Astro-Wise Training - photometric pipeline

Basic use and exercises

14 - 18 November 2005

Documentation about the photometric pipeline can be found at the Astro-Wise portal website: `http://portal.astro-wise.org/aw_howtos.shtml (under calibration pipeline > photometric pipeline`. of this documentation.

1 Exercises

1.1 Standard star catalog and extinction curve

More detailed information than given below about the standard star catalog and extinction curve can be read at:

`http://portal.astro-wise.org/man_howto_standardcals/man_howto_standardcals.shtml`

1. *Question1: Retrieve the standard star catalog from the database and explore its interfaces.*

```
Answer1: To retrieve the catalog:
awe> refcat = PhotRefCatalog.get()
awe> refcat.retrieve()
To explore it interfaces: see possibilities by typing:
awe> dir(refcat)
For example:
mag_dict = refcat.get_dict_of_magnitudes('JohnsonV')
```

2. *Question2: Overlay the content of the catalog over a frame of a standard field contained within the catalog.*

```
Answer2:
Determine the database location of the reduced standard field fits file:
awe> q=(ReducedScienceFrame.DATE_OBS==DateTime.DateTimeFrom(date_obs)) \&
...     (ReducedScienceFrame.chip.name==chip)}
where date_obs is the date_obs listed for your dataset in
Table 1 of the sheet named Introduction exercises.
For example:
awe> q=(ReducedScienceFrame.DATE_OBS==DateTime.DateTimeFrom('2001-08-08 01:28:30.00')) &\
...     (ReducedScienceFrame.chip.name=='ccd53')
Make sure you get the Reduced Science Frame which was created most recently:
reduced = q.max('creation_date')
Then retrieve the standard field fits file to local disk:
reduced.retrieve()
Retrieve the standard star catalog in skycat format to local disk:
```

```
awe> refcat.make_skycat()
Start skycat:
awe> os.system('skycat&')
Use the "File" menu in skycat to load and display the image and the
"Data-servers>Local Catalogs"
menu to load and overplot the standard catalog catalog and overplot it.
```

1.2 Making a photometric catalog

Detailed information on making a photometric catalog is available at :

http://portal.astro-wise.org/man_howto_makephotcat/man_howto_makephotcat.shtml

1. *Question3: Derive a photometric catalog from the observation of a standard field using the* `PhotcatTask` *recipe.*

```
Answer3:
Determine the database location of the reduced standard star field fits file:
see Question 2.
Then determine the name of the raw standard star field fits file:
rawfilename=reduced.raw.raw_fits_data.filename
Perform the recipes and inspect the result:
photcattask=PhotcatTask(image=<imagename>, chip=<chip_name>, inspect=1)
photcattask.execute()

The results are stored in the database only if the object has the "commit" switched on:
photcattask = PhotcatTask(image=<imagename>, chip=<chip_name>, inspect=1, commit=1)
```

2. *Question4: do the same as above, but now using the basic building blocks of the photometric pipeline. To get information about how the* `PhotSrcCatalog` *object can be configured just ask* `photcat.process_params.info()`.

```
Answer4:
awe> frame = (ReducedScienceFrame.filename == 'reducedstandardstarfield.fits')[0]
awe> query = (AstrometricParameters.reduced == frame) &
...          (AstrometricParameters.is_valid == 1)
awe> astrom_params = query.max('creation_date')
awe> refcat = PhotRefCatalog.get()
awe> photcat = PhotSrcCatalog()
awe> photcat.refcat = refcat
awe> photcat.frame = frame
awe> photcat.weight = frame.weight
awe> photcat.astrom_params = astrom_params
awe> photcat.refcat.retrieve()
awe> photcat.frame.retrieve()
awe> photcat.weight.retrieve()
awe> photcat.make()
awe> photcat.inspect()
awe> photcat.commit()
```

3. *Question5: Write the contents of the* `PhotSrcCatalog` *object made previously to a skycat catalog. Then overlay this catalog together with the standard star catalog in skycat.*

```
photcat.make_skycat()
To overlay catalogs on fits images is explained in Answer2.
```

1.2.1 Making a photometric catalog - transformation tables

Four u', g', r' and i' sloan bands can be photometrically calibrated as shown in the previous questions. For all other bands a transformation table is needed to convert the photometric information in `awe` to such a band. Such transformation tables are already present for some commonly used band such as Johnson/Cousins B, V, R and I. More information on transformation tables is available at :

`http://portal.astro-wise.org/man_howto_transformation/man_howto_transformation.shtml`

1. *Question6: Retrieve a transformation table for the date of your dataset and filter from the database and explore its contents.*

```
Answer6: for date 2001-10-10 and filter #842 use:
awe> transform = PhotTransformation.get('2001-10-10', '#842')
Using awe> dir(transform) one can see the various options to explore the content of
the table. For example:
awe> print transform.timestamp_start,transform.timestamp_end
returns the begin and end date of the period for which the transformation table is valid.
```

2. *Question7: Rederive the photometric catalog made earlier now using a (fitting) transformation table as an extra dependency. Do this using the* `PhotcatTask` *recipe.*

```
Answer7:
Determine the name of the raw standard star field fits file: see Question3.
Perform the recipe using a transformation table and inspect the result:
photcattask=PhotcatTask(image=<imagename>, chip=<chip_name>, inspect=1, transform=1)
photcattask.execute()
To not only perform the operation but also store the results in the database one can alter
above commands to:
photcattask=PhotcatTask(image=<imagename>,chip=<chip_name>,inspect=1,transform=1,commit=1)
photcattask.execute()
For reference: to do the same using the basic building blocks which the PhotcatTask combine
in a 'script':
awe> frame = (ReducedScienceFrame.filename == 'reducedstandardstarfield.fits')[0]
awe> query = (AstrometricParameters.reduced == frame) &\
...          (AstrometricParameters.is_valid == 1)
awe> astrom_params = query.max('creation_date')
awe> refcat = PhotRefCatalog.get()
awe> transform = PhotTranformation.get('2001-10-10', '#842')
awe> photcat = PhotSrcCatalog()
awe> photcat.refcat = refcat
awe> photcat.transform = transform
awe> photcat.frame = frame
awe> photcat.weight = frame.weight
awe> photcat.astrom_params = astrom_params
```

```
awe> photcat.refcat.retrieve()
awe> photcat.frame.retrieve()
awe> photcat.weight.retrieve()
awe> photcat.make()
awe> photcat.inspect()
awe> photcat.commit()
```

1.3 Deriving the zeropoint

The documentation about this topic can be found at :

`http://portal.astro-wise.org/man_howto_zeropoint/man_howto_zeropoint.shtml`

1. *Question8: Derive the extinction and zeropoint for the night using the* `PhotomTask` *recipe. This requires that the results from the* `PhotcatTask` *recipe (or using the more elaborate way) have been committed to the database.*

```
Answer8:
If your night is '2001-10-10' and chip is 'ccd53' and the filter is '#842'
for your standard star observations then define the photomtask object as:
awe> phtask=PhotomTask(date='2001-10-10',chip='ccd53',filter='#842',commit=1,inspect=1)
Alternatively you can enter the filename of the raw fits file
of the standard star field e.g.,:
awe>phtask=PhotomTask(image='WFI.2001-10-11T09:24:40.791_4.fits',chip='ccd53',filter='#842'
Then execute it:
awe> phtask.execute()
```

2. *Question9: Do the same as above, but now using the basic building blocks of the photometric pipeline. Use a standard extinction coefficient from the database. To get information about how to configure the* `PhotometricParameters` *object, just type* `photom.process_params.info()`.

```
Answer9:
awe> extinct = AtmosphericExtinctionCoefficient.get('#842', 'ccd53')
which will give you the value for the WFI #842 (i.e., B-band) filter and chip 'ccd53'.
awe> extinct.value
show the resulting value.
Query the database for the photometric catalog you made in the previous exercises e.g.,:
(the date_obs for your standard star field is as in Question2)
awe> photcat=(PhotSrcCatalog.date_obs ==DateTime.DateTimeFrom(date_obs) &\
      ...    (PhotSrcCatalog.filter.name == '#842') &\
      ...    (PhotSrccatalog.chip.name='ccd53') &\
      ...    (PhotSrcCatalog.is_valid == 1)
awe> photom = PhotometricParameters()
awe> photom.photcat = photcat
awe> photom.extinct = extinct
awe> photom.make()
awe> photom.inspect()
awe> photom.commit()
```

1.4 Something more...

Bonus question: Given three or more observations of standard fields observed throughout the night, it is possible to roughly monitor the overall atmospheric stability. Write a script, using the *basic building blocks of the photometric pipeline, that provides such a monitoring. Assume as null-hypothesis that the night* was *stable, and that a good first approximation of the atmospheric extinction is given by a standard value for the atmospheric extinction coefficient.*