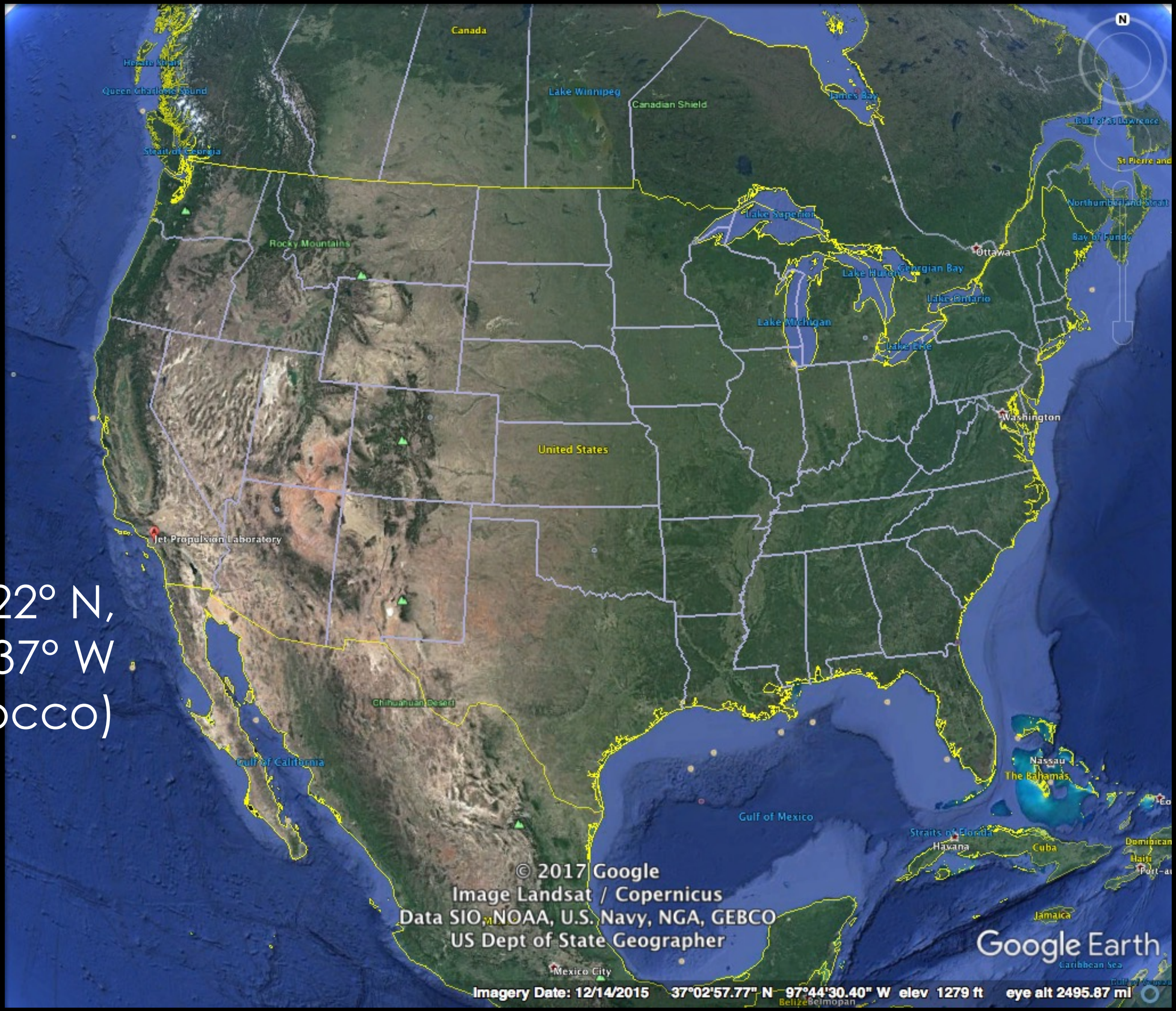
too kink to be square? mapping satellite polygon footprints onto regular grids by tessellation

karlsruhe institute of technology
2017-07-11

thomas p kurosu
jet propulsion laboratory, california institute of technology

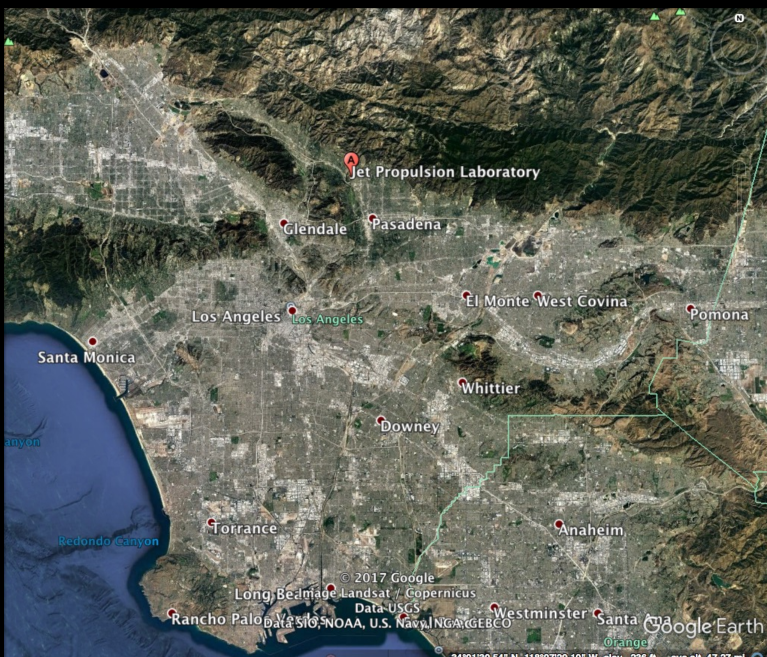
Jet Propulsion Laboratory
California Institute of Technology

prelude – a bit about jpl



34.0522° N,
118.2437° W
(rabat, morocco)

prelude – a bit about jpl



situated in pasadena, california, 20 minutes north of los angeles

prelude – a bit about jpl

in the foothill of the san gabriel mountains

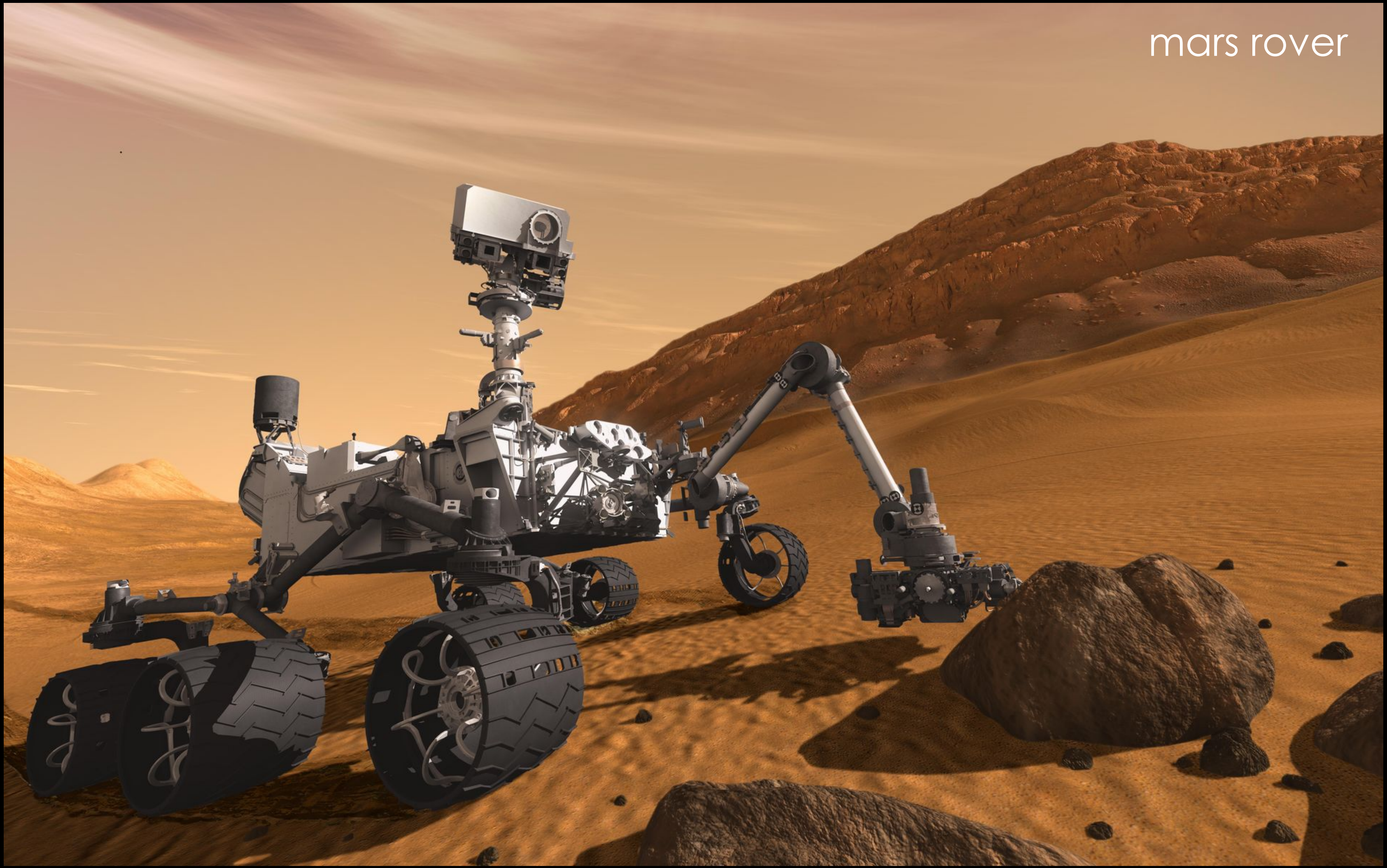
~7000 employees

planetary exploration
astronomy/astrophysics
earth sciences

prelude – a bit about jpl

space shuttle development

prelude – a bit about jpl



jpl atmospheric missions

- Quick Scatterometer (QuickScat; 1999)
- Advanced Spaceborn Thermal Emission and Reflection Radiometer (ASTER; Terra 1999)
- Multi-angle Imaging SpectroRadiometer (MISR; Terra 1999)
- Jason 1/2/3 (2001/2008/2016)
- Gravity Recovery and Climate Experiment (GRACE; 2002)
- Atmospheric Infrared Sounder (AIRS; Aqua 2002)
- SeaWinds (Midori 2 2002)
- Tropospheric Emission Spectrometer (TES; Aura 2004)
- Microwave Limb Sounder (MLS; Aura 2004)
- CloudSat (2005)
- Orbiting Carbon Observatory 2 (OCO-2; 2014)
- ECOsystem Spaceborne Thermal Radiometer Experiment on Space Station (ECOSTRESS; ISS 2019)
- Orbiting Carbon Observatory-3 (OCO-3; ISS 20??)

too kink to be square?
mapping satellite polygon footprints onto regular grids by tessellation

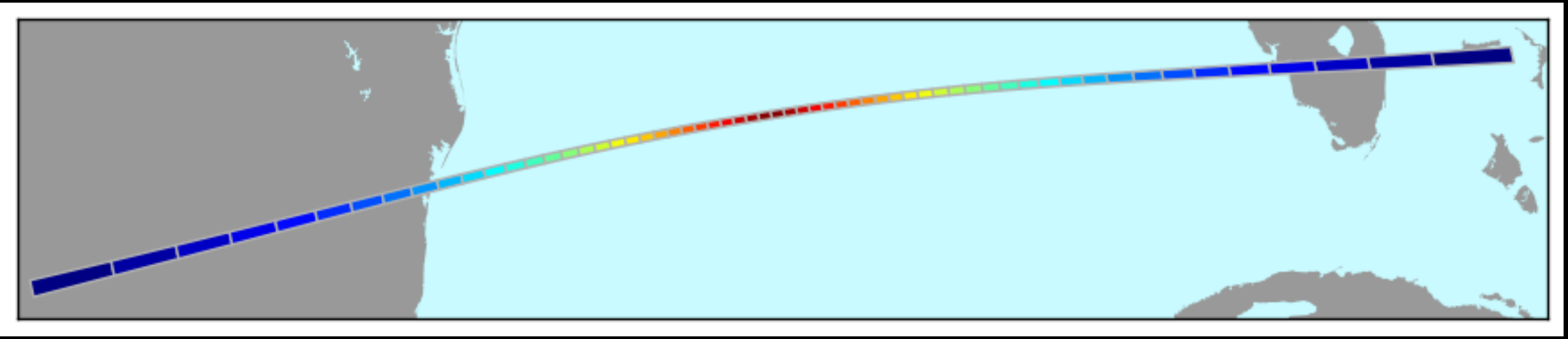
note
this talk is not so much about science,
but rather about a tool to enable scientific investigations

also: the principle is really really simple!

motivation #1

most applications that use satellite data require mapping of the (usually irregular) footprint polygons to rectangular grids, for subsequent averaging or comparison to other data

the commonly used “drop-in-the-box” approach either leads to significant loss of spatial coverage when the grid cells approach the size of the satellite footprint area, or jettisons spatial resolution of the sensor.



example of an OMI swath line

2,600 km swath divided into 60 cross-track pixels

range of pixel size: 340 km^2 (center) to 4,600 km^2 (edges)

motivation #2

comparison/mapping of two independent data sets with significantly different spatial resolution on the same spatial grid

e.g.,
satellite *vs.* satellite
satellite *vs.* aircraft *vs.* ground

motivation #3

long-term averaging with high spatial oversampling
may expose features on sub-footprint scale

this depends on the nature of the observations – mainly the variation in the observation target and how the satellite footprints sample them

the goal

develop an algorithm to map the satellite footprint polygon to a rectangular grid of arbitrary latitude/longitude resolution, without loss of information on spatial coverage

method: tessellation

tessellation

(imagine an m.c. escher image here)

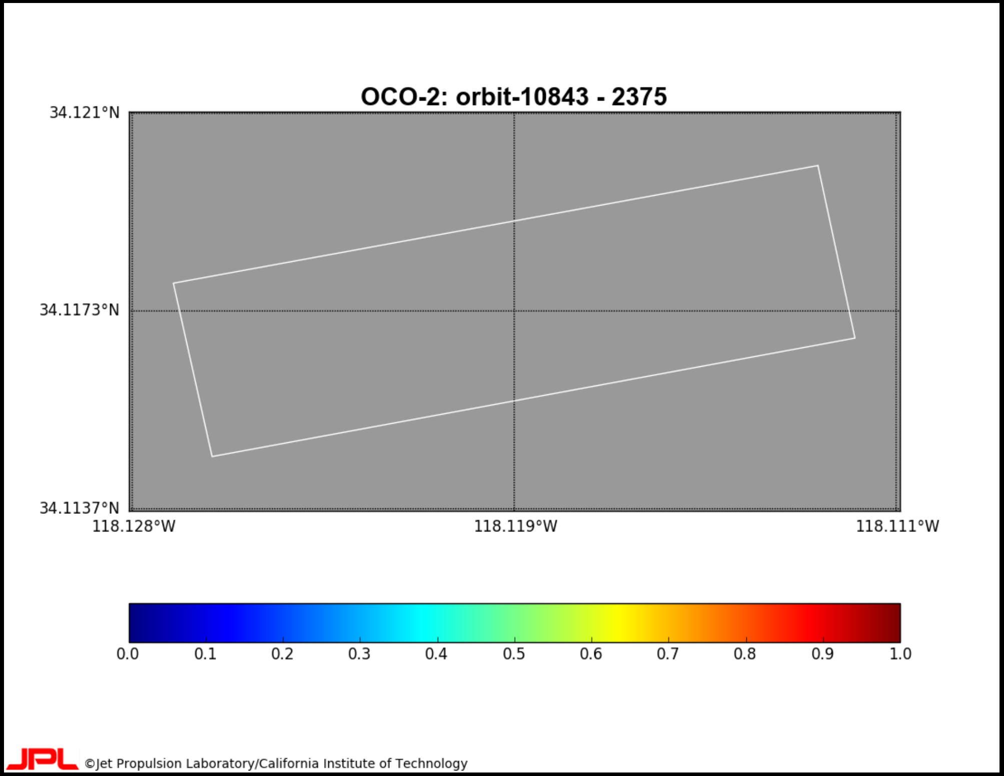
tessellation of a flat surface is the tiling of a plane using one or more geometric shapes, called tiles, with no overlaps and no gaps.
(https://en.wikipedia.org/wiki/Tessellation)

the process – overview

1. read in satellite data, including corner-coordinates of the satellite footprints
2. define a rectangular grid onto which to map the data
3. for each satellite footprint:
    1. determine which grid cells it covers
    2. compute fractional coverage of grid cells by the footprint
4. combine/average output with weights of your choice

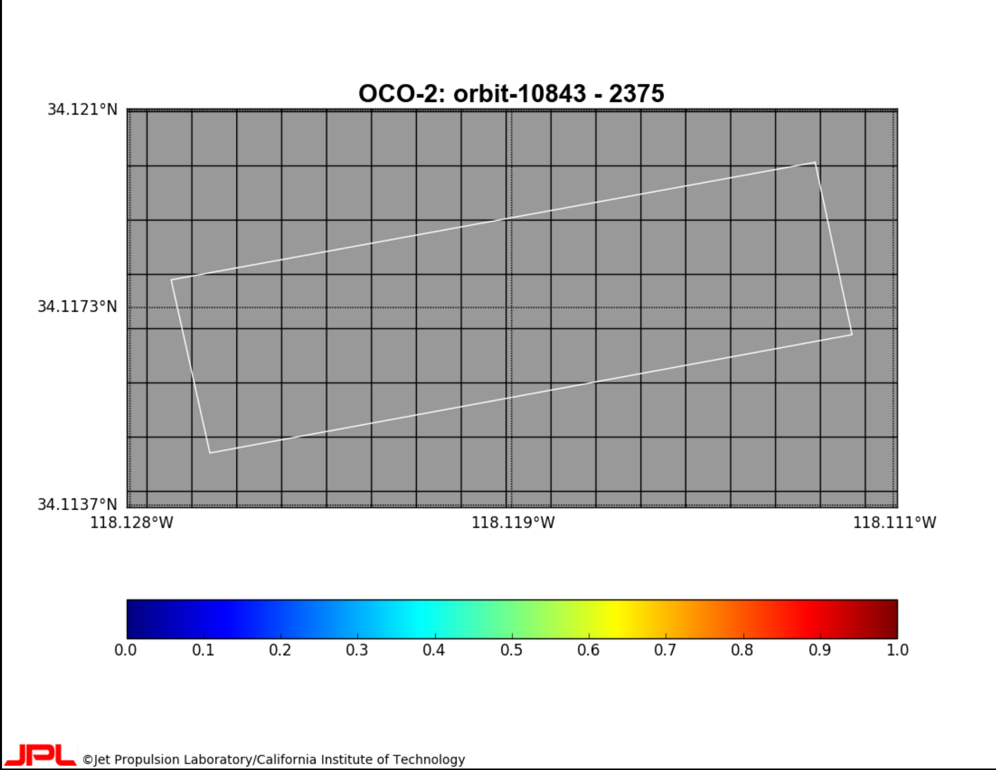
the tessellation process

1. start with a satellite ground pixel (OCO-2 target data, in this case)



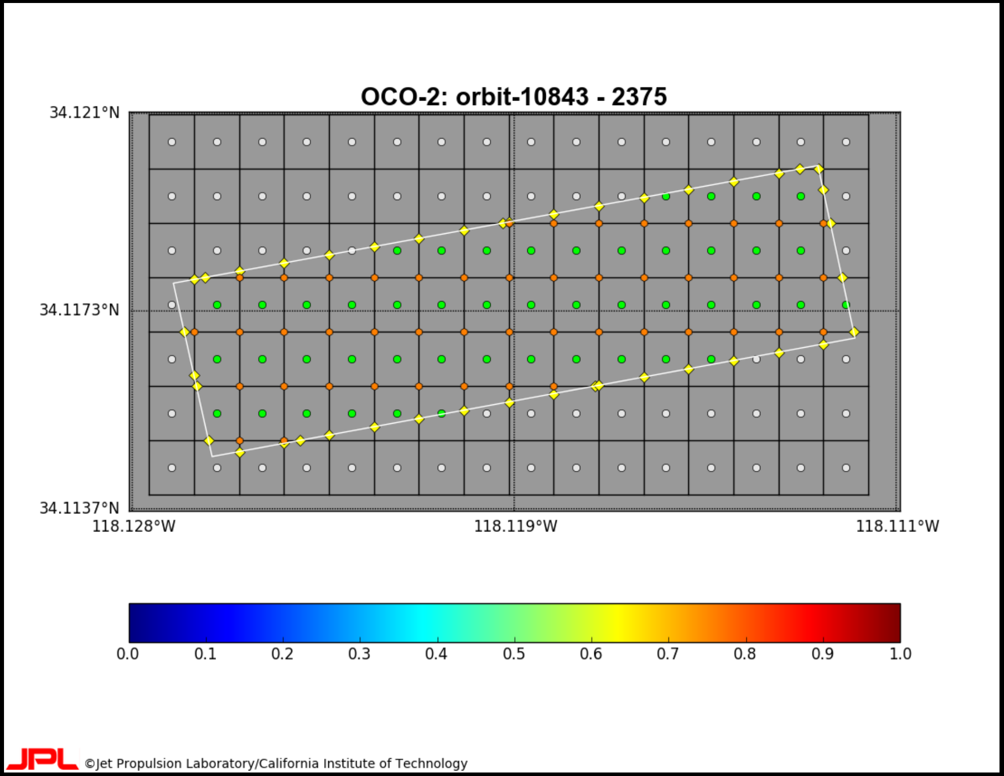
the tessellation process

2. define a rectangular grid, *e.g.*, 0.001°x0.001° resolution (~100 m)



the tessellation process

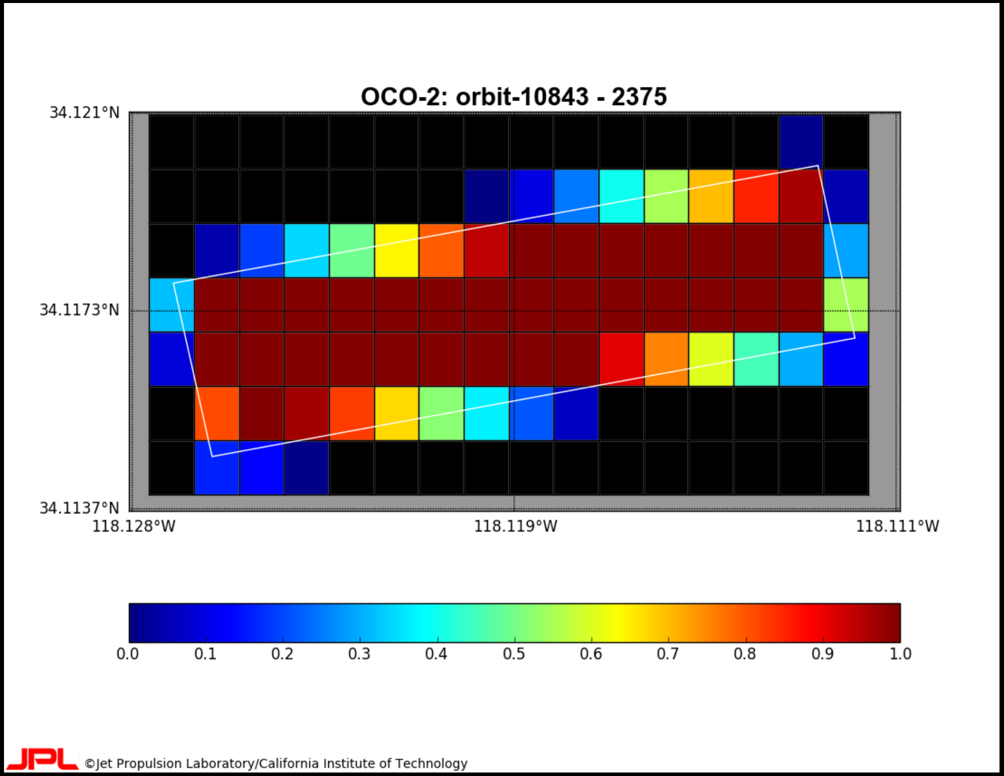
3.1 calculate intersects of footprint and grid cell boundaries



- line intersects
- grid cell centers inside footprint
- grid cell corners inside footprint

the tessellation process

3.2 Calculate covered cell area by built-in rules (triangles and rectangles)



tessellation – built-in rules

trivial cases:

- grid cell is completely outside footprint: F = 0
- grid cell is completely inside footprint: F = 1

simple cases:

- footprint boundary goes across a cell corner: F is a triangle
- footprint boundary slices cell east-west or north-south: F is a combination of a triangle and a rectangle

less simple case:

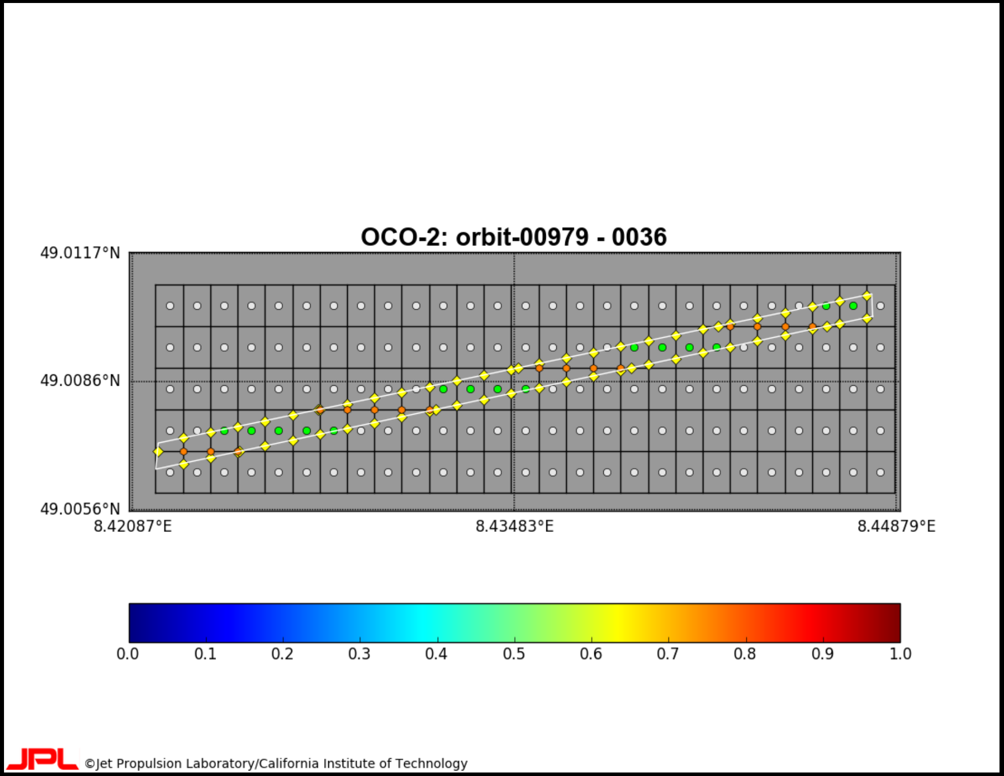
- a single footprint corner inside the grid cell: F is a combination of several triangles and rectangles

anything more complicated: “divide and conquer”

- recursive zoom processing: sub-divide cell with finer grid until built-in rules apply or we run up against the recursion limit

the tessellation zoom process

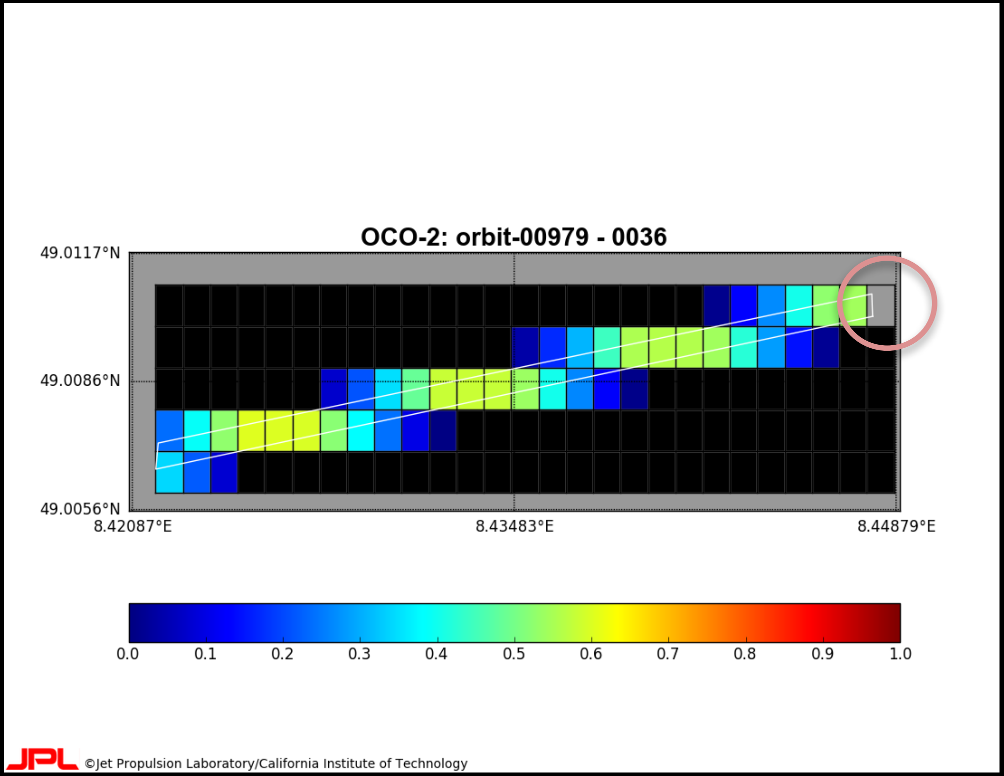
built-in rules for cell area computations don't cover all geometric cases



in the above case …

the tessellation zoom process

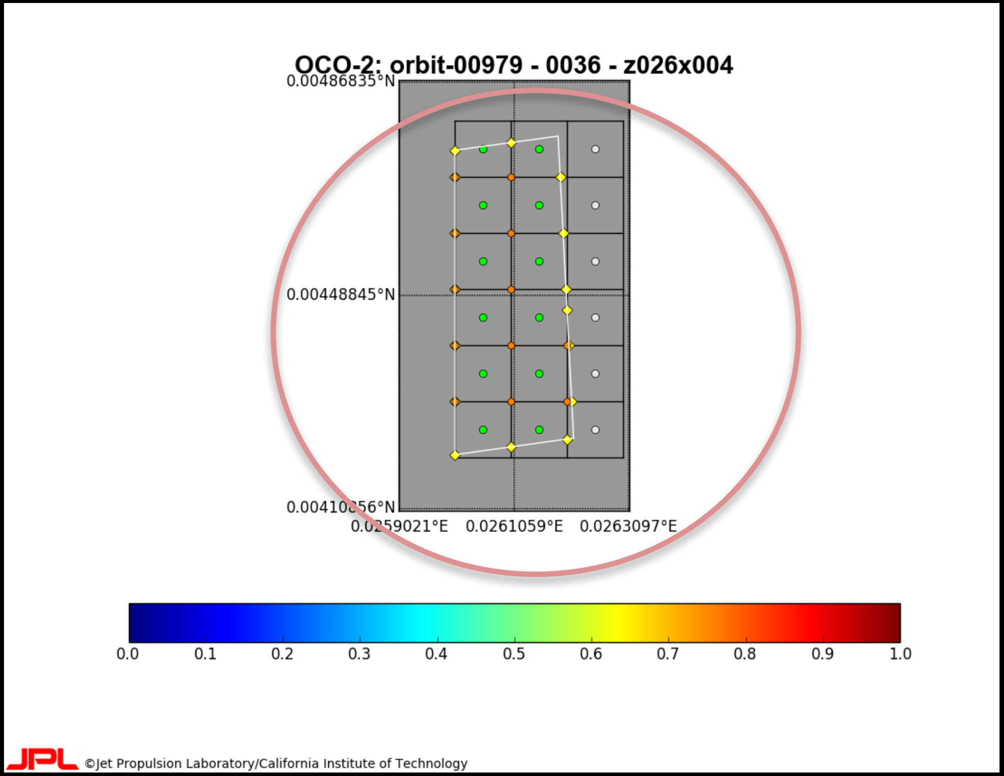
… the eastern-most grid cell can't be quanfied



all such unquantifiable grid cells enter zoom processing

the tessellation zoom process

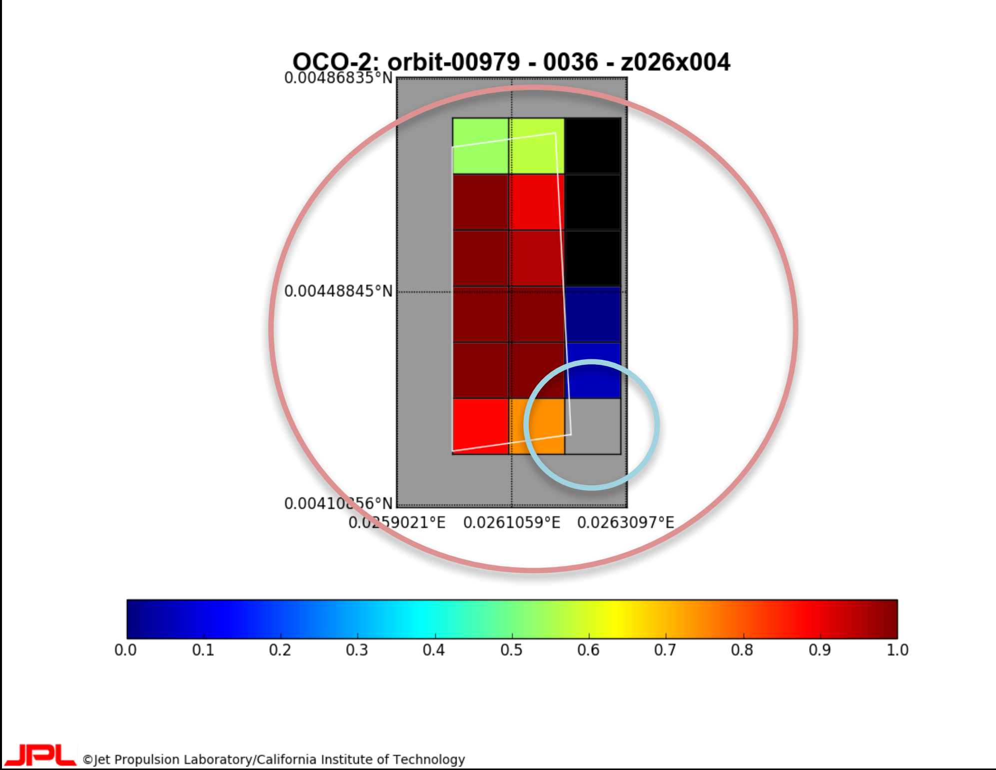
the grid cell is divided by a sub-grid and processed like the original footprint ...



… in the hope that this defines the coverage area.

the tessellation zoom process

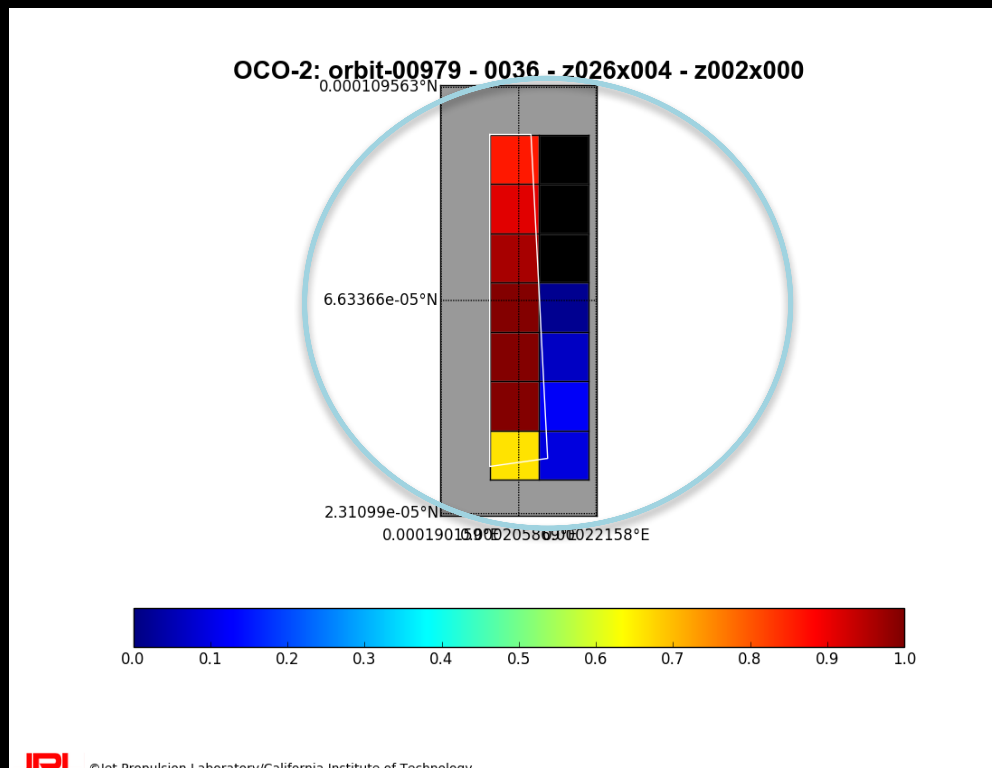
if this doesn't resolve the sub-cell, the process is continued recursively



… in the hope that THIS defines the coverage area.

the tessellation zoom process

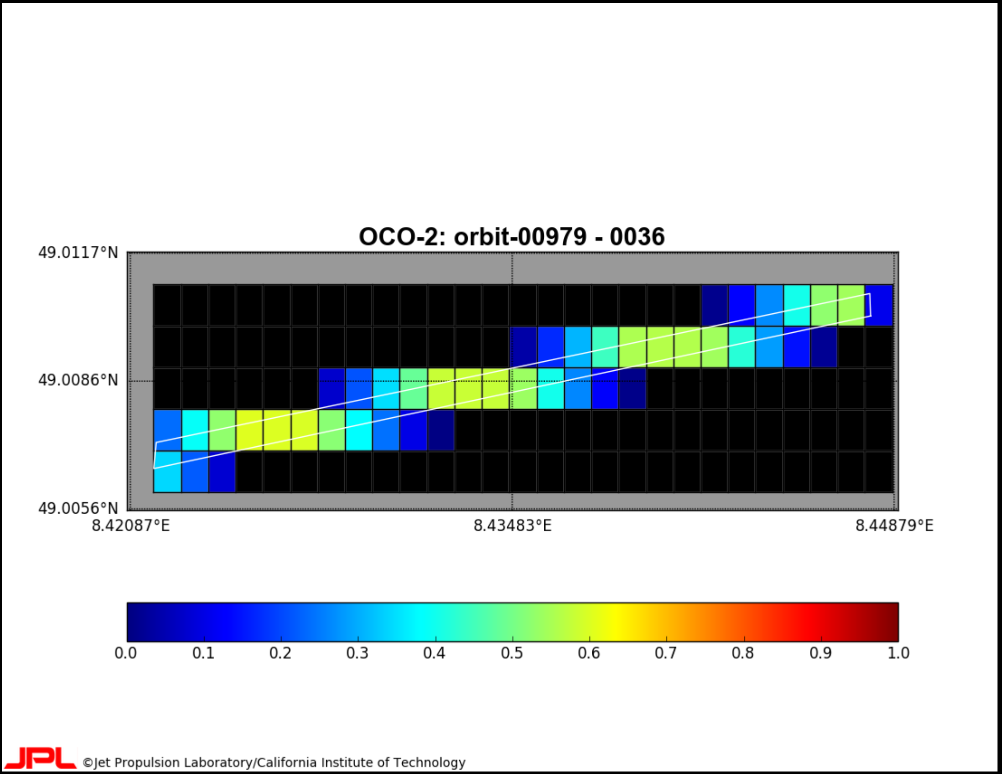
the recursive iteration stops either if the sub-sub(-sub…) cell is quantified ...



or until the limits of the recursion process are reached.

the tessellation zoom process

in this case, two iterative zoom processes resolved the grid cell.



the status quo

Python 3 implementation of a prototype algorithm, consisting of

core routines
which perform the mapping of the satellite footprint and calculate the fractional coverage of each rectangular grid cell underlying the footprint

wrapper routines
which interface the core routines with the satellite data and perform temporal and spatial averaging

successful applications
OCO-2 target data, OMI, OMPS, ASTER

from tessellation to gridding/averaging

for each satellite footprint, the calculated grid cell cover fractions *F*, (0 <= *F* <= 1) are used to weigh the contribution of the observation from this footprint to the total average.

overlapping footprints will update the the underlying grid cells that they share, and the total grid cell area *A* is usually >=1, depending on how many satellite footprints in the time series covered this particular cell.

A replaces the "total number of samples" in the final averaging step.

the data averaging process

the core routines only compute the fractional coverage of each grid cell underlying the satellite footprint

the actual averaging is done in the wrapper routine;
for OMI and OCO-2, this consisted of a weighted averaging, with the weights composed as

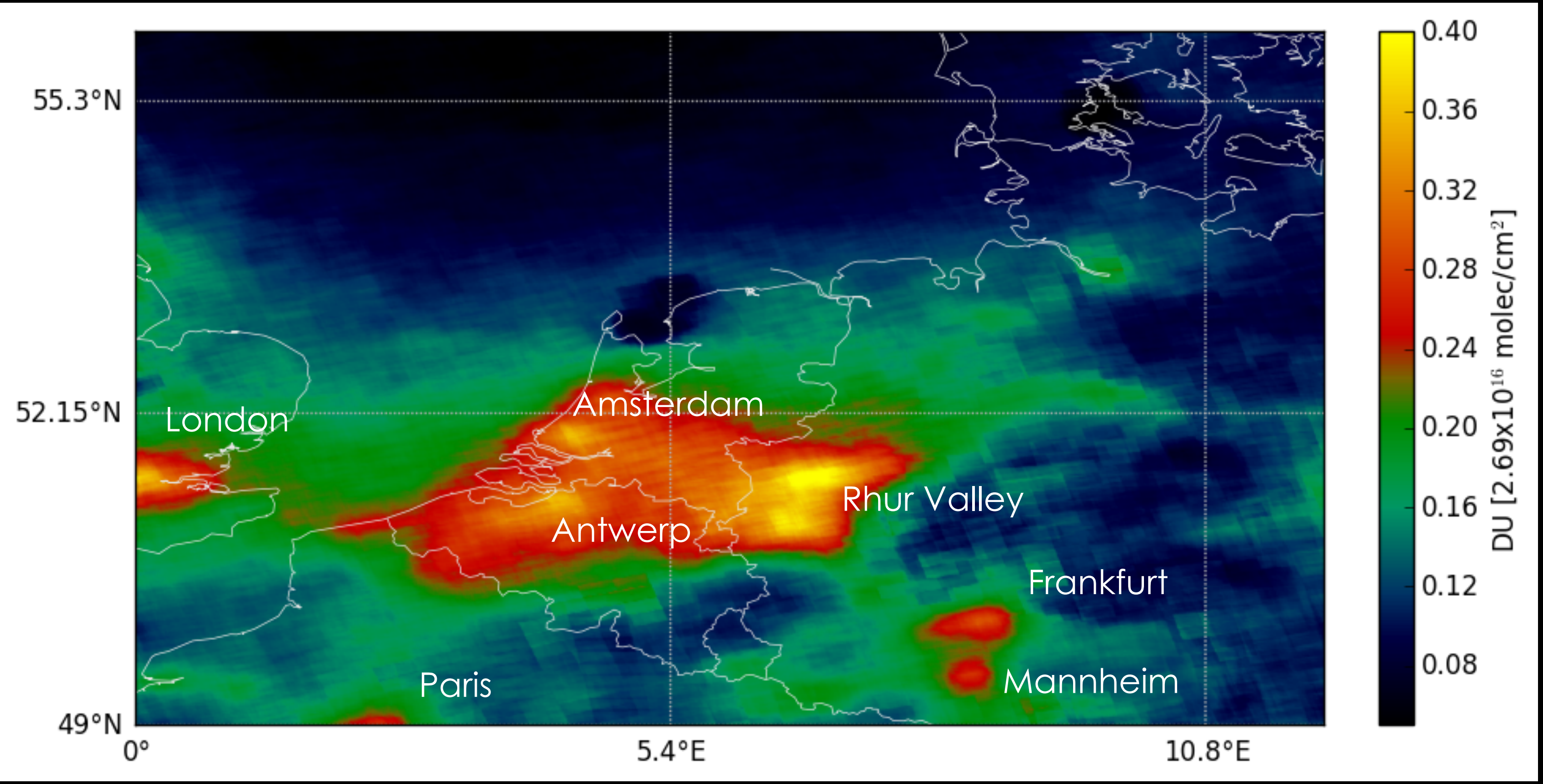
(1) proportional to the grid cell coverage
(2) inversely proportional to footprint size, or $1/\cos(\Theta_{view})$

data averaging – examples

instrument	native resolution	gridded resolution m/km	gridded resolution degree
OMI	340 km^2 – 4,600 km^2	1x1 km^2	0.01°x0.01°
OCO-2 (target data)	1x1 km^2	100x100 m^2	0.001°x0.001°
ASTER	90x90 m^2	10x10 m^2	0.0001°x0.0001°

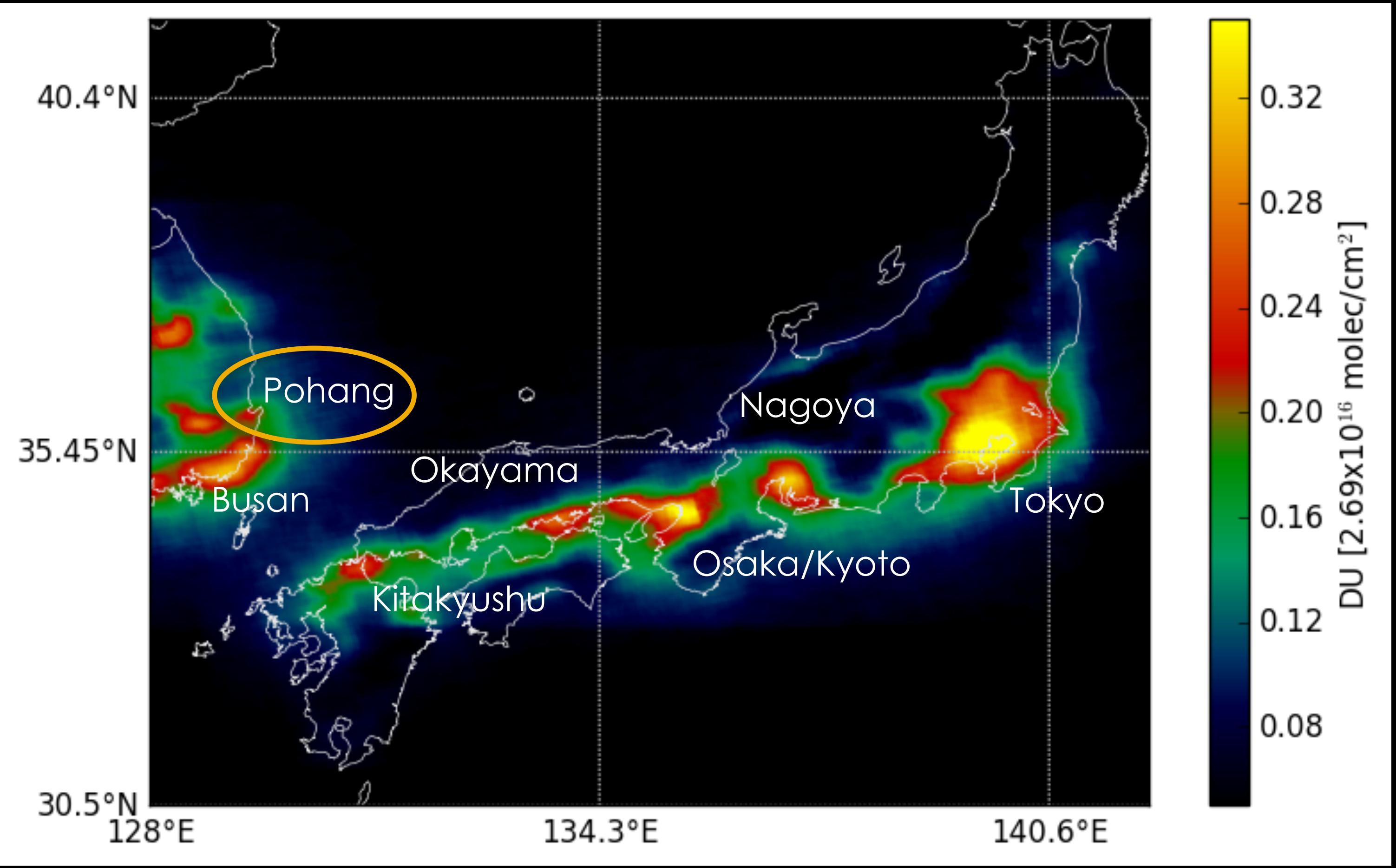
OMI tropospheric NO_2 – 2006, 0.01°x0.01°

Northern Europe



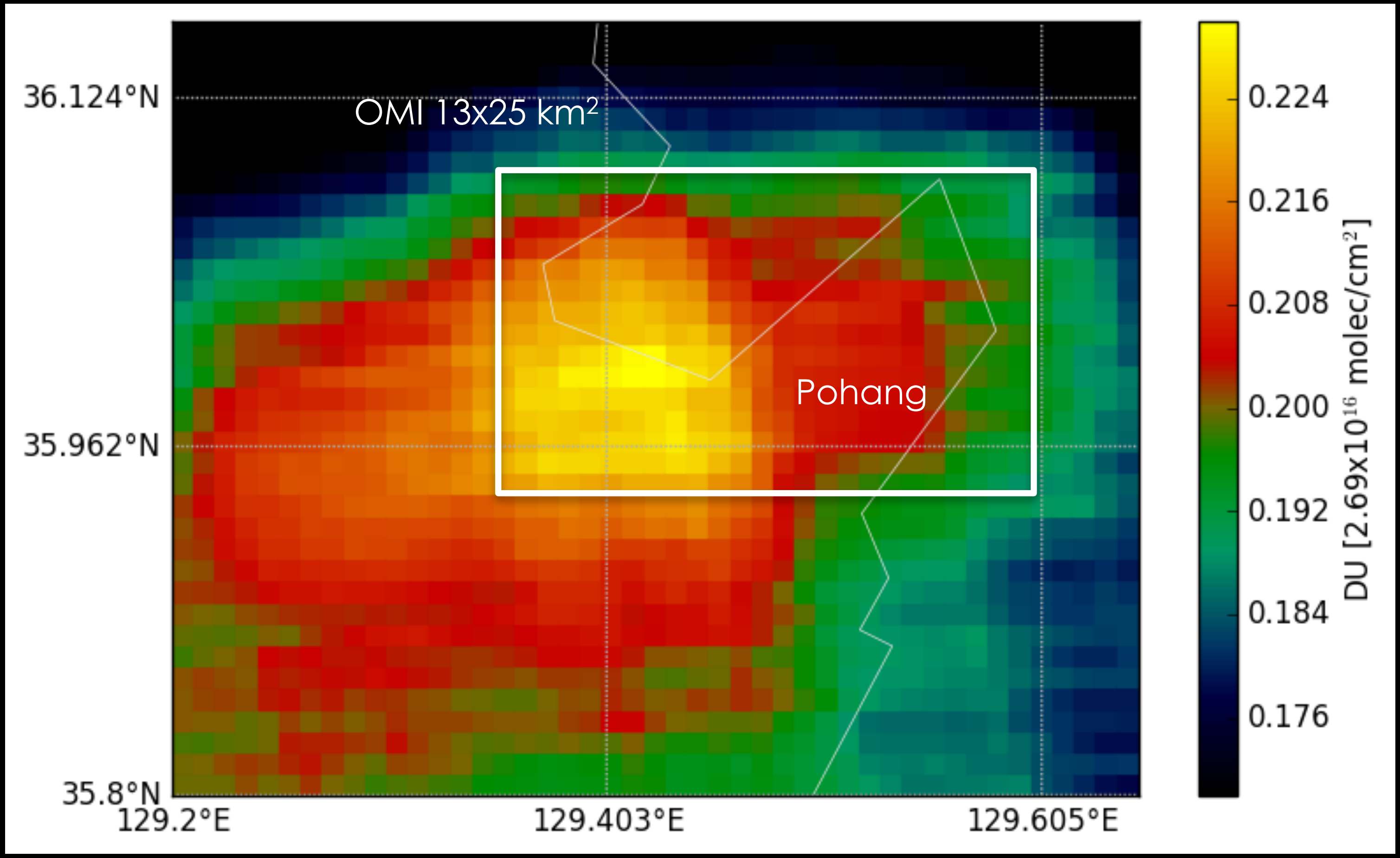
OMI tropospheric NO_2 – 2006, 0.01°x0.01°

Japan



OMI tropospheric NO_2 – 2006, 0.01°x0.01°

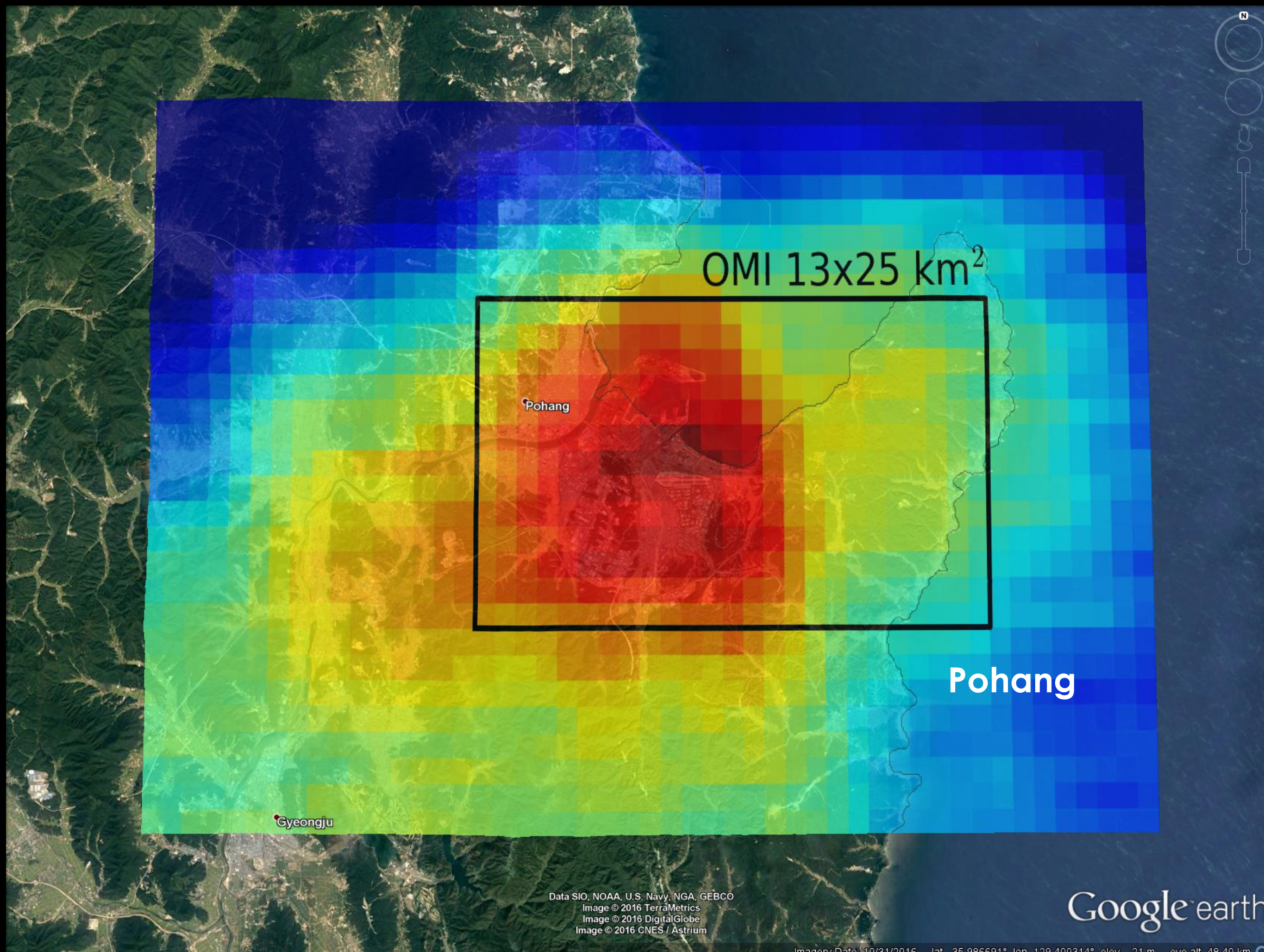
Pohang, Korea



13x25 km^2 is the smallest ground footprint size in the OMI swath

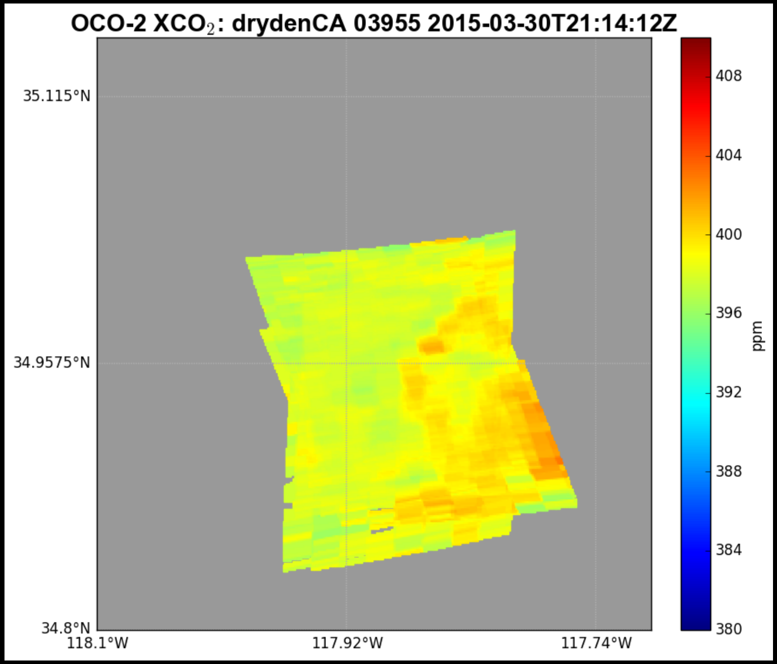
OMI tropospheric NO_2 – 2006, 0.01°x0.01°

google earth overlay

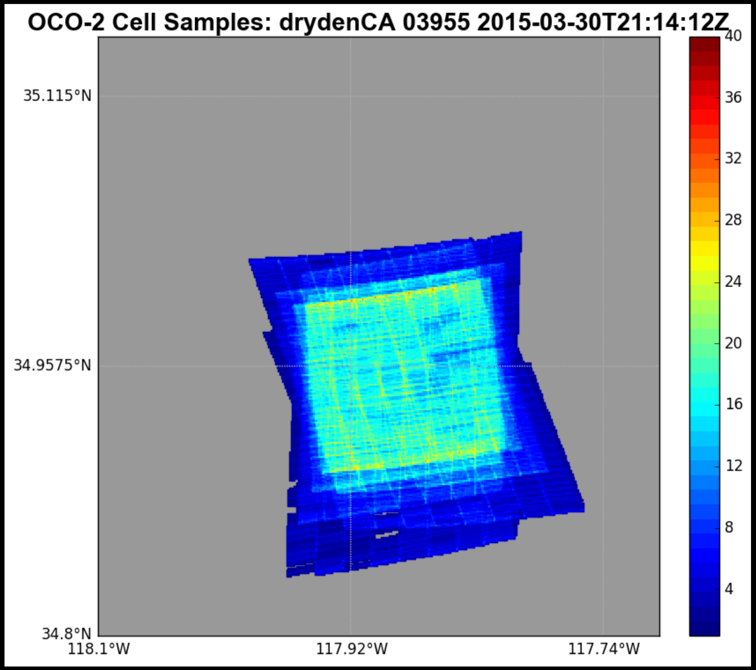


OCO-2 XCO_2 target data over Edwards AFB, CA 0.001°

single target overpass



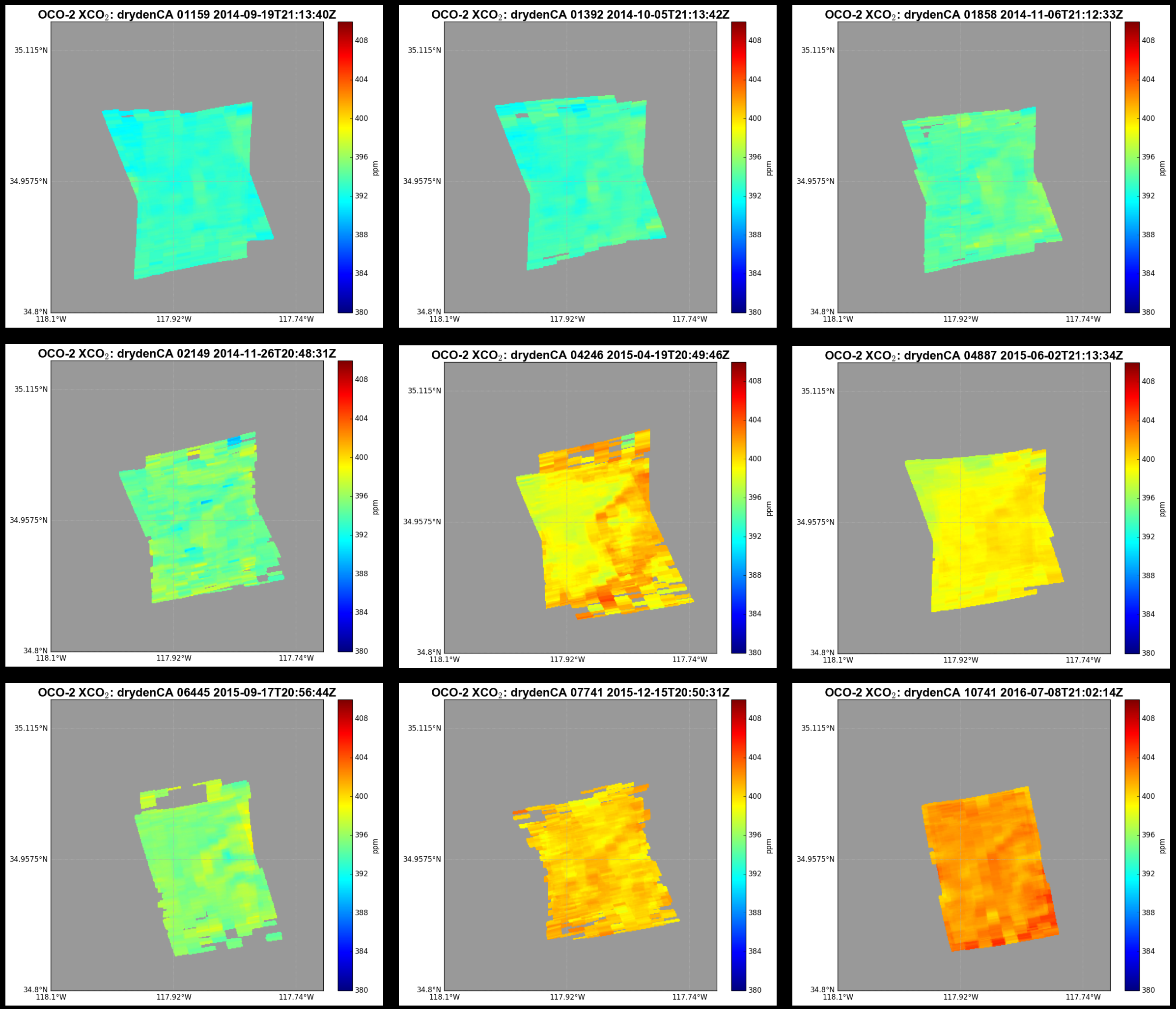
number of cell samples



Jet Propulsion Laboratory
California Institute of Technology

OCO-2 XCO_2 target data over Edwards AFB, CA 0.001°

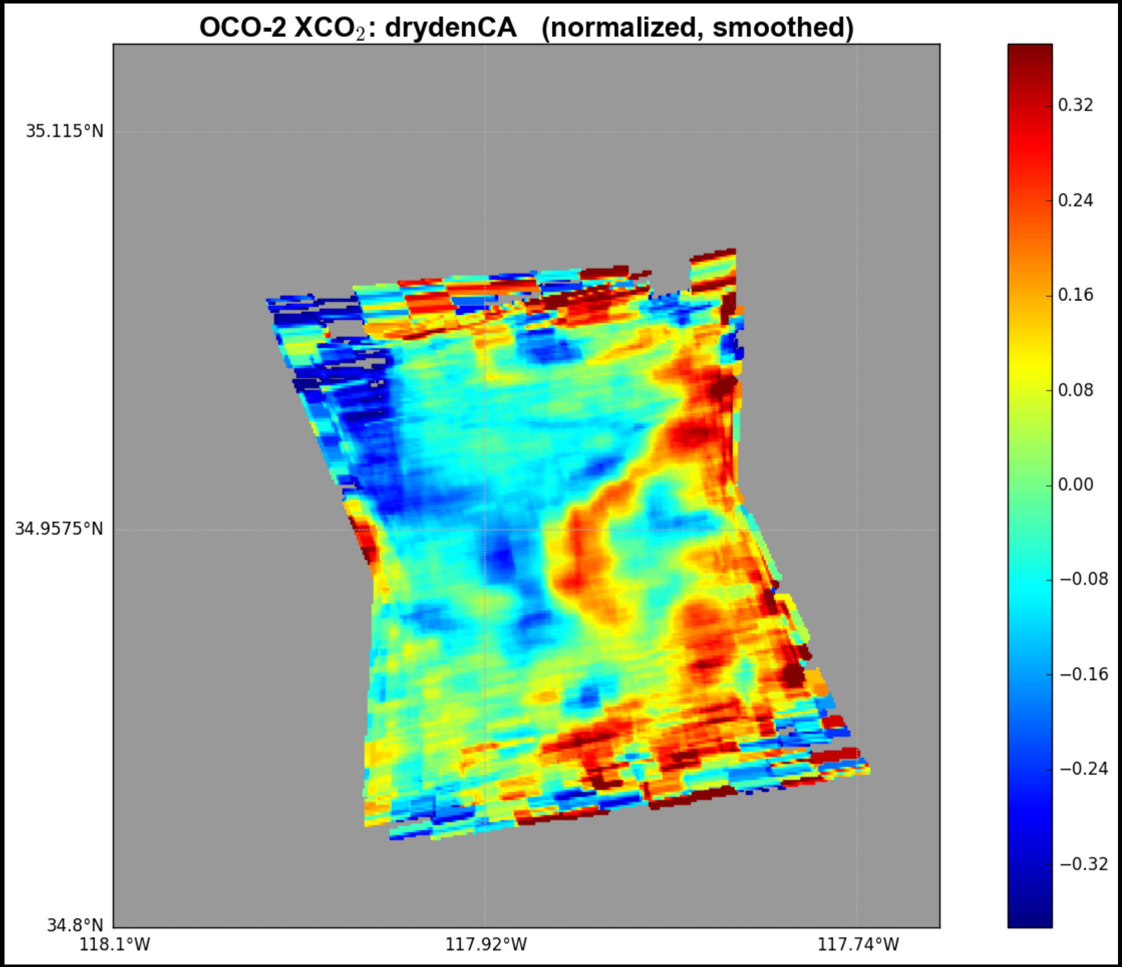
single-day target overpasses



Jet Propulsion Laboratory
California Institute of Technology

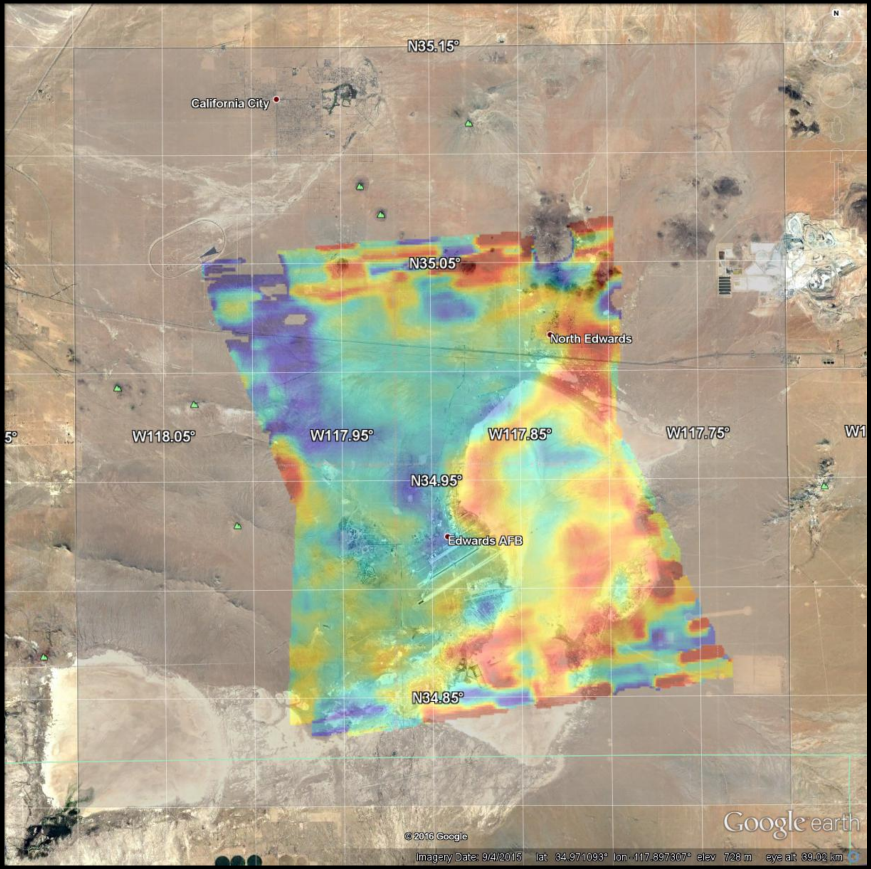
OCO-2 XCO_2 target data over Edwards AFB, CA 0.001°

normalized difference to the combined de-seasonalized median

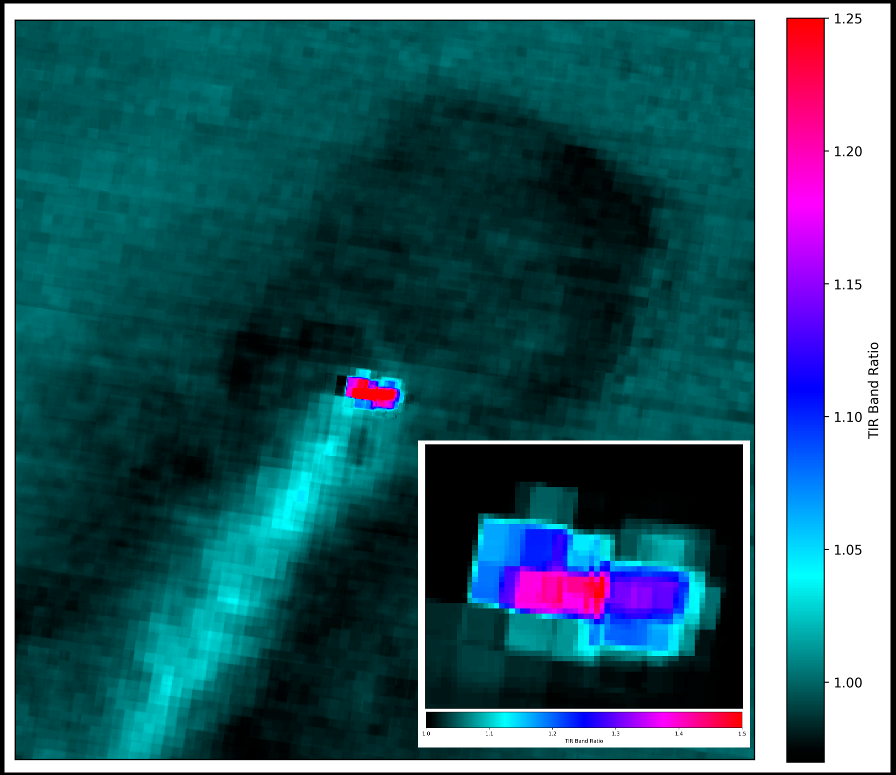


OCO-2 XCO_2 target data over Edwards AFB, CA 0.001°

google earth overlay



ASTER band ratio data, Kilauea Crater, Hawai'I 0.0001°



ASTER band ratios can be used as a proxy for SO_2 emissions

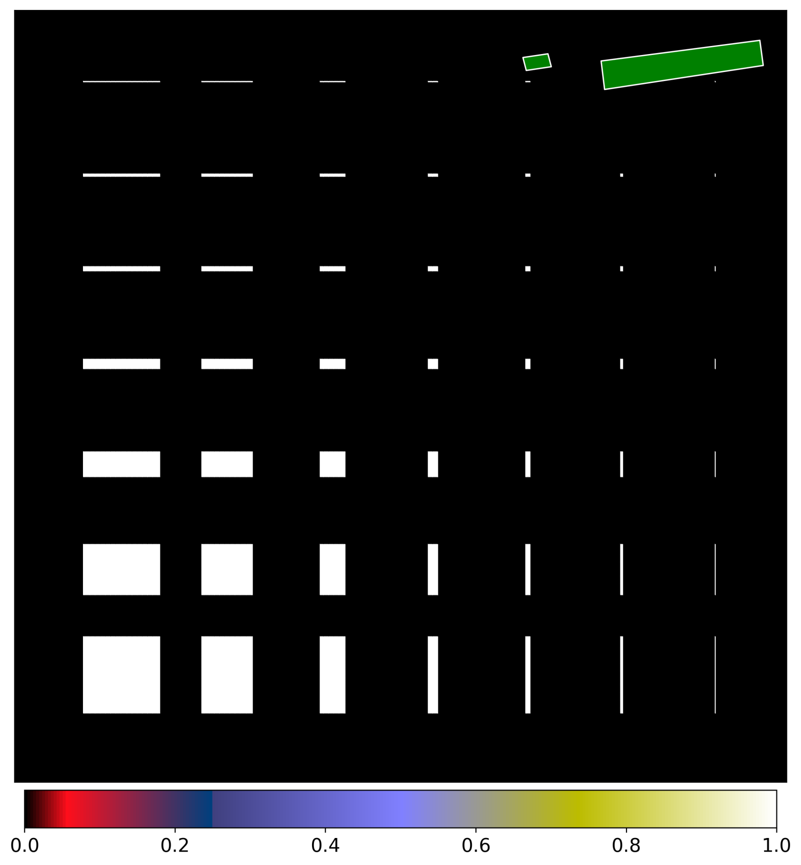
theoretical examples – gridding test patterns

define geometric test patterns to be sampled with satellite footprints.

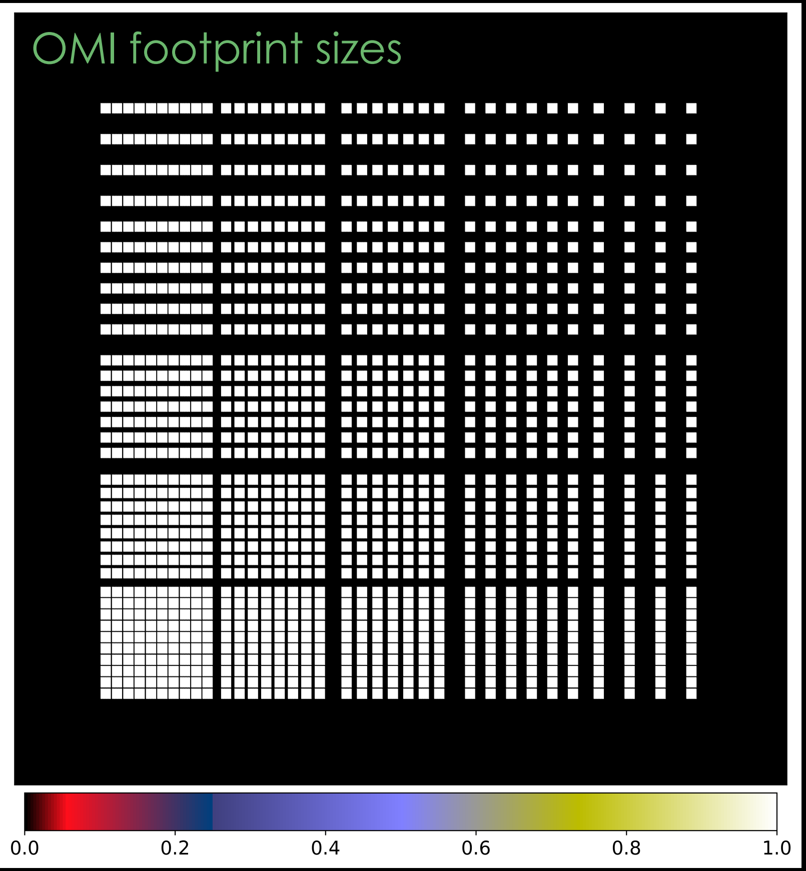
compare original and sampled (*e.g.*, averaged) pattern in the hope to find a metric for the "quality" of satellite observations.

(1) define pattern on a high-resolution, regular grid and place it on the Earth's surface
(2) run tessellation "in reverse", *i.e.*, compute the composite signal in a satellite footprint derived from that test pattern
(3) average those derived footprint signals onto the same regular. grid as the original test pattern

theoretical examples – gridding test patterns

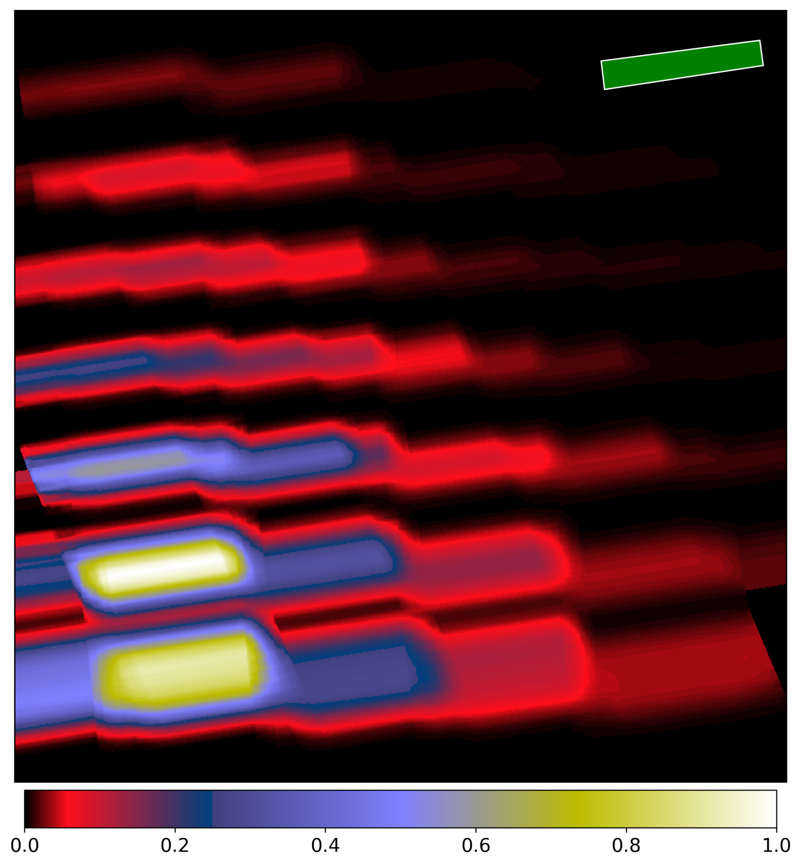


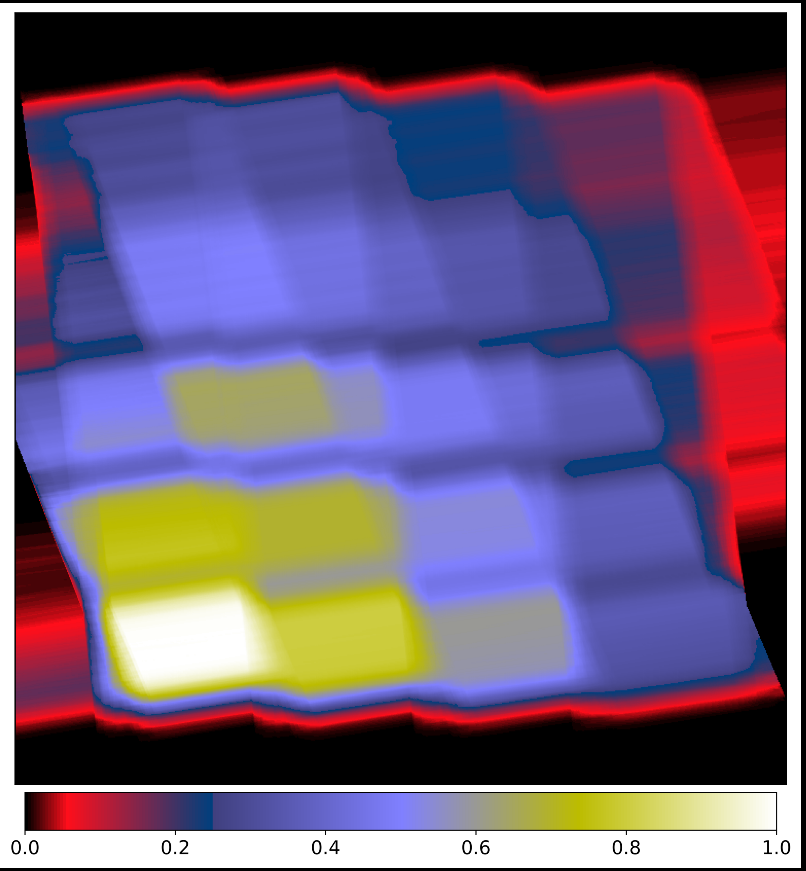
Test Pattern 1: Rectangles of 1 3 5, 10, 25, and 50 km width



Test Pattern 2: Squares of 10x10 km^2 spaced 1, 3, 5, 10, and 20 km apart

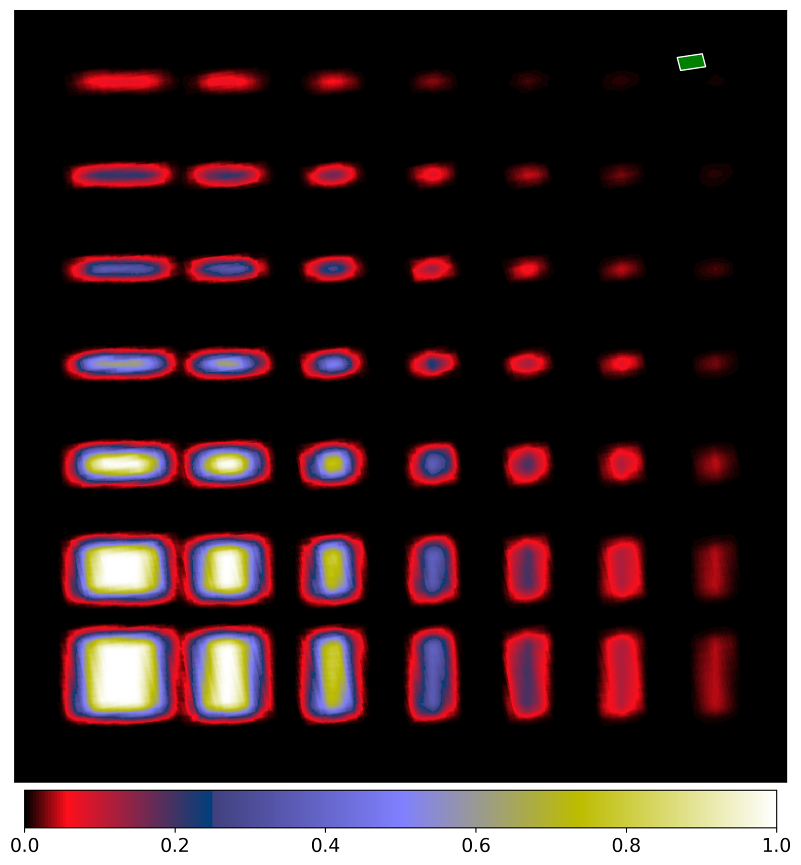
theoretical examples – gridding test patterns

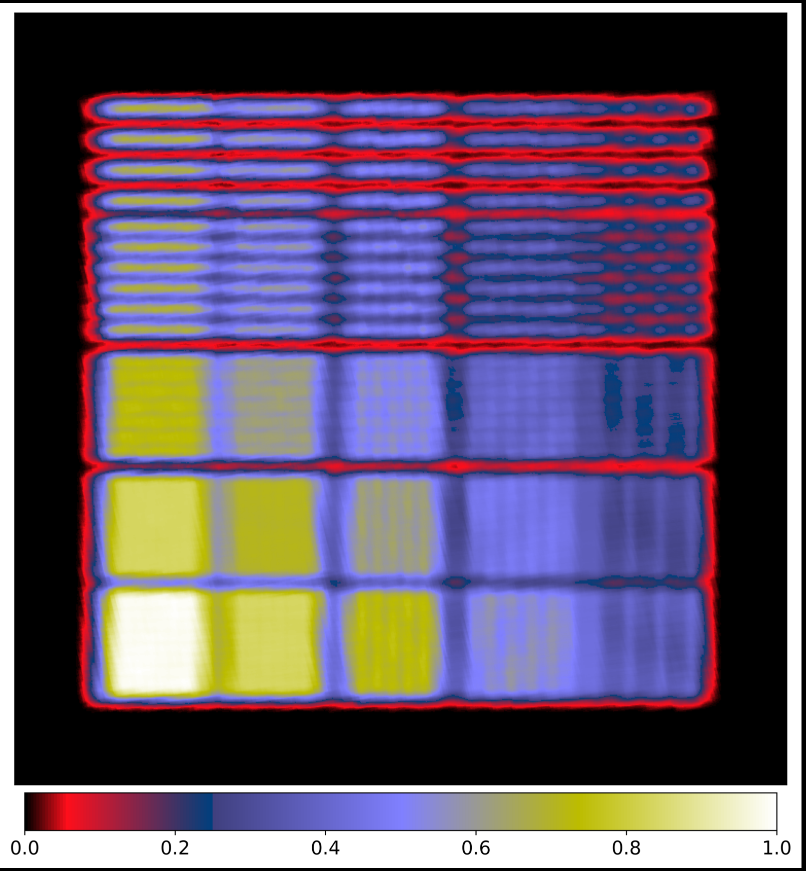




only largest OMI pixels (swath edges)

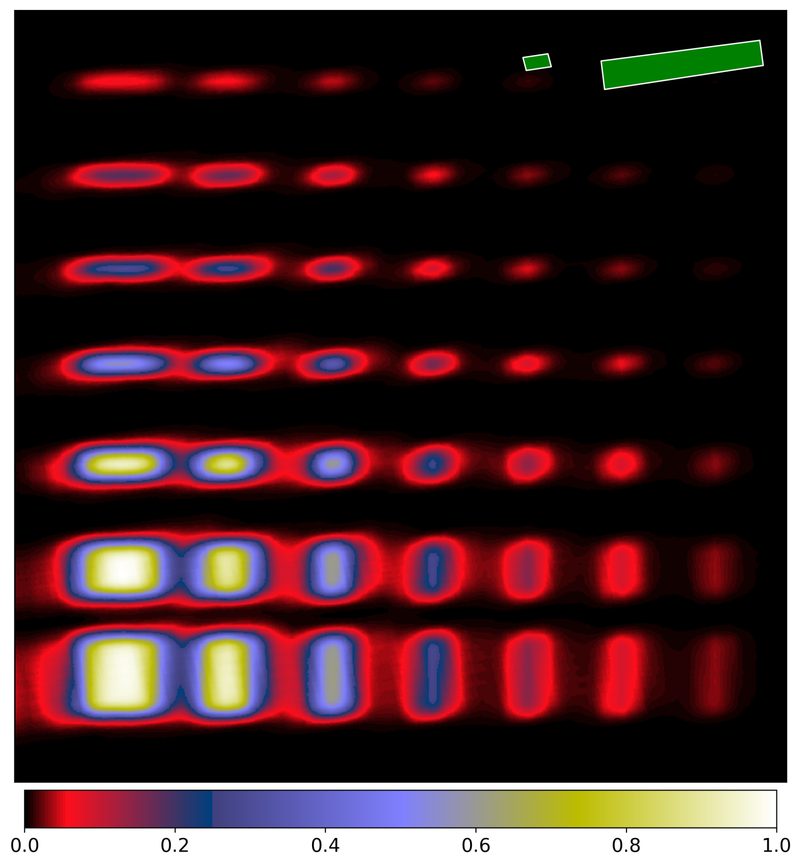
theoretical examples – gridding test patterns

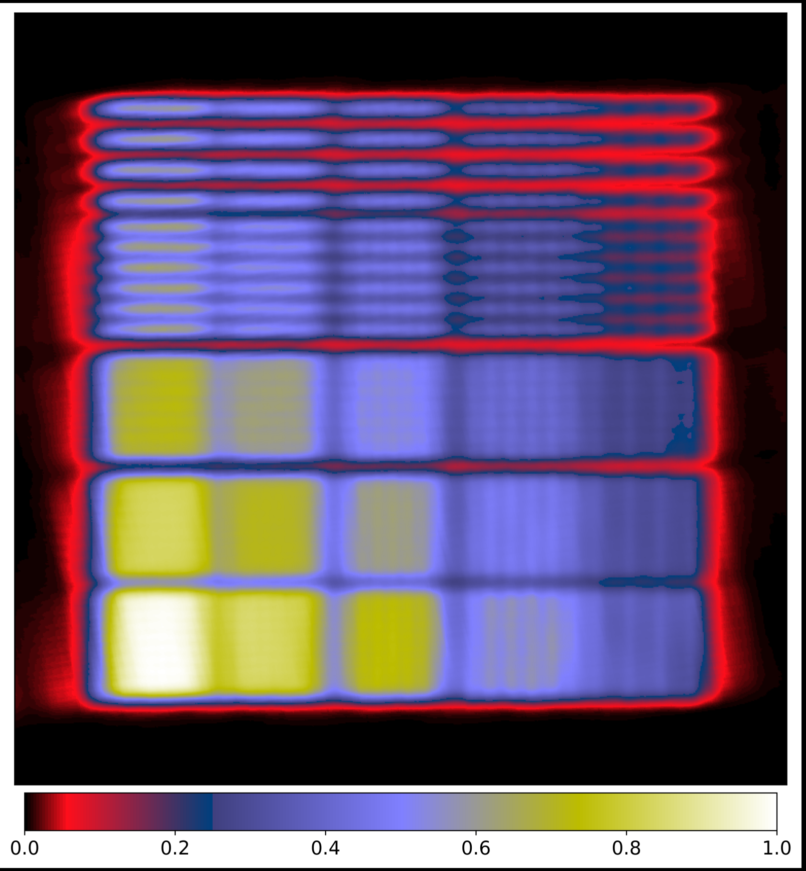




only smallest OMI pixels (swath center)

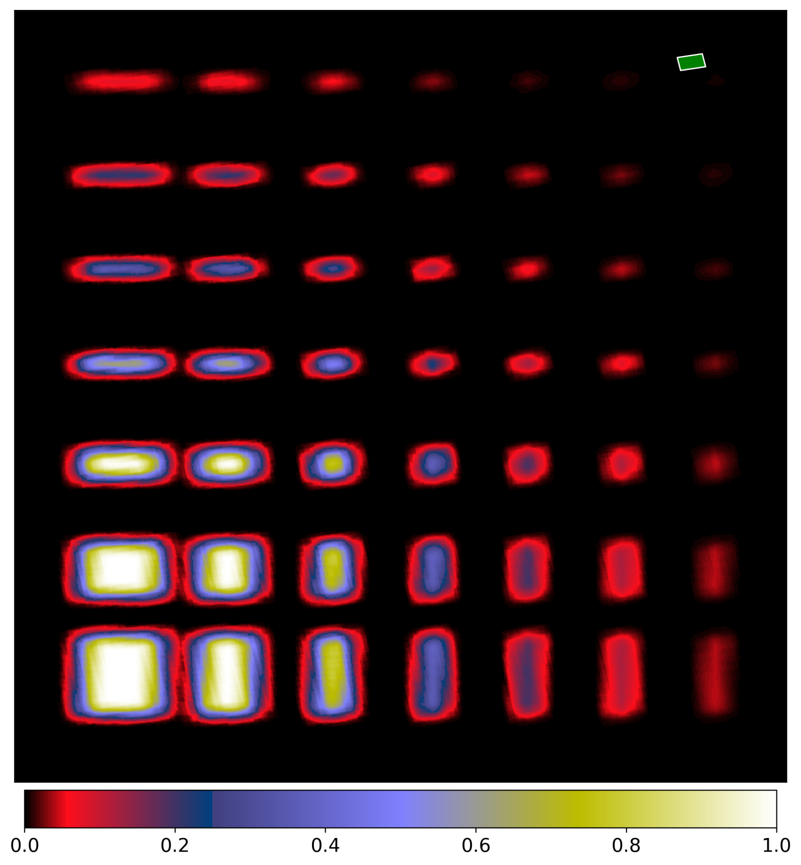
theoretical examples – gridding test patterns

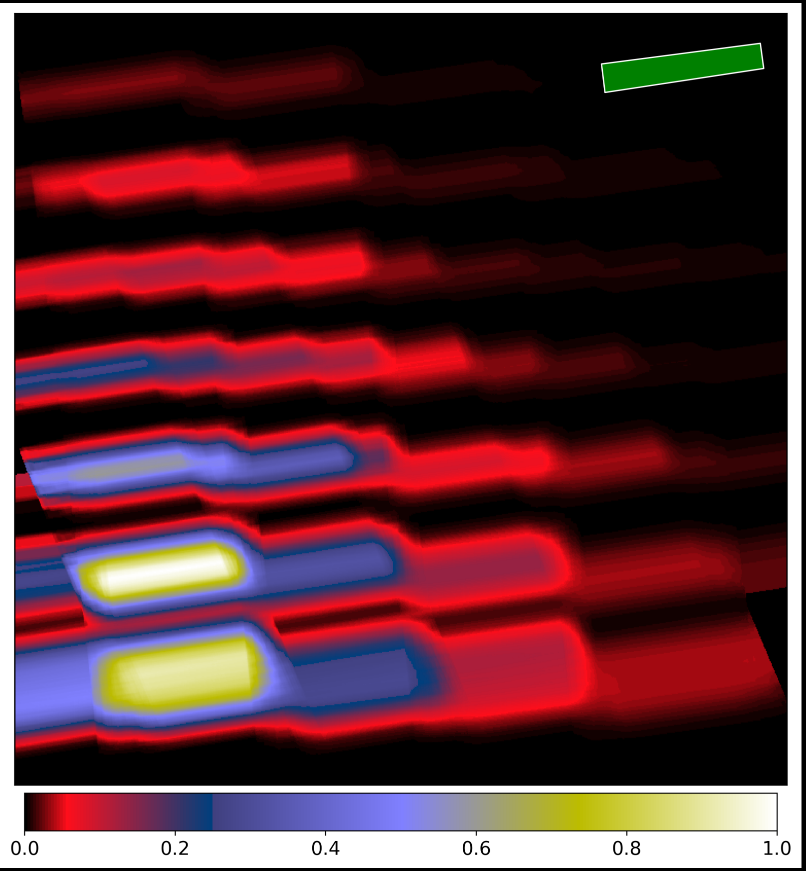




all OMI pixels (area-weighted)

theoretical examples – gridding test patterns





“smallest” *vs.* “largest” pixels – how to define a metric of pattern conservation?
ideas welcome!

next steps

prototype implementation needs to be improved
(robustness, turn-key/black-box, operational)

speed-up required
(execution times are too long; cython?)

beta-testers welcome!
(tessellate data at your own peril)

thank you