

Telecon May 31

Outline

- Implement the class TileFileCache

Questions

- Regarding tile drawing using affine transformation
(<https://github.com/ad1995/HIPS-to-Py/issues/3>)
 - Here's some step-by-step suggestions how to draw your first HIPS tile onto an all-sky image:
 - Use this example all-sky image (in AIT projection):
from astropy.io import fits
hdu_list =
fits.open('https://github.com/gammapy/gammapy-extra/blob/master/datasets/cata
logs/fermi/gll_psch_v08.fit.gz?raw=true')
hdu_list.info()
 - Open it up with <http://aladin.u-strasbg.fr/#AladinDesktop>
Find the Galactic center and the crab nebula, maybe note the pixel and sky coordinates roughly
 - Then open it up with Python, plot the sky image using wcsaxes. Put markers at the positions of the Galactic center and Crab nebula (as an exercise to understand the world coordinates (Galactic in this case) and pixel coordinates in that image.
 - Then the next task is to pick one HIPS tile to draw.
Let's use this tile (order 3, pixel 450, near the Galactic center, covering a large part of the sky):
[https://github.com/hipsipy/hips-extra/blob/master/datasets/samples/DSS2Red/Nor
der3/Dir0/Npix450.fits?raw=true](https://github.com/hipsipy/hips-extra/blob/master/datasets/samples/DSS2Red/Nor
der3/Dir0/Npix450.fits?raw=true)
 - Use function
<http://healpy.readthedocs.io/en/latest/generated/healpy.boundaries.html>
(in healpy) to retrieve position on the sky for the 4 tile corners (in xyz spherical coordinates.
 - Call
<http://healpy.readthedocs.io/en/latest/generated/healpy.pixelfunc.vec2ang.html> to convert these xyz to theta / phi.
 - The DSS2Red HIPS dataset is in RADEC, so you'll get sky coordinates like this:
radec = SkyCoord(ra=phi, dec=pi/2 - theta, unit='radian', frame='icrs')

- The Fermi all-sky image is in Galactic coordinates, so you'll have to convert to Galactic coordinates like this:
galactic = radec.galactic
Or equivalently:
galactic = radec.to_system('galactic')
- Plot the positions with wcsaxes -> email us if you have this plot or let us know if you have issues getting to here.
- Convert these galactic world coordinates to pixel coordinates (x, y).
wcs.wcs_world2pix(corners_sky_coordinates)
-> Again: print and plot the coordinates to check where they fall on the sky image.
- Then the next step is to compute the affine transformation using either of these two:
 - https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.ndimage.interpolation.affine_transform.html
Usage examples:
<https://stackoverflow.com/questions/20161175/how-can-i-use-scipy-ndimage-interpolation-affine-transform-to-rotate-an-image-ab>
 - <http://scikit-image.org/docs/dev/api/skimage.transform.html#skimage.transform.AffineTransform>
Examples:
http://scikit-image.org/docs/dev/auto_examples/xx_applications/plot_geometric.html
Another example:
http://www.scipy-lectures.org/packages/scikit-image/auto_examples/plot_features.html
 - You'll have to look for tutorial examples / try to understand the affine transformation (try to do this as a separate exercise).
- Then apply the affine transformation onto the HIPS data array -> plot the result.
 - You use the HIPS tile corner coordinates that you computed above to initialise the affine transformation (or in the case of scipy to pass those in the appropriate format to the scipy function)
 - Input data to the affine transformation is the 512 x 512 numpy array (the HIPS pixel data)
 - Output is a numpy array matching the shape of your all-sky sky image (something bigger)

○
○
○

- Also:
 - <http://healpy.readthedocs.io/en/latest/generated/healpy.pixelfunc.pix2ang.html>
- Implementation of classes listed in the API proposal document (<https://github.com/hips/hips/blob/master/notes/API-proposal.md>)
 - This is a good place to start: <https://github.com/hips/hips/pull/11>
 - What classes / methods do we need?

Basic idea: implement ~ 10 classes that each are simple and do one thing and do it well. Higher-level algorithms like `make_sky_image` below) then use these classes in concert:

Implement this (very similar to Aladin Lite) in `'hips.tiles'` :

 - HipsDescription -- maps properties file to attributes
 - We need at least the base URL, the `hips_frame`, `hips_norder`, `hips_tile_format` (JPG, FITS)
 - Use this file for writing tests:
 - Example of a properties file: <http://alasky.unistra.fr/DSS/DSSColor/properties>
 - Read (i.e. parse) into a `collections.OrderedDict`, store this as data member on the HipsDescription class.
 - Expose derived things (if any) as properties or methods.
 - Hips -- constructor: `Hips(HipsDescription)`
 - `setServer(URL)`: set the base URL
 - `getTileURL(order, ipix, tile_format)`
`getTileURL(order=7, ipix=12345, format='fits')→`
<http://alasky.../DSS2/red/Norder7/Dir10000/Npix12345.fits>
 - HipsTileCache
 - tbd
 - HipsTileFetcher
 - `addTileToQueue(hips_tile, success_callback_function)`
 - `reset()`: remove all tiles still in the queue
 - HipsTile / HipsTileMeta
 - Hold data and metadata for a single tile
 - Interacts with renderer
 - Or we could come from the use cases / caller side:


```
geometry = (wcs + array shape nx / ny)
hips_survey = ('dss', ...)
from hips import make_sky_image
image = make_sky_image(geometry, hips_survey)
```
 - `def make_sky_image(geometry, hips_survey):`
 - First compute list of tile needed ???
`tile_specs = list(HipsTileMeta)`
 - Second fetch the tiles
`cache = HipsTileCache()` # local folder on disk (or maybe optionally in

```
memory later)
fetcher = HipsTileFetcher(server_info, cache)
fetcher.fetch(tile_specs)
```

- Last step: `image = HipsTileRenderer(tile_specs, cache, geometry, tiles).render()`
- Updated blog. Suggestions / Improvements.
(<https://github.com/ad1995/HIPS-to-Py/blob/master/documents/blog.md>)
- Using a third-party application for tracking / assigning tasks. Suggestions: Trello, Slack.
 - Christoph: I suggest we use the Github-built-in features
 - Issues: <https://github.com/hipspy/hips/issues>
(if you want, we can file issues instead of putting suggestions what to implement in the coming days / weeks for you during our calls)
 - Milestones: <https://github.com/hipspy/hips/milestones>
(for now, I've created version 0.1 and 0.2 milestones)
 - Projects: <https://github.com/hipspy/hips/projects>
(not used yet)