Why do my google tiles look poor in a Cartopy map?

Asked 1 year, 8 months ago Active 10 months ago Viewed 2k times

6

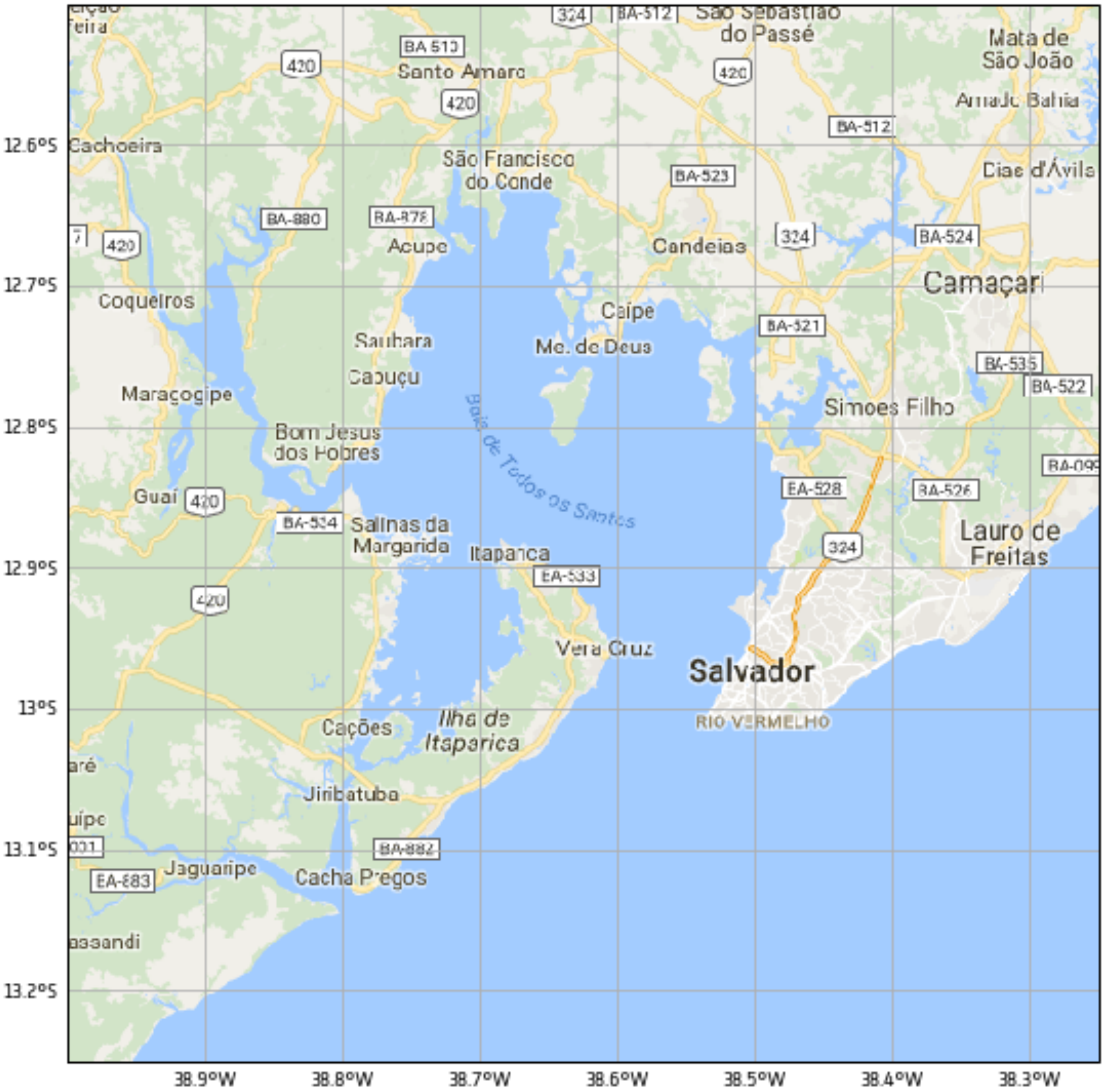
4

I am a bit puzzled by the rendering of google tiles with Cartopy. The map looks extremely poor compared to the standard google map look.

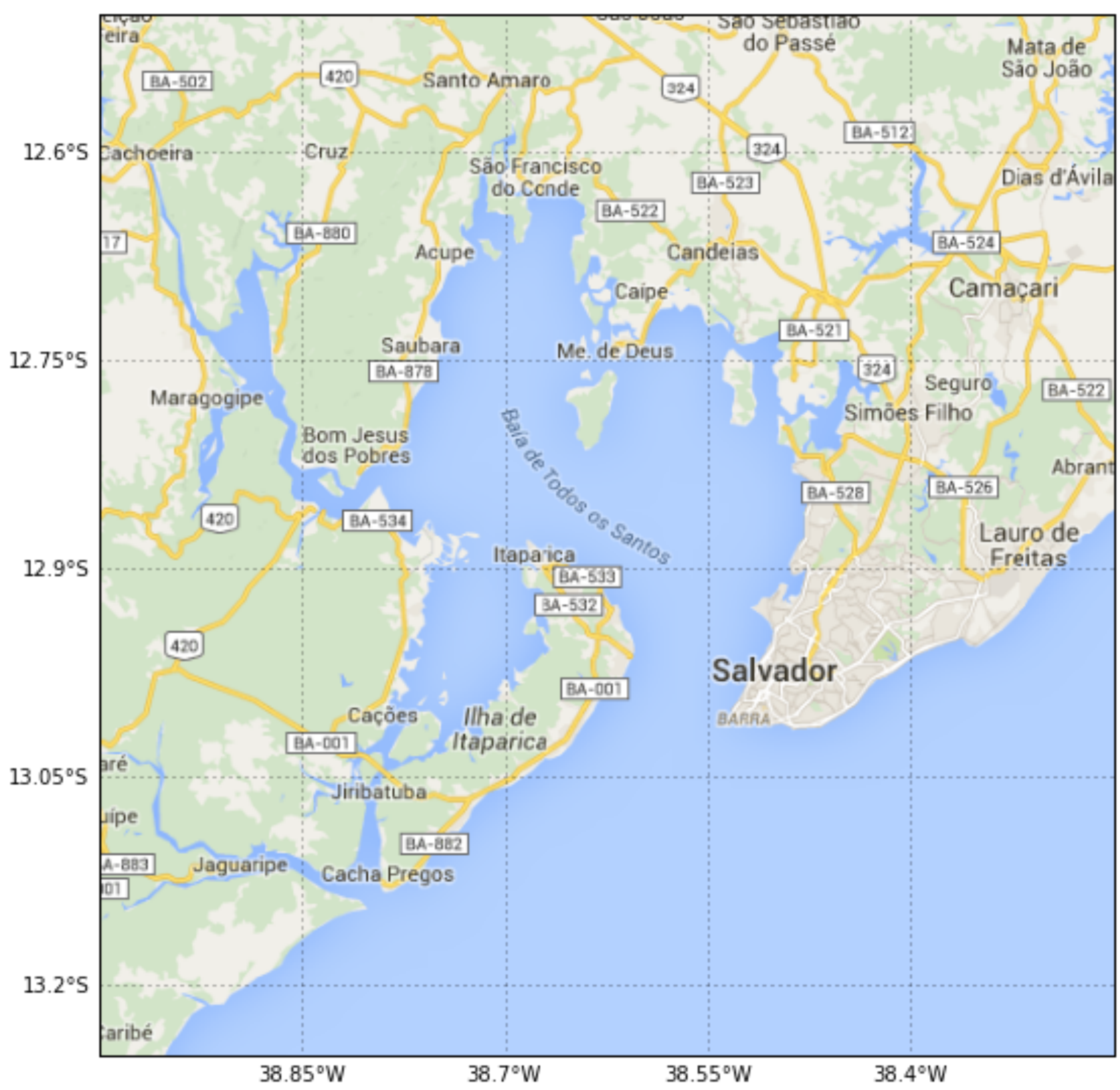
Example (code from https://ocefpaf.github.io/python4oceanographers/blog/2015/06/22/osm/):

```
import matplotlib.pyplot as plt

import cartopy.crs as ccrs
from cartopy.io import shapereader
from cartopy.mpl.gridliner import LONGITUDE_FORMATTER, LATITUDE_FORMATTER

def make_map(projection=ccrs.PlateCarree()):
    fig, ax = plt.subplots(figsize=(9, 13),
                           subplot_kw=dict(projection=projection))
    gl = ax.gridlines(draw_labels=True)
    gl.xlabels_top = gl.ylabels_right = False
    gl.xformatter = LONGITUDE_FORMATTER
    gl.yformatter = LATITUDE_FORMATTER
    return fig, ax
import cartopy.io.img_tiles as cimgt

extent = [-39, -38.25, -13.25, -12.5]

request = cimgt.GoogleTiles()

fig, ax = make_map(projection=request.crs)
ax.set_extent(extent)

ax.add_image(request, 10)
```

Generates:

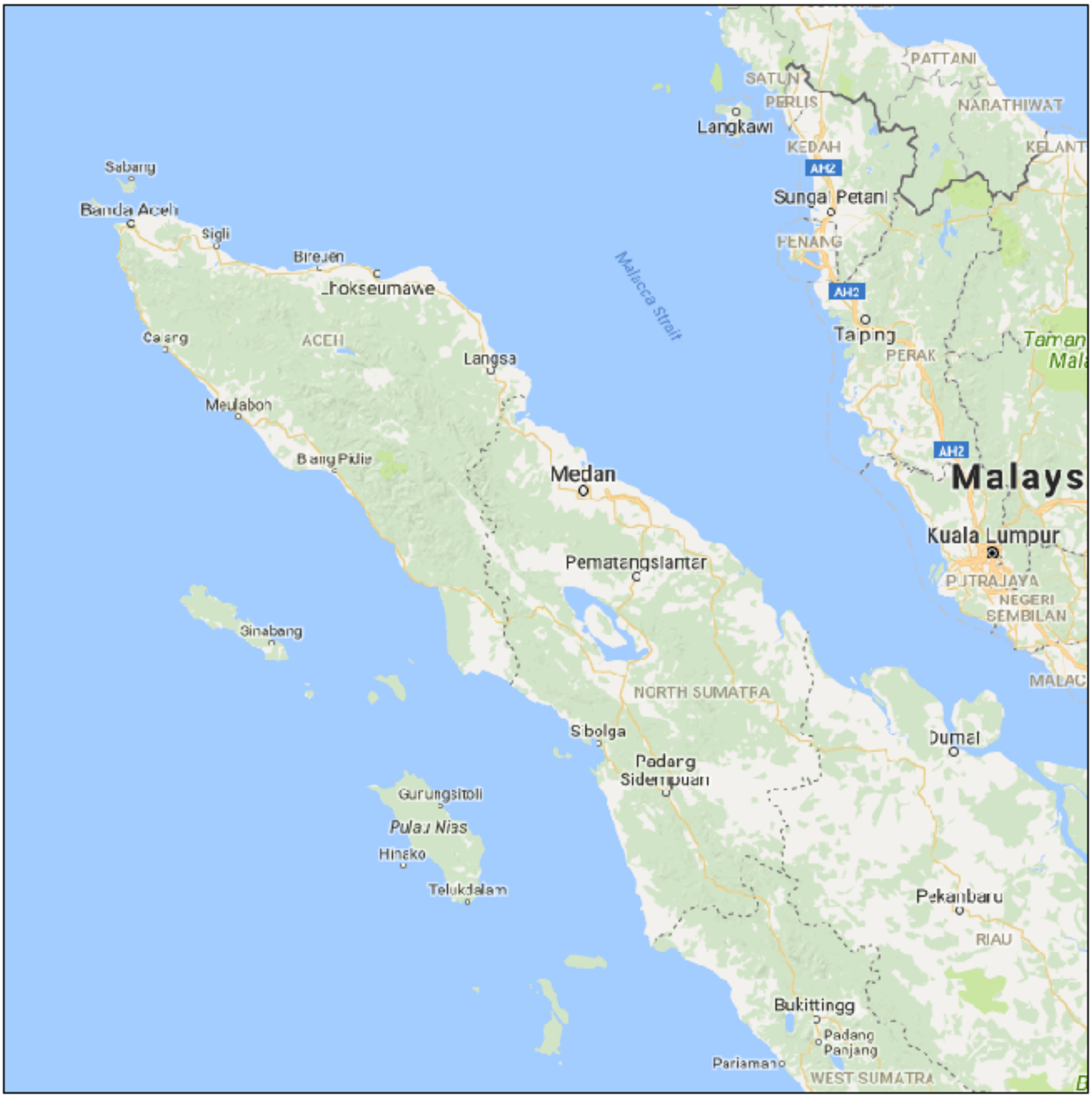


Which looks very poor—look at the pixelated rendering of text label and street number—compared to the same image shown on the linked website:

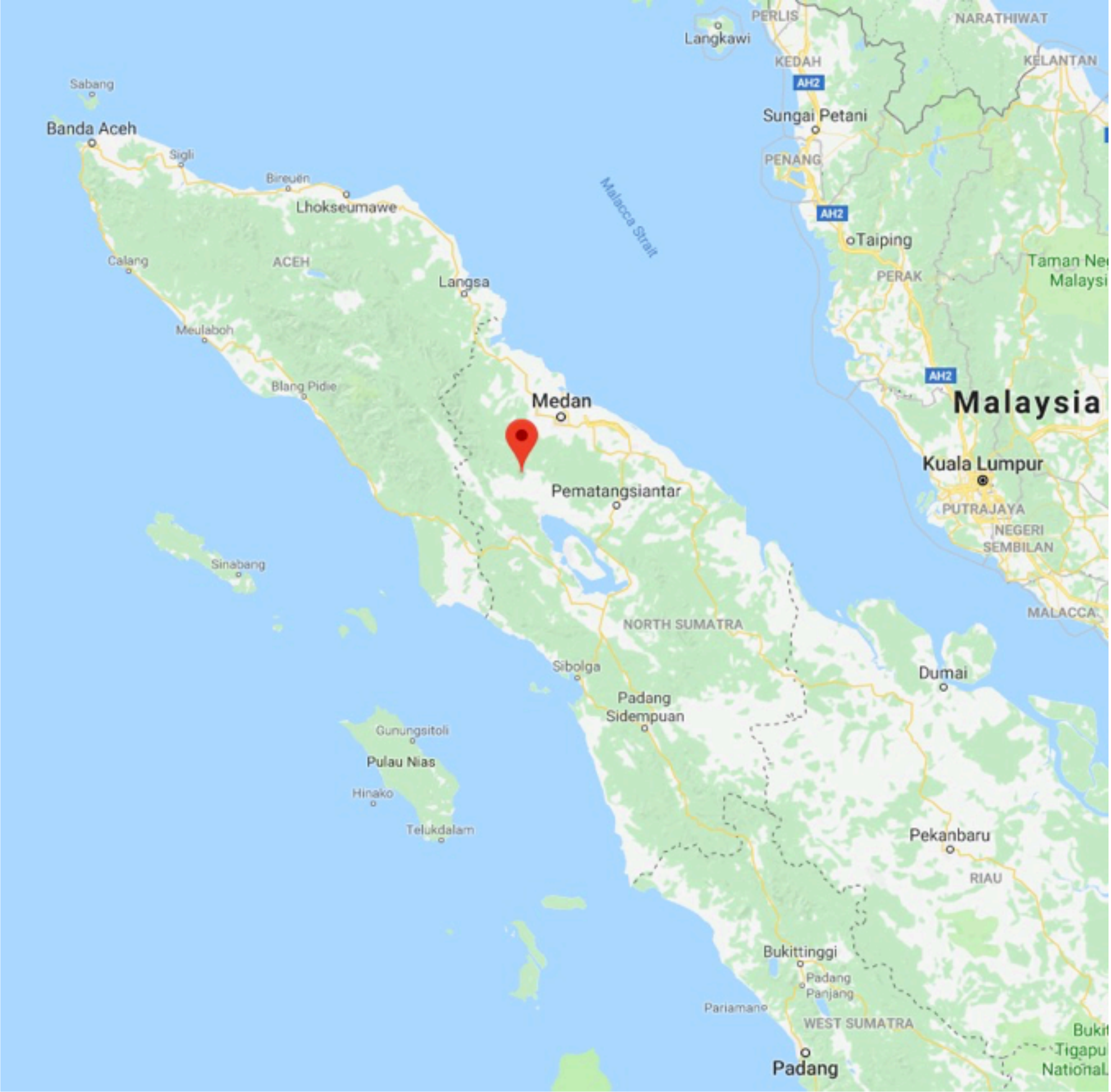


Changing zoom level does not seem to improve the situation.

This is another example on a map I was working on as rendered by Cartopy and googletiles():



Same map displayed in Google Maps



Does anybody know what could be the cause of this strange issue and how to address it?

python google-maps matplotlib cartopy

edited Mar 9 '18 at 17:19

asked Mar 7 '18 at 15:06

stm4tt
565 ● 3 ● 17

Could it be the mpl backend? What does `matplotlib.get_backend()` return? – Daniel Kirkham Mar 8 '18 at 11:44

@DanielKirkham I get 'module://ipykernel.pylab.backend_inline' on my Jupyter notebook and 'MacOSX' on my ipython console. Both give the same results. – stm4tt Mar 8 '18 at 11:47

1 I have no idea about cartopy's behavior but I recently came across salem which can do some cool stuff with google images salem.readthedocs.io/en/latest/auto_examples/... – Ray Bell Mar 24 '18 at 4:46

2 Answers

6

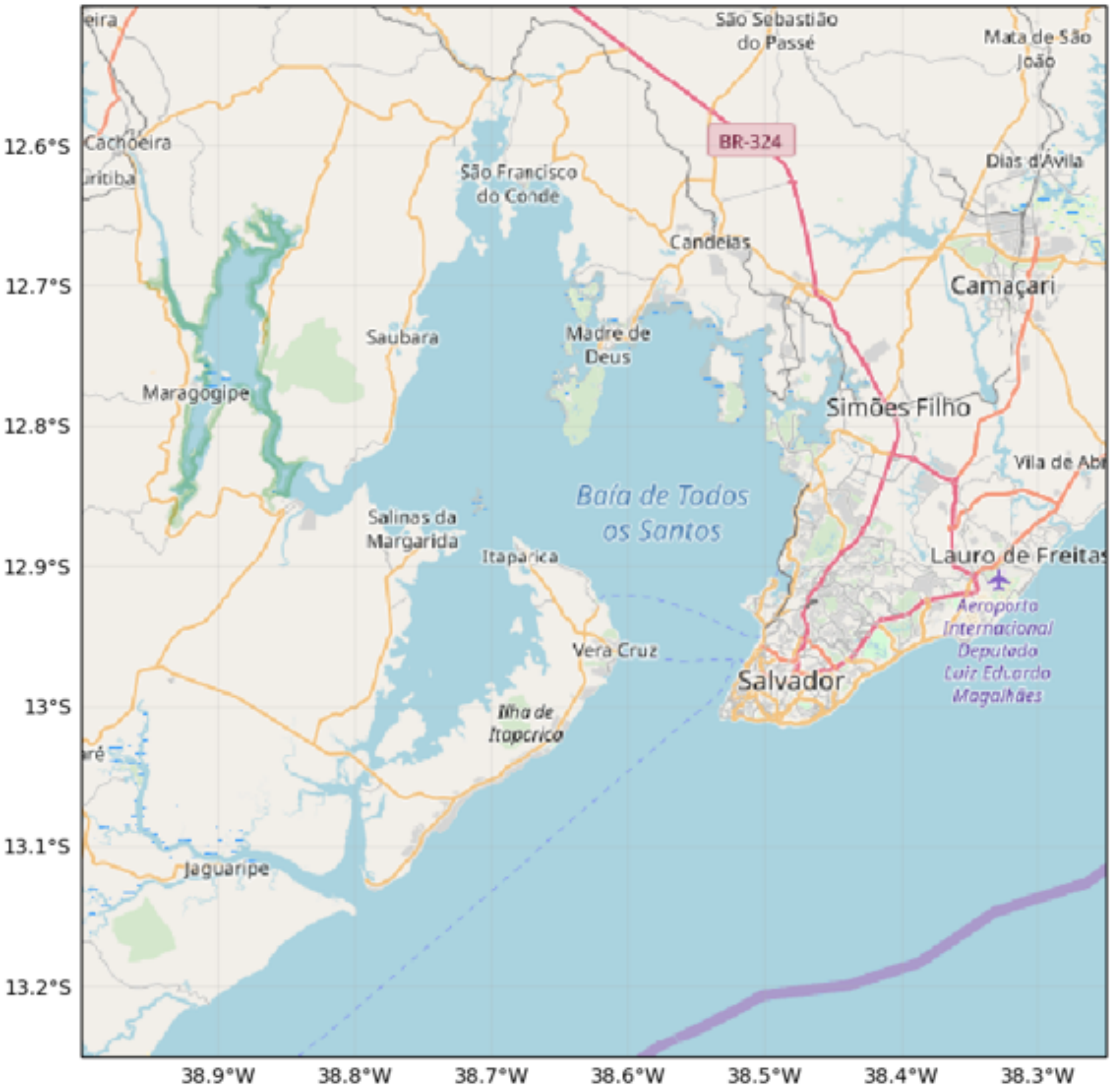
This question was also asked on the cartopy issue tracker at https://github.com/SciTools/cartopy/issues/1048, where it was suggested setting the `interpolation=` keyword argument. This is the standard matplotlib interpolation for imshow, which is documented at https://matplotlib.org/gallery/images_contours_and_fields/interpolation_methods.html.

We determined in the issue tracker that an interpolation of `nearest` is what you are seeing here. Changing that to `bilinear` gives a good result, and an even better result is achievable with different interpolation schemes. For example the `spline36` scheme results in a very pleasant image...

So, with your example code of:

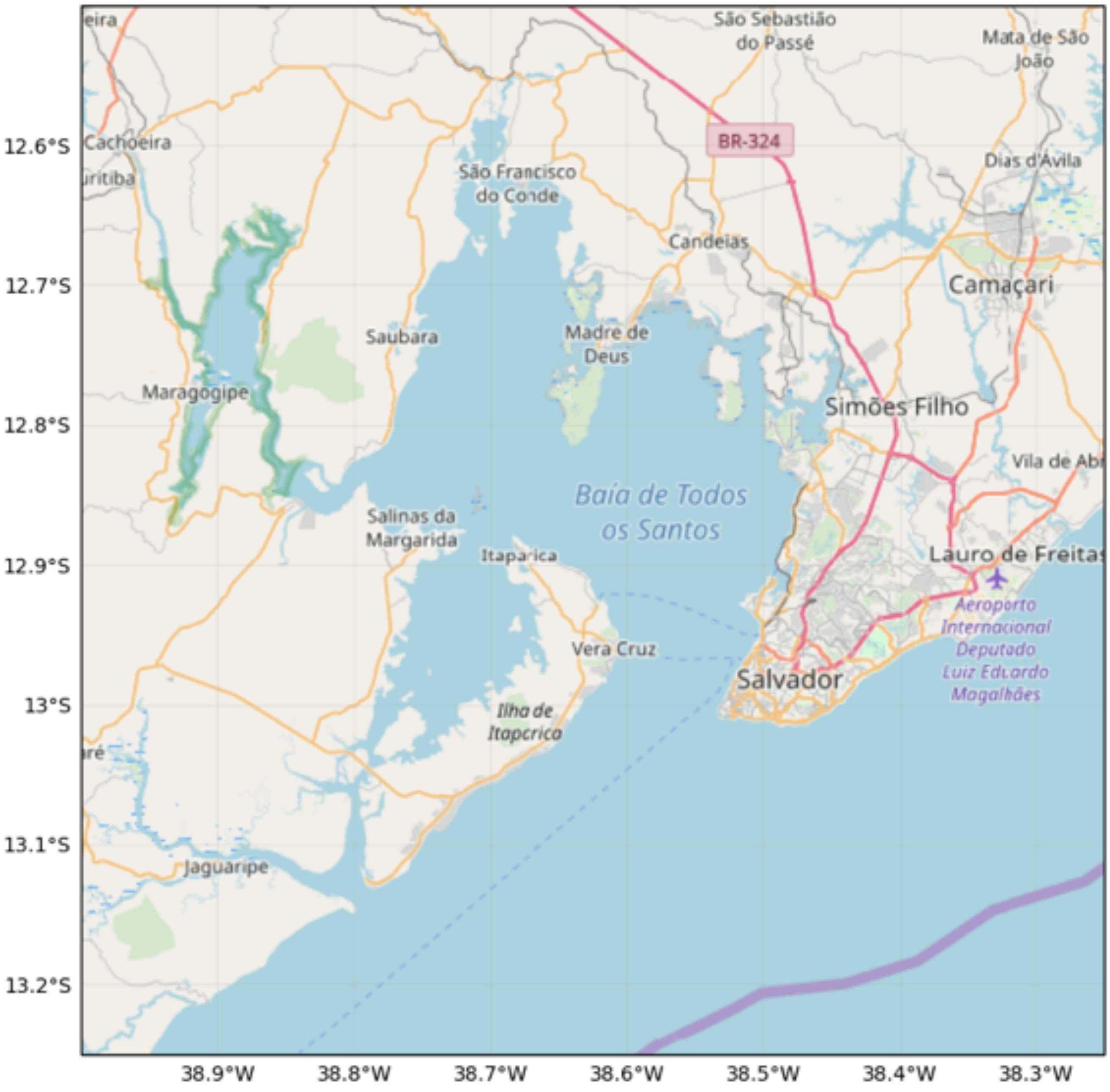
```
import matplotlib.pyplot as plt

import cartopy.crs as ccrs
from cartopy.io import shapereader
from cartopy.mpl.gridliner import LONGITUDE_FORMATTER, LATITUDE_FORMATTER

import cartopy.io.img_tiles as cimgt

extent = [-39, -38.25, -13.25, -12.5]

request = cimgt.OSM()

fig = plt.figure(figsize=(9, 13))
ax = plt.axes(projection=request.crs)
gl = ax.gridlines(draw_labels=True, alpha=0.2)
gl.xlabels_top = gl.ylabels_right = False
gl.xformatter = LONGITUDE_FORMATTER
gl.yformatter = LATITUDE_FORMATTER

ax.set_extent(extent)

ax.add_image(request, 10)

plt.show()
```

We get:



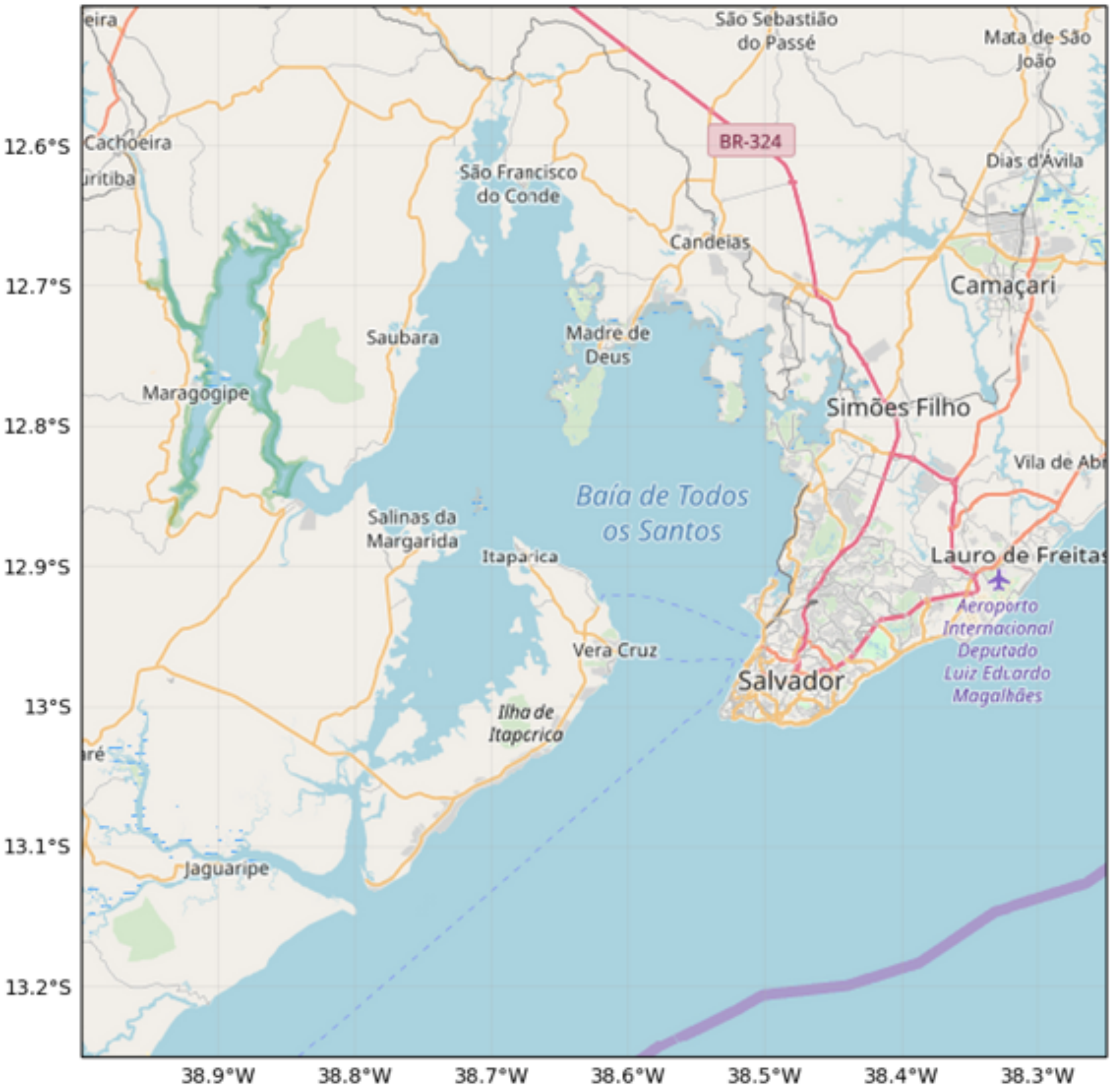
To set `bilinear` interpolation, we can change the `add_image` line to:

```
ax.add_image(request, 10, interpolation='bilinear')
```

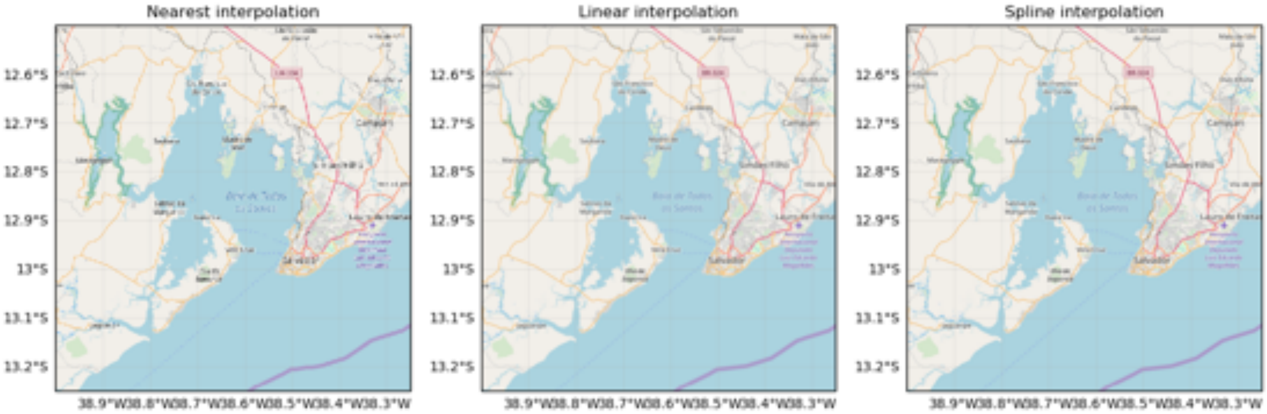


Even better, let's try something like spline36 with:

```
ax.add_image(request, 10, interpolation='spine36')
```



Putting these images side-by-side:



There is a caveat (as pointed out in https://github.com/SciTools/cartopy/issues/1048#issuecomment-417001744) for the case when the tiles are being plotted on their non-native projection. In that situation we have two variables to configure:

1) The resolution of the regridding from native projection to target projection 2) The interpolation scheme of the rendering of the reprojected image (this is what we have been changing in this answer).

Hope this is all useful information.

answered Aug 29 '18 at 16:06

pelson

16k ● 3 ● 69 ● 83

1

There is a small typo in the accepted answer.

```
ax.add_image(request, 10, interpolation='spine36')
```

should be

```
ax.add_image(request, 10, interpolation='spline36')
```

answered Jan 8 at 10:09

QuarKUS7

73 ● 5