Utilizing a Discrete Global Grid System For Handling Point Clouds With Varying Location, Time, and Scale

MSc Thesis Project, Fugro

Supervisors:

EDWARD VERBREE
PETER VAN OOSTEROM

Neeraj Sirdeshmukh

Delft University Of Technology

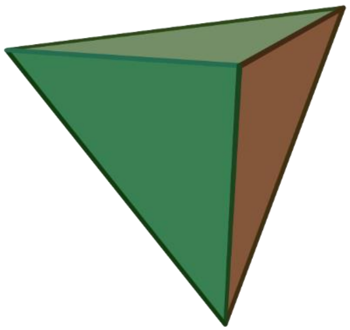
About me

Currently pursuing Master of Geomatics at Delft University Of Technology, Netherlands

Previous Experience/Education:

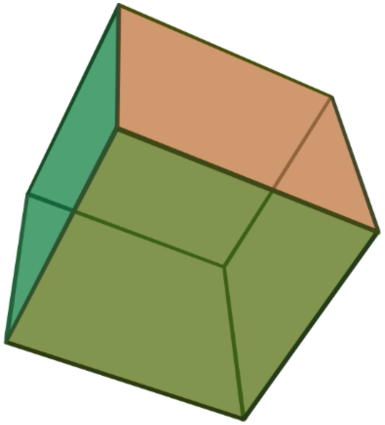
- 3 years work experience in GIS industry
- Master of Science in Geospatial Information Sciences, University of Texas at Dallas, USA
- Bachelor of Science in Geospatial Information Sciences, University of Texas at Dallas, USA

PLATONIC polyhedrons

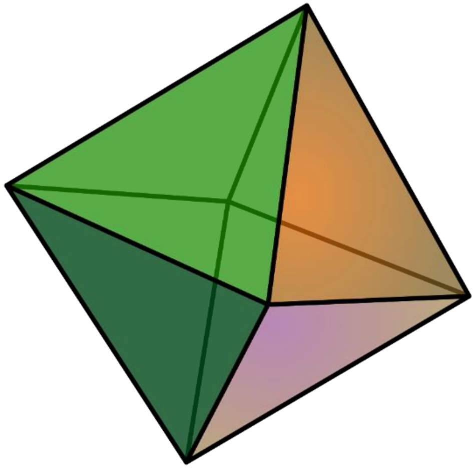


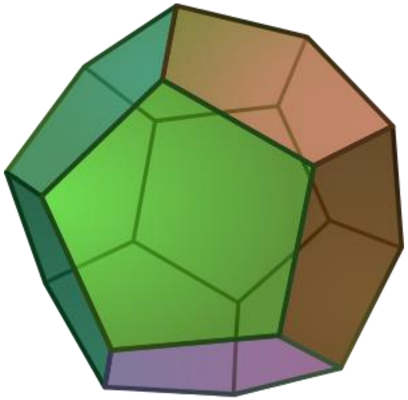
Tetrahedron

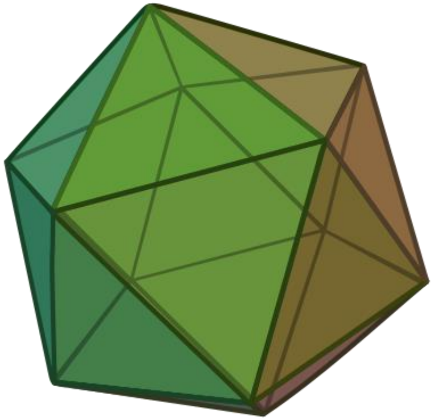
Cube



Octahedron



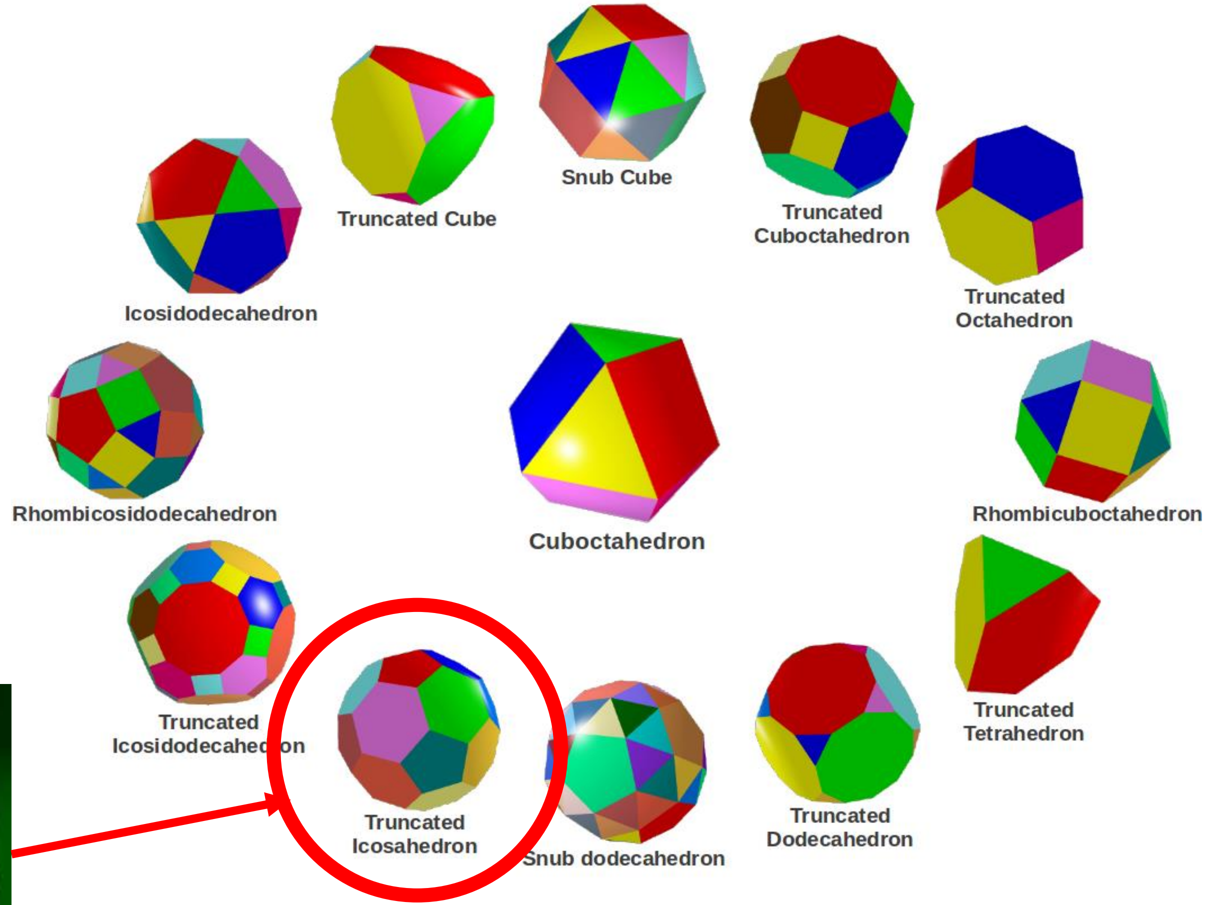




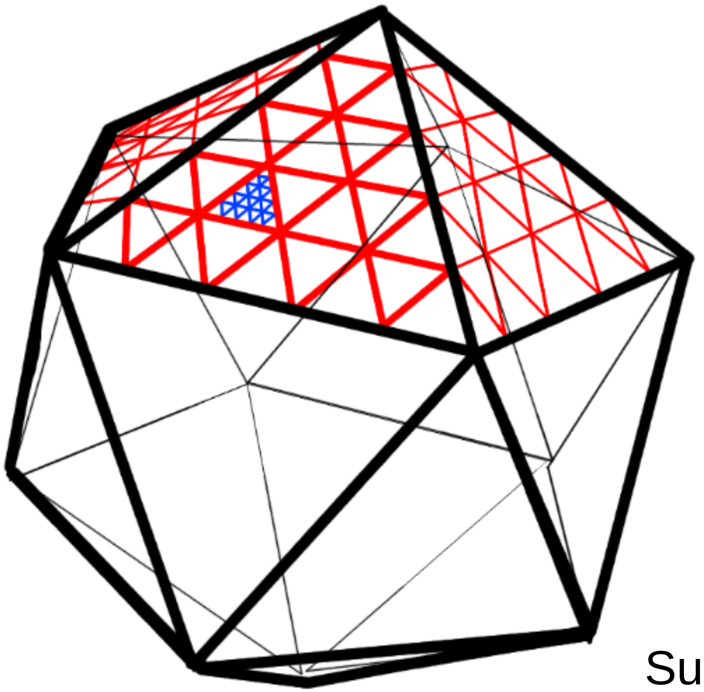
Dodecahedron

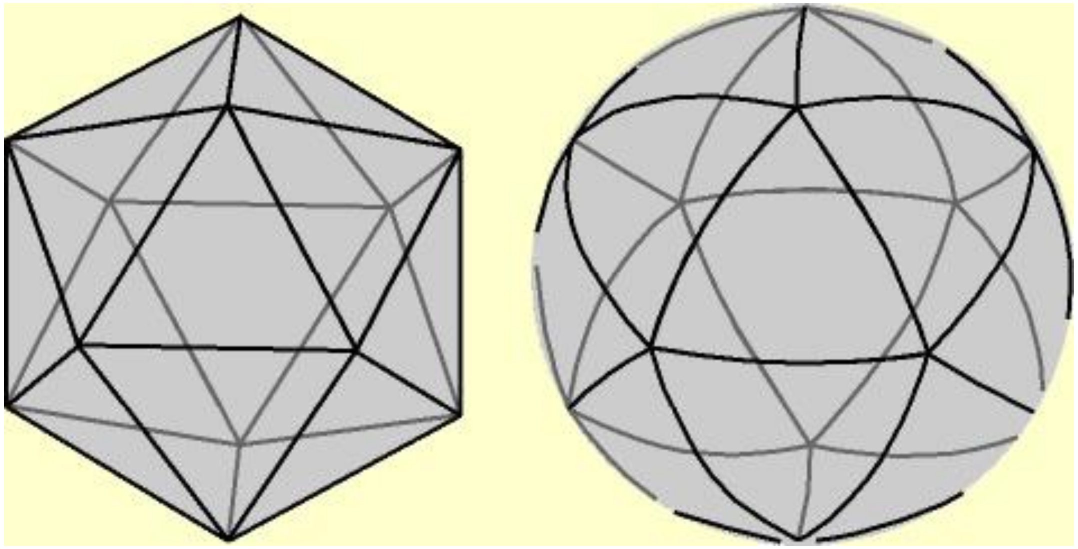
Icosahedron

ARCHIMEDEAN polyhedrons



…a novel representation of the Earth





Icosahedron

…..projected onto sphere with equal-area projection === equal-area cells!

Subdividing the polyhedron faces – equal-area cells!

Discrete Global Grid Systems (DGGS)

DGGS =

Polyhedron +
Polyhedron Orientation +
Subdivision shape +
Refinement ratio (aperture) +
[Projection]

IMPORTANT: Equal-area cells!

CHOICE OF DGGS

- Various kinds of DGGS exist, differing based on the choice of base polyhedron, polyhedron orientation, spatial partitioning method, aperture, and projection
- Icosahedron is a better approximation of the Earth than a tetrahedron, octahedron, cube, or dodecahedron
- Faces of the polyhedron can be subdivided into triangles, squares, rhombuses, or hexagons, and these can then be inversely projected back onto the Earth sphere

PROJECT INTRO

- Need to integrate point clouds originating from different **locations** (countries), each having different coordinate reference systems (CRS), into a common CRS
- Need to integrate point clouds with varying initial densities to achieve appealing visualization of point cloud data when viewed from any **scale**
- Need to determine how **time**-stamped point clouds acquired at different times can be analyzed for changes

Can DGGS be used to work with point clouds stemming from different locations, scales, and times?

Main research question:

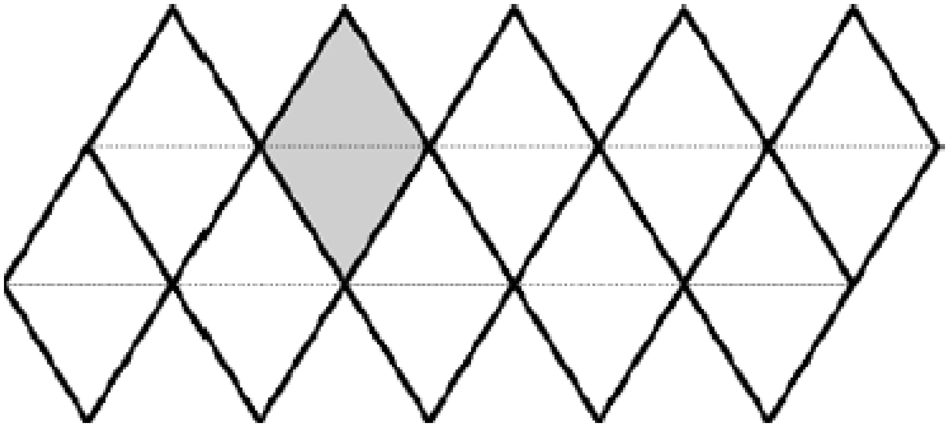
To what extent can a Discrete Global Grid System (DGGS) be used to handle point clouds with varying location, time, and scale ?

Sub - questions:

How can a variable-scale structure be implemented to support smooth-zoom in DGGS?

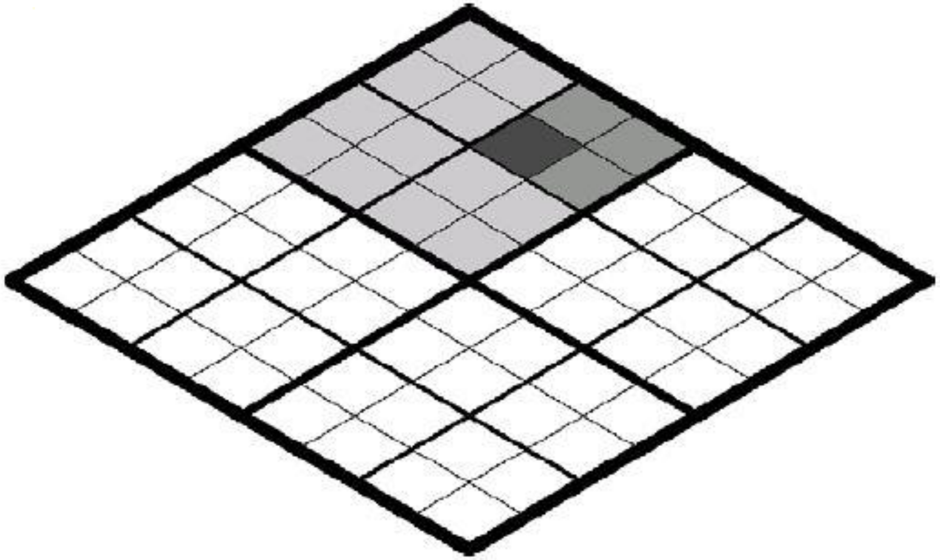
How can a DGGS be utilized with point cloud data to allow for a time-dynamic visualization of point cloud data of the same place acquired at different times?

What are the advantages and disadvantages of using a DGGS as compared with conventional coordinate reference systems?

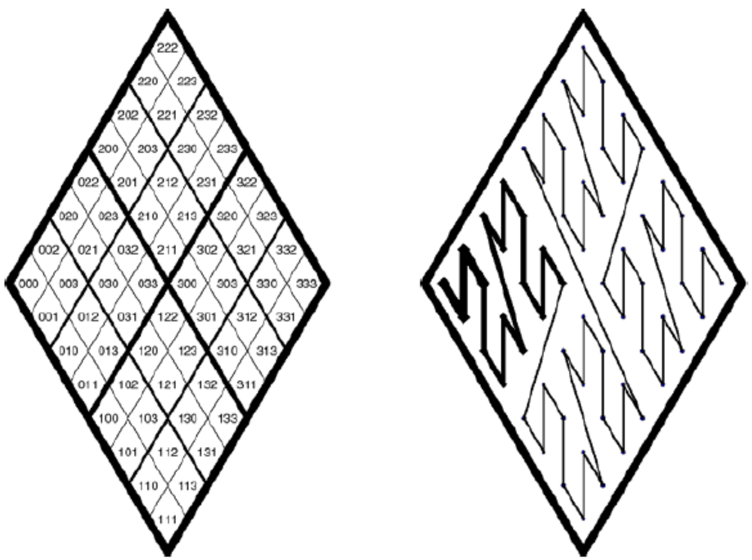


Flattened icosahedron composed of rhombus cells

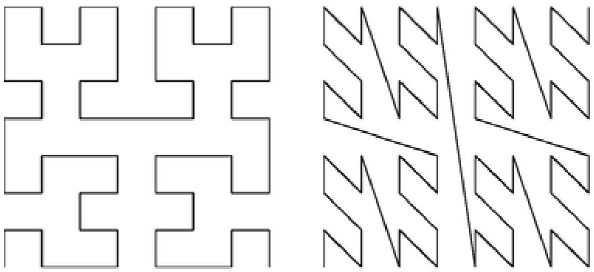
An aperture 4 rhombus nested hierarchy



- Morton indexing can be applied on a rhombus-based DGGS to ensure that each cell has a unique ID

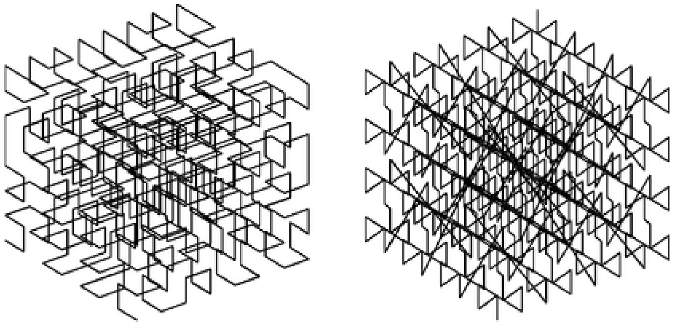


Quadrant-recursive Morton ordering on a rhombus cell



(a) 2D

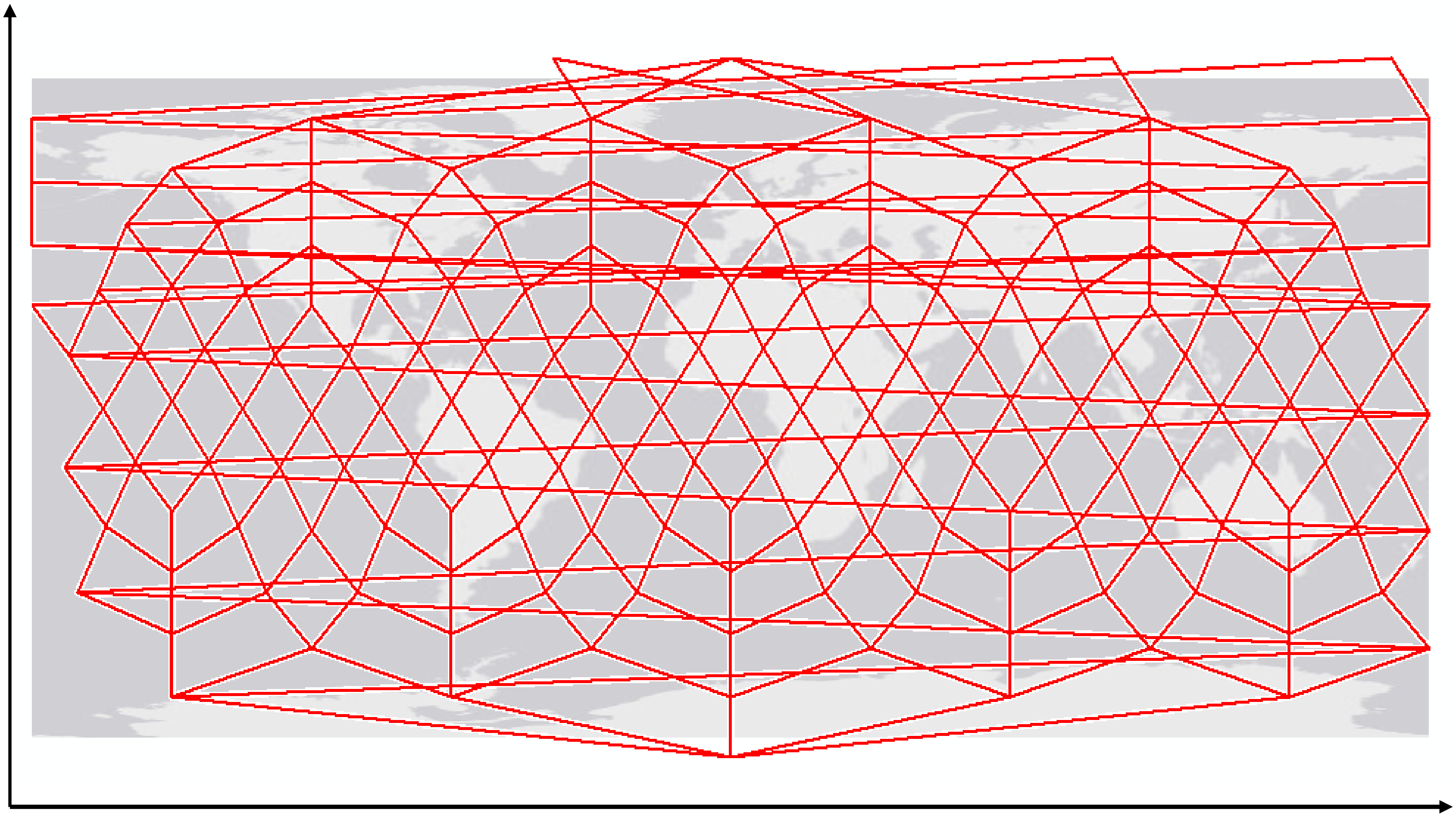
Hilbert and Morton orderings in 2D (above) and 3D (below)



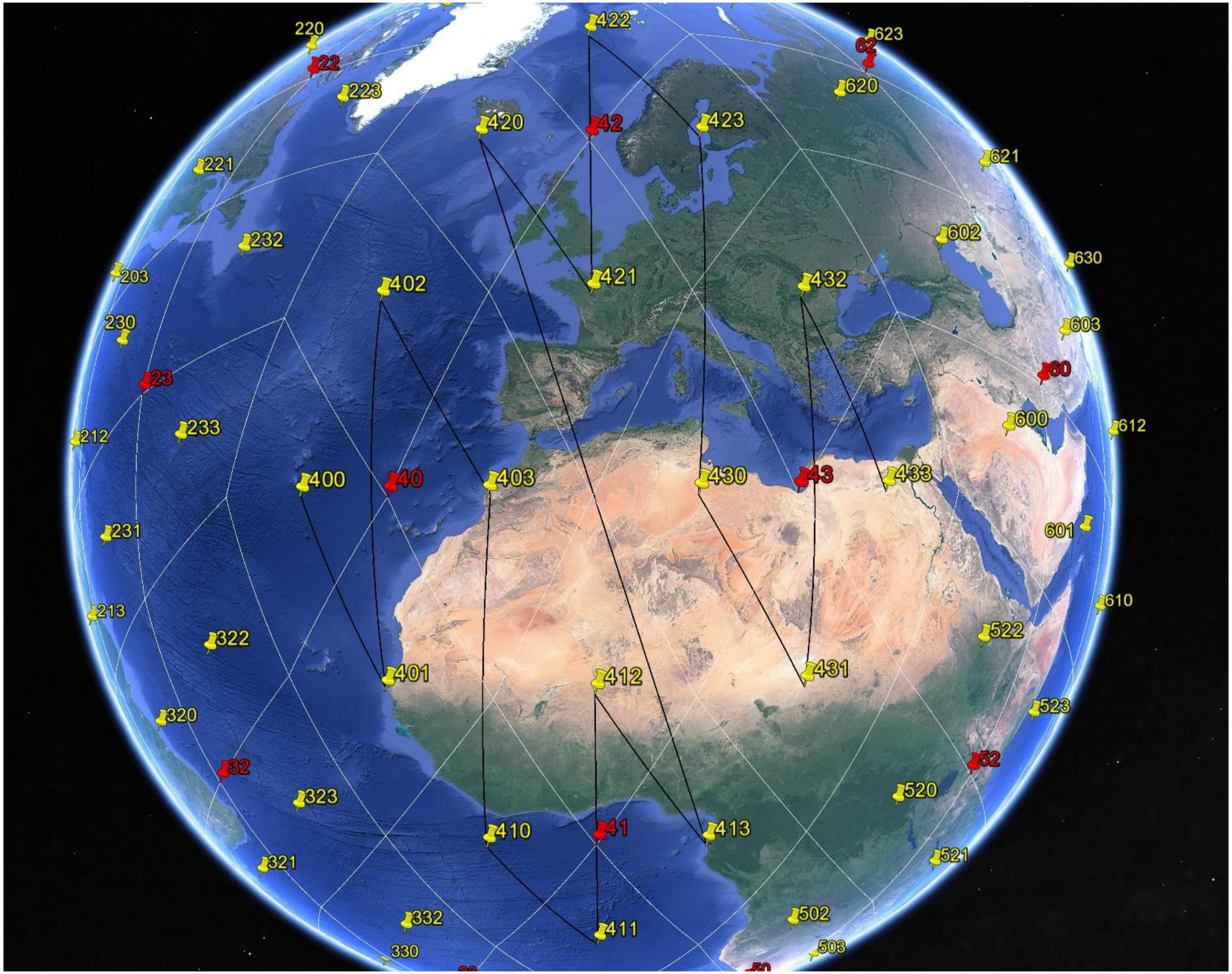
(b) 3D

Morton indexing using Plate Carrée projection

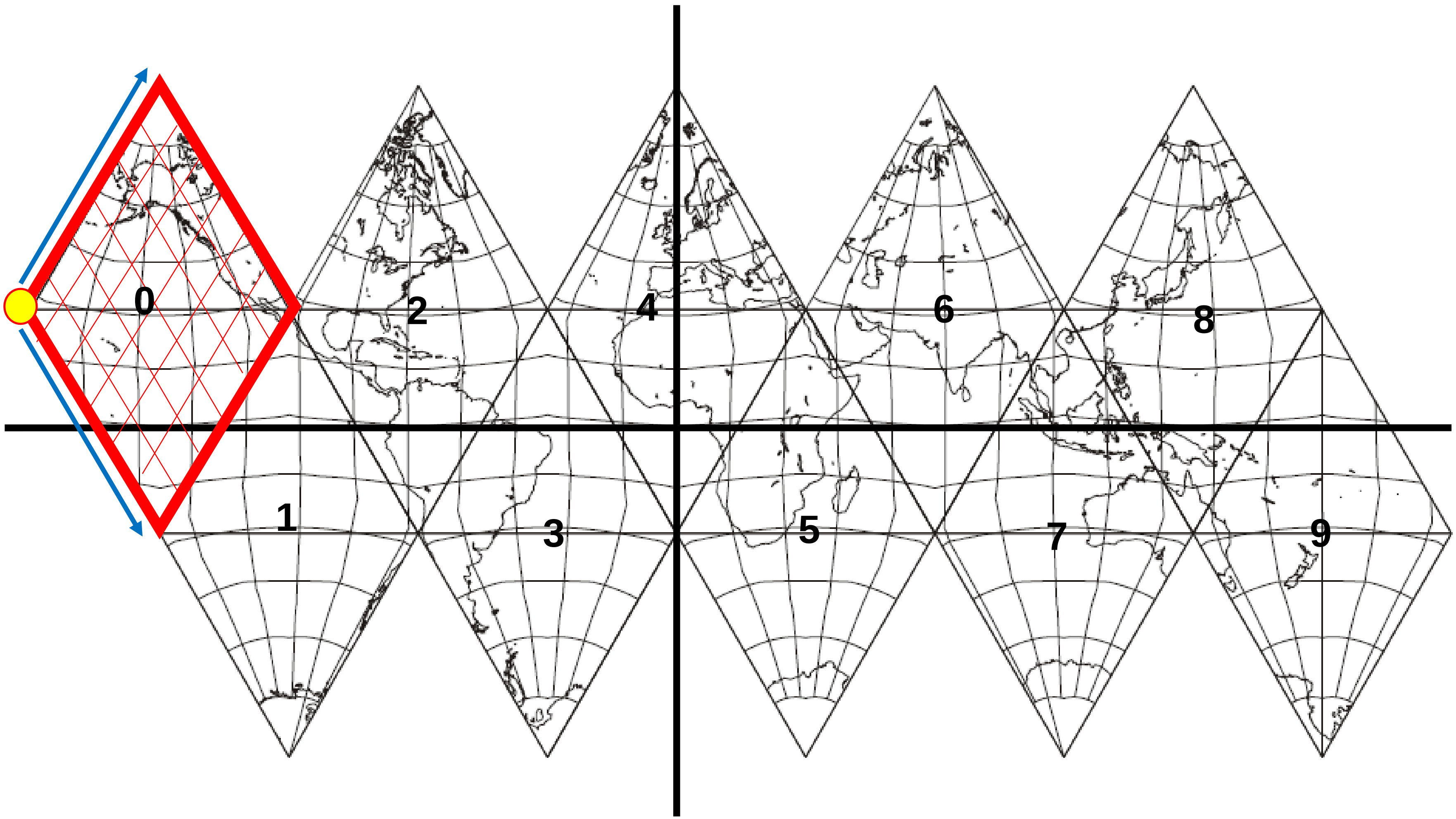
Bitwise interleaving (Long, Lat) coordinates does not work with *equal-area* DGGS!



Morton indexing using bearings on sphere

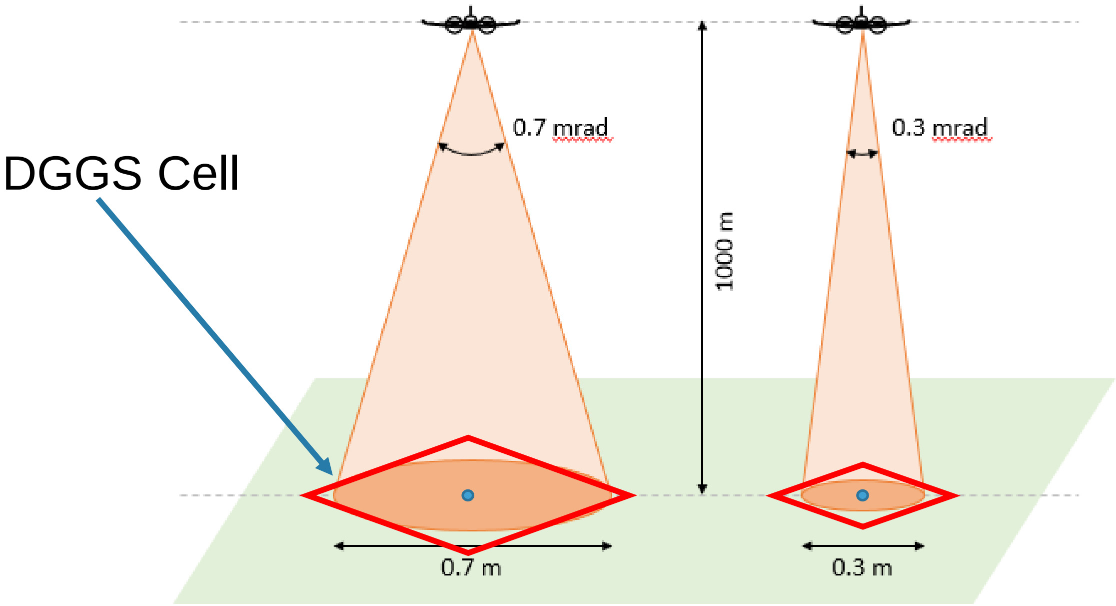


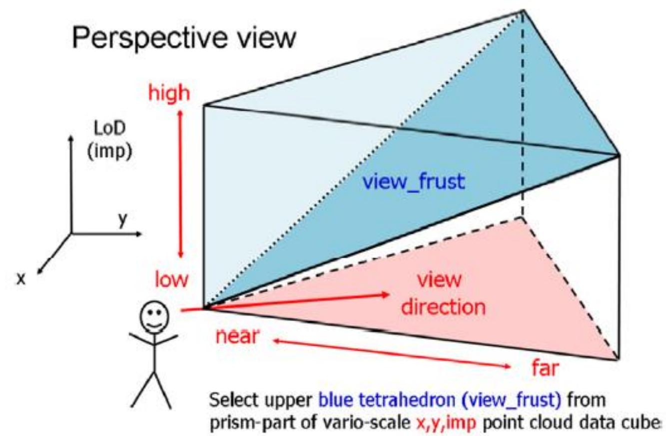
Icosahedral Snyder Equal Area (ISEA) Projection



Variable-scale DGGS

"Importance" value assigned to points based upon their precision --- relationship with LIDAR beam footprint



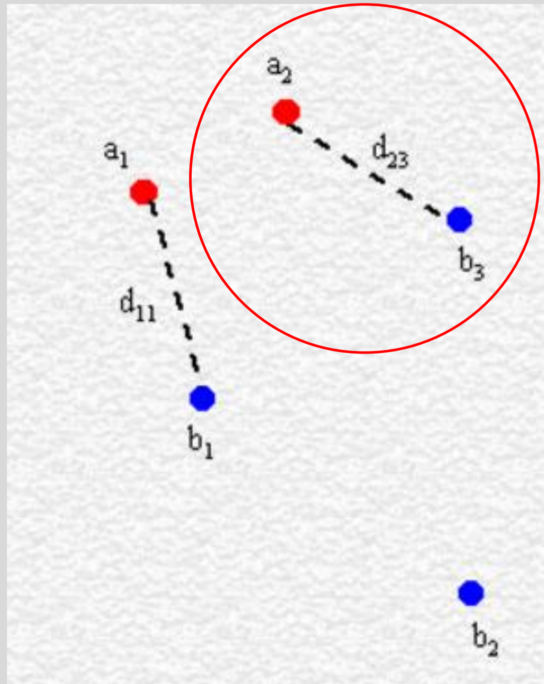


Length of Morton code will indicate precision and resolution of point, so no need to store these separately

DGGS FOR time-varying point clouds

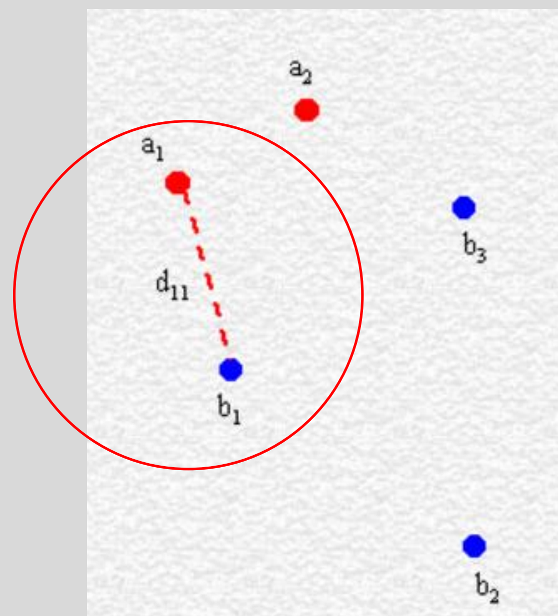
- The Hausdorff Distance between two points has been one of the most widely used methods of change detection in LIDAR data, and this will be investigated.
- Relies on a common octree structure between two different point clouds
- Compute for each point p in the 1st point cloud its distance to the nearest point in the other cloud
- Each 2D DGGS cell in one resolution layer can be assigned the mean Hausdorff Distance of all of its contained points and the cells color-coded based on these values.
- Points animated with CZML in Cesium JS

DGGS FOR time-varying point clouds



True (shortest) distance

Hausdorff distance



END PRODUCT

- DGGS-based web viewer for point clouds
    - Should provide appealing visualization of point clouds
    - Should have a “globe”-like interface, not a flat 2D representation
    - Should allow for time-dynamic visualization of changes between point clouds using DGGS
    - Should allow for smooth zoom in/out
    - Should allow for upload/download of point cloud data
    - Should allow for simple cell-based spatial queries using SQL

Example SQL Queries (will use Morton codes in back-end)

" Select all points in area with a certain precision"

" Find maximum elevation of points in a given DGGS cell"

" Retrieve all points in a lat/long bounding box"

"Select all points at certain precision level that were captured between time t1 and time t2"

"Find mean elevation of points in cell that were captured at time t1"

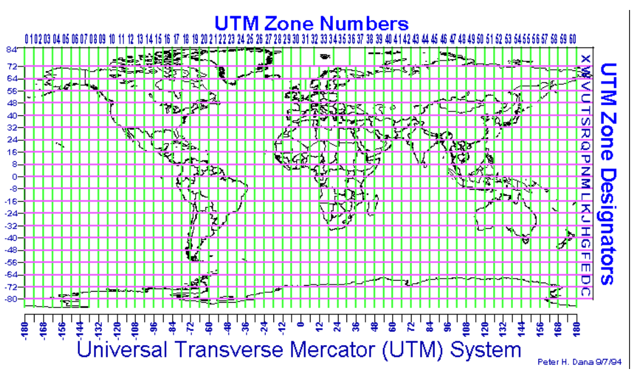
Comparison of DGGS with conventional CRS’s

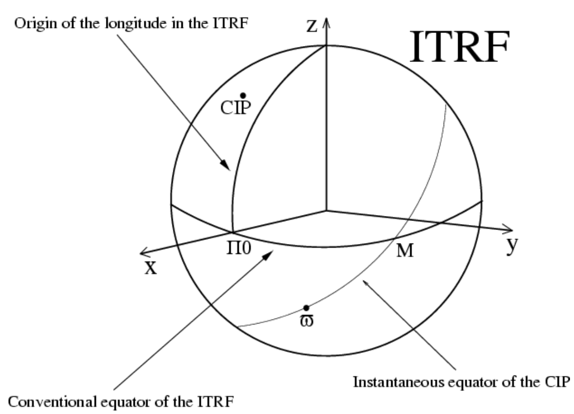
DGGS is meant to be used as an <u>information grid</u>, and not a navigation grid like conventional CRS’s.

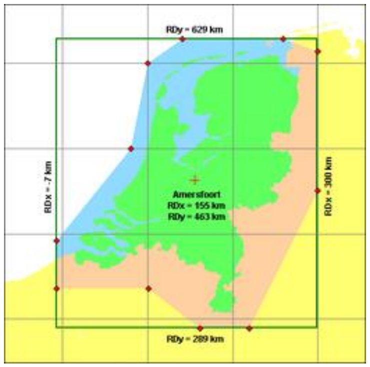
- Aim is also to investigate the advantages and disadvantages of using a DGGS as opposed to some commonly used national/international projections/coordinate reference systems such as:

  - Plate Carrée (Latitude/Longitude)
  - Universal Transverse Mercator (UTM)
  - International Terrestrial Reference System (ITRS)
  - European Terrestrial Reference System (ETRS)
  - Dutch *Rijksdriehoeksstelsel* system

  - Each of these systems has its own pros and cons, including DGGS!







Thanks for your attention!

EMAIL: n.sirdeshmukh@fugro.com

LINKEDIN: https://www.linkedin.com/in/neeraj-sirdeshmukh-37a1b353/