Grid Coverage Cell Span

(or Point vs. Area)

Proposal to add an optional concept of
a cell span for grid coverages to the
Coverage Implementation Schema

Knowing what values mean

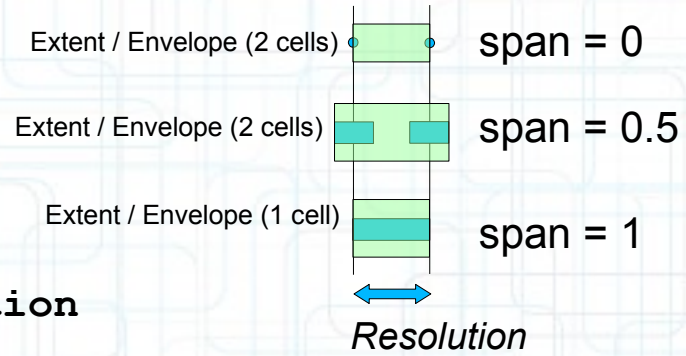
- There is a need to store and communicate precisely which portion of an axis values correspond to, e.g. whether they represents single points, or a certain area.
- For areas, this could potentially mean different things, e.g. an average, a minimum or maximum value, or an indication of spatial or temporal accuracy.

The Present Situation

- In CIS general grid coverage, each regular axis has:
  - `lowerBound`
  - `upperBound`
  - `resolution`
  - `uomLabel`
- Some standards like GeoTIFF and the GeoPackage gridded coverage extension have a concept of *Point* vs. *Area*
  - Some have 2 flavors of Point: *Corner* vs. *Center*
  - *Corner* simply adds a ½ resolution offset to the geospatial position represented by the point

Adding `span` to *RegularAxis*

- By adding a **`span`** factor relative to the axis resolution to *RegularAxis* we could store this information
- A **`span`** of 0 would indicate a point
- A **`span`** of 1 would indicate an area spanning the resolution between two cell values
- Values in between could be used
- This would support multi-dimensional scenarios



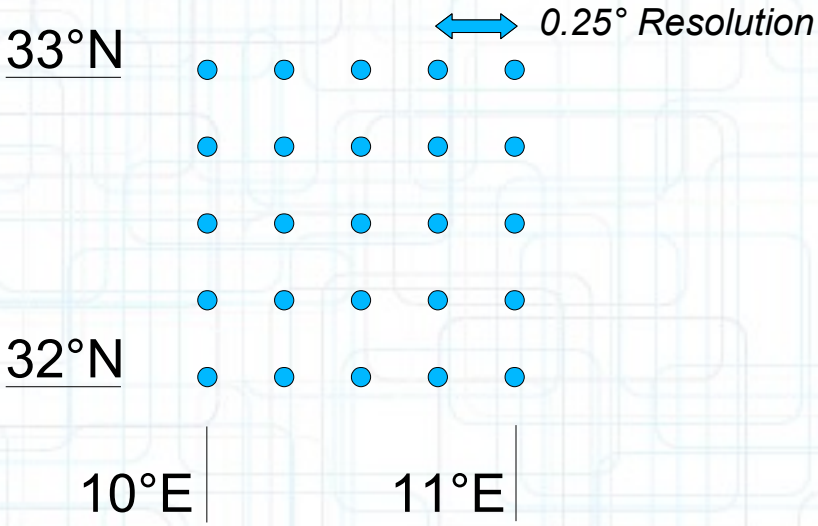

```
extent = ((numCells-1) + span) * resolution

Point (span=0): extent = (numCells-1) * resolution
Area  (span=1): extent = numCells     * resolution
```

Envelope / Extent / Grid axes

(32°N, 10°E) – (33°N, 11°E)

span = 0

(values represent theoretically infinitely small points)




```
axisLabel: Lat
lowerBound: 32.0
upperBound: 33.0
uomLabel: deg
resolution: 0.25
span: 0.0
```

```
axisLabel: Lon
lowerBound: 10.0
upperBound: 11.0
uomLabel: deg
resolution: 0.25
span: 0.0
```

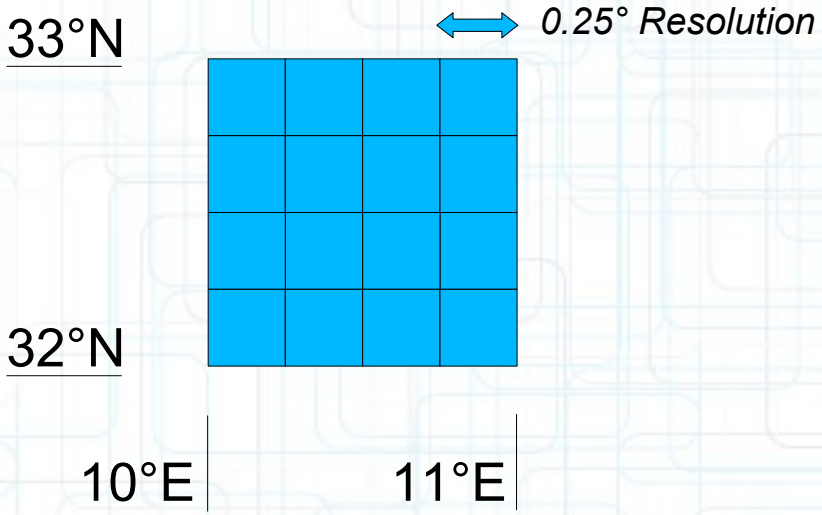
```
axisLabel: j
lowerBound: 0
upperBound: 4
```

```
axisLabel: i
lowerBound: 0
upperBound: 4
```

(32°N, 10°E) – (33°N, 11°E)

span = 1

(values represent e.g. an average over an area resolution x resolution)




```
axisLabel: Lat
lowerBound: 32.0
upperBound: 33.0
uomLabel: deg
resolution: 0.25
span: 1.0
```

```
axisLabel: Lon
lowerBound: 10.0
upperBound: 11.0
uomLabel: deg
resolution: 0.25
span: 1.0
```

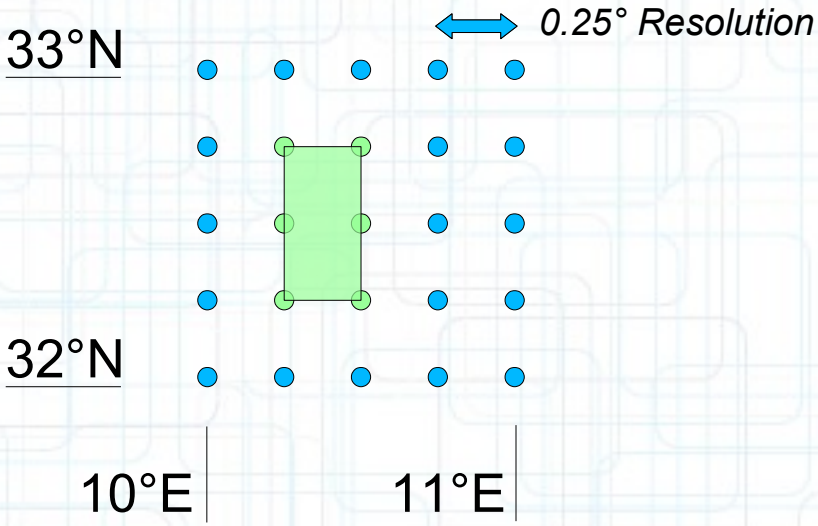
```
axisLabel: j
lowerBound: 0
upperBound: 3
```

```
axisLabel: i
lowerBound: 0
upperBound: 3
```

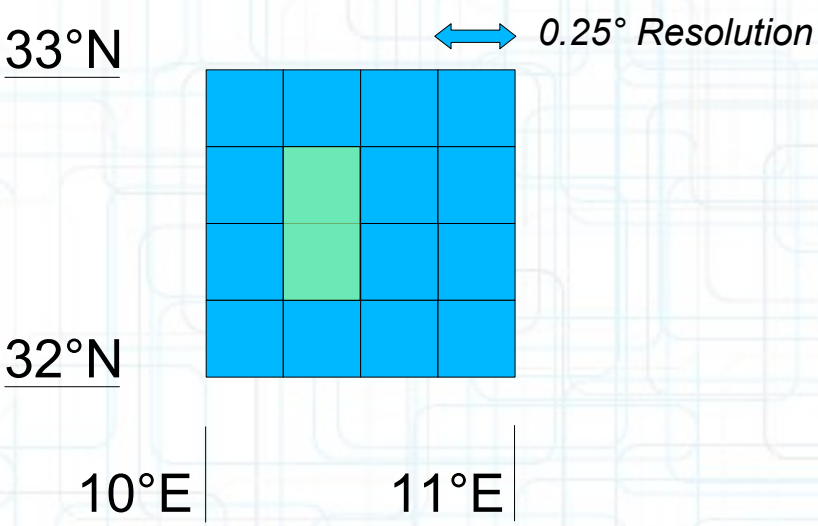
Subset

(32.25°N, 10.25°E) – (32.75°N, 10.50°E)

span = 0
(values represent theoretically infinitely small points)



span = 1
(values represent e.g. an average over an area resolution x resolution)




```
axisLabel: Lat
lowerBound: 32.25
upperBound: 32.75
uomLabel: deg
resolution: 0.25
span: 0.0
```

```
axisLabel: Lon
lowerBound: 10.25
upperBound: 10.50
uomLabel: deg
resolution: 0.25
span: 0.0
```

```
axisLabel: j
lowerBound: 0
upperBound: 2
```

```
axisLabel: i
lowerBound: 0
upperBound: 1
```

```
axisLabel: Lat
lowerBound: 32.25
upperBound: 32.75
uomLabel: deg
resolution: 0.25
span: 1.0
```

```
axisLabel: Lon
lowerBound: 10.25
upperBound: 10.50
uomLabel: deg
resolution: 0.25
span: 1.0
```

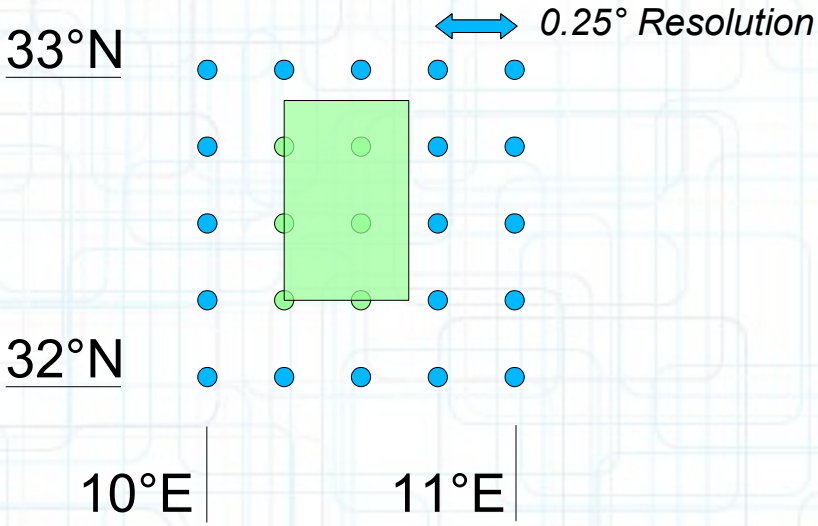
```
axisLabel: j
lowerBound: 0
upperBound: 1
```

```
axisLabel: i
lowerBound: 0
upperBound: 0
```

Subset (not aligned to grid)

(32.25°N, 10.25°E) – (32.85°N, 10.60°E)

span = 0
(values represent theoretically infinitely small points)



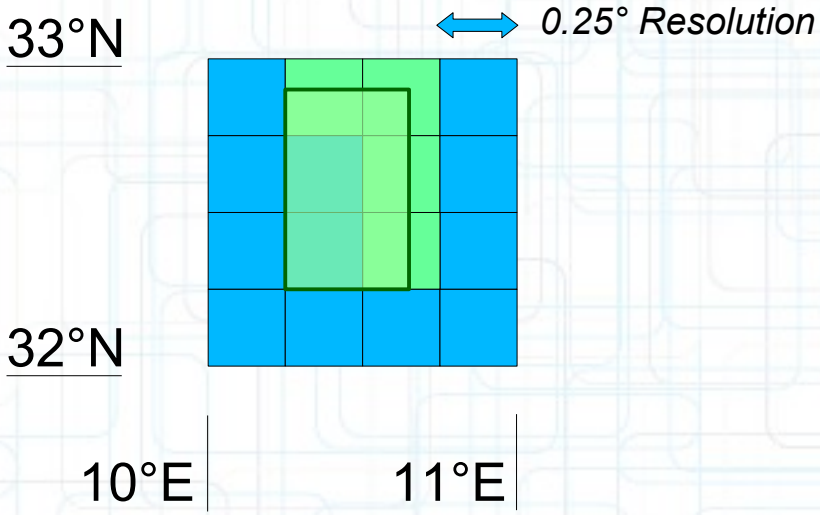

```
axisLabel: Lat
lowerBound: 32.25
upperBound: 32.75
uomLabel: deg
resolution: 0.25
span: 0.0
```

```
axisLabel: Lon
lowerBound: 10.25
upperBound: 10.50
uomLabel: deg
resolution: 0.25
span: 0.0
```

```
axisLabel: j
lowerBound: 0
upperBound: 2
```

```
axisLabel: i
lowerBound: 0
upperBound: 1
```

span = 1
(values represent e.g. an average over an area resolution x resolution)




```
axisLabel: Lat
lowerBound: 32.25
upperBound: 33.00
uomLabel: deg
resolution: 0.25
span: 1.0
```

```
axisLabel: Lon
lowerBound: 10.25
upperBound: 10.75
uomLabel: deg
resolution: 0.25
span: 1.0
```

```
axisLabel: j
lowerBound: 0
upperBound: 2
```

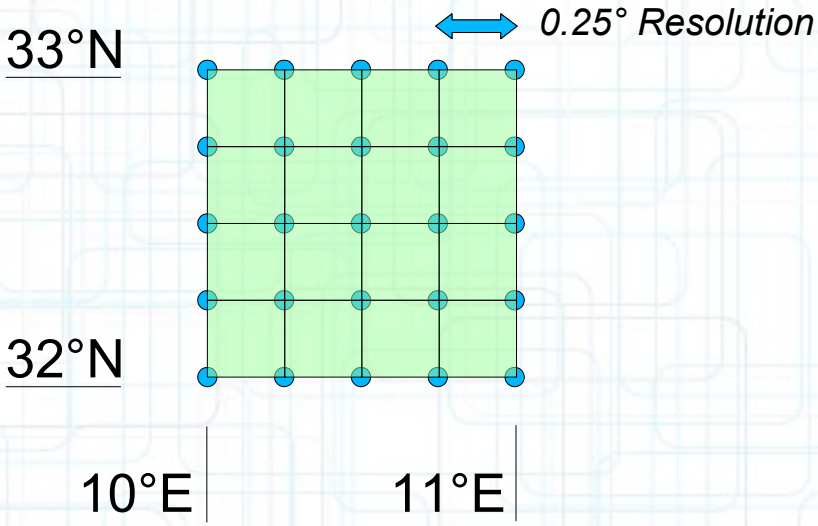
```
axisLabel: i
lowerBound: 0
upperBound: 1
```

Rendering Maps (area pixels)

(32°N, 10°E) – (33°N, 11°E)

span = 0

We need all edge values to interpolate area pixels

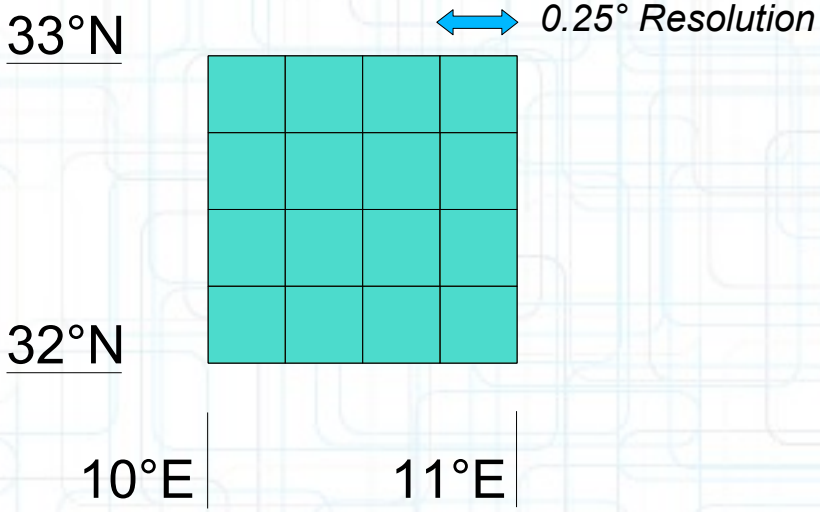


4 x 4 pixels map
5 x 5 source cells

(32°N, 10°E) – (33°N, 11°E)

span = 1

The values already represent pixel areas – no interpolation required



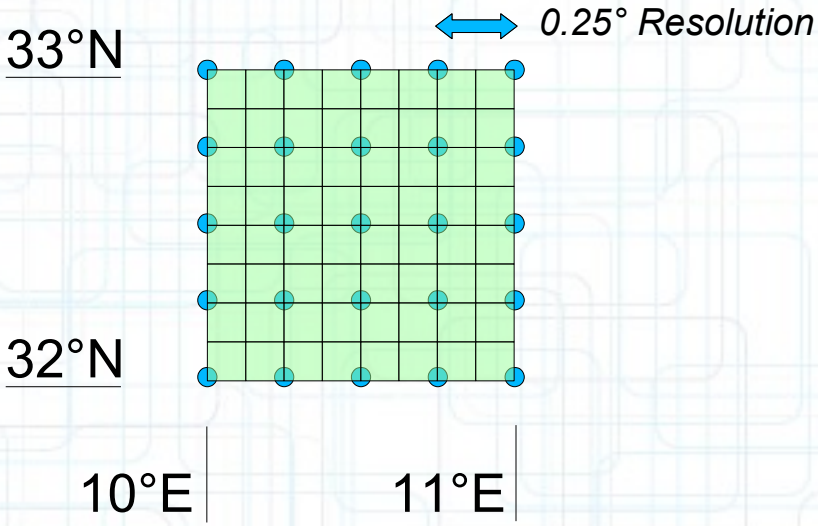
4 x 4 pixels map
4 x 4 source cells

Rendering Maps (larger map)

(32°N, 10°E) – (33°N, 11°E)

span = 0

We need all edge values to interpolate area pixels

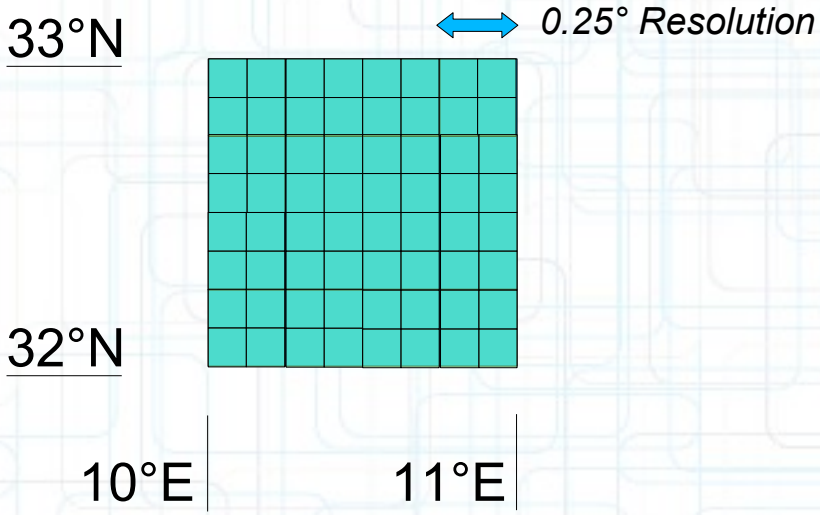


8 x 8 pixels map
5 x 5 source cells

(32°N, 10°E) – (33°N, 11°E)

span = 1

The values already represent pixel areas – different interpolation method



8 x 8 pixels map
4 x 4 source cells

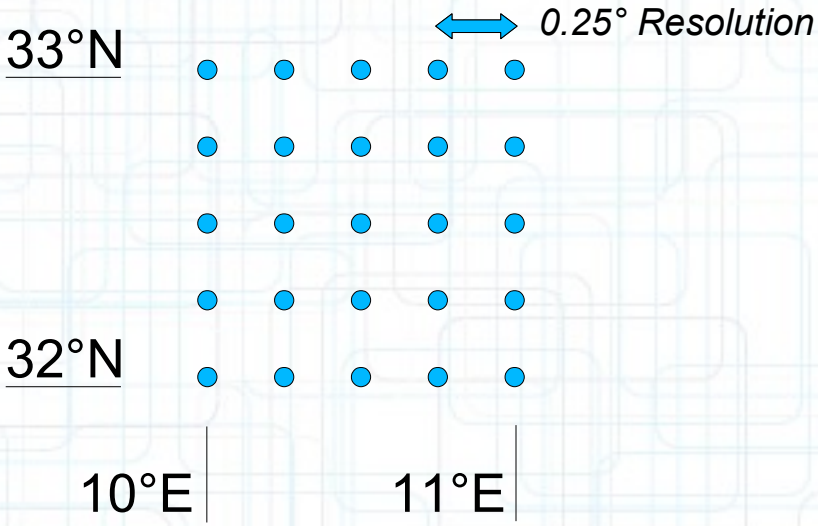
Coverage Tiles

(32°N, 10°E) – (33°N, 11°E)

span = 0

Use tile extent subset

Use TMS tile pixel size + 1



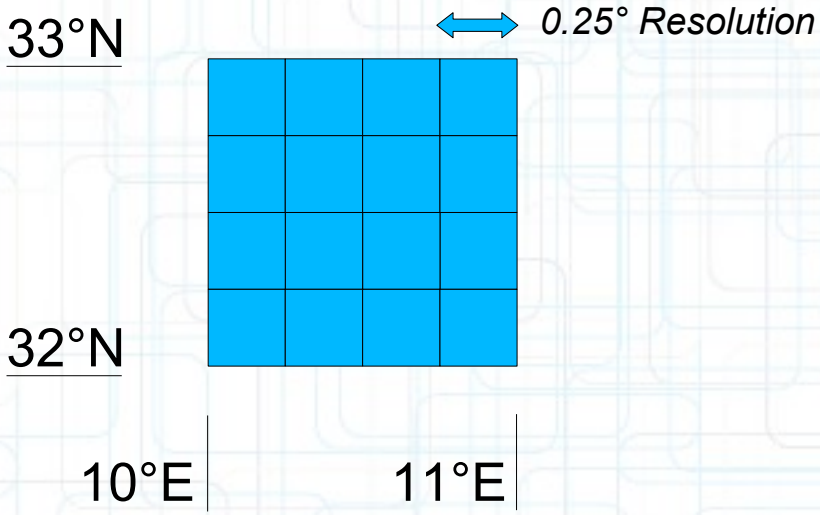
5 x 5 cells

(32°N, 10°E) – (33°N, 11°E)

span = 1

Use tile extent subset

Use TMS tile pixel size



4 x 4 cells

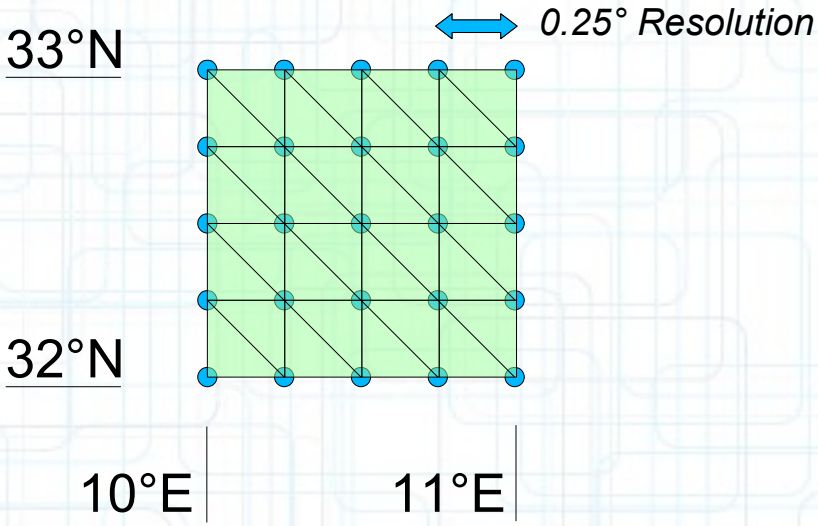
(4 x 4 TileMatrixSet tile pixel size)

3D Terrain Mesh

(32°N, 10°E) – (33°N, 11°E)

span = 0

Elevation values used to generate triangle mesh

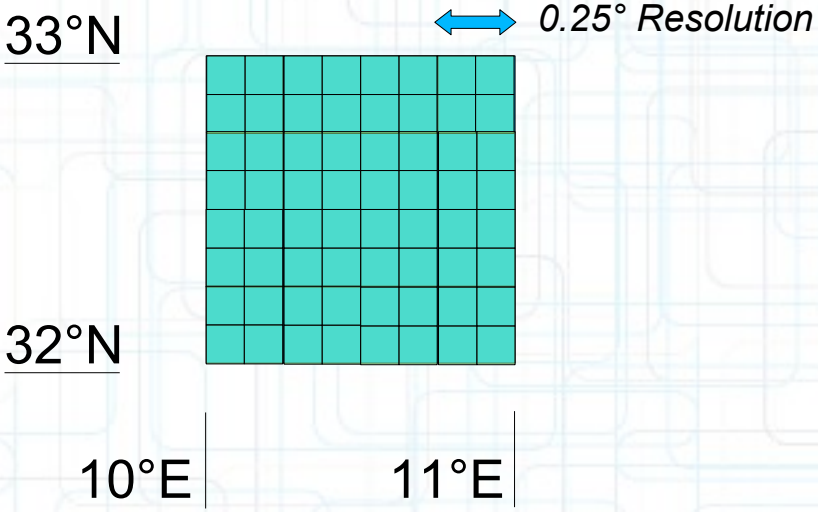


5 x 5 source cells

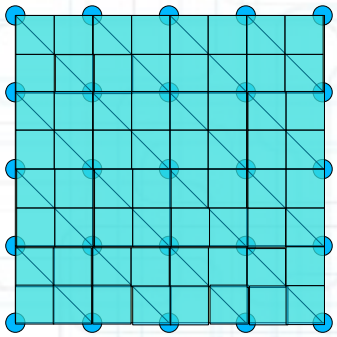
(32°N, 10°E) – (33°N, 11°E)

span = 1

Draped imagery and / or smooth hill shaded map



8 x 8 pixels hill shaded map
5 x 5 source elevation cells
8 x 8 imagery cells



Thank you!

Discussion context

https://github.com/opengeospatial/coverage-implementation-schema/issues/6

https://github.com/opengeospatial/ogcapi-coverages/issues/92

jerome@ecere.com

Observation and questions on *DomainSet*:

```
span =   (CRSAxis::upperBound - CRSAxis::lowerBound)
       / resolution
       - (IndexAxis::upperBound - IndexAxis::lowerBound)
```

Are such grid indices compatible with CIS 1.1 for both *span = 0* and *span = 1*?
Can the span be reliably inferred this way from current CIS 1.1 *DomainSet*?