Practical approach for modifying existing geocoding system from equal angular to equal area

Pavel PETROV[1]

[1] University of Economics, Varna, Bulgaria
petrov@ue-varna.bg

Abstract. Geocoding systems are essential for the rapid processing of large volumes of spatial data. There are numerous geocoding systems, each of which divides the space into subspaces with a unique geometric shape and employs a different approach of creating its unique identifier - the geocode. The article's main goal is to demonstrate how existing equal angular geocoding system can be modified to become equal areas geocoding system. The research focuses on the main characteristics of equal angular geocoding systems. Although there are many geocoding systems available today, the study is limited to Geohash, one of the equal angular geocoding systems widely used in practice. We propose a novel approach called Hashgeo that has many practical implications. Many advantages (practical implications) of using cells with equal area have been identified, including: facilitating data comparison between cells of different latitudes, simplifying the process of visualizing data on an online map when creating so-called "heat maps", facilitating statistical analysis of data, preserving the "units per area" indicator when summarizing and aggregating data in a larger scale from data for smaller cells and others. The areas of the spherical segments and the cells within them are calculated using a suitable derived formula.

Key words: geocoding, geographic information systems, equal area geocoding, Geohash, Hashgeo.
Received: 19.10.2023.
Revised: 14.11.2023
Accepted: 15.11.2023
Published: 29.12.2023

1. Introduction

Many devices support global or regional navigation satellite systems, like GPS (USA), Galileo (EU), BeiDou (China), GLONASS (RF), QZSS (Japan), or NavIC (India), and can easily provide geolocation coordinates. Automatic coordinate acquisition makes it easier to measure a variety of parameters, such as motion, speed, atmospheric parameters, etc., and it makes it simple to quickly link events to exact locations. The volume of data related to geographical locations is constantly growing, owing to the widespread use of smartphones and other mobile devices with built-in GPS receivers, as well as the prevalence of vehicles with automatic navigation, drones, IoT devices, and other technologies. GPS navigation, geotagging, geomarketing, and geofencing are becoming increasingly popular. These tools, devices, and services generate massive amounts of data that must be efficiently and timely processed in order to improve decision support.

Efficient processing of rapidly changing large volumes of geospatial data is a complex scientific challenge that necessitates solutions that are in line with current information technology. It necessitates a reconsideration of how data is transformed, stored, and processed in order to effectively meet specific needs. Geocoding systems are critical in the analysis of large amounts of geospatial data.

Different geocoding systems divide space into subspaces with different geometric shapes (triangles, quadrilaterals, or hexagons are the most common). Moreover, various systems are employed to generate distinct identifiers, commonly known as geocodes. The depth of recursive space subdivision, space properties (such as area and dimensions), the number of subspaces into which a space is directly divided, and other factors are also different among these systems. These distinctions have implications for the efficiency of geocode storage, processing, and transmission, as well as the convenience of data aggregation, statistical processing and analysis, visualization, and modelling.

Geocoding systems are categorized based on distinct characteristics, with recent years witnessing the establishment of Discrete Global Grid Systems (DGGS) as an abstract concept. DGGS adheres to specific rules

during recursive tessellation, a process involving the division or filling of space using geometric figures, typically identical regular polygons. This method, often referred to as tessellation, aims to eliminate overlaps and empty spaces (Okabe 2017; Hrach 2023).

In DGGS development, a foundational requirement involves using a geometric body with uniform walls as the basis for projecting geographic maps to generate the grid. Recursive tessellation then structures spaces and their corresponding subspaces hierarchically. Each space can, in turn, act as a subspace within a higher-level space. Essential operations within this geocoding approach include establishing relationships between spaces and subspaces, as well as determining nearest neighbouring spaces.

The "Goodchild Criteria," introduced in 1990, play a pivotal role in DGGS evolution (Kimerling et al. 1999; Goodchild & Kimerling 2002; Gibb 2016; Wang et al. 2021). Currently, the Open Geospatial Consortium (OGC) and working groups within the International Organization for Standardization are significantly involved in advancing DGGS (Gibb 2022).

2. Literature review

Geocoding is a fundamental technology for efficiently indexing spatial data. Geocoding, in its broadest sense, includes three primary components. It begins with the input data on which the encoding process is performed. Second, it includes the method (algorithm) for performing the encoding. Finally, it refers to the data structure of the resulting code, which can be numerical values for rapid automated processing or alphanumeric strings for simple human use in everyday life.

Geographic coordinates, usually latitude and longitude, are used as the input data in most geocoding systems. Classical address geocoding or geoparsing is frequently required before working with building addresses. The process of translating geographic names into geographic coordinates is known as geoparsing (Aldana-Bobadilla et al. 2020; Tateosian et al. 2017; Avvenuti et al. 2018; Fernandes et al. 2021). It is important to consider that geoparsing works with unstructured data, which is frequently textual, whereas geocoding primarily works with structured information (Horak et al. 2011). This distinction highlights the possibility of mistakes that come with using language as humans (Liu et al. 2022).

In a broader sense, a "geocode" is a geographic code, typically alphanumeric, that uniquely and unambiguously identifies a specific geographic location. The geographical location may or may not have a predetermined area, dimensions, or shape. Geocodes intended for human use in everyday life should be concise and simple to understand.

Today, geocodes are widely used in a variety of applications, including vehicle registration numbers that indicate the city of registration, postal codes, and addresses. Geocoding is the process of working with various types of geocodes. It can occur in two directions: forward and reverse geocoding. Forward geocoding, for example, determines that the city of Varna corresponds to postal code 9000, whereas reverse geocoding associates postal code 9000 with the city of Varna.

Within a more restricted context, certain sources employ the word "geocoding" interchangeably with the more focused idea of "address geocoding." We believe that this usage is deceptive and may lead to a misunderstanding of terminology. Geocoding, for example, is described as follows in Google's developer documentation (Google 2023): "The process of converting addresses (e.g., '1600 Amphitheater Parkway, Mountain View, CA') into geographic coordinates (e.g., latitude 37,423021 and longitude -122,083739) that can be used to place markers or positions on a map." The process of translating geographic coordinates into an address that can be read by humans is known as reverse geocoding." Here, "geocoding" refers only to using building addresses and Google Maps; it does not refer to the wider definition of "geocoding."

"Geocoding is the process of transforming a description of a location - such as a pair of coordinates, an address, or a name of a place - to a location on the earth's surface" says Environmental Systems Research Institute, Inc. (ESRI) documentation for ArcGIS Desktop. The locations that result are represented as geographic features with attributes that can be used for mapping and spatial analysis" (ArcMap 2023). In a nutshell, "geocoding" is "the assignment of a code, typically numerical, to a geographic location" (Krieger et al. 2004). In summary, geocoding is a programming technique that converts two sets of geographical coordinates into a short character string.

Geographic Information Systems (GIS) specify locations (Vasilev et al. 2022) in a variety of formats. The most basic format is Degrees, Minutes, Seconds (DMS): Latitude and longitude are expressed as three integers in degrees, minutes, and seconds, with the latter having a fractional part. One nautical mile (1 NM or 1852 meters) is equivalent to one minute of terrestrial meridian, and one second from the meridian is approximately 30.87 meters. These can be represented in two ways: latitude followed by longitude (navigation system) and longitude followed by latitude (Cartesian system).

Other formats are as follows WKT (Well-Known Text): A proposal by the Open Geospatial Consortium that has been incorporated into international standards (ISO 2016; ISO 2019); KML/KMZ (Keyhole Markup Language/Keyhole Markup Zipped): Initially developed for Google Earth and subsequently specified by the OGC (OGC 2015; Google 2023); GeoJSON: Defined in RFC 7946 by the IETF (IETF 2016); GPX (GPS Exchange Format): Commonly used in Garmin devices (Topografix 2004); MGRS: Employed in NATO; Plus Codes: Proposed and used by Google; Geohash: Proposed by G. Niemeyer from Canonical Ltd. (Ubuntu). Furthermore, data can be specified in universal unstructured formats, such as CSV (defined by IETF) (IETF 2005) or TSV (specified by IANA) (IANA 1993).

A practical and open-source tool for converting between different file formats is ogr2ogr (https://gdal.org/programs/ogr2ogr.html), developed by the GDAL (Geospatial Data Abstraction Library) project, with funding from the NumFOCUS organization. Convenience, performance, legacy considerations, and other factors are important to consider when selecting a format. A qualitative comparison of a few location specification formats used in GIS is given in Table 1, considering factors like size, footprint, and decoding ease.

Table 1.

Comparison of popular geolocation formats used in GIS

Format	Size	Footprint	Ease of Decoding
DMS	Expressed in degrees (°), minutes ('), and seconds (")	Can be relatively large in terms of characters, when specifying coordinates with high precision.	Human-readable and understandable for people familiar with geographic coordinates.
WKT	Variable, but typically concise for simple geometries	Compact for representing basic geometries	Readable and relatively easy for humans to understand. Widely supported by GIS software for parsing.
KML/ KMZ	XML-based, can vary widely depending on the complexity of the data	XML is verbose and can result in larger file sizes for complex data	KML is human-readable and can be understood visually. KMZ is a compressed format that can be easily opened in compatible software.
GeoJSON	JSON-based, compact for representing basic geometries	Compact and efficient for web-based applications	Human-readable and easy for software to parse, making it popular for web mapping
GPX	XML-based, variable depending on the number of waypoints, tracks, and routes	XML format can be verbose for large datasets	Human-readable XML structure, widely used for GPS data interchange
MGRS	Typically a combination of UTM coordinates and grid zone designators	Larger footprint in terms of characters compared to latitude and longitude	Requires knowledge of MGRS grid system and may involve parsing algorithms
Plus Codes	compact alphanumeric code	compact and designed for easy communication	Designed for simplicity and user-friendliness. Specialized decoding tools and services are available.
Geohash	Variable length depending on desired precision	Compact for representing locations but can get longer with higher precision	Requires specialized decoding algorithms

Source: Own elaboration

The Geohash geocoding system converts a latitude and longitude pair into a single alphanumeric character string to encode geographic coordinates. Each character in the Geohash string after that improves the accuracy of the location. This Geohash string can be used as a unique identifier for spatial indexing and quick lookups. There is also an online system for converting geographic coordinates into a Geohash code (as shown in fig. 1).

Geohash encoding is currently implemented in various relational database management systems and NoSQL type systems, including PostgreSQL (PostGIS 2023), MySQL (MySQL 2023), Redis (Redis 2023-b), MongoDB (MongoDB 2023), Lucene (Apache Software Foundation 2016), Elasticsearch (Elasticsearch 2023), Solr (Smiley et al. 2015), Cassandra (Shah & Chaudhary 2018), Memcached (Ni et al. 2020), CouchBase (Couchbase 2023), HBase (He et al. 2017; Van & Takasu 2018), DynamoDB (Jawarneh et al. 2020), and others.

Numerical values on the map are divided in half during the division process. For example, averaging (0 + 90) and dividing by 2 yields the latitude of 45° for the first horizontal line. This method of dividing the map into separate cells is based on simple arithmetic operations and yields cells with different areas and sizes. A single cell has the capacity to divide multiple times, giving rise to as many as 32 smaller cells (fig. 2). The division pattern of Geohash coding alternates, which makes it unique. The following division consists of 8 rows and 4 columns if a division of 4 rows and 8 columns is utilized.

Figure 1. Geocoding online service geohash.org with a Geohash code in the URL
Source: http://geohash.org/ry7cx4tp95

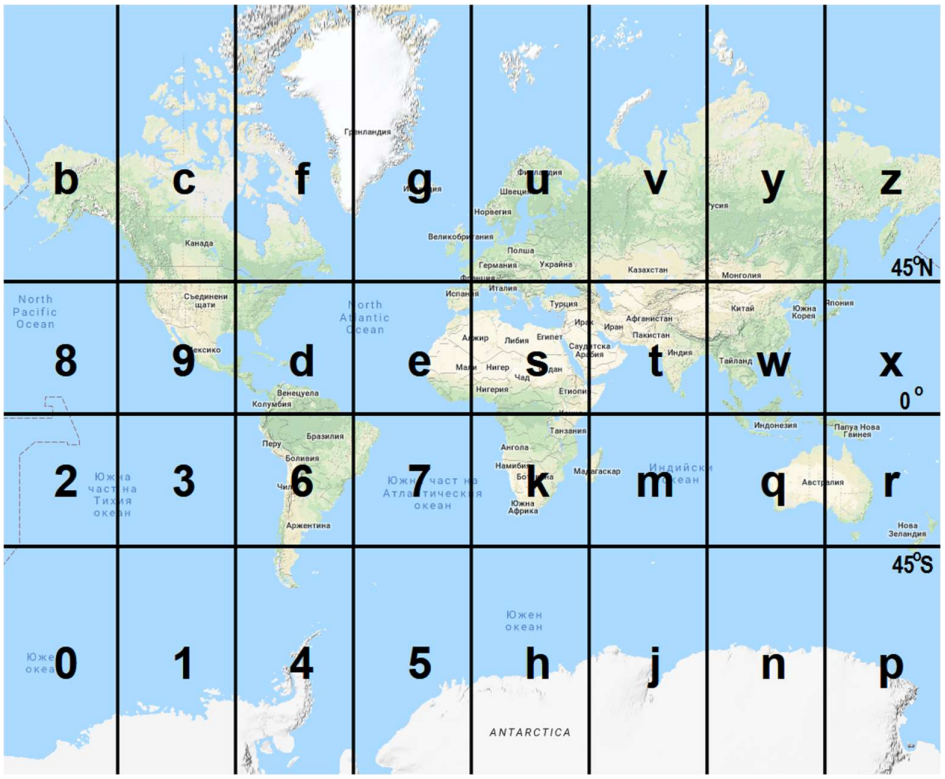

Figure 2. The map of the world divided on 32 cells at the Geohash's first level of precision
Source: Petrov et al. 2018

Each cell division into smaller cells adds a new character to the Geohash string. The accuracy of the location is directly impacted by the string's character count. The effect of removing characters from the end of the string is the same as removing digits from the end of the latitude and longitude numbers. The use of a Geohash code allows for the quick retrieval of objects that are nearby. These objects typically share common Geohash string prefixes, allowing for efficient querying via SQL statements. Consider the following query: "SELECT * FROM table WHERE geohash LIKE 'sxevg%';" retrieves objects located in the city of Varna (without peripheral districts), and the request: "SELECT * FROM table WHERE geohash LIKE 'sxevg3%';" extracts objects located around the UE-Varna building.

Following the widespread adoption of Geohash coding, information surfaced about the Quadkeys system (https://learn.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system), which operates in a similar manner (fig. 3). However, based on the algorithm description, it appears to be more closely related to the Quadtree algorithm (Samet 1984; Spann & Wilson 1985; Tobler & Chen 1986; Shaffer & Samet 1987).

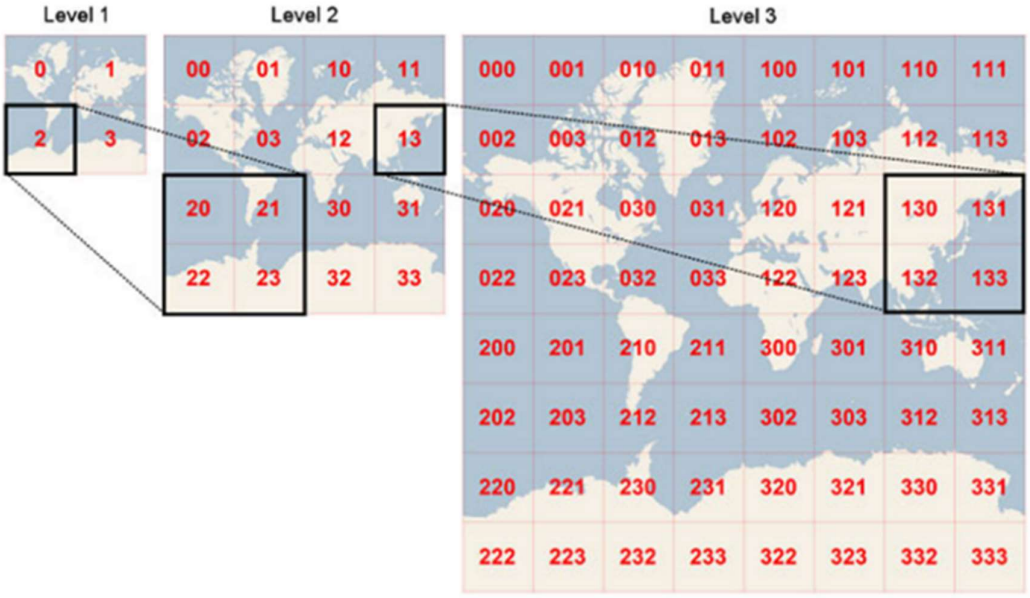

Figure 3. Visual representation of the tessellation process with the Quadkeys system
Source: Schwartz 2022

The primary similarity to Geohash coding is the practical alternation of latitude and longitude bits as well as z-curve traversal. The key difference between Geohash and Quadkeys geocode is the use of Base-4 for the geocode, specifically the digits "0123", and this system is primarily intended for specialists to use.

Furthermore, a similar system to Bing Maps' Quadkeys has been developed for Google Maps, known as Open Location Code/Plus Codes (OLC/PC). OLC/PC is a geocoding system that is based on quadrilateral spaces (Cantarero et al. 2020; Mai et al. 2022). The grid is divided into 9 rows and 18 columns at the first level, with each cell measuring 20 by 20 degrees. The following four recursive divisions of each space are 20 by 20 subspaces, i.e. the spaces on the individual levels have dimensions of 20°x20°, 1°x1°, 0,05°x0,05°, 0,0025°x0,0025°, and 0,000125°x0,000125°, with the areas of the spaces from one level located at different latitudes not being equal. The dimensions of the spaces near the equator are approximately (assuming that 1° = 111,321 km at the equator) 2226x2226 km, 111x111 km, 5,6x5,6 km, 278x278 m, and 13,9x13,9 km, respectively. The following divisions are 5 by 4 subspaces (dimensions near the equator: 3,5x2,8 m, 87x56 cm, and so on).

The location is encoded in an alphanumeric string using twenty characters from the set "23456789CFGHJMPQRVWX". The letters are all capitalized. To avoid confusion or mispronunciation, numbers and letters that are easily confused with others are excluded, as in other systems. The geocode begins with four pairs of symbols, the first of which represents latitude and the second of which represents longitude. Similar to how separators are used when writing telephone numbers, the "+" symbol is placed after the first eight characters to aid in the visual perception of the separate parts of the geocode. The "+" symbol is followed by another pair, followed by a series of only one symbol each, corresponding to a grid of 5 by 4 subspaces (Rinckes & Bunge 2022). This geocode has an interesting feature in that it can be shortened when combined with a predefined reference location. This makes the code section shorter, more usable, and easier to remember. In the case of short codes, the first 2 to 6 characters are removed, and the symbol "+" must be included; the following characters cannot be removed. A country, city, or other location name should also be included.

Since 2018, the organization "Addressing the Unaddressed" (https://www.addressingtheunaddressed.org/) has used OLC/PC in the placement of signs with building addresses in India, particularly in Calcutta (Wiebe, 2018; Mason, 2020; Mask, 2020). Although OLC/PC is licensed as free software, it can only be used in the online mapping service Google Maps.

Combining different map projections and grids can be used in a number of ways to translate a three-dimensional location on a spherical body to a two-dimensional point on a map. For instance, the Mercator projection is frequently used in navigation along with a roughly square grid. In addition to the obvious benefit of making course and other angles easy to determine, this projection has the serious drawback of significantly distorting the figures' sizes as they approach the poles.

On a regional map covering a country or a water basin, this distortion is usually not visible and therefore not necessary. When the entire Earth is covered, these distortions are easily noticeable. For this reason, Google Maps, a well-known online map service, has been displaying a 3D globe at large scale instead of a small-scale Mercator projection map since 2018. In order to address these issues, data analysis calls for the application of a set of coefficients connected to particular grid subspaces.

A geocode can be linked to a wide variety of data in the database, and all data associated with a given geocode that corresponds to a specific geographic space or neighbouring spaces can then be quickly retrieved. Small objects can be associated with larger spaces where they fit perfectly, just as large objects can be associated with several smaller spaces where they fit perfectly.

For ease of comparison, it is crucial that the spaces have the same area for multiple applications. For instance, there is a significant ease of calculation for the "number of objects per unit area" indicator, which forms the foundation of the so-called "heat maps". Owing to the peculiarities of the reference geometric bodies - the sphere, ellipsoid, and flattened ellipsoid, that are used to approximate the precise shape of the Earth (the geoid), it is difficult to obtain spaces in the grid that have the same dimensions and proper shape.

The grid is frequently formed during the rasterization process by spherical arcs spaced at equal angular distances in latitude and longitude. This causes disproportions as latitude increases - the spaces, which are square or rectangular at the equator, gradually change to trapezoidal in the middle latitudes and triangular at the poles, where all the meridians meet.

In contrast to the square, rectangle, or trapezoid, which have four neighbours on the sides and four on the vertices, the pole triangle has three neighbours on the sides, two neighbours on the vertices facing the equator, and numerous neighbours on the vertex at the pole. Due to these complications, some systems (such as Redis or Quadkeys) limit the latitude to about 85 degrees, avoiding the triangular shape of cells near the poles.

3. State of the problem

One major problem with current equiangular tessellation methods is the accuracy of a cell's predetermined height, measured in centimetres, meters, or kilometres, in relation to its width. If the Earth is a sphere, for example, the width in meters at the bottom edge of the cell "9" covering 45°x45° from the equator equals the height, but it is less than the width at the top edge in Geohash coding.

This discrepancy arises because the meridians converge at the poles, causing the latitude in meters at the top of cell "u" (and all cells on the same row) to approach zero. Assuming the Earth is a sphere with a radius of 6371 km, with satisfactory accuracy, we can state that: 1° at the equator equals 111,2 km for both the height and width of a cell; 1° along the 22,5° parallel is 102,5 km (2 * π * [6371 km * cos(22,5°)] / 360°) for cell width; at 45° – 78,6 km, and at 67.5° – 43,5 km for cell width.

With this data, the bottom and height of the "9" cell are 5003,8 km (45° * 111,2 km). However, the width of the top side of the "9" cell is 3538,2 km (45° * 78,6 km).

The width of the bottom of the "c" cell matches the width of the top side of the "9" cell - also 3538,2 km. The height is the same as the height of cell "9" - 5003,8 km, nonetheless, the span of the top side of cell "c" is zero. In contrast to the "9" cell, which is a spherical quadrilateral (with four 90° angles), a "c" cell is a spherical triangle (with two 90° angles at the base and a third 45° angle at the apex). This may not be visible on maps using projections such as the Google Web Mercator.

While dividing the map into subspaces by halving numbers involves simple arithmetic operations, each cell at precision level one has 45° width and 45° height dimensions. However, the lengths of the horizontal sides of cells in different rows are not equal when measured in meters, kilometres, miles, and so on.

Not only are the sides unequal, but the cells are also different in size. The "9" cell, for example, has a surface area of approximately 22,5 million square kilometres, whereas the "c" cell has a surface area of only 9,3 million square kilometres. According to the calculation formulas provided by (Donaldson & Siegel 2001), the area of the "9" cell is approximately 2,4 times that of the "c" cell (fig. 4, fig. 5).

Figure 4. The surface areas disproportion in cells located on latitudes 0°-45° and 45°-90°
Source: Petrov et al. 2018

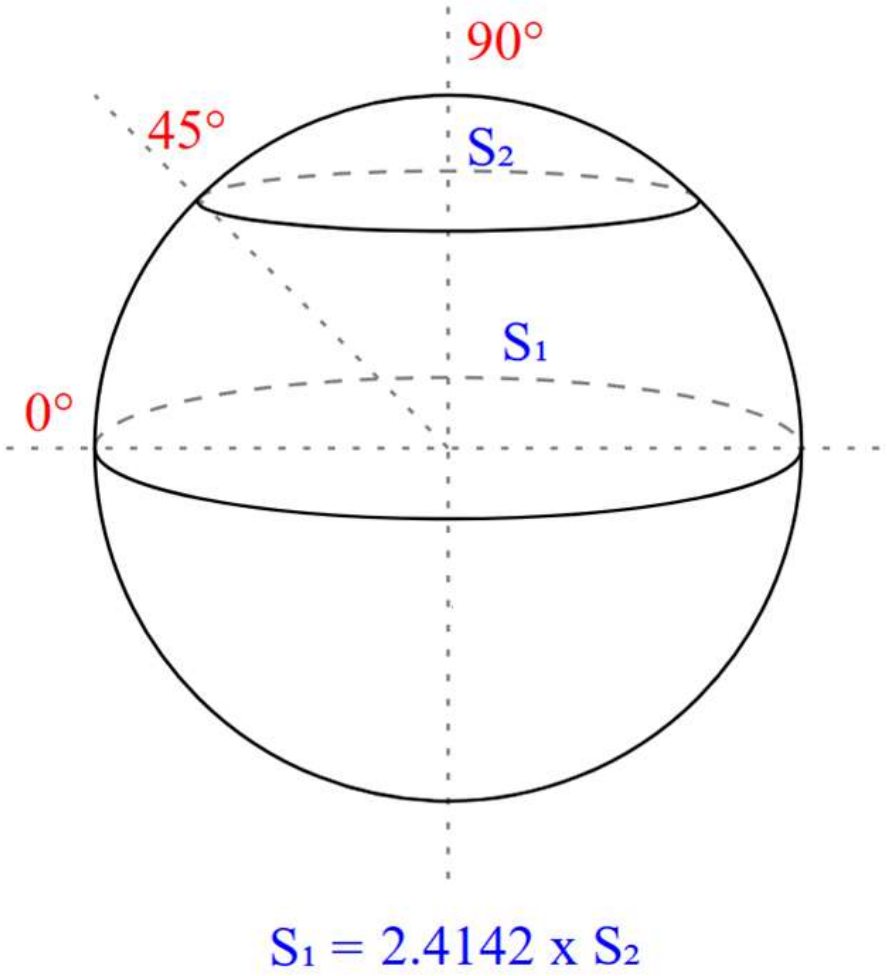

Figure 5. Difference in areas between spherical segments 0°-45° N. and 45°-90° N
Source: Own elaboration

Essentially, different rows of cells have different areas because latitude is represented by mechanically dividing angles into equal parts. As Figure 4 and 5 illustrates, this disparity is already noticeable in the northern hemisphere's original division.

The upper row's area accounts for approximately 29,3% of the total area of the northern hemisphere, while the lower row's area accounts for approximately 70,7% of the total area. Refer to (Weisstein 2022) for a detailed mathematical explanation of the formulas used in calculations. It is important to note that when using the traditional Mercator projection for navigation, this difference in area is visually inverted. While this projection distorts areas, it effectively maintains object angles.

The cells in the top row are actually spherical triangles, with vertical sides convergent at a single point at the top of the triangle. Furthermore, latitudes above 83° are not displayed in Figure 4 because doing so would cause the areas in the top row cells' visual representation to be even more distorted. The division is shown on a sphere in Figure 5, which makes the variations in areas clearly visible.

4. Study of the differences in the areas of cells of the same accuracy level in Geohash coding

In the calculations, we use the basic formula for the area of a spherical segment:

$A = 2 * \pi * r^2 * (1 - \sin(\text{latitude}))$,
where r is the radius of the Earth.

At precision level one (the first character of the Geohash code), every single cell in the bottom row is about 2,4 times bigger than every other cell in the top row. At the second level of accuracy (the first two characters of the Geohash code) the division is 8 rows by 4 columns, and when dividing each cell into 32 subspaces, this difference in areas between the cells of the second level of accuracy increases (fig. 6 and fig. 7). In fig. 6 the top eighth row of cells of the second precision level is not shown - i.e. above 84,4°. It is well known that the Mercator projection, as a result of the distortions, is not ideal for accurately representing geographical sizes and shapes, particularly in areas far from the equator. The areas surrounding the poles are infinitely stretched, resulting in exaggerated sizes and shapes. The poles themselves are depicted as lines when they are, in fact, points. The distortions are easily visible on fig.6.

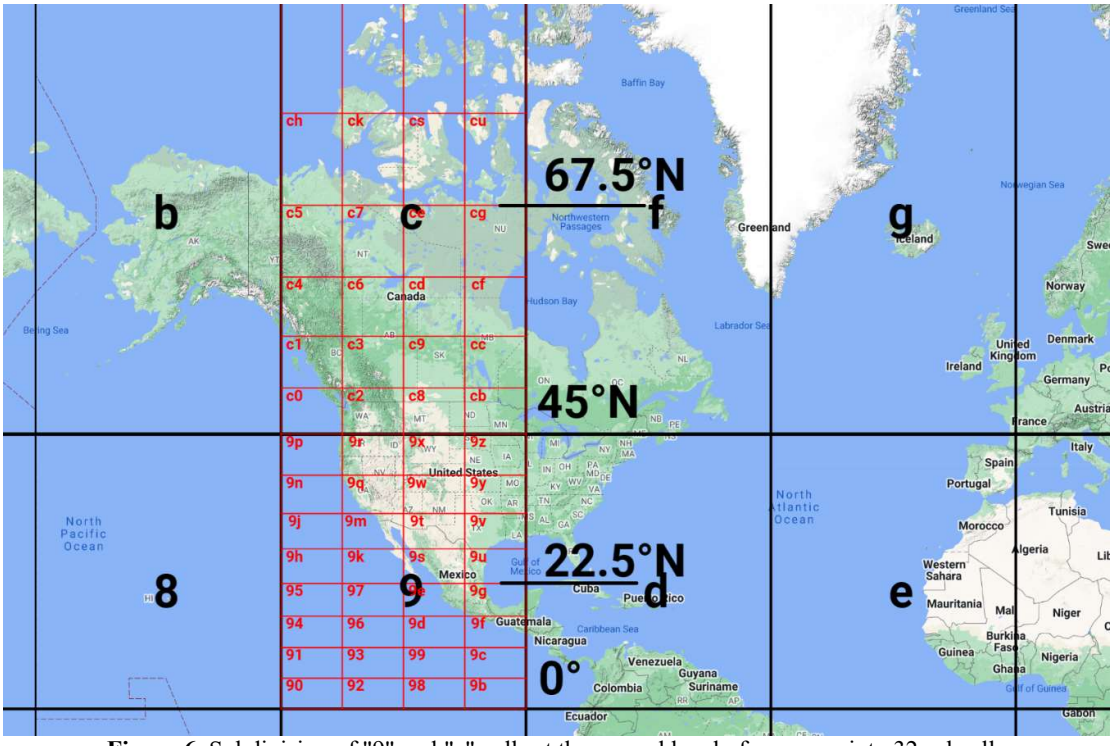

Figure 6. Subdivision of "9" and "c" cells at the second level of accuracy into 32 subcells
Source: Own elaboration, based on the Web Mercator projection from Google Maps

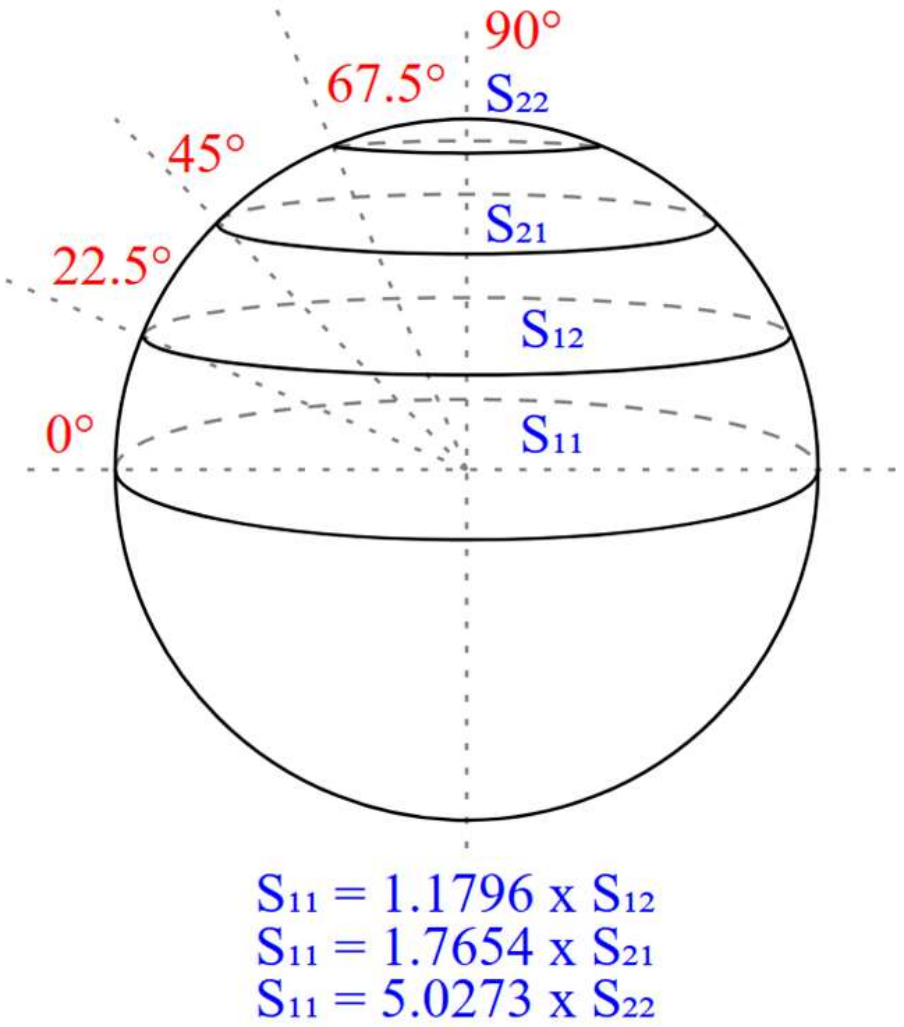

Figure 7. Area difference between spherical segments 0°-22.5° N, 22.5°-45° N, 45°-67.5° N and 67.5°-90° N
Source: Own elaboration

As a result of calculations for the individual spherical segments, the following is obtained:

Area_share_S_{11}(0°, 22.5°) ≈ 0.38268 ≈ 38.3%
Area_share _S_{12}(22.5°, 45°) ≈ 0.32442 ≈ 32.4%
Area_share _S_{21}(45°, 67.5°) ≈ 0.21677 ≈ 21.7%
Area_share _S_{21}(67.5°, 90°) ≈ 0.07612 ≈ 7.61%

The data are presented in fig. 8, from which we can conclude that in the equiangular tessellation, the closer the spherical segment is to the pole, the more its area decreases compared to the spherical segments that are located closer to the equator. For example, the area of the second accuracy level cell "9b" (the same area as "90", "92" and "98"), in the Pacific Ocean, is 2,9 times the area of the cell of the same precision level "ck" (the same area as "ch", "cs" and "cu") where the part of the Victoria Island, Canada is located.

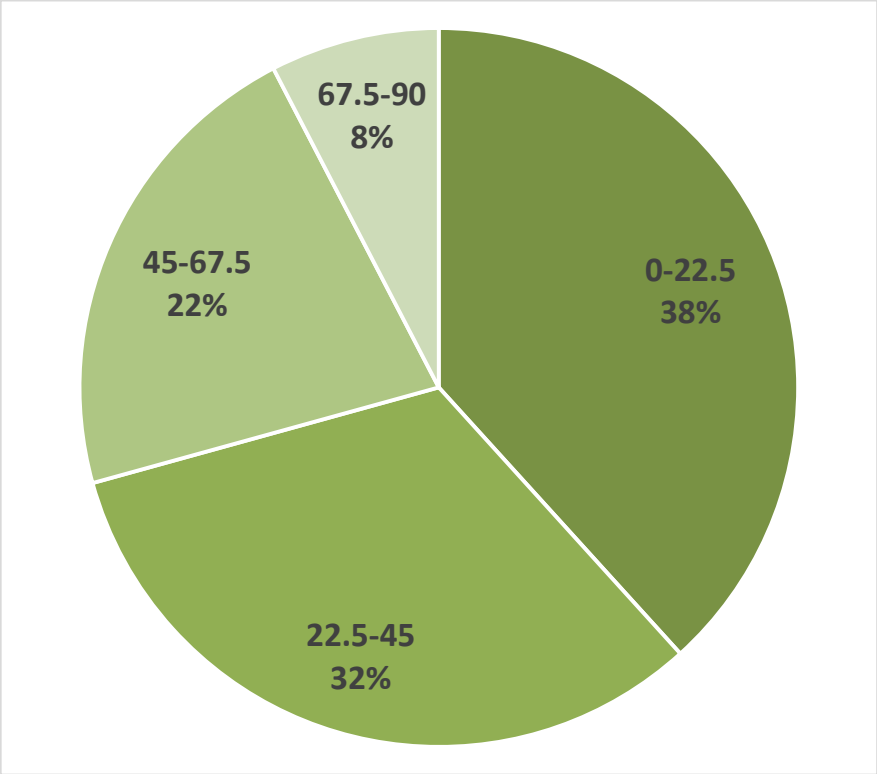

Figure 8. Shares of the areas of the four spherical segments 0°-22.5°, 22.5°-45°, 45°-67.5° and 67.5°-90° in the hemisphere in the equiangular tessellation
Source: Own elaboration

We can summarize that all subsequent divisions inherit disproportion in the surface areas of the parent cells and the tendency is for the disparity to increase. In Table 2 the areas of cells of different rows at the first and second level of accuracy in the Geohash code are presented.

Table 2. Bounding boxes around specific Geohash cells at first and second level of precision

Cell	West side	East side	South side	North side	Area of the cell (in thousands sq. km)
9	135° W	90° W	0°	45° N	22 542
c			45° N	90° N	9 337
9b	101,25° W	90° W	0°	5,63° N	781
9c			5,63° N	11,25° N	774
9f			11,25° N	16,88° N	759
9g			16,88° N	22,5° N	736
9u			22,5° N	28,13° N	707
9v			28,13° N	33,75° N	671
9y			33,75° N	39,38° N	628
9z			39,38° N	45° N	580
cb			45° N	50,63° N	525
cc			50,63° N	56,25° N	466
cf			56,25° N	61,88° N	402
cg			61,88° N	67,5° N	334
cu			67,5° N	73,13° N	263
cv			73,13° N	78,75° N	190
cy			78,75° N	84,38° N	115
cz			84,38° N	90° N	38

Source: Own elaboration

In Table 3, the ratio between the areas of cells, of the same level of accuracy, but located at different latitudes, is calculated. The base (unit) is the area of a cell bordering the equator at its lower end, i.e., latitude 0°. The next three cells border at their lower end the parallels of latitudes 22.5°, 45° and 67.5°. The last cell is bounded at its upper end by the parallel, which degenerates from a line or circle into a point, in latitude 90°. Calculations are made for different levels of accuracy - from 1 to 8.

Table 3. Ratio between the areas of cells located at different latitudes, when changing the level of accuracy

Level	Geohash codes of cells	Width in degrees	Height in degrees	Ratio
1	s u	45	45	1 0.414214
2	s0 sh u0 uh uz	11.25	5.625	1 0.90508 0.672369 0.337296 0.049127
3	s00 sh0 u00 uh0 uzz	1.40625	1.40625	1 0.919183 0.698429 0.371345 0.012272
4	s000 sh00 u000 uh00 uzzz	0.351563	0.175781	1 0.923293 0.706022 0.381266 0.001534
5	s0000 sh000 u0000 uh000 uzzzz	0.043945	0.043945	1 0.923733 0.706836 0.382329 0.000383
6	s00000 sh0000 u00000 uh0000 uzzzzz	0.010986	0.005493	1 0.923861 0.707073 0.382639 0.000048
7	s000000 sh00000 u000000 uh00000 uzzzzzz	0.001373	0.001373	1 0.923875 0.707098 0.382672 0.000012
8	s0000000 sh000000 u0000000 uh000000 uzzzzzzz	0.000343	0.000172	1 0.923879 0.707106 0.382682 0.000001

Source: Own elaboration

From Table 3 at the same level of accuracy the cells bordering the equator are many times larger than those bordering the pole. For example, at accuracy level 7, the difference is more than 83 thousand times, i.e., cell "s0000000" is about 83443 times larger in area than cell "uzzzzzz". At accuracy level 8, the difference is over 667 thousand times, at 9 – over 2,6 million times, at 10 – over 21 million times, at 11 – over 105 million times. For the cells bordering the 22,5°, 45°, and 67,5° parallels, the area differences gradually stabilize with increasing the accuracy.

As can be seen from the Table 2 and Table 3, the areas of cells of the same level of accuracy in Geohash coding vary widely, which creates great inconvenience in solving some applied problems.

Based on the provided data, switching to a grid of cells with equal areas, in our opinion, has the following main advantages:

1) Cells of equal area make it easier to compare data between cells located at different latitudes. For example, when studying population density, one can easily compare the density of this indicator in one cell with another cell, if the areas are of the same and previously known size.

2) Equal cell areas can simplify the process of data visualization on an online map when creating the so-called "heat map". If the cells are not the same in area, it may be more difficult to accurately represent the data on the map and may distort the visual representation of the data.

3) Cells with equal areas can facilitate statistical analysis of the data. For example, if the mean or median of a set of geospatial data is being calculated, cells of equal area help the statistical calculations to be more accurate.

4) Cells with equal areas can facilitate the implementation of algorithms that rely on equal areas. For example, some machine learning algorithms would perform better when the geospatial data is arranged in cells with equal areas.

5) When summarizing and aggregating data in a larger plan from data for smaller sized cells, the coefficient "indicator per unit area" is preserved - e.g., number of patients per square kilometre, number of vehicles per unit area, number of traffic accidents per unit area, etc.

5. Modified Geohash system with equal areas cells – Hashgeo system

Varying cell areas is sometimes inconvenient and undesirable, for example when analysing data by the number of objects per unit area. In these cases, the area of all cells of the same accuracy level must be the same. This is not achievable with the classical Geohash system, and therefore we propose to change it so that all cells of one accuracy level have the same area (Petrov et al. 2018). Additionally, in subsequent divisions, each cell of the next precision level has an area exactly 1/32 of the area of the parent cell.

To distinguish this system from the classical Geohash system, we use the name Hashgeo and the geocode is prepended with "#" (called hash, sharp, pound, hex, octothorp, etc.). For the conversion from geographic coordinates to Hashgeo to be as fast as possible, the calculations should be as simple as possible. The working algorithm is as follows:

1) There is no change in how the longitude of cell boundaries is calculated, i.e. vertical cell borders remain at the same longitude.

2) Calculation of latitude of cell borders is changed. In order to keep the area equal, a more complicated formula must be applied instead of splitting into equal angular portions. We obtained the following formula after transforming the spherical segment formula:

cell_latitude_boundary = arcsin(2 * n / N)

cell_latitude_boundary is the latitude of the cell boundary that is closer to the equator, i.e. the northern boundary in the southern hemisphere, the southern boundary in the northern hemisphere. For the latitude boundary of the opposite cell's side (i.e. near to the pole), it is necessary to calculate the latitude boundary of the next cell towards the pole. We use the fact that both cells have a common border.

N – total number of cells counted vertically. It relates on the level of accuracy i.e., of the string length: 1 character – 4 cells, 2 characters – 4*8 = 32 cells, 3 characters – 32*4=128 cells, 4 characters – 128*8=1024, etc.

n – the cell number, which is determined by counting beginning at 0 from the equator. In the northern hemisphere, the row number is positive. In the southern hemisphere, it is negative.

The spherical segments' areas are the same when employing this method (see fig. 9 and fig. 10), and since there are the same number of cells in each spherical segment (because the number of columns is fixed), their areas are also the same.

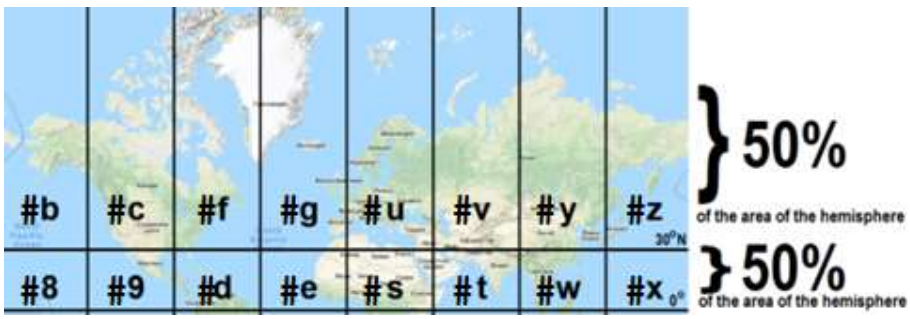

Figure 9. Areas of the two spherical segments separated by 30° parallel when using Hashgeo instead of 45° as in the classic Geohash system
Source: Own elaboration

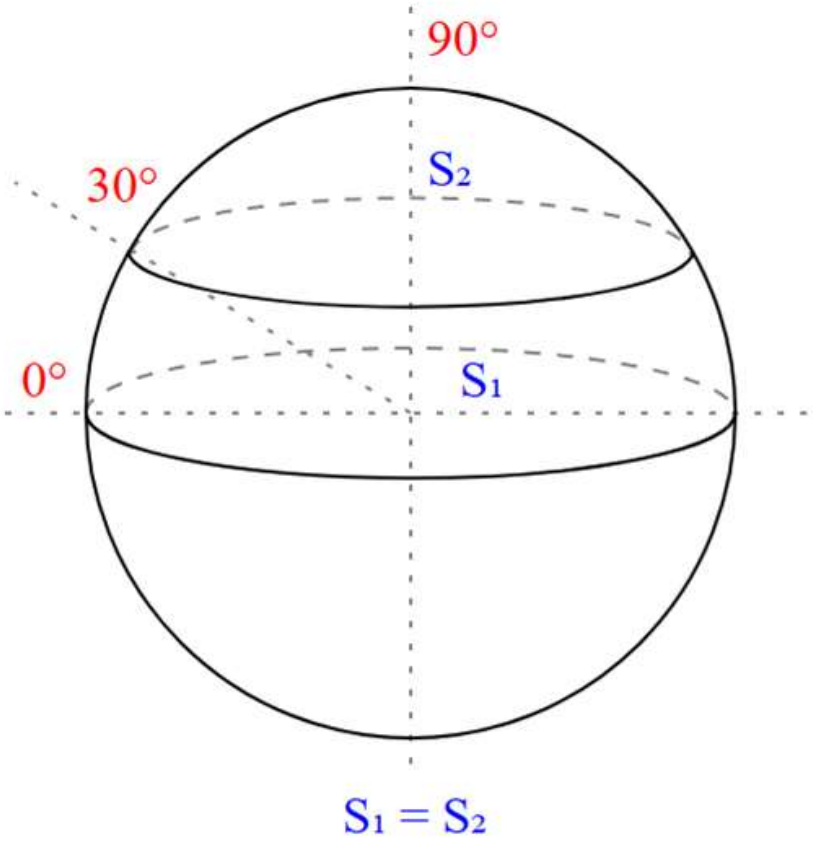

Figure 10. Area difference between spherical segments 0°-30° and 30°-90° in Hashgeo
Source: Own elaboration

To illustrate the operation of the algorithm, we have developed two examples with different levels of accuracy.

A. The calculation of the boundary latitudes at the first level of accuracy of cell "#6" covering part of the South America (fig. 11), using the Hashgeo system, is as follows:

1) For the boundary near the equator: N = 4 (since the string length is 1 character), n = 0, therefore:

cell_latitude_boundary1 = arcsin(2 * 0 / 4) = arcsin(0) = 0°.

2) For the boundary near the pole (in this case – the South Pole) the latitude of the next cell located towards the pole – cell "#h" must be calculated: N = 4, n = -1 (since it is in the southern hemisphere), therefore:

cell_latitude_boundary2 = arcsin(2 * -1 / 4) = arcsin(-0,5) = -30° = 30°S.

B. The boundary latitude computation of a cell "#ch" encompassing a part of British Columbia, Canada at the second accuracy level (fig. 12), using the Hashgeo system is as follows:

1) For the border to the equator: N = 32 (since the string length is 2 characters), n = +12 (since it is in the northern hemisphere and the row counting starts from 0), therefore:

cell_latitude_boundary1 = arcsin(2 * 12 / 32) = arcsin(0,75) = 48,59° = 48,59°N.

2) For the latitude of the border to the pole (in this case – the North Pole), the latitude of the next cell in the direction towards the pole – cell "#cj" must be found N = 32, n = +13, therefore:

cell_latitude_boundary2 = arcsin(2 * 13 / 32) = arcsin(0.8125) = 54.34° = 54.34°N.

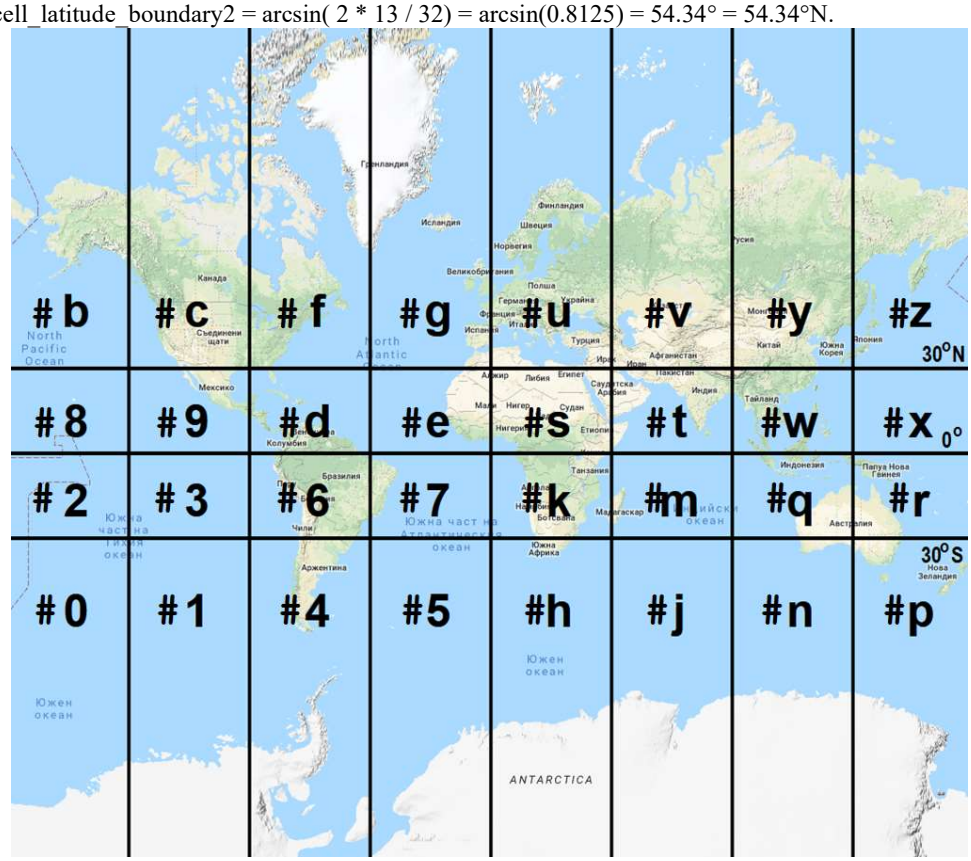

Figure 11. The map of the world divided on 32 cells at the Hashgeo's first level of precision
Source: Own elaboration

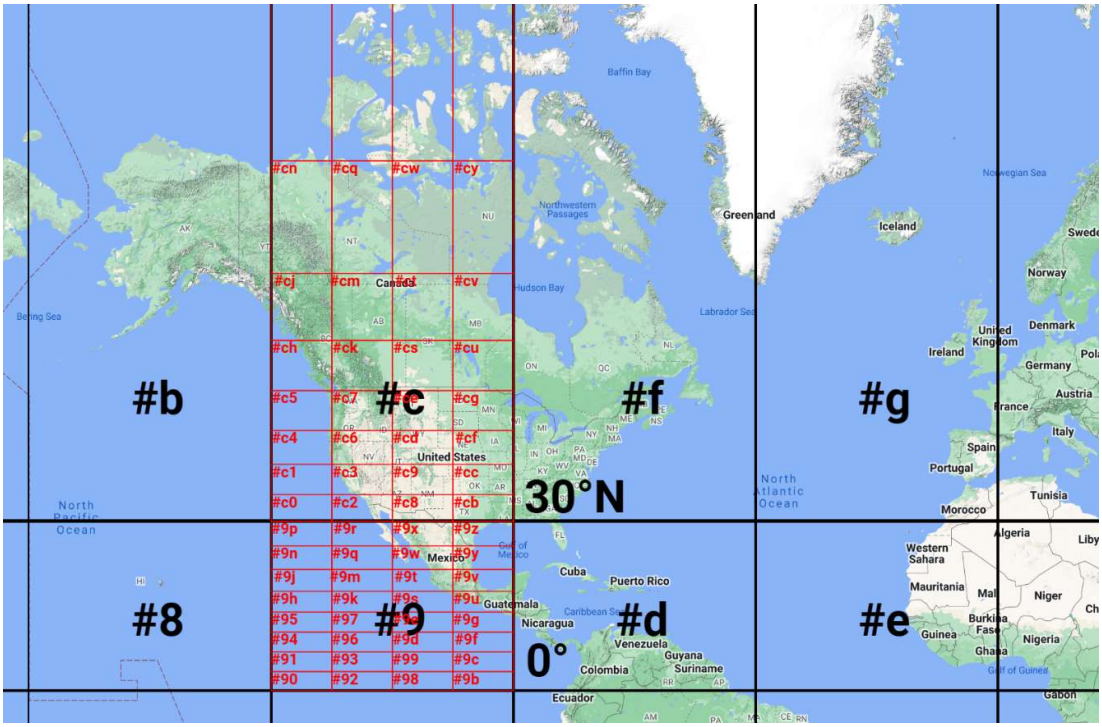

Figure 12. The Hashgeo system was used to divide the "#9" and "#c" cells into 32 subcells.
Source: Own elaboration

In an analogous way, the horizontal borders of the cells can be calculated at other levels of accuracy to ensure that all cells have the same area.

The subcells coordinates in the last column of Hashgeo cells "#9" and "#c", using the described computational method, are displayed in the Table 4.

Table 4. Bounding boxes around specific Hashgeo cells at first and second level of precision

Cell	West side	East side	South side	North side	Area of the cell (in thousands sq. km)
#9	135° W	90° W	0°	30° N	15 940
#c			30° N	90° N	15 940
#9b	101,25° W	90° W	0° N	3,58° N	498
#9c			3,58° N	7,18° N	498
#9f			7,18° N	10,81° N	498
#9g			10,81° N	14,48° N	498
#9u			14,48° N	18,21° N	498
#9v			18,21° N	22,02° N	498
#9y			22,02° N	25,94° N	498
#9z			25,94° N	30° N	498
#cb			30° N	34,23° N	498
#cc			34,23° N	38,68° N	498
#cf			38,68° N	43,43° N	498
#cg			43,43° N	48,59° N	498
#cu			48,59° N	54,34° N	498
#cv			54,34° N	61,05° N	498
#cy			61,05° N	69,64° N	498
#cz			69,64° N	90° N	498

Source: Own elaboration

The formula derived here uses the arcsine trigonometric function, which is relatively slow to execute, especially if there is no built-in support in the processor.

In order to speed up the computational operations when processing the so-called "big data" it is possible, when implementing the algorithm proposed here, to perform only one-time tessellation for equal areas at the beginning, and then to use equiangular tessellation. In this approach, at the beginning when the 0°-90° interval is divided into two parts, instead of 45°, the initial division is at 30°, and subsequently the 0°-30° and 30°-90° intervals are divided in half, i.e. values 15° and 60°.

Unfortunately, while the difference between the areas of the spherical segments locked between 0°-15° and 15°-30° is not significant – the area of the first segment is only 7.3% greater than the area of the second, the difference between the areas of the spherical segments locked between 30°-60° and 60°-90° is substantial. The area of the first segment is over 2.7 times that of the second (fig. 13).

Moreover, it turns out that the area of the spherical segment locked between 30°-60° is the largest and is 41.4% larger than the area of the 0°-15° segment and 51.8% larger than the area of the segment 15°-30°. The fraction of the areas of the four spherical segments relative to the area of the hemisphere are as follows (Hemisphere fraction 0°-90° = 100%):

Share of spherical segment 0°-15° ≈ 25.9%
Share of spherical segment 15°-30° ≈ 24.1%
Share of spherical segment 30°-60° ≈ 36.6%
Share of spherical segment 60°-90° ≈ 13.4%

Because of these large differences in areas, we consider it appropriate to apply the equal areas approach consistently for all levels of accuracy, and not just up to a certain level of accuracy, i.e., in the particular case, the 30°-90° interval needs to be divided into 48.6°, not 60° (fig. 14).

In this situation, since the calculation of arcsine can be a computationally intensive task, it is possible to apply two approaches in case of the need to speed up data processing processes:

1) Applying a caching approach (with or without precomputation). The already calculated values are stored in the cache and thus finding the corresponding latitudes is significantly faster.

2) Applying an approach to approximate arcsine calculations.

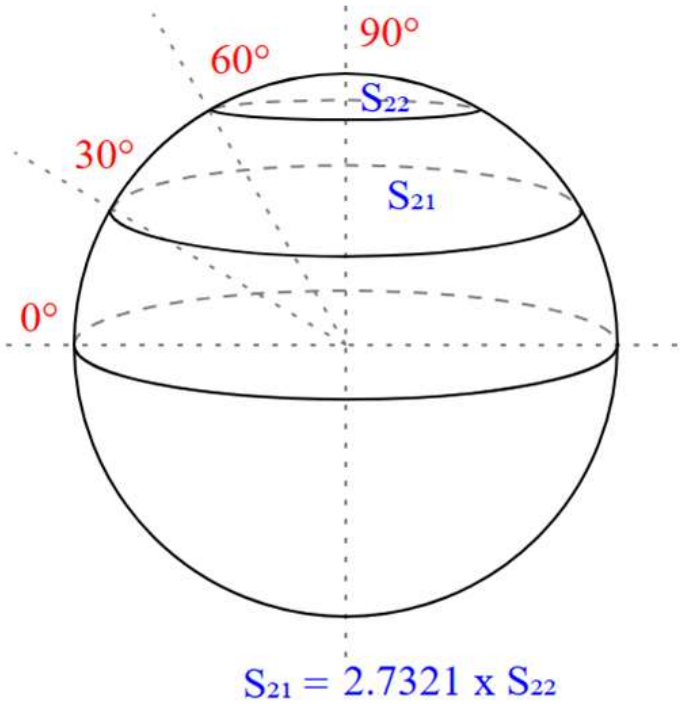

Figure 13. Area difference between spherical segments 30°-60° and 60°-90°
Source: Own elaboration

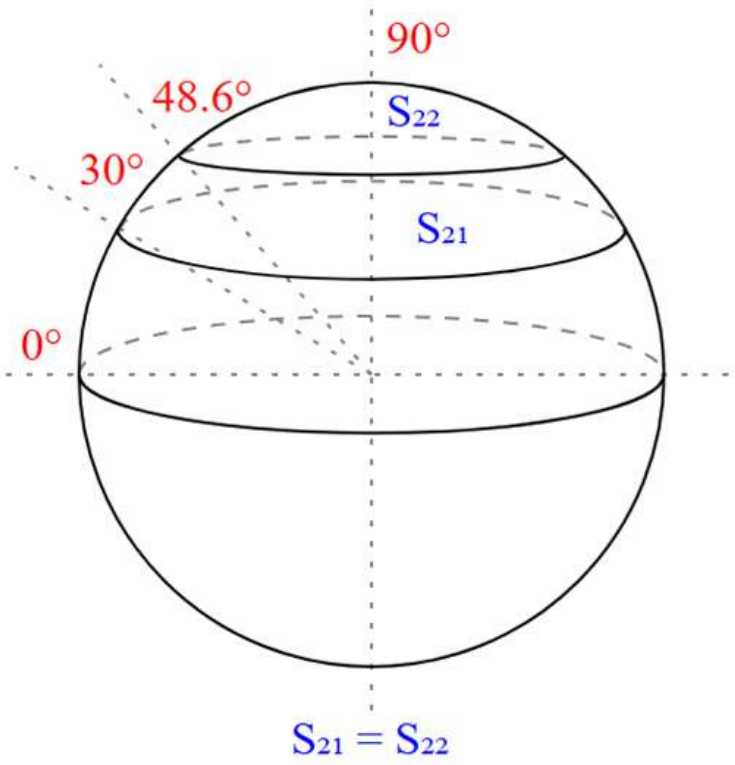

Figure 14. Area difference between spherical segments 30°-48.6° and 48.6°-90°
Source: Own elaboration

About the last option, approximate arcsine formulas could be used, in situation where the computational expense and complexity associated with calculating exact arcsine values should be small, i.e. in resource-constrained environments or real-time applications. Approximate arcsine formulas provide a practical solution by providing sufficiently accurate arcsine estimations while significantly reducing computational overhead. Approximation using polynomial expansions or rational approximations of the arcsine function within specific ranges of input values is one of the widely acclaimed approaches (fig. 15).

Function:

$$\arcsin X = \frac{\pi}{2} - \sqrt{1 - X}\,\Psi(X)$$

Range:

$$0 \leq X \leq 1$$

Approximation:

$$\Psi^*(X) = a_0 + a_1 X + a_2 X^2 + a_3 X^3$$

$$a_0 = 1.5707{,}288 \qquad a_2 = .0742{,}610$$

$$a_1 = -.2121{,}144 \qquad a_3 = -.0187{,}293$$

Figure 15. Formula for approximate arcsine calculations
Source: Hastings et al. 2015

One important advantage of these approximation formulas is that they are effective at increasing processing speed. Complex algorithms and series expansions are used in traditional methods of calculating arcsine, which require a significant amount of computational power. On the other hand, approximate formulas provide more straightforward mathematical operations, which makes them especially useful for applications that need quick calculations, like high-frequency trading in finance or signal processing in real-time systems.

Moreover, these approximation formulas considerably lessen the computational load while maintaining accuracy within predetermined ranges. They achieve a balance between computational cost and accuracy, enabling quick computations with adequate accuracy. This balance is especially useful for applications like scientific simulations, game development, and navigation systems that call for iterative calculations or real-time responses.

Another appealing feature of approximate arcsine calculations is their ability to work around hardware and software limitations. Using complex trigonometric functions may be impractical in situations where computational power or memory resources are limited. Approximate formulas provide a viable alternative by enabling efficient computations on devices with limited processing capabilities, thereby broadening the scope of applications to a variety of platforms. Furthermore, the simplicity and ease with which these approximate arcsine formulas can be implemented make them accessible to a broader audience, including programmers, engineers, and researchers. Their simple mathematical expressions and low complexity contribute to their widespread use in a variety of computational domains, allowing for rapid prototyping and development. However, it is important to note that the use of approximate arcsine formulas may introduce minor errors, particularly at the function's domain's extremes. While these errors may be acceptable in many applications, precision-critical scenarios may require the use of more accurate but computationally expensive methods.

6. Use of equal area division for data visualization

In various systems handling geospatial data analysis and visualization, there's a demand for uniformity in areas. This requirement results from the ease with which data concerning the distribution of measured values per unit area can be analysed within specific cells, as shown in fig. 16. Uniform areas allow for relatively simple data aggregation for computations such as total or average values. Additionally, proportions or percentages based on aggregated totals can be calculated. This consistency aids in the visualization of "heat maps" at various scales.

"Heat maps" are a powerful tool in the realm of geospatial data visualization, allowing for the comprehensive representation of complex information. Colour gradients are used in these visualizations to convey variations in data density across geographic areas. They facilitate the intuitive comprehension of information patterns using this technique, making them a cornerstone in fields ranging from urban planning and environmental studies to business analytics and epidemiology. One of the primary benefits of "heat maps" is their ability to represent large datasets succinctly. Geospatial data, which is frequently vast and intricate, can be dense and difficult to interpret conventionally. "Heat maps" provide a simplified yet detailed depiction of this complexity by transforming numerical data into a colour spectrum. This streamlined representation not only aids in identifying trends and outliers but also supports quick decision-making processes.

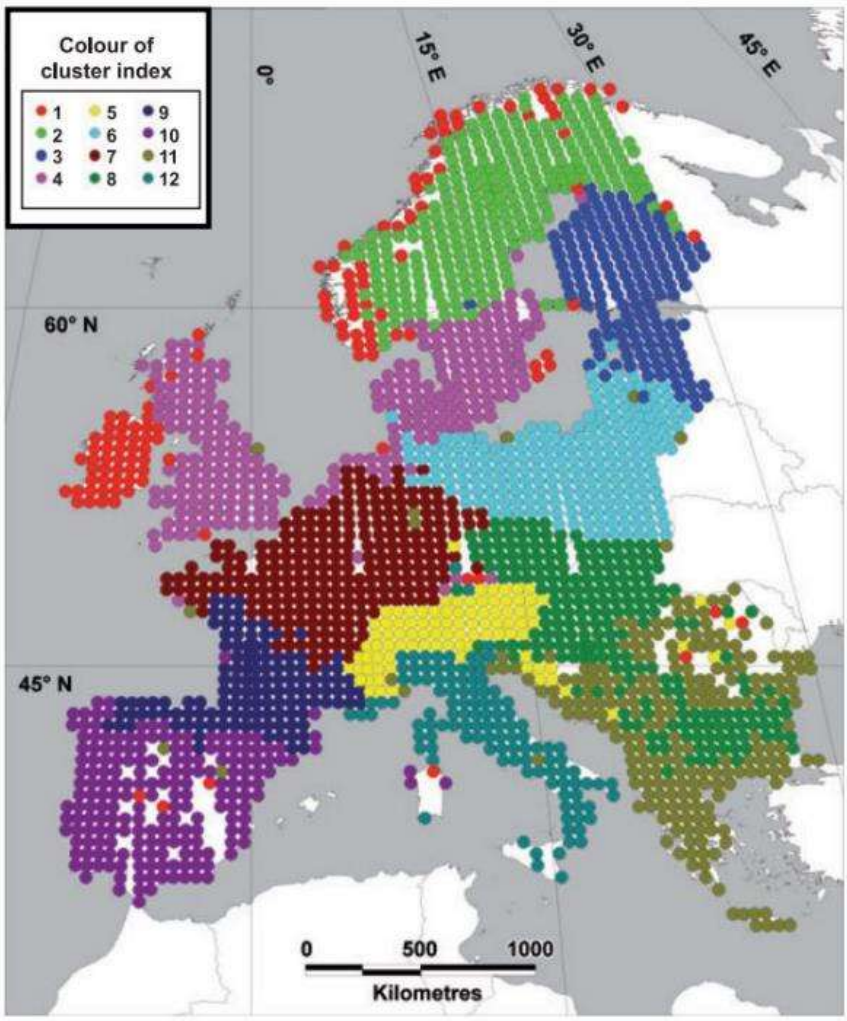

Figure 16. Example of cluster representation of European mammal data using equal area cells
Source: Heikinheimo et al. 2007

Because "heat maps" are inherently visual, they are incredibly approachable. The use of colour gradations in these maps contributes to the enhanced comprehension of visual information that humans are capable of processing. Regardless of their level of technical proficiency, stakeholders can understand spatial distributions and trends thanks to the intuitive nature of colour interpretation, which promotes more informed decision-making. Furthermore, "heat maps" facilitate efficient pattern identification. When data trends are visualized across geographic boundaries, correlations or spatial clusters that may not be visible in raw datasets are frequently found. Stakeholders can gain important insights by recognizing these patterns, which can be applied to anything from market analysis to epidemiology to understand consumer behaviour and identify disease outbreaks.

Furthermore, the adaptability of "heat maps" makes them a useful tool. These visualizations can support a wide range of data types and formats, allowing for seamless integration with various analytical methods. "Heat maps" provide a versatile platform for displaying diverse spatial attributes such as population density, traffic patterns, environmental changes, and real estate values. "Heat maps" are also a useful communication tool. Presenting complex spatial data to a variety of audiences, such as policymakers, researchers, or the general public, necessitates clarity and accessibility. These visualizations effectively communicate insights and findings through vivid and concise representations, fostering a shared understanding among stakeholders.

Notwithstanding their many advantages, it's critical to be aware of any possible drawbacks, such as the possibility of oversimplification or incorrect data interpretation. To guarantee that information is accurate and appropriately represented, caution should be used. To sum up, "heat maps" are an invaluable tool for the analysis

and visualization of spatial data. Decision-makers in a variety of fields are empowered by their capacity to condense complex data into understandable and perceptive representations, which promotes a deeper comprehension of spatial phenomena and motivates wise action.

7. Conclusion

In conclusion, the method for dividing the earth globe into horizontal segments (rows) implemented in the classical Geohash geocoding is easy and quick, but subsequently, the areas of the individual spherical segments (the rows in the grid) are totally different. The disparity, for example, gets up to 2,9 times in the variation in area between cells located close to the Equator and cells located close to the northern parts of Europe or North America, which are relatively densely populated and therefore of interest in terms of geospatial data processing. Furthermore, for the regions around 45° latitude, relative to the regions around the equator, the difference is about 1,3-1,5 times, such that the area of the cells near to the equator is bigger. This is problematic in some circumstances, and all cells from the same level of accuracy must have the same area.

The proposed approach for dividing the map into separate cells of equal area involves the use of the arcsine trigonometric function in the calculation of latitude boundaries. We call it Hashgeo because it generates cells with different locations and sizes than the original Geohash system. The area of every geocoded cell with a single symbol is the same, and around 15 939 515 sq. km. The cells geocoded with 2 symbols also have the same area – precisely 1/32 of the area of the parent cell which is geocoded with 1 symbol – 498 110 sq. km, etc. Thus, the areas of geocoded cells with 3, 4, 5, and more characters (i.e. the length of the geocoded string) can be readily determined by dividing by 32.

As directions for future work, we suggest that our approach for modifying existing geocoding system from equal angular to equal area could be applied to many other equal angular geocoding systems. For example, Quadkeys, Open Location Code/Plus Codes, Natural Area Code (NAC), etc. In this way geocoding systems will be more useful for data visualization, assuring equal area representation for better spatial analysis. For example, creating grids for ecological studies, biodiversity mapping, or natural resource management, census purposes, demographic analyses, climate modelling and analysis, climate change impacts, etc. where accurate area representation is crucial. For the benefit of urban planners, equal area zoning maps can help with land use assessments, infrastructure design, and resource distribution in an equitable manner. For disaster management and emergency response equal area planning can ensure fair resource allocation and coverage. For healthcare equal area planning can ensure fair access to medical facilities and resources across different regions. These and other potential prospects have yet to be explored.

Literature

Aldana-Bobadilla, E., Molina-Villegas, A., Lopez-Arevalo, I., Reyes-Palacios, S., Muniz-Sanchez, V., Arreola-Trapala, J. (2020). Adaptive Geoparsing Method for Toponym Recognition and Resolution in Unstructured Text. *Remote Sensing.* MDPI, Vol. 12, № 18, article № 3041, pp. 1 – 19. Available at: https://doi.org/10.3390/rs12183041.

Apache Software Foundation (2016). Lucene 5.5.0 spatial API. Class GeoHashUtils. Available at: https://lucene.apache.org/core/5_5_0/spatial/org/apache/lucene/spatial/util/GeoHashUtils.html.

ArcMap (2023). What is geocoding? Available at: https://desktop.arcgis.com/en/arcmap/latest/manage-data/geocoding/what-is-geocoding.htm.

Avvenuti, M., Cresci, S., Del Vigna, F., Fagni, T., Tesconi, M. (2018). CrisMap: a big data crisis mapping system based on damage detection and geoparsing. *Information Systems Frontiers.* Springer, Vol. 20, pp. 993 – 1011. Available at: https://doi.org/10.1007/s10796-018-9833-z.

Cantarero Navarro, R., Rubio Ruiz, A., Dorado Chaparro, J., Villanueva Molina, F. J., Santofimia Romero, M. J., Villa Alises, D., Lopez Lopez, J. C. (2020). A proposal for modeling indoor-outdoor spaces through IndoorGML, Open Location Code and OpenStreetMap. *ISPRS International Journal of Geo-Information.* MDPI, Vol. 9, № 3, article № 169, pp. 1 – 21. Available at: https://doi.org/10.3390/ijgi9030169.

Gibb, R. G. (2016). The rHEALPix discrete global grid system. *IOP Conference Series: Earth and Environmental Science.* IOP Publishing, Vol. 34, article № 012012, pp. 1 – 8. Available at: https://doi.org/10.1088/1755-1315/34/1/012012.

Gibb, R. G., Purss, M. B., Sabeur, Z., Strobl, P., Qu, T. (2022). Global reference grids for big earth data. *Big Earth Data.* Taylor & Francis, Vol. 6, article № 3, pp. 251 – 255. Available at: https://doi.org/10.1080/20964471.2022.2113037.

Goodchild, M. F., Kimerling, A. J. (2002). *Discrete global grids: A web book.* Available at: https://escholarship.org/uc/item/9492q6sm.

Couchbase (2023). Geospatial Geopoint Queries. Available at: https://docs.couchbase.com/server/current/fts/fts-supported-queries-geopoint-spatial.html#a-geohash.

Donaldson, S., Siegel, S. (2001). *Successful Software Development*, Prentice Hall Professional, p. 354. Available at: https://books.google.bg/books?id=lrix5MNRiu4C.

Elasticsearch (2023). Geohash grid aggregation. Available at: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-bucket-geohashgrid-aggregation.html.

Fernandes, C., Fernandes, J., Mathew, S., Raorane, S., Srinivasaraghavan, A. (2021). Automated Disaster News Collection Classification and Geoparsing. *Proceedings of the International Conference on Smart Data Intelligence (ICSMDI, 2021)*. Elsevier, article № 3852688, pp. 1 – 11. Available at: http://dx.doi.org/10.2139/ssrn.3852688.

Google, (2023). What is geocoding? Available at: https://developers.google.com/maps/documentation/geocoding/overview.

Hastings, C., Wayward, J., Wong, J. (2015). *Approximations for Digital Computers.* Princeton University Press, p. 159. Available at: https://books.google.bg/books?id=IRTWCgAAQBAJ.

Heikinheimo, H., Fortelius, M., Eronen, J., Mannila, H. (2007). Biogeography of European land mammals shows environmentally distinct and spatially coherent clusters. *Journal of Biogeography.* Wiley, Vol. 34, № 6, pp. 1053 – 1064. Available at: http://www.mgrs-data.org/data/documents/HeikinheimoetalJBI07.pdf.

He, S., Chu, L., Li, X. (2017). Spatial query processing for location based application on Hbase. *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*. IEEE, pp. 110 – 114. Available at: https://doi.org/10.1109/ICBDA.2017.8078787.

Horak, J., Belaj, P., Ivan, I., Nemec, P., Ardielli, J., Ruzicka, J. (2011). Geoparsing of Czech RSS news and evaluation of its spatial distribution. *Semantic Methods for Knowledge Management and Communication. Studies in Computational Intelligence.* Springer, Vol. 381, pp. 353 – 367. Available at: https://doi.org/10.1007/978-3-642-23418-7_31.

Hrach, R. (2023). Plane Tessellation. Learning Mathematics in the Context of 3D Printing. *Proceedings of the International Symposium on 3D Printing in Mathematics Education.* Wiesbaden: Springer, pp. 111 – 132. Available at: https://doi.org/10.1007/978-3-658-38867-6_5.

IANA (1993). Definition of tab-separated-values (TSV). Available at: https://www.iana.org/assignments/media-types/text/tab-separated-values.

IETF (2005). RFC 4180 Common Format and MIME Type for Comma-Separated Values (CSV) Files. Available at: https://www.rfc-editor.org/rfc/rfc4180.

IETF (2016). RFC 7946 The GeoJSON Format. Available at: https://www.rfc-editor.org/rfc/rfc7946.

ISO (2016). ISO/IEC 13249-3:2016 Information technology – Database languages – SQL multimedia and application packages – Part 3: Spatial. Available at: https://www.iso.org/standard/60343.html.

ISO (2019). ISO 19162:2019 Geographic information – Well-known text representation of coordinate reference systems. Available at: https://www.iso.org/standard/76496.html.

Jawarneh, I. M., Bellavista, P., Corradi, A., Foschini, L., Montanari, R. (2020). Efficient QoS-aware spatial join processing for scalable NoSQL storage frameworks. *IEEE Transactions on Network and Service Management.* IEEE, Vol. 18, № 2, pp. 2437 – 2449. Available at: https://doi.org/10.1109/TNSM.2020.3034150.

Kimerling, J. A., Sahr, K., White, D., Song, L. (1999). Comparing geometrical properties of global grids. *Cartography and Geographic Information Science.* Taylor & Francis, Vol. 26, № 4, pp. 271 – 288. Available at: https://doi.org/10.1559/152304099782294186.

Krieger, N., Waterman, P. D., Chen, J. T., Rehkopf, D. H., Subramanian S. V. (2004). *The Public Health Disparities Geocoding Project Monograph*. Available at: http://www.hsph.harvard.edu/thegeocodingproject/geocoding/.

Liu, Z., Janowicz, K., Cai, L., Zhu, R., Mai, G., Shi, M. (2022). Geoparsing: Solved or Biased? An Evaluation of Geographic Biases in Geoparsing. *AGILE: GIScience Series.* Copernicus Publications, Vol. 3, article № 9, pp. 1 – 13. Available at: https://doi.org/10.5194/agile-giss-3-9-2022.

Mai, G., Janowicz, K., Hu, Y., Gao, S., Yan, B., Zhu, R., Cai, L., Lao, N. (2022). A review of location encoding for GeoAI: methods and applications. *International Journal of Geographical Information Science.* Taylor & Francis, Vol. 36, № 4, pp. 639 – 673. Available at: https://doi.org/10.1080/13658816.2021.2004602.

Mask, D. (2020). *The address book: What street addresses reveal about identity, race, wealth, and power.* St. Martin's Press.

Mason, R. (2020). Addressing the unaddressed. *Land Journal.* Royal Institution of Chartered Surveyors (RICS), Issue January/February 2020, p. 30. Available at: https://www.proquest.com/openview/1f5eb9c28c4bc69e284ce25359ee0c4b/.

MongoDB (2023). MongoDB Manual. 2d Index Internals. Available at: https://www.mongodb.com/docs/manual/core/geospatial-indexes/.

MySQL (2023). Reference Manual, Spatial Geohash Functions. Available at: https://dev.mysql.com/doc/refman/8.0/en/spatial-geohash-functions.html.

Ni, L., Liu, Y., Liu, Y. (2020). Privacy protection model for location-based services. *Journal of Information Processing Systems.* Korea Information Processing Society (KIPS), Vol. 16, № 1, pp. 96 – 112. Available at: https://doi.org/10.3745/JIPS.04.0163.

OGC (2015). OGC KML 2.3. Available at: http://docs.opengeospatial.org/is/12-007r2/12-007r2.html.

Okabe, A. (2017). Spatial Tessellations. *International Encyclopedia of Geography: People, the Earth, Environment and Technology* (eds D. Richardson, N. Castree, M.F. Goodchild, A. Kobayashi, W. Liu, R.A. Marston). Wiley, pp. 1 – 11. Available at: https://doi.org/10.1002/9781118786352.wbieg0601.

Petrov, P., Dimitrov, P., Petrova, S. (2018). Geohash-EAS – a Modified Geohash Geocoding System with Equal-Area Spaces. *SGEM 2018 : 18 International Multidisciplinary Scientific Geoconference* : Conference Proceedings. Sofia: STEF92 Technology Ltd., Vol. 18 Informatics, Geoinformatics and Remote Sensing, № 2.2, pp. 187 – 194. Available at: https://doi.org/10.5593/sgem2018/2.2.

PostGIS (2023). ST_GeoHash. Available at: https://postgis.net/docs/ST_GeoHash.html.

Redis (2023-b). GEOHASH Syntax. Available at: https://redis.io/commands/geohash/.

Rinckes, D., Bunge, P. (2022). Open Location Code: An Open Source Standard for Addresses, Independent of Building Numbers And Street Names. Available at: https://github.com/google/open-location-code/blob/main/docs/olc_definition.adoc.

Samet, H. (1984). The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR).* ACM, Vol. 16, № 2, pp. 187 – 260. Available at: https://doi.org/10.1145/356924.356930.

Shaffer, C. A., Samet, H. (1987). Optimal quadtree construction algorithms. *Computer Vision, Graphics, and Image Processing.* Elsevier, Vol. 37, № 3, pp. 402 – 419. Available at: https://doi.org/10.1016/0734-189X(87)90045-4.

Shah, P., Chaudhary, S. (2018). Big data analytics framework for spatial data. *6th International Conference Big Data Analytics (BDA 2018), Lecture Notes in Computer Science.* Springer, Vol. 11297, pp. 250 – 265. Available at: https://doi.org/10.1007/978-3-030-04780-1_17.

Smiley, D., Pugh, E., Parisa, K., Mitchell, M. (2015). *Apache Solr enterprise search server.* Packt Publishing.

Spann, M., Wilson, R. (1985). A quad-tree approach to image segmentation which combines statistical and spatial information. *Pattern Recognition.* Elsevier, Vol. 18, № 3-4, pp. 257 – 269. Available at: https://doi.org/10.1016/0031-3203(85)90051-2.

Tateosian, L., Guenter, R., Yang, Y. P., Ristaino, J. (2017). Tracking 19th century late blight from archival documents using text analytics and geoparsing. *Free and open source software for geospatial (FOSS4G) conference proceedings.* University of Massachusetts, Vol. 17, № 1, article № 17, pp. 146 – 155. Available at: https://doi.org/10.7275/R5J964K5.

Tobler, W., Chen, Z. T. (1986). A quadtree for global information storage. *Geographical Analysis.* Wiley, Vol. 18, № 4, pp. 360 – 371. Available at: https://doi.org/10.1111/j.1538-4632.1986.tb00108.x.

Topografix (2004). GPX 1.1 Schema Documentation. Available at: https://www.topografix.com/GPX/1/1/.

Van Le, H., Takasu, A. (2018). G-HBase: A high performance geographical database based on HBase. *IEICE Transactions on Information and Systems.* Institute of Electronics, Information and Communication Engineers (IEICE), Vol. 101, № 4, pp. 1053 – 1065. Available at: https://doi.org/10.1587/transinf.2017DAP0017.

Vasilev, J., Kehayova-Stoycheva, M., Serbezova, B. (2022). A GIS-Based Approach in Support of Monitoring Sustainable Urban Consumption Variables. Education, Research and Business Technologies. *Proceedings of 20th International Conference on Informatics in Economy (IE, 2021)*, Smart Innovation, Systems and Technologies (SIST). Springer, Vol. 276, pp. 237 – 247. Available at: https://doi.org/10.1007/978-981-16-8866-9_20.

Wang, Z., Zhao, X., Sun, W., Luo, F., Li, Y., Duan, Y. (2021). Correlation analysis and reconstruction of the geometric evaluation indicator system of the discrete global grid. *ISPRS International Journal of Geo-Information. MDPI*, Vol. 10, № 3, article № 115, pp. 1 – 28. Available at: https://doi.org/10.3390/ijgi10030115.

Weisstein, E. (2022). "Spherical Segment." MathWorld – A Wolfram Web Resource. Available at: https://mathworld.wolfram.com/SphericalSegment.html.

Wiebe, K. (2018). Assessing the Utility & Feasibility of Geo-spatial Addressing in Mathare, Nairobi. McGill University. Available at: https://escholarship.mcgill.ca/downloads/6h440s55j.