lemaitre_bandpasses

January 11, 2024

1 Using the MegaCam, HSC & ZTF bandpasses

For the moment, this works only with the fork of sncosmo available from here:

`https://github.com/nregnault/sncosmo.git`

and the passbands, that are defined in this module:

`https://gitlab.in2p3.fr/lemaitre/bandpasses.git`

Both are pip-installable.

For the moment, the bandpass data and filter models are stored in hdf5 files, attached to the lemaitre.bandpasses module. This allows to update them frequently, without having to interact with the sncosmo people. We will nevetheless make the pull requests so that they are integrated into sncosmo.

[10]:
```
import numpy as np
import sncosmo
```

Importing this module registers all the new bandpasses into sncosmo.

[ ]:
```
from lemaitre import bandpasses
```

1.1 Obtaining passbands from sncosmo

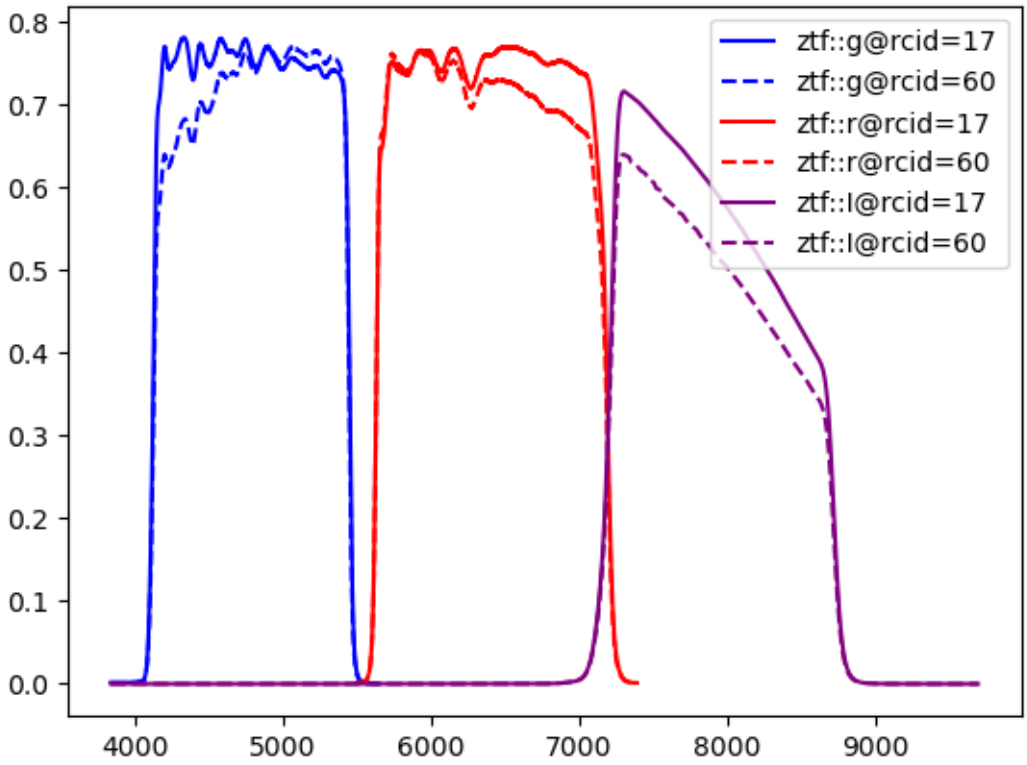
To obtain, for example, passband 'ztf::g' at location (x,y)=(1000,1000) on rcid number 12, we do:

[20]:
```
g = sncosmo.get_bandpass('ztf::g', x=1000, y=1000, sensor_id=12)
print(g.__class__)
print(g)
```

```
<class 'sncosmo.bandpasses.Bandpass'>
<Bandpass 'ztf::g' at 0x7f8dedb7b550>
```

[24]:
```
pl.figure()
colors = {'ztf::g': 'b', 'ztf::r': 'r', 'ztf::I': 'purple'}
for bn in ['ztf::g', 'ztf::r', 'ztf::I']:
    b = sncosmo.get_bandpass(bn, x=1000., y=1000., sensor_id=17)
    pl.plot(b.wave, b.trans, color=colors[bn], ls='-', label=f'{bn}@rcid=17')
    b = sncosmo.get_bandpass(bn, x=1000., y=1000., sensor_id=60)
```

```
    pl.plot(b.wave, b.trans, color=colors[bn], ls='--', label=f'{bn}@rcid=60')
_ = pl.legend(loc='upper right')
```



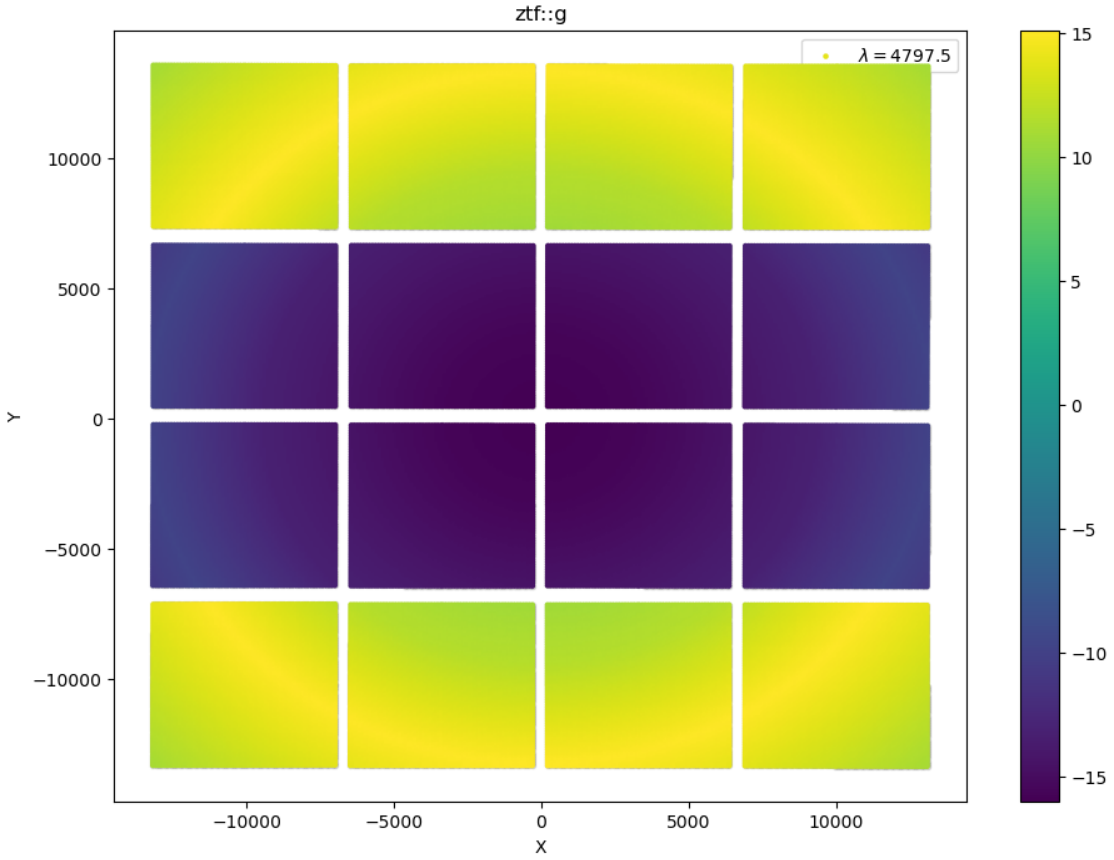
It is possible to obtain many instances of the same band, in a vectorized way

1.2 Passband non uniformities

Possible to obtain simulaneously a large number of passbands in a vectorized way.

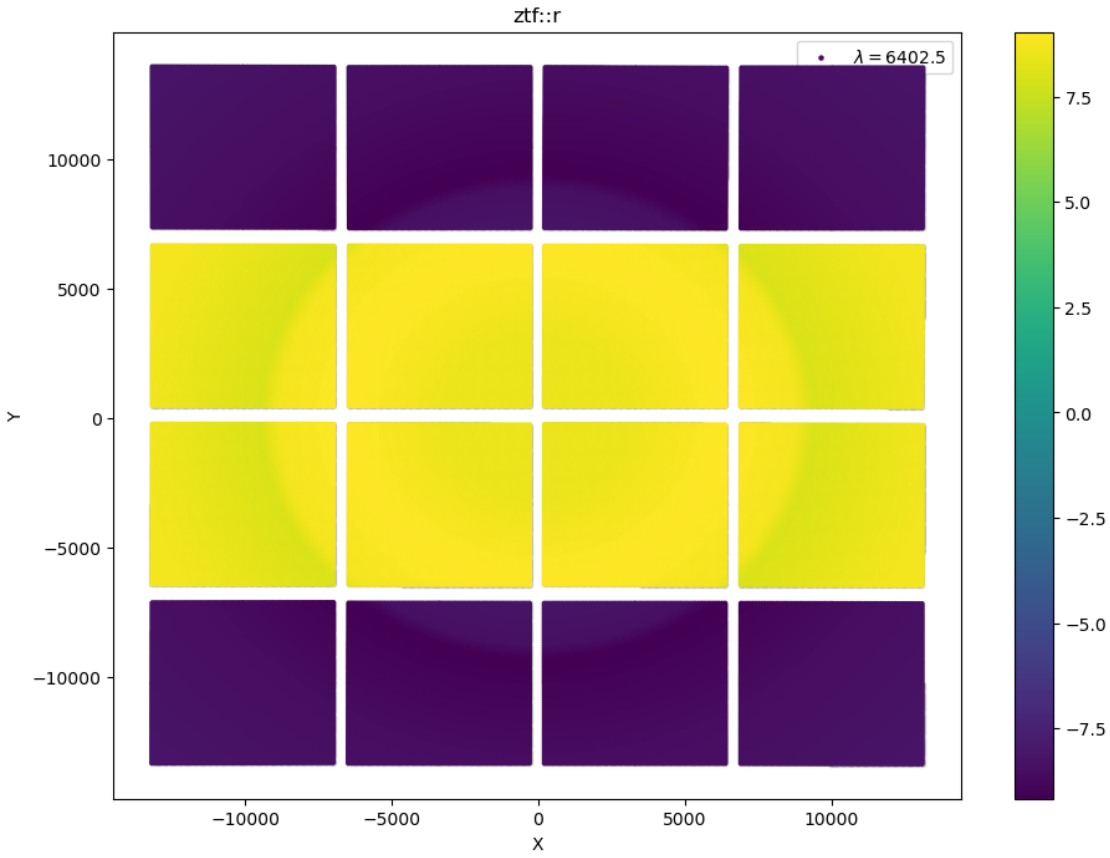
[28]:
```
from lemaitre.bandpasses import plots
x, y, ccd = plots.ztf_xyccd(delta=100)
plots.mean_wavelength_vs_position('ztf::g', x, y, ccd, title='ztf::g',␣
 ↪relative=True, markersize=5)
```

[28]:
```
array([13.87000517, 13.83333008, 13.79624205, …, 14.42212498,
       14.36804511, 14.31440419])
```



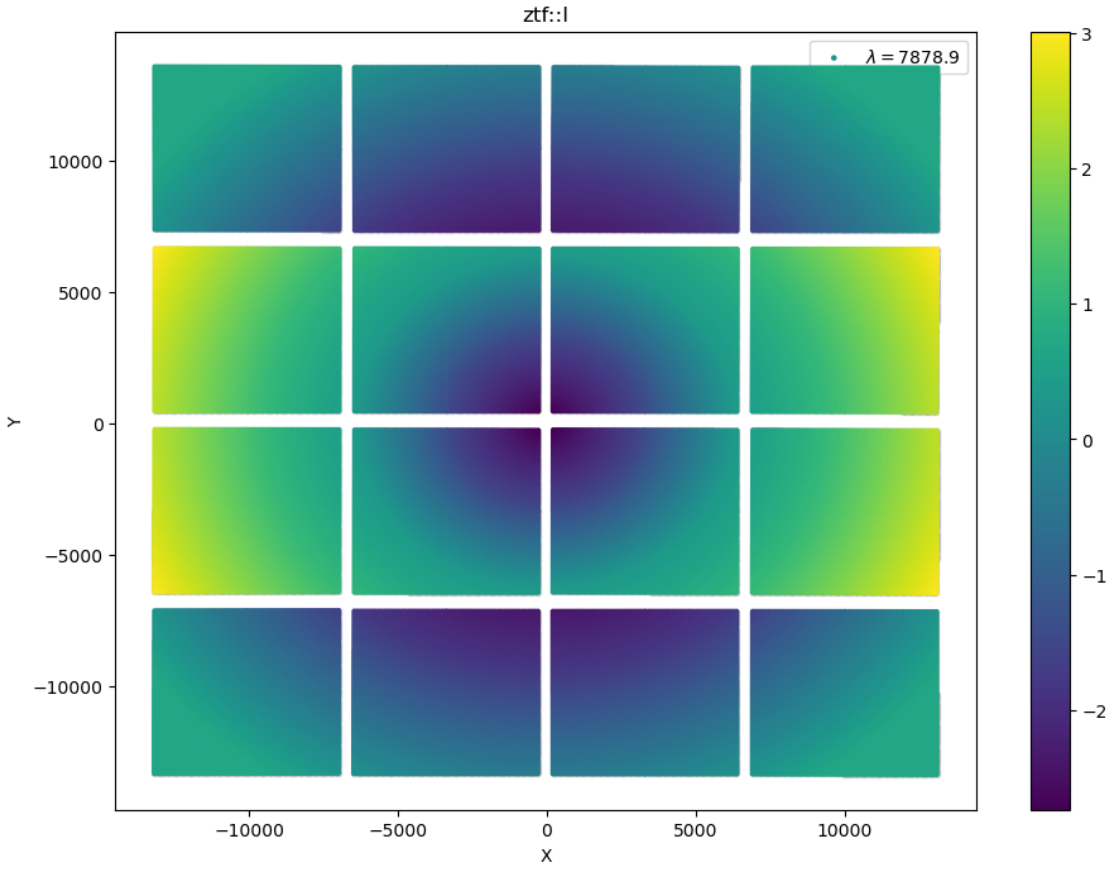
[29]:
```
x, y, ccd = plots.ztf_xyccd(delta=100)
plots.mean_wavelength_vs_position('ztf::r', x, y, ccd, title='ztf::r',␣
↪relative=True, markersize=5)
```

[29]:
```
array([-8.4373853 , -8.43399682, -8.4305713 , …, -8.67935295,
       -8.68902505, -8.69861555])
```



[30]:
```
x, y, ccd = plots.ztf_xyccd(delta=100)
plots.mean_wavelength_vs_position('ztf::I', x, y, ccd, title='ztf::I',␣
↪relative=True, markersize=5)
```

[30]:
```
array([ 0.2143785 ,  0.2321853 ,  0.25018373, …, -0.6889801 ,
       -0.71126972, -0.73336623])
```

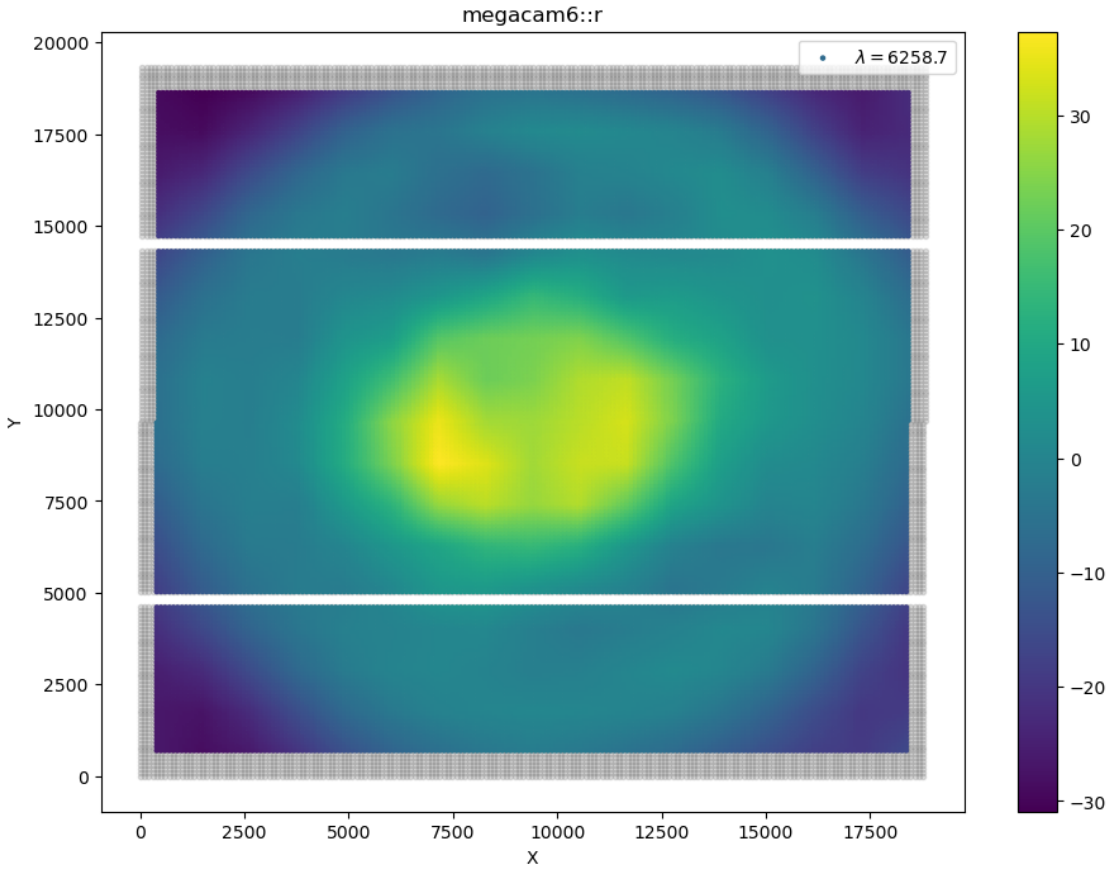


1.3 For comparison: MegaCam

[31]:
```
x, y, ccd = plots.megacam_xyccd(delta=100)
plots.mean_wavelength_vs_position('megacam6::r', x, y, ccd, title='megacam6::
  ↪r', relative=True, markersize=5)
```

```
/home/nrl/miniconda3/envs/lemaitre/lib/python3.11/site-
packages/lemaitre/bandpasses/plots.py:48: RuntimeWarning: invalid value
encountered in divide
  return (tr * wl).sum(axis=1) / tr.sum(axis=1)
```

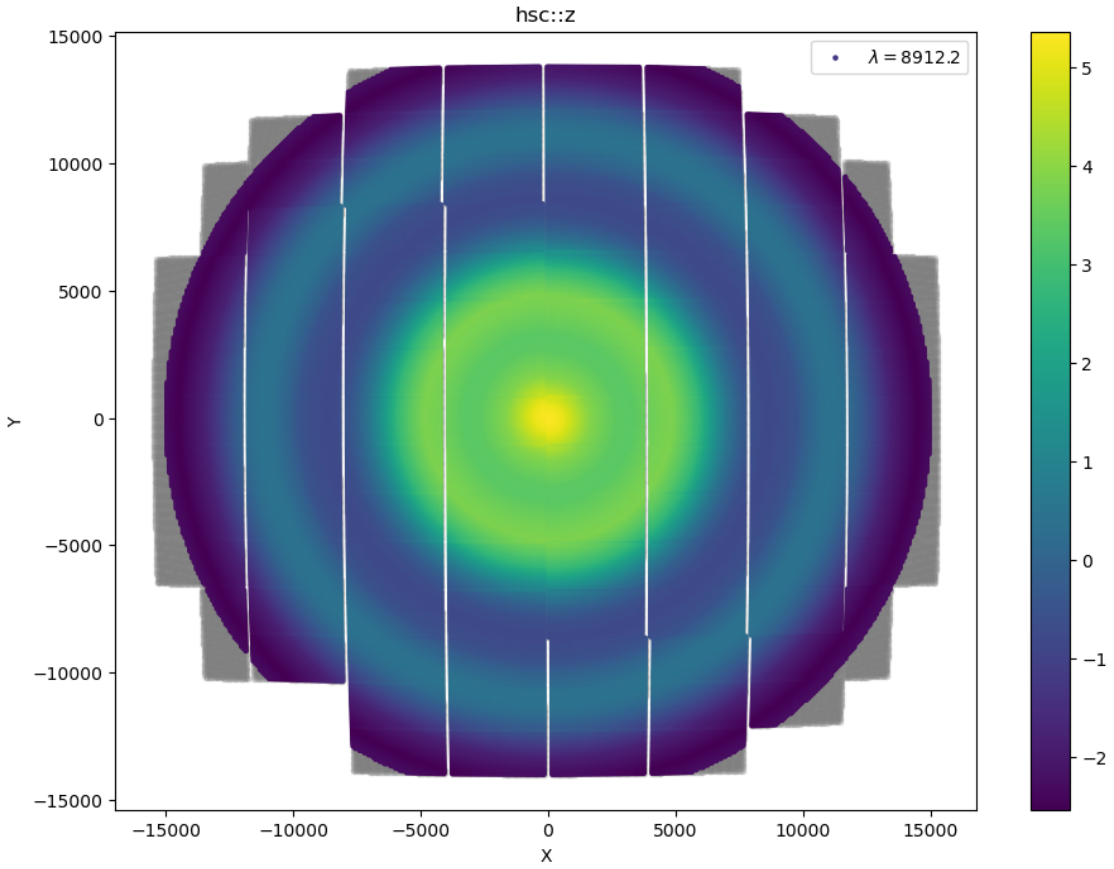
[31]:
```
array([        nan,         nan,         nan, …, -0.60991225,
       -6.6870578 ,         nan])
```



1.4 For comparison: HSC

[34]:
```
x, y, ccd = plots.hsc_xyccd(delta=100)
plots.mean_wavelength_vs_position('hsc::z', x, y, ccd, title='hsc::z',␣
↪relative=True, markersize=5)
```

[34]:
```
array([-1.22413629, -0.38290163, -1.12083179, …, -2.03425863,
       -1.84462574,         nan])
```


hsc::z
λ = 8912.2
15000
10000
5000
0
−5000
−10000
−15000
Y
−15000
−10000
−5000
0
5000
10000
15000
X
5
4
3
2
1
0
−1
−2

[ ]: