

Example:Varuna

This example demonstrates a data reduction process in which the optical lightcurve of the fastly rotating trans-Neptunian object ^[1] (TNO), (20000) Varuna ^[2] has been measured. Observations were made in the night of 2010 November 05/06, between 23:34 and 04:27 UT using the 60/90/180 cm Schmidt telescope ^[3] located on the Piszkeš-tető Mountain Station ^[4] of the Konkoly Observatory ^[5], Hungary. Varuna orbits the Sun: through this set of observations, it has an apparent proper motion of approximately 4", that is not negligible. Therefore, photometry should be done by correcting to the motion of the target itself, however, due to the relatively short time scale, a linear motion is a rather good approximation for the apparent motion.

All in all, scientific 89 frames have been acquired with an exposure time of 180 seconds and without any interruption, using a 4K × 4K CCD camera mounted on the primary focal plane of the telescope. With this configuration, the effective field-of-view (FOV) of the images were approximately 1.2 × 1.2 degrees (see later on for a more accurate estimation on the size of the FOV that is based on the analysis of auxiliary data yielded by the data reduction process).

Throughout the reduction process, we assume that the calibrated scientific frames (89 in total) are stored in a sub-directory named `./fits` (see other examples related to calibration). In this example, the FITS files are named as `vf_20101105-0001-E.fits ... vf_20101105-0089-E.fits`. Let us suppose that the "basename" of these files (i.e. the name of each FITS file without the `*.fits` extension) is stored in a file named `base.list`. In the following, all file products of the reduction process that is related to the individual scientific frames are 1/ stored in some sub-directories depending on the context (e.g. data related to astrometry will be placed into the directory `./astrom`) and 2/ named using the above mentioned "basename" with an appropriate extension. For instance, the list of stars detected on the first frame is found in the file `./astrom/vf_20101105-0001-E.stars`.

In principle, the series of scripts presented in this example do the following:

- Detecting stars.
- Doing both relative and absolute astrometry (using the list of stars and a reference catalogue).
- Registering images to a reference frame (using the astrometric solutions obtained in the previous step).
- Scaling the images to the same flux level (that might vary due to the changes in the transparency and other atmospheric conditions).
- Doing an astrometry on Varuna itself: this is a moving target, so the pixel coordinates even on the registered images are not the same.
- Using the astrometry for Varuna, photometry is performed on the appropriate (interpolated) centroid coordinates. Since The flux levels are scaled to the same reference level, there is no need for photometry on comparison stars.

We have to note here that the above listed steps of data reduction process is only one of the many possibilities. Of course, there are many alternatives, which might be implemented as well with approximately the same complexity, for example:

- Instead of flux level scaling, photometry is performed on Varuna and on a few comparison stars and the actual brightness of Varuna is computed as the difference between the instrumental magnitude of the comparison star(s) and Varuna.
- Instead of registering the frames to the same reference image (or coordinate system), astrometry of Varuna is obtained using the original frames and during the astrometry. Thus, only the individual pixel coordinates are transformed to the same celestial coordinate system.

These data reduction steps are not discussed here, however, other examples found in this web page demonstrate these options as well. If you intend to try this set of data reduction scripts in your system, the full archive of related data can be downloaded by downloading the individual files listed here: `varuna-20101105.list` ^[6]. This archive includes the 89 calibrated frames, the `base.list` file containing the "basenames" of these frames, the projected catalogue,

`varuna.cat` centered around the target, a script (named `vizquery-varuna.sh`) that downloads and creates this catalogue using the `cdsclient` package (and the `grtrans` task) and the `varuna.reg` file containing coordinates for some bright stars. Note that this archive linked above is large: the individual calibrated images have a size of 4096×4096 pixels, thus each image is approximately 64 megabytes (using 32-bit floating point representation), therefore the whole archive is around 5.8 gigabytes.

Detecting stars

The first script uses the task `fistar` to detect the stars on each image. The flux threshold is set to 50000: this is a relatively high value, however, it is enough to detect approximately 2500 stars on each image (since the field itself is in the Milky Way). Such a number of individual stars are sufficient for both absolute and relative astrometry.

Files containing the list of stars are placed in the directory `./astrom` while the individual files will have an extension of `*.stars`. This list of detected stars contain 8 columns in the respective order of: internal identifier (a simple integer), X and Y coordinates, three profile shape parameters (S, D and K), and an estimation for the brightness (both in flux units and magnitudes).

```
#!/bin/bash

FITS=./fits
ASTR=./astrom

cat base.list | \
while read base dummy ; do
    fistar $FITS/$base.fits \
        --flux-threshold 50000 --model elliptic \
        --format id,x,y,s,d,k,flux,mag \
        --output $ASTR/$base.stars
    echo "$base: done." >> /dev/stderr
done
```

Astrometry

The next script uses two subsequent calls of the task `grmatch` to obtain absolute and relative astrometry, respectively. The first call for `grmatch` uses the reference catalogue found in the file `varuna.cat`, that is an excerpt from the USNO-B1.0 catalogue for this field. The relative astrometry (done by the second call of `grmatch`) derives a spatial transformation which moves the frame to one of the images that has been selected for being a reference (`vf_20101105-0045-E`, in practice, this is the middle frame for this series of observations).

Files containing the absolute and relative astrometry are also placed in the directory `./astrom` while the individual files will have an extension of `*.atrans` and `*.dtrans`, respectively. These transformation files are suitable as an input for the tasks `grtrans` and `fitrans`, the latter one will also be used in the next step (registration).

```
#!/bin/bash

ASTR=./astrom

ref=vf_20101105-0045-E

cat base.list | \
while read base dummy ; do
```

```

grmatch --reference varuna.cat --col-ref 2,3 --col-ref-ordering -5 \
--input $ASTR/$base.stars --col-inp 2,3 --col-inp-ordering -8 \
--max-distance 1 \
--order 3 --triangulation auto,unitarity=0.01,maxref=1000,maxinp=1000 \
--weight reference,column=5,magnitude,power=2 \
--comment --output-transformation $ASTR/$base.atrans

grmatch --reference $ASTR/$ref.stars --col-ref 2,3 --col-ref-ordering -8 \
--input $ASTR/$base.stars --col-inp 2,3 --col-inp-ordering -8 \
--max-distance 1 \
--order 3 --triangulation auto,unitarity=0.01,maxref=1000,maxinp=1000 \
--weight reference,column=8,magnitude,power=2 \
--comment --output-transformation $ASTR/$base.dtrans

echo "$base: done." >> /dev/stderr

done

```

The absolute astrometric transformations (in the files `*.atrans`) can be used to precisely estimate the plate scale of the images and therefore the size of the field-of-view. Since the coordinates in the reference catalogue `varuna.cat` are expressed in degrees, the linear coefficients (in practice, the square root of the determinant of the matrix for the linear coefficients) directly give us the plate scale in pixels/degree units. This is approximately 3508.5 for this setup, thus the field-of-view has a size of $4096/3508.5 = 1.167$ degrees.

We should note here that regarding to the processing of astronomical images, "astrometry" might mean slightly different things, depending on the context. The default interpretation of the term "astrometry" is to obtain the celestial coordinates (α : right ascension, δ : declination) of an object by analyzing the images. The most simple way to do this is to fit the (x,y) pixel coordinates of the centroid of the given object (star, minor planet, ...), and then do an $(x,y) \leftrightarrow (\alpha,\delta)$ transformation. In the practice of the FITS package, this transformation is done in two steps. See the example related to Example:astrometry `astrometry` for more details.

Image registration

This script employs the task `fitrans` to register the original calibrated frames to the same system as the reference image (`vf_20101105-0045-E`). Since the transformation files, `./astrom/*.dtrans` describe how the reference system should be transformed to the individual frames, here we should use the reverse of this transformation. The command line switch `-c` for `fitrans` imply a spline-based inter-pixel interpolation instead of the default linear interpolation.

The output of this process step are placed in the directory `./reg`. Note that since the registered images are also stored in FITS format, these share the same extension, `*.fits` as the original images in the `./fits` directory.

Flux level scaling

The flux level scaling is done in two distinct steps. First, aperture photometry is performed on a pre-selected list of stars (found in the file `varuna.reg`). The results of this aperture photometry are stored in the directory `./phot`. These photometric results are used in the second step, where the background information is used to subtract the background of the image to a nearly zero level while the flux information is used to scale the intensity level of the images to a previously selected reference level (that is the same as the astrometric reference, `vf_20101105-0045-E`).

Since the FOV of these images is relatively large (1.167×1.167 degrees), both the background removal and the flux level scaling is done in a linear (first) order. To fit a certain background level or a flux scale level, the task lfit is used to obtain the coefficients. In order to subtract the background and scale the flux levels, the task fiarith is used to evaluate the coefficients obtained by lfit.

```
#!/bin/bash

REG=./reg
PHOT=./phot

cat base.list | \
while read base dummy ; do

    fiphot --input $REG/$base.fits \
           --input-list varuna.reg --col-xy 2,3 --col-id 1 \
           --aperture 8:15:25 --format IXY,FfBb \
           --output $PHOT/$base.phot

    echo "$base: done." >> /dev/stderr

done

#!/bin/bash

REG=./reg
PHOT=./phot
SCALED=./scaled

ref=vf_20101105-0045-E

cat base.list | \
while read base dummy ; do

    A=$( lfit $PHOT/$base.phot \
           -c x:2,y:3,bg:6 -v a,b,c -n 3 -r 3 \
           -f "a*(x-2048)/2048+b*(y-2040)/2048+c" -y bg ))

    B=$( paste $PHOT/$ref.phot $PHOT/$base.phot | \
           lfit \
           -c x:2,y:3,ri:4,ii:11 -v a,b,c -n 3 -r 3 \
           -f "a*(x-2048)/2048+b*(y-2040)/2048+c" -y "ri/ii" ))

    fiarith ["'$REG/$base.fits'"] ((a-((A[0]))*x+(A[1])*y+(A[2]))*((B[0])*x+(B[1])*y+(B[2]))) \
           --output $SCALED/$base.fits

    echo "$base: done." >> /dev/stderr

done
```

Astrometry of the moving target Varuna

This script also employs the task `fistar` to do astrometry on Varuna itself. A simple visual inspection of the registered (and scaled) frames gives us that Varuna is the only one prominent source in the region of [2021,2034] × [1835,1852]. Thus, we use this constraint for `fistar` to search stellar profiles only in this section of the scaled images.

This preliminary astrometry for Varuna is stored in a single file named `varuna.star`. Each line of this file contains the sequence number of the frame (i.e. an integer between 1 and 89) and the same 8 fields as it was described above in the very first reduction step of detecting stars.

Some notes:

- The filter of `head -n -2` in the script expunges the last two frames of the `basename` list, since due to twilight, these frames are rather noisy and Varuna can only barely be detected.
- In addition, Varuna might not be detected in some of the frames, thus the condition `if [-n "$l"] ; then ... ; fi` is used to print only those lines where there is a detection for this TNO.

```
#!/bin/bash

SCALED=./scaled

n=1

cat base.list | \
head -n -2 | \
while read base dummy ; do
    l=$(fistar $SCALED/$base.fits \
        --only-candidates --format id,x,y,s,d,k,flux,mag \
        --flux-threshold 2000 --section 2021:2034,1835:1852)
    if [ -n "$l" ] ; then
        echo $n $l
    fi
    n=$((n+1))
done > varuna.star
```

By inspecting the output file `varuna.star`, it can be seen that the *individual* centroid coordinates have a scatter of 0.3 pixels. However, there are numerous (89) frames that can be used for astrometry, so the initializing step of the photometry is to perform a regression on this file, by fitting a "straight line to the individual X and Y pixel coordinates, independently. In this process, we silently exploit the fact that all of the 89 frames were taken with exactly the same cadence, so using the frame sequence number for this linear interpolation is as good as the time for an independent value of this linear fit. Of course, if cadence varies in time, this frame number n must be replaced by the time (for instance, Julian Day).

Photometry of Varuna

The final step of the data processing is performing the aperture photometry on Varuna itself. The centroid coordinates for each frames are derived by a linear interpolation where the linear coefficients are computed by `lfit`, by fitting a "straight line" to the individual pixel coordinates. The results of these two fits (for X and Y coordinates) are stored in the shell arrays `$FX` and `$FY`, respectively, and subsequently used inside the shell loop to obtain the interpolated pixel coordinates.

These interpolated pixel coordinates are then simply used as an input for the photometry, in parallel with the respective scaled images. The flux scaling allows us to treat the output of the photometry directly as the magnitude

of Varuna. The individual calls of `fiphot` print the results to the standard output, that is collected by the shell loop itself by feeding the file `varuna-a2.0.phot`. This file contains the Julian Day as well, (that is derived using the appropriate FITS keywords with some simple tricks), so the final output file can directly be used by any plotting or analysis tool.

```
#!/bin/bash

FITS=./fits
SCALED=./scaled

FX=$(lfit varuna.star -c n:1,x:3 -v x0,dx -f "x0+dx*(n-45)" -y x)
FY=$(lfit varuna.star -c n:1,y:4 -v y0,dy -f "y0+dy*(n-45)" -y y)

n=1
cat base.list | \
while read base dummy ; do
    A=$(fihheader $FITS/$base.fits --get date-obs | tr "T:-" " ")
    ymd=`echo ${A[*]} | awk '{ print $1,$2,$3+($4*3600+$5*60+$6)/86400; }'`
    jd=`echo ${ymd[*]} | awk '{ printf("%.5f\n",2455500.5+$3); }'`

    x=`echo $n | awk -v x0=${FX[0]} -v dx=${FX[1]} '{ n=$1; printf("%.2f\n",x0+dx*(n-45)); }'`
    y=`echo $n | awk -v y0=${FY[0]} -v dy=${FY[1]} '{ n=$1; printf("%.2f\n",y0+dy*(n-45)); }'`
    echo $jd $x $y | \
    fiphot --input $SCALED/$base.fits \
          --input-list - --col-id 1 --col-xy 2,3 \
          --aperture 2.0:5:10 --sky-fit mode \
          --format IXY,MmBbFf --output -

    n=$((n+1))
done > varuna-a2.0.phot
```

Plotting the results

This script below simply calls the program `gnuplot` to plot the results of the photometry to a graphics file (in EPS format).

```
#!/bin/bash

gnuplot > varuna-lc.eps << EOF
set terminal postscript portrait color
set size 1,0.5

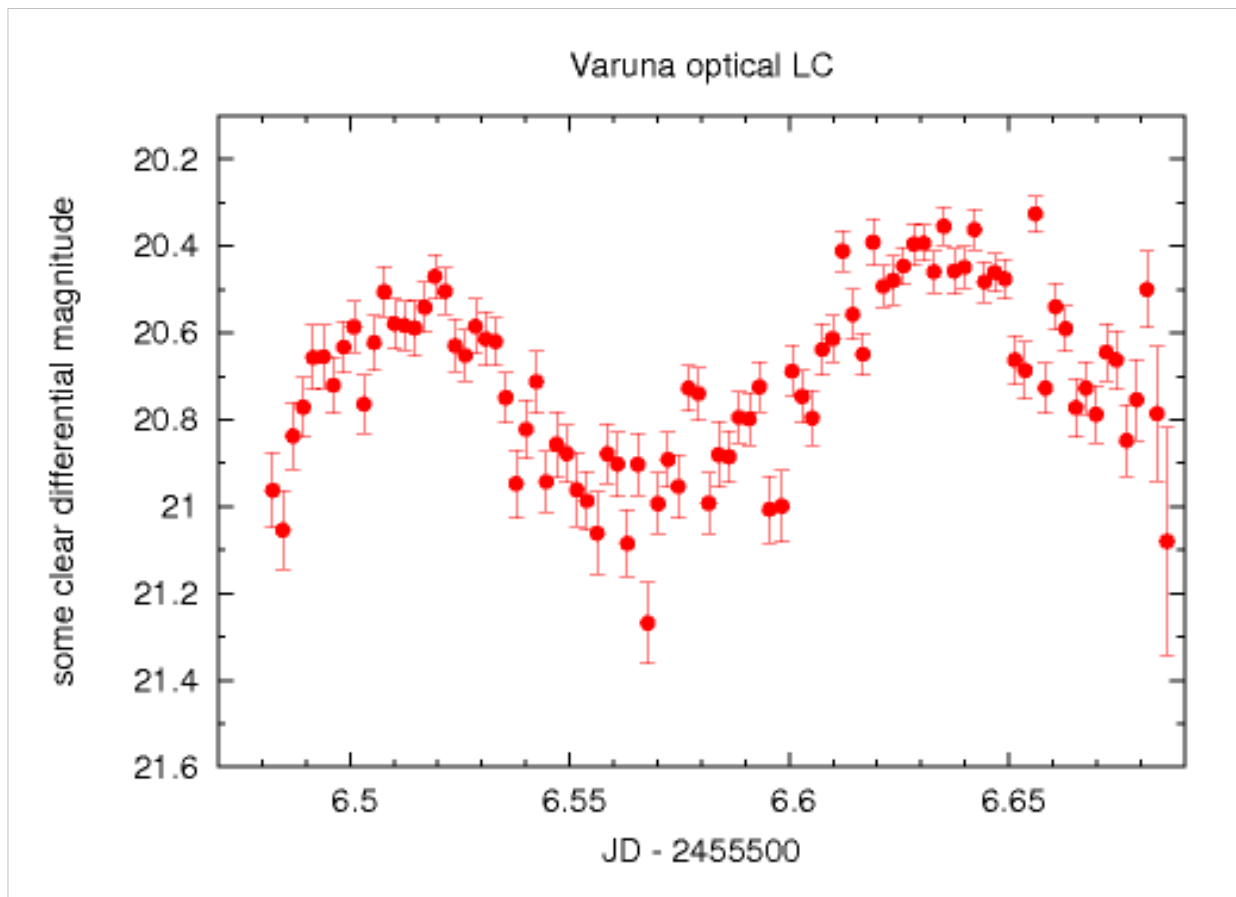
unset key

set title "Varuna optical LC"

set mxtics 5
set mytics 2
set xlabel "JD - 2455500"
set ylabel "some clear differential magnitude"
```

```
set yrange [:] reverse
plot [6.47:6.69] [21.6:20.1] \
      "varuna-a2.0.phot" u (\$1-2455500):(\$4+9):5 w err pt 7 ps 1
EOF
```

The output of this plotting script, i.e. the photometry for Varuna for (the second half of) the night of 2010 November 05/06 is then:



References

- [1] https://en.wikipedia.org/wiki/Trans-Neptunian_object
- [2] https://en.wikipedia.org/wiki/20000_Varuna
- [3] http://ccdsh.konkoly.hu/wiki/Schmidt_telescope
- [4] http://ccdsh.konkoly.hu/wiki/Piszk%C3%A9st%C5%91_Mountain_Station
- [5] http://www.konkoly.hu/index_en.shtml
- [6] <http://fitsh.net/archive/list/varuna-20101105.list>