Riskaware

capability through technology

OpenEAGGR

Literature Review & Prototype Evaluation

Issue 1

16th June 2017

Riskaware Ltd
Colston Tower
Colston Street
Bristol
BS1 4XE
+44 117 929 1058
www.riskaware.co.uk

Author: Ian Bush
ian.bush@riskaware.co.uk
+44 117 933 0539

For the Record

Table of Contents

1 **Introduction** ... 6

2 **DGGS Overview** ... 6

2.1 What is a DGGS? ... 6

2.2 Design Choices ... 6

3 **Requirements** ... 7

4 **Base Polyhedron** ... 8

4.1 Number of Faces ... 8

4.2 Polyhedron Options ... 8

5 **Orientation** ... 10

5.1 Orientation based on Application ... 10

5.2 Specifying Orientation ... 10

5.3 Examples of Different Orientations ... 10

6 **Projection** ... 12

6.1 Lambert Azimuthal Equal-Area Projection ... 12

6.2 Equal-Area Projection on a Polyhedron ... 12

7 **Grid Partitioning** ... 13

7.1 Terminology ... 13

7.2 Quaternary Triangular Mesh ... 13

7.3 Hexagonal Grid ... 14

7.4 Isosceles Triangle Grid ... 14

8 **Cell Indexing** ... 16

8.1 Axial Coordinates ... 16

8.2 Offset Coordinates ... 16

8.3 Hierarchical Index ... 17

8.4 Parent-Child Relationships ... 17

8.5 Working with Hexagonal Grids ... 18

9 **Existing Solutions** ... 19

9.1 HEALPix ... 19

9.2 SCENZ-Grid ... 19

9.3 PYXIS WorldView™ ... 19

9.4 DGGRID ... 19

9.5 Proj.4 ... 20

9.6 Other Solutions ... 20

10 **Prototyping** ... 21

11 **Evaluation Criteria** 23
11.1 Equal Area 23
11.2 Hierarchical Consistency 23
11.3 Conversion Speed 24
11.4 Conversion Accuracy 24
11.5 Ease of Human Visualisation 24
12 **Performance Testing** 26
12.1 Input Points 26
12.2 Speed Testing 26
12.3 Grid Partitioning Accuracy 26
13 **Prototype Evaluation** 27
13.1 Evaluation Framework 27
13.1.1 Prototype Limitations 27
13.2 Evaluation System 27
13.3 Conversion Speed 27
13.4 Conversion Error 28
13.5 Accuracy Error 30
13.6 Parent-Child Querying 31
13.6.1 Parent Querying 31
13.6.2 Child Querying 32
14 **Conclusions** 34
15 **Recommendations** 35
16 **Bibliography** 36

List of Figures and Tables

Figure 1 - The five Platonic solids [3] 8
Figure 2 - Truncated Icosahedron [5] 8
Figure 3 - Distortion on Polyhedral Globes using Snyder's Equal-Area Projection [4] 9
Figure 4 – Different orientations for an icosahedron base polyhedron [6] 10
Figure 5 - Lambert azimuthal equal-area projection of the world [7] 12
Figure 6 - The Earth projected on to a base polyhedron (octahedron) [8] 12
Figure 7 - Quaternary Triangular Mesh sub-division 13
Figure 8 – Aperture 3 partitioning with a hexagonal grid 14
Figure 9 - Partitioning in an Isosceles Triangle Grid 15
Figure 10 - Axial coordinates on a Hexagon Grid [11] 16
Figure 11 – Offset coordinates on a Hexagon Grid [11] 17
Figure 12 - Hierarchical Indexing on a QTM 17
Figure 13 – Parent-Child relationships between hexagons 18
Figure 14 - The first 4 resolution grids of the HEALPix system [14] 19
Figure 15 - Google Earth screenshot with KML grids produced by DGGRID 20

Figure 16 – Design choices for the DGGS prototypes 21
Figure 17 - Percentage of Earth's surface covered by non-equal area cells in an aperture 3 hexagonal grid [16]....23
Figure 18 - Representing a heart shape on hexagonal and triangle grids 25
Figure 19 – Evaluation system properties....27
Figure 20 – DGGS conversion performance....28
Figure 21 – Average DGGS conversion error....29
Figure 22 – Average conversion error as a percentage of the original error 29
Figure 23 – Average percentage conversion accuracy error....30
Figure 24 – Equilateral triangle (black), hexagon (red) and isosceles triangle (blue) each representing the same area 30
Figure 25 – Accuracy percentage error....31
Figure 26 – Average accuracy percentage error 31
Figure 27 – Performance of parent queries....32
Figure 28 – Performance of child queries....33

1 Introduction

This document details the output of the literature review task for the Open Equal Area Global GRid (OpenEAGGR) software. A wide-ranging study was conducted to investigate current DGGS approaches in order to provide insight into the techniques that will be evaluated in the prototyping phase.

The document provides an introduction into DGGS concepts and terminology. It goes on to outline the areas where design decisions are required and demonstrates the advantages and disadvantages of each. Based on the findings it suggests the techniques we believe are most appropriate for taking forward to the prototype evaluation phase.

2 DGGS Overview

2.1 What is a DGGS?

A Discrete Global Grid (DGG) is a set of cells that partition the Earth's surface. Traditionally locations on maps are defined by points – an exact location with no associated area. In a DGG, each cell defines an area rather than an exact point, which gives two distinct advantages:

1. A wider variety of data can be stored e.g. populations, rainfall, Twitter usage.
2. No point can be infinitely accurate, so there will always be an accuracy associated with each point. In a DGG this accuracy is implied by the area of the cell.

A Discrete Global Grid System (DGGS)[1] is made of up of a series of discrete global grids, usually consisting of increasingly finer resolution grids. Although DGGSs potentially have many different applications, they have come to be associated with polyhedral maps.

A polyhedral map attempts to solve the distortion problems of conventional 2-D maps by approximating the shape of the Earth as a polyhedron. By being closer to the shape of the Earth, the stretching and compressing required to fit the spherical Earth on to a flat map is greatly reduced, thus providing a more accurate map.

Once projected on to the flat side of the polyhedron the Earth can be partitioned using a DGGS. [1]

2.2 Design Choices

A polyhedral map DGGS can be fully described by specifying five design choices: [1]

1. Polyhedron to project the Earth on to (base polyhedron).
2. Orientation of the polyhedron relative to the Earth.
3. Projection method for mapping the Earth on to the polyhedron.
4. Spatial partitioning method for creating cells on the faces on the polyhedron.
5. Method of referencing the location of the cells.

As part of the literature review different options for each of these choices have been considered.

[1] Although a DGGS is made of a series of DGGs, the two terms are often used interchangeably.

3 Requirements

Whilst considering the possible options for a DGGS design, it is useful to bear in mind the specific requirements for this project. The important criteria for the DGGS are as follows:

- **Equal Area Projection** – A DGGS cell must represent exactly the same area anywhere on the surface of the Earth.
- **Multi-Resolution** – Be able to handle points on the Earth with different associated areas.
- **Equal Area Cells** – All the cells on the grid at a particular resolution must be the same area.
- **Parent-Child Queries** – Optimised for finding neighbouring cells and corresponding cells in the next lowest resolution.
- **Fast Conversion** – Points on the Earth must be converted to and from the DGGS as fast as possible.
- **Open Source** – Suitable for release under an Open Source License.
- **Limited Angular and Distance Distortion** – It is not possible to have an equal area projection DGGS which does not distort angles and distances to some extent, but any reduction in distortion is preferred.

4 Base Polyhedron

The first step in designing a DGGS is to decide on the base polyhedron.

4.1 Number of Faces

When choosing which polyhedron to use, a compromise must be made. Choosing a polyhedron that more closely resembles the shape of the Earth means that the Earth will be less distorted when it is projected on to it. However in order for the polyhedron to be more "spherical" usually requires the polyhedron to have a greater number of faces.

A polyhedron with more faces requires more computation to calculate which face a point on the Earth should be projected on to since the process of assigning a point to a face is iterative. There is also a problem with increased discontinuities, which can be illustrated by considering a line across the surface of the Earth. When the line is projected on to the polyhedron it will change in direction as it passes across the angle between two faces. The more faces a polyhedron has, the more edges it has and therefore there is a higher chance of encountering one of these inconsistencies. Although conversely the angle between the faces is generally reduced by having more faces, so the amount the line distorted is likely to be less.

4.2 Polyhedron Options

The Platonic polyhedrons (i.e. polyhedrons where all the faces are identical regular polygons with identical angles at each vertex) are natural candidates for a DGGS projection because of their consistency and simplicity. The icosahedron has the greatest number of faces, with 20, and therefore is a common choice for a base Platonic solid. However it is worth noting that the dodecahedron, with only 12 faces, is actually the closest approximation to a sphere and therefore has lower distortion.

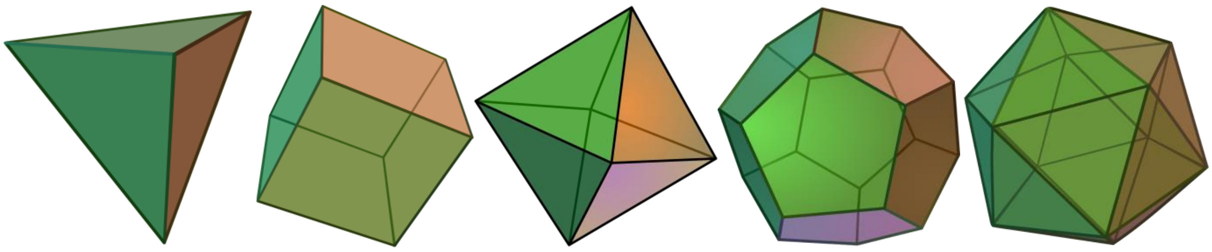

Figure 1 - The five Platonic solids [2]

In his paper "An Equal-Area Map Projection for Polyhedral Globes" [3], John P. Snyder identifies another choice of solid, the truncated icosahedron, which is an even closer approximation to a sphere than the icosahedron or dodecahedron.

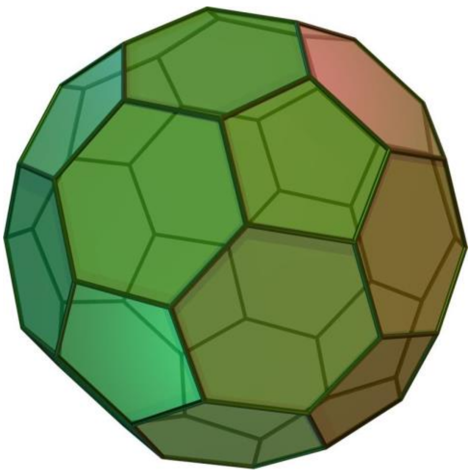

Figure 2 - Truncated Icosahedron [4]

In his paper, Snyder goes on to compare the worst-case angular and distance distortion for all five of Platonic solids and the truncated icosahedron when used as the base polyhedron for his projection. A copy of his findings is shown in Figure 3.

Polyhedron	Max Angular Deformation (in °)	Max Scale Factor	Min Scale Factor
Tetrahedron	52.07	1.601	0.624
Cube	25.17	1.248	0.801
Octahedron	34.45	1.357	0.737
Dodecahedron	10.24	1.094	0.914
Icosahedron	17.27	1.163	0.860
Truncated Icosahedron			
- **Hexagon Faces**	3.75	1.033	0.968
- **Pentagon Faces**	2.65	1.033	0.983

Figure 3 - Distortion on Polyhedral Globes using Snyder's Equal-Area Projection [3]

In the table, angular deformation is defined as the greatest difference in the bearing between two points on the Earth and the bearing between the same two points when they are projected on to the polyhedron. Scale factor is the relative change in the distance between those two points.

The results from Snyder’s findings can be used as a basis to determine the most promising base polyhedrons for a DGGS. Looking at the data, it is the dodecahedron which has the least angular and distance distortion of the five Platonic solids, with the truncated icosahedron having less still.

5 Orientation

After choosing the type of base polyhedron, a decision must then be made how it will be orientated relative to the actual surface of the Earth.

5.1 Orientation based on Application

Angular and distance distortion are not equal across the whole of the polyhedron. It is therefore better to orientate the base polyhedron so that the places of most distortion are furthest away from the areas of most interest. For example, this might mean orientating the polyhedron such that the greatest distortion will occur over water rather than on land.

To illustrate where the places of most distortion occur, consider placing the polyhedron around the Earth. Where the polyhedron touches the Earth's surface, in the centre of the faces, there is no distortion. As you move away from the centre of the face, the polygon gets further away from the Earth's surface causing angular and distance distortion to increase in order preserve equal-area. Distortion is greatest when the polyhedron is the furthest away from the Earth's surface, which is at its vertices.

Ideally then the base polyhedron should be orientated with its vertices away from the areas of most interest.

5.2 Specifying Orientation

The orientation of the base polyhedron can be specified by giving the coordinates of one of the polyhedron's vertices and the azimuth from that vertex to an adjacent vertex. For Platonic solids this information will completely specify the position of all the other vertices, but for the truncated icosahedron it must also be specified which vertex is being used as the reference because of the different face shapes. [5]

5.3 Examples of Different Orientations

Figure 4 shows examples of different orientations for an icosahedron when used as a base polyhedron for a DGGS:

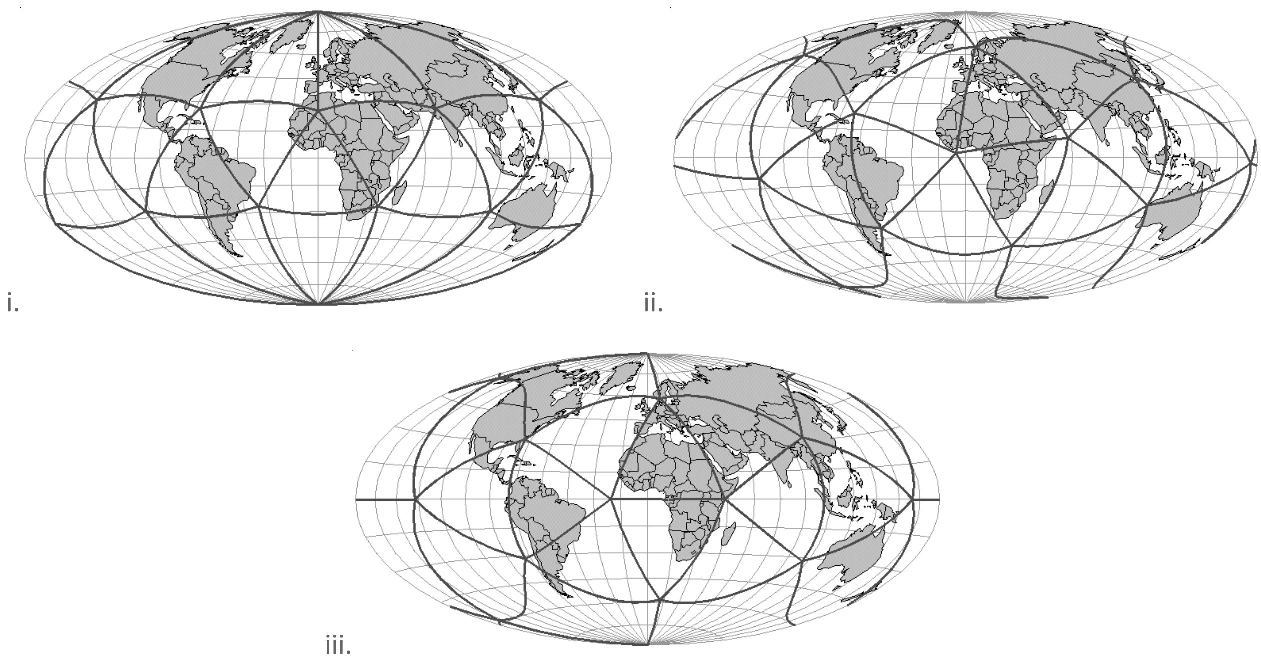

Figure 4 – Different orientations for an icosahedron base polyhedron [5]

i. Perhaps the most common, this orientation has a vertex at each of the poles and one of the edges emanating from the vertex at the North Pole aligned with the prime meridian. This orientation is good for human visualisation, because it uses points on the Earth that most people are familiar with – the poles and the prime meridian.
ii. Richard Buckminster Fuller's orientation for his Dymaxion icosahedral map projection. In the orientation he placed all 12 of the icosahedron vertices over water so that the net of the icosahedron could be unfolded without breaks in any land mass. This is the only known placement with this property.
iii. An orientation used for fluid flow simulations of global climate, where data points are moving around the Earth. The orientation makes the icosahedron symmetrical about the equator to avoid simulations that have a state symmetrical about the equator evolving into a state which is asymmetrical about the equator.

The different orientations demonstrate that no one orientation is suitable for all applications and that the choice must depend on the intended use of the DGGS. [5]

6 Projection

The requirements listed in Section 3 state that an equal-area projection is required, so a projection must be found that preserves the area between the Earth and the base polyhedron. For example when a country, such as the United Kingdom, is projected on to the polyhedron it may not be the same shape, but it should have exactly the same area.

6.1 Lambert Azimuthal Equal-Area Projection

It is possible to map the Earth with an equal area projection onto a single flat surface using the Lambert Azimuthal Equal-Area Projection, shown in Figure 5Figure 5 - Lambert azimuthal equal-area projection of the world . It is clear from Figure 5 that using this projection the features of the Earth would be heavily distorted in order to maintain equal-area.

Figure 5 - Lambert azimuthal equal-area projection of the world [6]

6.2 Equal-Area Projection on a Polyhedron

Irving Fisher was the first to successfully use the Lambert Azimuthal Equal-Area Projection to project the Earth on to an icosahedron. Snyder then generalised his method for all Platonic solids and the truncated icosahedron. By projecting on to a polyhedron rather a single surface, they were able to greatly reduce the angular and distance distortion.

Figure 6 - The Earth projected on to a base polyhedron (octahedron) [7]

Whilst other equal-area solutions have been proposed for specific solids, for example the HEALPix projection (discussed further in Section 9.1), Snyder’s projection is the only one that can be applied to a range of different polyhedrons. [7]

7 Grid Partitioning

Once a point has been projected on to the base polyhedron, its location and associated accuracy must be converted to a cell in the DGGS.

7.1 Terminology

Before discussing the possible methods for partitioning the faces of the base polyhedrons it is useful to explain some of the terminology associated with these grids:

- **Aperture** – Area of the cells in one resolution of the DGGS, relative to the area of the cells in the next highest resolution. For example, in an aperture 4 DGGS, the area of a cell at resolution *n*, will be four times that of a cell at resolution *n + 1*. Generally it is better for a DGGS to have a smaller aperture because it gives more potential grid resolutions, and therefore more options for different cell sizes.
- **Congruence** – A congruent DGGS is any DGGS where the cells of a higher resolution sub-divide those of lower resolution exactly, without any overlap. Congruent grids help to simplify cell location at different resolutions because there is a strong relationship between grids of different resolutions.
- **Parent Cell** – Cell in the next lowest resolution grid that contains the given cell.
- **Child Cells** – Cells in the next highest resolution grid that are contained by the given cell.

7.2 Quaternary Triangular Mesh

Three of the five Platonic solids have faces that are equilateral triangles, so they are well-suited to a grid system based on equilateral triangles. By using a grid based on the same shape as the faces, the complexities of having cells that overlap onto multiple faces can be avoided.

The most common grid based on equilateral triangles is the Quaternary Triangular Mesh (QTM) (also known as the Hierarchical Triangular Mesh), shown in Figure 7.

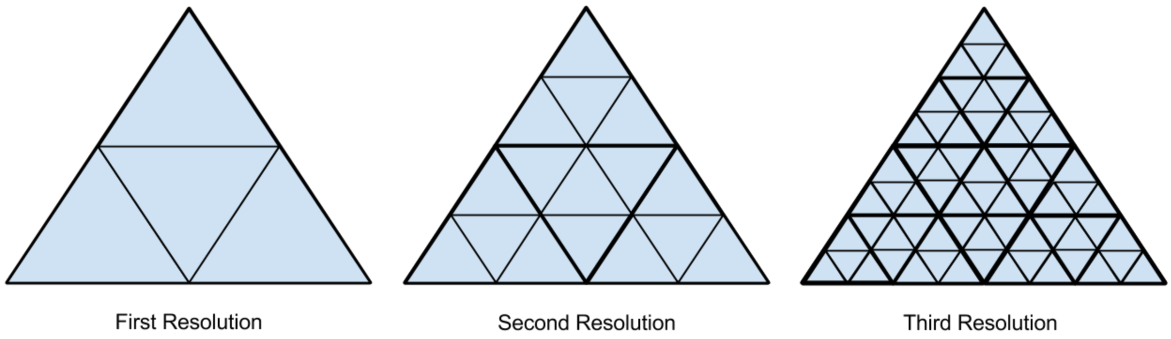

Figure 7 - Quaternary Triangular Mesh sub-division

The different resolutions of the QTM are created by dividing each of the triangular faces into four smaller equilateral triangles by connecting the midpoints of the sides of the triangle. The four triangles this generates can be subdivided again in the same way to form 16 triangles at the second resolution. The third resolution grid can be created by dividing all the triangles again to give 64 triangles, and so on. The QTM is an example of a congruent grid. [8] [9]

Applying the QTM to pentagonal faces of the dodecahedron and truncated icosahedron cannot be done in an equal-area DGGS, because it forms cells with different areas along the edges where the polyhedron's faces meet.

7.3 Hexagonal Grid

Like the QTM, it is difficult to apply hexagonal grids to any other base polyhedrons except those with equilateral triangle faces. Even on equilateral triangle faces hexagons cannot be made to cover the face without some hexagons spanning more than one face. Unlike pentagonal faces however the overlapping can be made to be consistent, forming regular shaped cells when combined with the overlapped cells on the adjacent faces.

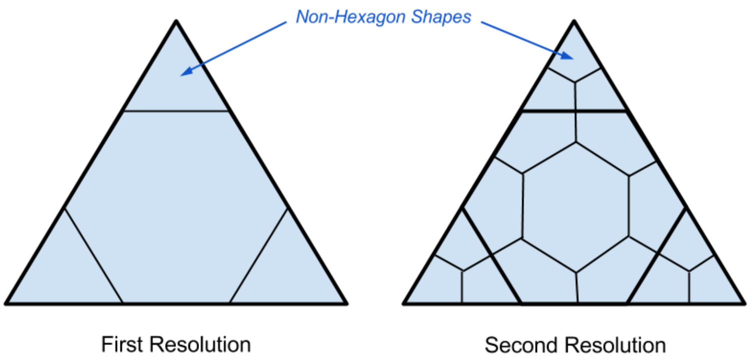

Figure 8 – Aperture 3 partitioning with a hexagonal grid

There are a number of different ways to partition triangular faces with hexagons, but the aperture 3 grid shown in Figure 8 allows the highest number of potential grid resolutions. In an aperture 3 system, to get round the fact that the grid is not congruent, the cells are rotated 30° relative to the cells in the resolution above, thus keeping the different resolution grids aligned.

In a hexagonal aperture 3 grid, the edge length of a cell is $\sqrt{3}$ of a cell in the next highest resolution, therefore the grid system is sometimes called "Square Root Three Subdivision".

Despite their extra complexity hexagonal grids have a number of advantages:

- Hexagons are the only tessellating shape where the centres of the neighbouring cells are all at equal distance. This means that distortion caused by approximating a point to the nearest cell is consistent in all directions.
- Hexagons are the most compact tessellating shape. Their cell's centre points are closer together than any other type of grid with cells of the same area. Being compact reduces distortion caused when a point is assigned a cell because the centre of the cell will be closer to the position of the original point.
- It is possible for a hexagonal grid to have a smaller aperture than a QTM.

A disadvantage of hexagonal grids is that they produce a small number of cells that are smaller than the others. Because it is not possible to tile all of the faces of the base polyhedron with hexagons, a non-hexagon cell will be formed at each of the polyhedron's vertices. The number of these non-hexagon polygons will be the same at every resolution. The number formed is dependent on the number of vertices of the polyhedron. In the case of the icosahedron they will be 12 pentagons. This non-hexagon cell will have a different area to the hexagon cells meaning that a small part of the DGGS will not be equal-area. [5]

7.4 Isosceles Triangle Grid

So far only grid systems that can be applied to polyhedrons with triangular faces have been considered, but of all the base polyhedrons discussed in Section 4.2, the two solids that provide the least angular and distance distortion are the dodecahedron and truncated icosahedron. These solids have pentagonal or hexagonal faces, not triangular faces, so to use either of these solids as a base polyhedron a different grid must be used.

No examples of partitioning a pentagon could be found in the literature review, so a new partitioning system has been proposed, shown in Figure 9. At the first resolution, the polyhedron's face (in this case a pentagon) is divided into isosceles triangles with their points at the face's vertices and centre. At the next resolution, each of those triangles is broken down into four smaller isosceles triangles, similar to the sub-division used for the QTM (Section 7.2). These triangles are then broken down into four more at the next resolution.

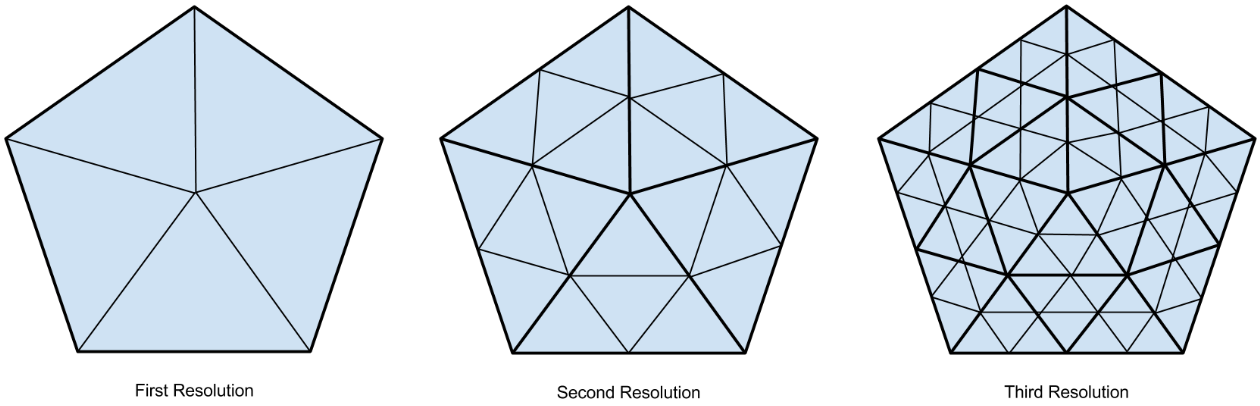

Figure 9 - Partitioning in an Isosceles Triangle Grid

This partitioning system can be applied to any regular shaped face, so it is suitable for partitioning any of the Platonic base polyhedrons. However it cannot be used with the truncated icosahedron because the cells on the hexagon faces will be a different area to those on the pentagon faces.

As well as being unsuitable for the truncated icosahedron, the grid system also has other disadvantages:

- Having the triangles at different orientations at the first resolution means that for pentagon faces there are ten different orientations that the triangle can be in, which increases the complexity when working with the grid.
- Isosceles triangles are not regular so they will have a greater maximum distance from the centre than an equilateral triangle of the same area. This will cause greater discrepancy between the projected point and the location of the centre of the DGGS cell.

The difficulty in partitioning the truncated icosahedron into equal area cells across the pentagonal and hexagonal cells means that this solid has to be discounted as a potential base polyhedron.

8 Cell Indexing

Grid indexing refers to the method for referencing a particular cell in the grid. Cells can either be indexed via coordinates that specify their location as an offset along a set of axes, or relative to the cells in lower resolutions.

8.1 Axial Coordinates

Axial coordinates are similar to Cartesian coordinates and are determined by working out the offset along two axes from a pre-defined origin. In order to apply them to hexagon or triangular grids the two axes are set at 60° to each other. An example of an axial coordinate system is shown in Figure 10. [10]

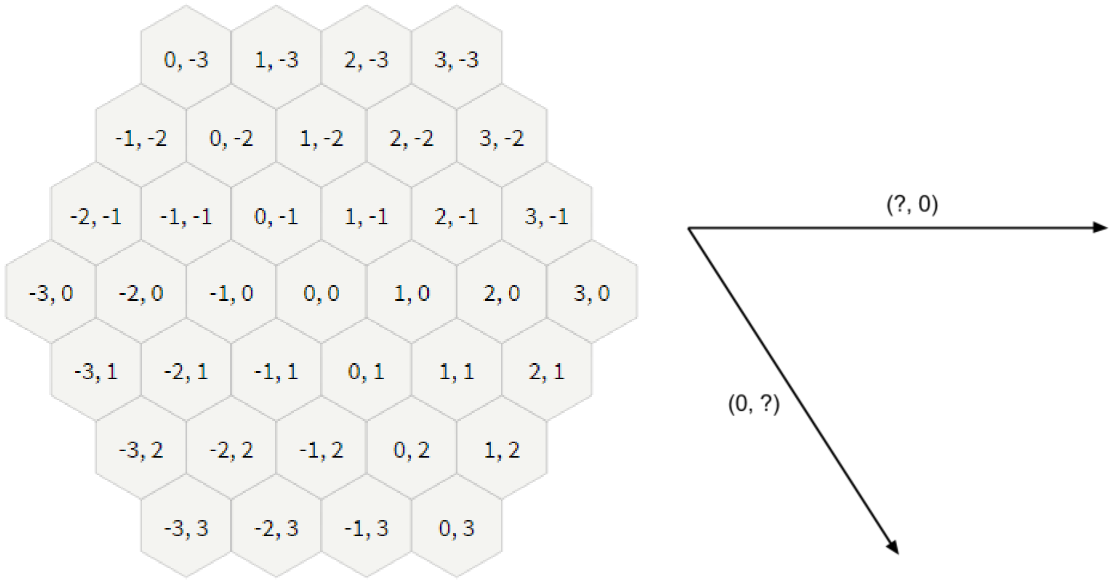

Figure 10 - Axial coordinates on a Hexagon Grid [10]

8.2 Offset Coordinates

Axial coordinates are effective for grids where the cells are aligned the same way at every resolution, as in a QTM. However in an aperture 3 hexagonal grid each resolution is made up from cells that are rotated 30° compared with the previous resolution. This means that the axes of the axial coordinate system must also be rotated.

An extension of the axial coordinate system is the offset coordinate system. In an offset coordinate system the axes remain fixed at each resolution but there is an offset in the axes between the odd and even rows/columns.

In a DGGS it is advantageous for the coordinate axes to remain constant across all the resolutions so the parent cells use the same axes as the child cells. Using the same axes makes determining the parent or children of a given cell a much simpler calculation.

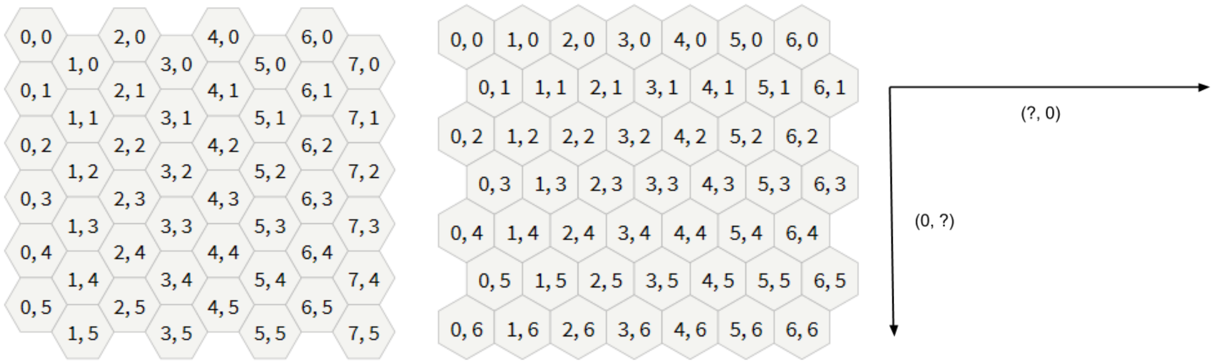

Figure 11 – Offset coordinates on a Hexagon Grid [10]

Figure 11 shows the two different possible orientations of hexagons on an aperture 3 hexagonal grid across its different resolutions. It demonstrates that the cells of the two grids are orientated at 30° relative to each other. In the left-hand diagram, the odd-numbered columns are offset by half a hexagon height in the y-axis direction. In the right hand diagram the odd-numbered rows are offset by half a hexagon in the x-axis direction. [10]

8.3 Hierarchical Index

In contrast to axial and offset coordinate systems, hierarchical indexing works by using the lower resolution grids to define a cell's location rather than with a set of axes. To calculate the hierarchical index of a cell, the lowest resolution grid is used to find the cell which contains the point. This process is repeated with grids at increasingly higher resolutions until the grid of the desired resolution is reached.

The index of the cell is a series of digits which specify the location of a cell relative to the cell chosen at the resolution before. A useful feature of a hierarchical index is that the number of digits indicates the resolution of the grid the cell is in.

An example of a hierarchical indexing system is shown in Figure 12.

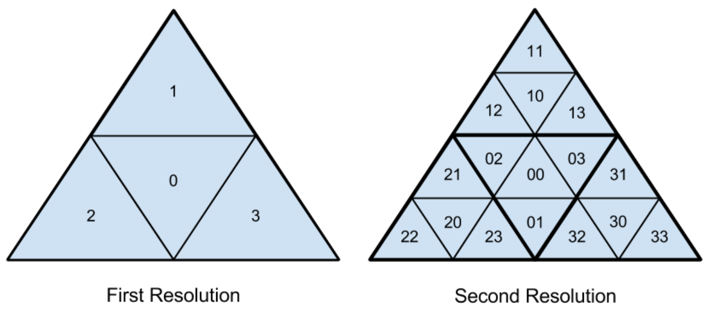

Figure 12 - Hierarchical Indexing on a QTM

8.4 Parent-Child Relationships

Given a cell index at a particular resolution it should be possible to calculate the cell index of that cell's parent and child cells. This section describes how the coordinate systems outlined above enable the parent-child relationships to be defined.

For the hierarchical indexing system, the parent-child relationship is trivial. A parent cell's index is simply the child cell's index with the final digit of the index removed. Similarly the child cell indices can be defined using the parent cell's index and appending the digits 0 to *N*, where *N* is the number of child cells contained in the parent. This can be observed from Figure 12 by comparing each parent cell in the first resolution with the child cells in the second resolution.

Axial and offset coordinate systems require more computation to determine parent or child cells but since the coordinates are simply an offset from the origin in two independent directions the geometry of the grid can be used to determine the parent or child cell indices. An exception to this is when the axial coordinate system does not have consistent axis orientations at all resolutions (e.g. in a hexagonal grid).

8.5 Working with Hexagonal Grids

Hexagonal grids have a further complexity in determining parent or child cells due to the fact that they are not congruent. Figure 13 shows how hexagons at successive resolutions are related. A cell at a low level resolution (large black hexagons) contains one complete higher resolution hexagon and a third of six other hexagons. Each low level resolution hexagon therefore has seven child cells, although six of those are not completely inside the parent. At the high level resolution some cells (e.g. the one marked in blue) are completely inside the parent hexagon and therefore have a single parent. Other cells (e.g. the one marked in red) span three parent hexagons.

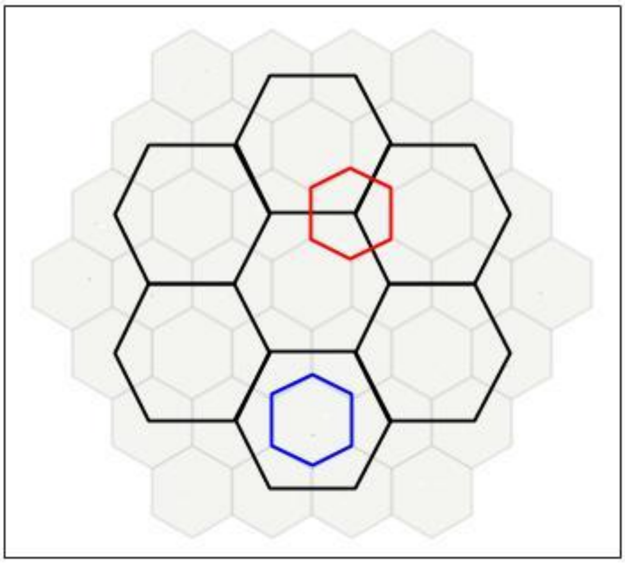

Figure 13 – Parent-Child relationships between hexagons

PYXIS, a company that have created a DGGS for commercial use called WorldView™ (see Section 9.3 for more information), have developed their own indexing system, which defines two distinct types of parent such that each cell has a single parent. [11] In Figure 13 the central large black hexagon would have children defined by the grey hexagon located at the centre of the hexagon and in addition the grey hexagons located at its vertices. The surrounding black hexagons would only have a single child cell (that at the centre of the hexagon).

This system has two potential problems. Firstly, the PYXIS indexing system is associated with a US patent [12] which may restrict the ability to utilise this type of indexing. Secondly, it would mean that the child cells of the low level resolution hexagons do not cover the same area (since some cells have seven children and some only one). Therefore the data from the child cells of one parent cell is not directly comparable with that from another since they are not equal in area.

It is proposed that, for hexagonal grids, each low level resolution hexagon should include all seven cells that are either fully or partly contained in the parent cell as child cells. Similarly for parent relationships, each cell that spans more than one parent cell should define multiple parent cells. This method would mean that child cells can belong to multiple parent cells but the DGGS would provide sufficient information to client applications such that it would be able to process the parent-child relationship as it sees fit and spread the data for the child cell over the parent cells as required.

9 Existing Solutions

Before developing a new DGGS for this project, the literature review looked at some of the existing DGGS implementations to see if any would be suitable as a basis for this project. Each solution provides functionality that satisfies some of the DGGS requirements but a direct comparison of the solutions is difficult due to the variation in the functionality available. Most solutions provide a projection onto one or more polyhedrons but do not provide functionality to assign each DGGS cell a unique index.

9.1 HEALPix

HEALPix is a DGGS developed by NASA's Jet Propulsion Laboratory (JPL) to store background cosmic microwave energy measured by their satellite missions. The purpose of HEALPix differs from that of this project and other existing DGGS software, because it aims to create a map of the sky around the Earth rather than of the Earth's surface.

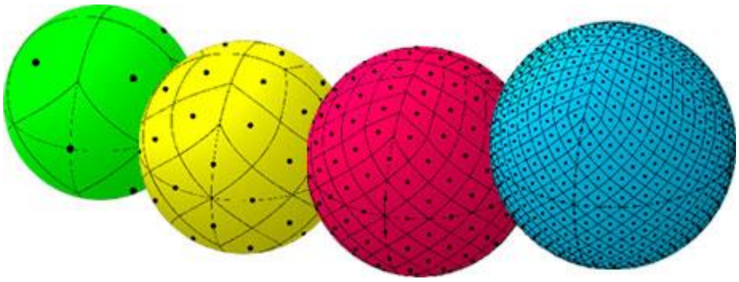

Figure 14 - The first 4 resolution grids of the HEALPix system [13]

The system uses an octahedron as a base polyhedron with an aperture 4 quadrilateral grid. The HEALPix software is open source and can be downloaded freely, without registration. Because HEALPix is not intended for the Earth's surface it is not suitable for use in this project.

The latest version of HEALPix, v3.20, was released in December, 2014. [13]

9.2 SCENZ-Grid

SCENZ-Grid is an open source DGGS currently being developed by the New Zealand government research institute, Landcare Research. It is based on the HEALPix projection, with the projection reversed to map the surface of the Earth instead of the sky (known as rHEALPix). Although it is open source, the SCENZ-Grid website does not contain any links to the source code or references to any releases. It is assumed therefore that the project is still in development and no official release of the code has yet been made. [14]

9.3 PYXIS WorldView™

WorldView™ is a commercial implementation of Snyder's projection on to an icosahedron with an aperture 3 hexagonal grid (commonly referred to as an ISEA3H DGGS). PYXIS has used their DGGS to allow their clients to integrate multiple sources of geospatial information. Because it is a commercial product, WorldView™ cannot be released under an open source license so is not suitable for use in this project. [11]

9.4 DGGRID

Developed by the Southern Terra Cognita Laboratory, DGGRID is a public domain software program for creating and manipulating DGGSs. The software uses an icosahedron as its base polyhedron, but allows the other design aspects of the DGGS to be customised. DGGRID supports grids of triangles or diamonds with an aperture of 4, or hexagons with apertures of 3, 4, or a mixed sequence of apertures 3 and 4.

DGGRID can only create grids and not convert specific points to cells. It is therefore not able to be used directly in this project. However it does include the ability to output the grids it creates as KML files which can be overlaid on Google Earth making it a useful tool when trying to visualise different DGGS designs.

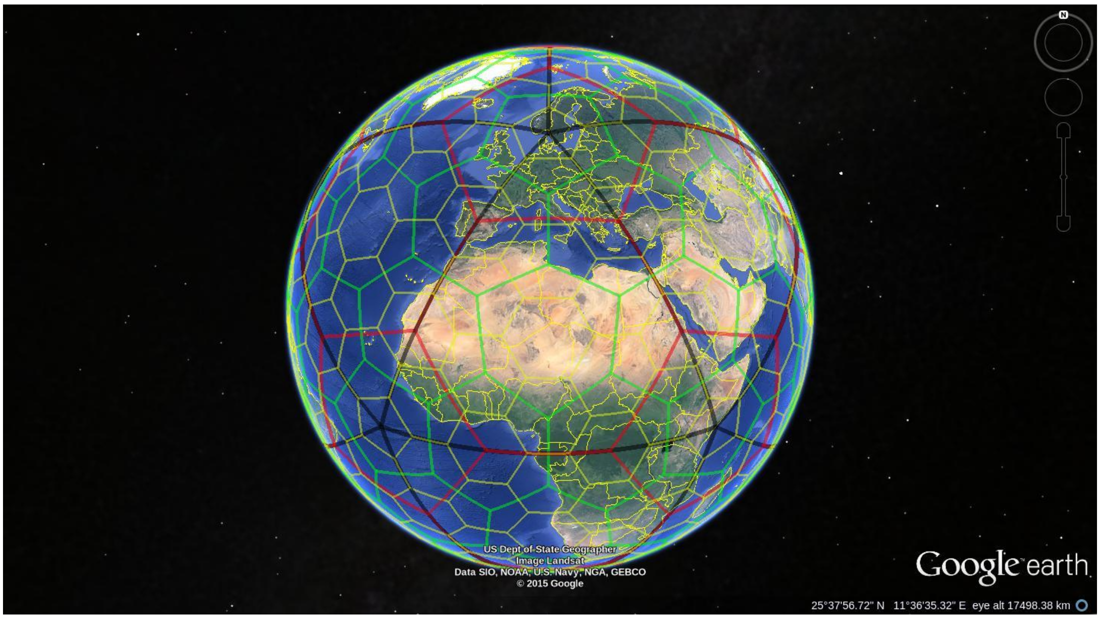

Figure 15 - Google Earth screenshot with KML grids produced by DGGRID

The latest version of DGGRID, version 6.1, was released in May, 2013. [15]

9.5 Proj.4

Proj.4 is a cartographic projections library written in C that can convert latitude and longitude co-ordinates into a wide range of co-ordinate systems.

The following DGGS projections are supported:

1. Icosahedral Snyder Equal Area (ISEA3H)
2. Reverse HEALPix (rHEALPix)
3. Quadrilateralized Spherical Cube (QSC)

Being open source, Proj.4 showed promise as a useful starting point for the development of this project. Although the core functionality of Proj.4 is reliable and well established, the development of its DGGS functionality still appears to be in its infancy. There is very little documentation available on these projections and when the code was trialled it appeared to contain bugs. Some of the projections also have limited functionality. For example points can be converted to the Snyder Icosahedron Equal Area projection, but there is no inverse projection to convert them back.

The latest version of Proj.4, v4.9.1, was released in September, 2014. [16]

9.6 Other Solutions

A list of other known DGGS implementations has been included below:

- **Spherekit** – Uses a DGGS for spatial interpolation and comparison of spatial interpolation algorithms. [17]
- **SCRIP (Spherical Coordinate Remapping and Interpolation Package)** – DGGS implementation written in Fortran 90. [18]

There is currently very little information available about these legacy solutions.

10 Prototyping

The literature review has provided an understanding of DGGS design, but to ascertain which of the suitable design options identified will best meet the requirements of this project a prototyping phase will be undertaken.

Figure 16 shows the choices for the prototypes:

Prototype	Base Polyhedron	Grid Partitioning	Cell Indexing
ISEA4T	Icosahedron	Quaternary Triangular Mesh	Hierarchical Index
ISEA3H	Icosahedron	Hexagonal Grid	Offset Index
DSEA4I	Dodecahedron	Isosceles Triangle Grid	Hierarchical Index
OSEA4T	Octahedron	Quaternary Triangular Mesh	Hierarchical Index

Figure 16 – Design choices for the DGGS prototypes

The DGGS prototype names are encoded using a standard notation that identify the polyhedron, projection and partitioning system. The 'SEA' part of the name signifies that it uses the **S**nyder **E**qual **A**rea projection which all of the prototype systems use. The first letter indicates which polyhedron is used (**I**cosahedron, **D**odecahedron or **O**ctahedron). The number and final letter signify the aperture and partitioning shape (**T**riangle, **I**sosceles Triangle or **H**exagon). The only part of the DGGS that is not encoded in this standard notation is the indexing system.

The design choices for each prototype are justified below:

- **Base Polyhedron** – After discounting the truncated icosahedron, the two polyhedrons that provide the least distortion are the icosahedron and dodecahedron so they have been chosen for the prototypes. For the icosahedron, the orientation will be with the top and bottom vertices at the poles. For the dodecahedron, the poles will be in the centre of the top and bottom faces.
- **Orientation** – The prototypes will use a consistent orientation for the base polyhedron because the chosen orientation will depend on the main areas of interest which are yet to be identified. The evaluation of the prototype will ensure that the DGGS is exercised at a uniform spread of points covering the globe.
- **Projection** - All the prototypes use the Snyder projection because this is the only known equal-area projection that could be applied to the chosen base polyhedrons.
- **Grid Partitioning** – The only grid partitioning option for pentagon faces is the Isosceles Triangle Grid so this has been used for the dodecahedron prototype. The other two options will be tested with the icosahedron.
- **Cell Indexing** – It is expected that the hierarchical indexing will be faster for parent/child querying so it will be used for two of the three prototypes. Offset coordinates will be used for the Hexagonal Grid on the icosahedron to avoid any chance of infringing on the existing patent held by PYXIS. [12]

In the prototype systems defined above, the approaches obtained from the literature review have been supplemented by additional methods proposed by Riskaware that may benefit DGGS performance. Firstly, the isosceles triangle indexing system that is used to partition the pentagonal faces of the dodecahedron. In the literature studied for this review, the dodecahedron is identified as having the least distortion of the Platonic solids from the equal-area projection, but is often dismissed due to the difficulty in partitioning the pentagonal faces using a regular shape. Isosceles triangles are proposed to partition the pentagonal faces. With this approach, increased distortion could be encountered since isosceles triangles are more skewed than equilateral ones. This will be included in the total distortion that will be measured in the evaluation of the prototype.

The second proposed method is the offset coordinate system used by the ISEA3H prototype. Due to the patent held by PYXIS for a hierarchical indexing system for hexagonal grids an alternative is required. The literature review highlighted an axis-based approach for identifying cells in the grid, however this system would cause difficulties when determining parents or children due to the fact that the hexagonal grid rotates by 30 degrees at each successive resolution. Offset coordinates, which are used in other fields, extend the axial concept but allow the coordinate axes to be fixed for all grid resolutions, enabling the parent/child relationships to be determined using the grid geometry.

11 Evaluation Criteria

In order to objectively assess the suitability of the prototypes it is necessary to first define a set of evaluation criteria. This section describes the criteria used to compare the different designs for the OpenEAGGR library with details about how each one will be measured.

11.1 Equal Area

An important requirement given for the DGGS was that it should preserve the area across the projection. All the prototypes use the equal-area projection developed by Snyder in his his paper “An Equal-Area Map Projection for Polyhedral Globes” [3]. This projection guarantees that areas on the Earth’s surface will be preserved when they are projected on to the base polyhedron. The implementation of the projection for the prototypes will be verified for correctness.

If all the cells in the DGGS are the same area for a given resolution, as is the case for those prototypes with a triangle-based gridding system, then every cell will represent the same area on the Earth’s surface and therefore fully meet the criteria for an equal area projection.

The gridding system used in the ISEA3H prototype however has a small number of cells which are of a different area to those of the rest of the grid due to the non-congruent nature of its hexagonal grid. This design therefore is not truly equal area. For the prototype’s base polyhedron, the icosahedron, there are 12 of these cells at any given resolution, which are $\frac{5}{6}$ the area of the other cells.

Resolution	Number of Cells	Area of Cells (in km²)	Inter-Cell Distance (in km)	% covered by Non-Equal Area Cells
15	143,489,072	3.554735017	1.862107055	8.363e-6
16	430,467,212	1.184911672	1.075088010	2.788e-6
17	1,291,401,632	0.394970557	0.620702352	9.292e-7
18	3,874,204,892	0.131656852	0.358362670	3.097e-7
19	11,622,614,672	0.043885617	0.206900784	1.032e-7
20	34,867,844,012	0.014628539	0.119454223	3.442e-8

Figure 17 - Percentage of Earth's surface covered by non-equal area cells in an aperture 3 hexagonal grid [15]

Figure 17 shows at the resolutions at which the DGGS is likely to be used at these cells will cover only a very small area of the Earth’s surface. It is unlikely therefore that these cells will significantly impact the performance of the DGGS so the design can still be considered equal-area.

Because the non-equal area cells are always located at the vertices of the base polyhedron, the effect can be further reduced by positioning the polyhedron’s vertices away from areas of interest on the globe.

11.2 Hierarchical Consistency

Hierarchical consistency refers to how the grids at different resolutions are related, and the ease with which the parents or children of a particular cell can be determined.

For congruent grids, like the triangle-based partitioning systems of the ISEA4T, DSEA4I and OSEA4T prototypes, determining the parent or children of a cell is straightforward because the cells will always have one parent and four children (except at the first resolution of the isosceles triangle grid where there will be five children).

For non-congruent grids, like the hexagonal grid used for the ISEA3H prototype, determining the parents and children is more complicated. The grids for different resolutions are at different orientations and the children of a cell do not fit exactly inside the parent. This means that whilst some cells are contained by a single parent, others may be shared between multiple parents (explained in more detail in Section 8.5). This extra complexity will increase computation time, but will also impact software using the OpenEAGGR library which will need to handle multiple parent cells.

The speed at which parent and child cells can be determined by the OpenEAGGR library for the different prototypes will be benchmarked by the performance tests, described in Section 12.

11.3Conversion Speed

The OpenEAGGR library must be able convert a large number of points as quickly as possible; therefore conversion speed should be an important consideration when selecting a design. Timings will be made of how long it takes to convert a point with latitude and longitude coordinates on the Earth's surface to a cell in the DGGS, and then how long it takes to convert the cell back again.

Conversion speeds for converting points to and from DGGS cells will be recorded separately, and performed for a range of points around the world using the performance tests, described in Section 12.

11.4Conversion Accuracy

When a point on the Earth's surface is converted to a DGGS cell its position will be distorted. The conversion accuracy is the distance the point moved away from its original location after conversion back to latitude and longitude coordinates.

There are three factors that affect the accuracy:

- **Projection** – It is impossible to create an equal-area projection which can perfectly preserve a point's location. However, since all the prototypes use the same projection this source of inaccuracy will be constant across all of them.
- **Base Polyhedron** – The more closely the base polyhedron resembles the spherical nature of the Earth the more accurate the conversion will be. The distortion introduced by the different polyhedrons is well understood and discussed further in Section 4.2. As a result, no further testing will be conducted to assess the effect of the base polyhedron on the conversion accuracy.
- **Grid Partitioning** – Once the point has been projected on to the polyhedron its location must be approximated to the nearest DGGS cell. For a given cell size, grid partitions with the lowest inter-cell distance will introduce the least distortion because on average the cell centres will be closer to the projected point.

The grid partitioning effect of the conversion accuracy will be tested in the performance tests, described in Section 12.

11.5Ease of Human Visualisation

The ease of which each design can be visualised by humans is difficult to quantify and therefore will not be tested specifically in the evaluation. Instead a discussion has been included of the how the aspects of the design impact the visualisation of the DGGS.

How points on the Earth are projected on to the base polyhedron could potentially have the most impact on human visualisation. For example, Figure 5 shows a heavily distorted projection of the Earth where the land masses have been heavily distorted in order to maintain an equal-area projection. Humans would find it difficult to use this map for tasks like navigation because the distance and angles are so radically different from the types of projection they are used to seeing. Although all the prototypes use the same projection, this type of distortion can be reduced by using a base polyhedron that is a closer approximation to the shape of the Earth. The findings in Section 4.2 show that using the dodecahedron as a base polyhedron produces slightly less distortion than the icosahedron. However neither distorts the Earth's features to a level where a human would struggle to recognise them.

Human visualisation of the base polyhedron can be helped by orientating it relative to well-known points on the Earth. Perhaps the most easily recognised are the poles, so positioning the polyhedron with its vertices or face centres at the poles would be convenient for humans. It may be of further assistance to rotate the polyhedron so the Earth's meridian (longitude = 0°) is aligned with an edge or face centre.

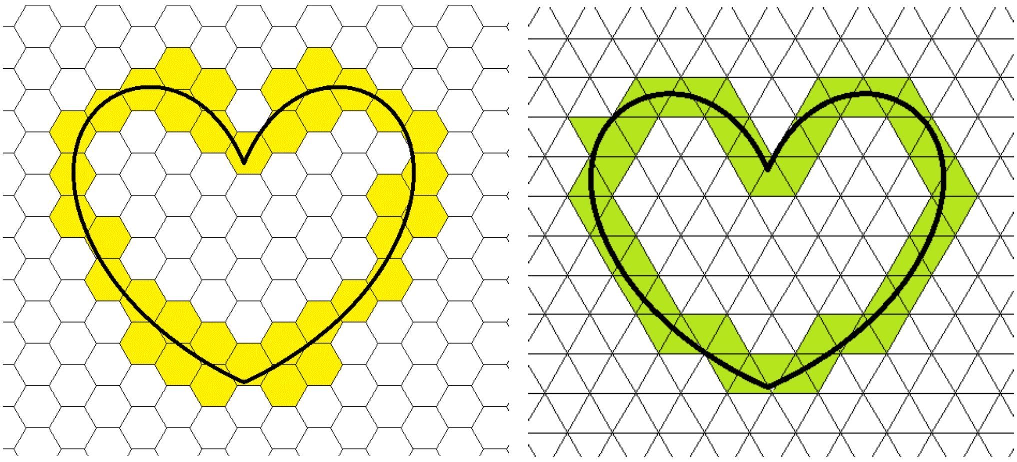

Figure 18 - Representing a heart shape on hexagonal and triangle grids

The ease at which the different grid systems can be visualised is likely to be quite subjective and application dependent. The congruency of triangular grids means that cells of different resolutions are aligned and perhaps more easy to visualise than cells on a hexagonal grid where the cells at different resolutions are orientated differently and overlap. However hexagonal grids have the same distance between all neighbouring cell centres, unlike triangular grids, and therefore could potentially more accurately display different shapes, see Figure 18.

The preferred method for visualising the location of cells using the indexing system is also likely to be subjective. Hierarchical indexing allows parent and child cells to be determined easily, but humans are more familiar with axis based coordinate systems (e.g. Cartesian), so they may prefer the offset coordinate system for that reason.

12 Performance Testing

To evaluate the software implementations of the three prototypes, a set of performance tests will be implemented to compare their accuracy and speed. The tests will perform a set of operations on points located at different positions around the world with a variety of associated accuracies.

12.1 Input Points

In order not to favour a particular design, care will be taken to ensure the points cover the whole surface of the Earth and are evenly spaced (in terms of spherical distance). It is expected that the prototypes will perform better closer to the face centres of their base polyhedron, so the fact the points are evenly spaced is important. The points will be spaced a set distance apart with a uniform distribution around the world.

The resolution the DGGS uses for the cell is determined by the accuracy associated with the point. The more accurate a point is, the smaller the area of its corresponding DGGS cell. It is expected that the OpenEAGGR library will be used with points with accuracies in the range of 1m to 1km, so each set of points will be repeated with a different associated accuracies in that range.

12.2 Speed Testing

The following tests will be performed for each point:

- **Conversion to DGGS cell** – Converting latitude and longitude coordinates to a cell in the DGGS.
- **Query parent of cell** – Calculating the parent cell or cells for the converted cell.
- **Query children of cell** – Calculating the child cells for the converted cell.
- **Conversion back to latitude and longitude** – Converting the generated DGGS cell back to a point with latitude and longitude coordinates.

In order to assess how efficiently each prototype could perform these tasks, each task will be timed separately with the average time taken for each calculated from all the points and accuracies.

12.3 Grid Partitioning Accuracy

In addition to checking the speed of each operation, the accuracy of the grid partitioning will also be measured in two ways:

- **Conversion Error** – Spherical distance between the original latitude and longitude coordinates and those returned when the cell was converted back to a point. This value represents the accuracy of the gridding system because it gives the distance the centre of the chosen cell was from the original point.
- **Accuracy Error** – The accuracy associated with the point determines the resolution of the grid used in the DGGS. The accuracy error is defined as how closely the chosen cell's area approximates the point's associated accuracy. Grid partitions with a lower aperture should have a reduced accuracy error because there will less difference between the area of cells at consecutive resolutions.

The maximum, mean average and standard deviation of these error values will be calculated at each accuracy level.

The results from running these tests are presented in Section13.

13 Prototype Evaluation

13.1 Evaluation Framework

The DGGS prototypes described in Figure 16 were evaluated using a test framework developed to generate the metrics outlined in Section 12. The framework utilised the google test framework, which has also been used to implement the unit testing for the prototype system, allowing the evaluation framework to be run from Eclipse and providing integrated functionality to output the metrics to an XML file.

The prototypes have been programmed to a set of interfaces covering each of the DGGS design choices (polyhedral globe, projection, grid partitioning and grid indexing). This allows various DGGS implementations to be built up from these component parts to create the prototypes for evaluation.

The framework measures the performance and accuracy for each prototype system by generating a grid of latitude/longitude points across the globe which are converted to DGGS cells and back again. The latitudes of the points are spaced at 10km intervals and at each latitude the longitudes of the points are selected at a spacing corresponding to 10km. The prototypes are evaluated using this grid of points with a range of input accuracies ranging from 0.5m to 1km.

13.1.1 *Prototype Limitations*

The test points were defined with latitude/longitude values in spherical Earth coordinates since the projection converts from that coordinate system to the chosen polyhedron. In the alpha release of the OpenEAGGR library, the API will accept coordinates in standard WGS84 latitude/longitude and convert these to spherical Earth coordinates before projecting them. This additional processing will be required for whichever prototype is selected, and has therefore not been included in the prototype evaluation.

Since a hexagonal grid is not congruent, ISEA3H grid cells will span more than one polyhedron face at the joins between the faces of the icosahedron. In the ISEA3H prototype, these grid cells will be assigned different IDs depending on the face on which the original point lies. Additionally, the pentagonal cells at the vertices of the icosahedron have been treated as hexagonal cells in the indexing system. This should be insignificant in the context of the evaluation since the conversion between the original point and the DGGS cell will have the same speed and accuracy. It may affect the parent/child performance for points that map to cells on the edges or corners of the icosahedron faces, but these are such a small percentage of cases that it should not affect the results. The alpha release of the OpenEAGGR library will include handling of these corner cases if the ISEA3H implementation is chosen.

13.2 Evaluation System

The prototype evaluation was run on a system with the following specifications.

System Property	Specification
Machine Type	Dell Precision M4800 Laptop
Operating System	Windows 7 Professional SP1 64 bit
Processor	Intel Core i7-4910MQ 2.90GHz
RAM	16.0GB

Figure 19 – Evaluation system properties

13.3 Conversion Speed

This metric measures the number of conversions per second for converting latitude/longitude points to DGGS cells and back again. Each conversion includes projection of the point onto the polyhedron, determination of the cell containing the point at the appropriate resolution and the generation of the cell index. The reverse conversion includes calculating the cell based on the DGGS cell ID and re-projection from the polyhedron back on to the Earth.

Figure 20 shows the results from the evaluation.

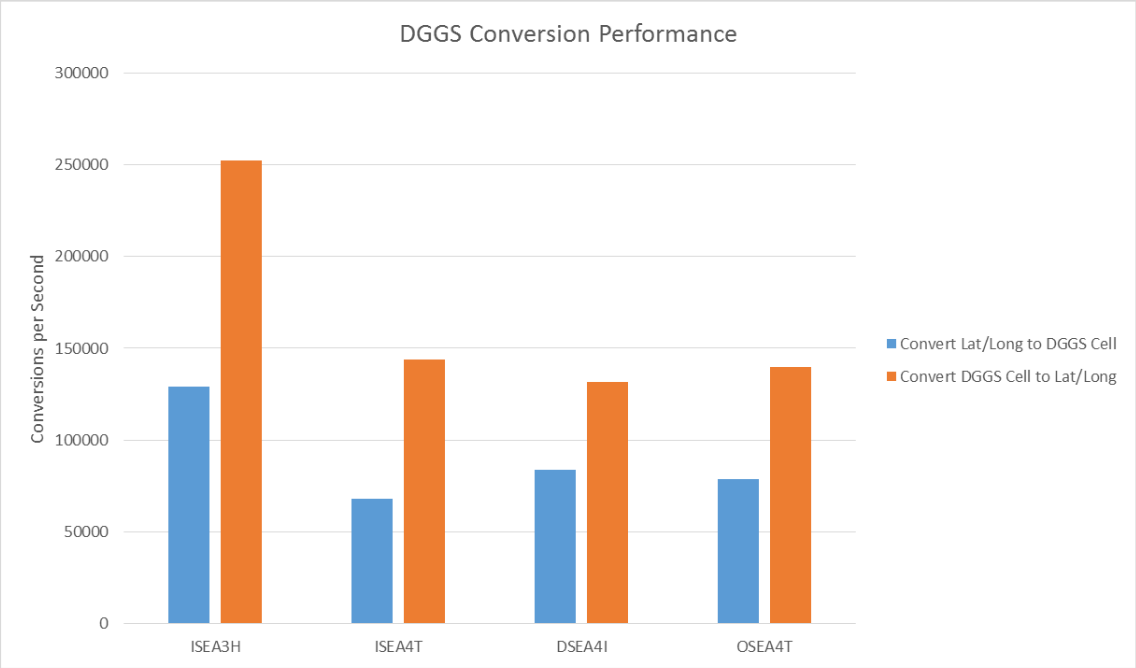

Figure 20 – DGGS conversion performance

The best performing DGGS prototype is the ISEA3H implementation which significantly out-performs the other prototypes in both the forward and reverse conversions. The main difference between this prototype and the others is that it uses offset coordinates to index the cells rather than a hierarchical system. The offset coordinate system has better performance since the cell index can be calculated from the geometry of the DGGS, whereas the hierarchical index has to be calculated by iterating through each of the resolution levels to determine which cell partition contains the point until the required resolution is reached.

Out of the other prototypes, the DSEA4I implementation has the best performance converting from latitude/longitude points to DGGS cells. In the reverse direction there is little between the three prototypes. There are two competing factors that determine the performance of these systems. Firstly, the fewer the number of faces on the polyhedron the faster the conversion will be. This is due to the fact that determination of the face the point is located on is performed by trying each face in turn until the correct face is found. The fewer faces that have to be tried the faster the conversion will be. Secondly, the smaller each face is (i.e. the more faces on the polyhedron) the faster the cell id generation will be in a hierarchical system. This is because the smaller the initial face is the fewer iterations are required to reach the required resolution level.

The OSEA4T implementation has the fewest faces (8), but the area covered by each face is 2.5 times larger than for the ISEA4T implementation (with 20 faces). The DSEA4I implementation has the advantage that it has an intermediate number of faces (12), but also that the pentagonal faces are partitioned into five triangles which means that the area covered by the first resolution partitions is $^{1}/_{5}$ of the face, rather than $^{1}/_{4}$ in the case of the ISEA4T and OSEA4T implementations.

13.4Conversion Error

This metric measures the distance between the original latitude/longitude point passed to the DGGS and the resulting latitude/longitude point generated by converting it to a DGGS cell centre and back again. It measures the error caused by mapping the projected point to a DGGS cell at the corresponding grid resolution then re-projecting it.

Figure 21 shows the average conversion error for each of the DGGS prototypes over a range of original point accuracies. The error bars represent the standard deviation of the data. It shows that all of the prototype systems have lower errors as the accuracy of the original point increases. This is expected since a higher original accuracy will be represented by a higher resolution grid in the DGGS with smaller grid cells, so the difference between the original point and the centre of the allocated DGGS cell will be smaller.

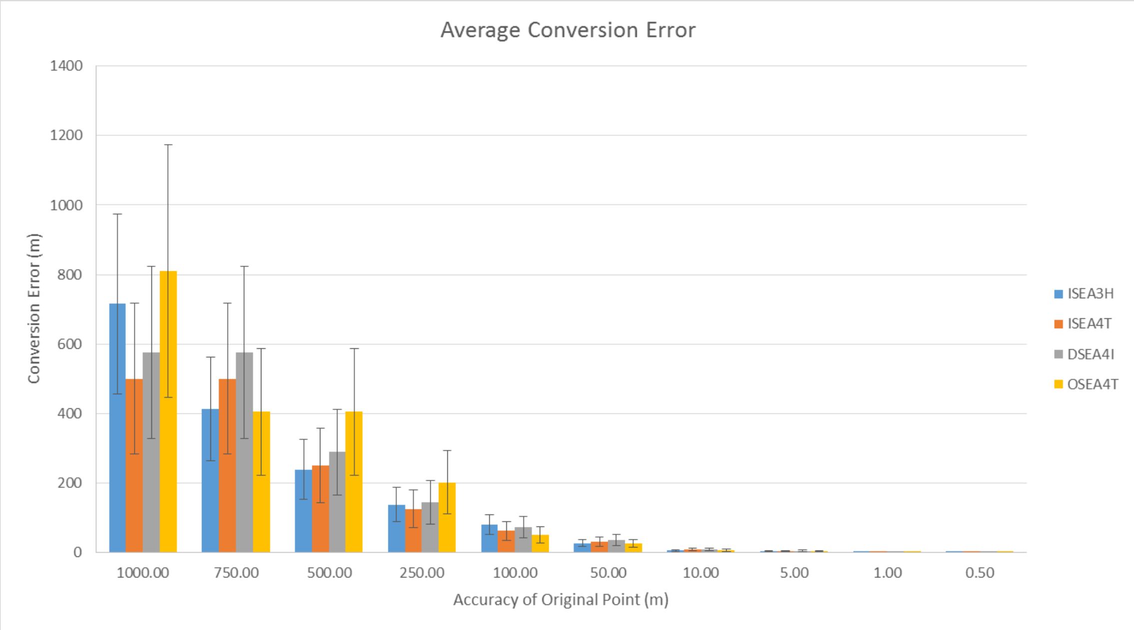

Figure 21 – Average DGGS conversion error

A more meaningful measure of the conversion error is the error as a percentage of the accuracy supplied with the original point. This is shown in Figure 22, and allows the error at each accuracy level to be compared.

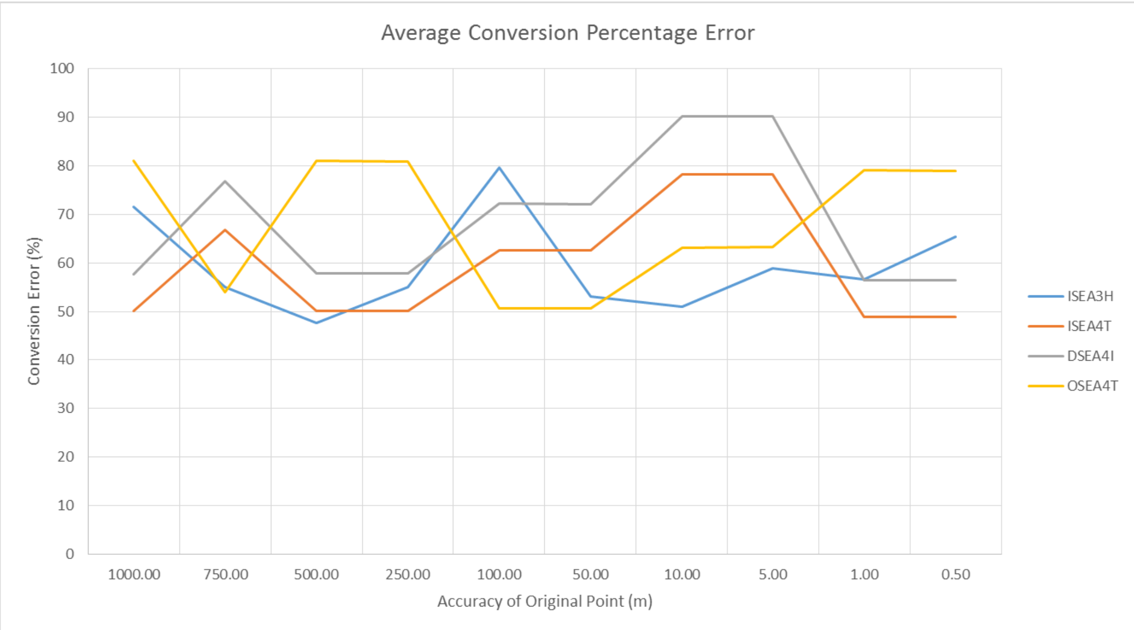

Figure 22 – Average conversion error as a percentage of the original error

Across all accuracy levels, the percentage conversion accuracy error ranges between about 50% and 90% for all of the DGGS prototypes. While there is some fluctuation in the percentage error over the different accuracy levels, it can be seen that the ISEA3H prototype generally has a lower error than the others with the ISEA4T prototype also performing well at some resolutions. The average percentage accuracy error across all resolutions for each prototype is shown in Figure 23.

DGGS	Average Percentage Error
ISEA3H	59.38%
ISEA4T	59.62%
DSEA4I	68.73%
OSEA4T	68.23%

Figure 23 – Average percentage conversion accuracy error

The factors that determine the conversion error include the polyhedron being projected onto, the size of the DGGS cell representing the point, the shape of the cell and the location of the point within that cell. In terms of the projection, the dodecahedron has the lowest percentage error (9%), the icosahedron is next (16%) and the octahedron worst (35%). These values are the maximum errors and have been taken from Snyder's paper [3]. The error from the projection will be much smaller than the errors resulting from representing the point by a DGGS cell whose centre is offset from the original location.

Clearly the larger the DGGS cell, the larger the potential error (since points further away from the cell centre will be represented by that cell). The cell size chosen will be dependent on the stated accuracy of the original point – the resolution with a cell size closest to the original accuracy will be selected. It follows that a DGGS that has a smaller difference between the areas of the cells at consecutive resolutions will tend to have smaller conversion error. This favours the ISEA3H implementation over the other DGGS prototypes since it is an aperture 3 system, rather than aperture 4.

The cell shape also affects the accuracy since the average distance between a point in the shape and the centre is different for each shape. A hexagonal grid is best in this regard (ISEA3H) as the vertices are closer to the centre for a given area. Equilateral triangles (ISEA4T and OSEA4T) are next best as the vertices are further from the centre but all vertices are the same distance away. Isosceles triangles (DSEA4I) are worse since the two vertices at the base of the triangle extend further from the shape centre. These shapes are shown in Figure 24.

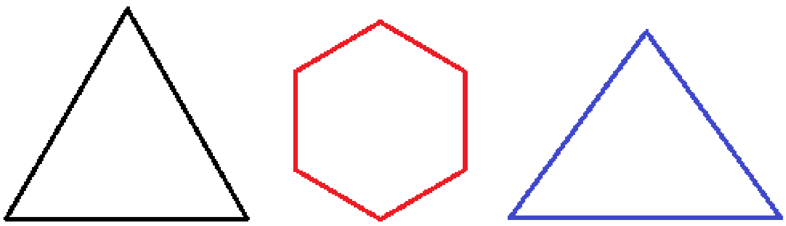

Figure 24 – Equilateral triangle (black), hexagon (red) and isosceles triangle (blue) each representing the same area

The effect of using isosceles triangles rather than equilateral triangles can be seen by comparing ISEA4T with DSEA4I. The former uses equilateral triangles to form its grid whereas the latter uses isosceles triangles. The conversion error shows the same profile in Figure 22, with DSEA4I offset to a higher percentage error, despite a lower projection error due to the polyhedron.

To summarise, the OSEA4T implementation has highest projection error. The DSEA4I implementation benefits from less projection error, but the isosceles triangles cause higher partitioning errors. The ISEA3H and ISEA4T implementations have the lowest conversion error. From Figure 22 it can be seen that at some stated accuracies the ISEA3H implementation has the lowest conversion error, and at others the ISEA4T implementation has the lowest conversion error.

13.5 Accuracy Error

This metric measures the error between the accuracy supplied with the original latitude/longitude point and the accuracy that is returned when the point is converted to a DGGS cell and back again. The returned accuracy will not be the same as the supplied accuracy since when the point is converted to a DGGS cell since a resolution whose cell area is closest to the supplied accuracy is used. When the DGGS cell is converted back to a latitude/longitude point the cell area is used to define the point's accuracy[2].

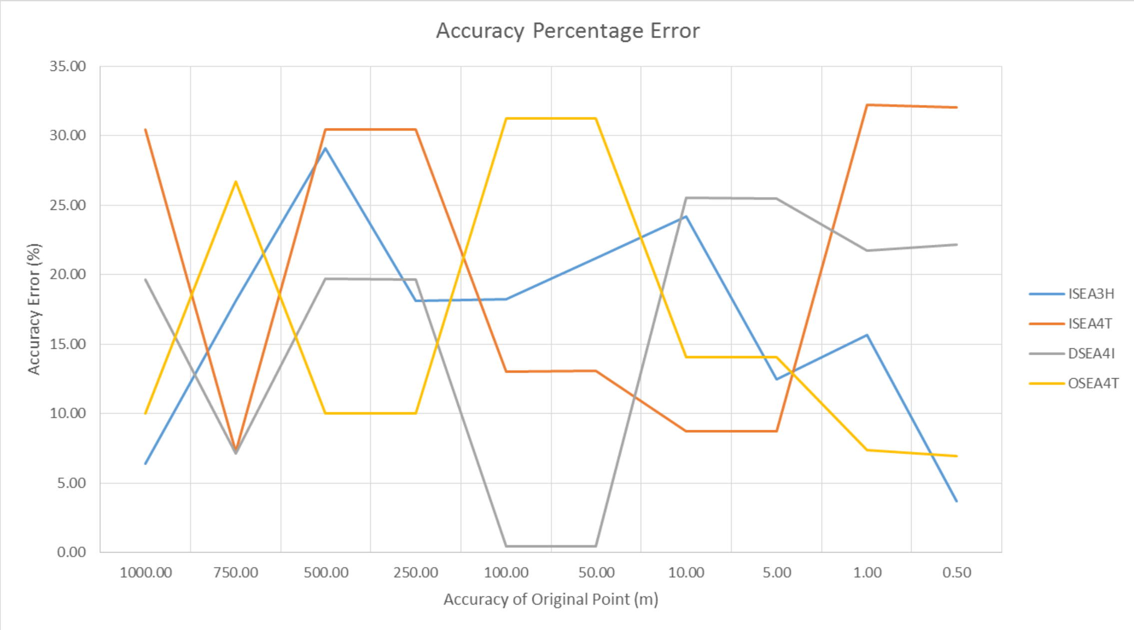

Figure 25 – Accuracy percentage error

Figure 25 shows the percentage accuracy error for each of the DGGS prototypes over a range of initial accuracies. The accuracy error varies considerably for all of the prototypes, but the maximum error is around 32%. The variation is caused by the proximity of the initial accuracy to the cell area at the resolution representing that accuracy level. Some accuracies will be well-matched to the resolution cell areas while others will not.

The average error across all of the accuracy levels for each prototype is shown in Figure 26. It shows that there is not much difference between the prototype systems. ISEA4T has a higher percentage error, but this may just be due to the accuracy levels chosen in the evaluation (there is no reason that this implementation should be worse than the other hierarchical triangular cell indexing systems).

DGGS	Average Percentage Error
ISEA3H	16.70%
ISEA4T	20.63%
DSEA4I	16.19%
OSEA4T	16.16%

Figure 26 – Average accuracy percentage error

13.6Parent-Child Querying

These metrics measure the number of parent and child queries per second for each prototype.

13.6.1 *Parent Querying*

The performance of the parent queries for each prototype is shown in Figure 27. It highlights the difference between the different indexing systems used in the prototype implementations. The ISEA3H implementation performs significantly worse than the others due to its offset coordinate indexing. In a hierarchical indexing system, determination of the parent cell's ID is simply a matter of removing the last

---

[2] A spherical cap is created based on the stated point accuracy. The area of this determines the DGGS resolution chosen. A spherical cap with the same area as the DGGS cell is used to calculate the accuracy of the resulting point.

digit from the child's cell ID. In an offset coordinate system, the geometry of the grid must be used to work out the parent cell.

The hexagonal grid is further complicated because some cells lie completely within a parent cell while others are covered by three parents (Figure 13). The ISEA3H implementation returns all of the parent cells which adds to the processing time. For the hierarchical triangle system, only one parent is required since the grid is congruent.

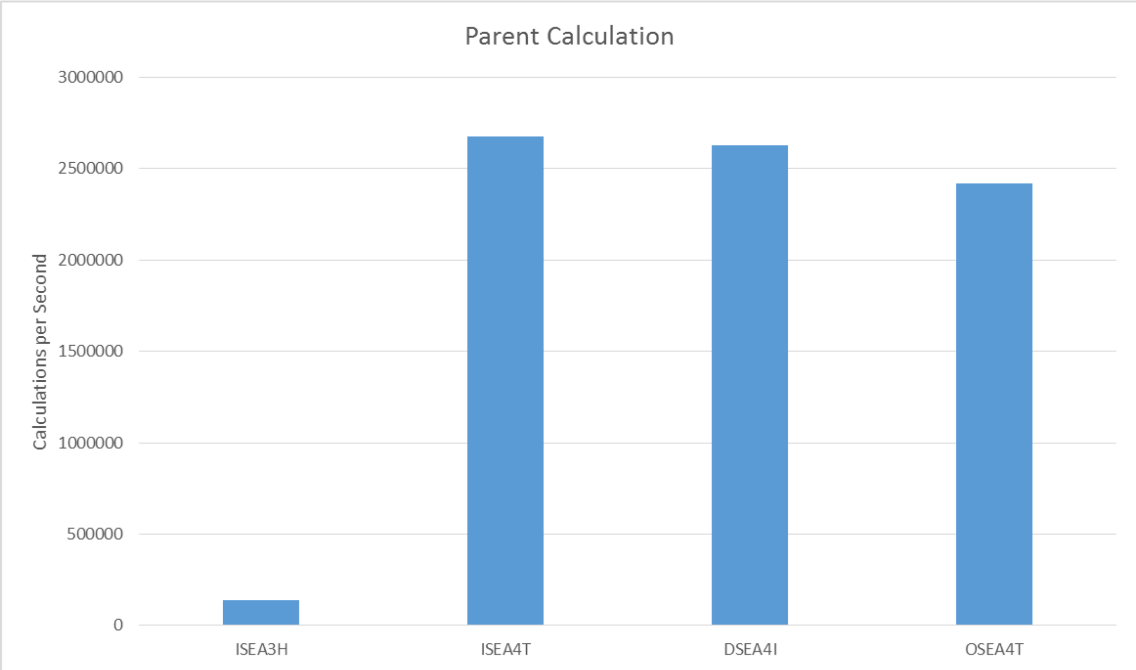

Figure 27 – Performance of parent queries

13.6.2 *Child Querying*

The performance of the child queries for each prototype is shown in Figure 28. There is not as large a difference between the systems as in the parent query case, but for these queries the ISEA3H implementation performs better than the other systems. Again these differences are caused by the way child cells are calculated for the offset and hierarchical coordinate systems. In the hierarchical coordinate system, the child cells are more complicated to process since they contain their entire hierarchy in their definition. An offset coordinate system is simpler because the child cells can been determined from the geometry of the grid.

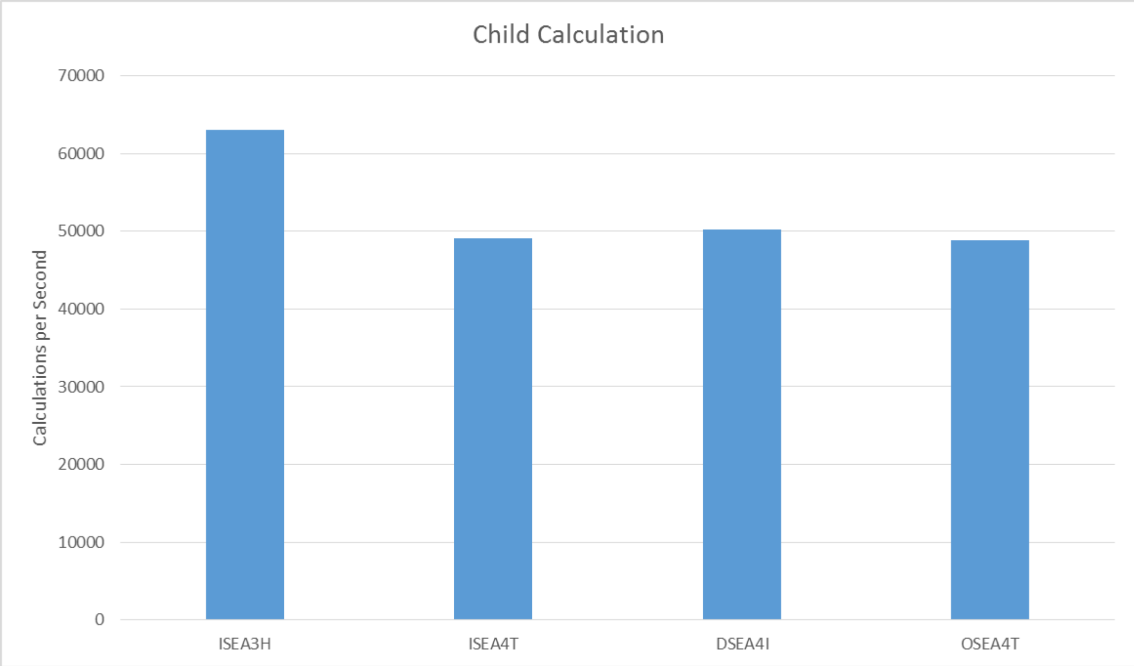

Figure 28 – Performance of child queries

14 Conclusions

The performance evaluation has shown that the ISEA3H prototype provides the best performance in terms of conversion between latitude/longitude points and DGGS cells (129000/252000 conversions per second in the forward/reverse directions). The DSEA4I prototype is the next best closely followed by the ISEA4T and OSEA4T implementations. The offset coordinate system lends itself well to DGGS cell indexing since the processing time is independent of the resolution level.

In the evaluation of the parent querying, the ISEA3H prototype performed worse than the other systems (141000 calculations per second compared to ~260000 calculations per second) due to the fact that all of the other systems use hierarchical grid indexing which makes determination of the parent cells trivial. For child queries, the ISEA3H prototype performs best (63000 calculations per second) with the other systems following close behind.

In terms of conversion error, the ISEA3H and ISEA4T prototypes perform best (59% average percentage accuracy error) with the DSEA4I and OSEA4T implementations following close behind (68% average percentage accuracy error). The best performing system varies at each accuracy level indicating that the performance of the DGGS is very much dependent on the accuracy at which the points are specified. There may be a benefit in further investigation at other stated accuracies.

The prototypes all have a similar average accuracy error despite the ISEA4T implementation appearing to have a slightly higher error than the others. There is no reason to think that this implementation should differ from the other prototypes that use an aperture 4 triangle grid, so this is more likely to be an artefact of the range of stated accuracies that were used in the evaluation.

15 Recommendations

Based on the results of the prototype evaluation, the best candidate for adoption in terms of both performance and accuracy is the ISEA3H implementation (with the novel offset coordinate grid indexing developed for the prototyping phase). It is significantly faster than the other prototype implementations in terms of conversion and child queries. The performance of determining parent cells is not as high as the other prototypes, but unless it is anticipated that this will be used extensively by client applications, advantages in the other areas should outweigh this disadvantage.

Another factor to consider for the ISEA3H implementation is that the results of parent/child queries will put some onus on the client application to interpret the output from the library. Since the ISEA3H system uses hexagonal grids (which are not congruent) there is not a one-to-one mapping between parent and child cells. Every DGGS cell will have seven child cells, but six of those are not wholly contained within the parent cell. These six cells will also be children of other DGGS cells, so if a set of cells are queried for their children then some child cells may be repeated as they have multiple parents. Similarly when a DGGS cell is queried for its parent cells there may be one or three parents. The OpenEAGGR library will return all three parents to the client application which will need to process this information.

16 Bibliography

[1] Southern Terra Cognita Company, “What is a Discrete Global Grid (DGG)?,” [Online]. Available: http://discreteglobalgrids.org/new-page-2/.

[2] Wikipedia, “Platonic Solid,” [Online]. Available: http://en.wikipedia.org/wiki/Platonic_solid.

[3] J. P. Snyder, “An Equal-Area Map Projection for Polyhedral Globes,” *CARTOGRAPHICA Vol 29 No 1,* pp. 10-21, Spring 1992.

[4] Wikipedia, “Truncated Icosahedron,” [Online]. Available: http://en.wikipedia.org/wiki/Truncated_icosahedron.

[5] K. Sahr and D. White, “Discrete Global Grid Systems,” *Computing Science and Statistics,* no. 30, 1998.

[6] Wikipedia, “Lambert azimuthal equal-area projection,” [Online]. Available: http://en.wikipedia.org/wiki/Lambert_azimuthal_equal-area_projection.

[7] Progonos Consultoria, “Polyhedral Maps,” [Online]. Available: http://www.progonos.com/furuti/MapProj/Normal/ProjPoly/projPoly.html.

[8] J. Baia, X. Zhao and J. Chenb, “Indexing of the Discrete Global Grid using Linear Quadtree,” *ISPRS Workshop on Service and Application of Spatial Data Infrastructure.*

[9] P. Z. Kunszt, A. S. Szalay and A. R. Thakar, “The Hierarchical Triangular Mesh,” Johns Hopkins University, Baltimore.

[10] Red Blob Games, “Hexagonal Grids,” [Online]. Available: http://www.redblobgames.com/grids/hexagons/.

[11] PYXIS innovation inc., “PYXIS WorldView,” [Online]. Available: http://www.pyxisinnovation.com/.

[12] P. Peterson, “Close-packed, uniformly adjacent multiresolutional, overlapping spatial data ordering”. United States of America Patent US 8400451 B2, 1 August 2003.

[13] NASA Jet Propulsion Laboratory, “HEALPix,” [Online]. Available: http://healpix.jpl.nasa.gov/index.shtml.

[14] Landcare Research, “SCENZ-Grid,” [Online]. Available: https://www.seegrid.csiro.au/wiki/SCENZGrid/WebHome.

[15] Southern Terra Cognita Laboratory, “DGGRID,” [Online]. Available: http://discreteglobalgrids.org/software/.

[16] G. Evenden, “Proj.4,” [Online]. Available: http://trac.osgeo.org/proj/.

[17] National Center for Geographic Information and Analysis (NCGIA), “Spherekit,” [Online]. Available: http://www.ncgia.ucsb.edu/pubs/spherekit/.

[18] Los Alamos Software, “SCRIP,” [Online]. Available: http://oceans11.lanl.gov/trac/SCRIP.

Riskaware Ltd
Colston Tower
Colston Street
Bristol
BS1 4XE
+44 117 929 1058
www.riskaware.co.uk