From raw images to lightcurves: *how to make sense of your data*

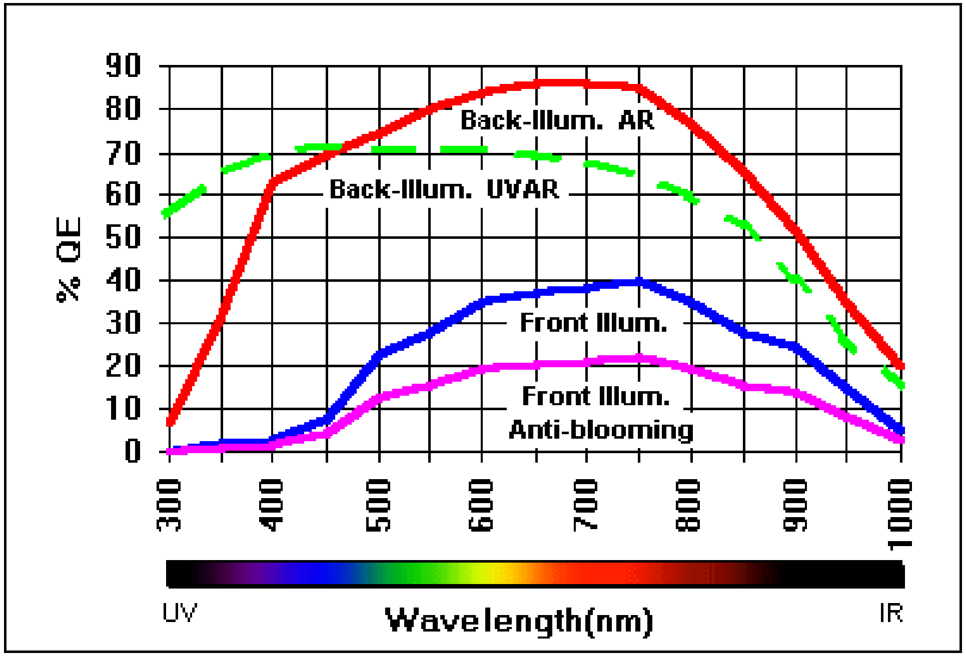
Yiannis Tsapras
LCOGT.net Queen Mary University of London

Talk Outline

- CCD performance
- Sources of noise
- Calibrating the images
- Performing the photometry
- Difference Imaging
- Constructing the lightcurve

CCD performance

- Sensitivity described by it's quantum efficiency (q.e.).
  - Perfect detector → 1 e- for each photon.
  - In practice q.e. depends on wavelength and is maximized in the red, around 600-700nm.
    - Typical values in that range are ~80-90% q.e.

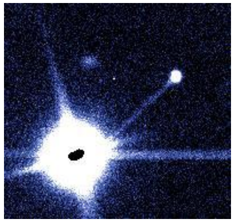


Converting photons to digital counts

- Light sensitive pixels convert photons to e-.
  - e- read and converted into numerical values [ADU].
- The CCD gain is a scale factor that converts the number of e- to ADU. G = (nr e- per pix) / (ADU per pix).

Converting photons to digital counts

- Light sensitive pixels convert photons to e-.
  - e- read and converted into numerical values [ADU].
- The CCD gain is a scale factor that converts the number of e- to ADU. G = (nr e- per pix) / (ADU per pix).
- Pixels “bloom” when charge capacity is exceeded.



  - 16-bit A/D CCD unit → the maximum value is 2^16 = 65536.
  - Gain set so that the maximum possible charge packet corresponds to this maximum value.
  - Anti-blooming systems are available but reduce sensitivity by ~1/2.
  - Easier to take multiple short exposures and stack them.

Reading the CCD

- Reading consists in moving charges from different pixels.
  - Detectors possess an extra line known as the line register.
  - Each line of pixels is transferred to the line register and read out.

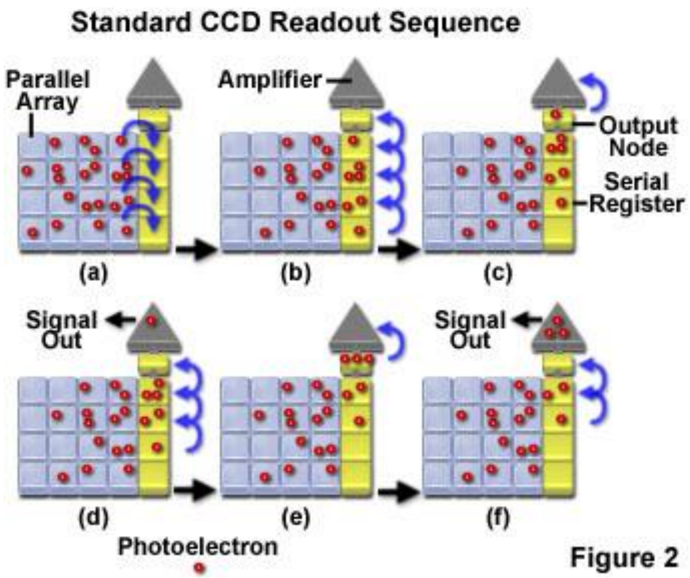


Figure 2

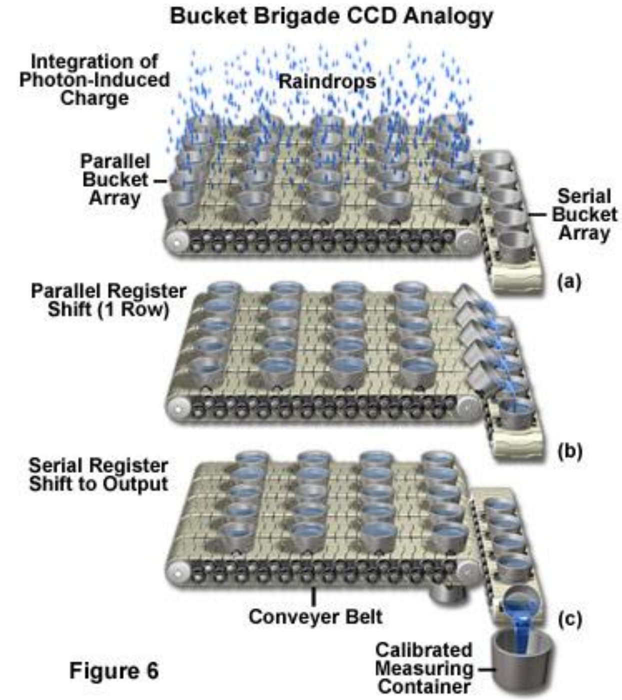
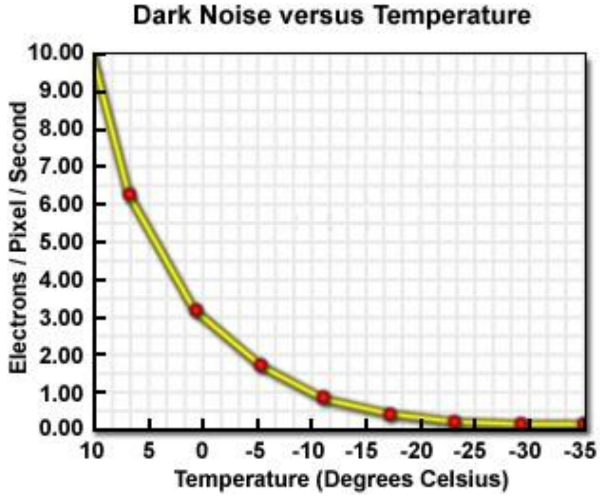


Figure 6

Sources of noise: The dark (thermal) current

- Even when no incoming light → e- still generated due to thermal fluctuations in silicon.
- But:
  - Reproducible phenomenon: at same temperature each pixel → same nr of e- per unit time.
  - The dark current depends on temperature → intensity decreases when detector is cooled.



Sources of noise: The dark (thermal) current

- Even when no incoming light → e- still generated due to thermal fluctuations in silicon.
- But:
  - Reproducible phenomenon: at same temperature each pixel → same nr of e- per unit time.
  - The dark current depends on temperature → intensity decreases when detector is cooled.
- To account for dark current → obtain a second CCD exposure of equal length with shutter closed and at same temperature → subtract from the original image.
- Becomes negligible at very low temperatures.

Sources of noise: Readout noise

- Is a consequence of the imperfection of the CCD itself.
  - Electric charge of each pixel transferred to the amplifier → Voltage induced by charge is measured.
  - Amplifier can't do a perfect job of measuring voltage: correct average value + some scatter.
  - Readout noise = measure of this scatter around the average value.
  - Independent of exposure time.

Sources of noise: Readout noise

- Is a consequence of the imperfection of the CCD itself.
  - Electric charge of each pixel transferred to the amplifier → Voltage induced by charge is measured.
  - Amplifier can't do a perfect job of measuring voltage: correct average value + some scatter.
  - Readout noise = measure of this scatter around the average value.
  - Independent of exposure time.
- A single 10 min exposure and the sum of 10x1 min exposures will both give us signal ***S***.
  - Readout noise occurs only once in 10 min exposure while 10 times in 1 min exposures.
  - Noise is random so law of averages applies: noise for the sum of K images is read-noise x $\sqrt{K}$

Sources of noise: Photon (shot) noise

- Depends on the amount of light reaching the CCD.
  - Due to random fluctuations of sky background and the light source itself.
  - Uncertainty in the photon counts N is $\sigma(N) = \sqrt{\langle N \rangle}$
  - Proportional to the exposure time.
- Sources of noise are uncorrelated. The variance of the total noise is:

$$\sigma^2 = N_{pix}\sigma^2_{read} + N_{pix}\sigma^2_{thermal} + N_{pix}\sigma^2_{sky} + \sigma^2_{star}$$

- For a given pixel measurement the SNR is:

$$\mathrm{SNR} = \frac{f_* q_e t}{[(f_* + f_s) q_e t + Dt + R^2]^{1/2}}$$

f= flux (phot/pix/s), D = dark current value(e-/pix/s), R = readout noise(e- rms/pix), qe = quantum efficiency, t = integration time (s)

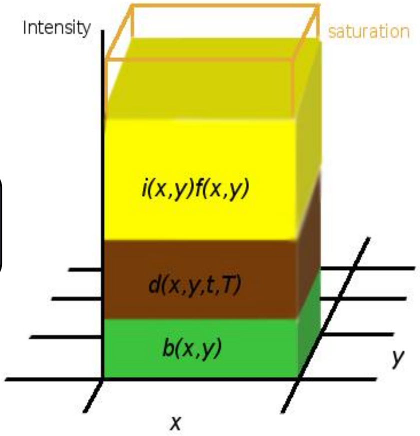
Calibrating the images: preprocessing

- Images need to be cleaned of artifacts.
- The different contributions to measured intensity $I(x,y)$:
  - Bias $b(x,y)$ → precharge value of CCD.
  - Dark $d(x,y,t,T)$ → value of accumulated thermal load during exposure.
  - Flat $f(x,y)$ → response factor of pixel determined from flat field image.
  - Source $i(x,y)$ → contribution from observed source.

Calibrating the images: preprocessing

- Images need to be cleaned of artifacts.
- The different contributions to measured intensity *I(x,y)*:
  - Bias *b(x,y)* → precharge value of CCD.
  - Dark *d(x,y,t,T)* → value of accumulated thermal load during exposure.
  - Flat *f(x,y)* → response factor of pixel determined from flat field image.
  - Source *i(x,y)* → contribution from observed source.

$$i(x,y) = [\, I(x,y) - \{b(x,y) + d(x,y,t,T)\}\,] : f(x,y)$$



Calibrating the images: calibration frames

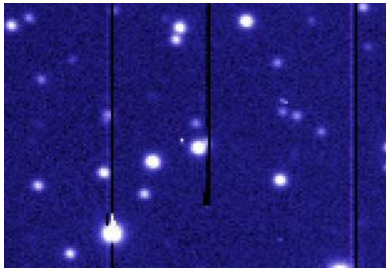
- Obtain calibration images to determine values of $b(x,y)$, $d(x,y,t,T)$, $f(x,y)$.
- Bias frame → 0s exposure with shutter closed.
  - They measure DC offset built into signal chain.
  - Used to fix constant to be subtracted from all raw images.
  - Master bias → median of N bias frames .
    - minimizes noise contribution.
    - eliminates cosmic ray hits.

Calibrating the images: calibration frames

- Dark frame → exposure of time *t* with shutter closed and at same *T* as the image to be processed.
  - Also contains the precharge.
  - Can be ignored if the CCD is cooled to low temperature.
- Flat field → exposure of uniformly illuminated background.
  - Takes into account optical transmission irregularities, e.g. dust, vignetting.
  - Master flat → average of many flats
    - normalized so average pixel value is 1.

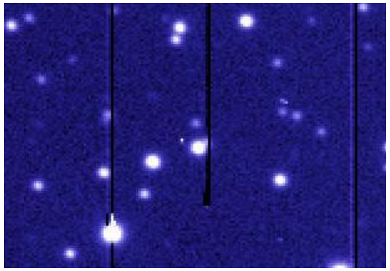
Calibrating the images: other problems

- CCDs often have defective pixels, lines & columns.
  - Some are “dead”, some “hot”.
  - Create a bad pixel mask.



Calibrating the images: other problems

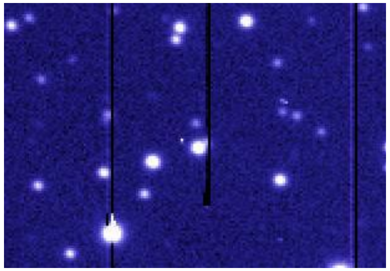
- CCDs often have defective pixels, lines & columns.
  - Some are “dead”, some “hot”.
  - Create a bad pixel mask.



- Cosmic ray hits on CCD
  - Liberate e- and cause bright pixels.
  - Identified and removed by software.

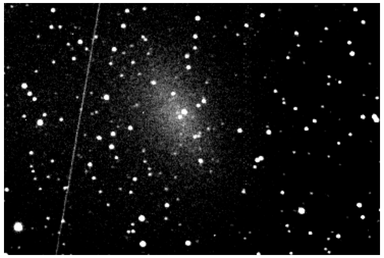
Calibrating the images: other problems

- CCDs often have defective pixels, lines & columns.
  - Some are “dead”, some “hot”.
  - Create a bad pixel mask.



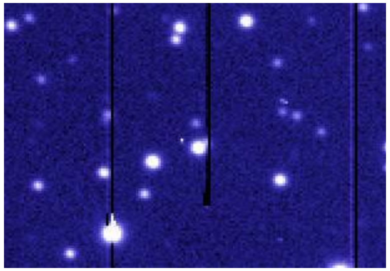
- Cosmic ray hits on CCD
  - Liberate e- and cause bright pixels.
  - Identified and removed by software.

Satellite tracks



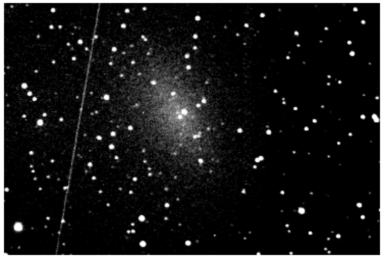
Calibrating the images: other problems

- CCDs often have defective pixels, lines & columns.
  - Some are “dead”, some “hot”.
  - Create a bad pixel mask.

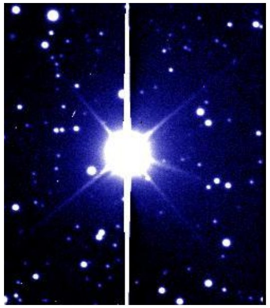


- Cosmic ray hits on CCD
  - Liberate e- and cause bright pixels.
  - Identified and removed by software.

Satellite tracks

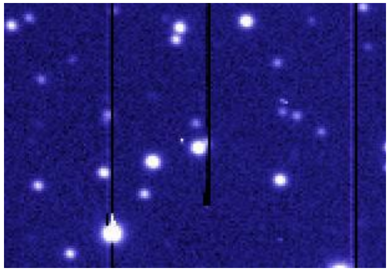


Diffraction spikes



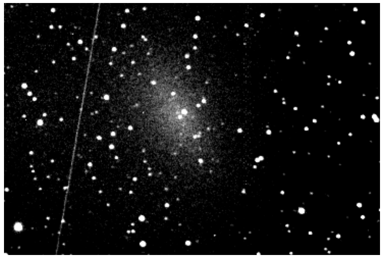
Calibrating the images: other problems

- CCDs often have defective pixels, lines & columns.
  - Some are “dead”, some “hot”.
  - Create a bad pixel mask.

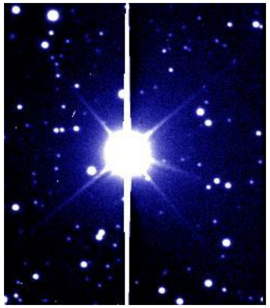


- Cosmic ray hits on CCD
  - Liberate e- and cause bright pixels.
  - Identified and removed by software.

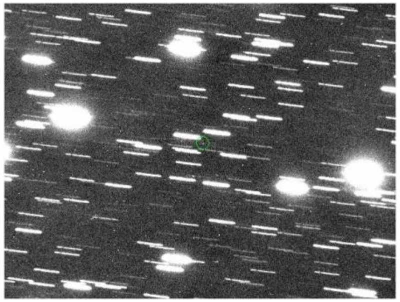
Satellite tracks



Diffraction spikes

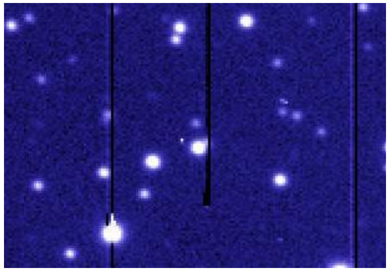


Trailing



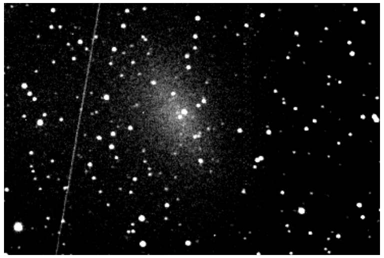
Calibrating the images: other problems

- CCDs often have defective pixels, lines & columns.
  - Some are “dead”, some “hot”.
  - Create a bad pixel mask.

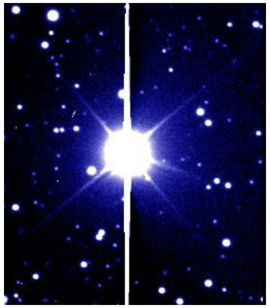


- Cosmic ray hits on CCD
  - Liberate e- and cause bright pixels.
  - Identified and removed by software.

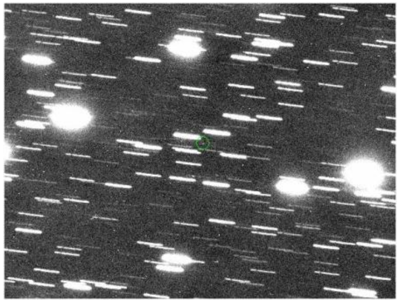
Satellite tracks



Diffraction spikes



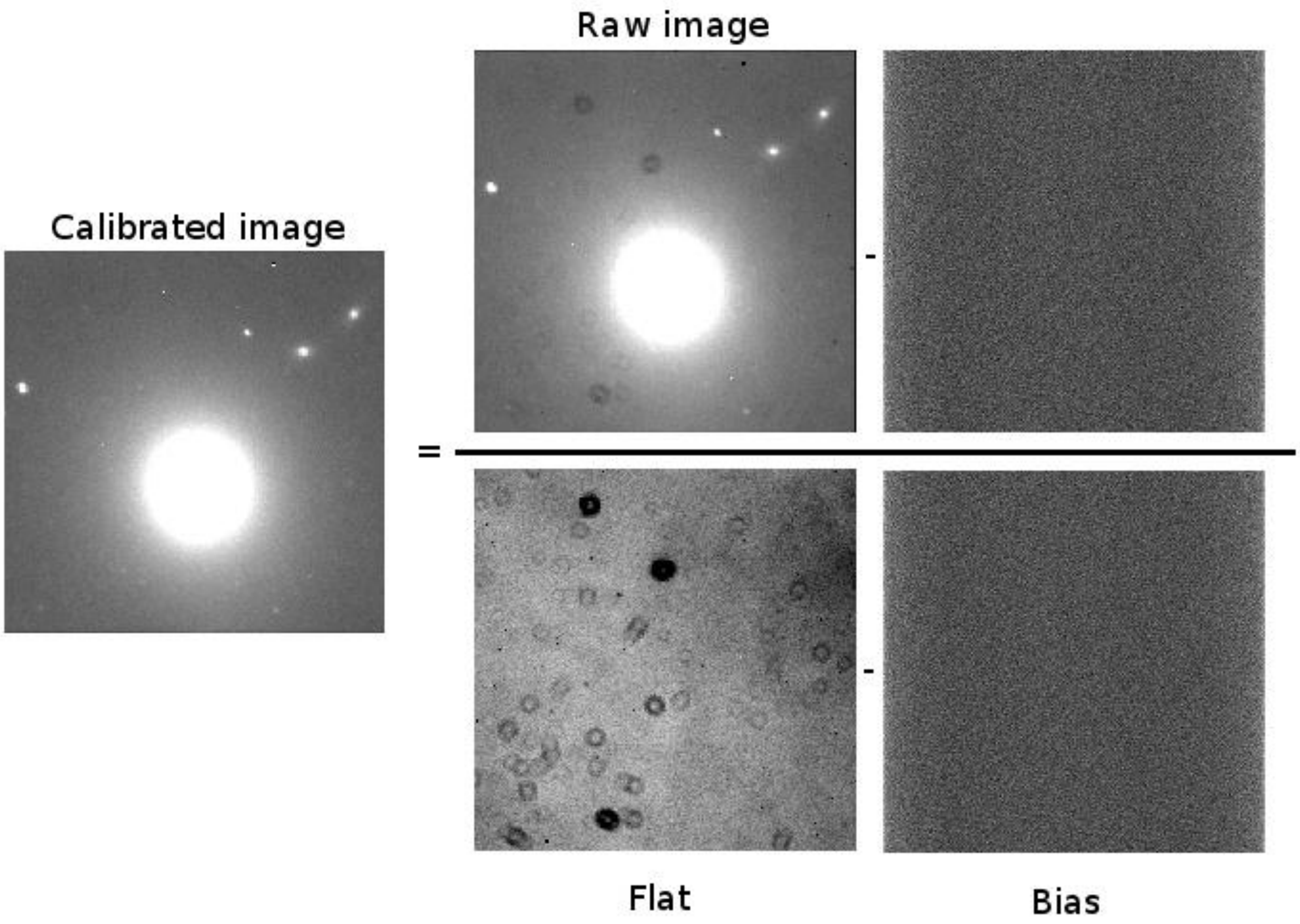
Trailing



also vignetting, variable sky background...

Taken into account during image processing phase.

Calibrating the images: producing a calibrated image

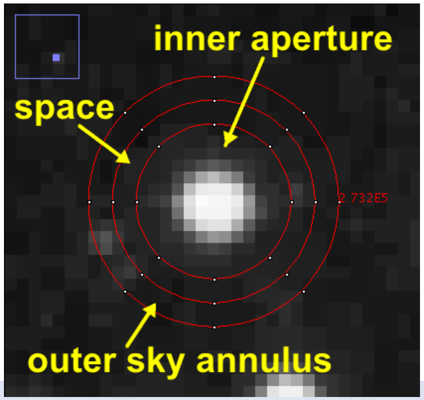


Preparing for the photometry

- Telescope pointing not 100% accurate.
- All calibrated images need to be registered.
  - Perform necessary shifts & rotations so that the same stars always fall on the same pixels.
- Exposure times for images of same target not always the same.
  - Photometry package will need to scale each image to match a reference with a fixed exposure time.
- Photometry options:
  - Aperture.
  - PSF fitting.
  - Difference imaging.

Aperture photometry

- Several available packages: DAOPHOT, PHOTOM, VAPHOT, Source Extractor...
  - Set up parameter files.
    - Approximate FWHM, gain, readout noise, saturation limits etc...
  - Identify stars on image.
  - Place an aperture over each star to measure total flux.
    - Aperture usually circular or elliptical.
  - To estimate sky background place a second larger aperture around the first and measure the total flux in the area between the two.
- Not so good for crowded fields.



PSF fitting photometry

- Several available packages: DAOPHOT, DoPHOT...
  - Set up parameter files.
    - Approximate FWHM, gain, readout noise, saturation limits etc...
  - Identify isolated stars on image above threshold.
  - Build point-spread function (PSF) model.
  - Fit PSF model to each star on image $\rightarrow$ total flux summed up over the stellar profile.

PSF fitting photometry

- Several available packages: DAOPHOT, DoPHOT...
  - Set up parameter files.
    - Approximate FWHM, gain, readout noise, saturation limits etc...
  - Identify isolated stars on image above threshold.
  - Build point-spread function (PSF) model.
  - Fit PSF model to each star on image → total flux summed up over the stellar profile.
- Errors from PSF fitting usually attributable to:
  - Bad determination of PSF model.
  - Flawed preprocessing & image registration stage.
  - Incorrect masking of artifacts.
  - Heavily blended objects not fit separately.
- DAOPHOT Available through IRAF

http://iraf.noao.edu/scripts/irafhelp?daophot

Difference imaging

- First developed by Tomaney & Crotts (1996) and Alard and Lupton (1998).
- Several packages in use: PySIS, DANDIA, ISIS...
- Involves matching the seeing variations between two images and subtracting them.
- Constant stars leave no residuals on the subtracted image.
- Variable stars will leave a positive or negative signal.
- Choice of reference image is crucial:
  - Must have good seeing.
  - Sometimes created by stacking several good images.
- Reference R is convolved to match image I.
- Determining the proper kernel solution, K, is difficult.

Difference imaging: determining the kernel solution

- Make no assumptions about PSF or background.
- Find the kernel by finding least-squares solution to:
  $R(x,y) \otimes K(u,v) = I(x,y) + bg(x,y)$
  - In principle: non-linear problem.
  - Choose a set of basis functions to model kernel.
  - In practice: linear problem for kernels made of 3 Gaussian components with associated polynomial degrees in range of 2-6 → subtracted images with very small residuals. *(Alard and Lupton 1998).*
  - Differential background (bg) can be fitted simultaneously with the kernel.
- Measures difference flux, not total flux as with PSF fitting.

Difference imaging: kernel as a discrete pixel array

- Alternatively, consider kernel as a discrete pixel array.
- Solve for kernel pixel values using linear least squares. *(Bramich 2008)*
- More flexible, no prior assumptions about “basis functions”.
- Does not require that the two images are perfectly aligned.
  - Flexible enough to correct for misalignments.
  - In the case of simple translation → no need to resample.
- Can deal with distorted star shapes but slow.
- If the model is *M*, image is *I* and *σ* the pixel uncertainties need to minimize

$$x^2 = \sum_{i,j} \left(\frac{I_{i,j} - M_{i,j}}{\sigma_{i,j}} \right)^2$$

Difference imaging: An implementation - DanDIA

- All DIA pipelines follow a similar basic structure.
- DanDIA written by Dan Bramich (http://www.danidl.co.uk/)
- Full pipeline with 8 stages starting from raw images.
- Code is well documented.
- Parameter file for each stage.
- Detailed logs at each stage.
- Produces difference lightcurves for all detected stars.
- Can easily be wrapped and automated (RoboNet).
- Requires IDL astronomy library (http://idlastro.gsfc.nasa.gov/).

Difference imaging: An implementation - DanDIA

- DanDIA is a set of wrapper routines for a set of IDL libraries.
- The IDL libraries will soon be publicly available.
  - Most of the tools you need:
    - Star finder, PSF construction, matching algorithms, image subtraction etc...
- Descriptions of the functions:

  http://www.danidl.co.uk/danidl.features.html

Difference imaging: An implementation - DanDIA

- Main code in IDL, some routines in C++.
  - Stage 1 – Image preprocessing.
  - Stage 2 – Cosmic ray removal (optional).
  - Stage 3 – Creation of reference image.
  - Stage 4 – Star identification & fitting.
  - Stage 5 – Image registration.
  - Stage 6 – Image subtraction.
  - Stage 7 – Difference photometry.
  - Stage 8 – Post processing (optional).

Stage 1 – Image preprocessing

- Aim: create calibrated images from raw images.
    - Requires raw images & calibration frames.
    - Requires .fits format.
    - Requires parameter file s1.preprocess.par.
    - Makes use of bad pixel mask file (optional).
- Expects particular directory structure:
    - All processing done under a master directory **<master_dir>**.
    - All images to be processed → subdirs of **<master_dir>**.
    - Image data organised to directories by day of observation.

Stage 1 – Preparing the parameter file

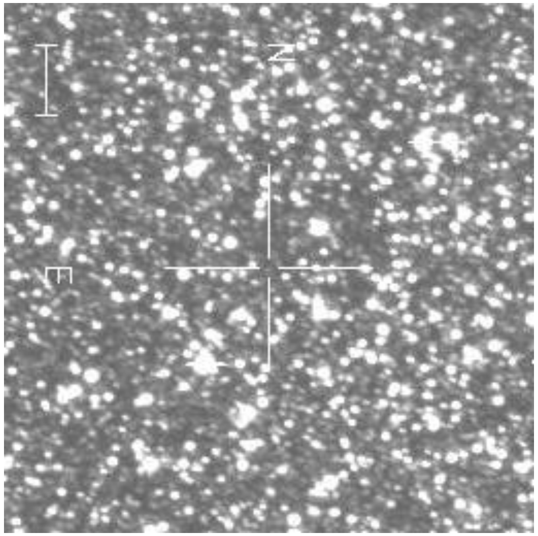
- s1.preprocess.par Must exist in **<master_dir>**.
- Declare
  - image header keywords.
  - overscan regions of CCD.
  - which corrections are required (Dark, Bias, Flat).
  - usable image region limits.
  - maximum ADU counts above which CCD saturates.
  - minimum useful pixel value (ADU).

Stage 1 – execution steps

- Logs are available in **<master_dir>**/s1.log.
- Checks directory structure and parameter file.
- Finds all images to be processed and calculates HJDs at mid-exposure.
- Checks for existence of bad pixel mask.
- Calculates overscan correction for all images.
- Creates **<master_dir>**/biases is bias frames are available for processing.
    - Produces master bias frame.
- Ditto for darks and flats.
- Applies corrections to raw science images and places them in **<master_dir>**/imred/**<filter>**/.
    - Also estimates the sky bg*(ADU)*, FWHM*(pix)* & ellipticity of the PSF.

Example

- MOA-2009-BLG-319
  - Sub-saturn mass planet (Miyake et al - http://arxiv.org/abs/1010.1809)
- Demo using 6 images from FTS, SDSS-I filter.
- Compile at IDL prompt with:
  - IDL>.r preprocess.pro
- Execute with:
  - IDL>preprocess, **'<master_dir>'**
- Examine output at each stage...
- Each stage comes with a log file.

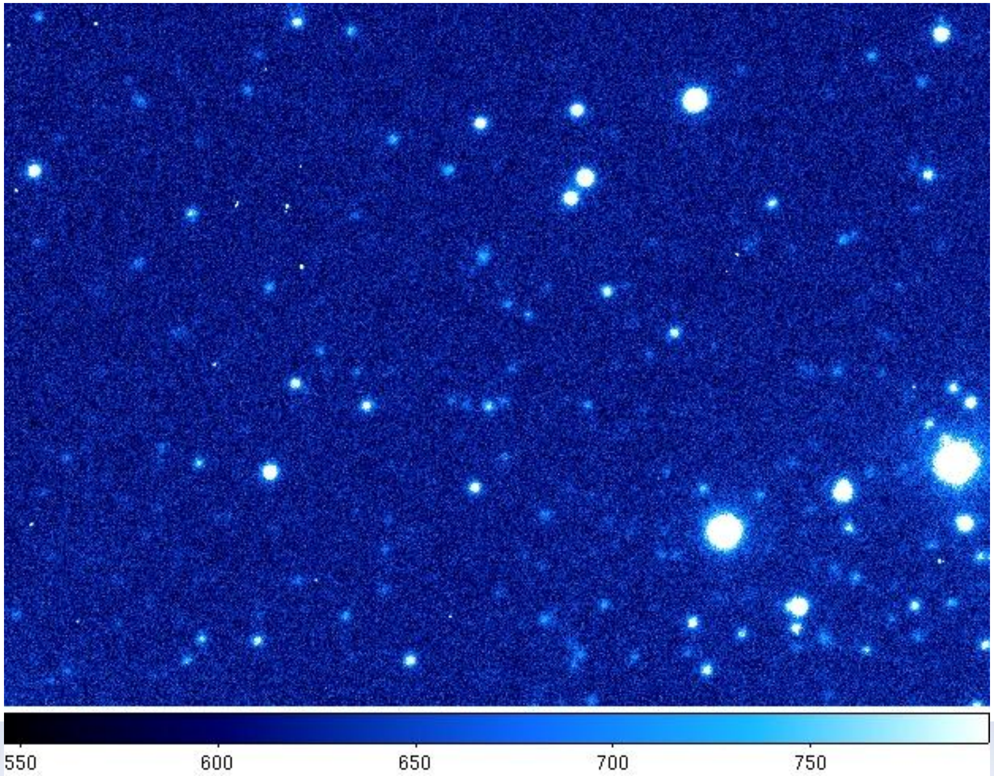


Stage 2 – Cosmic ray removal

- Aim: clean images of cosmic ray hits.
- Requires parameter file s2.cosmic.<filter>.par.
  - Must exist in **<master_dir>**.
  - Must specify read-noise, gain, clipping threshold...
  - Must specify path to precompiled C++ code.
- Goes through calibrated images produced by s1.
- Automatically removes cosmic ray hits.
  - Based on an algorithm employing Laplacian edge detection (van Dokkum, P. G., 2001, PASP, 113, 1420).
  - Takes into account bad pixel mask.
- Multiple instances can be run in parallel.
- Cleaned images placed in **<master_dir>**/cimred/**<filter>**/
- Log files created under **<master_dir>**: s2.<filter>.log

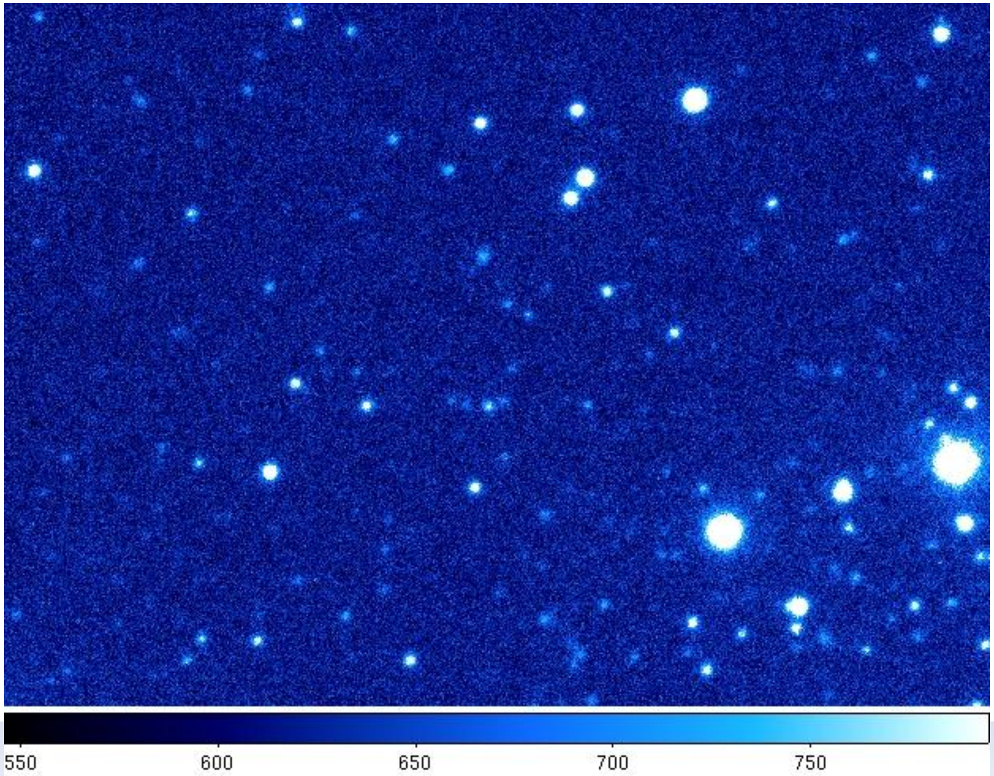
Stage 2 – Cosmic ray removal - example

- Note: May need to adjust parameters if images are too sharp (FWHM < 3.0 pix).
- Affected pixels are replaced by median of surrounding good pixels in a 5x5 box.
  - compile: IDL>.r cosmic.pro
  - execute: IDL>cosmic,**”<master_dir>”**,**”<filter>”**



Stage 2 – Cosmic ray removal - example

- Note: May need to adjust parameters if images are too sharp (FWHM < 3.0 pix).
- Affected pixels are replaced by median of surrounding good pixels in a 5x5 box.
  - compile: IDL>.r cosmic.pro
  - execute: IDL>cosmic,”**<master_dir>**”,”**<filter>**”



Stage 3 – Creation of reference image

- Aim: create a reference image for use in difference image analysis.
- Requires s3.reference.<filter>.par.
- Reference constructed by
    - Using best seeing image. (default)
    - Combining set of best seeing images.
- If **<master_dir>**/reflist.<filter>.txt exists, use it.
    - else autorun & create file.

Stage 3 – Creation of reference image

- Aim: create a reference image for use in difference image analysis.
- Requires s3.reference.<filter>.par.
- Reference constructed by
  - Using best seeing image. (default)
  - Combining set of best seeing images.
- If **<master_dir>**/reflist.<filter>.txt exists, use it.
  - else autorun & create file.
- Option to "smooth" image.
  - Convolves image with Gaussian kernel.
  - Can sometimes improve subtraction at loss of resolution.
  - Based on interpolation algorithm of Blu T., Thevenaz P. & Unser M., 2001, IEEE Tsansaction on Image Processing, 12, 11

Stage 3 – Creation of reference by combining images

- For best results:
  - Images must be taken on the same night.
  - Similar seeing conditions, low sky background.
  - Avoid constructing reference by combining images from different nights → your microlensing target may have changed brightness.
  - Is target star visible on reference frame?

Stage 3 – Creation of reference by combining images

- For best results:
  - Images must be taken on the same night.
  - Similar seeing conditions, low sky background.
  - Avoid constructing reference by combining images from different nights → your microlensing target may have changed brightness.
  - Is target star visible on reference frame?
- Images registered to the coordinate system of first image.
  - Detect bright stars. (Stetson P. B., 1987, PASP, 99, 191)
  - Match star positions between pairs of images.
    - Uses Delaunay triangulation and triangle matching.
    - Based on Pal A. & Bakos G.A., 2006, PASP, 118, 1474
  - Derive linear transformations.
  - Interpolate each image to first image coordinates.

Stage 3 – Preparing the parameter file

- s3.reference.<filter>.par must exist in **<master_dir>**.

- Declare:
  - type of reduced image to use for constructing reference (*imred*/*cimred*).
  - number of images to use.
  - maximum allowed sky bg value (ADU).
  - smallest acceptable value for PSF ellipticity.
  - f.o.v. of the camera.
  - average star spacing (pix) to be considered for matching purposes.

- Apply smoothing on/off.

Stage 3 – Creating the reference frame – example...

- At IDL prompt:
  - IDL>.r reference.pro
  - IDL>reference, **'<master_dir>'**,**'<filter>'**[,/REDO]
- Creates subdir **<master_dir>**/ref/**<filter>**/
  - Reference image → stackref.**<filter>**.fits
- Bad pixel map also available in fits format.
  - stackref.**<filter>**.bpm.fits
- .info file contains information on the images used for the construction.

Stage 4 – Star identification & fitting

- Aim: find the position and reference flux of all stars on reference.
- Requires s4.starfit.<filter>.par
- Based on "STARFINDER" IDL algorithm (Diolaiti, E. et al. 2000, A&A Sup. Ser., 147, 335).
- At all stages images registered using cubic O-MOMS* resampling (Blu et al 2001, IEEE Trans. Image Process., 10, 1069).
- Uses a discrete pixel array model for the image PSF.
- Can solve for spatially variable PSF.

* **O**ptimal **M**aximum-**O**rder **M**inimum-**S**upport synthesis functions.

Stage 4 – Preparing the parameter file

- s4.starfit.<filter>.par must exist in **<master_dir>.**
- Declare:
  - polynomial degree of the PSF model.
  - detection threshold in units of sky sigma.
  - PSF size in units of FWHM (PS).
  - PSF detection threshold for candidate stars in units of sky sigma.
  - companion distance in units of FWHM (CP).
    - Every star within 0.5*CP*PS of another star is considered a companion for PSF selection purposes.
  - maximum allowed flux ratio that any companion star can be allowed to have for a star to be considered for PSF generation.
  - minimum correlation coeff. a star can have to be considered for PSF generation.

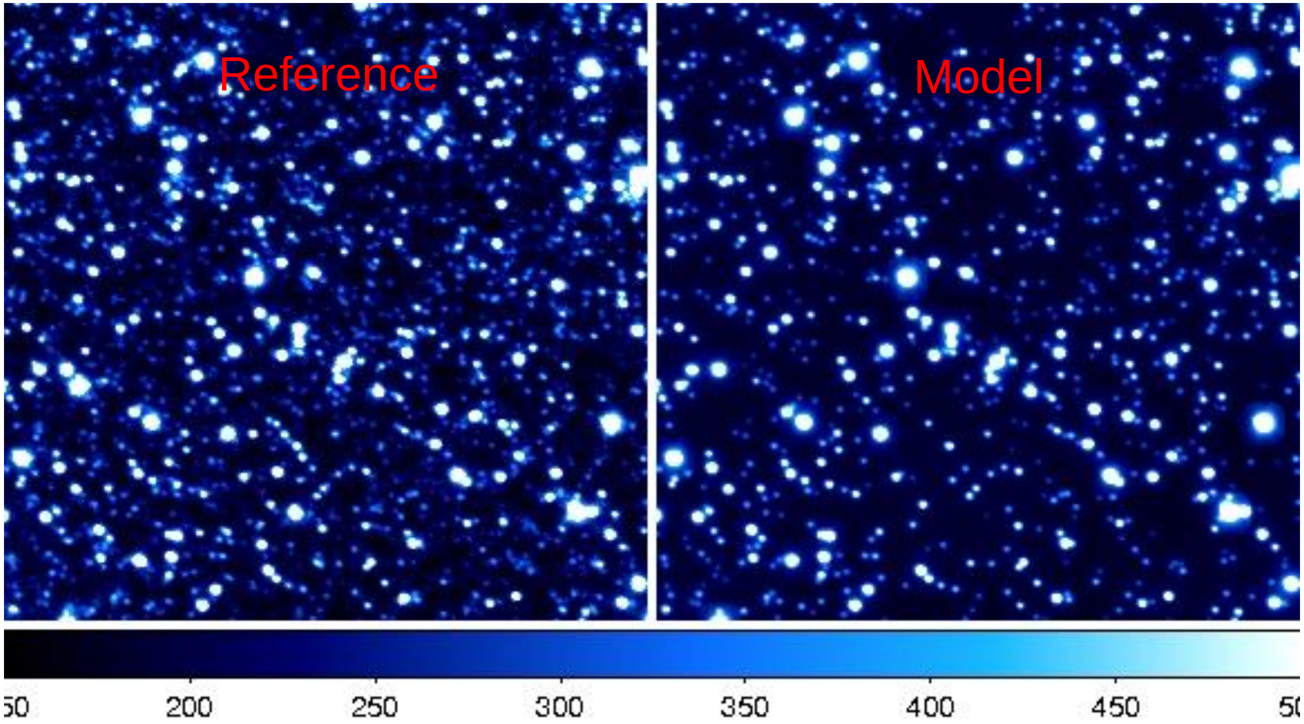
Stage 4 – Performing the star fitting - example

- At IDL prompt:
  - IDL>.r starfit.pro
  - IDL>starfit,**'<master_dir>'**,**'<filter>'**[,/REDO]
- Creates subdir **<master_dir>**/psf/**<filter>**
- **Step 1:**
  - Read in reference and make first estimate of sky bg.
  - Identify candidate PSF stars (bright, isolated, away from bad pixels).
  - Create initial estimate of reference image PSF.
    - Normalize PSF stars.
    - Resample to same coordinates.
    - Use median combining.

Stage 4 – Performing the star fitting - example

- **Step 2:**
  - Run STARFINDER
  - Look for stars using initial PSF estimate.
  - Attempt to deblend.
  - Redetermine PSF and repeats search & fitting.
- **Step 3:**
  - Remove stars close to bad pixels from final list.
  - Reconstruct reference model using star list. (.mod.fits)
  - Produce residual image = reference – model. (.sub.fits)
  - Write out sky background image for reference. (.bgd.fits)
  - Write out final starlist under **<master_dir>**/ref/**<filter>**/stackref.starlist.**<filter>**.txt

Stage 4 – Example results



11171 detected stars.

Identifying the target star

- Once the starlist is available → identify target position.
- Manually → Use finder chart and record x,y coordinates.
- Automatically (idtarget):
  - Use 2MASS catalogue to extract entries for all stars within FOV given field centre of reference.
  - Use a star finder algorithm to construct a catalogue of star positions on reference.
  - Compare the two catalogues and derive a WCS fit using WCStools (http://tdc-www.harvard.edu/wcstools/).

FITS WCS: using World Coordinate System

- Set of FITS conventions that specify a set of real world coordinates to associate with each pixel of an image.
- A common example is to specify the RA, Dec on the sky associated with a certain pixel of the sky image.
- WCS conventions described in:
    - Greisen, E. W., and Calabretta, M. R., Astronomy & Astrophysics, 395, 1061-1075, 2002
    - Calabretta, M. R., and Greisen, E. W., Astronomy & Astrophysics, 395, 1077-1122, 2002
    - Greisen, E. W., Calabretta, M. R., Valdes, F. G., and Allen, S. L., Astronomy & Astrophysics, 446, 747-771, 2006

Identifying the target star

- Once starlist is available → identify target position.
- Manually → Use finder chart and record x,y coordinates.
- Automatically (idtarget):
  - Use 2MASS catalogue to extract entries for all stars within FOV given field centre of reference.
  - Use a star finder algorithm to construct a catalogue of star positions on reference.
  - Compare the two catalogues and derive a WCS fit using WCStools (http://tdc-www.harvard.edu/wcstools/).
  - Calculate target star position based on RA, Dec given by survey.
  - Compare position with list of objects detected → closest one to position is accepted as target (within certain limits).

Identifying the target star

- Typically target position identified to within 2 pixels (~0.5 arcsec)
- Problems when the target is not visible on reference or extremely blended.

- Fallback: use finder chart assuming target is in the centre.
  - Resample and register reference to match pixel scale and orientation of finder chart.
  - Detect stars on both images.
  - Calculate coefficients of the 6-parameter coordinate transformation.
  - Transform known location of target on finder → original reference image coordinates.

Target identified at (x,y) = 692, 405

Stage 5 – Image registration

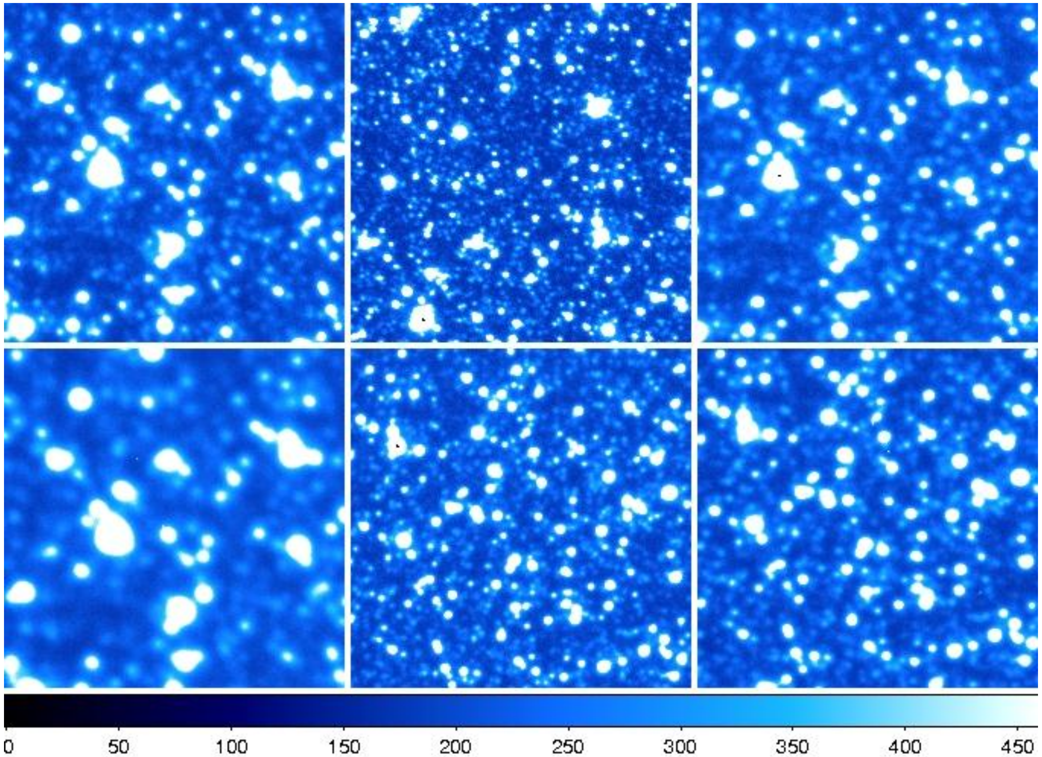
- Aim: register all images to reference image.
- Requires s5.register.<filter>.par.
- For every image...
  - Detect bright stars.
  - Match stars between image and reference using similar triangles. (Pal & Bakos 2006, PASP, 118, 1474)
  - Derive linear transformation from coord. system of image to that of reference.
  - Interpolate image to reference using O-MOMS (see stage 4)
- Multiple instances can be run in parallel.

Stage 5 – Preparing the parameter file

- s5.register.<filter>.par must exist in **<master_dir>**.
- Declare:
  - which image set to use. (imred/cimred)
  - transformation scale factor limits. (default values ok)
  - f.o.v. of CCD camera (deg).
  - star spacing limit.
    - This is the average spacing (pix) between the x,y coordinates of stars to be considered for matching.
    - Lower value →use more stars.
  - distance threshold to reject false matches (pix).
    - If transformed position of star is not within this threshold value from a registered star position on reference → do not use for matching.
  - Use image smoothing: yes/no.

Stage 5 – Image registration - example

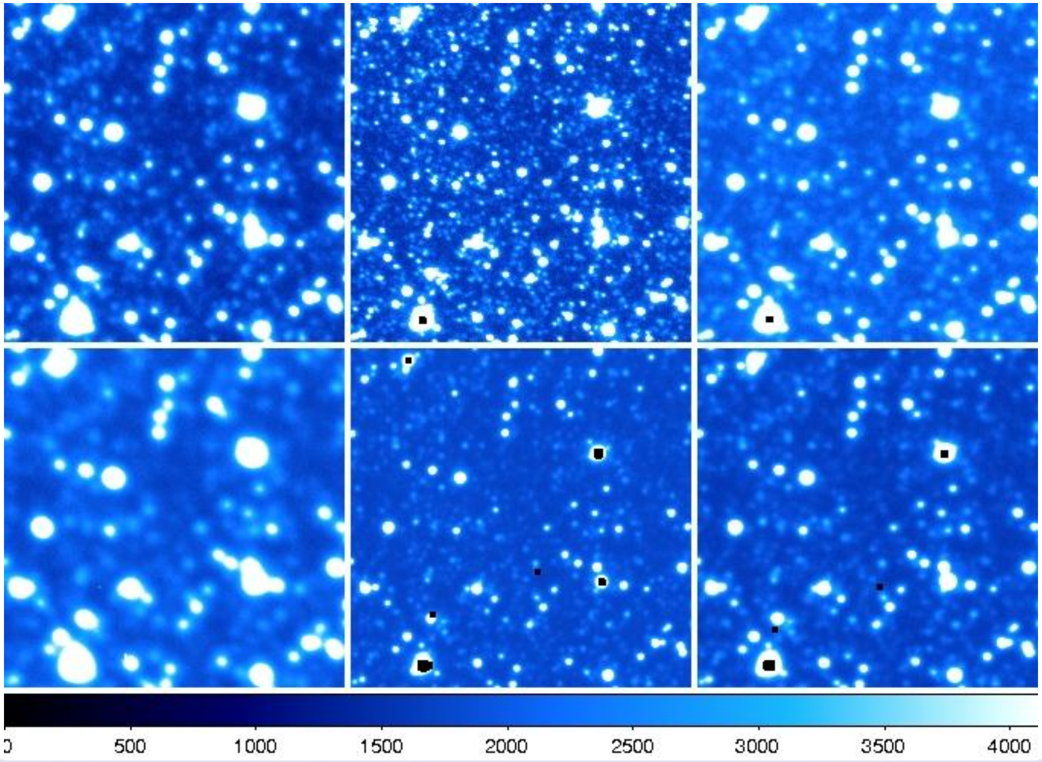
- At IDL prompt:
  - IDL>.r register.pro
  - IDL>register, **'<master_dir>'**,**'<filter>'**[,/REDO]
- Creates /**<master_dir>**/gimred/**<filter>**/
- Registered images stored with extension .geo.fits.



unregistered

Stage 5 – Image registration - example

- At IDL prompt:
  - IDL>.r register.pro
  - IDL>register, **'<master_dir>'**,**'<filter>'**[,/REDO]
- Creates /**<master_dir>**/gimred/**<filter>**/
- Registered images stored with extension .geo.fits.



registered

Stage 6 – Image subtraction

- Aim: produce difference images.
- Requires s6.subtract.<filter>.par.
- For every image:
  - Determines kernel solutions that best match reference. (Bramich D.M., 2008, MNRAS, 386, 77)
  - Photometric scale factor.
  - Differential sky bg.
- Solves for a discrete pixel array.
  - Allows spatially invariant & spatially variant kernels.
    - For spatially variant:
      - split image in subregions.
      - solve for individual pixel kernels.
      - interpolate the solutions to obtain kernel at any pixel.
  - Iterate using sigma clipping on the residuals.

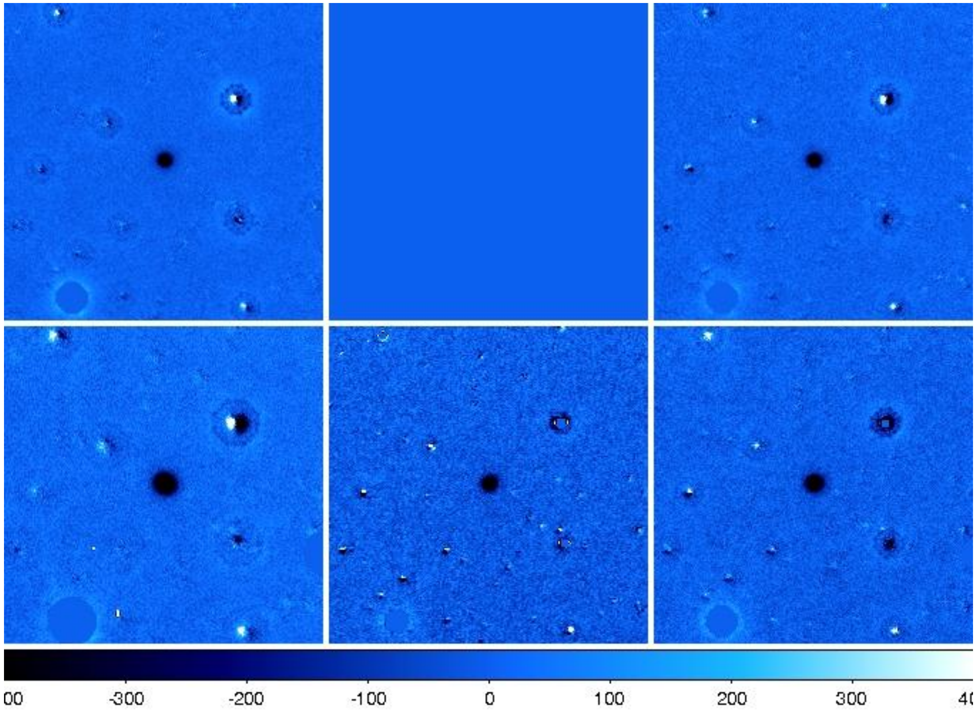
Stage 6 – Preparing the parameter file

- s6.subtract.<filter>.par must exist in **<master_dir>**.

- Declare:
  - CCD gain, readout noise.
  - kernel radius in units of image FWHM.
  - kernel threshold radius → pixels beyond this threshold value will have lower resolution.
  - number of subframes to use in x and y axis.
  - the amount of allowed subframe overlap between neighbouring regions.
  - number of allowed iterations in determining kernel solution.
  - spatially variable photometric factor: yes/no?
  - spatially variable differential bg: yes/no?

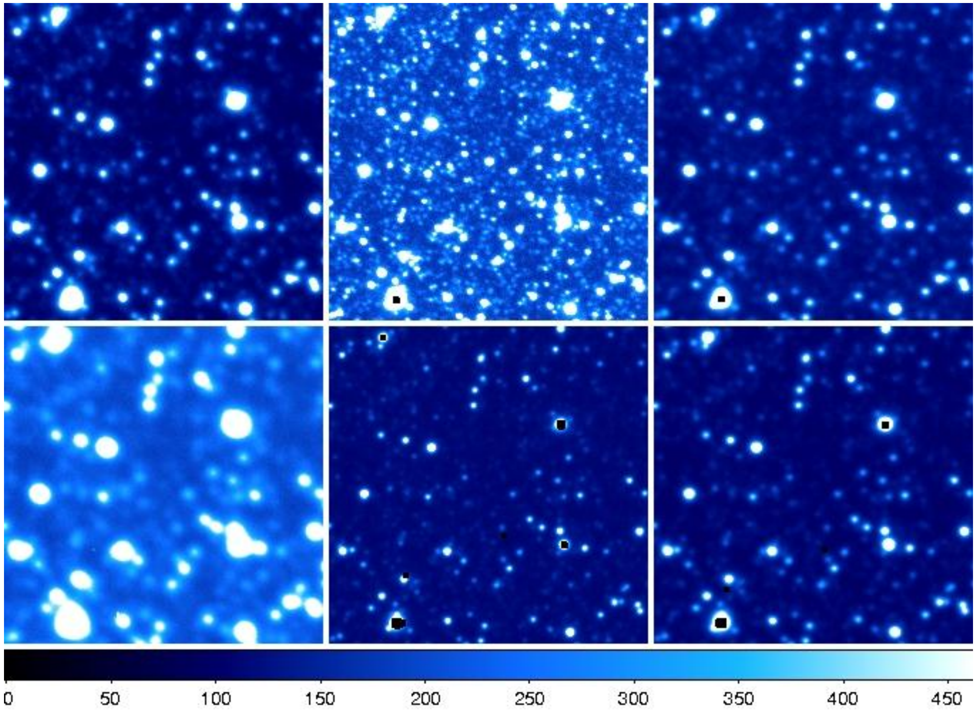
Stage 6 – Image subtraction - example

- At IDL prompt:
  - IDL>.r subtract.pro
  - IDL>subtract,**'<master_dir>'**,**'<filter>'**[,/REDO]
- Creates /**<master_dir>**/dimred/**<filter>**/
  - Contains .dif.fits and .res.fits (normalized residuals) files.



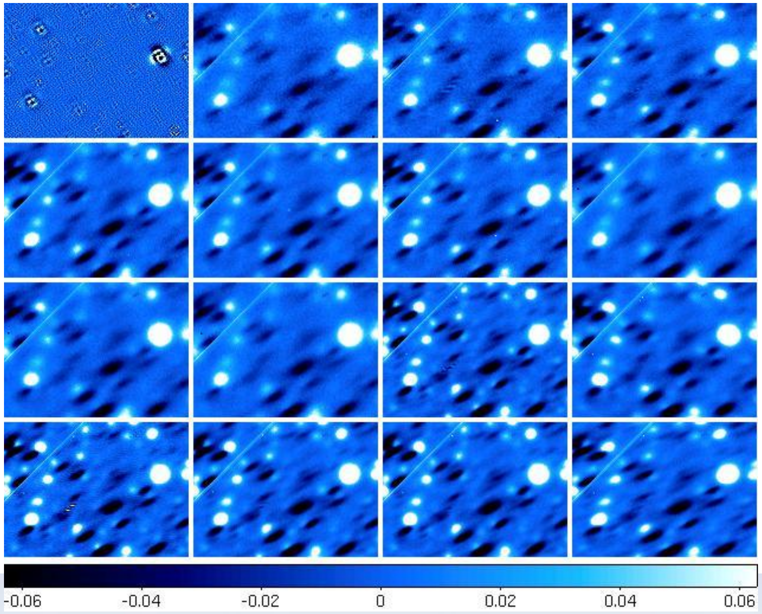
Stage 6 – Image subtraction - example

- At IDL prompt:
    - IDL>.r subtract.pro
    - IDL>subtract,**'<master_dir>'**,**'<filter>'**[,/REDO]
- Creates /**<master_dir>**/dimred/**<filter>**/
    - Contains .dif.fits and .res.fits (normalized residuals) files.



Stage 6 – Image subtraction - checks

- Check that difference images look okay.
- If not:
  - Does the PSF look okay?
  - Kernel size too small/large?
  - Reference a good image?
  - Allowed for enough stars to be detected in reference?



Stage 7 – Difference photometry

- Aim: measure difference flux & error for each star on set of difference images.
- Requires s7.diphot.<filter>.par.
- For each identified star position (from stage 4):
  - Measure difference flux:
    - Scale PSF to match star on difference image.
    - PSF model obtained by convolving PSF model from reference at the coordinates of the star with relevant kernel solution.
  - Iterative fitting using 4-sigma clipping of outliers.
    - 1st iteration used to calculate difference image pixel variances.
  - Process terminated when no more pixels are rejected.
  - Multiple instances can be run in parallel.

Stage 7 – example

- Prepare parameter file:
  - Declare CCD gain, readout noise.
  - Allow spatially variable photometric scale factor?
  - Allow spatially variable differential background?
- At IDL prompt:
  - IDL>.r diphot.pro
  - IDL>diphot, **'<master_dir>'**,**'<filter>'**[,/REDO]
- Creates **<master_dir>**/lc/**<filter>**/rawlc/ subdirectory.
  - Contains lightcurves of stars.
- Creates **<master_dir>**/lc/**<filter>**/sri/ subdirectory.
  - Contains the star residual image (.sri.fits)

Stage 7 – lightcurve files

- Output lightcurves have the format: lc_xcoord_ycoord_t
  - Plain text file → 29 Columns...
    - 2 → HJD.
    - 3 → difference flux.
    - 4 → difference flux error.
    - 11 → reference flux.
    - 12 → reference flux error.
    - 13 → reference exposure time.
    - 14 → magnitude.
    - 15 → magnitude error.
    - 18 → FWHM.
    - 19 → photometric scale factor.

Stage 7 – Equations used for conversions

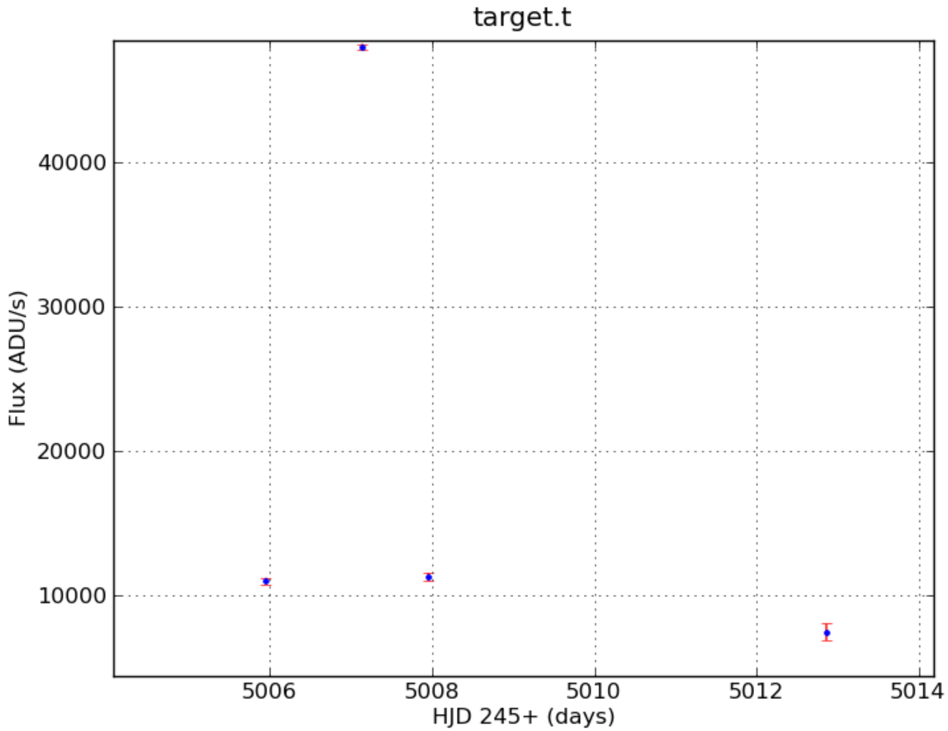
- total flux = reference flux + (difference flux / photometric scale factor)
- Mag = 25 - [2.5 * log10(total flux / reference exposure time)]
- Mag error = [2.5 / ln(10)] * [difference flux error / (photometric scale factor * total flux)]

Stage 8 – Post processing

- Stage under development.
- Aim:
  - Produce RMS diagrams.
  - Detrend lightcurves using sys-rem (Tamuz, Mazeh, Zucker 2005, MNRAS, 356, 1466).

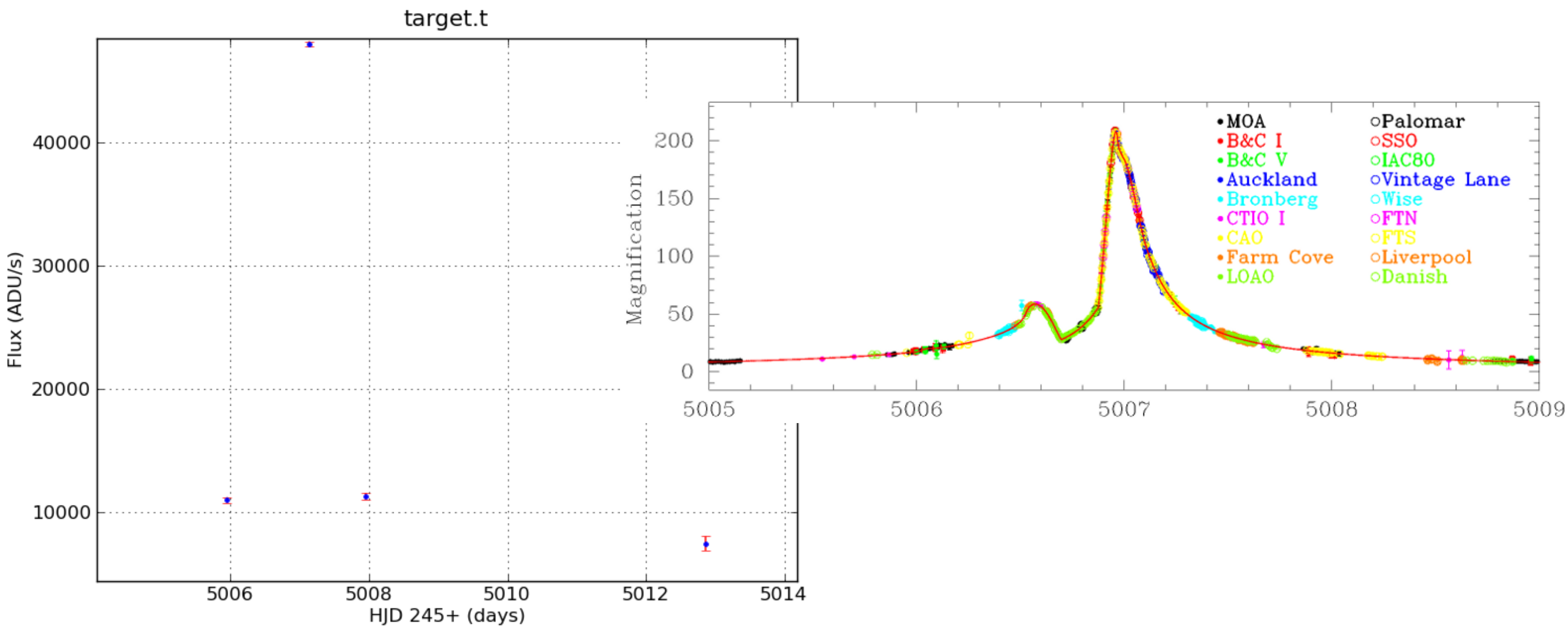
Constructing the lightcurve - example

- Extract the information from the lightcurve file of interest.
- We found the target at ~ (x,y) 692, 405 on reference.
- Closest identified star in star list: 691.703, 404.608
- Use lightcurve lc_00691.703_00404.608_t



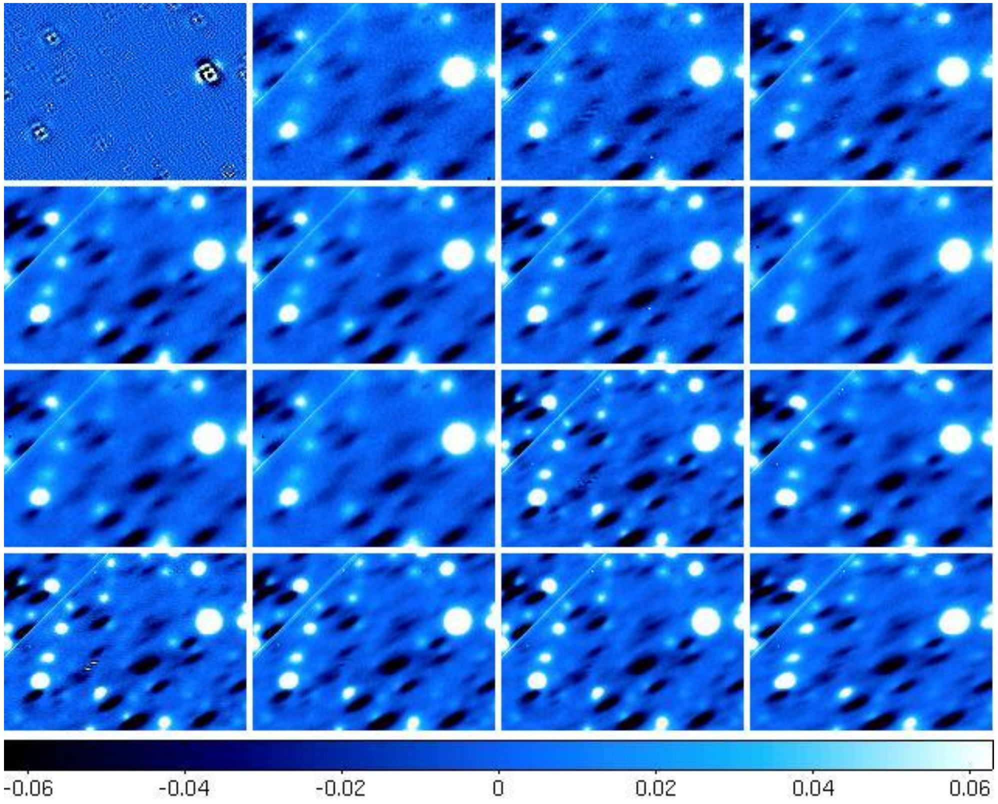
Constructing the lightcurve - example

- Extract the information from the lightcurve file of interest.
- We found the target at ~ (x,y) 692, 405 on reference.
- Closest identified star in star list: 691.703, 404.608
- Use lightcurve lc_00691.703_00404.608_t

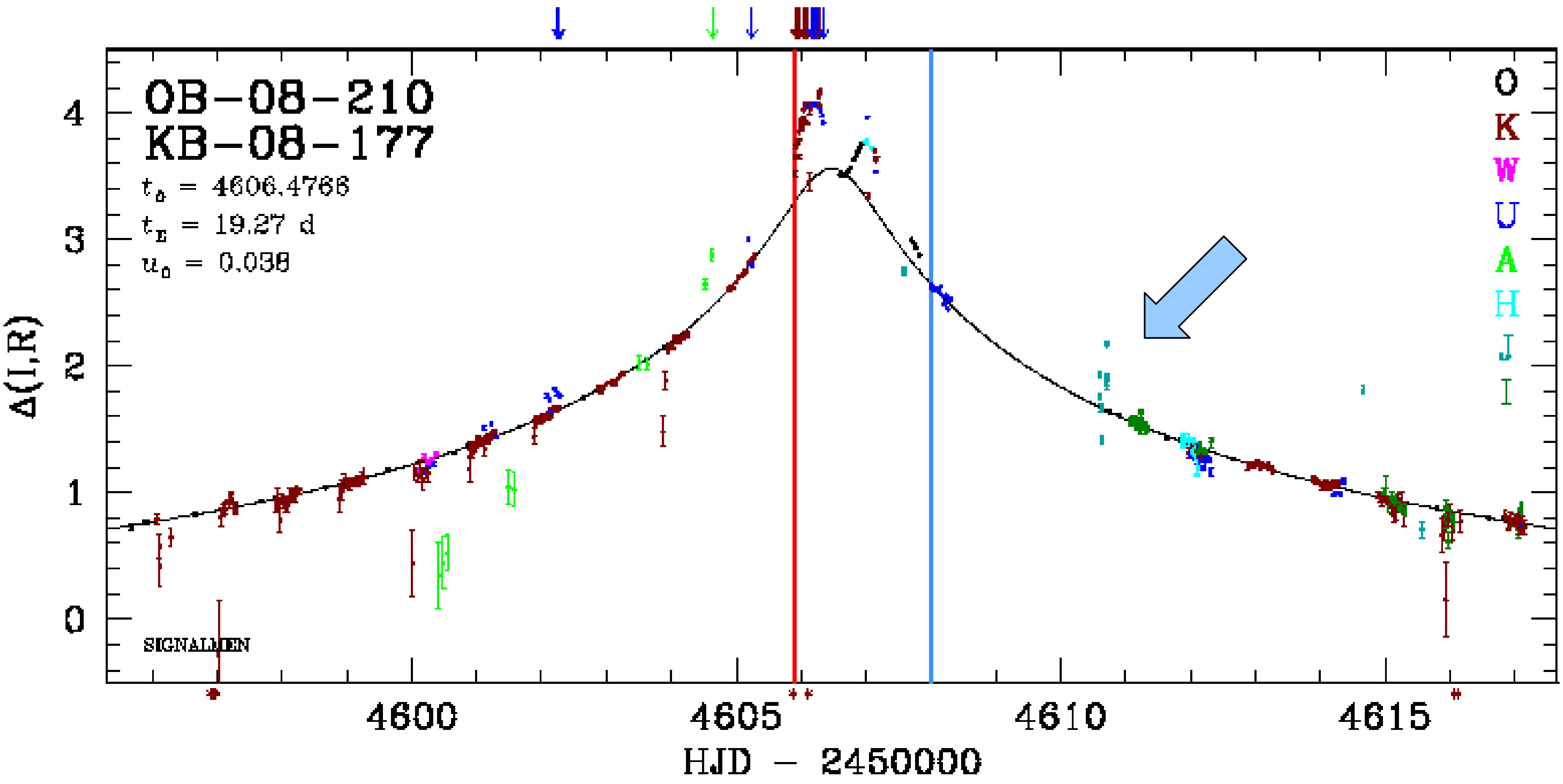


Things to watch out for

Bad subtractions

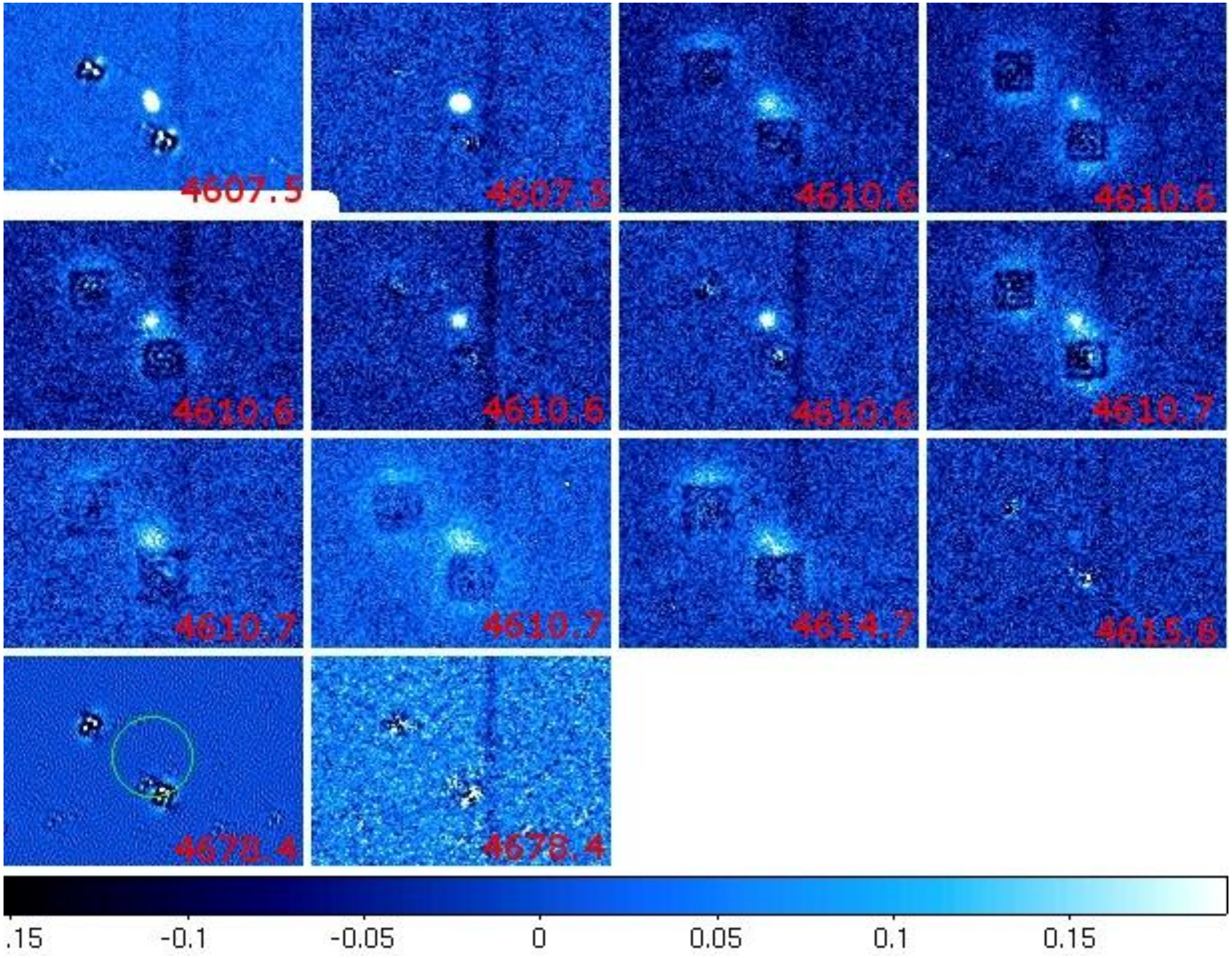


Things to watch out for



Things to watch out for

Bright stars close to target



Some Tips

- Always use your best seeing as reference (with low sky).
  - Algorithms may be confused by clouds – check number of detected stars.
- Make sure your PSF looks okay.
- Is the target (and surrounding stars) identified on the reference?
- Is the target visible on the reference?
- Is the target identified at the correct coordinates (Albrow et al. 2009 MNRAS, 397, 2099)?

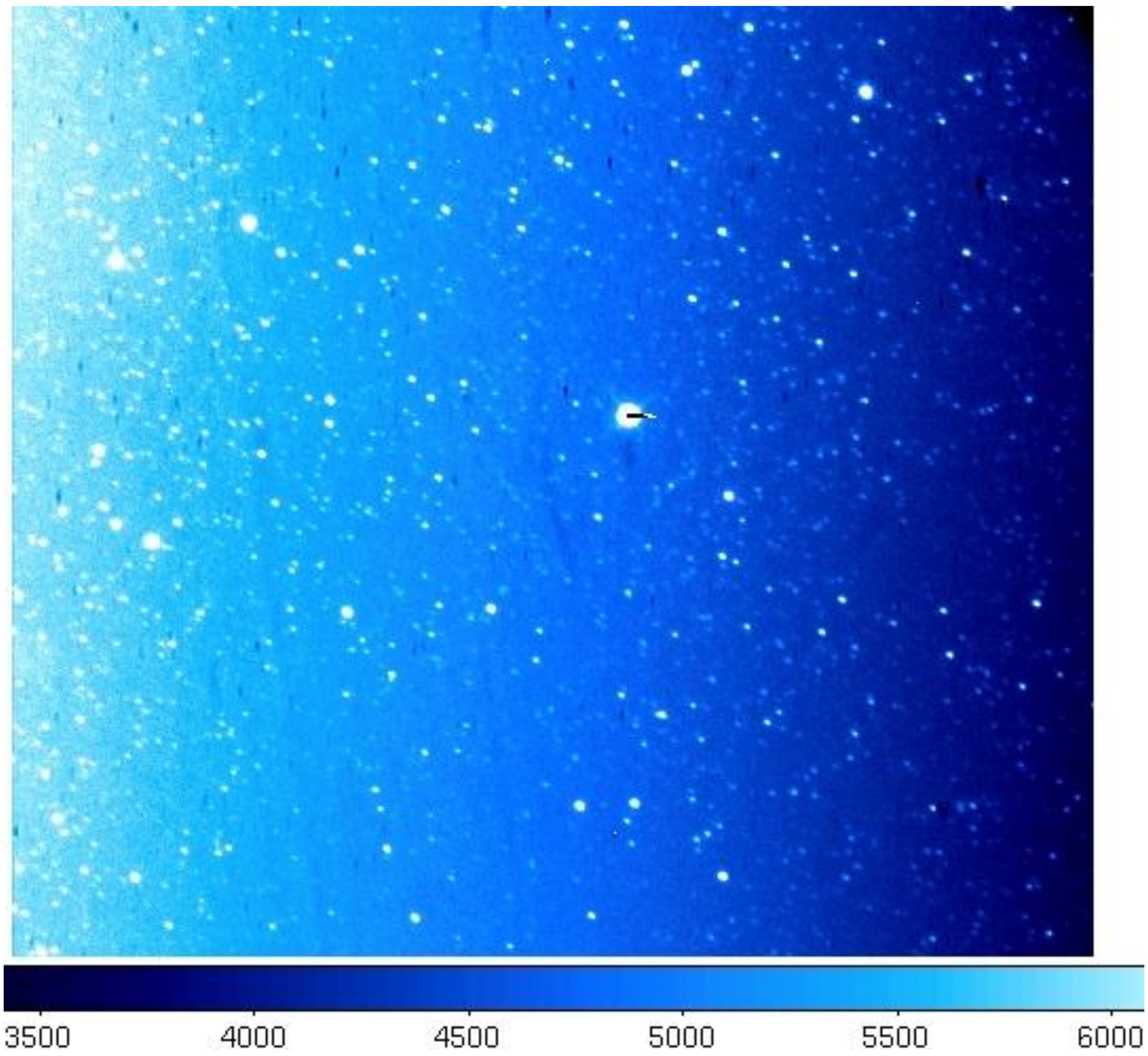
Some Tips

- Replace your reference if a better quality image becomes available and redo the photometry.
- Optimize the parameters at each stage according to instrument specifications and image quality.
- If only interested in one star, do not process whole image.
    - After registration → create image of smaller dimensions around target.
    - Crop all images to same dimensions & redo photometry.
    - It will speed up your processing.

THE END

Thank you for your attention

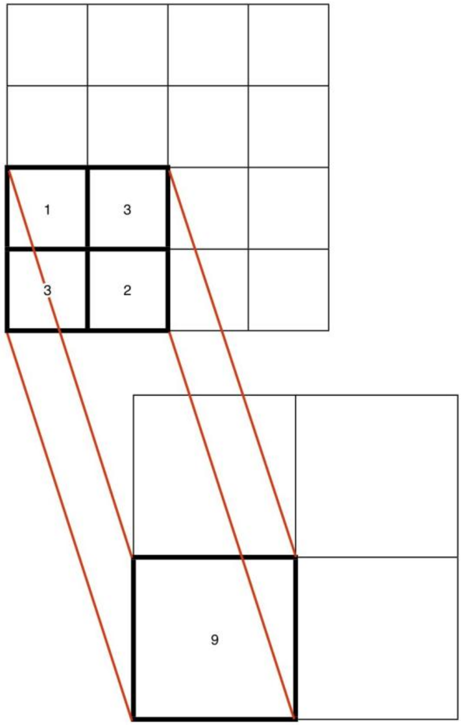
Variable sky background



Binning

- Consists of adding the charges contained in neighboring pixels to create a larger pixel.
  - 2x2 binning → groups the counts of 4 adjacent pixels.
- Advantages:
  - Increases the S/N of each pixel.
  - Allows for shorter exposure times.
  - Faster CCD readout.
- Disadvantages:
  - Loss of resolution.

Note: Resolution mitigated by seeing at site. 0.1"/pix at 1x1 bin. → 0.2"/pix at 2x2 has no effect if typical seeing is greater.



Stage 7 – Note on calculating the flux

- PySIS → each image aligned to reference by dividing with the photometric scale factor, then subtracted from reference.
  - flux measured on difference image on same scale as reference.
- DanDIA → scales reference by photometric scale factor then subtracts from target image.
    - Constructs model of data instead of scaling real image data.
    - Scale on difference image and reference not the same, must be adjusted according to model.
    - Aim: avoid introducing extra noise by direct manipulation of image.