

On-the-fly generation of JPEG HiPS tiles



Thomas Boch

IVOA Interop, College Park, MD, Apps 2



□ Outline

- Generation of JPEG HiPS tiles
 - context: streamlining Pan-STARRS HiPS generation
 - implementation
 - Apache HTTPD configuration
 - benchmarking
 - extensions to all existing CDS HiPS
 - demo page
 - in production for Pan-STARRS z band
- HiPS standards updates
 - description of additional parameters
- HiPS tiles mixing service
 - prototype
 - demo

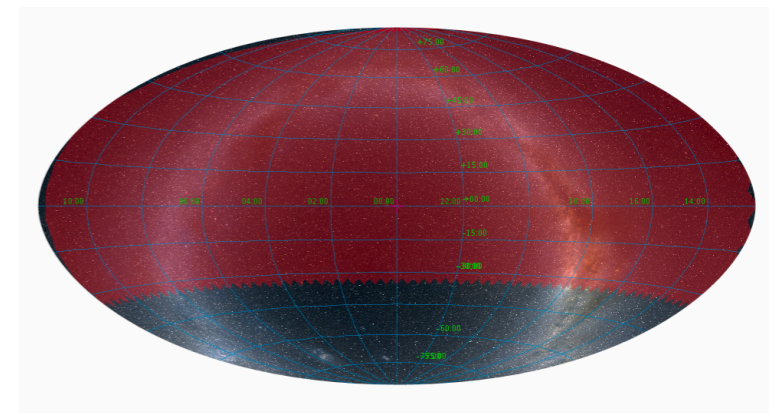
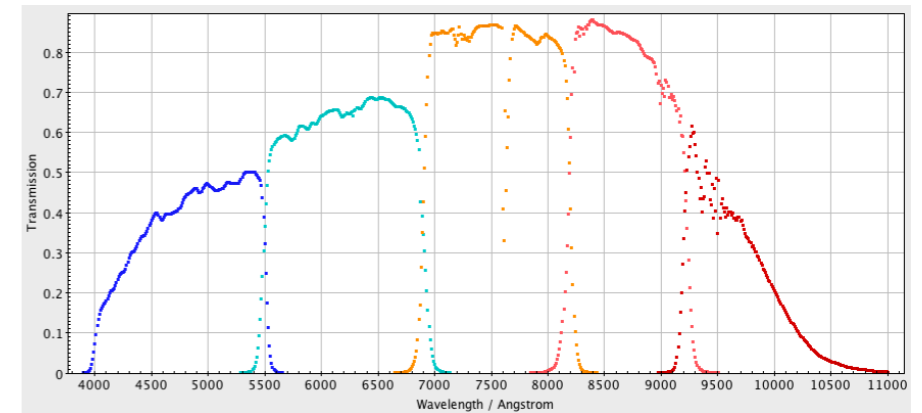
□ Pan-STARRS HiPS

- Pan-STARRS PS1 images

- 5 bands: g, r, i, z, y
- coverage: 3/4 of the sky
- *RICE* compressed
- resolution: 0.25"/pixel
- 15 TB per band

- HiPS generation

- resolution: 200 mas (HEALPix order 20)
- with 512x512 tiles: 47 million tiles to be generated
- 10 trillion pixels per band



□ Pan-STARRS HiPS generation

Step	Tool	Duration	
Download from STScI	<code>parallel wget</code>	4 days	14 TB
FITS tiles generation	<code>Hipsgen</code>	20 days	18 TB
JPEG tiles generation	<code>Hipsgen</code>	20 days	6 TB
Transfer to production machine	<code>parsync</code>	3 days	

□ JPEG HiPS tiles generation - implementation

- Web service developed in Python
 - Falcon framework (WSGI compatible)
 - heavy-lifting done by `astropy[.visualisation]`
 - read FITS image tile
 - apply image cuts retrieved from HiPS *properties*
 - executed on `gunicorn` WSGI server



```
def apply_stretch(input_image, stretch='linear', min_cut=None, max_cut=None):  
    from astropy.visualization import *  
    image_normalizer = simple_norm(input_image, stretch=stretch, min_cut=min_cut,  
max_cut=max_cut)  
    image_scaled = image_normalizer(input_image)  
    image_scaled = np.flipud(image_scaled)  
  
    return image_scaled
```

□ Apache rewrite rules

<http://alasky../hips-image-services/convert?url=/PS1/z/Norder9/Dir0/Npix4564.jpg>

```
# Apache rewrite rules ...  
RewriteMap hips-list-jpeg-gen "txt:/data/list-hips-for-jpeg-  
generation.txt"  
  
RewriteCond ${hips-list-jpeg-gen:$1|NOT-FOUND} !=NOT-FOUND  
RewriteRule ^/(.*)/Norder.*/*Dir.*/*Npix.*.jpg$ /hips-image-services/  
convert?url=%{REQUEST_URI}&%{QUERY_STRING} [P]
```

<http://alasky../PS1/z/Norder9/Dir0/Npix4564.jpg>

- Access to these server-side generated JPEG tiles is transparent for HiPS clients

□ Benchmarking

- Cloning of a small part of Pan-STARRS z JPEG tiles
 - using `Hipsgen`
- Result
 - 250,000 tiles retrieved in 18 minutes
 - 240 tiles/second
 - *gunicorn* server configured with 20 workers
 - tile **generation** time: **100ms** on average

□ Extensions

- allowing user to select *min/max cuts, color map, stretch function*
 - requires additional parameters added to the query string
 - `http://alasky../PS1/z/Norder9/Dir0/Npix4564.jpg?cmap=cubehelix&stretch=log&min_cut=.42`
 - support for those parameters implemented in Aladin Lite

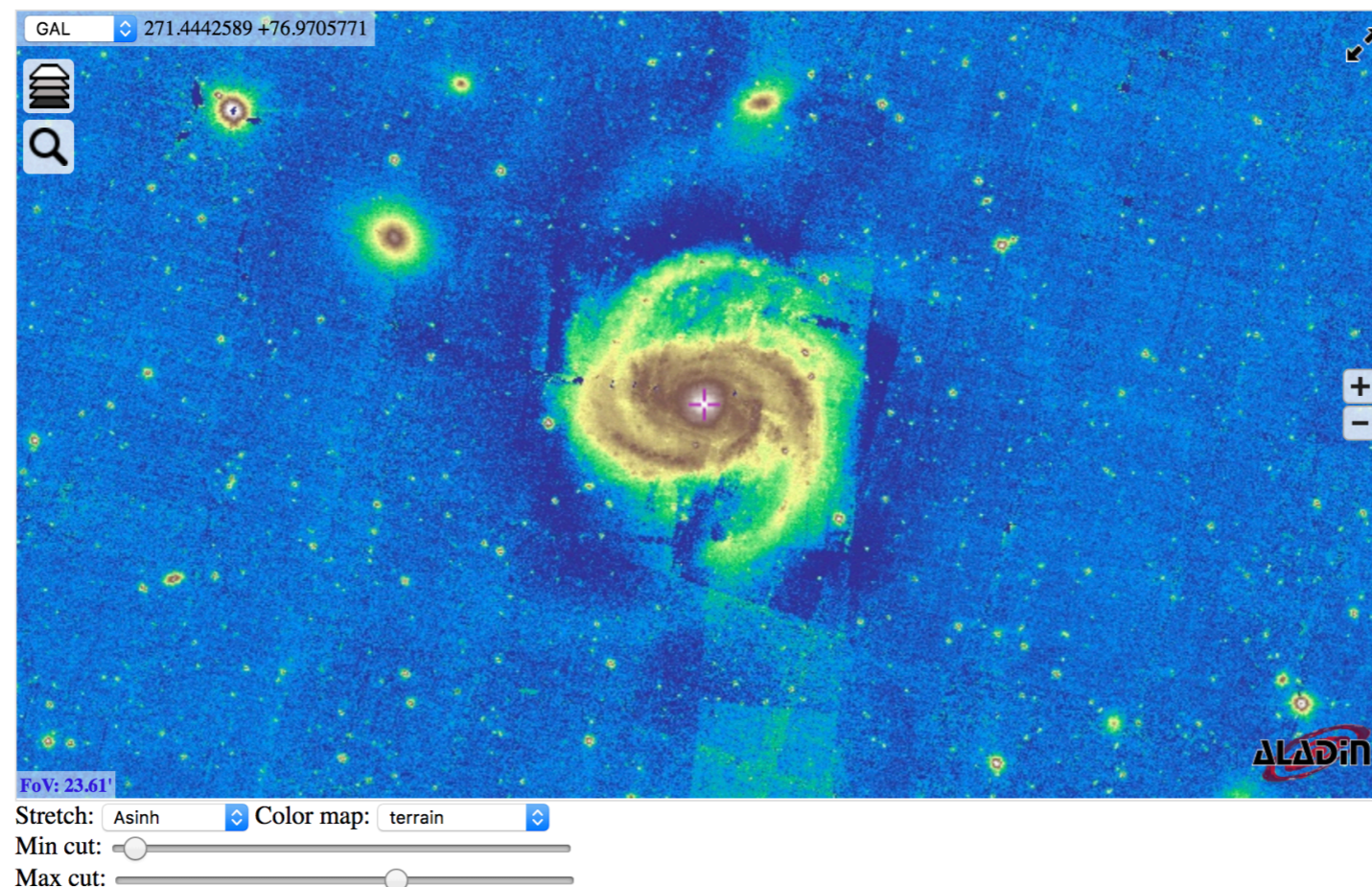
```
aladin.setBaseImageLayer(  
    aladin.createImageSurvey(<id>, <name>,  
        "http://alasky../PS1/z", "equatorial", 10,  
        {additionalParams: 'cmap=cubehelix&stretch=log&...' }));
```
- extensions to all HiPS served by CDS
 - use pre-generated JPEG tile if no parameter specified
 - generate on-the-fly otherwise
- support of remote HiPS
 - <http://alasky.u-strasbg.fr/http://skies.esac.esa.int/Herschel/PACS160/Norder5/Dir0/Npix7673.jpg?cmap=terrain>

□ Demonstration page

- aladin.unistra.fr/AladinLite/showcase/dynamic-tiles-generation/

HiPS tiles generation on the fly

Play with the sliders or change the color map to see the tiles being updated

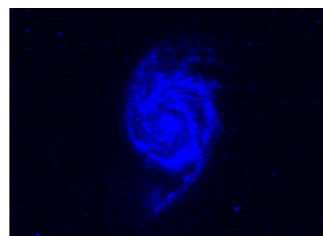
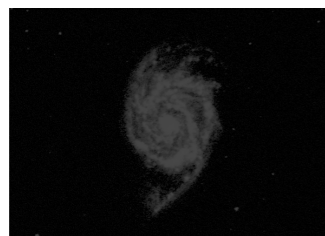
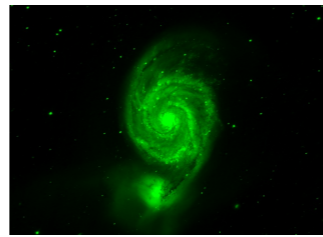
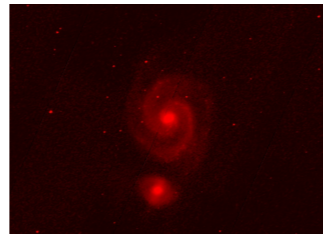
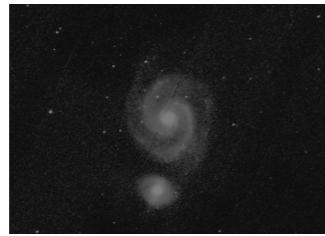


□ Updates to HiPS standard

- Support of parameterized HiPS URL requires updates to HiPS document
- How to describe parameters?
 - *properties* file
 - For each parameter
 - name
 - description
 - type
 - required/optional
 - range of allowed values
- which syntax?
 - HiPS *properties* is a **key = value** text file

□ HiPS tiles mixing service

- Idea: combine data from different HiPS into a single image tile



- Server-side implementation
 - no support for FITS tiles in Aladin Lite (yet)
 - extending the service previously shown
 - alternative: client-side mixing
- Prototype demo

□ Conclusion

- Generation on-the-fly of JPEG HiPS tiles in production at CDS
- Minimal changes for the client applications
- Parameterized HiPS URLs will require updates to the HiPS standard
- Server-side generation of HiPS tiles promising for data exploration and analysis
 - FITS HiPS tiles are needed for scientific exploitation