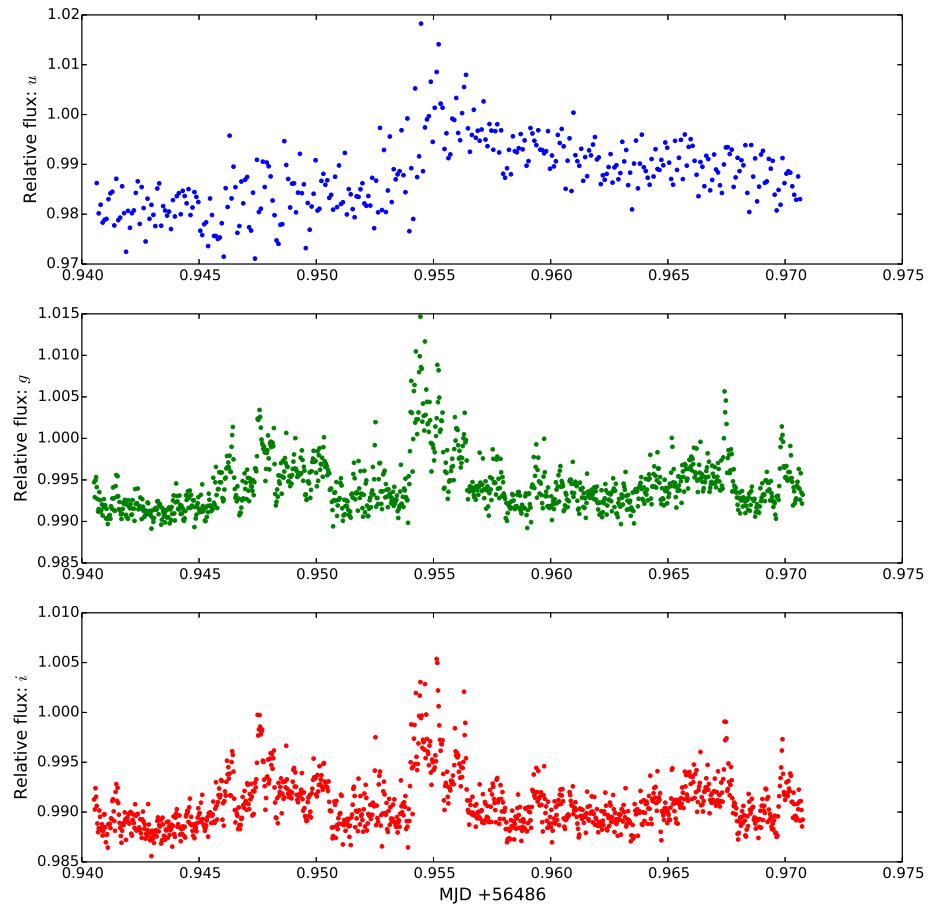


Figure 4.8: Object 0: A comparison of the flux measurements for a single object labeled ‘0’ in figure 4.1. The plot is produced by dividing the flux counts as measured by the *automated* pipeline by the flux counts as measured by the *traditional* pipeline.

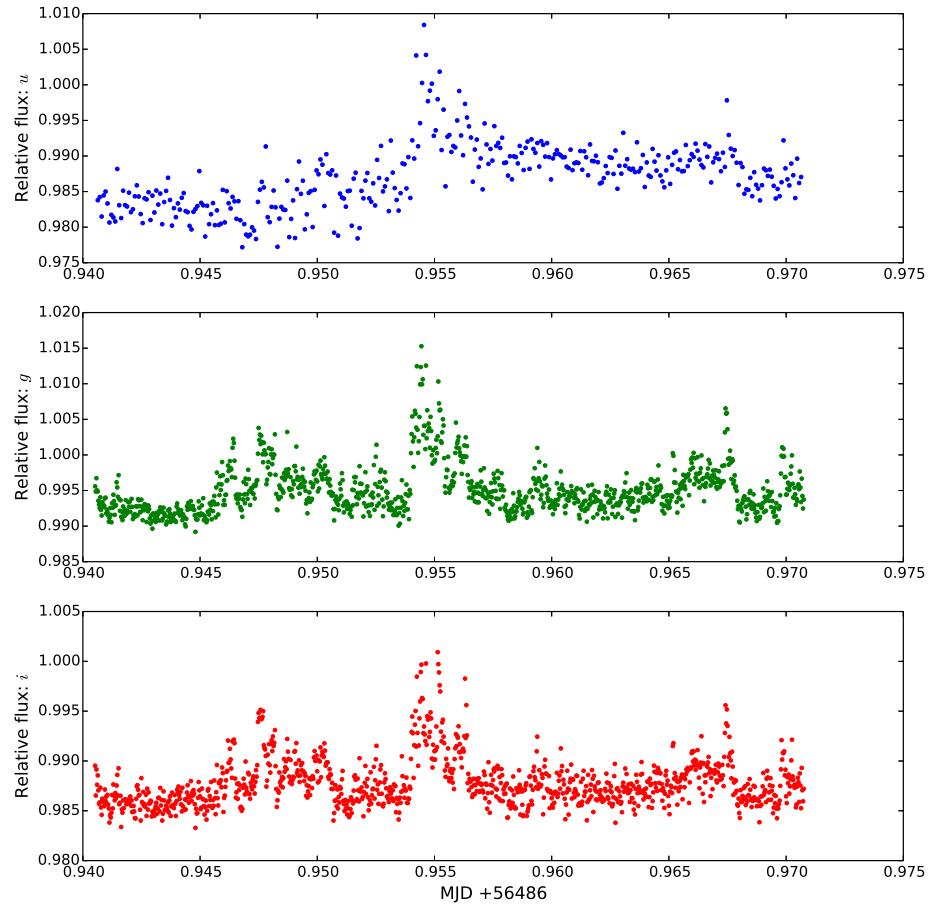


Figure 4.9: Object 2: A comparison of the flux measurements for a single object, labeled ‘2’ in figure 4.1. The plot is produced by dividing the flux counts as measured by the *automated* pipeline by the flux counts as measured by the *traditional* pipeline. Comparing this plot to the similar plot for object ‘0’ in figure 4.8 show very similar systematics.

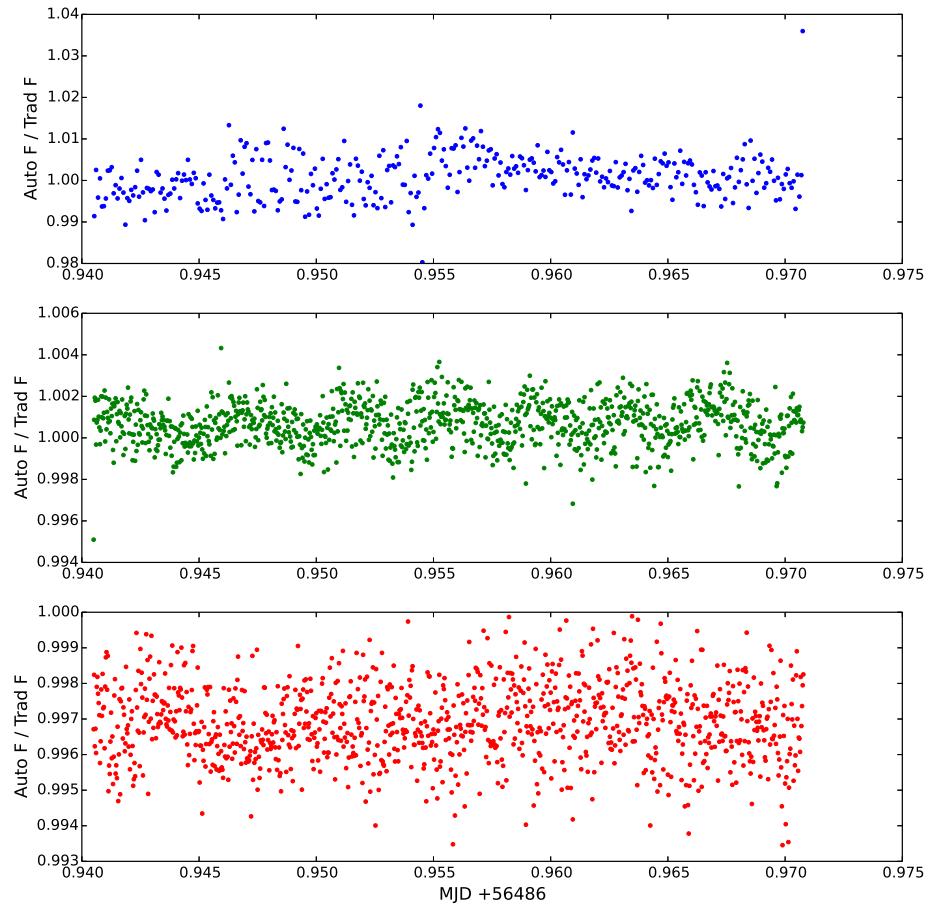


Figure 4.10: Traditional pipeline vs the Automated pipeline: A comparison of the flux ratio F , defined as the flux ratio of the Counts for object ‘2’ to object ‘0’ for the automated pipeline vs the same set of values derived from the traditional pipeline. There appears to be a regular signal in the green channel at an amplitude of about 0.1%. This is not seen in other similar plots generated for other runs.

channels, so the pipeline has mistakenly assigned all three measurements as belonging to the same object.

By plotting colour-colour and colour-magnitude diagrams for a few of the crowded fields in the ULTRACAM archive we can get an indication of the severity of this mis-matching problem. Although the automated pipeline does not perform a calibration of the magnitudes of the objects, by using standard stars and extinction corrections, it is still possible to create colour-colour diagrams provided that we are not concerned with the correct offsets for our $(u - g)$ and $(g - r)$ axes. We can also produce a colour-magnitude diagram if we have a field that contains objects that are all at the same distance from us. In these automatically produced plots, there are some outliers showing colours that are too extreme to be genuine astronomical bodies.

In figure 4.11 we have plotted colour-colour diagram of *run010* taken on 2013-07-21. In this diagram it is clear that there are some outliers. If we zoom in these (see figure 4.12) and show the labels the objects as they have been identified in the automated pipeline, we can go back to the web interface to check if the cross-identification has worked correctly. An outlier in figure 4.12 is identified as object ‘465’ in the run. Looking at figure 4.13 we can see that the pipeline has incorrectly identified a neighbour of object in the blue channel as it was brighter in blue and relatively nearby. Figure 4.11 uses a colour map to show the relative distances of the object from channel to channel. The relative distance has been calculated by taking the Pythagorean distance (D) of the object’s separation from the red to the green channel, (D_{rg}) and the object’s separation from the red to the blue channel, (D_{rb}).

$$D = \sqrt{D_{rg}^2 + D_{rb}^2}, \text{ where}$$

$$D_{rg} = \sqrt{(x_r - x_g)^2 + (y_r - y_g)^2} \text{ and}$$

$$D_{rb} = \sqrt{(x_r - x_b)^2 + (y_r - y_b)^2}.$$

It seems that this separation distance D is not clear discriminator of whether an object is matched correctly. In crowded fields, confusion occurs when objects are close together. Setting the minimum matching distance threshold to a lower value does not significantly reduce the number of mismatches. In fact, the systematic differences between each of the channels, especially towards the edge of the image, are larger than the separation of individual objects on each field. This can also be seen in figure 2.9. We con-

clude that in order to address this issue with the pipeline we need a more robust algorithm for clearly identifying each object’s position in each of the channels (perhaps with an accurate WCS solution) and then performing the cross identification.

4.4 Finding variable objects

In this section we present two examples of how the web interface makes browsing of the ULTRACAM archive accessible and fast and allows us to discover new variable objects. The first example is the serendipitous discovery of GU Mus, the X-ray transient object that was observed in quiescence in May 2005 at the VLT. We originally suspected that it was a cataclysmic variable. GU Mus was at magnitude of 20.65 in Sloan g at the time, (Shahbaz et al., 2010), and was considered to be rather faint for a typical ULTRACAM target. Since the ULTRACAM data does not include the pixel position of the target object in the field, it was not obvious to which one out of the 205 objects identified by the pipeline was GU Mus. The normal method for finding the target object is to revert to finding charts and existing catalogs. Figure 4.17 contains image captures from the web browser interface showing how the light curves and the field are represented to the user. By pressing the ‘left’ and ‘right’ arrow keys, the user can scroll through all of the light curves like pages in a book. For this run, there were 205 individual light-curves available for browsing. By quickly flipping through the light curves the user can spot any obvious variability. Most objects show light-curves similar to the upper one shown in figure 4.17 with no obvious variability above the noise. The object with the identification number ‘58’ however, was showing clear evidence of flickering. We assumed that we had discovered a new CV. Since GU Mus is a faint object, it took us a day to find an accurate finding chart. When we did so, we realised that our ‘new’ variable was actually GU Mus itself. Although our initial excitement was dampened, this incident was seen as good evidence that the automated pipeline and the browser interface is capable of surfacing faint variable objects. As a bonus, the object just a few arc minutes to the right turned out to be variable too. It is a W UMa variable and is discussed in chapter 5.

The second example from the browser interface is the detection of an

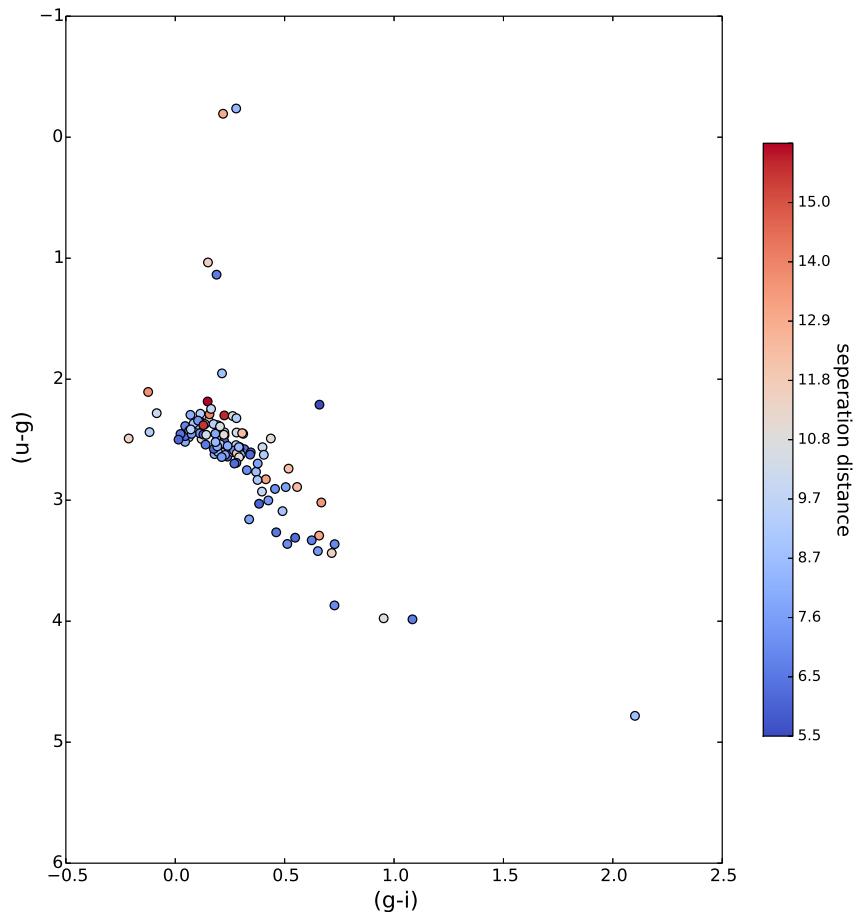


Figure 4.11: The colour-colour plot of run: *2013-07-21/run010*. The plot contains 110 objects located near the Kepler exoplanet host KIC5115978. The offsets on the x and y axes are both arbitrary as the photometry has not been calibrated with photometric standards. The outliers with extreme red and blue colours are due to mistaken classification by the automated pipeline. The colour map indicates how separated the (x, y) positions for the object are in each of the 3 channels.

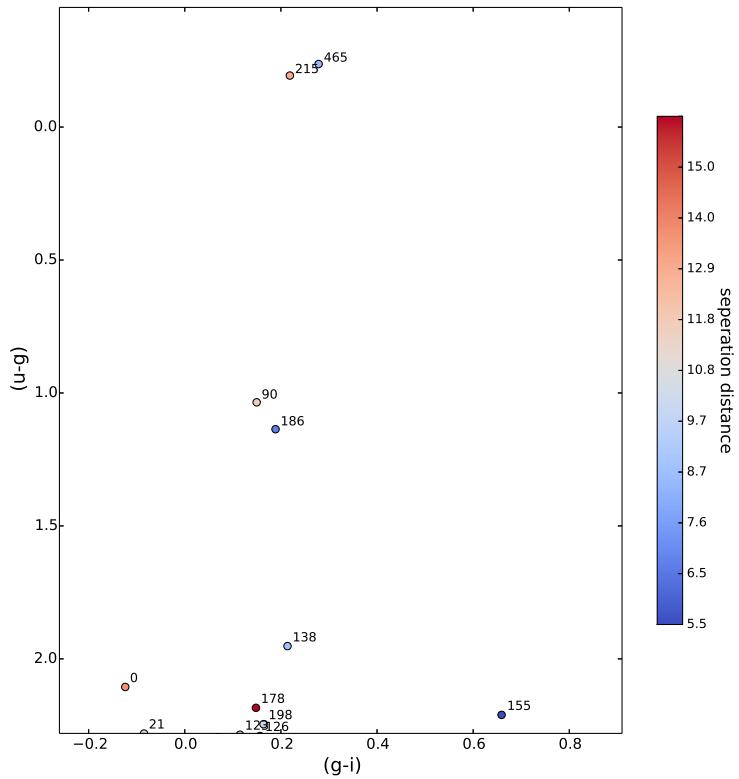


Figure 4.12: A closer look at some of the outliers in figure 4.11 showing the automated pipeline's identification label for each object. Going back to the web browser and inspecting them shows that both of the object's labeled '465' and '215' were mis-identified in the blue channel.

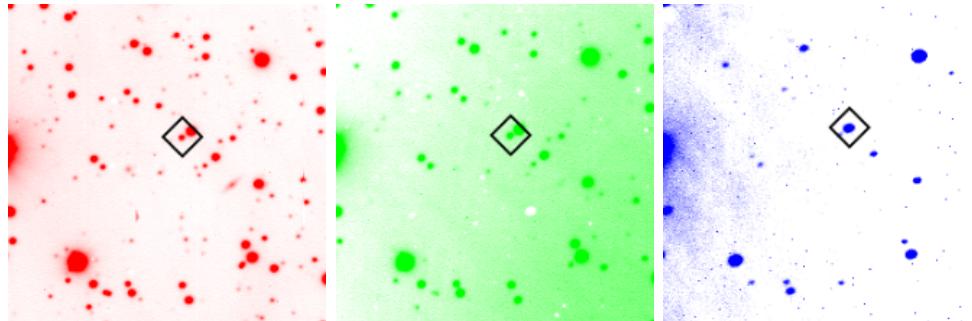


Figure 4.13: Checking the matching of the colour-colour outlier labeled '465' in figure 4.12. The object has been incorrectly identified with a neighbour that is brighter in the blue channel.

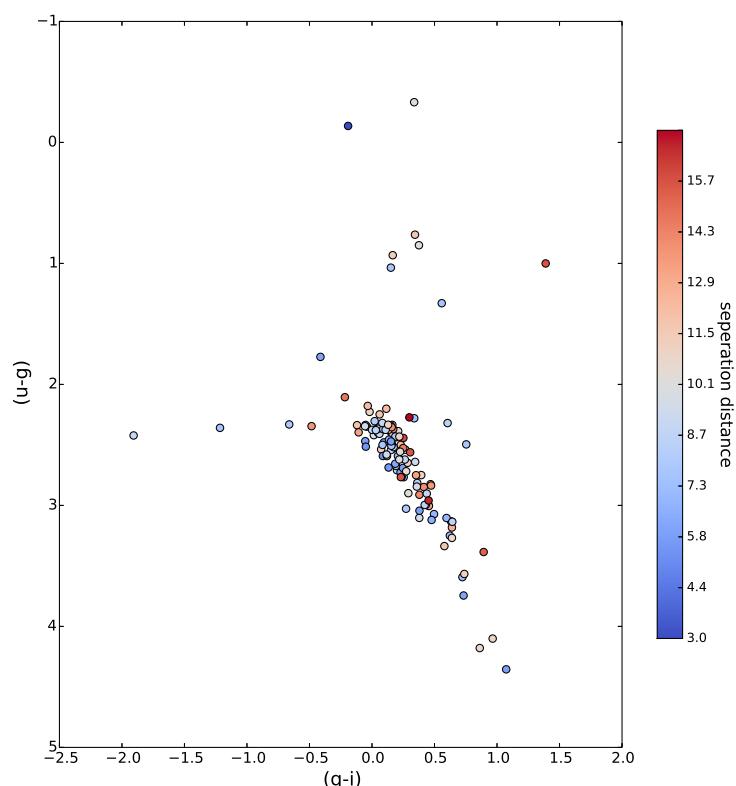


Figure 4.14: The colour-colour plot of run: *2013-07-21/run011*. The offsets on the x and y axes are both arbitrary as the photometry has not been calibrated with photometric standards. The outliers with extreme red and blue colours are due to mistaken classification by the automated pipeline.

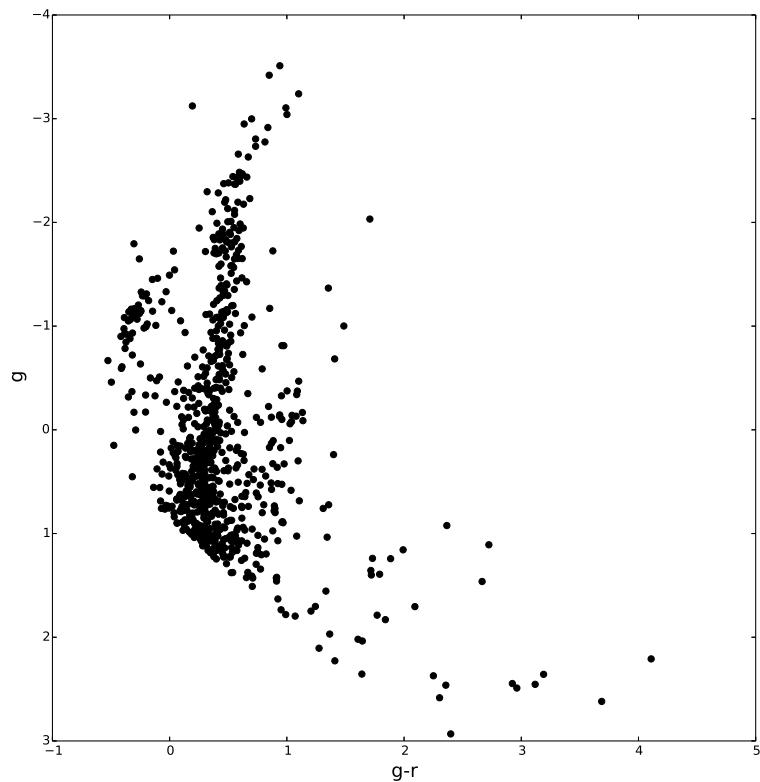


Figure 4.15: A colour-magnitude plot of run: `2011-04-22/run019`. The plot includes 790 objects found in a field taken of the outer perimeter of the globular cluster *Omega Centaurus*. The offsets on the x and y axes are both arbitrary as the photometry has not been calibrated to photometric standards. We have magnitude limited our sample to the 790 brightest objects in the Sloan 'r' filter.

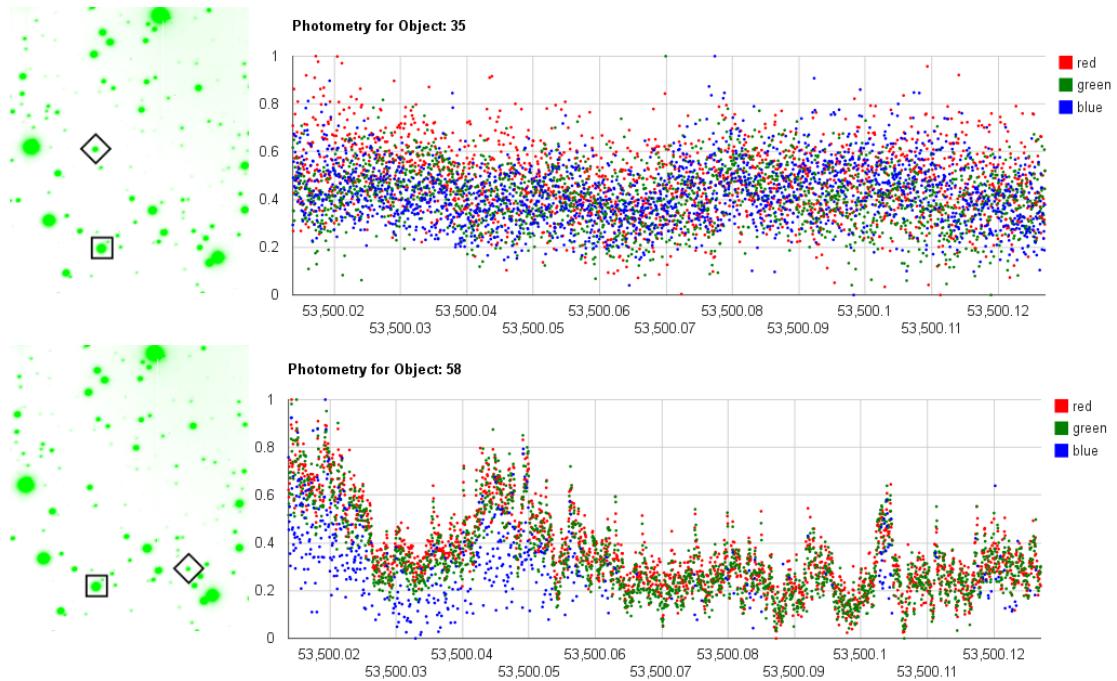


Figure 4.16: Flickering: A screen-capture from the web browser interface showing how variable objects reveal themselves when scrolling through the light-curves. The upper plot shows a non-variable object. The lower plot shows an object that is exhibiting flickering. The currently selected object is indicated by the diamond and the comparison object is indicated by the square. The y-axis of the light-curve is the relative flux of the selected object to the comparison object normalised to include the full range of variability within the same set of axes. The scaling of the y-axis was chosen to accentuate any variability in the light-curve and make it clearly visible in all three colours.

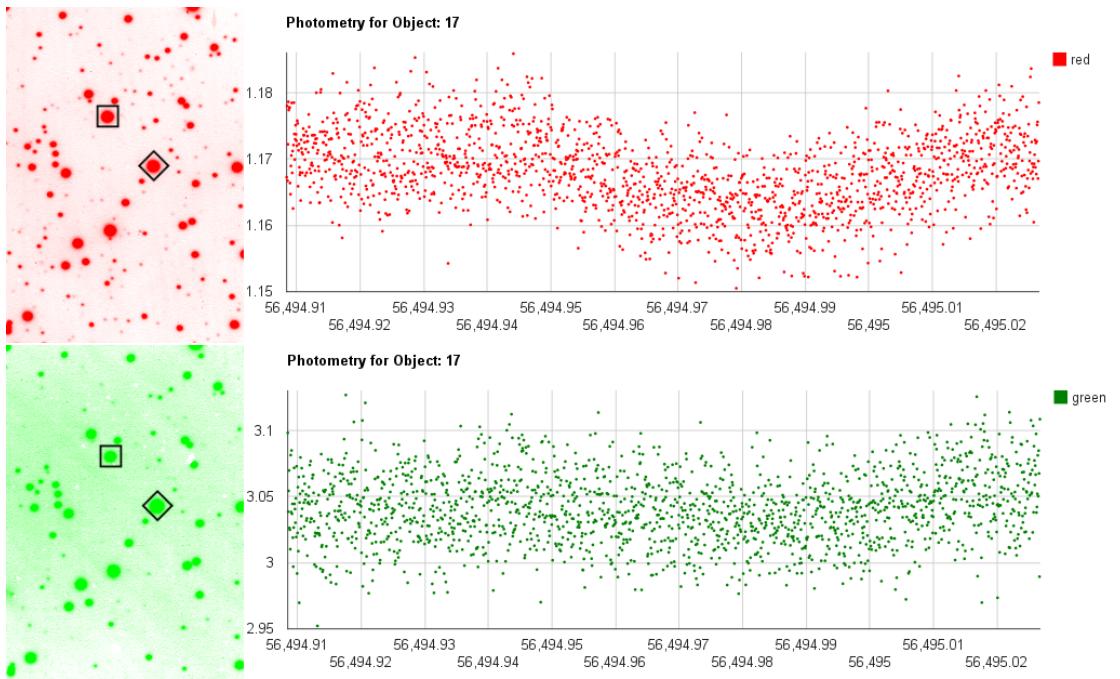


Figure 4.17: Exoplanet transit: A screen-capture from the web browser interface showing an exoplanet transit for KIC 5115978 in the Sloan i and g bands. The target object is indicated by the diamond and the comparison object is indicated by the square. The y-axis of the light-curve is the relative flux of the selected object to the comparison object.

exoplanet transit through inspection of the light-curves produced by the automated pipeline. The run was taken at the WHT in July as a follow up of the KOI-823, or KIC 5115978. This object is known to have at least one exoplanet, (Borucki et al., 2011). Although the transit does not have a large amplitude (approximately 1% in relative flux), it is still clearly visible when browsing through the light-curves in the browser interface.

4.5 Reducing the entire ULTRACAM archive

In order to make the processing of the ULTRACAM archive as free from manual intervention as possible, a few additional scripts were written to co-ordinate the steps of the pipeline as described in the previous two chapters. These wrapper scripts were designed to trigger the pipeline for a full day's worth of data with a single command. The University of Warwick has a high

performance computing facility called ‘Cluster of Workstations’ (CoWS) that uses idle computing time on all of the desktop machines in the department. A script was written that sends the automated pipeline processing jobs to this shared facility. Using this approach, it was possible to process nearly all of the ULTRACAM archive in about 2 weeks. At present, 347 nights, out of a total of 406, have been processed and are available for viewing on a web server hosted at the University of Warwick.

The runs that have not been processed are a few very high cadence runs with exposure times of less than 0.1 seconds and numbers of frames exceeding 50,000. The automated pipeline can take more than 8 hours of processing time for these runs. The Cluster of Workstations has sufficient computing power to complete the processing of these runs. However, it was felt that a different and more optimised version of the pipeline should be built that will treat these runs differently. These runs usually have very few objects in the fields, typically only the target object and the comparison and this situation is handled very well by the traditional pipeline. The processing of high cadence runs with very few objects was not a goal for this project.

Of these 347 nights, approximately 20% of the science runs have been investigated for objects with variability. The method of investigation is to perform by a visual check of the light-curve. The web interface is designed such that it is easy for the viewer to examine the light-curves of all of the objects systematically. The user-interface allows the users to see each light curve in turn by pressing the ‘right’ and ‘left’ arrow keys on the keyboard. More information on how to use this interface can be found in the User Manual, appendix A.

Chapter 5

Objects identified by the automated pipeline

5.1 Object identification

The pipeline is usually invoked by running a single command on a night's worth of data. For example, to build the light-curves for the night of, say, *2014-08-21*, then a single command, `daybuilder.py 2014-08-21` is issued from the command line. The pipeline then runs through all of the data for that night and generates a set of web pages. Depending on the amount of data for that night, this can take 1 hour to 8 hours.

For each night, an index page, which shows a list of all of the runs in the night along with thumbnail images of the field of view, is created. This allows the user to quickly navigate to the runs that are of interest. In other words, runs that contain science data, rather than acquisitions, biases or flat-fields. By clicking on the thumbnail of the run, the user is taken to a run page. This page shows the full image for each of the three channels. These images are created by stacking all of the individual frames in the run. The page also shows all of the objects that have been identified and have light-curves available. The user can view the light-curves by using the mouse to click on each object, or can scroll through all of the light-curves systematically, by using the left and right arrow keys.

Scrolling through the light curves in a systematic fashion makes it easy

for the user to quickly identify which objects are showing an obvious variability. All of the objects listed below were discovered in this way.

With this manual inspection it is possible to inspect the light curves at a rate of about 1-2 objects per second. In future, we plan to apply some automated tests to these data to perform the light curve inspection as an integral stage of the automated pipeline. Algorithms to perform these sorts of tests are already known and becoming increasingly more widespread as more large scale sky surveys are being used throughout astronomy research. We plan to re-use work from one or more of these surveys. The recently published Astrokit software, (Burdanov et al., 2014) is one tool that we plan to trial in the next version of the automated pipeline.

5.2 Discovered objects

As discussed in the introduction, chapter 1, we expect to find some new variables in the ULTRACAM archive. These will be objects displaying some kind of variability that is clearly visible over the length of the run and with the cadence of the observations. This will include eclipsing binaries, contact binaries, flare stars, RR Lyrae and δ Scuti stars. We might also expect to see other short period variables like cataclysmic variables and DV white dwarfs. Since the pipeline is able to track slow moving objects, we can also expect that we will capture some photometry of asteroids that drift through the field.

We have only inspected approximately 20% of the reduced data but we have found a few dozen variable objects so far. Below we list a selection of these objects.

Table 5.1: Table of a few of the interesting objects that have been revealed using the automated pipeline. This list contains just a few highlights of the dozen objects detected so far.

Type	ID	Position, J(2000)
W UMa	2005-05-10-run012-73	11:26:26.2 –68:40:50
W UMa	2013-07-21-run010-48	19:44:09.3 +40:16:34.0
W UMa	2013-07-21-run010-163	19:44:10.1 +40:18:09.1
Eclipsing binary	2013-07-21-run011-162	19:54:01.7 +40:37:34
δ Scuti	2013-07-21-run010-23	19:44:19.7 +40:16:45.3
Asteroid 1998 SU139	2011-08-26-run014-110	20:51:12.0 –08:31:25 at MJD=55800.038
Asteroid 9108 Toruyusa (1997 AZ6)	2009-01-04-run024-61	08:04:52.3 +16:18:10.6 at MJD=54836.26642

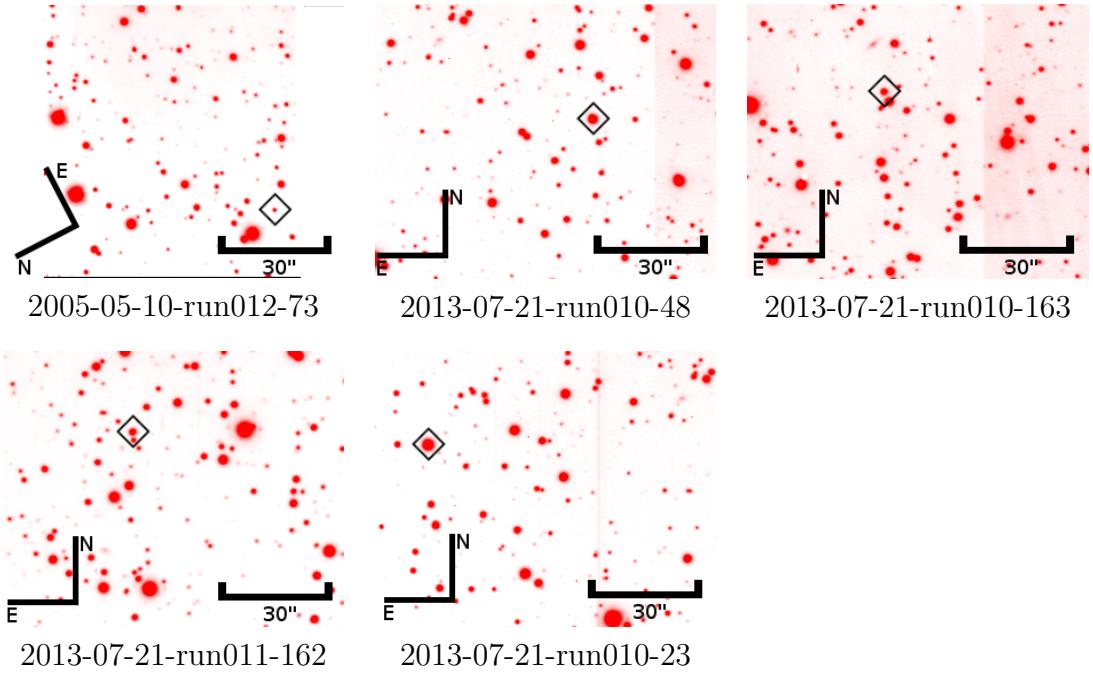


Table 5.2: Finding charts for the objects listed in table 5.1, excluding the asteroids. These charts are taken from the Web interface of the processed ULTRACAM archive and are to be used to help identify the target when reviewing the data in the web browser.

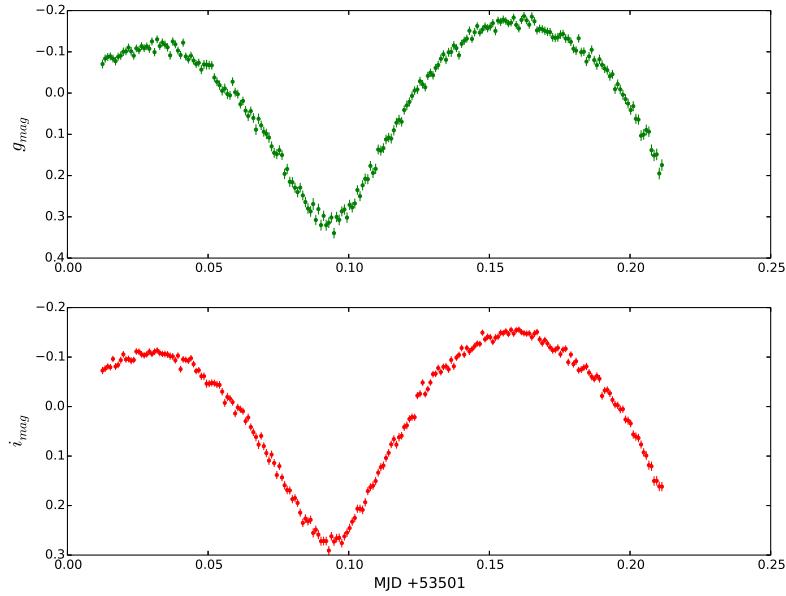


Figure 5.1: Sloan i and g light curves of object: 2005-05-10-run012-73. The data points are binned by a factor of 16.

5.2.1 W UMa: 2005-05-10-run012-73

The original target of this run was the X-ray transient, GU Mus, that was observed in quiescence in 2005. The compact object, thought to be a black hole, was accreting at the time and flickering is apparent in the optical light curve in all three of the Sloan i, g, and u bands, (Shahbaz et al., 2010). About 4 arc minutes to the west we have detected a suspected W UMa contact binary with an apparent magnitude of ~ 21 in Sloan g and period of ~ 390 minutes or 0.27 days. Visual inspection of the light-curves in figure 5.1 shows evidence of the O’Connell effect where the peak brightness of the second maximum is larger than the first. The object was too faint in the Sloan u band for us to produce any photometry.

The cause of the O’Connell effect is something that is still under some debate. Differences in the depth of the minima of the light-curves is something that is expected and is caused by the different sizes and temperatures of each component in the system. Differences in the height of the maxima though is less obvious. Based on a geometric model of the eclipse though, we would

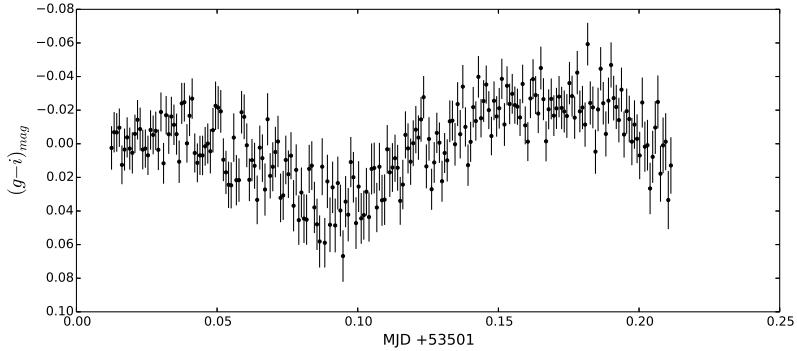


Figure 5.2: The $(g - i)$ colour light-curve of object: 2005-05-10-run012-73. The data points are binned by a factor of 16.

expect the maxima to be of equal brightness. When the light-curve is at a maximum we are seeing the system side-on with the maximum area ellipsoidal faces presented to the observer. Current explanations for the O’Connell effect consist of star-spots that are locked in rotational synchronisation with the orbit and are presented preferentially on one side of the system; gas streams that create a hot-spot on or near one of the stars; and circumstellar material that is kinetically heated and gathered preferentially on the leading edge of the orbiting bodies, (Wilsey and Beaky, 2009).

In figure 5.2 we plot the $(g - r)$ colour of the object. There is a clear modulation in the colour which must be due to temperature differences around the geometry of the system. The amplitude of this colour variation is $\Delta(g - r) = 0.06$ magnitudes. The minimum or coolest $(g - r)$ colour corresponds with the minimum of the light-curve. This is consistent with the idea that the hemispheres of the components in contact with each other are hotter due to a mutual reflection effect, while the outer hemispheres are cooler. The amplitude of the colour variation is $\Delta(g - r) \sim 0.06$ magnitudes.

5.2.2 W UMa: 2013-07-21-run010-48

This variable was the first one picked up by the automated pipeline and revealed itself during the very early development of the software. It appears on a relatively crowded Kepler field which has at least 5 newly discovered variable objects. Although this field has been monitored by the Kepler satellite,

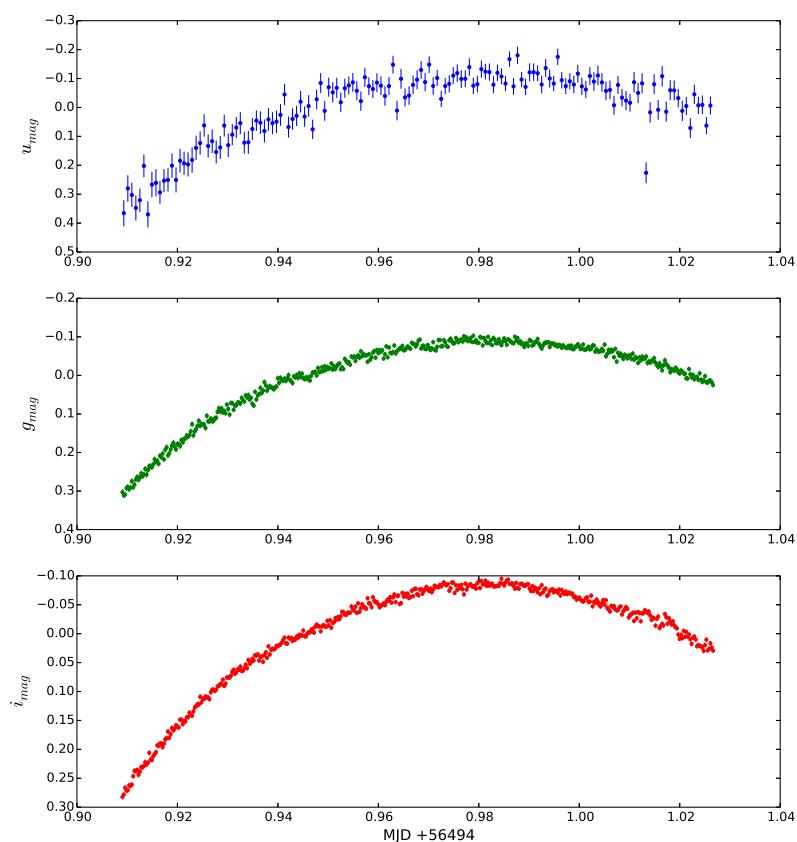


Figure 5.3: Sloan i, g and u light curves of object: 2013-07-21-run010-48. The data points are binned by a factor of 4.

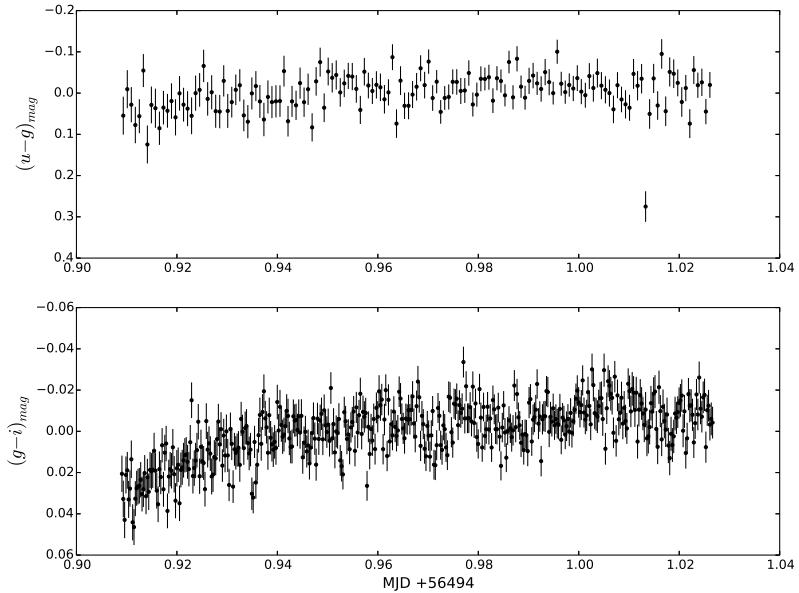


Figure 5.4: Colour plots of $(u-g)$ and $(g-i)$ for object: 2013-07-21-run010-48. Data points are binned by a factor of 4.

a search through the Kepler online archive ¹ has not resulted in any data for this object. Since the aim of the Kepler mission is to discover exoplanets via observation of transits on stars that are relatively constant, it usually discards stars that are clearly variable.

We suggest that this object is another W UMa category variable with an approximate period 400 minutes or 0.27 days. Unfortunately we do not have complete period coverage of the object's orbit and therefore this period is a very rough estimate. The target, KOI-823, has been observed with ULTRACAM on another occasion, 2013-07-28, but the field was set up differently (more to the west) and this object was not included.

The colour plots for the object are shown in figure 5.4. As for the previous object, it seems that the $(g-i)$ colour peak coincides with the brightness maximum. However, it appears that the peak of the $(u-g)$ plot occurs significantly before the peak in brightness. *Comment: Explanation for this?*

¹https://archive.stsci.edu/kepler/data_search/search.php

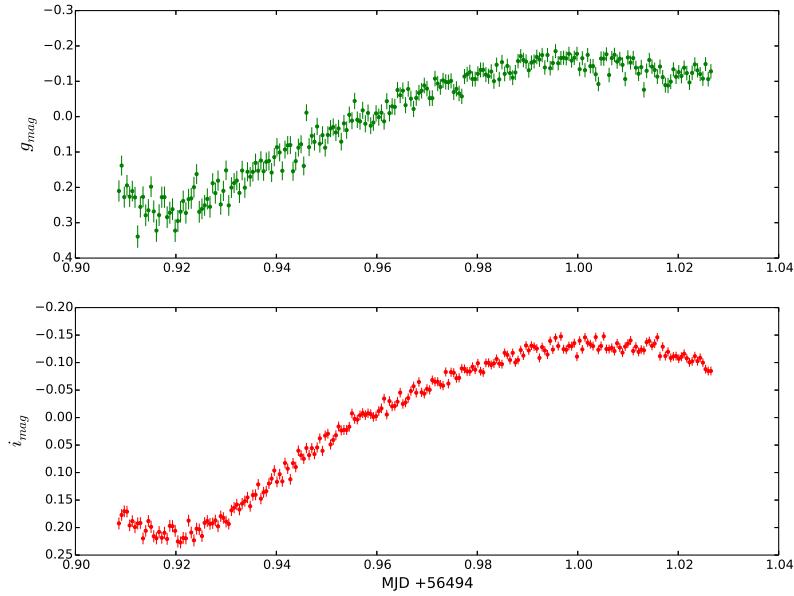


Figure 5.5: Sloan i and g light curves of object: 2013-07-21-run010-163. Data points are binned by a factor of 8.

5.2.3 W UMa: 2013-07-21-run010-163

This object is another suspected W Uma variable found on the same field as the previous object. The light-curve for this object includes one minimum and one maximum, although not a full orbit. From this light-curve we can estimate a period of about 460 minutes or 0.32 days.

5.2.4 Eclipsing binary: 2013-07-21-run011-162

The light curve of this object includes a primary eclipse. Since the eclipse profile has a 'flat-bottom' we can conclude that this eclipse is total (ie that the primary is completely obscured by the secondary during the eclipse). This also means that the primary has a smaller diameter than the secondary. The primary eclipse duration is 37 minutes. The depth of the eclipse is about 0.7 magnitudes which corresponds to a drop in flux of 50%.

It is notable that the ingress and the egress demonstrates broad 'wings' suggesting that the object being eclipsed (primary) is extended and the shape

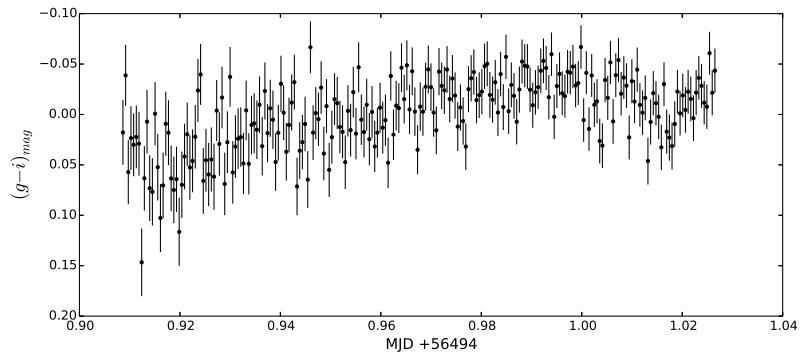


Figure 5.6: The $(g - i)$ colour light-curve of object: 2013-07-21-run010-163.
Data points are binned by a factor of 8.

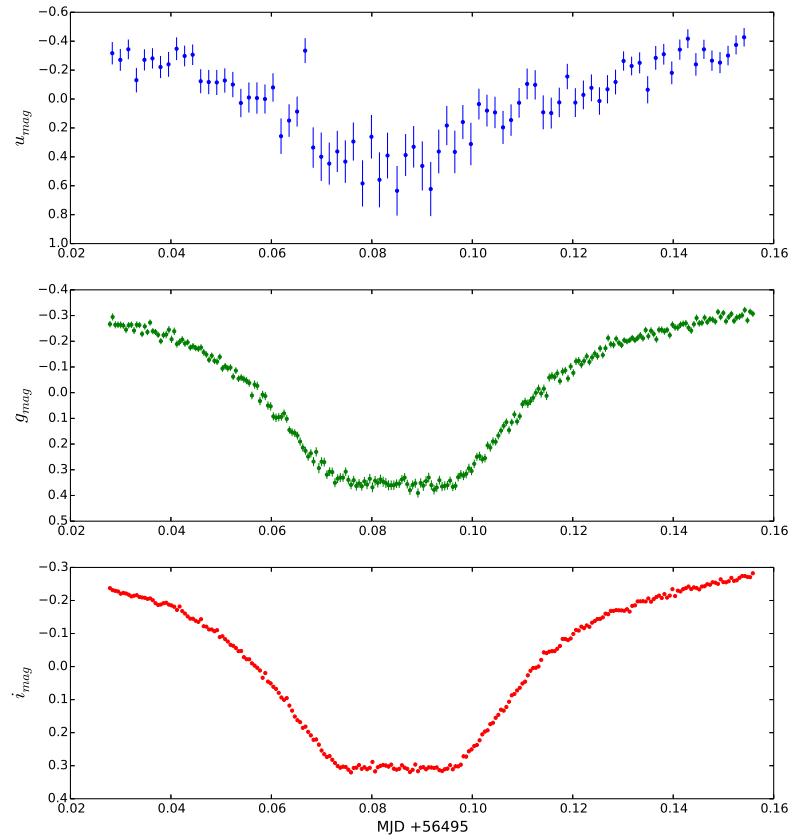


Figure 5.7: Sloan i, g and u light curves of object: 2013-07-21-run011-162.
Data points are binned by a factor of 8.

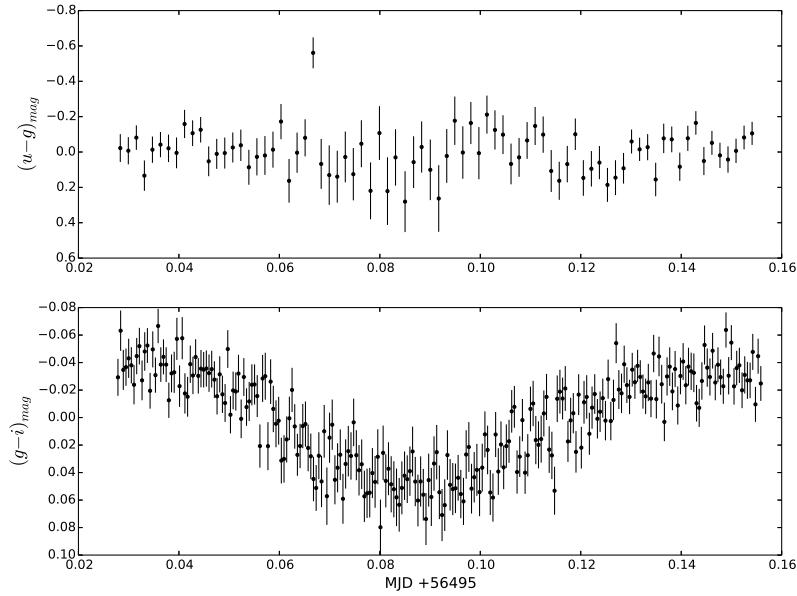


Figure 5.8: $(u - g)$ and $(g - i)$ plots for the object: 2013-07-21-run011-162. Data points are binned by a factor of 8.

of the curve suggests tidal distortion of the secondary. The colour plots in figure 5.8 show that the minimum of the $(g - i)$ coincides with the center of the eclipse. The colour plots do not show the clear flat-bottom that we see in the light-curves for i and g .

Although the object is in a Kepler field and is, in fact, close to KOI-1546, a search through the Kepler archive² reveals that it is not listed.

5.2.5 δ Scuti: 2013-07-21-run010-23

This object was found on a run that included the exoplanet host, KIC5115978, which has at least one planet, (Borucki et al., 2011). The new variable is located about 6 arc minutes away from the target. A search of the Kepler data archive³ gives no results for this object. Unfortunately this object is not in the Kepler field of view, but lies in a position that does not fall on the Kepler CCD. We have no further photometry for this object.

²https://archive.stsci.edu/kepler/data_search/search.php

³http://archive.stsci.edu/kepler/kepler_fov/search.php

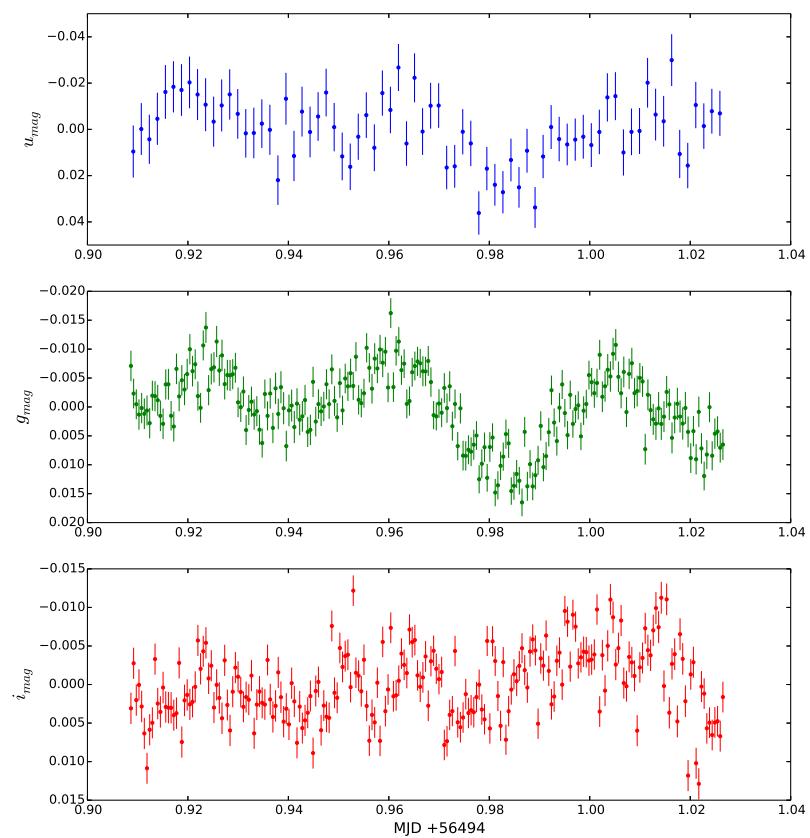


Figure 5.9: Sloan i, g and u light curves of object: 2013-07-21-run010-23. Data points are binned by a factor of 8.

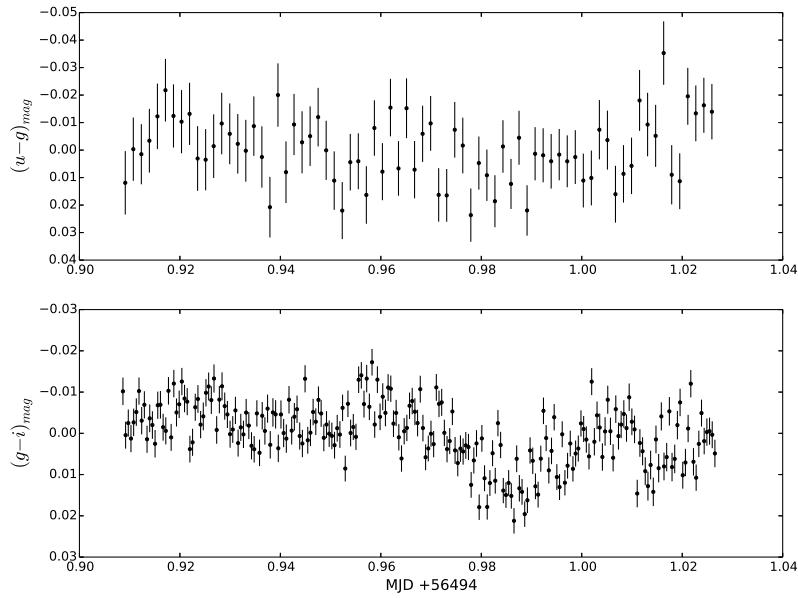


Figure 5.10: $(u - g)$ and $(g - i)$ plots for the object: 2013-07-21-run010-23. Data points are binned by a factor of 8.

There is evidence that the colour is varying in phase with magnitude, most noticeably in $(g - i)$, see figure 5.10. The light-curve is the shape we would expect, with a steeper slope during increasing flux than decreasing. The period is approximately 0.04 days, consistent with the expected range 0.03–0.3 days.

5.2.6 Asteroid: 1998 SU139

The light curve of this object displays a clear sinusoid which must be caused by variations in reflected sunlight that is modulated as the object rotates. A periodogram of these data peaks at a frequency of 0.0114 minutes⁻¹ (a spin period of 1.46 hours) for the 'i' filter and 0.0107 minutes⁻¹ (a spin period of 1.56 hours) for the 'g' filter. This assumes that one period in the light curve is equal to exactly one rotation of the asteroid. This might not be true if the asteroid has several light and dark patches on its surface.

Rotation of asteroids is driven by a process known as the YORP effect, (Bottke et al., 2006).

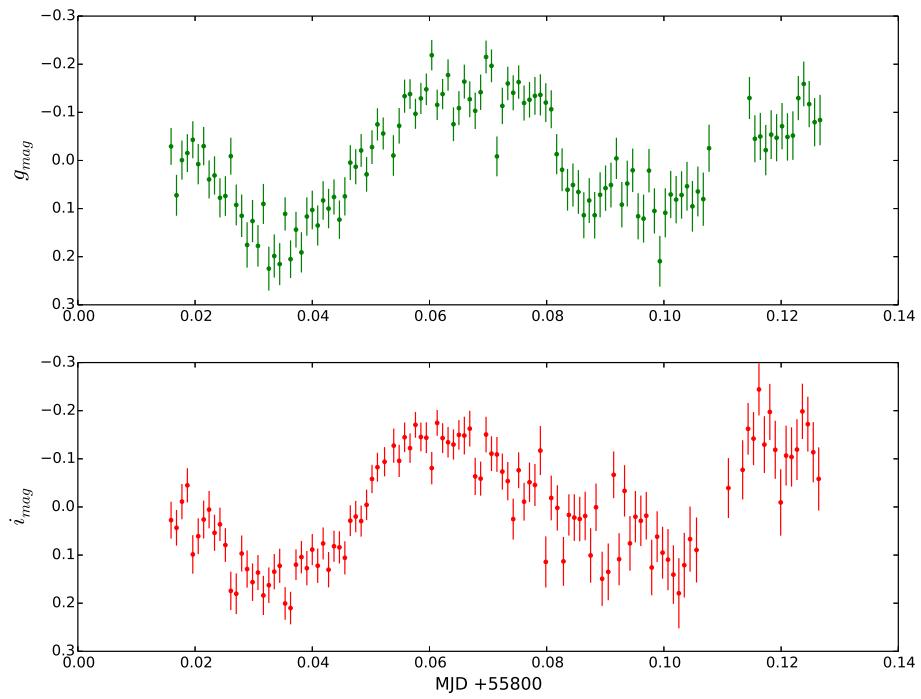


Figure 5.11: Sloan i, g light curves of asteroid: 2011-08-26-run014-110. Data points are binned by a factor of 4.

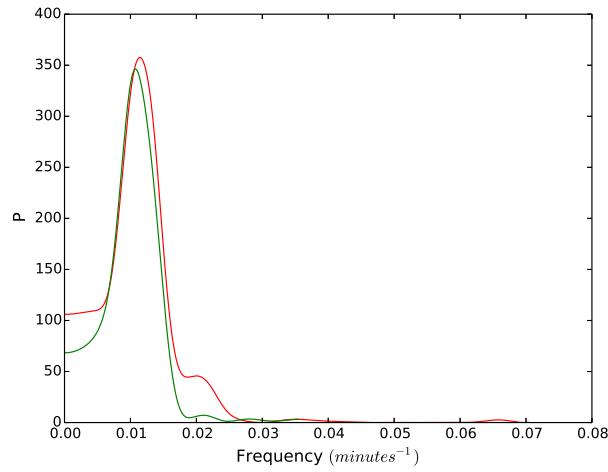


Figure 5.12: Periodogram of both the i and g light curves for the object: 2011-08-26-run014-110. The peaks occur at $g_{peak} = 0.0107 \text{ minutes}^{-1}$ and $r_{peak} = 0.0114 \text{ minutes}^{-1}$

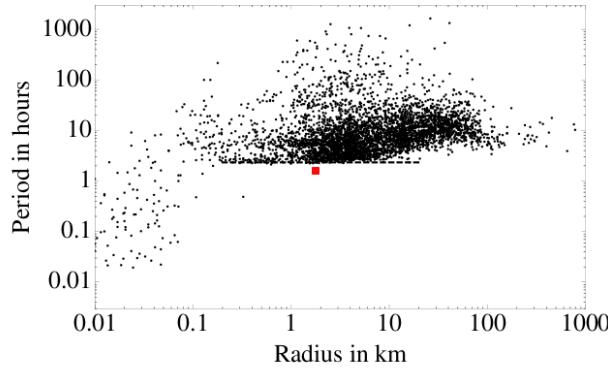


Figure 5.13: The spin period distribution as a function of radius for near-Earth (NEA), Mars crossing (MCA) and Main Belt (MBA) asteroids as reported in the Asteroid Lightcurve Database (Warner et al. (2009)). The dashed lines indicate the critical surface disruption period $P_d \sim 2.33$ hours for radii $R > 250$ metres. Taken from Jacobson et al. (2014). The red square on the plot shows the position of our object as estimated in this text.

A look-up using the *NEOChecker* tool on the website of the IAU's Minor Planet Center⁴ returns a result for this candidate as, most probably, asteroid *1998 SU139*. The world coordinates reported by the tool differed from our own determination by about $6.5'$.

There is a clear relationship between the size of the asteroid and the minimum spin period. It has been shown through observations that, for an asteroid with a diameter greater than 250 metres, the spin period cannot be less than 2.33 hours, (Jacobson et al., 2014). The theoretical reasoning for this 'spin cut-off' is that, assuming these asteroids are 'rubble-piles' then, above certain angular velocity, the centrifugal forces pulling the rubble pile apart will be stronger than the gravitational forces holding it together and the rubble pile will break up. At the moment, only two asteroids have been found that are exceptions to this rule, *2001 OE84* and *2005 UW163* with periods of 0.486 and 1.290 hours respectively, (Chang et al., 2014).

There are several more unconfirmed super-fast rotators (SFR) asteroids reported by Masiero et al. (2009) and Dermawan et al. (2011). Since these objects have low brightness and fast rotation, periods have not yet been accurately determined. Many of the light curves only have few tens of data points.

⁴<http://www.minorplanetcenter.net/cgi-bin/checkneo.cgi>

The fact that we have managed to determine a spin period for this object demonstrates that ULTRACAM is a suitable instrument to use for follow up observations of these other candidates.

The diameter of the asteroid can be estimated by using its absolute magnitude H and an assumption for the albedo p using the formula, adapted from Jewitt et al. (2013)

$$D = \frac{1130}{\sqrt{p}} 10^{-H/5}$$

We have taken $H = 15.2$ from the JPL Small-Body Database ⁵ and assumed an albedo of $p = 0.297$ for a V-type asteroid. This gives a diameter, $D = 1.89$ km for this asteroid.

If our period is correct, this could mean that we have discovered a new member of this rare, fast rotator class of asteroid.

5.2.7 Asteroid: 910 Toruyusa

This asteroid was discovered moving through the field of an ULTRACAM run from

Using the NEOChecker ⁶ tool, the object detected in this run appears to be a well known asteroid, 9108 Toruyusa. The light curve, which lasts approximately 1 hour shows no significant variation and it has not been possible to determine a spin period for this asteroid.

⁵<http://ssd.jpl.nasa.gov/sbdb.cgi>

⁶<http://www.minorplanetcenter.net/cgi-bin/checkneo.cgi>

Chapter 6

Conclusion

The purpose of this project was to create a pipeline that automatically reduces the data contained in the ULTRACAM archive and to make it easily accessible to interested users around the world. To that extent we have succeeded. We wrote an automated pipeline that was able to process all of the data archive and produce web pages allowing the user to browse all of the ULTRACAM runs from first-light in 2004 to the present day.

The automated pipeline enabled the checking of thousands of objects in hundreds of runs by rapidly scanning each light-curve by eye. With this process we found a few dozen objects that exhibited variability that had not yet been documented. Some of these objects are discussed in chapter 5. We were also able to reproduce light-curves for the objects that were the original targets of the run without having prior knowledge about which were the intended targets. For example, we ‘discovered’ GU Mus, an X-ray binary, serendipitously, but after checking the finding charts, realised that this was a known variable and the intended target of the observer. Despite causing premature excitement, this ‘error’ demonstrates that the automated pipeline is a legitimate and useful method of processing the ULTRACAM data. It enables rapid inspection, browsing and sharing of the output of ULTRACAM.

The pipeline lacks robust photometric reduction processes. So, while it is useful for producing light-curves that are easily inspected by eye, it does not produce calibrated magnitudes and colours. The output of the pipeline can be used as input for a final calibration task, but this needs manual intervention. Manual steps are needed to find the relevant runs that contain the standard

stars and to find the appropriate runs containing flat fields and biases to apply to the reduction.

6.1 Current status of the pipeline

Running the reduction of the automated pipeline is simple. As all of the tasks are automated, one single command is all that is needed to reduce a full night's worth of data and prepare all of the web pages used for browsing that night.

A macro script was created to allow the pipeline to make use of the internal distributed computing facility at the University of Warwick called Cluster of Workstations (CoWs). This has enabled us to go back through the entire ULTRACAM archive and process the vast majority of the runs. We have processed 373 nights out of a total of 406.

6.1.1 Areas for immediate improvement

Finding a good astrometric solution for any particular field fails in many cases and, at the moment, we only have astrometric solutions for 10% of the runs. The issues with finding good astrometric solutions are discussed in section 2.3.4. This is an area that could be improved by experimenting with different astrometry packages, such as Scamp from Astromatic.net¹. It is also worth trying to use different reference catalogs for Astrometry.net software if we can find bluer and deeper catalogs to index.

ULTRACAM was designed to be used as a high speed photometric camera and many of the science runs use very short exposure times for runs lasting longer than 30 minutes or so. For these runs, the raw data could contain more than 100,000 frames. The automated pipeline is not optimised to process these high cadence runs. It could be argued that this is not particularly important since these runs have very few objects on them. Actually, many only have two objects in the windowed portion of the CCD, the target and a comparison. In these situations, the traditional pipeline is more suited to the task of reduction, since setting up the apertures is a very quick and easy process. Nevertheless, if we want the automated pipeline to be suited for *all*

¹<http://www.astromatic.net/software/scamp>

of the ULTRACAM data, then some optimisation for these high cadence runs is desired.

For runs with high cadence and for runs with many objects, the volume of data stored in the JSON files is fairly large ~ 300 Mbyte. This taxes the memory management of the browser. With a standard desktop machine (8 Gbyte RAM, 1.6GHz Intel i5 with 6Mbyte cache) this is not a critical problem as all of the run pages load and operate correctly in Firefox and Chrome. Nevertheless, the lag times when loading the data and plotting the light-curves can be several tens of seconds and this makes the user interface sluggish. Javascript relies on internal garbage collection to handle its memory management and it has become clear that in these high load runs, memory leaks will eventually cause the browser to reach a memory maximum. Although it doesn't crash, the lag becomes longer than a minute and the page needs to be reloaded to make it useable. The obvious solution to this problem is to reduce the amount of data that is loaded into the browser. This would require a change in the application architecture to balance the load between the client (browser) and the web server. This approach would increase the amount of code required on the server and is discussed in section 3.2.

In an early effort to limit the amount of data being loaded into the browser, the error estimates on the photometry were dropped from the final output. This means that the browser view of the light-curve does not include any error data and we cannot plot error bars on the light-curves. This is an urgent issue that will be addressed in the next version of the pipeline.

ULTRACAM is used to observe transits of exoplanets with high precision photometry and timing. The exoplanet targets are relatively bright compared to most objects observed by ULTRACAM and, in order to avoid saturating the CCDs, the telescope is deliberately de-focussed to spread the light over many more pixels. This results in star images that resemble discs, rather than the expect PSF of a Moffat profile. The source extraction software, SExtractor, is able to accurately segment these extended images and measure the total flux, but this requires a change to the configuration parameters. This step is not automated by the pipeline at the moment, but this is planned for the next version.

6.2 Future enhancements

If this automated pipeline becomes popular with the ULTRACAM community and is a useful as part of the research then we can make enhancements further in the future. Below we list some of the potential areas for longer term enhancements.

- Photometric calibration: If we are able to improve the astrometric solution finding of the pipeline then we could try to find known photometric standard stars in the fields and use their known fluxes to calibrate the photometry for the run. Another approach might be to search through the comments fields as entered by the principal observer (PI) on the same night looking for the text ‘standard’ and use these as calibration runs.
- Automatic variability detection and light-curve classification: At the moment, we are detecting new variable objects by visual inspection of the light-curves, but this could be automated by using techniques such as those developed for automated surveys. Surveys such as the Catalina Survey are producing catalogs of variable stars found by automatically analysing and classifying millions of stars, (Drake et al., 2014).
- Alternative source extraction and flux measurement: It would be prudent to investigate alternative methods for source extraction and flux measurement. New versions of Astropy include source extraction and photometry libraries and these will be trialed in future versions of the pipeline.
- Automatic tweaking of the source extractor parameters: The pipeline has a fairly ‘brute-force’ approach to reducing the photometry as it has to deal with a very diverse set of input data and therefore relies on ‘best-guess’ values for parameters such as aperture-size, background variability, object-detection thresholds and distance matching. Many of the reductions could be improved by tweaking these parameters to match the specific conditions of the particular run. It is conceivable that these tweaks could be automated by a pipeline that first analyses the run and

tries to classify into a category, such as ‘exoplanet transit, defocused’ and ‘high-cadence, few objects’. These categories could have a pre-defined list of parameters appropriate for each.

- Ability to combine ‘consecutive’ runs: On some occasions, several consecutive runs are taken for the same target. There are also occasions where the same target has been observed multiple times over the course of several nights. It should be possible for the automated pipeline to combine these runs to produce a web page containing a combined set of light-curves for the full set of observations.

6.3 Recommendations for ULTRACAM users

During the processing of the 10 year long ULTRACAM data archive, it has become clear that, by following a few simple guidelines, the observers can aid the automatic photometric reduction of the data.

- Accurate entry of the target info: Finding an accurate WCS solution for each of the fields is improved when we have a world coordinate for the target object. Since the camera does not automatically acquire pointing data from the telescope, this information is dependant on the observer entering this information into the observing log, either as explicit coordinates or by providing an accurate and recognised object identifier that can later be referenced.
- Flat-fields, biases and photometric standards: The automated pipeline is not able to calibrate the photometry using standard stars since, at the moment, there is no reliable way to determine which runs contain measurements of relevant photometric standards. This process is performed manually in the traditional version of the pipeline. This is also true for flat-fields and for bias frames. If a standard practice of entering certain metadata into the observing logs was adopted, then we might be able to automate the application of flat-fields, biases and photometric standards to the automated pipeline.

- Avoid using small windows: Finding astrometric solutions for the fields is made very hard when the images we have are very small and contain very few objects in them. Where possible, the observer should try to include as much of the full field as is reasonable.

Appendix A

Automated software user manual

A.1 Running the pipeline

The automated pipeline is run after the data has been obtained at the telescope. It operates on the raw ULTRACAM data, reducing it and preparing output files that are used for the Web browsing interface. The pipeline can be run anytime after the data has been recorded and is therefore useful for browsing and sharing ULTRACAM observations immediately after performing the observations, or much later, when wishing to reduce the data for a run that exists in the ULTRACAM archive.

A.1.1 Prerequisites

The pipeline is written in the Python programming language and needs a working Python environment. It is currently running on Python version 2.6.9, but has also been tested on version 2.7.6. Within the Python path, the following well-known Python libraries should be installed.

- *Numpy* NumPy is a well-known package for scientific computing with Python. It is available from <http://www.numpy.org>.
- *Astropy* Astropy Project is a community effort to develop a single core package for Astronomy in Python and foster interoperability between

Python astronomy packages. Available at <http://www.astropy.org/>.

- *Matplotlib* matplotlib is a python 2D plotting library it can be used in python scripts, the python and ipython shell. It is available at <http://matplotlib.org/>
- *Image* The Image package contains the Python Imaging Library. This library is used for loading, creating and modifying bitmap images and is used by the pipeline to create the PNG images that are used on the web pages. It can be found at <http://effbot.org/imagingbook/pil-index.htm>.
- *Jinja* The Jinja2 package contains tools for creating and using template files and merging them with dynamic data. The pipeline uses this module to create the HTML files for the web site. It can be found at <http://jinja.pocoo.org/>.

Along with these packages, the following standard Python modules are also used by the pipeline. These packages are nearly always included by default in Python distributions.

- *Math* A module to perform some basic mathematical operations.
- *Argparse* A module that aids the creation of 'command-line parameters' for the scripts in the pipeline.
- *Time* A module for performing time functions and operations.
- *DateTime* A module for formatting and managing date and calendar objects.
- *JSON* A module for reading, writing and parsing JSON-formatted data objects.

The source extraction and flux measurement activity in the pipeline is performed by the third party software, called *SExtractor*, Bertin and Arnouts (2006). SExtractor can be downloaded and installed from <http://www.astromatic.net/software/sextarctor>. If downloading and compiling from the source,

there is a good guide available at: http://wiki.ipb.ac.rs/index.php/SExtractor_installation.

In order to serve the web site, a web server is needed. Fortunately, since the web site consists purely of static files, a simple HTTP server set up is required. For example, an instance of the Apache web server with no server-side add-ons is perfectly adequate.

A.1.2 Installing the Python code

The core code of the automated pipeline exists in the 'git' repository. It is therefore easy to download the code into a local directory, by typing the command: `git clone https://github.com/rashley2712/ucambuilder`. This will create a sub directory called `ucambuilder` which will contain all of the required Python code for running the automated pipeline. Since, it is likely that you will not be running the pipeline from this directory, an important next step is to add this directory to your PATH and PYTHONPATH environment variables.

A.1.3 Config file

It is a good idea to create a separate directory to keep configuration files and temporary files. It is also good practice to run the pipeline from this folder as it will, by default, look in the local directory for the configuration files. In this folder, you should create a few configuration files that you can modify as desired before running the pipeline. In order to get a pre-built set of configuration files, you can use a 'git' repository to create and download a folder with the default set. Typing `git clone https://github.com/rashley2712/ultracam-auto` will download the configuration files into a folder called `ultracam-auto`.

The following files are needed as configuration files:

- *ucambuilder.conf* This is the pipeline's main configuration file and is used to store the parameters that specify where the pipeline should look for the raw data, where it should write the output files, etc. This is discussed in more detail in the next section of this document.
- *default.sex* Is the configuration file that is loaded when SExtractor. This

files contains many parameters that instructs SExtractor on how to perform the source extraction and flux calculation in the image. More details on these parameters can be found in the SExtractor User Manual ¹.

- *default.param* This file lists the columns that we want SExtractor to include in the output catalog. These columns are read by the automated pipeline in order to build up the master catalog. We need SExtractor to output flux measurements, pixel coordinates and measurement flags. Generally these parameters do not need to be edited.
- *default.conv* This file defines the profile of the convolution filter that SExtractor applies to the image before source extraction.

A.1.4 ucambuilder.conf

A sample of the main configuration file for the automated pipeline is shown below.

```
DEBUG      1          # The debug level to be used by the various scripts. Can be 1, 2, 3
SITE_PATH    /storage/astro2/www/phrnaw/sitedev      # The path to the website
ULTRACAMRAW   /storage/astro1/phsaap/ultracam/raw_data      # Path to the Ultracam raw da
WRITE_FITS    0          # Write a fits file or not?
WRITE_JSON    1          # Write a fits file or not?
KEEP_TMP_FILES 0          # Keep the temporary (sextractor) files?
RUNTEMPLATE   /storage/astro2/www/phrnaw/sitedev/sitecode/runxxxx.jinja
DAYTEMPLATE   /storage/astro2/www/phrnaw/sitedev/sitecode/day-xxxx-xx-xx.jinja
MINPIXELDISTANCE 10
RUNINFO       /storage/astro1/phsaap/ultracam/logs/ultra.json
WORKINGDIR     /storage/astro2/phrnaw/workingdir
ROOTURL        http://deneb.astro.warwick.ac.uk/phrnaw/sitedev/
SEX_MAGNITUDE  FLUX_AUTO
COMPARISON_THRESHOLD 95
```

These parameters should be modified before running the pipeline.

- SITE_PATH Specifies where the document root for the web folder is. The pipeline will write the HTML and JSON files to this folder. For each date in the archive, the pipeline will create sub-directories in the format YYYY-MM-DD. A web server should be configured to serve HTTP requests from this folder.

¹<https://www.astromatic.net/pubsfn/software/sextorator/trunk/doc/sextorator.pdf>

- **ULTRACAMRAW** This should point to the root folder where the raw ULTRACAM data is stored. Within this folder there will be sub-folders corresponding to each date on which the ULTRACAM was active. These folders will contain .dat and .xml files corresponding to each run recording during that observing night.
- **ROOTURL** Specifies where the SITE_PATH is accessed in URL space. In other words, this is the URL to the web site that is configured to serve files from the SITE_PATH folder.
- **WORKINGDIR** This is a folder where the pipeline will place temporary working files used to connect the intermediate stages of the pipeline. They are not required for the web version of the output, but they can be useful to debug and diagnose the running of the pipeline to better understand how it has arrived at the final results.
- **RUNINFO** The location of file that contains important meta-data about the ULTRACAM archive, such as run numbers, durations, comments and RA and DEC locations of the field. This data is used to add information to the web pages and to aid the astrometry solution.
- **RUNTEMPLATE & DAYTEMPLATE** Specify the location of the HTML template files that are used to create the final versions that for the web pages.
- **SEX_MAGNITUDE** This parameter instructs the pipeline which value of the flux estimate produced by SExtractor to use for the brightness measurement of the object in the master catalog. Possible values are **FLUX_AUTO**, **FLUX_APER**, **MAG_AUTO** & **MAG_APER**.
- **MINPIXELDISTANCE** The minimum pixel difference used to allow a match of an object across the 3 channels. If the object is more than this distance from the same object in a different channel, then it is treated as a new object.
- **COMPARISON_THRESHOLD** The pipeline tries to find an object that can act as a comparison object for the other objects in the field. It does this by

performing a statistical test for consistency of the differential flux measurements of the top 20 brightest objects in the field and then choosing the brightest of the objects that has the lowest standard deviation of the mean with another object in the field. This comparison is not used during the data reduction portion of the pipeline, but it is automatically selected as the default comparison object when the web browser session is first loaded. This parameter defines the minimum percentage of the run that the object has to persist on in order to be considered as a comparison object.

A.1.5 Producing the output for a particular run

`runbuilder.py`

The quickest way to create the output for a particular run is to use the macro-script, `runbuilder.py`. This script chains together the various stages of the pipeline to produce the HTML and JSON output required to view a particular run. It runs each of the following scripts in the pipeline in turn: `objectdbcreator.py`, `postprocessor.py`, `wcssolver.py`, `mergeobjects.py` and `create_html.py`.

`runbuilder.py` takes the following command line parameters:

- **`runName`** This is a path to the `.xml` and `.dat` for a specific run and is specified in the format `YYYY-MM-DD/runXXX` (for example `2013-07-21/run011`).
- **`-n [n] --numframes [n]`** Specifies the number of frames you would like the script to process. The default is all of the frames in the run. Making this number smaller is useful for running a quick test. For example, `-n100` will run through 100 frames only.
- **`-c [filename] --configfile [filename]`** Allows you to specify an alternative configuration file. By default, the script will look for a file called “ultracam.conf” in the local directory.
- **`-w`** Use this switch to disable the astrometric solution step in the pipeline. This can be used to save time when we have a run where we do not expect to find an astrometric solution. This occurs when the run has only two

or three objects on it and/or the windowed portions of the CCD are very small.

- **-v[version] --version [version]** Specify a unique string to act as an identifier for an alternate version of the output of the pipeline. This can be used if we want to re-run the pipeline, but with different SExtractor parameters and compare the outputs. The resulting web pages will have a URL that has this version string appended.

For more information on the output of `runbuilder.py` see the section below, describing 'Output while running'.

When `runbuilder.py` has completed, it will display the URL to the output run in the terminal window. This URL can then be copied into a web browser's location bar in order to access the results of the reduction.

objectdbcreator.py

This is the most important script in the automated pipeline. It takes the raw image data in the ULTRACAM archive and sends it to SExtractor for processing. Based on the SExtractor output, it compiles and maintains a list of objects across all of the frames and each of the channels. These are given as three output catalogs when the script finishes running.

`objectdbcreator.py` takes the following command line parameters:

- **runName** This is a path to the `.xml` and `.dat` for a specific run and is specified in the format `YYYY-MM-DD/runXXX` (for example `2013-07-21/run011`).
- **-d[n] --debug [n]** Use this parameter to determine how much output you would like to see while the program is running. There are 3 debug levels, `1` is silent (except for errors) and is the default debug level; `2` shows general progress of the pipeline; `3` shows detailed info to help with debugging. Note that the default is 'silent' and therefore, unless there are errors, you will not see anything on the command line and a long run through the data could last an hour or more. It is recommended that you use `-d2` in most cases.
- **-n[n] --numframes [n]** Specifies the number of frames you would like the script to process. The default is all of the frames in the run. Making

this number smaller is useful for running a quick test. For example, `-n100` will run through 100 frames only.

- `-s[n] --startframe [n]` Specifies which frame to start at. The default is frame 1 (the first frame in the run).
- `-c[filename] --configfile [filename]` Allows you to specify an alternative configuration file. By default, the script will look for a file called `ultracam.conf` in the local directory.
- `-C[r,g,b] --channels [r,g,b]` Which channels to operate the pipeline over. By default, the script will process all three channels, namely, r, g, and b. This parameter allows you to specify a subset of these channels. For example, you could omit the processing of the ‘green channel’ by passing in `-Crb`.
- `-p --preview` Specifying this parameter enables a preview window for each frame and each channel using *Matplotlib*. This allows you to see each frame as it is being processed. The colour palettes match the channel, red for r, green for g and blue for b. The preview window also draws a green circle around each object that SExtractor has identified on that particular frame. Warning: This preview slows down the pipeline significantly so should only be used for information and debugging purposes.
- `-t[n] --sleep [n]` Time to pause (in seconds) between the processing of each frame. Useful for debugging in ‘preview’ mode.
- `-r --crop` For ‘preview’ mode, crop the windows to show only the areas that were not masked in the original data. Useful for runs where the windows are fairly small.
- `-v[version] --version [version]` Specify a unique string to act as an identifier for an alternate version of the output of the pipeline. This can be used if we want to re-run the pipeline, but with different SExtractor parameters and compare the outputs. The resulting web pages will have a URL that has this version string appended.

Output while running

If the `--debug` option is left to the default value of 1 then the output will be mostly *silent* with only errors appearing in `stdout`. This mode is designed for use during the running of the pipeline across a complete night where we want to suppress a lot of the output. If you are running `objectdbcreator.py` in standalone mode, then `-d2` is recommended.

The output of the script with `-d2` set looks like this:

```
[10:31:25] 00:10:22 Frame: [1681,1681 87%] MJD:56495.1395466 r:1899 g:1030 b:551
[10:31:27] 00:10:19 Frame: [1682,1682 87%] MJD:56495.1396132 r:1900 g:1030 b:551
[10:31:29] 00:10:17 Frame: [1683,1683 87%] MJD:56495.1396799 r:1900 g:1030 b:551
[10:31:32] 00:10:14 Frame: [1684,1684 87%] MJD:56495.1397465 r:1900 g:1030 b:552
```

Where,

- [10:31:25] is the current time, in HH-MM-SS format;
- [00:10:22] is the estimated time remaining, in HH-MM-SS format, until this stage of the pipeline has completed. ;
- Frame: [1681, 1681 87%] The first number is the absolated frame number being processed (starts at first frame of the run = 1), the second number is the relative frame being processed (different if the start frame was not = 1), and the percentage completed;
- MJD:56495.1395466 is the MJD for this frame;
- r:1899 g:1030 b:551 shows the number of objects being tracked in each of the r, g, b channels.

postprocessor.py

The second stage of the pipeline performs a filtering of the data as described in section 2.3.3. It also creates output catalog files containing an ordered list of the pixel coordinates and fluxes for the brightest objects in the field for each of the r, g, b channels. These can be used as inputs to the Astrometry.net software for the solving of the WCS coordinates for the field.

`postprocessor.py` takes the following command line parameters:

- `runName` This is a path to the `.xml` and `.dat` for a specific run and is specified in the format `YYYY-MM-DD/runXXX` (for example `2013-07-21/run011`).

- **-d[n] --debug [n]** Use this parameter to determine how much output you would like to see while the program is running. There are 3 debug levels, 1 is silent (except for errors).
- **--xyls** Use this switch to create the output catalogs for Astrometry.net. The script will create three catalogs (r, g, b) in FITS format for input into the Astrometry.net software.
- **-c[filename] --configfile [filename]** Allows you to specify an alternative configuration file. By default, the script will look for a file called `ultracam.conf` in the local directory.
- **-v[version] --version [version]** Specify a unique string to act as an identifier for an alternate version of the output of the pipeline. This can be used if we want to re-run the pipeline, but with different SExtractor parameters and compare the outputs. The resulting web pages will have a URL that has this version string appended.

wcssolver.py

This script configures and runs the WCS solving step of the pipeline. It is really just a script that prepares and runs the Astrometry.net package.

`wcssolver.py` takes the following command line parameters:

- **runName** This is a path to the `.xml` and `.dat` for a specific run and is specified in the format `YYYY-MM-DD/runXXX` (for example `2013-07-21/run011`).
- **-d[n] --debug [n]** Use this parameter to determine how much output you would like to see while the program is running. There are 3 debug levels, 1 is silent (except for errors).
- **-f --forcesolve** The script usually checks the `WORKINGDIR` to see if a WCS solution for this field already exists and will skip Astrometry.net if it finds one. Use this switch to force the script to call Astrometry.net even if a solution already exists.
- **-c[filename] --configfile [filename]** Allows you to specify an alternative configuration file. By default, the script will look for a file called `ultracam.conf` in the local directory.

- **-v [version] --version [version]** Specify a unique string to act as an identifier for an alternate version of the output of the pipeline. This can be used if we want to re-run the pipeline, but with different SExtractor parameters and compare the outputs. The resulting web pages will have a URL that has this version string appended.

mergeobjects.py

This script loads the three separate catalogs and performs a merge of the objects into one ‘master’ catalog. It also writes the JSON files that are placed in the web server’s directory, ready to be loaded by the HTML and Javascript for viewing the results.

`mergeobjects.py` takes the following command line parameters:

- **runName** This is a path to the `.xml` and `.dat` for a specific run and is specified in the format `YYYY-MM-DD/runXXX` (for example `2013-07-21/run011`).
- **-d [n] --debug [n]** Use this parameter to determine how much output you would like to see while the program is running. There are 3 debug levels, `1` is silent (except for errors).
- **-c [filename] --configfile [filename]** Allows you to specify an alternative configuration file. By default, the script will look for a file called `ultracam.conf` in the local directory.
- **-v [version] --version [version]** Specify a unique string to act as an identifier for an alternate version of the output of the pipeline. This can be used if we want to re-run the pipeline, but with different SExtractor parameters and compare the outputs. The resulting web pages will have a URL that has this version string appended.

create_html.py

This script takes the run meta-data and merges this with the HTML templates stored in `RUNTEMPLATE` to create the HTML files used for web browsing.

`create_html.py` takes the following command line parameters:

- **runName** This is a path to the .xml and .dat for a specific run and is specified in the format YYYY-MM-DD/runXXX (for example 2013-07-21/run011).
- **-d[n] --debug [n]** Use this parameter to determine how much output you would like to see while the program is running. There are 3 debug levels, 1 is silent (except for errors).
- **-c[filename] --configfile [filename]** Allows you to specify an alternative configuration file. By default, the script will look for a file called ultracam.conf in the local directory.
- **-v[version] --version [version]** Specify a unique string to act as an identifier for an alternate version of the output of the pipeline. This can be used if we want to re-run the pipeline, but with different SExtractor parameters and compare the outputs. The resulting web pages will have a URL that has this version string appended.

A.1.6 Producing the output for a full night's observing

There is a macro Python script called `daybuilder.py` that runs the pipeline on all of the runs in any particular night. The script effectively runs the `runbuilder.py` script for all of the runs found on the date specified. It also produces a summary web page showing all of the runs with descriptions and thumbnails, making navigating the reductions for that night quite easy.

`daybuilder.py` takes the following command line parameters:

- **runDate** The date for the night of interest. Specified in the format YYYY-MM-DD.
- **-d[n] --debug [n]** Use this parameter to determine how much output you would like to see while the program is running. There are 3 debug levels, 1 is silent (except for errors).
- **-c[filename] --configfile [filename]** Allows you to specify an alternative configuration file. By default, the script will look for a file called ultracam.conf in the local directory.

- **-r --buildruns** Build the run output for each run (if no existing output found).
- **-f --forcebuildruns** Force build of each run (even if existing data is found).

A.2 Using the archive

A.2.1 Night summary page

Once the pipeline has been run, the HTML and JSON files will be ready to view through a web. You can access the output of any night of observing by entering a URL into the web browser of the following format, `http://deneb.astro.warwick.ac.uk/phrnaw/sitedev/YYYY-MM-DD/index.html` where you need to substitute the YYYY-MM-DD portion of the URL with the date in question. This will load an HTML page showing you all of the runs that occurred during that night. This list will include acquisition runs, biases and flat fields as well as the science runs. The page shows a thumbnail of each run along with a description of the target object, RA and DEC, run duration and the comments entered by the observer at the telescope. Clicking on the run thumbnail will take you to the web page for that particular run.

A.2.2 Run page

When viewing the web page for a particular run, the user can navigate the light-curves for all of the objects identified by the pipeline. Interaction with the data in the page is through the mouse and keyboard. Many of the actions are triggered by a single key press. There are also tickboxes, and radio buttons that allow the selection of various options.

Page loading process

The page has three main components, HTML, Javascript and JSON. The HTML and Javascript define the page structure and the interactions that can occur, while the JSON contains the reduction data for all of the objects in the run. The JSON datafile is requested from the web server as soon as the

Night of 2013-07-21

Preview	Info	Comments
	run001 ? 0 minutes 0 : 0	
	run002 ? 0 minutes 0 : 0	Low red bias problem. Focal plane mask at -50 px, Agcomp in, Dome lights off
	run003 ? 0 minutes 0 : 0	
	run004 ? 0 minutes 0 : 0	
	run005 ? 0 minutes 0 : 0	More good bias frames
	run006 ? 0 minutes 0 : 0	Blank twilight sky near zenith, telescope spiralling
	run007 SDSS J152419.33+220920.0 4 minutes 15.4053694444 : 22.1555833333	Acquisition. Sky still very bright
	run008 SDSS J152419.33+220920.0 40 minutes 15.4053694444 : 22.1555833333	Science data - primary eclipse. Sky still bright at start. Tweaked focus near start. 0.7" seeing!
	run009 KOI-823 7 minutes 19.7338 : 40.2954444444	Acquisition
	run010 KOI-823 2 hours, 50 minutes 19.7338 : 40.2954444444	Science data - hard to see transit
	run011 KOI-1546 3 hours, 5 minutes 19.9009138889 : 40.6396111111	Science data - nice transit
	run012 HAT-P-23b 1 hour, 56 minutes 20.4082555556 : 16.7621666667	Telescope defocused. Science data. Partial transit observed - bright sky at end
	run013 ? 0 minutes 0 : 0	Biases for SDSS J152419.33+220920.0 data taken tonight - no-clear mode
	run014 ? 0 minutes 0 : 0	Biases for SDSS J152419.33+220920.0 data taken tonight - clear mode
	run015 ? 0 minutes 0 : 0	Biases for SDSS J152419.33+220920.0 data taken tonight - more no-clear mode

Figure A.1: Example of the web page summarising a full night's observing.

HTML and Javascript has finished loading (within 1 second or so). For long runs with many (>20) objects, this file can be quite large and will take some time to download. Depending on the data size and the speed of the internet connection, this download could take up to a minute or more, although in most cases it is usually completed in a few seconds. The status window at the top of the page will give an indication of when this data has completed the download. The request for the JSON data is made using an ‘asynchronous’ call and this means that the page is still working and some interactions can take place, like choosing the base image for example, but since no main data has not yet arrived, it won’t be possible to display a light-curve.

Selecting a base image

The pipeline produces a deep image for each channel (r, g, b) based on stacking all of the individual frames captured by ULTRACAM. The page loads all three images in the background and, by default, displays the ‘green’ channel initially. Switching between these images can be performed by pressing the **r**, **g** and **b** keys on the keyboard. You can also switch these images by selecting either the **Red**, **Green** or **Blue** of the **Base Image** radio options found below the image itself.

Viewing a light-curve

The quickest way to view a light-curve of any particular object is to simply click on the object with the mouse. If the object has been identified by the pipeline and there are sufficient data to display, then a light curve for this object will appear in the panel below the radio buttons and checkboxes on the page. While moving the mouse over the base image, the cursor that displays the (x, y) coordinates (or the (RA, DEC)) coordinates will display a green background if the object underneath it has the required data for a light-curve.

Object markers and Object labels

Each object that has been identified by the pipeline has a unique identifier in the master catalog. In order to view these IDs, you can toggle the object ‘labels’ on and off. This is done by pressing the (letter ‘l’) **1** key, or clicking the

Show labels checkbox. This will draw labels next to each object identified by the pipeline that has photometry available in the channel that is currently selected for the base image. If you want to see each of the identified objects shown as a circle centred on each object, rather than a label, then pressing the **m** key or checking the **Show circles** check box will toggle the drawing of circles of radius 15 pixels around each object.

Selecting a comparison star

During the final stage of the pipeline, a test is performed on the brightest objects to see if any can act as comparison stars for the run. The test takes into account the standard deviation of the star's flux measurements in comparison to the other bright stars in the field. If an object is deemed to be significantly 'constant' enough, then it is flagged as a potential comparison star. When the data has finished loading, this comparison star (if one exists) is selected. The test is performed independantly for each channel. It is possible for the user to select a different comparison star for each channel. This is done by selecting the object and pressing the key **c**. When the object is selected, a black diamond is drawn around the object. When the object is selected as the comparison object, this diamond will change to a square. To indicate that a comparison object for this channel has now been selected, the **Comparison objects** status area will now indicate the Object ID, or the comparison and the light-curve of the comparison object will be displayed below the light-curve of the target object.

Re-scaling the light curve

The vertical axis on the light curve of the target object can be adjusted to adapt to the dynamic range of the flux measurements. By default, when the page is loaded, the light-curve of the target is calculated as the ratio of the flux of the target object to the flux of the comparison object. This value is then also normalised to fill the range from [0to1]. The normalisation step can be disabled by unchecking the **Normalise the chart** check box on the web page. Feedback from early users of this page has suggested moving to a magnitude (\log_{10}) scale by default. This will be implemented in the next iteration of the

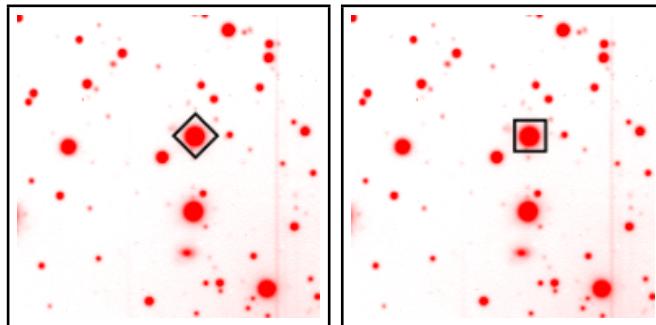


Figure A.2: Selecting an object to act as the comparison. First select the object, then press the **c** key on the keyboard. The cursor will change from a diamond to a square, indicating that this object is now the comparison for this channel ('r' in this case).

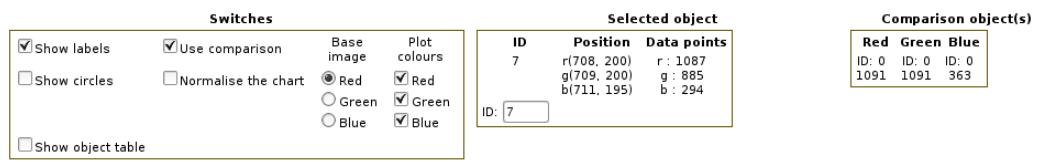


Figure A.3: Sample of the control panel of the run web page. This screenshot shows some of the radio buttons and checkboxes available to the user to manipulate the displayed light-curves.

pipeline. In order to view the light-curve in raw flux measurements, rather than as a ratio with the comparison, then the **Use comparison** check box can be unchecked.

Plotting the position of the object

Once an object is selected, it is possible to produce a chart of the object's (x, y) pixel position during the duration of the run. Pressing the **p** key will produce this plot, which will display a chart of how the object's position varies during the course of the run. An example of this plot is shown in figure 2.8.

Exporting to a CSV file

Once a light-curve is displayed, pressing the **e** key will start the download of a comma-separated value (CSV) file. Most browsers will allow you to rename this file and save to the local disk. The data saved will reflect the light-curve that is currently displayed. ie. If the current light curve is displaying

normalised values, then these are the ones that will be exported in the CSV file.

Bibliography

- C. Aerts, J. Christensen-Dalsgaard, and D. W. Kurtz. *Asteroseismology*. Springer, 2010.
- G. Á. Bakos, J. Lázár, I. Papp, P. Sári, and E. M. Green. System Description and First Light Curves of the Hungarian Automated Telescope, an Autonomous Observatory for Variability Search. *PASP*, 114:974–987, September 2002. doi: 10.1086/342382.
- E. Bertin. Automatic astrometric and photometric calibration with SCAMP. In *ASP Conference Series, Vol. 351, 2006, C. Gabriel, C. Arviset, D. Ponz, and E. Solano, eds*, page 112, 2006.
- E. Bertin and S. Arnouts. SExtractor: Software for source extraction. *Astronomy & Astrophysics Supplement*, 317:393, 2006.
- W. J. Borucki, D. G. Koch, G. Basri, N. Batalha, A. Boss, T. M. Brown, D. Caldwell, J. Christensen-Dalsgaard, W. D. Cochran, E. DeVore, E. W. Dunham, A. K. Dupree, T. N. Gautier, III, J. C. Geary, R. Gilliland, A. Gould, S. B. Howell, J. M. Jenkins, H. Kjeldsen, D. W. Latham, J. J. Lissauer, G. W. Marcy, D. G. Monet, D. Sasselov, J. Tarter, D. Charbonneau, L. Doyle, E. B. Ford, J. Fortney, M. J. Holman, S. Seager, J. H. Steffen, W. F. Welsh, C. Allen, S. T. Bryson, L. Buchhave, H. Chandrasekaran, J. L. Christiansen, D. Ciardi, B. D. Clarke, J. L. Dotson, M. Endl, D. Fischer, F. Fressin, M. Haas, E. Horch, A. Howard, H. Isaacson, J. Kolodziejczak, J. Li, P. MacQueen, S. Meibom, A. Prsa, E. V. Quintana, J. Rowe, W. Sherry, P. Tenenbaum, G. Torres, J. D. Twicken, J. Van Cleve, L. Walkowicz, and H. Wu. Characteristics of Kepler Planetary Candidates

Based on the First Data Set. *Astrophysical Journal*, 728:117, February 2011.
doi: 10.1088/0004-637X/728/2/117.

W. F. Bottke, Jr., D. Vokrouhlický, D. P. Rubincam, and D. Nesvorný. The Yarkovsky and Yorp Effects: Implications for Asteroid Dynamics. *Annual Review of Earth and Planetary Sciences*, 34:157–191, May 2006. doi: 10.1146/annurev.earth.34.031405.125154.

A. Y. Burdanov, V. V. Krushinsky, and A. A. Popov. Astrokit-an efficient program for high-precision differential CCD photometry and search for variable stars. *Astrophysical Bulletin*, 69:368–376, July 2014. doi: 10.1134/S1990341314030122.

J. R. Burton, C. A. Watson, S. P. Littlefair, V. S. Dhillon, N. P. Gibson, T. R. Marsh, and D. Pollacco. z'-band Ground-based Detection of the Secondary Eclipse of WASP-19b. *ApJS*, 201:36, August 2012. doi: 10.1088/0067-0049/201/2/36.

Chan-Kao Chang, Adam Waszczak, Hsing-Wen Lin, Wing-Huen Ip, Thomas A. Prince, Shrinivas R. Kulkarni, Russ Laher, and Jason Surace. A new large super-fast rotator: (335433) 2005 UW163. *Astrophysical Journal*, 391:L35, 2014.

James R. A. Davenport, Andrew C. Becker, Andrew A. West, John J. Bochanski, Suzanne L. Hawley, Jon Holtzman, Heather C. Gunning, Eric J. Hilton, Ferah A. Munshi, and Meagan Albright. The very short period m dwarf binary sdss j001641000925. *The Astrophysical Journal*, 764(1):62, 2013. URL <http://stacks.iop.org/0004-637X/764/i=1/a=62>.

B. Dermawan, T. Nakamura, and F. Yoshida. Subaru lightcurve observations of sub-km-sized main-belt asteroids. *Publ. Astron. Soc. Japan*, 63:555–576, 2011.

V. S. Dhillon, T. R. Marsh, M. J. Stevenson, D. C. Atkinson, P. Kerry, P. T. Peacocke, A. J. A. Vick, S. M. Beard, D. J. Ives, D. W. Lunney, S. A. McLay, C. J. Tierney, J. Kelly, S. P. Littlefair, R. Nicholson, R. Pashley, E. T. Harlaftis, and K. O'Brien. ULTRACAM: an ultrafast, triple-beam

- CCD camera for high-speed astrophysics. *Monthly Notices of the Royal Astronomical Society*, 378:825–840, July 2007. doi: 10.1111/j.1365-2966.2007.11881.x.
- A. J. Drake, M. J. Graham, S. G. Djorgovski, M. Catelan, A. A. Mahabal, G. Torrealba, D. García-Álvarez, C. Donalek, J. L. Prieto, R. Williams, S. Larson, E. Christensen, V. Belokurov, S. E. Koposov, E. Beshore, A. Boattini, A. Gibbs, R. Hill, R. Kowalski, J. Johnson, and F. Shelly. The Catalina Surveys Periodic Variable Star Catalog. *ApJS*, 213:9, July 2014. doi: 10.1088/0067-0049/213/1/9.
- P. P. Eggleton and A. A. Tokovinin. A catalogue of multiplicity among bright stellar systems. *Monthly Notices of the Royal Astronomical Society*, 389: 869–879, 2008.
- Gerald Handler. Asteroseismology. In D. Oswalt, T. and M. A. Barstow, editors, *Planets, Stars and Stellar Systems, Volume 4: Stellar Structure and Evolution*, chapter 4, pages 208–239. Springer Science, 2013.
- D. W. Hogg and D. Lang. Astrometry.net. <http://www.astrometry.net>, 2012.
- Seth A. Jacobson, Francesco Marzari, Alessandro Rossi, Daniel J. Scheeres, and Donald R. Davis. Effect of rotational disruption on the size-frequency distribution of the main belt asteroid population. *Monthly Notices of the Royal Astronomical Society: Letters*, 439:L94, 2014.
- David Jewitt, Masateru Ishiguro, and Jessica Agarwal. Large particles in active asteroid P/2010 A2. *Astrophysical Journal*, 764:L5, 2013.
- A. F. Kowalski, M. Mathioudakis, S. L. Hawley, E. J. Hilton, V. S. Dhillon, T. R. Marsh, and C. M. Copperwheat. White Light Flare Continuum Observations with ULTRACAM. In C. Johns-Krull, M. K. Browning, and A. A. West, editors, *16th Cambridge Workshop on Cool Stars, Stellar Systems, and the Sun*, volume 448 of *Astronomical Society of the Pacific Conference Series*, page 1157, December 2011.
- R. G. Kron. Photometry of a complete sample of faint galaxies. *Astrophysical Journal Supplement Series*, 43:305, 1980.

- L. B. Lucy. The Structure of Contact Binaries. *Astrophysical Journal*, 151: 1123, March 1968. doi: 10.1086/149510.
- Joseph Masiero, Robert Jedicke, Josef Ďurech, Stephen Gwyn, Larry Denneau, and Jeff Larsen. The thousand asteroid light curve survey. *Icarus*, 204:145, 2009.
- A. F. J. Moffat. A Theoretical Investigation of Focal Stellar Images in the Photographic Emulsion and Application to Photographic Photometry. *A&A*, 3: 455, December 1969.
- Isabella Paganano. Stellar activity. In D. Oswalt, T. and M. A. Barstow, editors, *Planets, Stars and Stellar Systems, Volume 4: Stellar Structure and Evolution*, chapter 10, pages 588–547. Springer Science, 2013.
- E. S. Phinney and S. R. Kulkarni. Binary and millisecond pulsars. *Annual Review of Astronomy and Astrophysics*, 32(1):591–639, 1994. doi: 10.1146/annurev.aa.32.090194.003111. URL <http://dx.doi.org/10.1146/annurev.aa.32.090194.003111>.
- D. L. Pollacco, I. Skillen, A. Collier Cameron, D. J. Christian, C. Hellier, J. Irwin, T. A. Lister, R. A. Street, R. G. West, D. R. Anderson, W. I. Clarkson, H. Deeg, B. Enoch, A. Evans, A. Fitzsimmons, C. A. Haswell, S. Hodgkin, K. Horne, S. R. Kane, F. P. Keenan, P. F. L. Maxted, A. J. Norton, J. Osborne, N. R. Parley, R. S. I. Ryans, B. Smalley, P. J. Wheatley, and D. M. Wilson. The WASP Project and the SuperWASP Cameras. *PASP*, 118:1407–1418, October 2006. doi: 10.1086/508556.
- T. Shahbaz, V. S. Dhillon, T. R. Marsh, J. Casares, C. Zurita, and P. A. Charles. Observations of the quiescent X-ray transients GRS 1124-684 (=GUMus) and Cen X-4 (=V822Cen) taken with ULTRACAM on the VLT. *Monthly Notices of the Royal Astronomical Society*, 403:2167–2175, April 2010. doi: 10.1111/j.1365-2966.2010.16262.x.
- David L. Shupe and Richard N. Hook. The SIP convention for representing distortion in fits imageheaders, 2008. URL http://fits.gsfc.nasa.gov/registry/sip/SIP_distortion_v1_0.pdf.

- Ovidiu Vaduvescu. Observing near earth asteroids with a small telescope. *Romanian Astronomical Journal*, 2005.
- B. Warner. *Cataclysmic Variable Stars*. Cambridge University Press, September 2003.
- B. D. Warner, A. W. Harris, and P. Pravec. The asteroid lightcurve database. *Icarus*, 202:134–146, July 2009. doi: 10.1016/j.icarus.2009.02.003.
- N. J. Wilsey and M. M. Beaky. Revisiting the O’Connell Effect in Eclipsing Binary Systems. *Society for Astronomical Sciences Annual Symposium*, 28: 107, May 2009.
- D.E. Winget and S.O. Kepler. Pulsating white dwarf stars and precision asteroseismology. *Annual Review of Astronomy and Astrophysics*, 46(1):157–199, 2008. doi: 10.1146/annurev.astro.46.060407.145250.
- J. N. Winn. Transits and Occultations. *ArXiv e-prints*, January 2010.