# Towards Single Value Coordinate System (SVCS) for Earthquake Forecasting Using Single Layer Hierarchical Graph Neuron (SLHGN)

Benny Benyamin Nasution(✉), Rahmat Widia Sembiring, Indra Siregar, Ermyna Seri, and Rina Walmiaty Mardi

Politeknik Negeri Medan, Jalan Almamater 1, Kampus USU, Medan 20155, Indonesia
bennynasution@polmed.ac.id

**Abstract.** The current coordinate system has been the major challenge for the development of earthquake forecasting technology using Single Layer Hierarchical Graph Neuron (SLHGN). First, the accuracy of the longitude value is not distributed equally, and the accuracy gets worse towards the poles. Second, the distance of the same longitude difference varies following the difference of the latitude values. The extreme one is again on the poles, where the longitude value becomes unity. Third, there is no way to have a coordinate of an area. As an alternative the Single Value Coordinate System (SVCS) has been scrutinized and elaborated. The coordinate system treats every area on the earth equally on the equator until the poles. It means that the accuracy is everywhere the same and the calculation of a distance and an area is not dependent on the location (e.g. near the equator, near the North Pole, etc.). At this stage the algorithm for measuring a distance and the conversion from and to the current coordinate system are available. The distance between two locations is directly discovered from the value of the coordinate itself. The coordinate system is fundamentally dedicated to pinpoint an area, not a point. The smaller an area is the more precise the location will be. Using the SVCS, the characteristic of the earth as a spherical shape suits the SLHGN architecture.

**Keywords:** Earthquake forecasting · Single Layer Hierarchical Graph Neuron (SLHGN) · Single Value Coordinate System (SVCS) · Hierarchical Graph Neuron (HGN)

## 1 Introduction

The issues related to forecasting an earthquake are not only about when it will occur and how big the tremor will be, but also about where the epicenter will be and which area and how big the area will suffer from it. The research in earthquake forecasting using Single Layer Hierarchical Graph Neuron (SLHGN) that has been started since two years ago has also been working on those challenging issues. Some promising results have shown that the SLHGN would come up with good results. Therefore, the SLHGN has


Published by Springer Nature Switzerland AG 2021
Y. Murayama et al. (Eds.): ITDRR 2020, IFIP AICT 622, pp. 73–89, 2021.
https://doi.org/10.1007/978-3-030-81469-4_7

been improved to increase the accuracy of its results. Not only the accuracy of the time and the magnitude of the earthquake occurrences, the SLHGN would also attempt to identify the epi centrum and the areas of the impact about several hours prior to the incidence [1–3]. The description of HGN and mHGN can be found in [3–7].

The main barrier why the accuracy of SLHGN is not yet as good as expected, is that the SLHGN has used the current coordinate system that represents a coordinate through longitude and latitude values. Assuming that the earth is a perfect sphere, the value of the longitude would be more and more inaccurate towards the poles, since the longitude value becomes unity. Similarly, the distance between two points of different longitude values but the same latitude value, would become smaller when the latitude value is closer to a pole (90 degree of latitude value).

Furthermore, the ultimate problem that makes SLHGN ineffective when working on earthquake forecasting is that the current coordinate system does not provide a way to represent a coordinate of an area. Originally, an earth coordinate system was proposed for the first time in 1866 [8, 9]. The current coordinate system is only able to represent a coordinate of a point. However, it is very unlikely that an earthquake forecasting system could determine a point on earth where the upcoming earthquake would occur. In fact, most earthquake forecasting techniques measure and calculate the possibility of an earthquake that will hit an area, not a point on earth. Some other researchers have used a grid approach [10].

So, due to some unfortunate characteristics of the current coordinate system, SLHGN has lost some part of its accuracy while recognizing patterns of earthquakes. The first issue is that in most cases the surface of the earth will be transformed into a two dimensional diagram [11]. This transformation would produce two problems for SLHGN: 1) Although in reality they are close to each other, Japan and Hawaii look very far from each other, that is from the east to the west. 2) It is difficult to locate and allocate an observation area close to a pole, because there are a lot of "blank" areas in between.

The second issue is the measurement or the calculation of a distance of two coordinates. There are different formulas (at least two) for calculating a distance. The first one is the formula for the case when both coordinates are in the area around the equator. The other one is the one for the case when the coordinates are far from the equator. This issue raises a number of questions. First, how far from the equator is the formula still valid? Second, which formula will be used, if one coordinate is in the area of the equator and the other one is very far from the equator? Many researchers have tried to figure out all of those problems [11].

The third issue is the mapping of reported earthquake locations into SLHGN architecture. Since the SLHGN needs to be trained with historical data of previous earthquakes, the location of the epicenter and the affected areas need to be stored/taught in the SLHGN architecture. However, it is very complicated to calculate the transformation of longitude and latitude into something like Cartesian coordinates.

These three main issues have been the reason to develop something that can solve those problems, so that the SLHGN would work more accurately and be able to forecast an earthquake.

## 2 Some Issues When Using Current Coordinate System

The idea of developing the Single Value Coordinate System (SVCS) has been triggered by those above-mentioned issues: 1) inaccurate physical location, 2) inaccurate distance calculations, and 3) not suitable for SLHGN architecture. The next sections discuss them.

### 2.1 Inaccurate Physical Location

Physical location is important for SLHGN to forecast earthquakes. The location of an epicenter would be used to be fed to the SLHGN during both training phase and forecasting phase. Depending on the accuracy of the results required, there will be a lot of epicenter locations that need to be elaborated and composed. For instance, for 9X9 grid size 81 epicenter locations would be used within SLHGN architecture.

In order to know the coordinate of an earthquake, the location of an epicenter is normally first calculated through the data from at least four seismograph stations that have received the physical wave of an earthquake across the earth crust [12]. To calculate the longitude and the latitude values of the epicenter the data from the four seismograph stations will then be combined with the longitude and the latitude values of each seismograph station [13]. It is known that using a GPS device, the measurement error of coordinates received is around 10 m. Therefore, if there are at least four seismograph stations that have been observed, then there would be around 40 m of coordinate measurement error. Additionally, the physical wave of an earthquake would travel through the crust and sometimes also through the core of the earth. It is possible that the wave would be distorted or the path were swerved that would affect the traveling time and in the end affect the location accuracy [14, 15]. This is another contributing factor to the epicenter measurement error using the current coordinate system.

### 2.2 Inaccurate Distance Calculations

Another important part for SLHGN architecture is the calculation of a distance between two location points. During a training phase, earthquake data is taken from locations that have regular distances amongst them. Therefore, every distance must be calculated accurately in order to get accurate results. During the real-time earthquake monitoring phase, the data must also be taken from locations with regular distances. So, distance calculation is very important for having accurate results. Unfortunately, the current coordinate system does not offer a fixed formula for calculating a distance. There is a formula for the case when both location coordinates are in the area around the equator only. On the other hand, when the coordinates are far from the equator, another formula must be used. Such an uncertainty would make the distance calculation using the current coordinate system not applicable, because SLHGN will be used for all areas of the earth surface.

Another challenging issue related to distance calculation is the fact that it is not easy to find a targeted location (gained through a distance calculation) in which earthquakes have occurred. Ideally, the whole surface of the earth will be used as locations of the to-be-measured earthquake magnitude.

### 2.3 Not Suitable for SLHGN

Originally, the type of data that can be fed to the architecture of SLHGN could be one-dimensional, two-dimensional, three-dimensional, or other multi-dimensional data. In case of earthquake forecasting, all the stations of earthquake data are located on the surface of the earth. This is another challenge as the earth itself is a three-dimensional shape, the source of data is located only on the surface. It is true that such kind of data on the surface can be mapped into two-dimensional data [9], but the spherical nature of the earth would not allow that. The reason for this is—for instance—that, in a two-dimensional shape Tokyo and Honolulu would look far away separated, but in reality both are close to each other.

For the SLHGN, in order to allow the data to be fed the same way as the real situation of the data, a kind of spherical structure of data needs to be developed within SLHGN. With such a spherical surface all close neighbors will be treated as close neighbors, and vice versa all far neighbors will be treated as far neighbors. In principle, the SLHGN will treat a seismic location the same as where it is located on the earth. This concept is important when working with time-series pattern recognition. The observed earthquake patterns will be verified by the SLHGN in order to forecast dangerous earthquakes several hours earlier. Different to the previous approach using mHGN that observed local areas, the latest approach observes the whole area of the earth in real-time. In this approach, the accuracy is dependent on the number of faces SLHGN covers earth surface, either with: 32 faces, or 122 faces, or 482 faces, and so on. The details about the number of faces are described in the next section.

## 3 Single Value Coordinate System (SVCS)

As already discussed, the development of the Single Value Coordinate System (SVCS) is for solving those above-mentioned issues: 1) inaccurate physical location, 2) inaccurate distance calculations, and 3) not suitable for SLHGN architecture.

The inaccurate physical location and inaccurate distance calculation are fundamentally caused by the usage of the transformation of the physical units of seismic wave and electromagnetic wave to a distance. To avoid such a transformation, SVCS attempts to calculate the location and the distance through elaborating the earth surface directly. To this purpose, the first step is finding the composition of polygons that cover the earth surface, and then determine a polygon as the reference area to other polygons. The most suitable composition of the polygons are the following figure (see Fig. 1).

There are 12 pentagons and 20 hexagons (32 faces) that can evenly cover up a spherical shape of the earth. The following is the composition of the polygons represented in a two-dimensional way (see Fig. 2).

The polygons above are marked with letters from A till P, and from AA till PP. Note that hexagons A and AA are located opposite to each other on the surface of the ball. The same applies for B and BB, C and CC, and so forth. As the length of the edge of all hexagons is the same as the length of the edge of all pentagons, all hexagons and pentagons can be decomposed differently. The following figure shows it (see Fig. 3).

To increase the accuracy of locations and distances, every hexagon can be decomposed into 4 (four) smaller hexagons, and every pentagon can be decomposed into 1

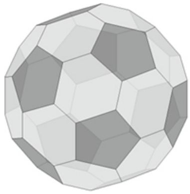

**Fig. 1.** A Football with 12 pentagons and 20 hexagons

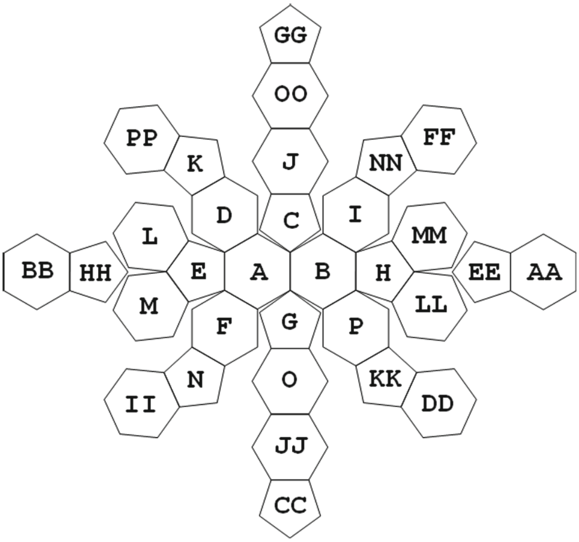


**Fig. 2.** A representation of polygons in a 2-dimensional form

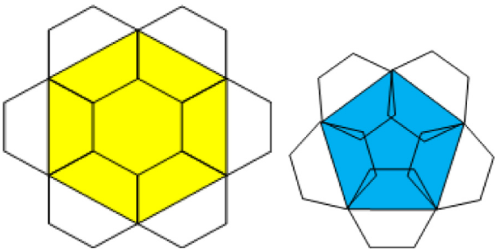

**Fig. 3.** A decomposition of hexagon and pentagon in different form

small pentagon and 2.5 (two point five) smaller hexagons. So, in total there will still be 12 (twelve) pentagons and $12 * 2.5 + 20 * 4 = 110$ hexagons. The number of faces is now 110 hexagons + 12 pentagons = 122 faces. A piece of the ball after the decomposition is like the following (see Fig. 4).

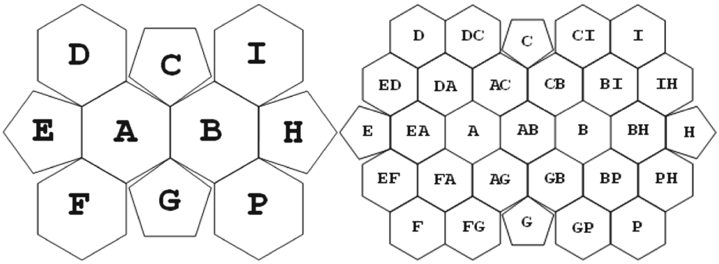


**Fig. 4.** The condition before (left) and after (right) the decomposition

It can be seen that the number of hexagons increases whereas the number of pentagons stays the same. When the same decomposition to all the pentagons and hexagon is implemented recursively, the following shows the changes of the polygons.

```
32      faces: 12 pentagons + 20 hexagons
122     faces: 12 pentagons + 110 hexagons
482     faces: 12 pentagons + 470 hexagons
1922    faces: 12 pentagons + 1910 hexagons
7682    faces: 12 pentagons + 7670 hexagons
30722   faces: 12 pentagons + 30710 hexagons
122882  faces: 12 pentagons + 122870 hexagons
```

The current coordinate system is called Geographic Coordinate Systems. Beside the coordinate system, there is also another thing that is related to the coordinate system called datum. Datum functions as the mechanism to determine the location of a place on the earth surface. There are some other coordinate systems for specific purposes, they are: geodetic coordinate system and plane coordinate system. All those coordinate systems have a general characteristic that they all rely on geometric calculations which require electronic equipment such as satellites and may generate inaccurate results.

On the other hand, the SVCS uses the pentagon G (see Fig. 2) as the reference area. This pentagon is located exactly on the South Pole and one of its corners points to the North Pole through the GMT mark, which is through the zero degree of longitude value. This corner is marked (big dot) and the identity of the corner is a small letter (a) on the pentagon G. All the pentagons and hexagons will be fixed where the mark is located. As already mentioned, in all pentagons the mark denotes the location of the "a" corner (out of a, b, c, d, and e corners), and for hexagons the mark denotes the location of the "f" corner (out of f, g, h, i, j, and k corners) (see Fig. 5).

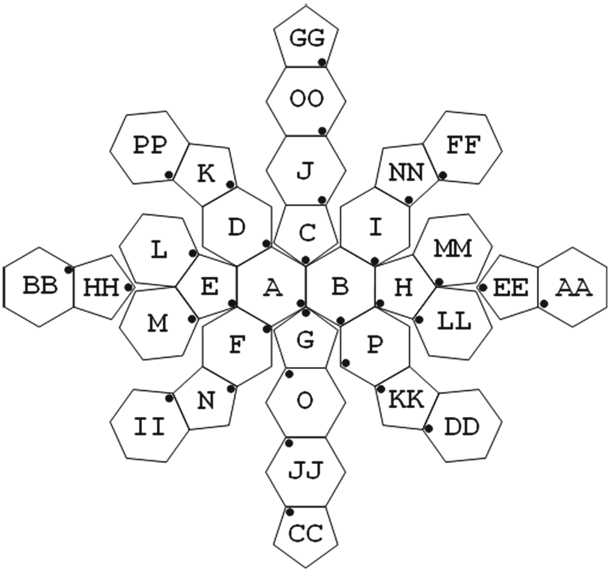


**Fig. 5.** A representation of polygons in a 2-dimensional form with marks

From the reference area (pentagon G), the SVCS-coordinate of each polygon will be determined and elaborated. The algorithm of the coordinate determination follows the following direction guidelines (see Fig. 6).

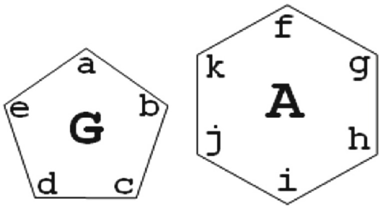


**Fig. 6.** All marked corners of pentagons and hexagons

In the Fig. 7, the red colors are the reference for determining the values of a coordinate along the side (edge) of polygons (1, 2, 3, 4, 5, 6), whereas the blue colors are the reference for determining the values of a coordinate into or out of a polygon (a, b, c, d, e, f, g, h, i, j, k). The following are two examples of SVCS-coordinates of the pentagon GG and the hexagon I.

```
GG-Coordinate is   0:a>1331221>a
I-Coordinate is    0:b>12>f
```

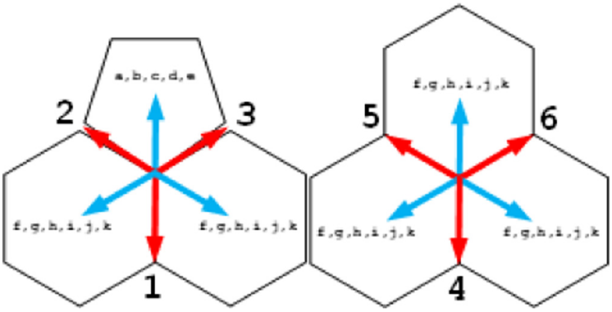


**Fig. 7.** Guidelines to determine the SVCS coordinates (Color figure online)

The following figure shows the path of the coordinates (see Fig. 8).

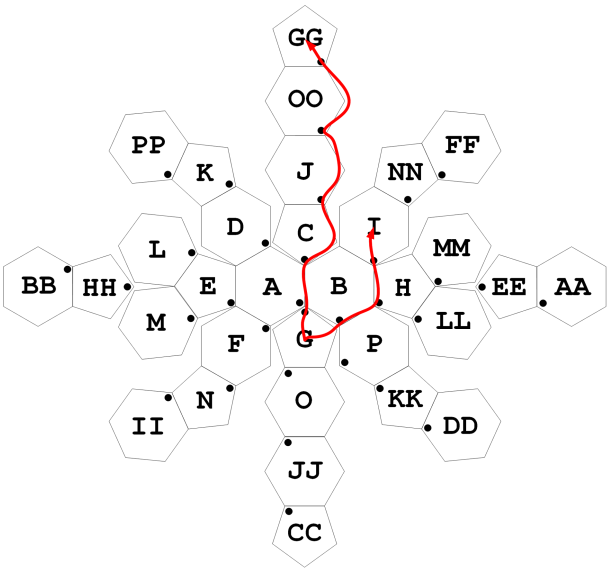


**Fig. 8.** The paths of the coordinates of GG and of I (Color figure online)

This is the description of those coordinates. As already mentioned, the pentagon G is the (starting) reference of all coordinates of the SVCS. The coordinate of GG starts with zero. The first number of a coordinate shows the level of the polygons. The zero means that the polygons have 32 faces. If the first number is one, the polygons have 122 faces, number two means 482 faces, and so on. After the first number of the coordinate is a delimiter (:), followed by the information about from which corner the coordinate is constructed. In the case of the coordinate of GG, from the pentagon G the path goes through the corner "a", followed by another delimiter (>). After the delimiter, the coordinate of GG shows that the values are determined by the direction of red arrows. Using the above references (red and blue arrows), the coordinate is constructed as: 1331221 followed by another delimiter (>) and finished with a letter either one of these: a, b, c, d, and e corners. In the case of the coordinate of GG, the path goes through the corner "a" of the pentagon GG. Note that values 4, 5, and 6 will only appear in coordinates on level 1 and higher.

Similar to the coordinate of GG, the coordinate of I starts with zero. It means that the polygons have 32 faces. After the first number of the coordinate is a delimiter (:), followed by a letter b, meaning that the path goes through the corner "b", then followed by another delimiter (>). After the delimiter, the coordinate of I shows that the coordinate is constructed as: 12 followed by another delimiter (>) and finished with a letter either one of these: f, g, h, i, j, k. In the case of the coordinate of I, the path goes through the corner "f" of the hexagon I.

The following (see Fig. 9) are the coordinates of all 32 polygons including their distances measured from the pentagon G. Note the s denotes the side (edge) length of a pentagon.

| | | | |
|---|---|---|---|
| A | → | 0:a>>f | : Distance = a + f = 0.85 * s + s = 1.85 * s; |
| B | → | 0:b>>f | : Distance = b + f = 0.85 * s + s = 1.85 * s; |
| C | → | 0:a>1>a | : Distance = a + s + a = 2 * 0.85 * s + s = 2.7 * s; |
| D | → | 0:a>12>f | : Distance = a + 2 * s + f = 0.85 * s + 3 * s = 3.85 * s; |
| E | → | 0:e>1>a | : Distance = e + s + a = 2 * 0.85 * s + s = 2.7 * s; |
| F | → | 0:e>>f | : Distance = e + f = 0.85 * s + s = 1.85 * s; |
| G | → | 0:>> | : Distance = 0 |
| H | → | 0:b>1>a | : Distance = b + s + a = 2 * 0.85 * s + s = 2.7 * s; |
| I | → | 0:b>12>f | : Distance = b + 2 * s + f = 0.85 * s + 3 * s = 3.85 * s; |
| J | → | 0:a>133>f | : Distance = a + 3 * s + f = 0.85 * s + 4 * s = 4.85 * s; |
| K | → | 0:a>1221>a | : Distance = a + 4 * s + a = 2 * 0.85 * s + 4 * s = 5.7 * s; |
| L | → | 0:e>133>f | : Distance = e + 3 * s + f = 0.85 * s + 4 * s = 4.85 * s; |
| M | → | 0:e>12>f | : Distance = e + 2 * s + f = 0.85 * s + 3 * s = 3.85 * s; |
| N | → | 0:d>1>a | : Distance = d + s + a = 2 * 0.85 * s + s = 2.7 * s; |
| O | → | 0:d>>f | : Distance = d + f = 0.85 * s + s = 1.85 * s; |
| P | → | 0:c>>f | : Distance = c + f = 0.85 * s + s = 1.85 * s; |
| AA | → | 0:b>133133>f | : Distance = b + 6 * s + f = 0.85 * s + 7 * s = 7.85 * s; |
| BB | → | 0:e>122133>f | : Distance = e + 6 * s + f = 0.85 * s + 7 * s = 7.85 * s; |
| CC | → | 0:d>1221>a | : Distance = d + 4 * s + a = 2 * 0.85 * s + 4 * s = 5.7 * s; |
| DD | → | 0:c>133>f | : Distance = c + 3 * s + f = 0.85 * s + 4 * s = 4.85 * s; |
| EE | → | 0:b>1331>a | : Distance = b + 4 * s + a = 2 * 0.85 * s + 4 * s = 5.7 * s; |
| FF | → | 0:b>12213>f | : Distance = b + 5 * s + f = 0.85 * s + 6 * s = 6.85 * s; |
| GG | → | 0:a>1331221>a | : Distance = a + 7 * s + a = 2 * 0.85 * s + 7 * s = 8.7 * s; |
| HH | → | 0:e>1221>a | : Distance = e + 4 * s + a = 2 * 0.85 * s + 4 * s = 5.7 * s; |
| II | → | 0:d>133>f | : Distance = d + 3 * s + f = 0.85 * s + 4 * s = 4.85 * s; |
| JJ | → | 0:d>12>f | : Distance = d + 2 * s + f = 0.85 * s + 3 * s = 3.85 * s; |
| KK | → | 0:c>1>a | : Distance = c + s + a = 2 * 0.85 * s + s = 2.7 * s; |
| LL | → | 0:b>13>f | : Distance = b + 2 * s + f = 0.85 * s + 3 * s = 3.85 * s; |
| MM | → | 0:b>133>f | : Distance = b + 3 * s + f = 0.85 * s + 4 * s = 4.85 * s; |
| NN | → | 0:b>1221>a | : Distance = b + 4 * s + a = 2 * 0.85 * s + 4 * s = 5.7 * s; |
| OO | → | 0:a>13312>f | : Distance = a + 5 * s + f = 0.85 * s + 6 * s = 6.85 * s; |
| PP | → | 0:e>13312>f | : Distance = e + 5 * s + f = 0.85 * s + 6 * s = 6.85 * s; |

**Fig. 9.** The coordinates of all 32 polygons and their distances (s is the side length)

The following (see Fig. 10) are the coordinates of level 0 and its corresponding level 1 of 10 polygons including their distances measured from the pentagon G.

| | | | |
|---|---|---|---|
| A | → | 0:a>>f | : Distance = a + f = 0.85 * s + s = 1.85 * s; |
| A | → | 1:a>12>f | : Distance = a + 2 * s + f = 0.85 * s + 3 * s = 3.85 * s; |
| B | → | 0:b>>f | : Distance = b + f = 0.85 * s + s = 1.85 * s; |
| B | → | 1:b>12>f | : Distance = b + 2 * s + f = 0.85 * s + 3 * s = 3.85 * s; |
| C | → | 0:a>1>a | : Distance = a + s + a = 2 * 0.85 * s + s = 2.7 * s; |
| C | → | 1:a>16556>a | : Distance = a + 5 * s + a = 2 * 0.85 * s + 5 * s = 6.7 * s; |
| D | → | 0:a>12>f | : Distance = a + 2 * s + f = 0.85 * s + 3 * s = 3.85 * s; |
| D | → | 1:a>156565>f | : Distance = a + 6 * s + f = 0.85 * s + 7 * s = 7.85 * s; |
| E | → | 0:e>1>a | : Distance = e + s + a = 2 * 0.85 * s + s = 2.7 * s; |
| E | → | 1:e>16556>a | : Distance = e + 5 * s + a = 2 * 0.85 * s + 5 * s = 6.7 * s;; |
| F | → | 0:e>>f | : Distance = e + f = 0.85 * s + s = 1.85 * s; |
| F | → | 1:e>12>f | : Distance = e + 2 * s + f = 0.85 * s + 3 * s = 3.85 * s; |
| G | → | 0:>> | : Distance = 0 |
| G | → | 1:>> | : Distance = 0 |
| H | → | 0:b>1>a | : Distance = b + s + a = 2 * 0.85 * s + s = 2.7 * s; |
| H | → | 1:b>16556>a | : Distance = b + 5 * s + a = 2 * 0.85 * s + 5 * s = 6.7 * s; |
| I | → | 0:b>12>f | : Distance = b + 2 * s + f = 0.85 * s + 3 * s = 3.85 * s; |
| I | → | 1:b>156565>f | : Distance = b + + 6 * s + f = 0.85 * s + 7 * s = 7.85 * s; |
| P | → | 0:c>>f | : Distance = c + f = 0.85 * s + s = 1.85 * s; |
| P | → | 1:c>12>f | : Distance = c + 2 * s + f = 0.85 * s + 3 * s = 3.85 * s; |

**Fig. 10.** Ten coordinates in both level 0 and level 1

# 4 Characteristics of SVCS

In this section, the characteristics of SVCS will be discussed. These characteristics are important for finding the direction from an area to another area, finding the shortest path between areas, and converting a SVCS coordinate from and to a current coordinate system.

## 4.1 Reversed Coordinates

The SVCS is not only for representing the location of an area, either a pentagon or a hexagon, it is also for representing a direction from the origin to the destination and its distance. Therefore, the SVCS is reversible. Such a characteristic is important for future usages, such as finding a reversed direction, calculating a distance of a complex path, and also finding the shortest path between two areas.

The following (see Fig. 11) are the rules to make a reversed direction of a coordinate. Reversed 1 (!1) is 1, reversed 2 (!2) is 3, reversed 3 (!3) is 2, and reversed 4 (!4) is 4. To get a reversed 5 or reversed 6 it is determined by the previous direction. If the previous direction is 5, then the reversed 5 or reversed 6 is 6, and if the previous direction is 6, then the reversed 5 or reversed 6 is 5.The following shows some examples of coordinates and the corresponding reversed coordinates (denoted by the exclamation mark).

```
J    → 0:a>133>f
!J   → 0:f>221>a
K    → 0:a>1221>a
!K   → 0:a>1331>a
FF   → 0:b>12213>f
!FF  → 0:f>21331>b
BB   → 0:e>122133>f
!BB  → 0:f>221331>e
GG   → 0:a>1331221>a
!GG  → 0:a>1331221>a
```

**Fig. 11.** Some coordinates and their corresponding reversed directions

### 4.2 Shorten the Sequence of a Direction

As also applies in other coordinate systems, there are some circumstances that the sequence of a direction can be shorten. Through such a characteristic, there is a possibility that within the SVCS the shortest path of a direction can be found. Additionally, the shortest path is also important to disclose the distance between two areas.

**Opposite Direction.** The following are rules and samples of shortening the sequence of a direction through opposite directions.

- 23 = !33 = 2!2 = empty, for example 0:b>131233>c can be shorten to 0:b>1313>c
- 233 = 3, or 223 = 2
- 32 = !22 = 3!3 = empty, for example 0:b>131322>c can be shorten to 0:b>1312>c
- 322 = 2, or 332 = 3
- 11 = 1!1 = !11 = empty, for example 0:b>1112133>c can be shorten to 0:b>12133>c
- 44 = 4!4 = !44 = empty, this rule applies for level of polygons>0

**Alternative Direction.** The following are rules and samples of shortening the sequence of a direction through alternative directions.

- 333 = 22, for example 0:b>13331>c can be shorten to 0:b>1221>c
- 222 = 33, for example 0:b>12221>c can be shorten to 0:b>1331>c
- 3333 = 223 = 322 = 2, for example 0:b>133331>c can be shorten to 0:b>121>c
- 2222 = 233 = 332 = 3, for example 0:b>122221>c can be shorten to 0:b>131>c
- 33333 = 3322 = 32 = empty, for example 0:b>1333333>c can be shorten to 0:b>13>c
- 33333 = 2233 = 23 = empty

- 22222 = 3322 = 32 = empty, for example 0:b>3122222>c can be shorten to 0:b>31>c
- 22222 = 2233 = 23 = empty
- 1212 = 31, or 2121 = 13
- 3131 = 12, or 1313 = 21

**From a Pentagon Out or to a Pentagon in.** The following are rules and samples of shortening the sequence of a direction that goes from a pentagon out or to a pentagon in.

- 0:a>131 = 0:b>12, the same way rules for sides: b, c, d, and e
- 0:a>121 = 0:e>13, the same way rules for sides: b, c, d, and e
- 131>a = 21>e, the same way rules for sides: b, c, d, and e
- 121>a = 31>b, the same way rules for sides: b, c, d, and e

**From a Hexagon out or to a Hexagon in.** The following are rules and samples of shortening the sequence of a direction that goes from a hexagon out or to a hexagon in.

- 0:f>313 = 0:g>21, the same way rules for sides: g, h, i, j, and k
- 0:f>212 = 0:k>31, the same way rules for sides: g, h, i, j, and k
- 0:f>222 = 0:g>3, the same way rules for sides: g, h, i, j, and k
- 0:f>333 = 0:k>2, the same way rules for sides: g, h, i, j, and k
- 313>f = 12>k, the same way rules for sides: g, h, i, j, and k
- 212>f = 13>g, the same way rules for sides: g, h, i, j, and k
- 222>f = 3>k, the same way rules for sides: g, h, i, j, and k
- 333>f = 2>g, the same way rules for sides: g, h, i, j, and k

### 4.3 Sequence Modification

There are some other circumstances that the sequence of a direction that is built over a pentagon or over a hexagon can be modified. Through such a characteristic, there is another possibility that within the SVCS the shortest path of a direction can be found.

**Over a Pentagon.** The following are rules and samples of the modification of the sequence of a direction that is built over a pentagon.

- 1>ae>1 = 131, the same way rules for: ba, cb, dc, and ed
- 1>ad>1 = 1331, the same way rules for: be, ca, db, and ec
- 1>ac>1 = 1221, the same way rules for: bd, ce, da, and eb

- 1>ab>1 = 121, the same way rules for: bc, cd, de, and ea

**Over a Hexagon.** The following are rules and samples of the modification of the sequence of a direction that is built over a hexagon.

- 3>fg>3 = 333, the same way rules for: gh, hi, ij, jk, and kf
- 3>fh>2 = 3312, the same way rules for: gi, hj, ik, jf, and kg
- 3>fi>3 = 33133 or 31213, the same way rules for: gj, hk, if, jg, and kh
- 3>fj>2 = 3122, the same way rules for: gk, hf, ig, jh, and ki
- 3>fk>3 = 313, the same way rules for: gf, hg, ih, ji, and kj
- 2>fg>2 = 212, the same way rules for: gh, hi, ij, jk, and kf
- 2>fh>3 = 2133, the same way rules for: gi, hj, ik, jf, and kg
- 2>fi>2 = 21312 or 22122, the same way rules for: gj, hk, if, jg, and kh
- 2>fj>3 = 2213, the same way rules for: gk, hf, ig, jh, and ki
- 2>fk>2 = 222, the same way rules for: gf, hg, ih, ji, and kj

### 4.4 Samples of Directions

The following are two examples of directions (from K to M and from M to K) that have been shortened using above rules. Note that the coordinate of K is actually the direction from G to K and similarly the coordinate of M is actually the direction from G to M (see Fig. 12 and Fig. 13).

```
The coordinate of K, or GK is 0:a>1221>a
The coordinate of M, of GM is 0:e>12>f

The direction of KM =?
KM = KG + GM = !K + M
KM = 0:a>1331>ae>12>f
KM = 0:a>1331312>f
KM = 0:a>13122>f
KM = 0:b>1222>f
KM = 0:b>13>k

The direction of MK =?
MK = MG + GK = !M + K
MK = 0:f>31>ea>1221>a
MK = 0:f>3121221>a
MK = 0:f>33121>a
MK = 0:f>3331>b
MK = 0:k>21>b
```

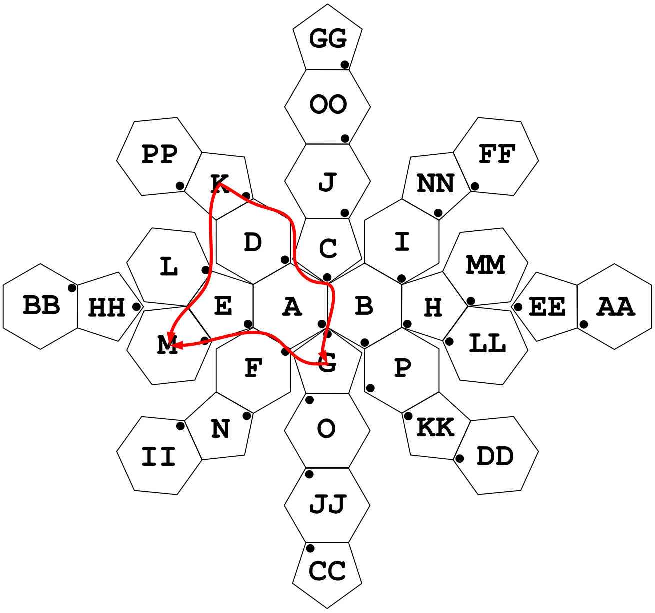


**Fig. 12.** The paths of the direction KG + GM = KM = 0:b > 13 > k

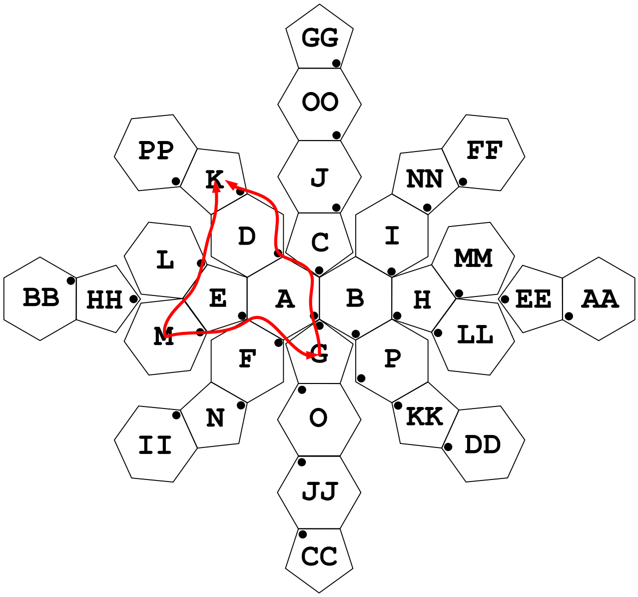


**Fig. 13.** The paths of the direction MG + GK = MK = 0:k > 21 > b

### 4.5 Coordinate Conversion

As all the 21 pentagons and 20 hexagons are so composed that they are symmetrical and regularly distributed, for coordinate conversion purposes it is adequate to only focus on a piece of those polygons, that is 5% of the surface. Through this piece of surface, all the areas on the earth surface can be represented. For discussion purposes this piece is named Half Sector (HS). So there will be 20 HSs in total.

The index of HS in the Southern Hemisphere will be between 0 and 9, whereas the one on the Northern Hemisphere will be between 10 and 19. For a location with a longitude value between 0.0 and 35.999 the coordinate will be within HS with index of 0 or 10, for longitude value between 36.0 and 71.999 the coordinate will be within HS with index of 1 or 11, and so on. Note that the HS with index 0 is the same as the HS with index 2, 4, 6, and 8. Similarly, the HS with index 1 is the same as the HS with index 3, 5, 7, and 9. Additionally, the HS with index 1 is actually the vertically mirrored HS with index 0, and the HS with index 10 is actually the 180 degree rotated HS with index 0. Furthermore, within a HS there are subarea indexes between 0 and 4.

As the side borders of a HS can be formulated as a collection of linear lines, as well as the borders within the HS, the coordinate conversion calculation requires linear equations only. So, the two values of longitude and latitude will determine in which HS and in which subarea index the coordinate is located. The following is the image of the HS with index 0 (see Fig. 14).

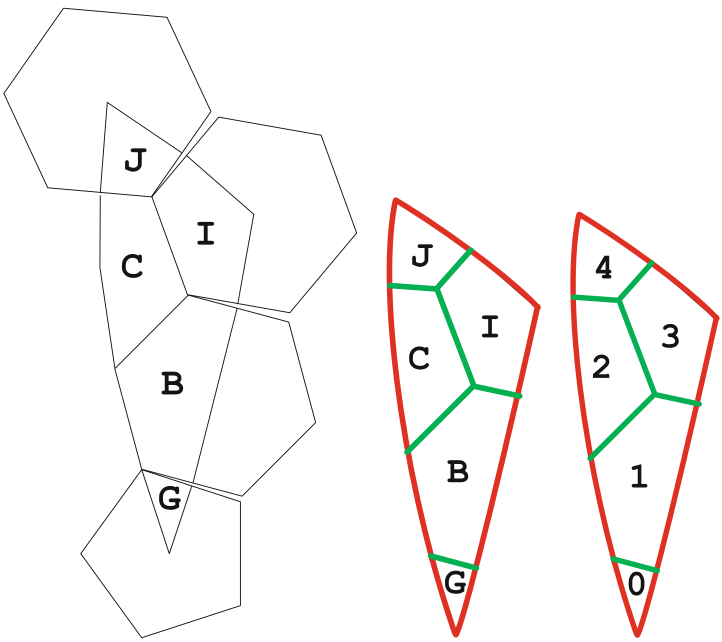


**Fig. 14.** The position of a Half Sector (HS) index 0 and its subarea indexes

## 5 Conclusion

The research on a more suitable coordinate system and the development of Single Value Coordinate System (SVCS) has revealed that the issues within the current coordinate system have been scrutinized and solved. The inaccurate physical location is handled through the regular and symmetric structure of polygons on the earth surface. It means that the accuracy is everywhere the same and the calculation of a distance and an area is not dependent on the location (e.g. near the equator, near the North Pole, etc.). Related to inaccurate distance calculation the SVCS offers a single value of a coordinate and using a direction and shortest path approach. The suitability of the SVCS is solved through utilizing the original shape of the earth, which is a spheroid. With such an approach, the hierarchical characteristic of SLHGN would be maintained. At this stage of the SVCS development, the algorithm for measuring a distance and the conversion from and to the current coordinate system have been made available. The SVCS coordinate system has successfully been developed to work on the coordinate and the direction of an area, not a point. The smaller an area is the more precise the coordinate and the direction will be. When SVCS is completed, the characteristic of the earth as a spherical shape will suit the SLHGN architecture.

## References


1. Nasution, B.B., et al.: An earth coordinate system for earthquake forecasting using SLHGN. In: Murayama, Y., Velev, D., Zlateva, P. (eds.) ITDRR 2019. IAICT, vol. 575, pp. 107–118. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-48939-7_10
2. Nasution, B.B., et al.: Real-time tornado forecasting using SLHGN. In: Murayama, Y., Velev, D., Zlateva, P. (eds.) ITDRR 2018. IAICT, vol. 550, pp. 97–119. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32169-7_8
3. Nasution, B.B., Khan, A.I.: A hierarchical graph neuron scheme for real-time pattern recognition. IEEE Trans. Neural Networks*,* 212–229 (2008)
4. Nasution, B.B.: Towards real time multidimensional hierarchical graph neuron (mHGN). In: The 2nd International Conference on Computer and Information Sciences 2014 (ICCOINS 2014), Kuala Lumpur, Malaysia (2014)
5. Nasution, B.B., et al.: Realtime weather forecasting using multidimenssional hierarchical graph neuron (mHGN). In: The 16th International Conference on Neural Networks (NN 2015), Rome, Italy (2015)
6. Nasution, B.B., et al.: Forecasting natural disasters of tornados using mHGN. In: Murayama, Y., Velev, D., Zlateva, P., Gonzalez, J. (eds.) IFIP Advances in Information and Communication Technology. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-319-68486-4_13
7. Nasution, B.B., et al.: Weather data handlings for tornado recognition using mHGN. In: Murayama, Y., Velev, D., Zlateva, P. (eds.) ITDRR 2017. IAICT, vol. 516, pp. 36–54. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-18293-9_5
8. VIeux, B.E.: Distributed Hydrologic Modeling using GIS. Kluwer Academic Publishers, Oklahoma (2001)
9. Dutton, G.H.: A Hierarchical Coordinate System for Geoprocessing and Cartography. Springer, Heidelberg (1999). https://doi.org/10.1007/BFb0011617
10. Wheatley, D., Gillings, M.: Spatial Technology and Archaeology: The Archaeological Applications of GIS. Taylor & Francis, New York (2002)

11. Harvey, F.: A primer of GIS: Fundamental Geographic and Cartographic. The Guilford Press, New York (2008)
12. Dehant, V., Creager, K., Karato, S.-I., Zatman, S.: Earth's Core: Dynamics, Structure. Rotation. AmericanGeophysical Union, Washington, DC (2003)
13. Einstein, A.: Relativity: The Special and General Theory. Methuen & Co Ltd. (2002)
14. Lambeck, K.: The Earth's Variable Rotation: Geophysical Causes and Consequences. Combridge University Press, New York (2005)
15. Tipler, P.A., Llewellyn, R.A.: Modern Physics. W. H. Freeman and Company, Basingstoke (2012)