Step1: get ‘corrected frame’ FITS file.

Code: Explorer.py

Search the FITS file for one object, procedures:

```
Searching by using ‘navigator’:
1. go to http://skyserver.sdss.org/dr14/en/tools/chart/navi.aspx.
2. click 'explore'.
3. click 'FITS'.
4. click 'Corrected Frames' to see the description of the FITS file.
As described, field images are saved as ALLSKY float32[256,ny].
Interpolation vectors (bilinear interp) are stored as XINTERP and YINTERP.
```

Output: FITS file URLs for ‘corrected frames’ FITS.

[u_url, g_url, r_url, i_url, z_url]

SDSS provides query methods for the FITS URLs, but when I tried it, I got:

Your SQL command was:

```
select dbo.fGetUrlFitsCFrame(1237645876865991117,'r')
```

Your query output (max 500,000 rows):

Column1

This is the reason why I currently use this slower method to query the URLs.

Step2: Download FITS file. Get the cropping ‘image’ for one object.

The FITS file we got is for one field of the sky. If we want the ‘image’ for one object, we should crop the sky image by using the bound stored in atlasOutline.

Code: getRec.py

```
Methods:
      1. download the image files by using command line.
      2. Unzip .bz2 file we got in step 1.
```

```
    3. Get the bound information by querying atlasOutline (one by one).
    4. Cropping. method1: rescale the bound vector (done, the one you see)
    method2: interpolate the image, back to 2048*1489.(Working on it)
```

Output: the cropped image for one object.

[crop_u, crop_g, crop_r, crop_i, crop_z]

Each object has images with different size.

Note: For some objects, we will see “photometric not reliable” on its navigator website. These objects would not have a shape as [256, ny], usually the shape is [196, 256]. These objects currently shown as ‘[None]’ for u, g, r, i, z in our dataset.

Step3: Get the cropped image for objects in one field.

Because the shape of the image data for each object differs, I saved them in hdf5 file.

Filename: “GALAXY.hdf5”

Format:

Objects are in separated database named by its galaxy_id.

Inside, there are 5 columns named as ‘u’, ‘g’, ‘r’, ‘i’, ‘z’.

The data for each band is saved as numpy array.

Get the data:

```
import h5py
f = h5py.File("GALAXY", 'r')
f[“galaxy_id”].attrs[‘u’]
f.close() # important
```

Step4: Get the cropped image by using method 2.

In the corrected-frame's documentation, it says: These dn values are in the same units as the "data numbers" stored by the raw data files that come off the instrument. They are related to the detected number nelec of photo-electrons by: nelec= dn*gain. (https://dr14.sdss.org/datamodel/files/BOSS_PHOTOOBJ/frames/RERUN/RUN/CAMCOL/frame.html). Also, I found the corrected-frames data does not look like the one they reported on their website as magnitude-u, g, r, i, z. Therefore, we may need to interpolate the image and do some calibration. The method is unclear. I am reproducing these procedures with the guide provided by

https://acrider.wordpress.com/2015/01/20/extracting-the-data-number-image-dn_image-from-sdss/