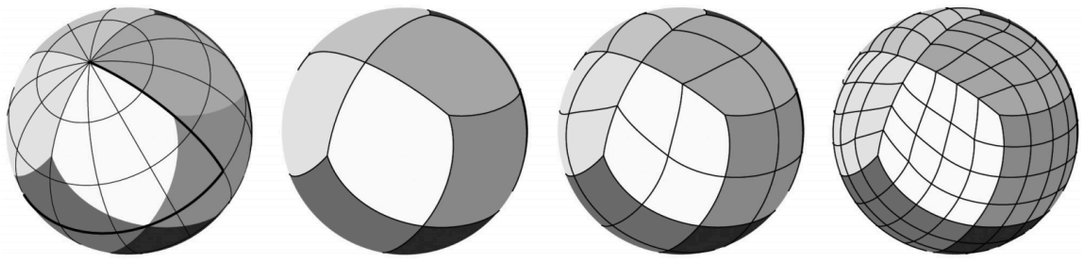
Combine HiPS at your will!

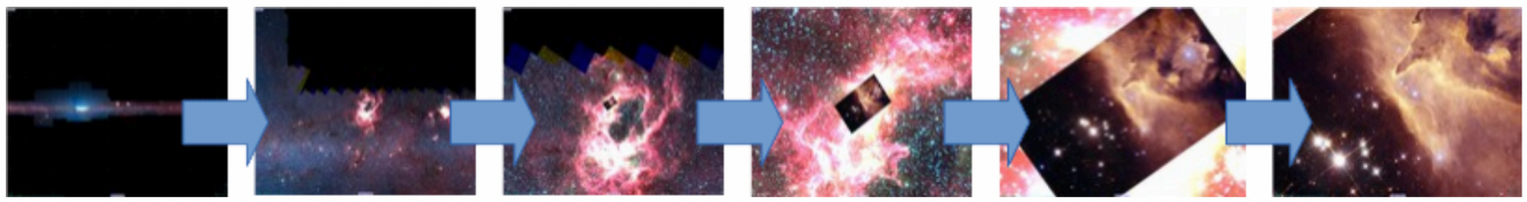
Thomas Boch

IVOA Apps 2 - June 5 2025

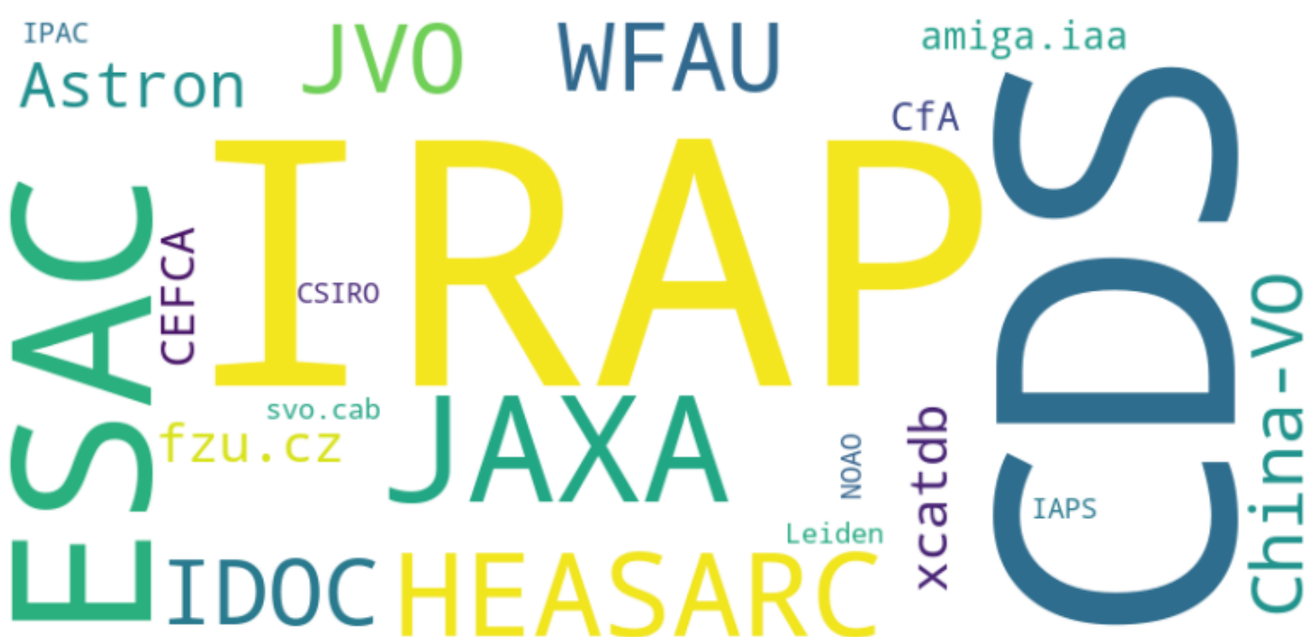
HiPS: the format

- **Hi**erarchical **P**rogressive **S**urveys
- CDS implementation of slippy maps suited for astronomy
  - started in 2009
- HEALPix-based
  - no singularity at poles
  - minimal deformation overall
- preserves original dynamic range (FITS tiles)
- Described in [2015A&A...578A.114F]
- IVOA standard since 2017





- 1400+ published HiPS
- By 24 publishers
- 820 TB
  - 2 billion tiles



Motivations

- Benefit from the best of each individual HiPS dataset
  - Extend coverage
  - Resolution

- Without loading each HiPS client-side
  - Features available in each client
  - Save bandwidth

- In one new resulting HiPS product
  - Conform to HiPS standard

2 combinations modes

- **By order of priority**

- **Color HiPS from 3+ HiPS**

By order of priority

By order of priority

By order of priority

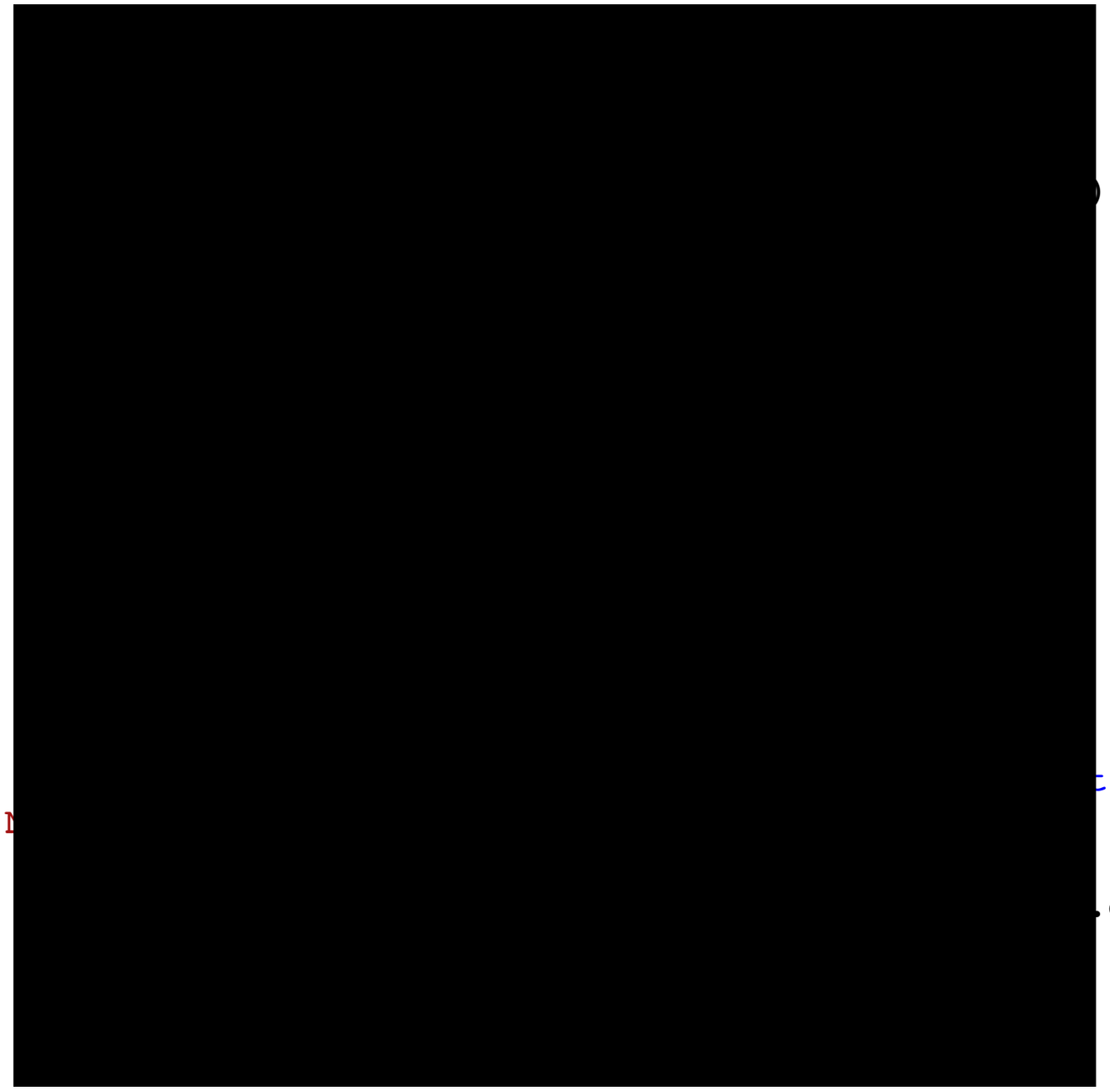
- We consider JPEG/PNG tiles

```
from PIL import Image
import numpy as np

  tile = np.zeros((512, 512, 4), dtype=np.uint8)
  for idx, hips in enumerate(hipses):
      mask = (tile[..., 3] == 0)
      if np.sum(mask)==0: # all done?
          return tile
```

```
      tile_format = hips_params[hips]['format']
      if 'png' in tile_format:
          tile_format = 'png'
      else:
          tile_format = 'jpg'

      root = hips_params[hips]['root']

      tile_path = f'{root}/Norder{order}/Dir{int(10000 * (ipix//10000))}/
Npix{ipix}.{tile_format}'

      cur_tile = np.array(Image.open(tile_path).convert('RGBA'))

      tile[mask] = cur_tile[mask]
```

By order of priority

- We consider JPEG/PNG tiles



```
(10000 * (ipix//10000))}/
.convert('RGBA'))
```

By order of priority

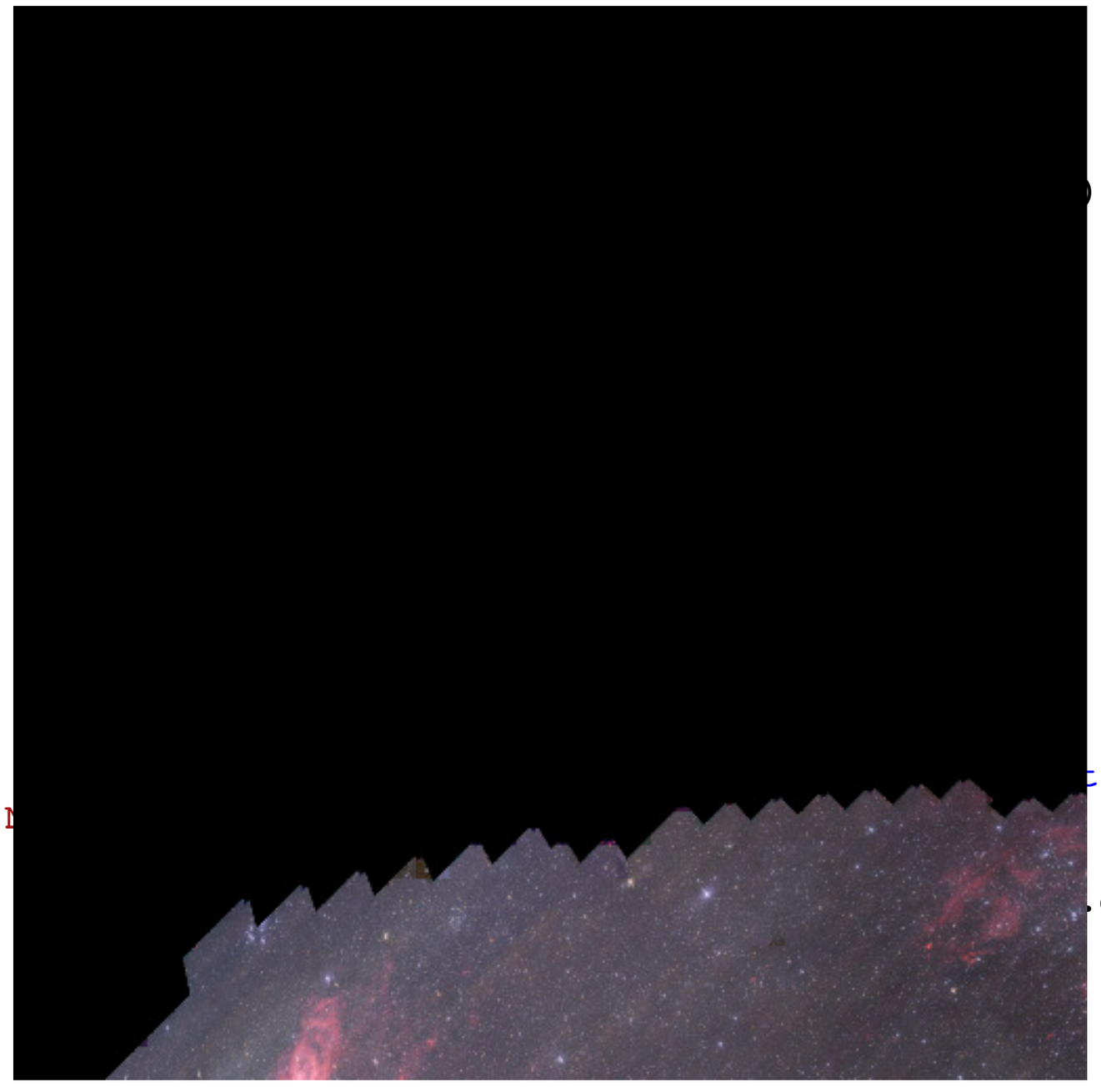
- We consider JPEG/PNG tiles

```
(10000 * (ipix//10000))}/

.convert('RGBA'))
```

VPHAS
DECaPS
DES DR2
DESI
SkyMapper
DSS2

By order of priority

- We consider JPEG/PNG tiles

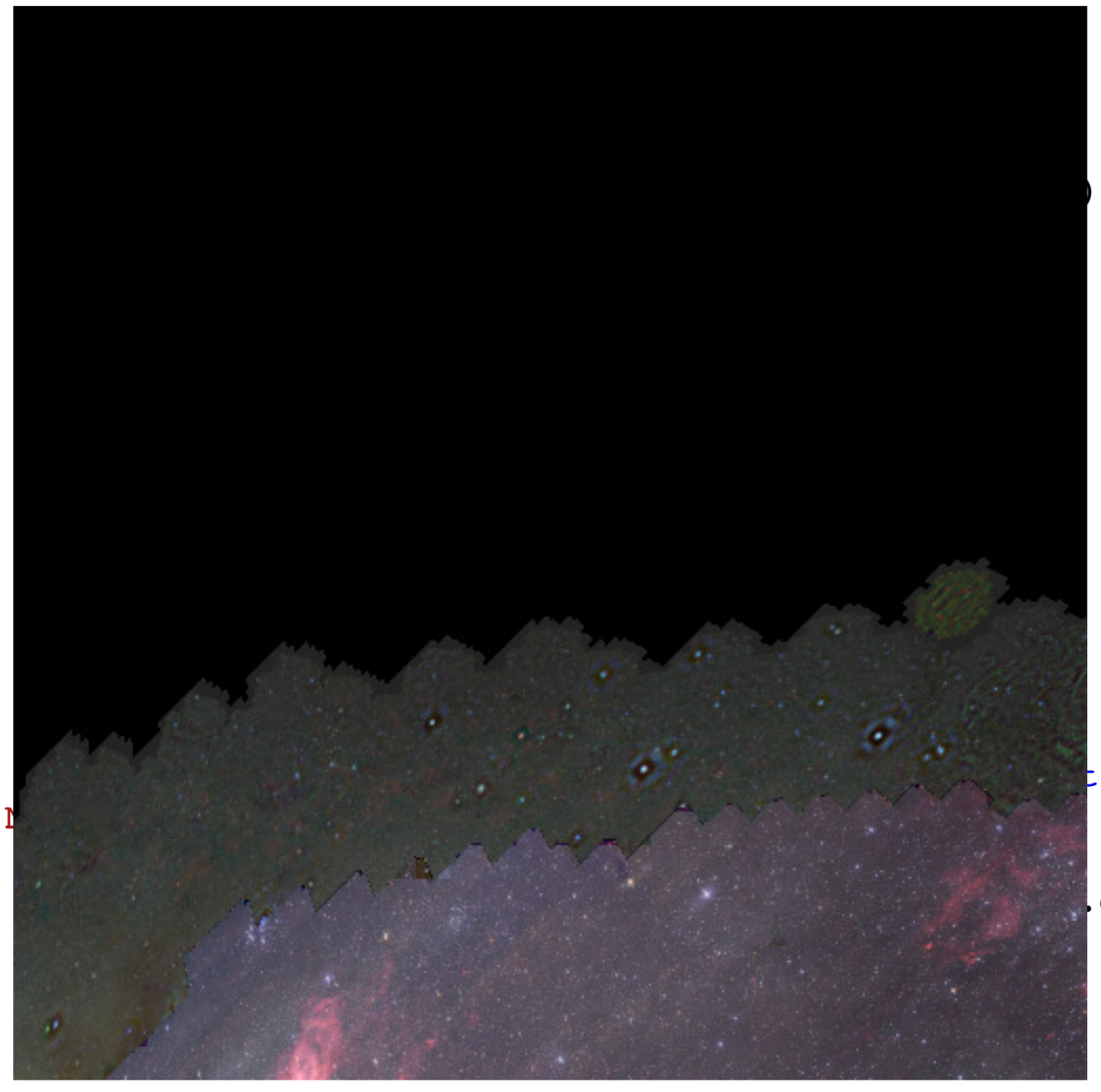


```
(10000 * (ipix//10000))}/

.convert('RGBA'))
```

VPHAS
DECaPS
DES DR2
DESI
SkyMapper
DSS2

By order of priority

- We consider JPEG/PNG tiles

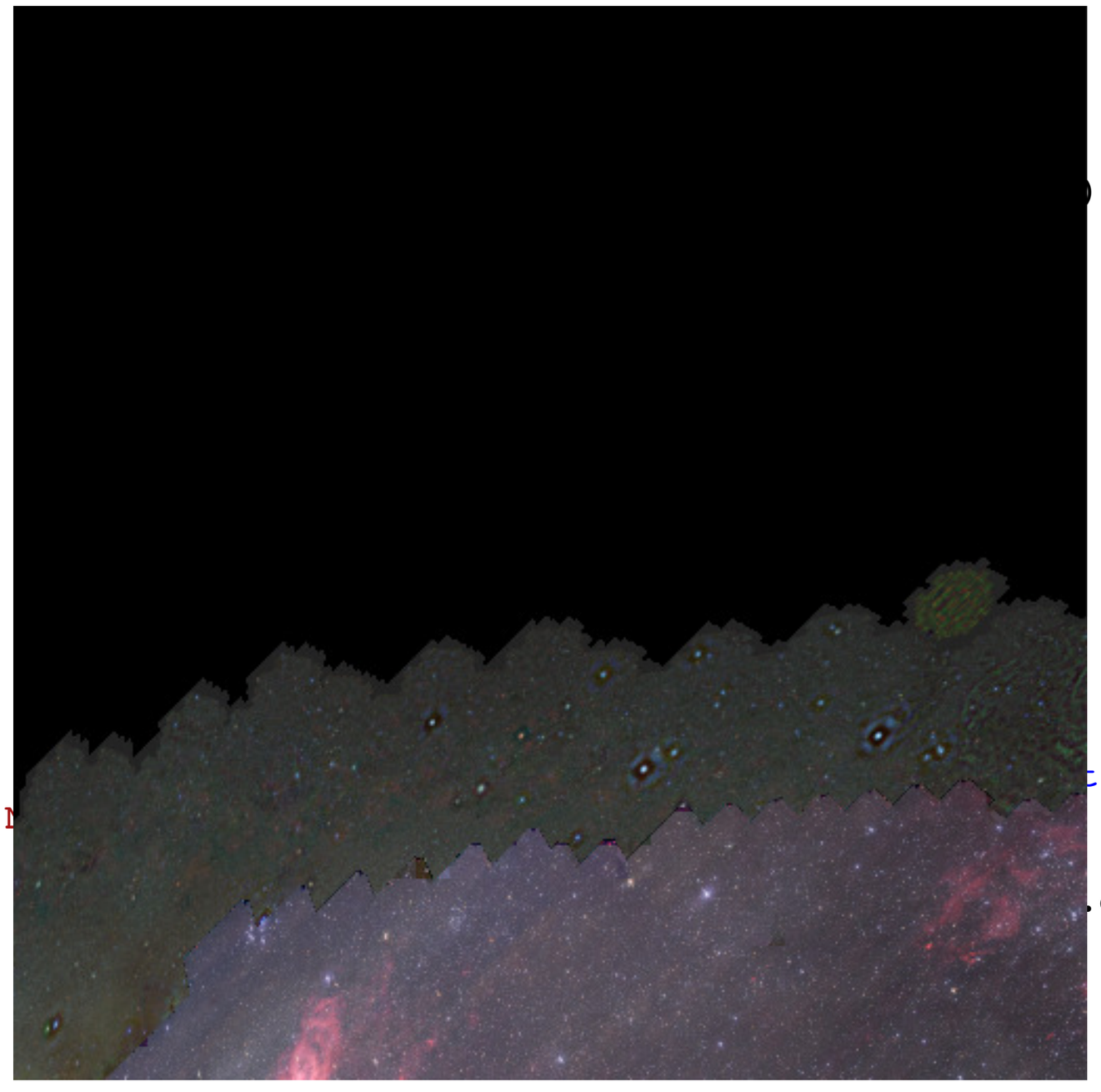


VPHAS
DECaPS
DES DR2
DESI
SkyMapper
DSS2

```
(10000 * (ipix//10000))}/

.convert('RGBA'))
```

By order of priority

- We consider JPEG/PNG tiles

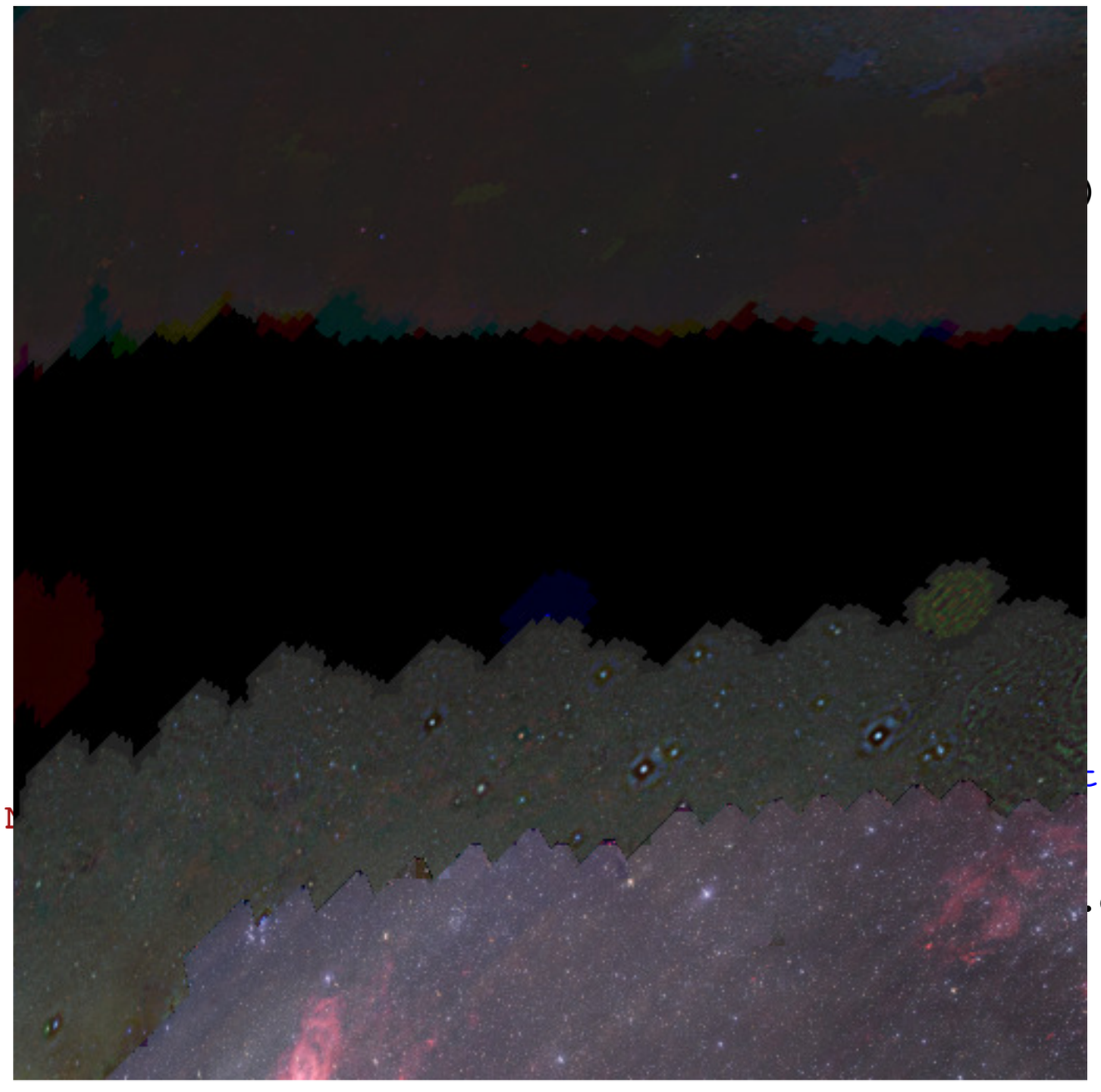


VPHAS
DECaPS
DES DR2
DESI
SkyMapper
DSS2

```
(10000 * (ipix//10000))}/

.convert('RGBA'))
```

By order of priority

- We consider JPEG/PNG tiles

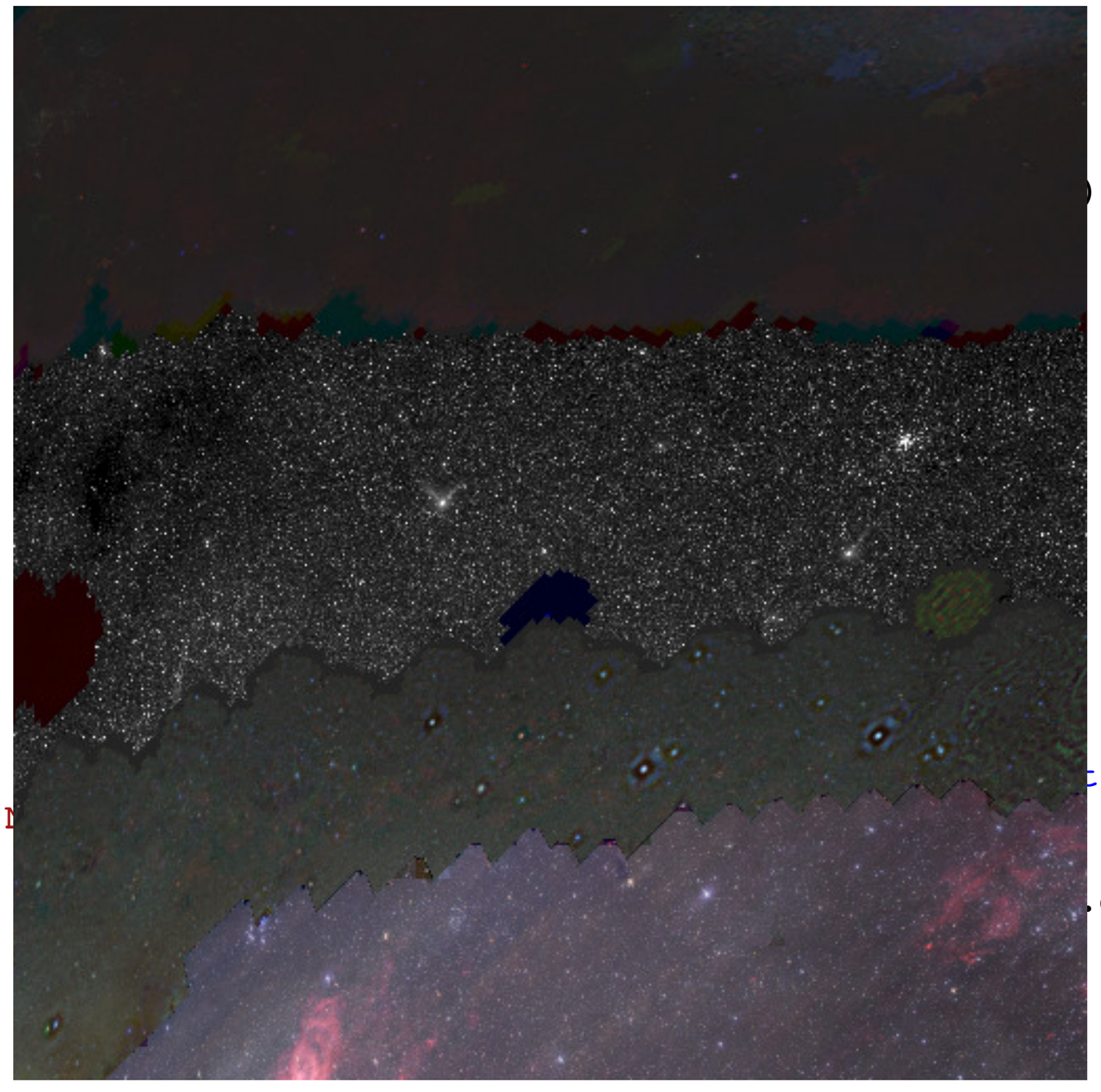


VPHAS
DECaPS
DES DR2
DESI
SkyMapper
DSS2

```
(10000 * (ipix//10000))}/

.convert('RGBA'))
```

By order of priority

- We consider JPEG/PNG tiles

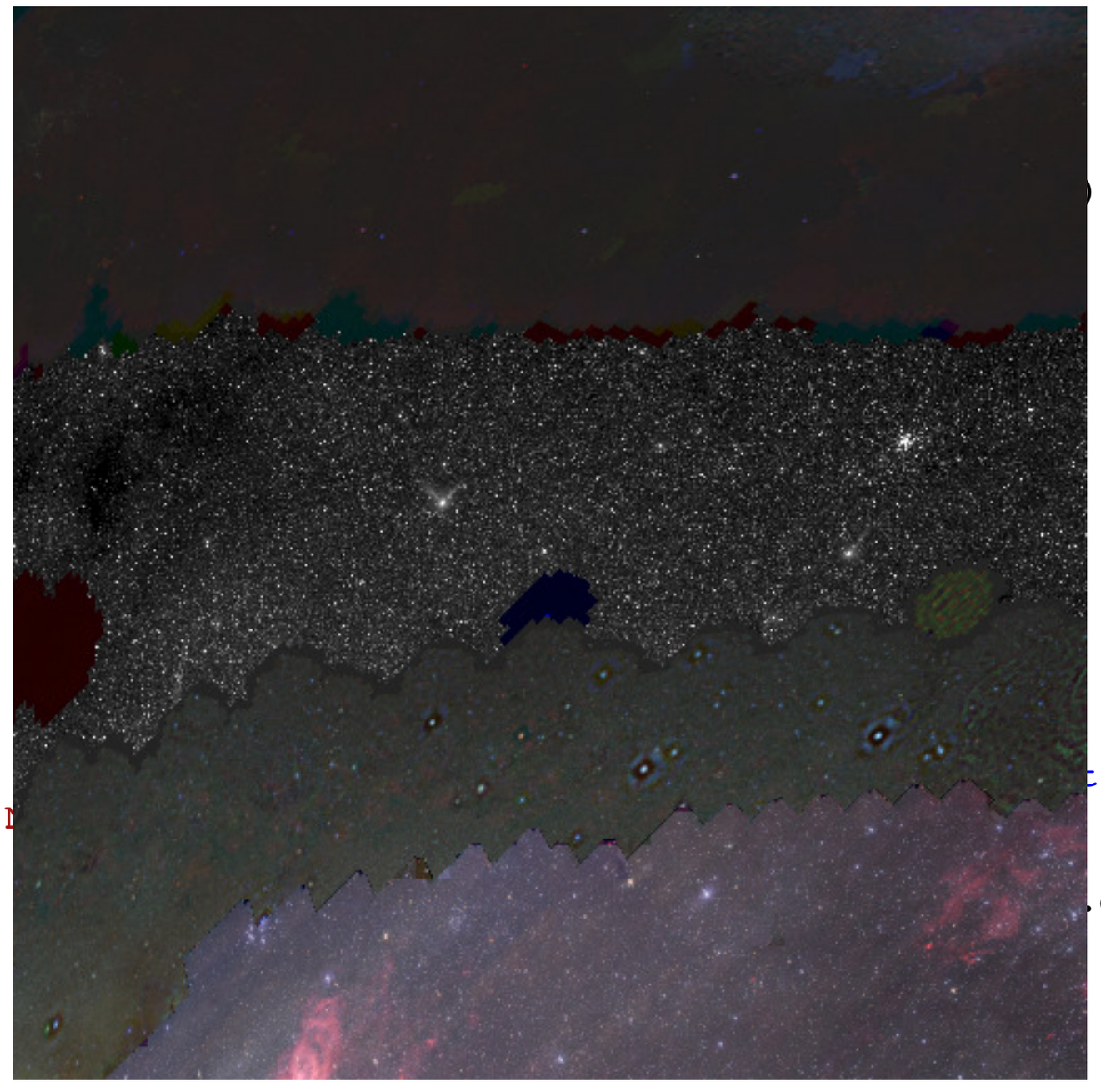


VPHAS
DECaPS
DES DR2
DESI
SkyMapper
DSS2

```
(10000 * (ipix//10000))}/

.convert('RGBA'))
```

By order of priority

- We consider JPEG/PNG tiles



VPHAS
DECaPS
DES DR2
DESI
SkyMapper
~~DSS2~~

```
(10000 * (ipix//10000))}/

.convert('RGBA'))
```

MOC (spatial coverage)

- Union of spatial coverages

```
moc = MOC.new_empty(max_order)
for hips in hipses:
    cur_moc =
MOCServer.query_region(return_moc="smoc",
criteria=f'ID={hips}')
    moc = moc.union(cur_moc)
```

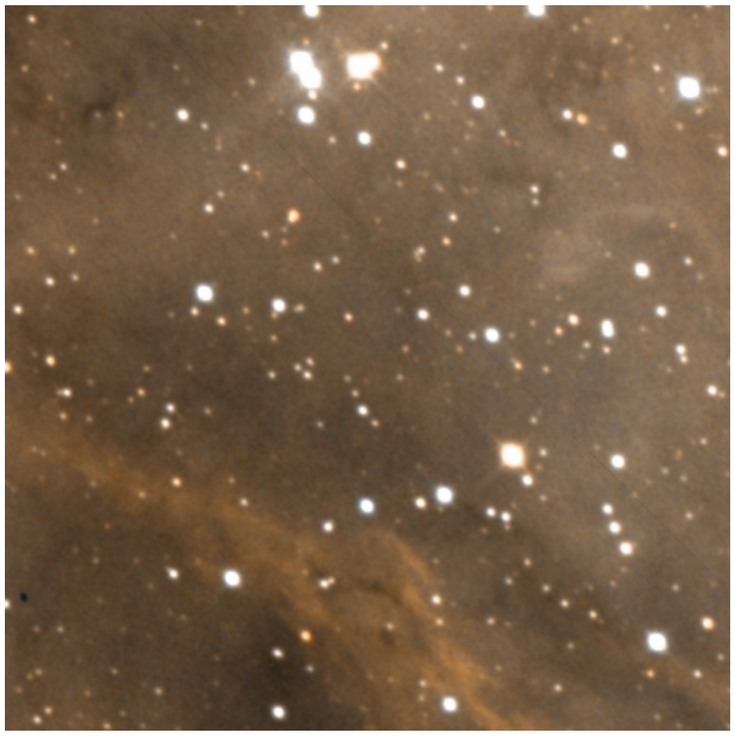
Properties file

```
obs_title              = HiPS combined
hips_frame             = equatorial
hips_order             = 13
hips_tile_width        = 512
hips_initial_ra        = 266.416817
hips_initial_dec       = -29.0078249
hips_initial_fov       = 360
dataproduct_type       = image
hips_tile_format       = png
dataproduct_subtype    = color
obs_description = Created from HiPS CDS/P/Euclid/Q1/NISP.H,CDS/P/Euclid/
Q1/NISP.J,CDS/P/Euclid/Q1/NISP.Y,CDS/P/Euclid/Q1/VIS
```

Heterogeneous orders

- order_combined = max(order_hips*1*, …, order_hips*n*)

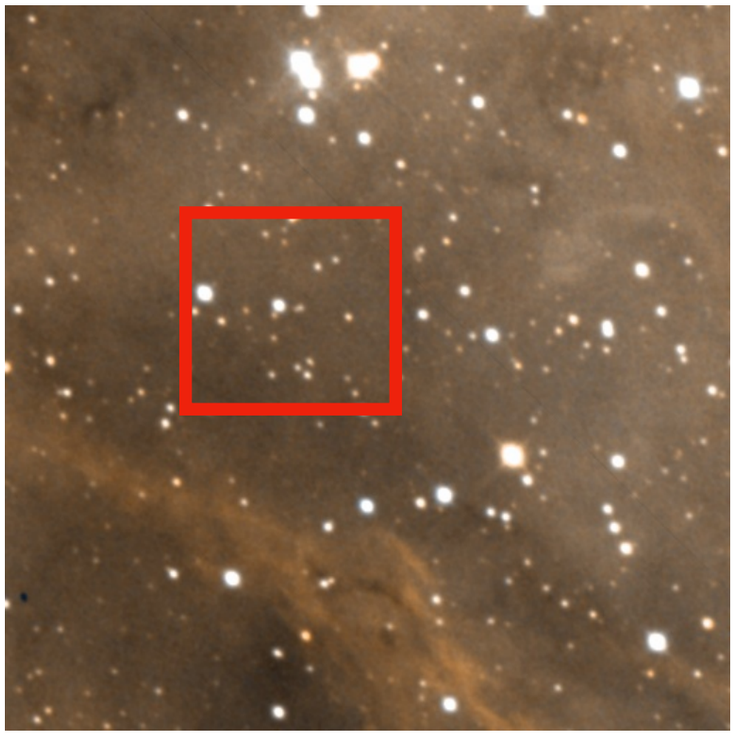
  If order HiPS < order max
  DSS2 ==> request tile *12345* at order *11*
  Retrieve tile 12345//16 at DSS2 max order (9)
  Cut the needed part and expand it



Heterogeneous orders

- order_combined = max(order_hips*1*, …, order_hips*n*)

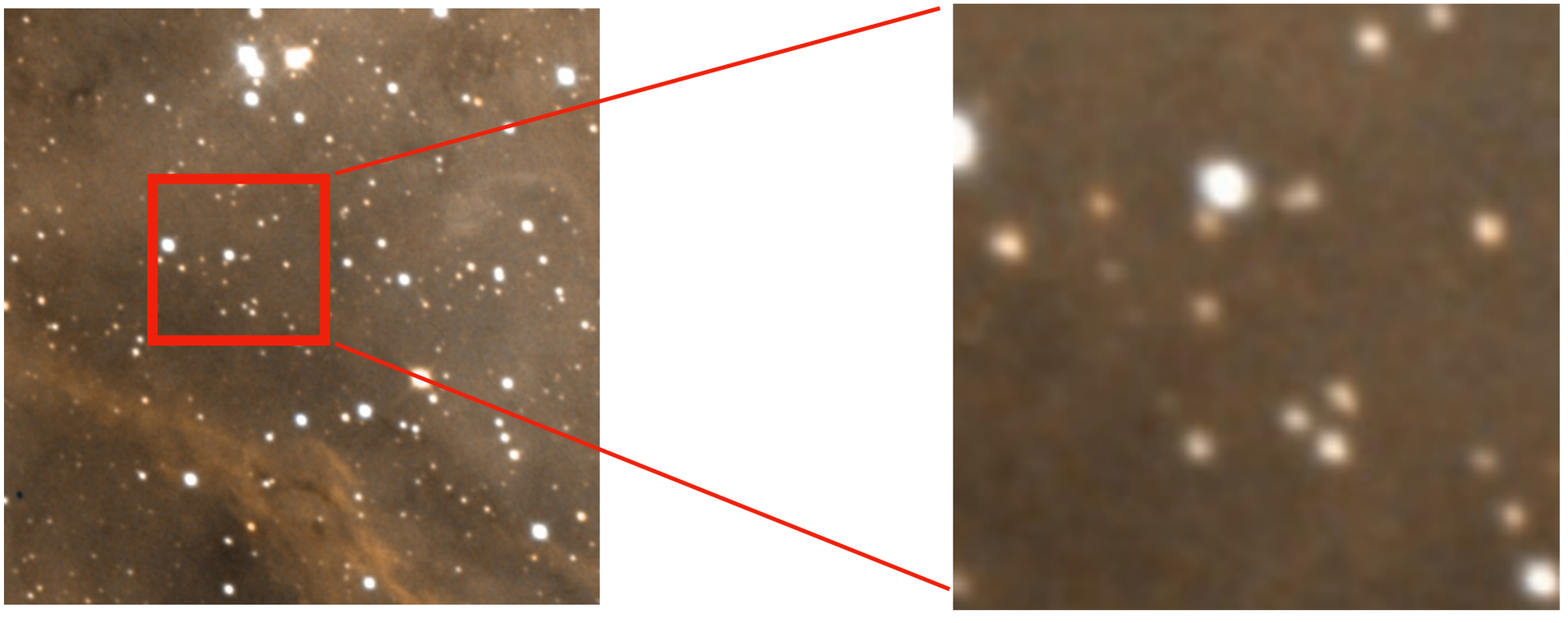
  If order HiPS < order max
  DSS2 ==> request tile *12345* at order *11*
  Retrieve tile 12345//16 at DSS2 max order (9)
  Cut the needed part and expand it



Heterogeneous orders

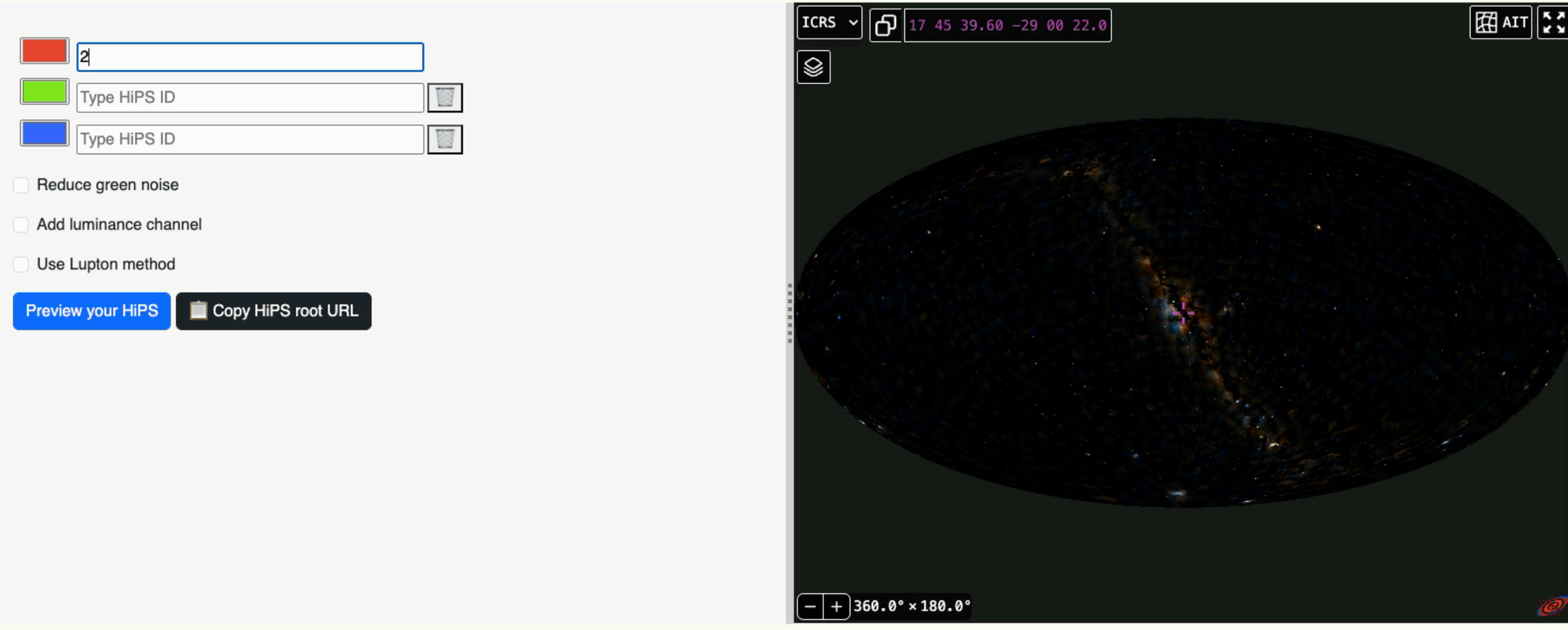
- order_combined = max(order_hips*1*, …, order_hips*n*)

  If order HiPS < order max
  DSS2 ==> request tile *12345* at order *11*
  Retrieve tile 12345//16 at DSS2 max order (9)
  Cut the needed part and expand it



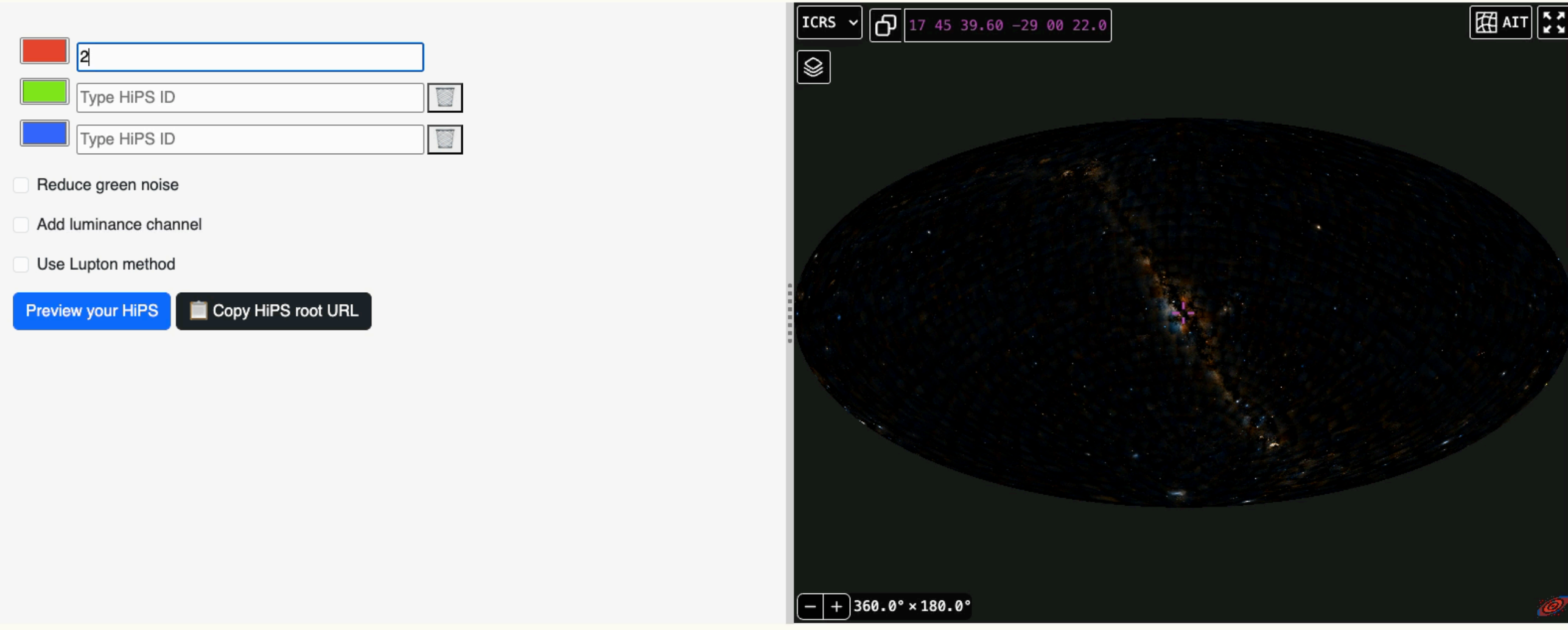
Virtual color HiPS

HiPS Mixer - 🌈 Create a color HiPS from 3+ HiPS



Virtual color HiPS

HiPS Mixer - 🌈 Create a color HiPS from 3+ HiPS



Color HiPS creation

- We work from FITS files
- Compute contribution of each channel

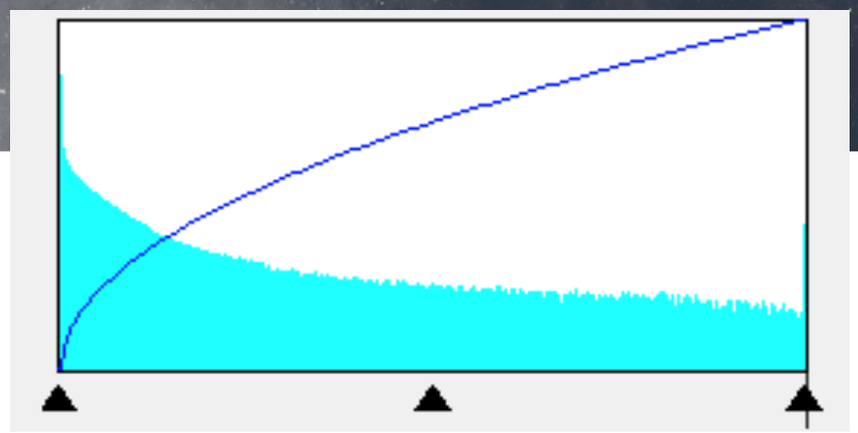
HiPS	Red	Green	Blue
2MASS/K	1.0	0.0	0.0
2MASS/H	0.0	1.0	0.0
2MASS/J	0.0	0.0	1.0
VPHAS Hα	1.0	0.75	0.8

$$\text{Red} = \frac{1 \cdot \text{2MASS/K} + 1 \cdot \text{VPHAS H}\alpha}{2}$$
$$\text{Green} = \frac{1 \cdot \text{2MASS/H} + 0.75 \cdot \text{VPHAS H}\alpha}{1.75}$$
$$\text{Blue} = \frac{1 \cdot \text{2MASS/J} + 0.8 \cdot \text{VPHAS H}\alpha}{1.8}$$

- Weighted mean

  (⚠️ taking NaN/missing into account)

Cuts and stretch



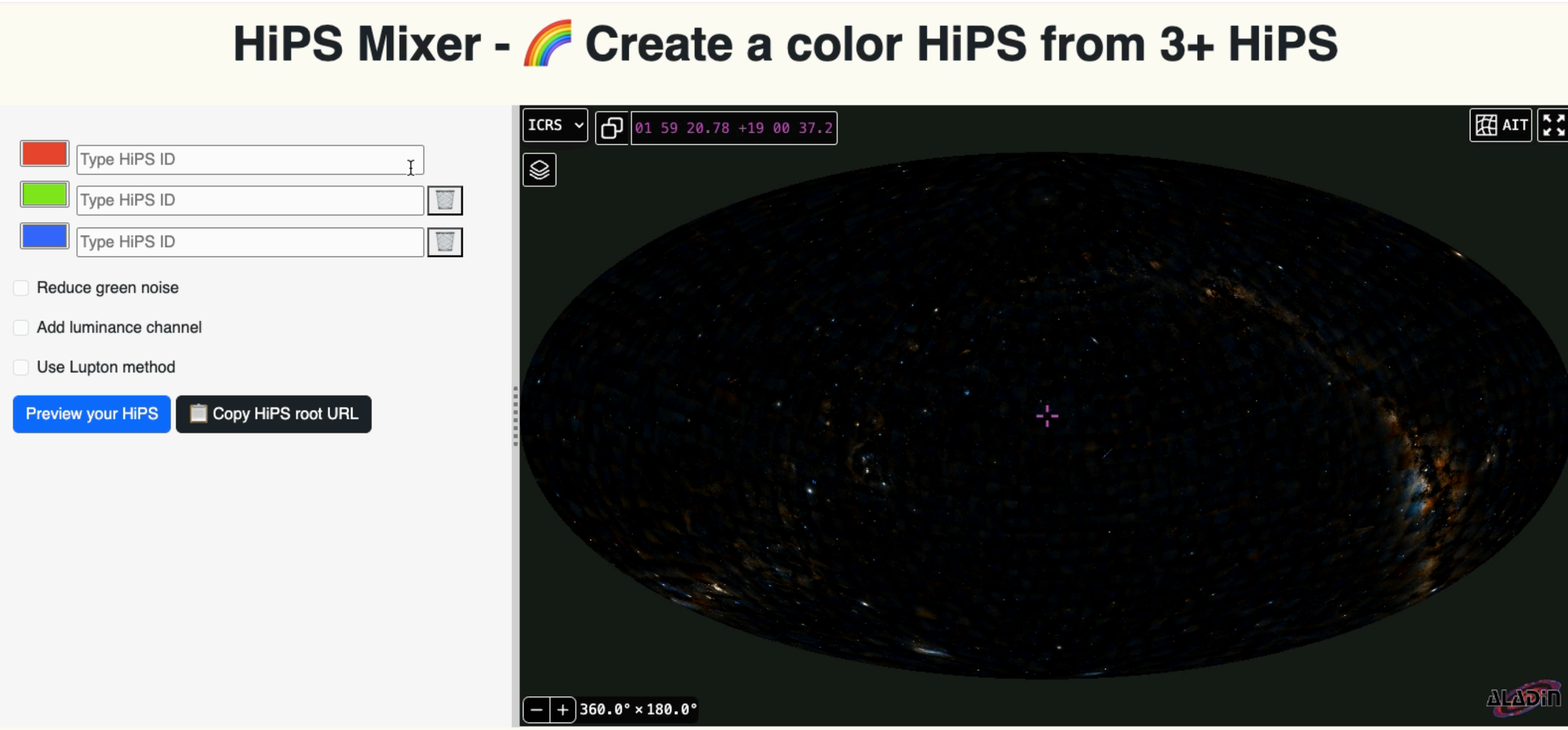
- Providing *good defaults* cut values
  - Pre-compute percentiles on all pixels
    for tiles at order 2
    (at most: 192 x 512 x 512 = 50M pixels)

- Default min_cut = percentile 0.5
- Default max_cut = percentile 99.9

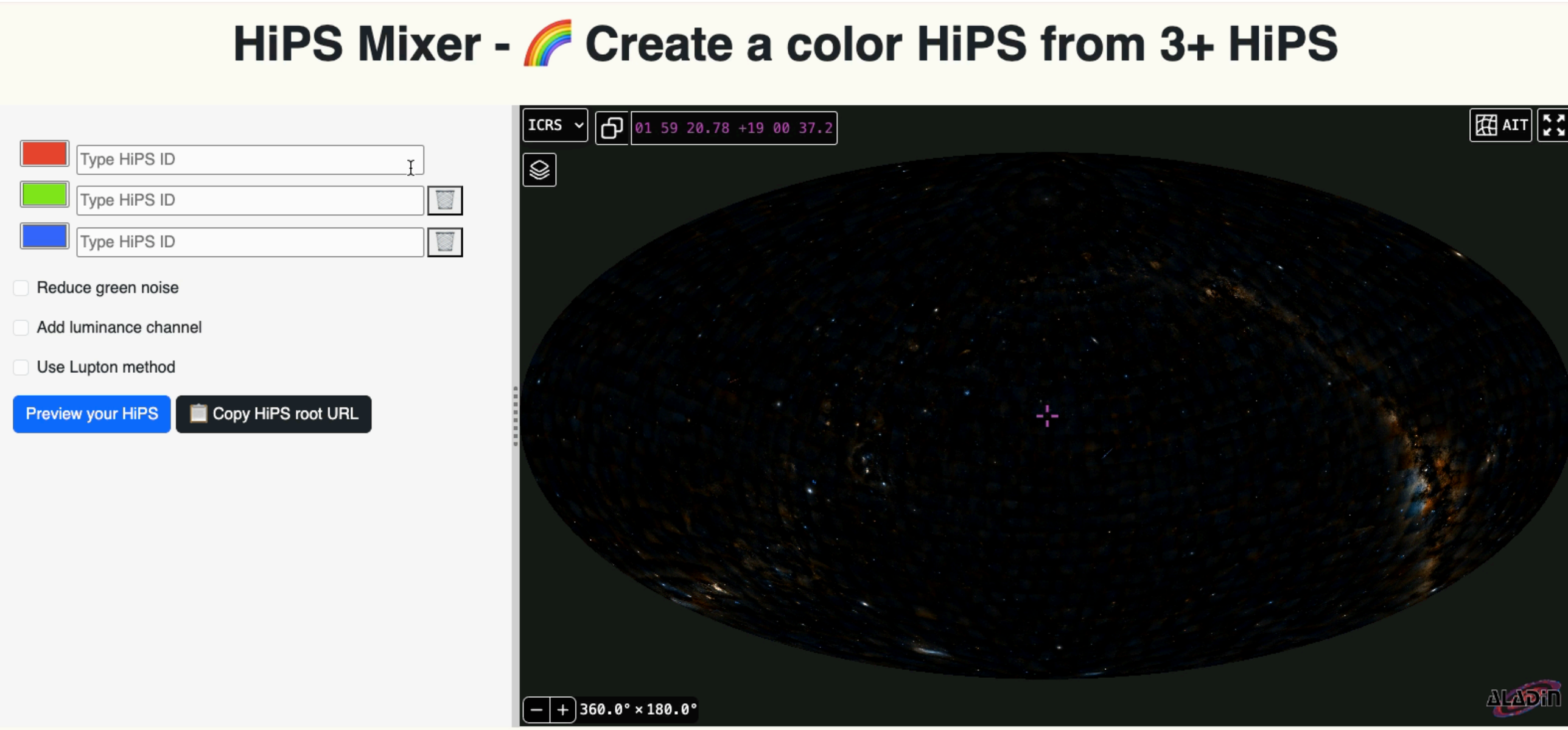
Other options

- Green noise reduction
- Lupton method vs classical RGB
- Luminance channel

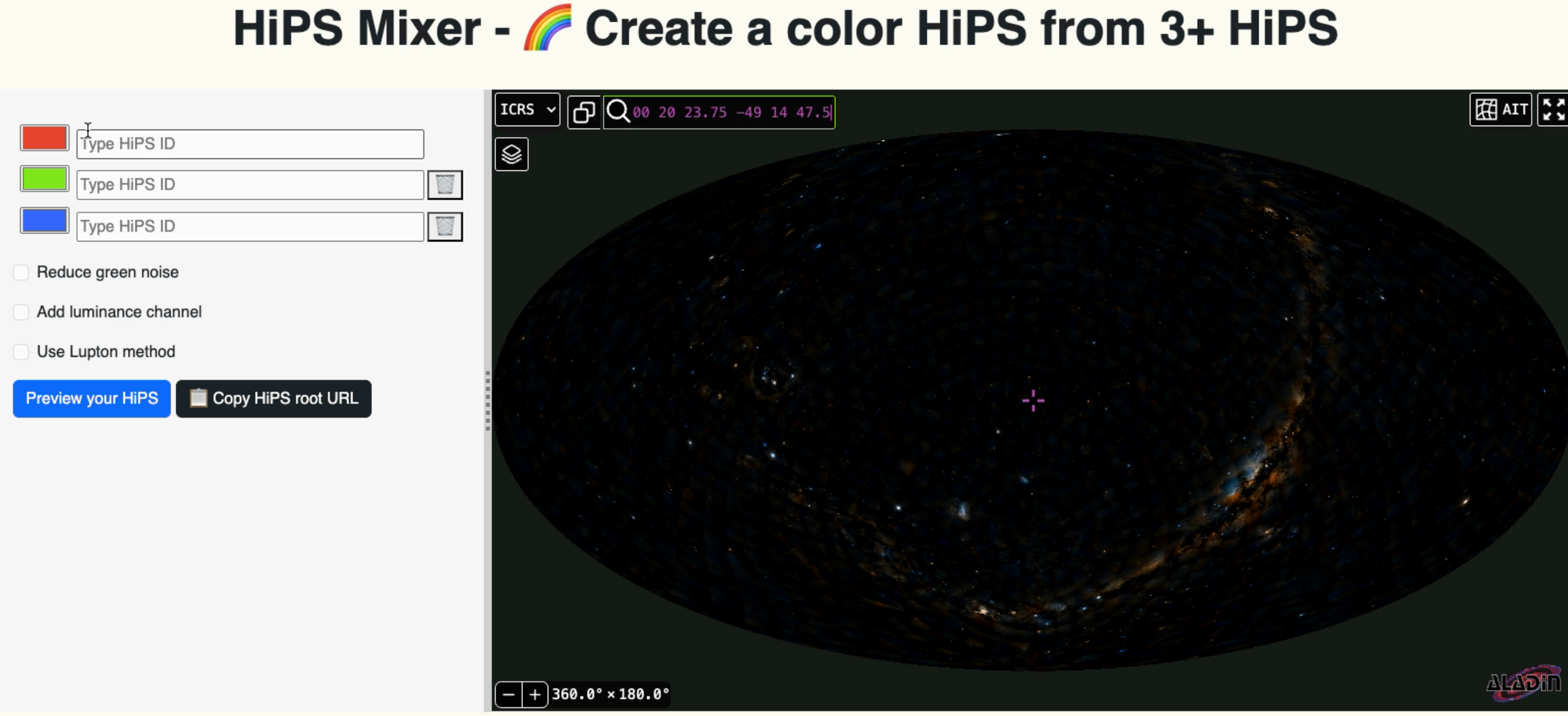
Video: Lupton method



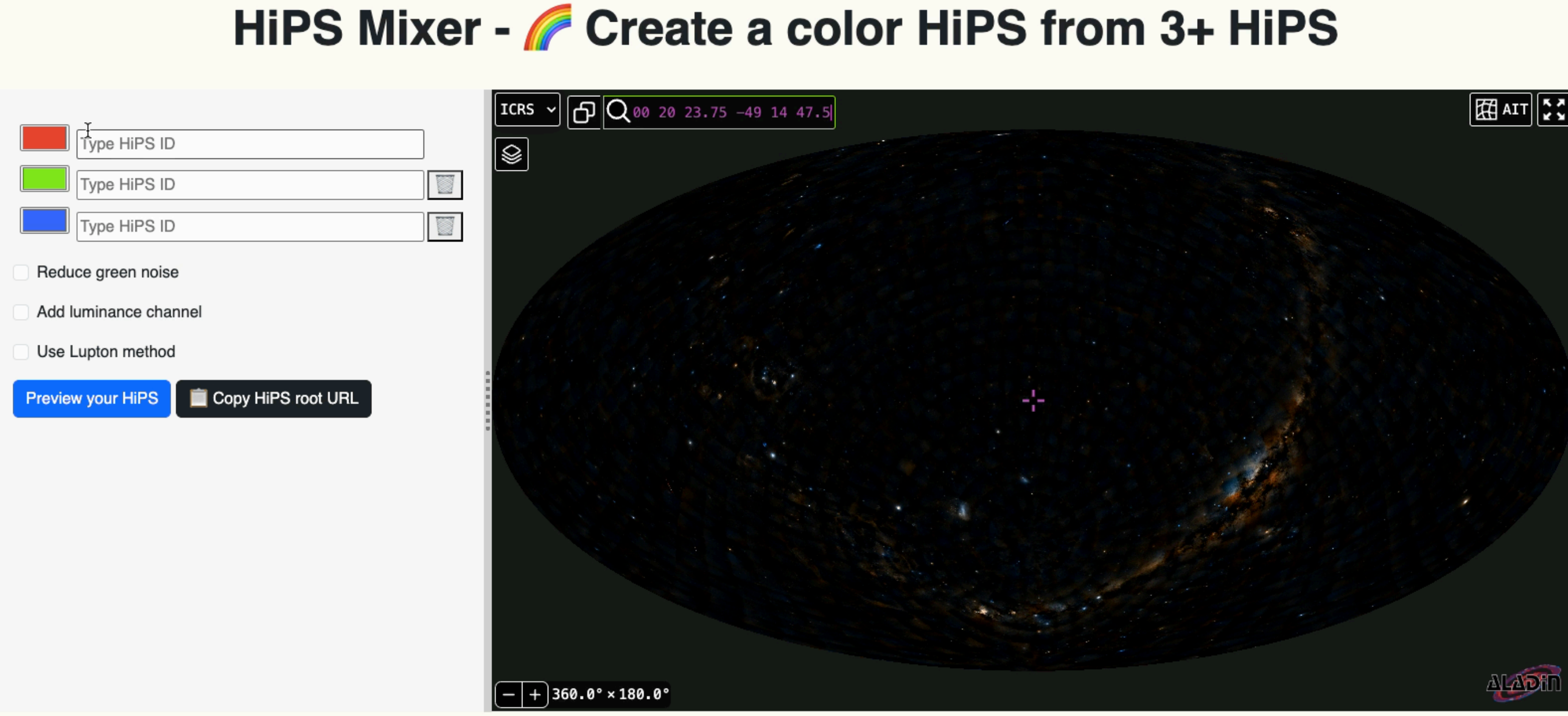
Video: Lupton method



Video: luminance channel



Video: luminance channel



Limitations

- Heterogeneous frames (equatorial vs galactic)
  - Not supported
  - Would need time-consuming reprojections

- Only HiPS hosted at CDS
  - Created by CDS or mirrored at CDS

Future developments

- **Cache** for MOC, Allsky, properties, tiles at order <=3
- Support **remote HiPS**
  - Huge distributed datasets, eg SKA
- Virtual HiPS from **arithmetic combination**
  - Eg: color HiPS from 2 flux maps HiPS
- **Stacking** HiPS
  - Hard, need to have compatible quantities
- HiPS cubes, **HiPS 3D** support
  - Combine cube slices

Implementation : **Falcon** Web framework

- Definition of *routes* associated to class methods
  - Parameter mapping
  - Example :

```
prioritize_hips_combiner = PrioritizeHiPSCombiner(...)
app.add_route('/prioritize/{params_str}/', prioritize_hips_combiner,
suffix='survey_hips_root')
app.add_route('/prioritize/{params_str}/properties',
prioritize_hips_combiner, suffix='survey_hips_properties')
app.add_route('/prioritize/{params_str}/Moc.fits',
prioritize_hips_combiner, suffix='survey_hips_moc')

app.add_route('/prioritize/{params_str}/Norder{order:int}/
Dir{dir_nb:int}/Npix{ipix:int}.{tile_format}',
prioritize_hips_combiner, suffix='survey_hips_tile')
```

```
    def on_get_survey_hips_tile(self, req, resp, params_str, order, dir_nb, ipix
, tile_format):
        params = decode_params_string(params_str)
        hipses = params['hipses']
        tile = create_tile(order, ipix, tile_format, hipses, self.hips_params, s
elf.mapping_functions)
        buffer = io.BytesIO()
        mimg.imsave(buffer, tile, format='png', dpi=42)

        resp.content_type = f'image/png'
        resp.data = buffer.getvalue()
```

Conclusion

- HiPS mixer:
  smart proxy, building virtual HiPS in various ways
  - Prototype at https://alasky.cds.unistra.fr/hips-mixer/

- Valid HiPS
  - use in any client
- Good performances
  - 30-50 tiles / second
- Advice to HiPS producers
  - Keep your FITS tiles
  - Equatorial frame if possible
  - Use PNG for partial coverage / pointed survey

Additional slides

Green noise reduction

- https://siril.readthedocs.io/en/stable/processing/colors.html#remove-green-noise

```
if reduce_green_noise:
    green_p = 0.5 * red + 0.5 * blue
    # this is SIRIL Average Neutral Protection
    mask = green > green_p
    green[mask] = green_p[mask]
```

- https://alasky.cds.unistra.fr/hips-combiner/color/hipses:CDS=P=2MASS=K,CDS=P=2MASS=H,CDS=P=2MASS=J;colors:red,green,blue;stretch:sqrt/?fov=10

Lupton

$$I = \frac{r + g + b}{3}$$

$$f(I) = \frac{\sinh^{-1}(Q \cdot I)}{\sqrt{Q}}$$

$$R = \frac{F(I) \cdot r}{I + \varepsilon}$$

$$G = \frac{F(I) \cdot g}{I + \varepsilon}$$

$$B = \frac{F(I) \cdot b}{I + \varepsilon}$$

https://alasky.cds.unistra.fr/hips-combiner/color/hipses:CDS=P=DESI-Legacy-Surveys=DR10=i,CD
Legacy-Surveys=DR10=r,CDS=P=DESI-Legacy-Surveys=DR10=g%3Bcolors:red,green,blue;stretch:log
fov=0.2&ra=318.776309&dec=-47.2208105
https://alasky.cds.unistra.fr/hips-combiner/color/hipses:CDS=P=DESI-Legacy-Surveys=DR10=i,CD
Legacy-Surveys=DR10=r,CDS=P=DESI-Legacy-Surveys=DR10=g%3Bcolors:red,green,blue;method:lup
fov=0.2&ra=318.776309&dec=-47.2208105