STARE dataframes for geospatial analysis

- a high level STARE interface -

Niklas Griessbaum[1] James Frew[1], Michael Lee Rilee[2], Kwo-Sen Kuo[3], James Gallagher[4], Kodi Neumiller[4]
[1]Bren school of Environmental Science UCSB, [2]Rilee Systems Technologies LLC, [3]Bayesics LLC, [4]OPeNDAP Inc

STARE

The Spatio-Temporal Adaptive Resolution Encoding (STARE) is a hierarchical geospatiotemporal referencing scheme. The spatial component of STARE is based on a hierarchical triangular mesh (HTM). The base of the HTM is an octahedron comprised of eight spherical triangles, named S0 through S3 and N0 through N3 (S for south, N for north). Each triangle is recursively subdivided into four smaller triangles. The name/ID of a new triangle is composed by appending the triangle number to the parent triangle's name. At the 24th level of subdvision, the triangle size is ≈ 1 m.

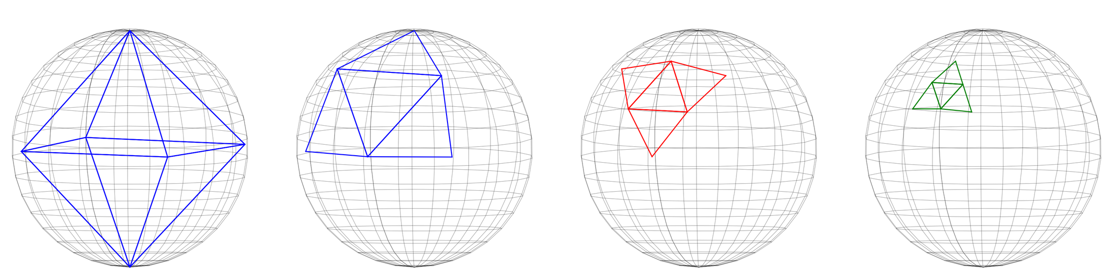


Figure: HTM Schema

STARE dataframes

PySTARE is a Python module that exposes the STARE library to Python. STARE dateframes further abstract PySTARE and provide high-level functions for users to explore and interact with STARE.

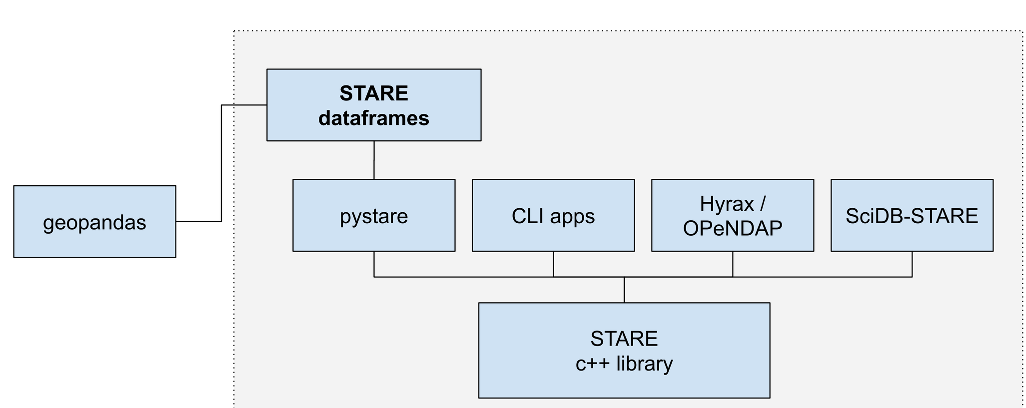


Figure: STARE dataframes

STARE dataframes represent geometries as sets of STARE triangles or "trixels" (analogous to GeoPandas geodataframes which represent geometries as WKT.) In STARE dataframes, points are represented as STARE trixels at the HTM tree's leaf level. Polygons are represented as sets of STARE trixels that cover the polygon. The trixels are stored as integerized STARE index values.

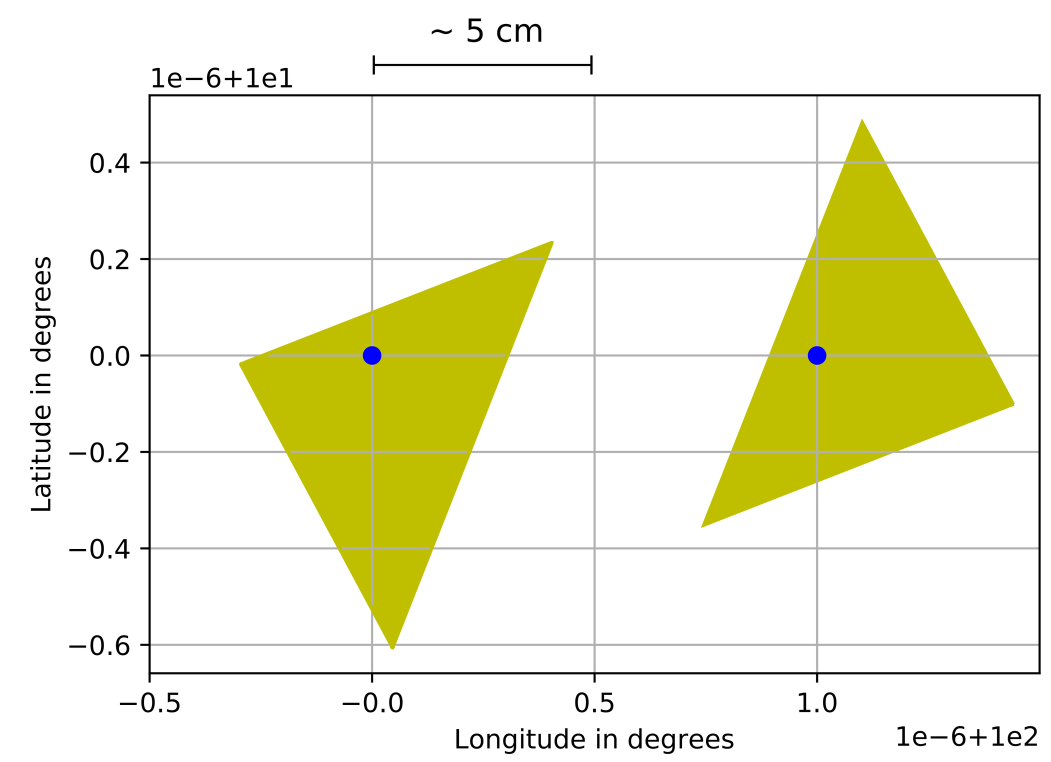


Figure: The blue points (10.0°N, 100.0°E) and (10.0°N, 100.000 001°) represented as STARE leaf trixel (yellow).

STARE dataframe methods

StareDataFrame.stare_intersects(self, other)

Spatial STARE intersects test. Returns a boolean Series with value True for each geometry that intersects other.

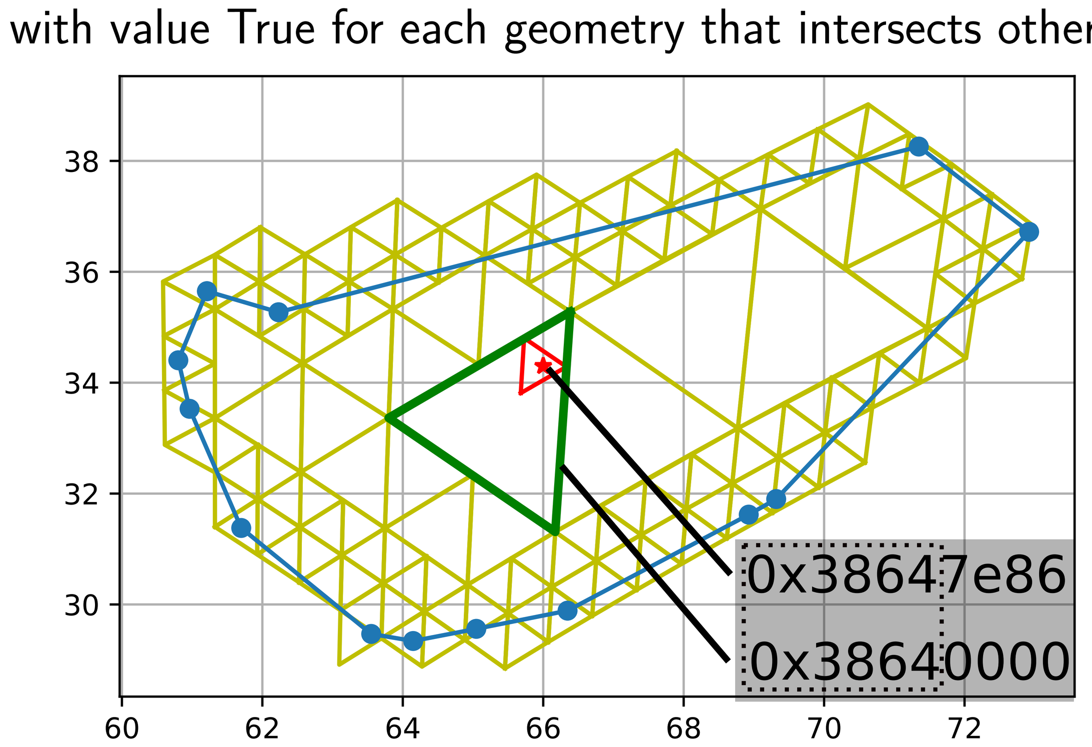


Figure: STARE intersects test

StareDataFrame.plot(self, **kwargs)

Generates a plot of the trixels representing each geometry.

starepandas.stare_join(left, right)

STARE spatial left join of left and right STARE dataframe on the predicate 'intersects'.

StareDataFrame.stare_intersection(self, other)

Returns the set of index values / trixels that represent the intersection for each geometry with other.

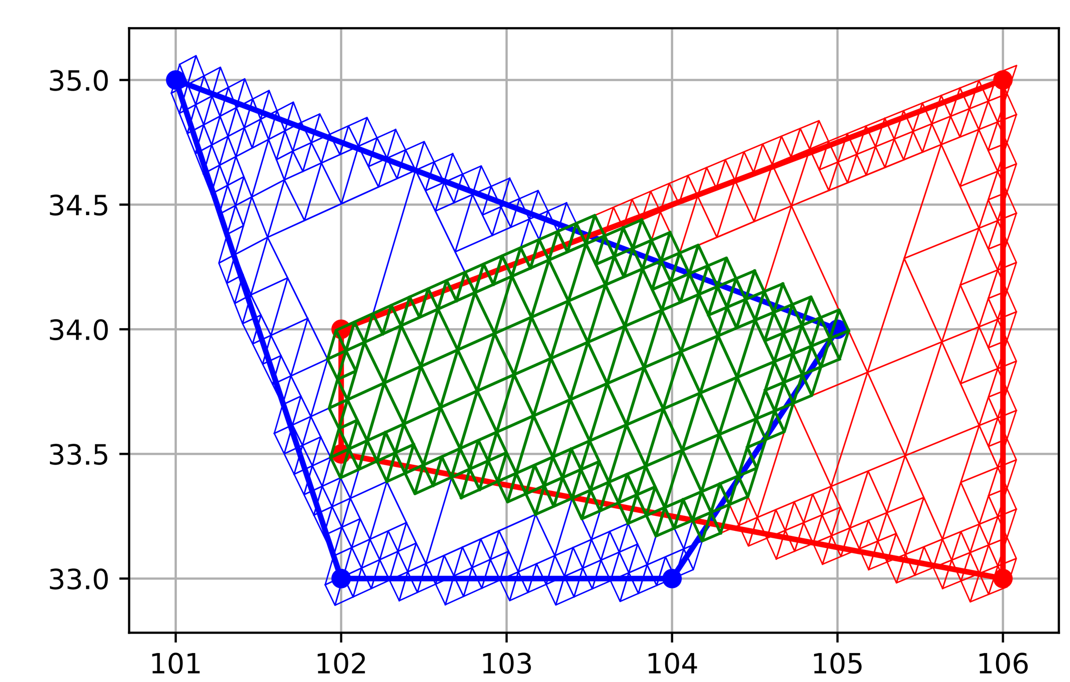


Figure: STARE intersection (green) of two STARE dataframes (red and blue)

Example: SWATH classification

Loading spatial regions

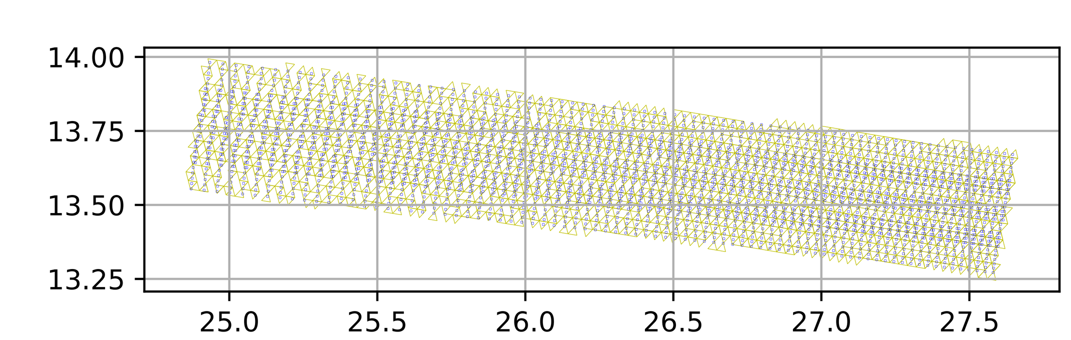
```
import starepandas

countries = starepandas.read_file('regions.gpkg')
countries = countries.add_stare(level=12)
countries.head()
```

name	pop_est	iso_a3	gdp_md_est	stare
Algeria	40969443	DZA	609400.0	[4485585228861014020, 4492340628302069764, 422...
Angola	29310273	AGO	189000.0	[4464281091561226248, 4464659323561181192, 446...
Benin	11038805	BEN	24310.0	[4481363104210354183, 4481398288582443015, 448...
Botswana	2214858	BWA	35900.0	[21392098230009861, 21955048183431173, 2589569...
Burkina Faso	20107509	BFA	32990.0	[4470526317606993926, 4470667055095349254, 447...

Loading MOD09 Modis SWATH data

```
file_path = 'MOD09.A2019317.0815.006.2019319020759.hdf'
modis = starepandas.read_mod09(file_path)
modis = modis.add_stare()
modis = modis.set_index('stare')
modis.trixels().plot()
```



Calculating the NDVI, spatial join and mean NDVI

```
modis['ndvi'] = (modis.b2−modis.b1)/(modis.b2+modis.b1)
modis = starepandas.stare_join(modis, regions)
modis_grouped = modis.groupby('country').mean()
modis_grouped.head()
```

Takeaway and Future

STARE dataframes expose STARE functionality on a high level through a well-known syntax, simplifying exploration and facilitating understanding of STARE. STARE dataframes may provide a performant alternative for tasks such as spatial classification of diverse data. Future improvements will address:

1. Dask Integration

Since spatially coincidental STARE index values share the same predicate, STARE can be used for distributed co-alignment.

2. Non-Convexity

STARE currently can only index the convex hulls of polygons.

3. Add spatial and temporal methods

Integration of further spatial relationship tests and spatial operations.

Checkout starepandas on github:

`github.com/NiklasPhabian/starepandas`

Acknowledgments

NASA ACCESS program provided support to this effort, award ID 80NSSC18M0118.