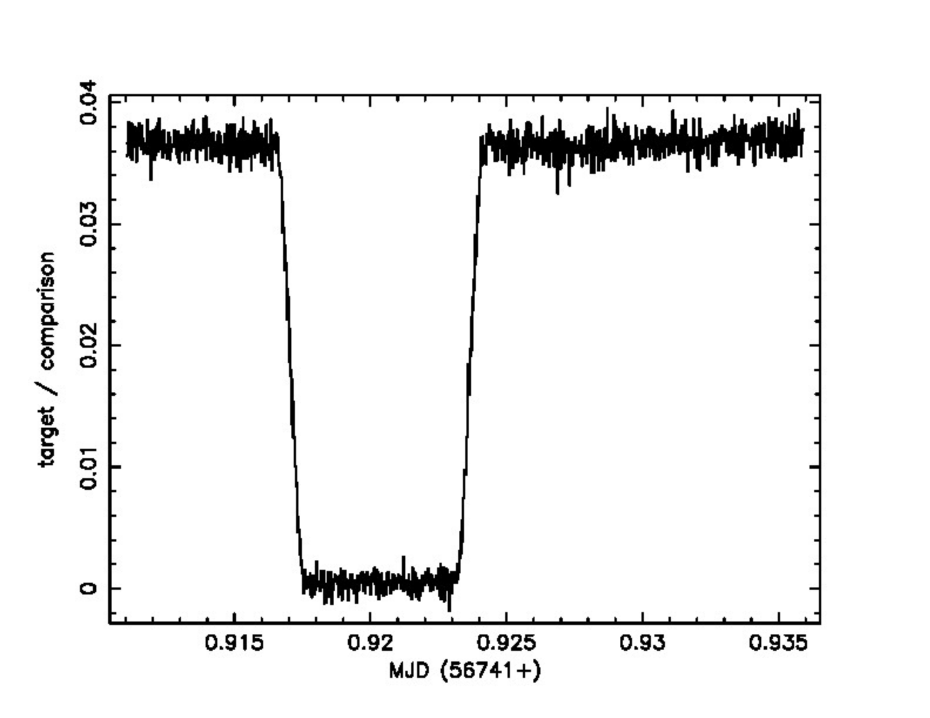
How to reduce ULTRASPEC data

Tom Marsh
University of Warwick
t.r.marsh@warwick.ac.uk



Talk outline

1) Installation hints
2) Principles of photometric data reduction
3) The ULTRACAM / ULTRASPEC pipeline
4) Trouble-shooting

Installation

See:

www.astro.warwick.ac.uk/people/marsh/software

You want the C++ package “ULTRACAM”

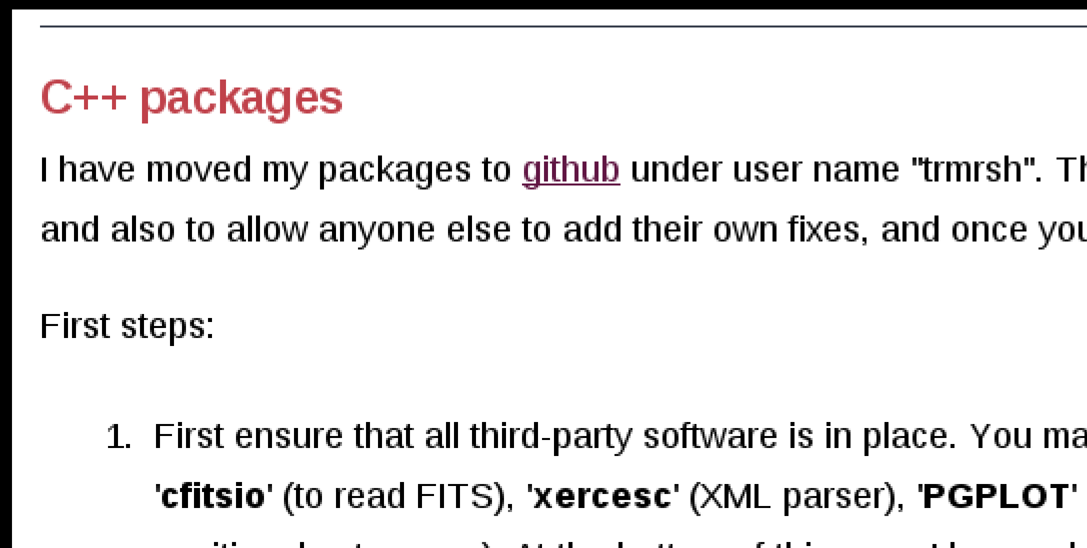
Linux: ✓✓
Mac: ✓ – but expect a rough ride
Windows: ✗ – convert your data to FITS! (see later)

If you know Python and like to do your own thing then you might also investigate trm.ultracam

Installation tips

Read & follow the instructions to the letter:

www.astro.warwick.ac.uk/people/marsh/software


C++ packages

I have moved my packages to github under user name "trmrsh". Thi
and also to allow anyone else to add their own fixes, and once you

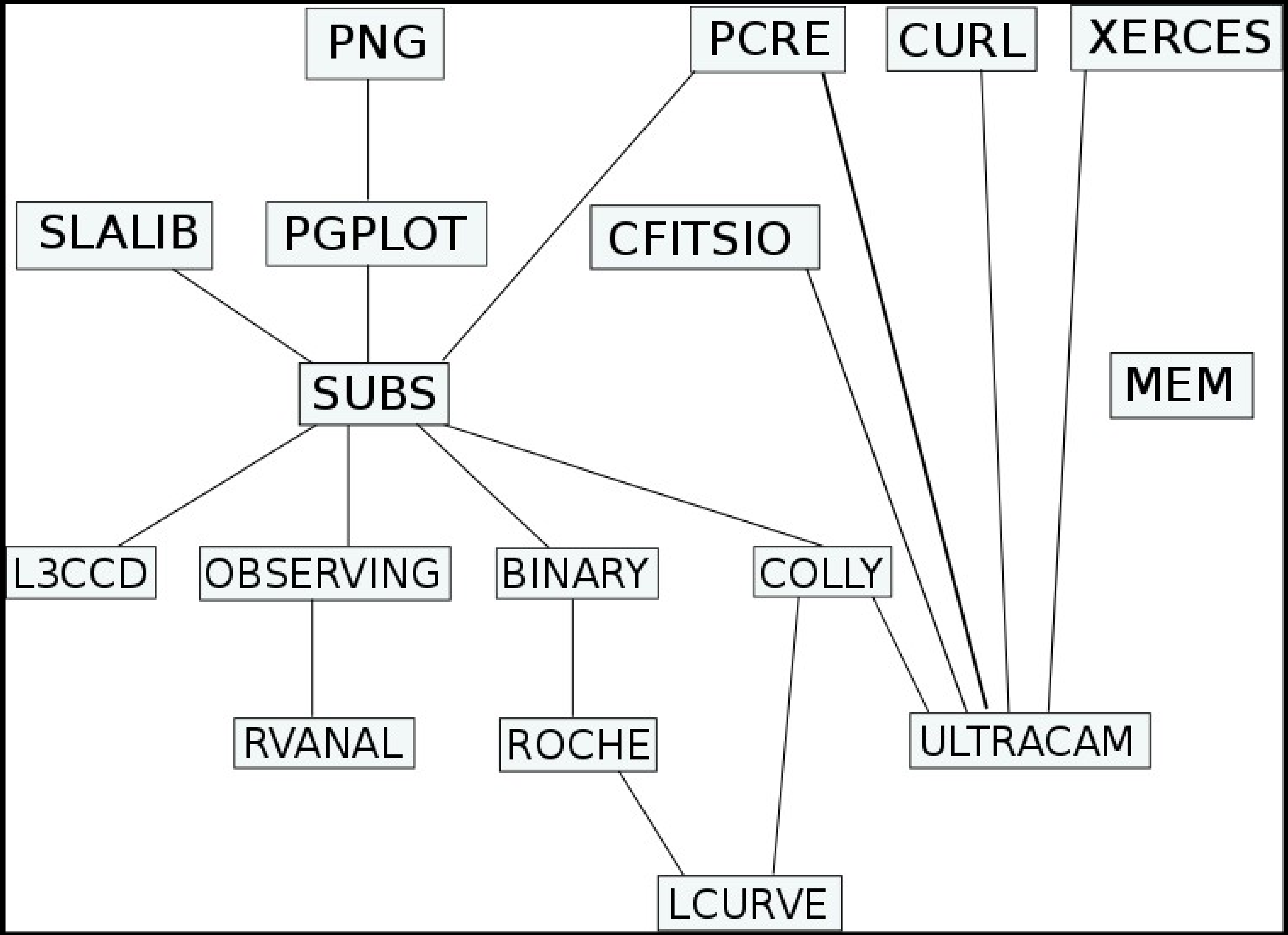
First steps:

1. First ensure that all third-party software is in place. You may
'cfitsio' (to read FITS), **'xercesc'** (XML parser), **'PGPLOT'** (

etc, etc, etc

Top secret package “slalib”:

CENSORED



PNG, PCRE, CURL, XERCES → SLALIB, PGPLOT, CFITSIO → SUBS → COLLY → ULTRACAM

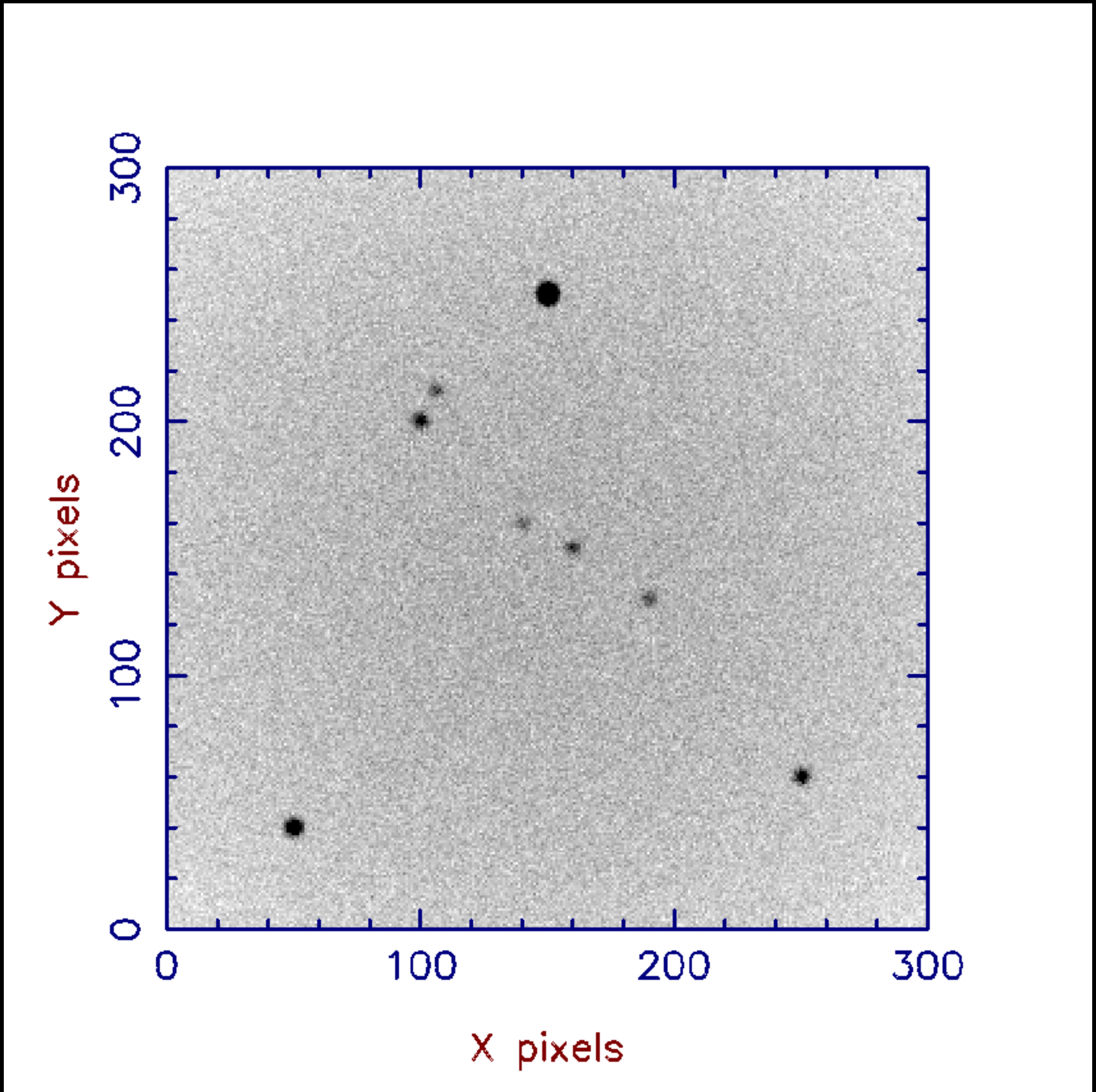
Third party *General astro* *ULTRASPEC specific*

Talk outline

1) Installation hints
2) Principles of photometric data reduction
3) The ULTRACAM / ULTRASPEC pipeline
4) Trouble-shooting

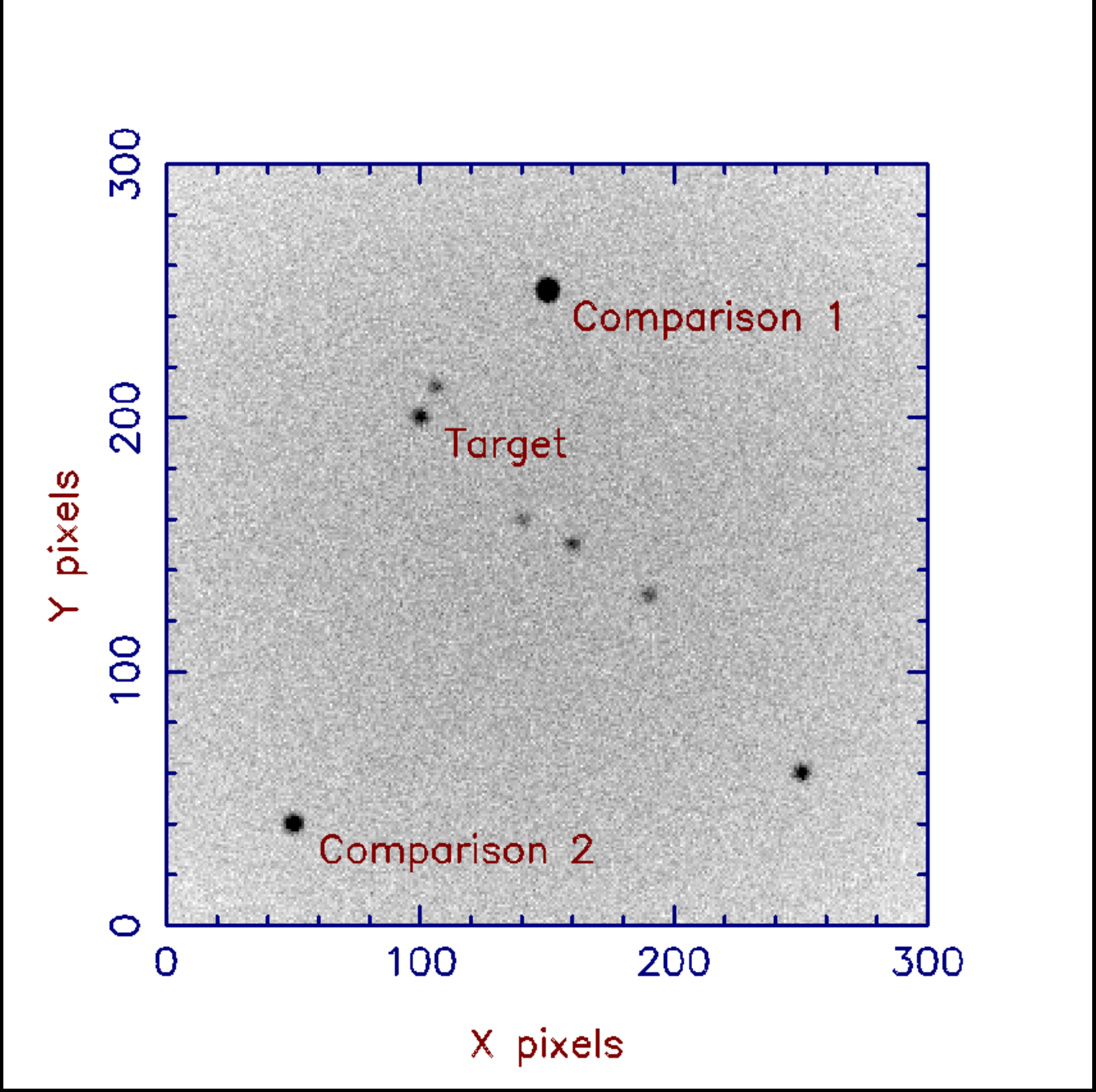
Aperture Photometry

Right: an (artificial) CCD image.



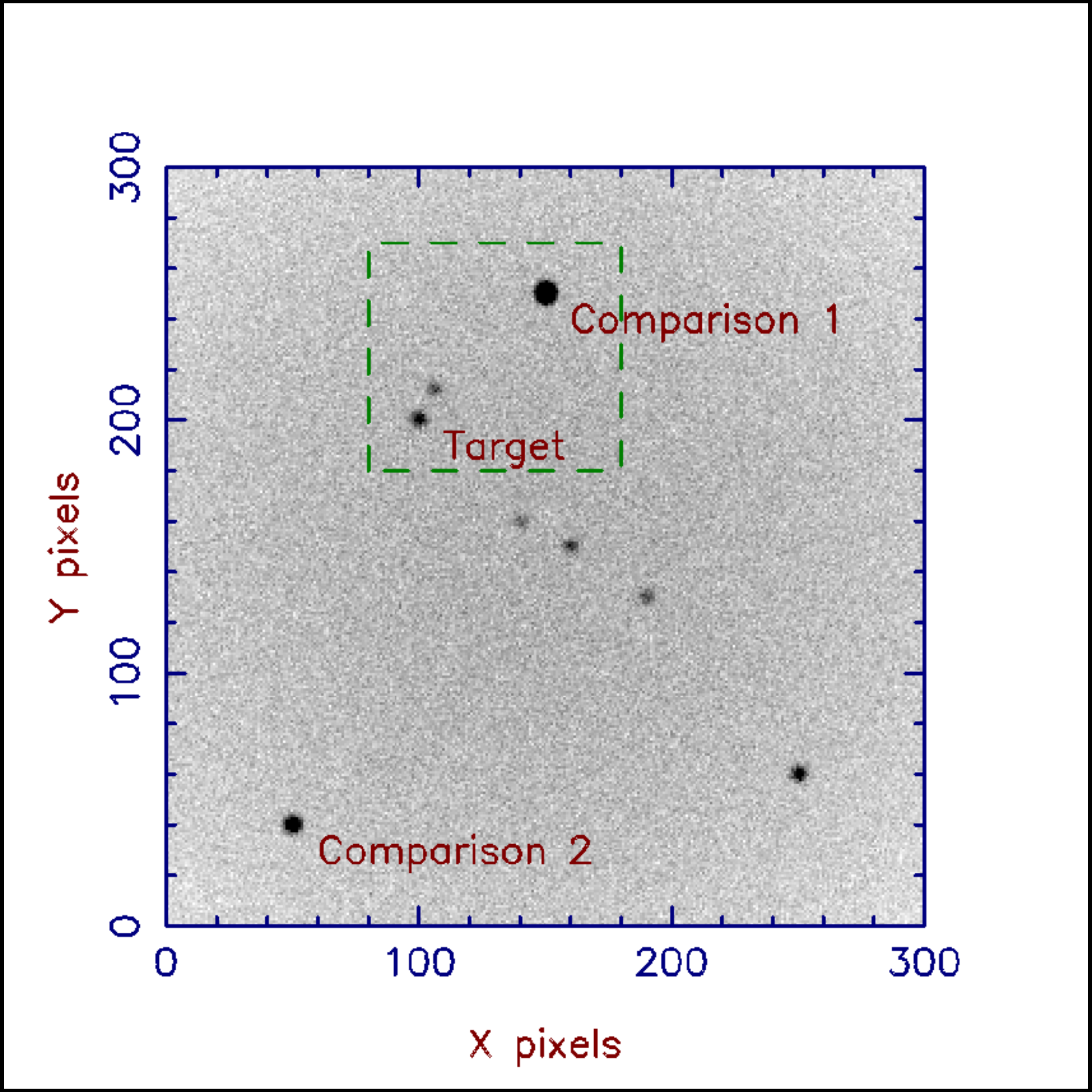
Aperture Photometry

First task: identify target and comparison stars



Aperture Photometry

First task: identify target and comparison stars

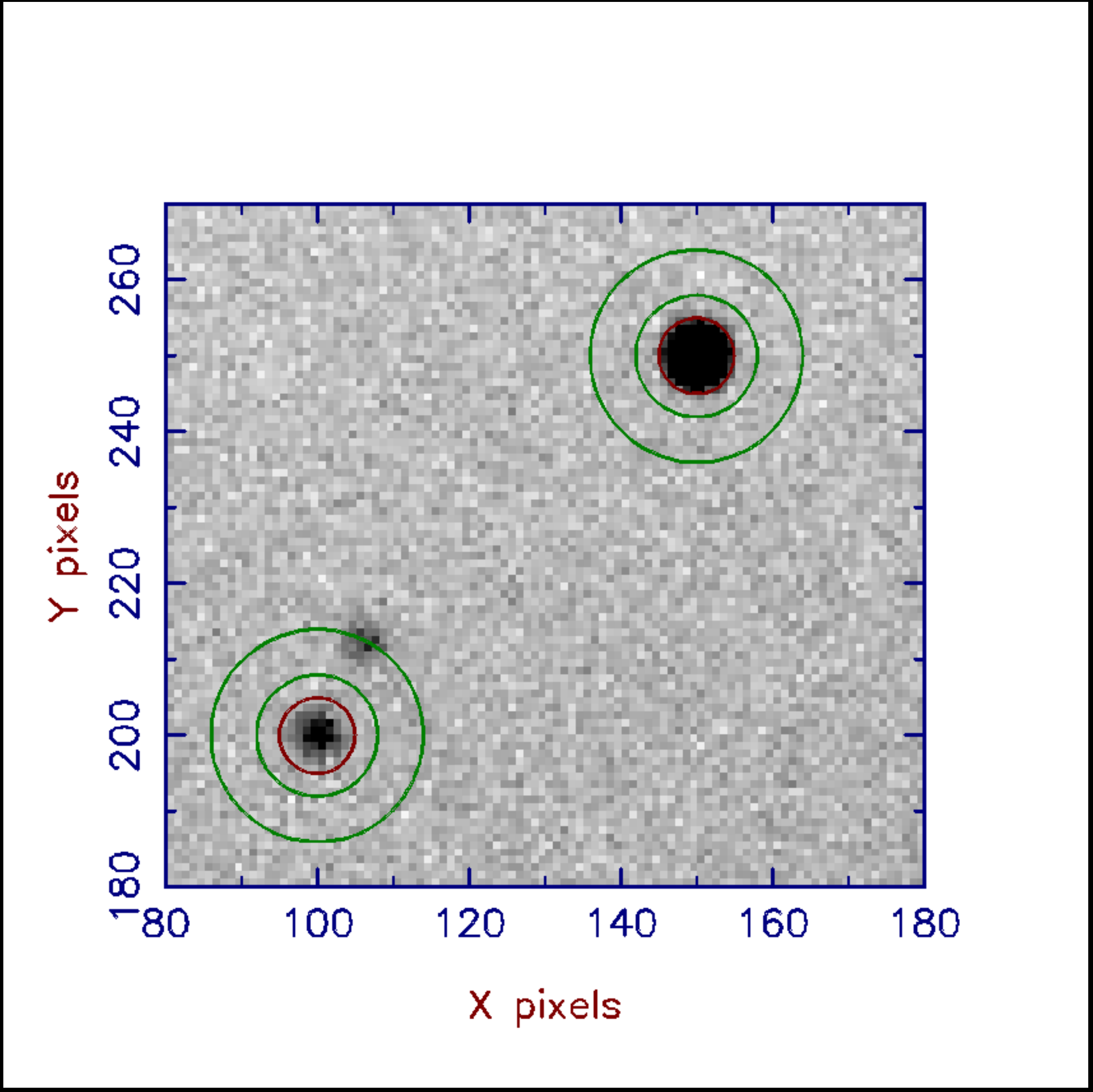


Aperture Photometry

Second task: set object and sky apertures.

NB. Use same sizes for all stars!

(ULTRASPEC pipeline does this automatically.)



Calibration

If $I(x,y,t)$ is the ideal image of the stars & sky, and $O(x,y,t)$ is what you observe [functions of position on the CCD and time], then ideally (but approximately):

$$O(x,y,t) = T(t)\, F(x,y)\, I(x,y,t) + B(x,y) + D(x,y)\, E$$

Where $F(x,y)$ is the “flat field”, $B(x,y)$ is the “bias”, $D(x,y)$ is the “dark count rate” [functions of position only], E is the exposure time, and $T(t)$ is the transmission [function of time only].

Biases

$$O(x,y,t) = T(t)\, F(x,y)\, I(x,y,t) + B(x,y) + D(x,y)\, E.$$

Set $E = 0$, $I = 0$ (zero length exposure, no light) then (since O will now not vary with t):

$$B(x,y) = O(x,y).$$

Such exposures are called “bias frames” or just “biases” for short.

Darks

$$O(x,y,t) = T(t)\, F(x,y)\, I(x,y,t) + B(x,y) + D(x,y)\, E.$$

Having measured B, set $I = 0$ (no light) then:

$$D(x,y) = (\, O(x,y) - B(x,y)\,) / E$$

Such exposures are called “dark frames” or just “darks” for short.

(For ULTRASPEC the dark count rate is low (< 10 counts / hour), and I will ignore it from now on.)

Flats

$$O(x,y,t) = T(t)\, F(x,y)\, I(x,y,t) + B(x,y).$$

Having measured B, observe a uniform source (e.g. twilight clear sky), so $I = I(t)$. Then:

$$F(x,y) = (\, O(x,y,t) - B(x,y)\,) \,/\, T(t)\, I(t)$$

We don't know $T(t)\, I(t)$ so we usually set it to make $\langle F \rangle = 1$. Such “flat fields” thus correct for *relative* sensitivity variations, e.g. (x1,y1) vs (x2,y2).

Reduction: first steps

The first steps in reduction are encapsulated in this expression:

$$O'(x,y,t) = (O(x,y,t) – B(x,y)) / F(x,y)$$

Which breaks down into a "debiassing" step

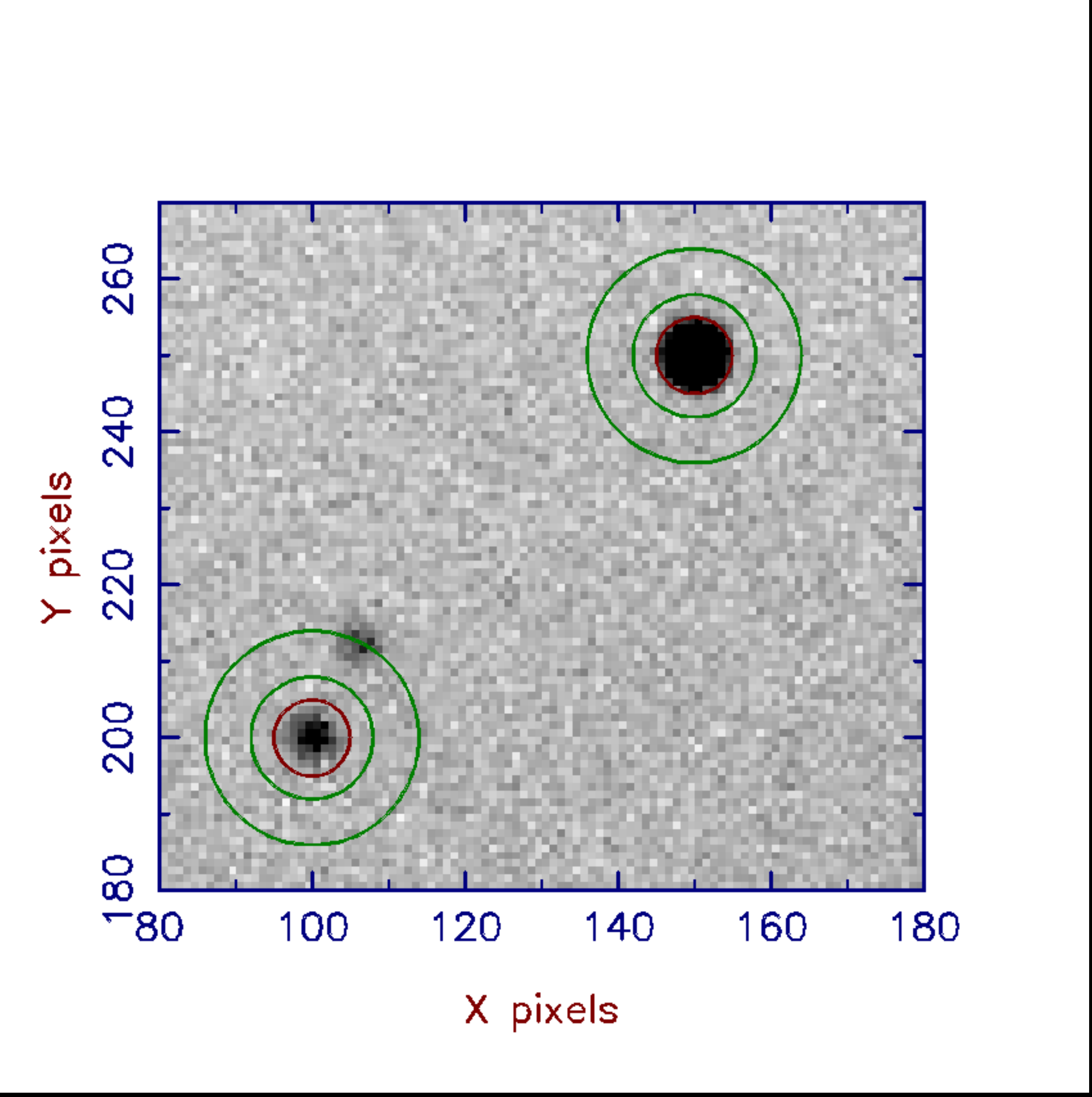
"$O(x,y) – B(x,y)$",

and a "flatfielding" step

"$/ F(x,y)$".

Extraction

In “aperture photometry” we extract fluxes by summing all counts in the inner circle (“aperture”) minus a background sky estimated from the region between the two outer circles.



… and (almost) finally

We end up with

$$F'_T = T(t)\, F_T$$

$$F'_C = T(t)\, F_C$$

F'_T, F'_C bias-subtracted, flat-fielded, sky-background-subtracted, summed-over-aperture fluxes.

$T(t)$, the transmission, varies because of absorption along the changing path through the atmosphere, dust and clouds.

Remove by <u>division</u>: $F''_T = F'_T \,/\, F'_C = F_T \,/\, F_C$

Talk outline

1) Installation hints
2) Principles of photometric data reduction
3) The ULTRACAM / ULTRASPEC pipeline
4) Trouble-shooting

The ULTRASPEC pipeline

We call the software written to look at and reduce ULTRACAM & ULTRASPEC data the “pipeline”.

It consists of ~90 standalone programs. See:

http://deneb.astro.warwick.ac.uk/phsaap/software/ultracam

In practice, only ~15 of these are generally useful. e.g. rtplot, setaper, reduce, grab

Typically start the software by typing 'ultracam' (depends on precise installation)

A first look at your data – rtplot

rtplot – “real time plot”

```
demos/nnser>
demos/nnser> rtplot run042 pause=0.2 iset=p
FIRST - first file to access (0 for last) [1]:
TRIM - trim junk lower rows from windows? [no]:
parseXML warning: data status = WARNING
parseXML warning: version >= 120813; will assume 0.1 millisecond time exposure delay steps, valid as of August 2012
parseXML warning: ULTRASPEC file
parseXML warning: version number = 140331
BIAS - do you want to subtract a bias frame before plotting? [yes]:
BIASFRAME - name of bias frame [bias]:
THRESHOLD - do you want to threshold to get 0 or 1 photons/pix? [no]:
NACCUM - number of frames to accumulate before displaying [1]:
XLEFT - left X limit of plot [0.5]:
XRIGHT - right X limit of plot [1056.5]:
```

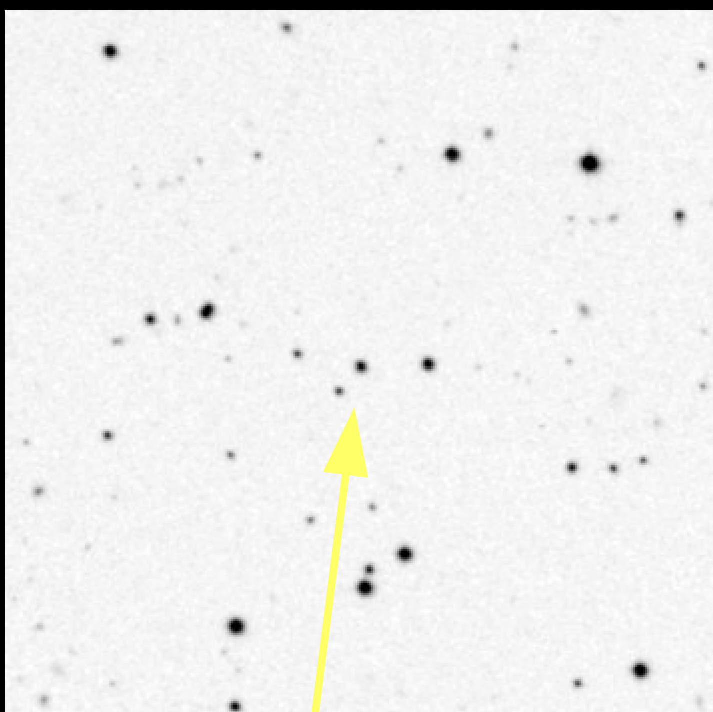
- Command-line oriented: parameters can be specified on the command line.
- Others prompted for: <CR> keeps default e.g. “[1056.5]”
- Once run can re-run with “rtplot \\” ←keeps all default values

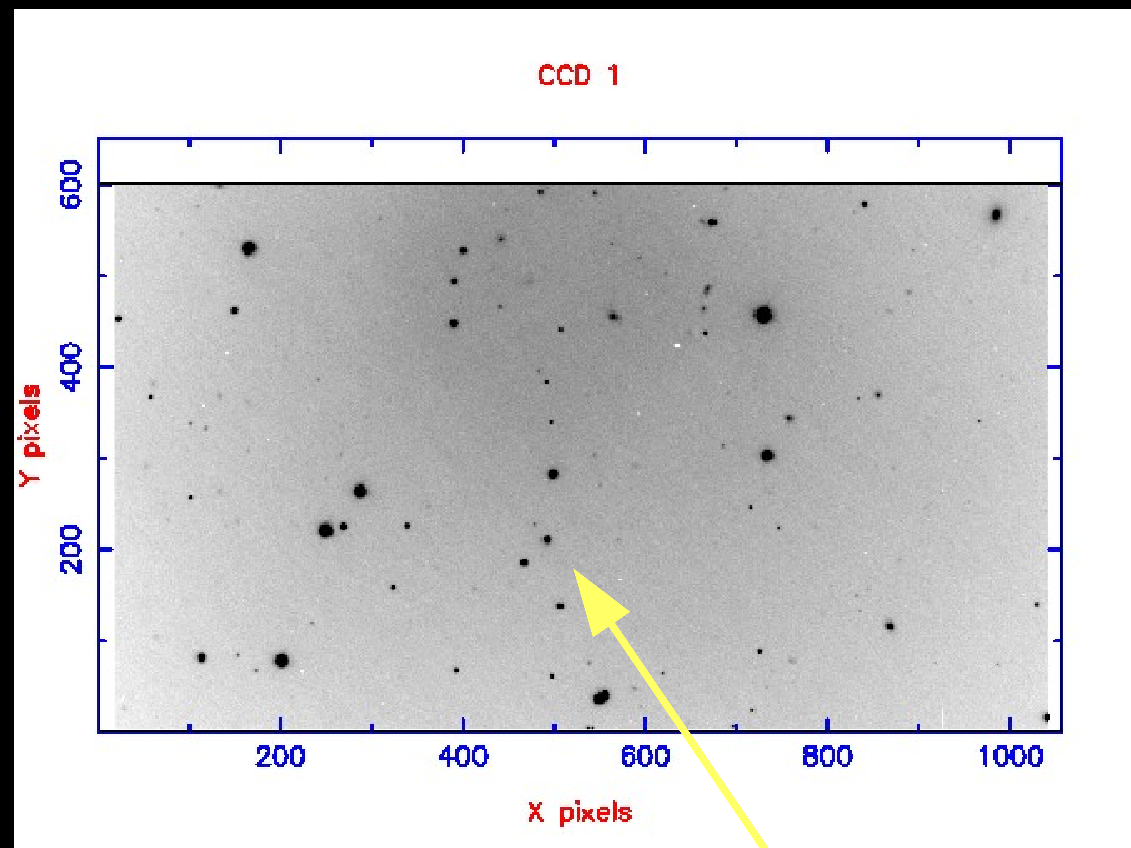
… time for a demo ...

Object identification

Compare ULTRASPEC image with sky survey: useful routine averun, e.g. “averun run002 3 50”

DSS





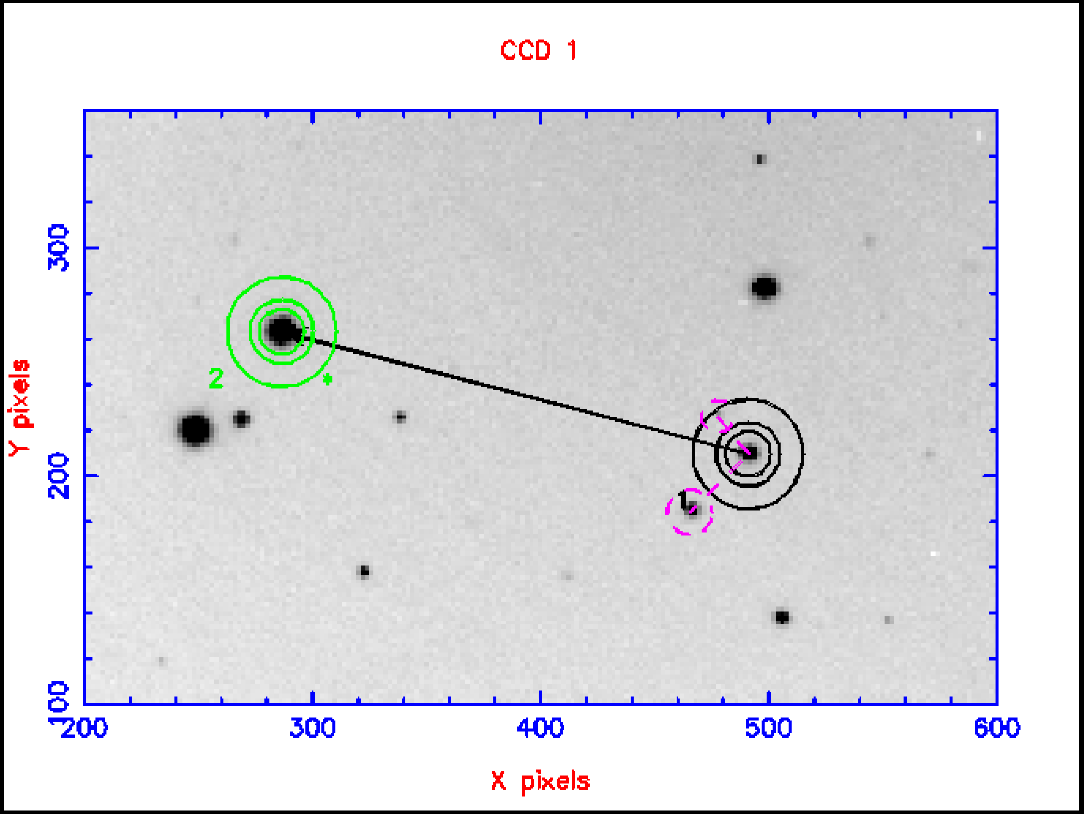
NN Ser

NN Ser

Defining apertures

Photometry apertures are defined with setaper

- Target (NN Ser) is “linked” to comparison
- Two nearby stars have been “masked”



… time for a demo ...

Flux extraction

Flux extraction is carried out with reduce

Philosophy:

- Must be able to keep up with fast frame rates.
- Therefore minimal I/O: single load of bias, flat, dark. Load data once per frame.
- Single program to subtract bias, flat field and extract fluxes.
- Lots of parameters: most loaded from a file: “reduce.red”

reduce.red [section of]

```
# Aperture parameters

aperture_file              = aper                # file of software apertures for each CCD
aperture_reposition_mode   = reference_plus_tweak  # relocation method: static, individual, individual_plus_tweak, re
aperture_positions_stable  = yes        # whether to weight search towards last position or not
aperture_search_half_width = 35         # half width of box for initial search around last position, unbinned pixels
aperture_search_fwhm       = 14.0       # FWHM for gaussian used to locate objects, unbinned pixels
aperture_search_max_shift  = 24.0       # maximum allowed shift in object positions, frame to frame, unbinned pixels
aperture_tweak_half_width  = 20         # half width of box for tweak after a search, unbinned pixels
aperture_tweak_fwhm        = 8.0        # FWHM for gaussian used in tweaking object position, unbinned pixels
aperture_tweak_max_shift   = 4.0        # maximum allowed shift when tweaking object positions, unbinned pixels.
aperture_twopass           = no         # twopasses to fit relative position drift or not
aperture_twopass_counts    = 20.0       # minimum number of counts for a position to be included in the fits
aperture_twopass_npoly     = 3          # number of polynomial coefficients for the fits
aperture_twopass_sigma     = 3.0        # mrejection threshold, multiple of RMS, for fits

# Extraction control parameters. One per line with the format nccd
#
# aperture_type extraction_method star_scale star_min star_max inner_sky_scale
# inner_sky_min inner_sky_max outer_sky_scale outer_sky_min outer_sky_max
# aperture_type can be 'fixed' or 'variable' (i.e. fixed or variable radii);
# extraction_method can be 'normal' or 'optimal'.  The aperture radius scale
# factors are multiples of the FWHM so if either of 'variable' or 'optimal' are
# set, profile fitting will be carried out. The minimum and maximum ranges
# allow you to control the sky aperture radii, for instance to avoid a nearby
# bright star.

extraction_control         = 1 variable normal 1.7 6.0 30.0 2.5 17.0 35.0 3.0 20.0 40.0
```

Outer aperture scale factors

Inner aperture scale factor

Scale apertures with seeing

reduce demos

1) NN Ser: 3.1h eclipsing, detached white dwarf+M dwarf binary (with planets!). Look out for extremely deep eclipse of white dwarf.

2) ASASSN-14ag: 1.4h eclipsing cataclysmic variable star. Look out for rise of flux as spot at edge of disc comes into view, followed by eclipse of the white dwarf then the spot.

Observing with reduce

During observing reduce can keep up with the frames as they come in. This allows you to

- Monitor conditions (seeing, transparency)
- Get a good idea of data quality
- Decide whether your target is in the right state
- Optimise the telescope focus (which is time-variable)

Trouble-shooting

If you use Windows, you should write your data to FITS. This is best done with a Python script called “tofits.py” available at the telescope.

Some pipeline commands have hidden parameters. Specify “prompt” on the command line to reveal them.

You can stop commands with ctrl-C, but sometimes this confuses the plot windows causing problems with setaper: destroy the windows if this happens.

Four key resources

- Pipeline docs: deneb.astro.warwick.ac.uk/phsaap/software/ultracam
- Everything on ULTRASPEC at the TNT: www.vikdhillon.staff.shef.ac.uk/ultracsepc/ultraspec_tnt.html
- Pipeline command user input: deneb.astro.warwick.ac.uk/phsaap/software/ultracam/html/UserInput.html
- ULTRASPEC finding chart tool: www.slittlefair.staff.shef.ac.uk/usfinder