[Ouranos 4D Spatio-temporal ID]

Definitions of Spatial ID, Spatial Voxel, and Extended Specifications
(Version 1.1)

Ministry of Economy, Trade and Industry
Ministry of Land, Infrastructure, Transport and Tourism
New Energy and Industrial Technology Development Organization
Information-technology Promotion Agency

Revision History

Revision Date	Revised Section	Description of Revision
Mar 3, 2025	-	Release of gamma version
Oct 24, 2025	-	Release version 1.1 ・Review of the text and addition of extended specifications.

Preface - Positioning of this document

The Ouranos Ecosystem is an initiative promoted by the Ministry of Economy, Trade and Industry in Japan to collaborate and utilize data across companies and industries.

In the Ouranos Ecosystem, "Ouranos 4D Spatio-temporal ID" (abbreviated as Spatio-temporal ID) serves as an index for easily integrating, retrieving, and freely distributing a wide variety of spatial information (including temporal information).

"Spatio-temporal ID" is a combination of "Spatial ID," which is an identifier in three-dimensional space, and "Temporal ID," which is an identifier in the time axis.

Table of Contents

1. Introduction

1.1. Overview of Spatial ID

A Spatial ID uniquely identifies a specific region on Earth. Each region, called a spatial voxel, is a rectangular cuboid in a three-dimensional grid. Spatial voxels can be subdivided to represent areas from global to several centimeters in size, and each voxel receives a unique Spatial ID.

By linking information within each spatial voxel to its spatial ID and using the voxel as the unit of distribution, information sharing becomes standardized and searching or integrating data is easier. Abstracting data by voxel also helps to understand the overall spatial area.

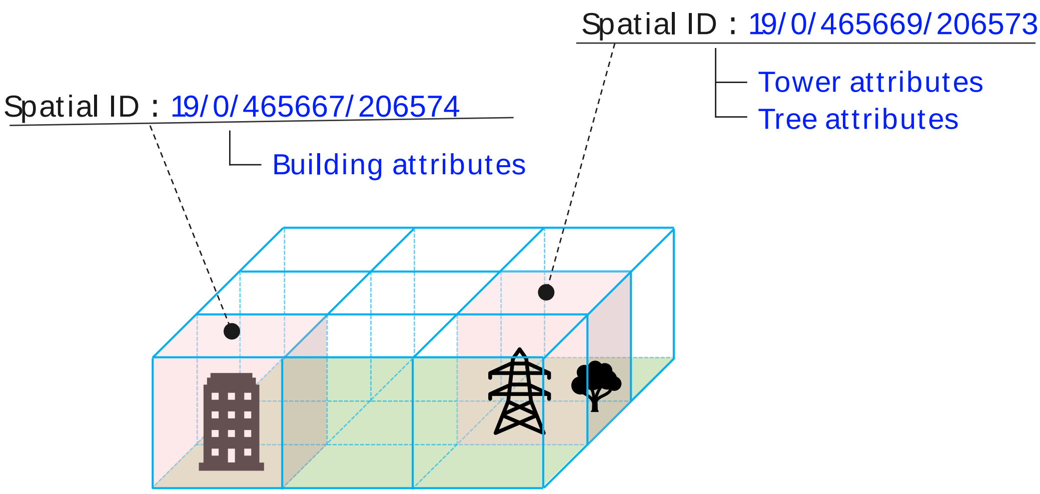


Figure 1-1 Image of spatial ID

1.2. Features and Significance of Spatial IDs

Spatial IDs offer features like 4D space-time handling, unified assignment rules, computable ID generation, and hierarchical structure. They enable interoperability, efficient spatial information use, and reduce processing and communication loads. Concrete benefits depend on each use case. Figure 1-2 summarizes these features, significance, and their relationships.

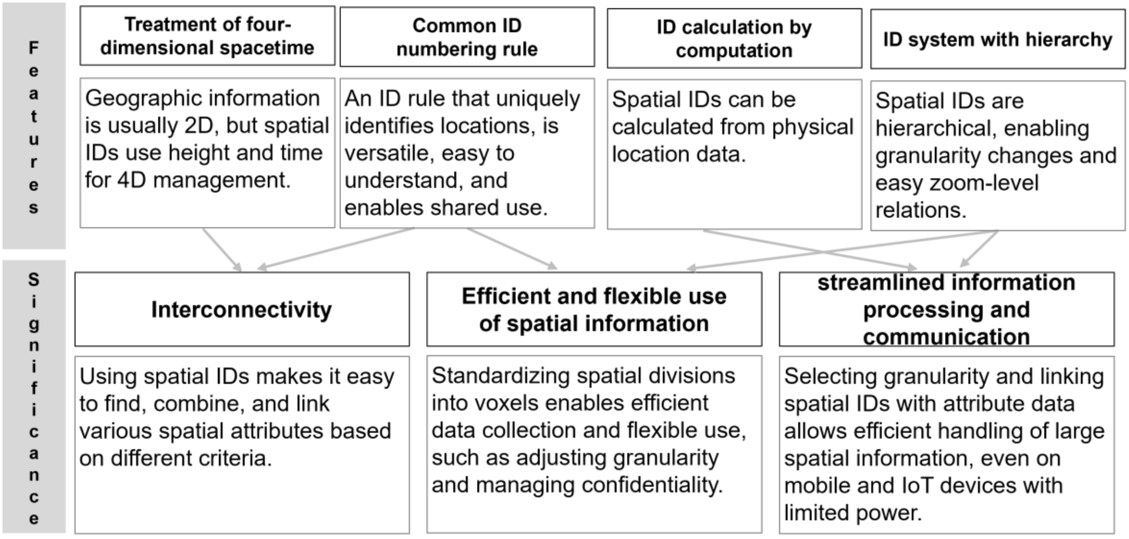


Figure 1-2 Significance of Spatial ID

1.3. Spatial Voxel

1.3.1. Concept of Spatial Voxel

The spatial area of each rectangular parallelepiped obtained by dividing every space on the earth, including the air, the ground, the underground, indoors, and the sea, into a rectangular grid is called a "spatial voxel." "spatial ID" is assigned to each spatial voxel, which can uniquely identify its position even if the spatial information is based on different specification.

Spatial voxels have the following structure.

- The highest hierarchical level is set to zoom level 0, and the spatial voxel is divided into 8 segments each time the zoom level is increased.
- The spatial voxel of the upper zoom level and the spatial voxel of the lower zoom level divided from it have a parent-child relationship.
- There are no overlapping spatial voxels at the same zoom level.

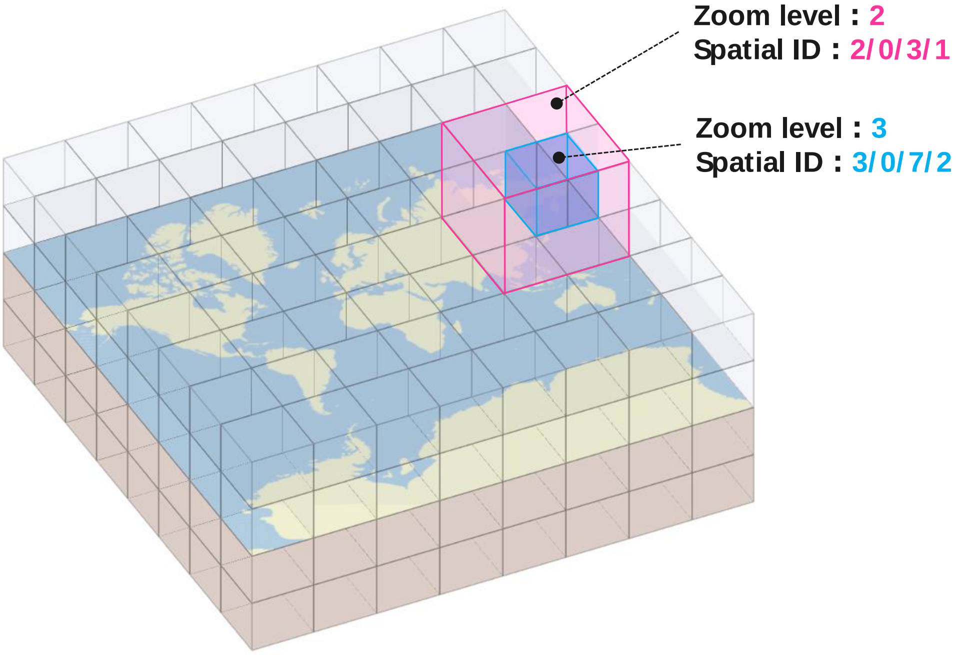


Figure 1-3 Image of spatial voxels and spatial ID

1.3.2. Basic Elements of Spatial Partitioning Methods with Spatial Voxels

There are three basic elements to define a spatial voxel partitioning method. The following sections describe each element in detail.

(1) The height basis for placing spatial voxels

(2) Horizontal spatial partitioning method

(3) Vertical spatial partitioning method

In deciding the spatial partitioning method, the following were taken into consideration: affinity with generally distributed geospatial information and related services, convenience of voxel shape and size, and global availability.

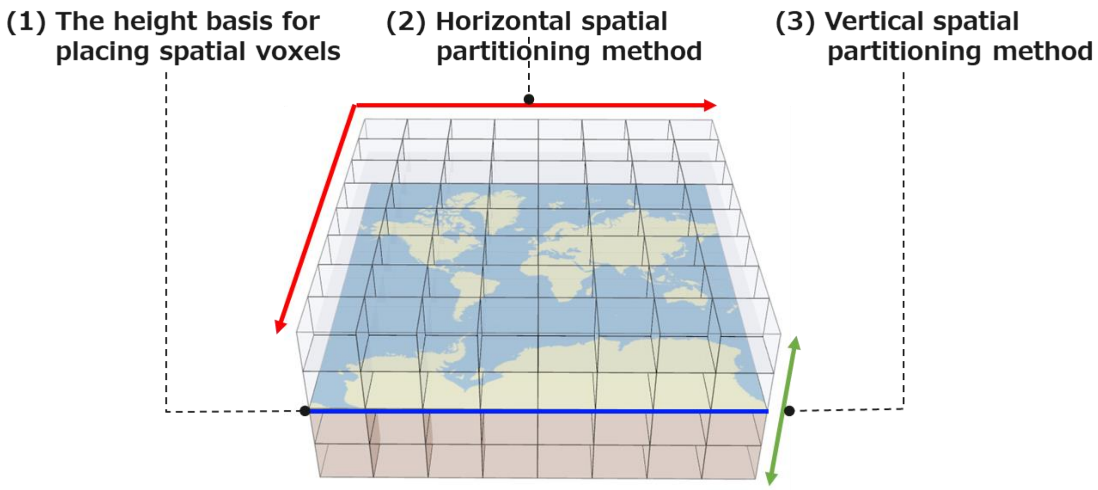


Figure 1-4 Basic elements of the spatial partitioning scheme

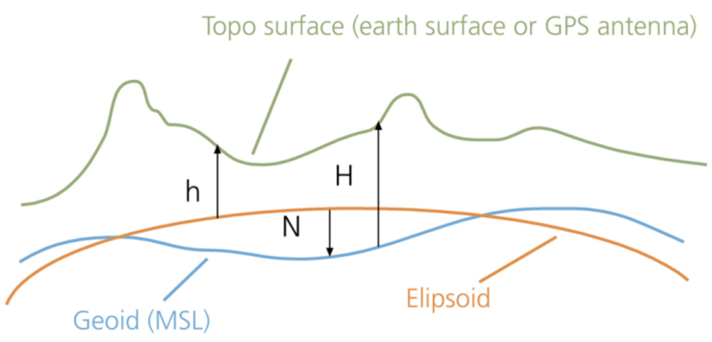
(1) The height basis for placing spatial voxels

i. Reference plane for placing spatial voxels

The reference plane for placing spatial voxels in 3D space on the earth is a geoid (plane with equal potential energy due to the earth's gravity), and the spatial voxels are placed in a coordinate space with a plane with an elevation of 0 m as a flat plane.

The height value of the spatial voxel corresponds to the elevation value (Orthometric Height), and the spatial attribute information created based on the elevation can be directly related to the spatial voxel.

$$h=H+N$$



h=elipsoid height
H=orthometric height
N=geoid height

Figure 1-5 Relationship between ellipsoid height, geoid height and orthometric height

Source: Witold Fraczek, Esri Applications Prototype Lab
https://www.esri.com/news/arcuser/0703/geoid1of3.html

ii. Geoid model

The "geoid model" is used as the data to obtain the elevation from the ellipsoid height obtained by satellite positioning. The geoid model is used depending on the country or region, and the basis of the height should be specified as metadata. In Japan, the geoid is defined by the mean sea level of Tokyo Bay and is used as the reference surface for elevation.

(2) Horizontal spatial partitioning method

i. Horizontal spatial partitioning

The horizontal division is performed over the entire map area defined by the Web Mercator projection.[1].

The target area of the earth is as follows.

Longitude (X): 180 degrees west longitude to 180 degrees east longitude

Latitude (Y): 85.0511 degrees south latitude to 85.0511 degrees north latitude

The above area is set to zoom level 0, and 4 divisions (Divide east-west direction in two, divide north-south direction in two) are repeated each time the zoom level increases by 1. This partitioning method is the same as XYZ tiles (Slippy map tile names) adopted by many Web map services.

ii. Geodetic system

The geodetic system of the map used for horizontal partitioning shall be the World Geodetic System (Japan Geodetic System 2011 (JGD2011) or WGS84). If necessary, the geodetic system adopted in the application area is specified as metadata of the data related to the spatial ID.

iii. Assigning x and y indexes

The x-index for longitude (east-west direction) and the y-index for latitude (north-south direction) are assigned as identification numbers for each divided grid.

Starting from the northwest point (180°W, 85.0511°N) at (0,0), x increments eastward, y increments southward.

x-index value

Left-most columns: x = 0, right-most columns: x = 2 ^ Zoom level – 1

(Increments from 0 of the left-most columns to the next one on the right)

y-index value

[1] The entire map area follows the Web Mercator projection. Any map projection can be used in spatial ID systems, but spatial IDs are calculated using latitude and longitude (decimal) on the reference ellipsoid (JGD2024 or WGS84).

Top rows: y = 0, Bottom rows: y = 2 ^ Zoom level – 1
(Increments from 0 of the top-most rows to the next one on the bottom)

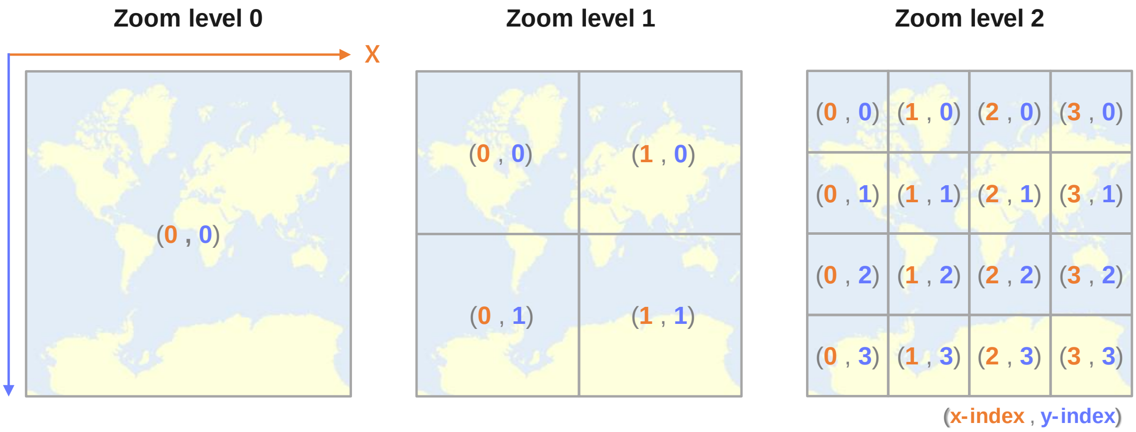


Figure 1-6 Horizontal partitioning and index assignment

(3) Vertical spatial partitioning method

i. Vertical spatial partitioning

Vertical spatial partitioning is based on the positive and negative height ranges. Each range is as follows.

Positive height range (elevation): 0 m to 33,554,432 m

Negative height range (elevation): -33,554,432 m to 0 m

The above positive height range and negative height range are set to zoom level 0, and equal division is repeated every time the zoom level increases by 1.

ii. Assigning an f-index

A f-index is assigned to each range divided in the vertical direction. For each additional zoom level, the positive direction assigns a value incremented by 1 from 0 in the positive direction. The negative direction assigns a value decremented by 1 from -1 in the negative direction.

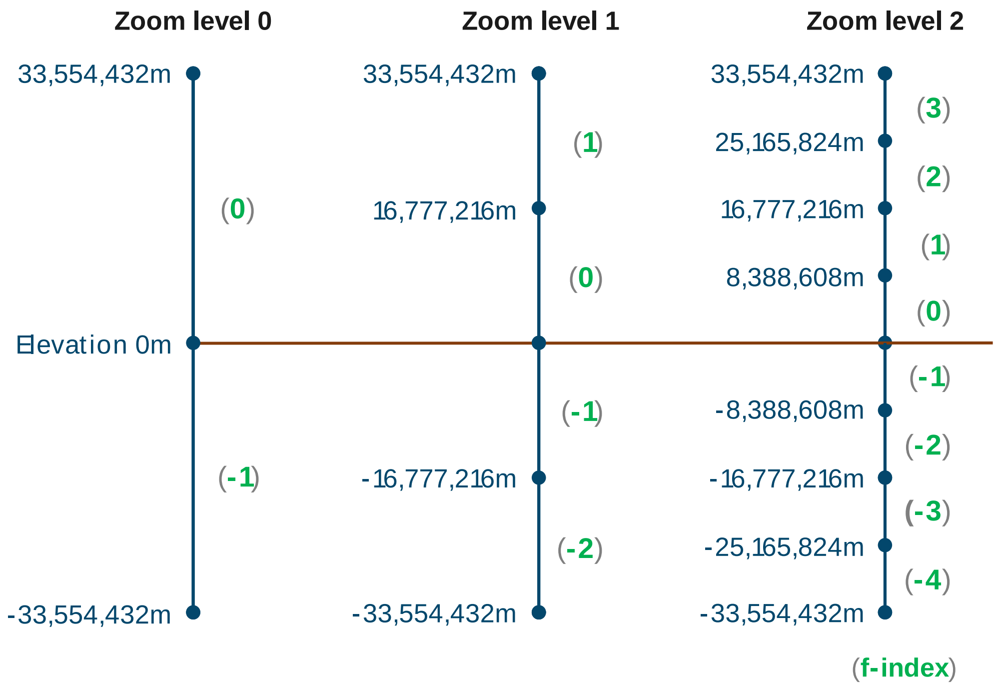


Figure 1-7 Vertical divisions and assigning Indexes

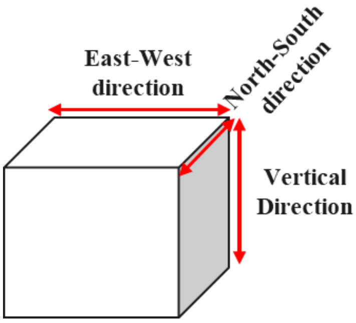
* Elevation values that serve as split points are included in the upper index.
Example) The f-index of 8,338,608 m at zoom level 2 is 1.

1.3.2.1. Zoom Level and Size of Spatial Voxels

A spatial voxel is divided into 8 segments each time the zoom level increases by 1 from zoom level 0, decreasing in size. Examples of the horizontal (east-west and north-south directions) and vertical sizes of spatial voxels at zoom level 0-26 are shown below. Zoom levels 27 and higher can also be defined.

Table 1-1 Examples of spatial voxel sizes at -1 zoom level 0~26

Zoom Level	Horizontal East-West direction (m)	Horizontal North-South direction (m)	Vertical direction (m)
0	40,075,016.68	40,075,016.68	33,554,432.00
1	20,037,508.34	20,037,508.34	16,777,216.00
2	10,018,754.17	10,018,754.17	8,388,608.00
3	5,009,377.09	5,009,377.09	4,194,304.00
4	2,504,688.54	2,504,688.54	2,097,152.00
5	1,252,344.27	1,252,344.27	1,048,576.00
6	626,172.14	626,172.14	524,288.00
7	313,086.07	313,086.07	262,144.00
8	156,543.03	156,543.03	131,072.00
9	78,271.52	78,271.52	65,536.00
10	39,135.76	39,135.76	32,768.00
11	19,567.88	19,567.88	16,384.00
12	9,783.94	9,783.94	8,192.00
13	4,891.97	4,891.97	4,096.00
14	2,445.98	2,445.98	2048.00
15	1,222.99	1,222.99	1024.00
16	611.50	611.50	512.00
17	305.75	305.75	256.00
18	152.87	152.87	128.00
19	76.44	76.44	64.00
20	38.22	38.22	32.00
21	19.11	19.11	16.00
22	9.55	9.55	8.00
23	4.78	4.78	4.00
24	2.39	2.39	2.00
25	1.19	1.19	1.00
26	0.60	0.60	0.50



The distances in the east-west and north-south directions in the above table indicate distances at 0 degrees of latitude, and the distances become shorter at higher latitudes.

The vertical distances are the same for all longitudes and latitudes as shown in Table 1.

The east-west distance of the spatial voxel at each latitude can be calculated by the following formula:

East-west distance = Equatorial radius * 2 * PI * cos (latitude)/2 ^ Zoom level

(The equatorial radius for geodetic systems JGD2011 and WGS84 is 6,378,137 m.)

Source: OpenStreetMap
https://wiki.openstreetmap.org/wiki/Zoom_levels

For reference, the following shows the spatial voxel sizes at zoom level 16-26 for cities in Japan (Naha, Tokyo and Sapporo).

Table 1-2 Examples of spatial voxel sizes at zoom level 16~26 for Naha, Tokyo, and Sapporo

Naha
(Naha City Hall)
Latitude：26.21 degrees north

Zoom Level	Horizontal East-West direction (m)	Horizontal North-South direction (m)	Vertical direction (m)
16	548. 98	546. 01	512. 00
17	274. 49	273. 00	256. 00
18	137. 24	136. 50	128. 00
19	68. 62	68. 25	64. 00
20	34. 31	34. 13	32. 00
21	17. 16	17. 06	16. 00
22	8. 58	8. 53	8. 00
23	4. 29	4. 27	4. 00
24	2. 14	2. 13	2. 00
25	1. 07	1. 07	1. 00
26	0. 54	0. 53	0. 50

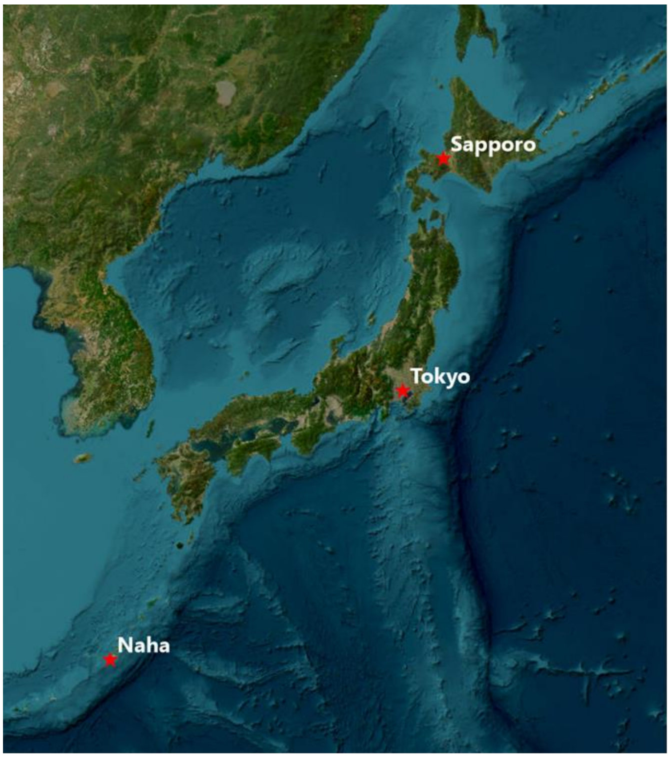
Tokyo
(Tokyo Metropolitan Government Building)
Latitude：35.89 degrees north

Zoom Level	Horizontal East-West direction (m)	Horizontal North-South direction (m)	Vertical direction (m)
16	497. 22	495. 01	512. 00
17	248. 61	247. 51	256. 00
18	124. 31	123. 75	128. 00
19	62. 15	61. 88	64. 00
20	31. 08	30. 94	32. 00
21	15. 54	19. 11	16. 00
22	7. 77	7. 73	8. 00
23	3. 88	3. 87	4. 00
24	1. 94	1. 93	2. 00
25	0. 97	0. 97	1. 00
26	0. 49	0. 48	0. 50

Sapporo
(Sapporo City Hall)
Latitude：43.06 degrees north

Zoom Level	Horizontal East-West direction (m)	Horizontal North-South direction (m)	Vertical direction (m)
16	447. 48	445. 86	512. 00
17	223. 73	222. 92	256. 00
18	111. 86	111. 46	128. 00
19	55. 93	55. 73	64. 00
20	27. 97	27. 87	32. 00
21	13. 98	13. 93	16. 00
22	6. 99	6. 97	8. 00
23	3. 50	3. 48	4. 00
24	1. 75	1. 74	2. 00
25	0. 87	0. 87	1. 00
26	0. 44	0. 44	0. 50

※ Horizontal length is given by the geodesic distance on the GRS80 ellipsoid.
If the east-west distance between the top edge (north side) and bottom edge (south side) of the horizontal plane of the spatial voxel is different, indicate the distance of the bottom edge.



© Earthstar Geographics

Figure 1-8 Locations of Naha, Tokyo, and Sapporo

1.4. Spatial ID

1.4.1. Spatial ID Format

A spatial ID consists of z, f, x, and y elements, which are connected by forward slashes (/).

Elements that make up a spatial ID

{z}: Zoom level

{f}: Elevation (vertical direction) index

{x}: Longitude (east-west direction) index

{y}: Latitude (north-south direction) index

Array of spatial ID

{z}/{f}/{x}/{y}

Example: 20/1/931369/413142

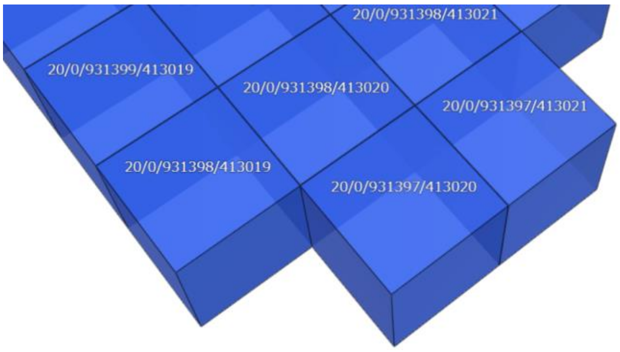


Figure 1-9 Image showing the spatial IDs of spatial voxels

1.4.2. Formulas for Calculating Each Element of Spatial ID

The formulas for calculating each element (f, x, y) of spatial ID from longitude, latitude, elevation, and zoom level are as follows.

----

lng: Longitude [Decimal Degree]

lat: Latitude [Decimal Degree], lat_rad: Latitude [Radians]

h: Elevation $[m]$

z: Zoom Level

$n = 2^z$

$Z = 25$ (The zoom level at which the voxel height is 1m)

$H = 2^Z\ [m]$

$$f = floor\left(n * \frac{h}{H}\right)$$

$$x = floor\left(n * \left(\frac{lng + 180}{360}\right)\right)$$

$$y = floor\left(n * \frac{1 - \frac{\log\left(\tan(lat\_rad) + \left(\frac{1}{\cos(lat\_rad)}\right)\right)}{PI}}{2}\right)$$

Source: UN Vector Tile Toolkit
https://github.com/unvt/zfxy-spec

1.4.3. Spatial ID Format for Data without Heigh Value

In spatial data, there are many two-dimensional data (E.G., demographics, zoning, land use) without height value. When these data are related to spatial IDs, the following ID format excluding the f-index is applied because the f-index indicating elevation is not necessary.

Array of spatial IDs corresponding to data without height value

{z}/{x}/{y}

Example: 20/931369/413142

This is the same array as XYZ tiles numbers in web map services such as Geographical Survey Institute Maps.

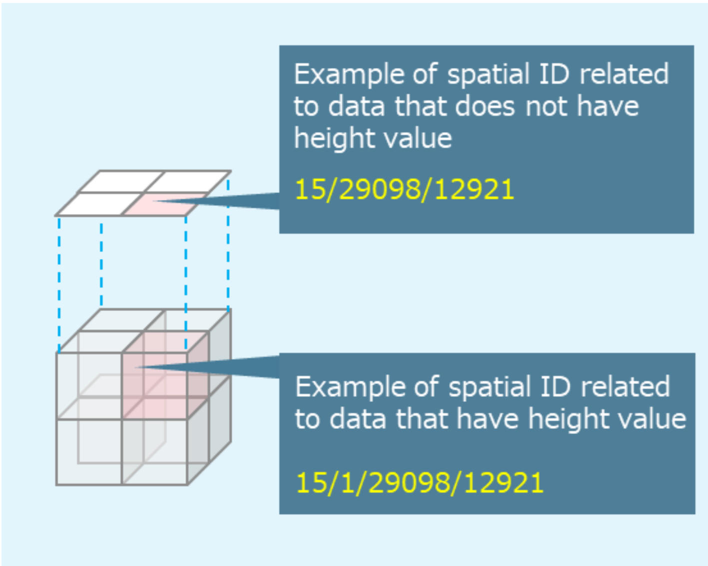


Figure 1-10 Example of spatial IDs corresponding to data without height value

1.5. Extended Specifications

This section describes the specifications for "local spatial IDs," which are for applying the role of spatial IDs to manage local spatial information, and "Spatio-temporal ID," which are for applying the role of spatial IDs to manage information that changes or moves over time. Besides, because local spatial IDs and Spatio-temporal ID are optional extensions, their application is optional; therefore, their implementation shall be determined depending on the needs of the use case.

1.5.1. Local Spatial ID

Local spatial ID is a common rule that can be used to manage spatial information when information needs to be shared within a local area. Specifically, local spatial IDs are intended to be applied in cases such as when data managed in a local coordinate system indoors, e.g., in buildings is handled; when real spaces and digital spaces are linked in entertainment use cases to utilize spaces at various positions, and when the local coordinate system itself moves, such as inside a vehicle's bed.

In this section, the term “global spatial ID” is used to mean the common “spatial ID” for the purpose of explicitly contrasting with local spatial ID.

(1) Spatial partitioning method for local spatial IDs

The spatial partitioning method for local spatial IDs in the horizontal direction is as follows.

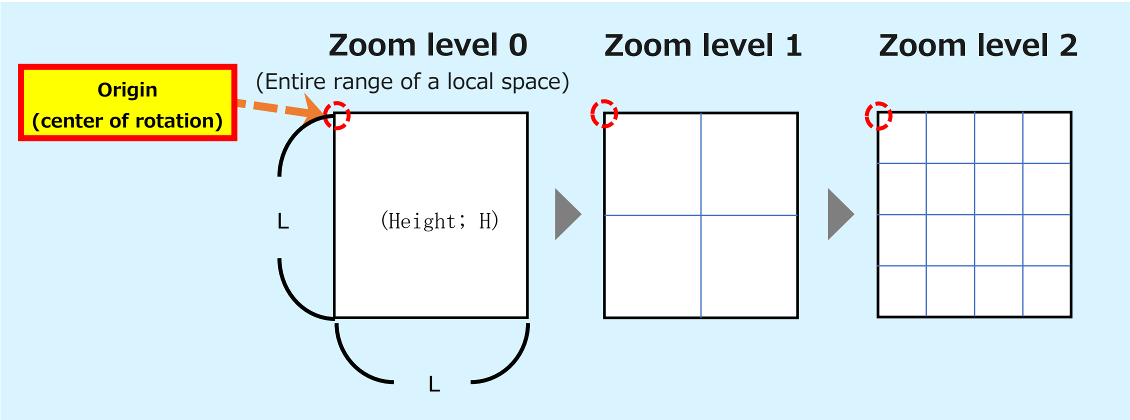


Figure 1-11 Spatial partitioning method for local spatial IDs (horizontal direction)

The entire range of local spaces is defined in accordance with the interoperability and the reusability of codes based on the same concepts and logic as global spatial IDs, and subdivision into four parts is repeated in a two-dimensional space; subdivision into eight parts is repeated in a three-dimensional space by the same method as global spatial IDs.

The spatial partitioning method for the vertical direction of local spatial IDs is as follows.

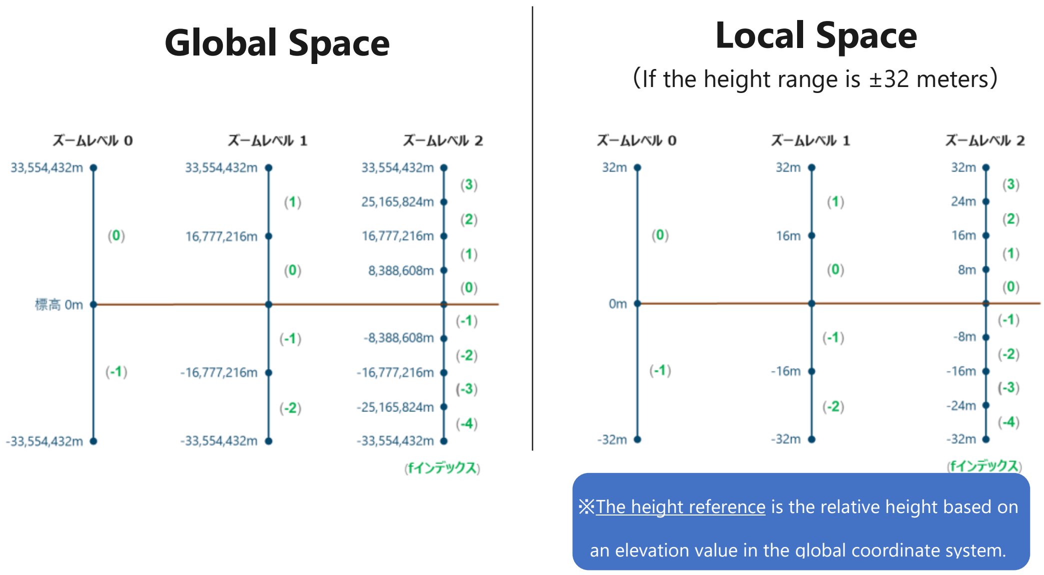


Figure 1-12 Spatial partitioning method for local spatial IDs (vertical direction)

(2) Local spatial ID format

A spatial ID consists of z, f, x, and y elements, which are connected by forward slashes (/). The format is the same as that of global spatial ID.

Elements that make up a spatial ID

{z}: Zoom level

{f}: f-index

{x}: X-index

{y}: Y-index

Array of spatial ID

{z}/{f}/{x}/{y}

Example: 2/0/1/3

The illustrative image below shows the local spatial IDs for each spatial voxel[2]. This example uses “zoom level 2,” creating an array of “4 × 4 cells” in a two-dimensional space.

[2] SDK demo for local spatial ID https://geolonia.github.io/local-spatial-id-js-sdk/

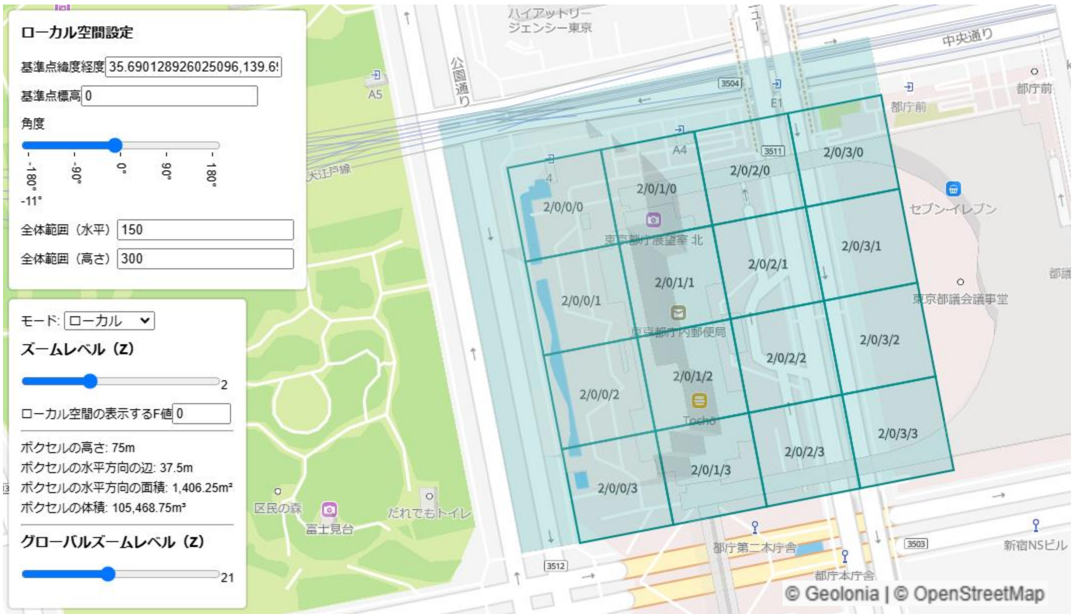


Figure 1-13 Image showing the spatial IDs of spatial voxels

(3) Formulas for calculating each index of spatial ID

The formulas for calculating each index (f, x, y) of local spatial ID from local space coordinates (X, Y), height, and zoom level are as follows.

[Formulas for calculating each index of local spatial ID]

$x \ = \ floor \ (\ n \ * \ X \ / \ L \ )$

$y \ = \ floor \ (\ n \ * \ Y \ / \ L \ )$

$f \ = \ floor \ (\ n \ * \ h \ / \ H \ )$

※$n = 2\text{^}z \ (z\text{: Zoom level})$

$(X, Y) = \text{Local space coordinates}[m]$

$h = \text{Height（based on the elevation of the origin in the global coordinate system）[m]}$

$[L, H]\text{: Length of one side of the entire range(L; horizontal, H; vertical)}$

$f\text{: f} - \text{index}$

$x\text{: X} - \text{index}$

$y\text{: Y direction index}$

The coordinate axes and entire range of local spatial IDs are defined as follows.

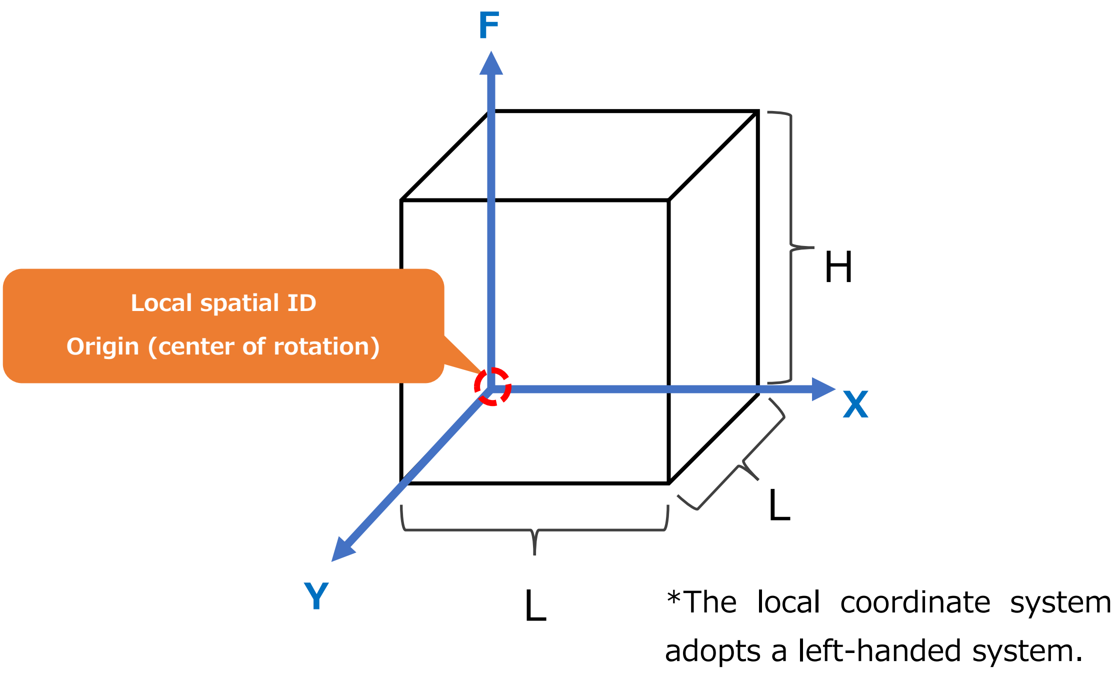
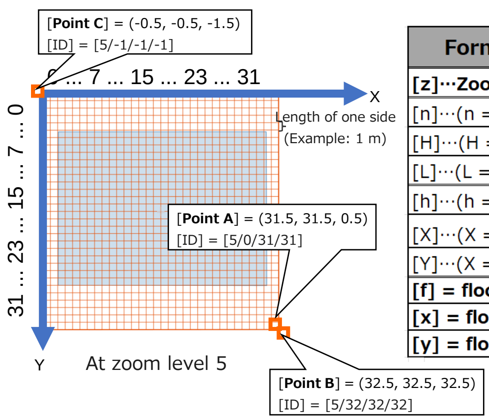


Figure 1-14 Definition of coordinate axes and entire range of local spatial IDs

The local coordinate system adopts a left-handed system. The figure shows the origin of the local spatial IDs as a red circle, which is the center of rotation for positioning relative to the global coordinates. Besides, the length of one side of the entire range of local spatial IDs can be defined independently as L (length of X and Y are the same) in the horizontal direction and H in the vertical direction.

Example calculations based on the formulas for local spatial IDs are as follows.



Formula and variables/settings	Type	Point A	Point B	Point C
[z]…Zoom level	**Setting value** (spatial definition)	**5**	**5**	**5**
[n]…(n = 2^z)		32	32	32
[H]…(H = total range height)		32	32	32
[L]…(L = horizontal length of the entire range)		32	32	32
[h]…(h = elevation/H coordinate value [m])	**Reference point** (input coordinate value)	0.5	32.5	-0.5
[X]…(X = local space/X coordinate value [m])		31.5	32.5	-0.5
[Y]…(X = local space/Y coordinate value [m])		31.5	32.5	-0.5
[f] = floor(n * h/H)	**Spatial ID** (index value formula)	**0**	**32**	**-1**
[x] = floor(n * X/L)		**31**	**32**	**-1**
[y] = floor(n * Y/L)		**31**	**32**	**-1**

Figure 1-15 Example calculations using calculation formulas for local spatial IDs

In the local spatial ID specifications, the start point of the indexes in the horizontal (X- and Y-) direction is based on the origin of the local coordinate system. The results of trial calculations with simple settings using formulas in the local spatial ID specifications are shown as an example. Below is an example calculation using the local spatial ID calculation formulas (assuming the entire range is a cubic voxel measuring 32 × 32 × 32 m).

[Point A]: The first voxel with the largest coordinate values and a height value close to 0 within the entire range in the local coordinate system

=> (X,Y,h) = (31.5,31.5,0.5)

[Point B]: The first voxel outside the entire range in the positive direction ⇒ (X,Y,h) = (32.5,32.5,32.5)

[Point C]: The first voxel outside the entire range in the negative direction => (X,Y,h) = (-0.5,-0.5,-0.5)

(4) Definition of coordinate in local spatial IDs

The definition of coordinate in local spatial ID consists of the following elements. Defining coordinate information enables correspondence with global spatial IDs.

① Length of each side of the entire range [in meters]
⇒ Let the length of both sides in the horizontal direction (X, Y) be [L].
The length in the vertical direction can be defined with a length [H], which is different from [L].

② Origin position* (latitude, longitude, and elevation)
⇒ Set the upper left corner of the entire range as the coordinate origin and the center of rotation.

③ Rotation angle* (Euler angle; -180 to +180°) [degrees]
⇒[Rotation angle (ψ) around the Z axis (X-Y plane)]

*When used only locally, the definition is optional.

※In cases where the local coordinates themselves move, a possible operation is to update the information of ii and iii after the movement is complete, rather than updating the coordinates dynamically during the movement.

The following is an example of the entire range of local spatial IDs overlaid on global spatial IDs based on coordinate definition information.

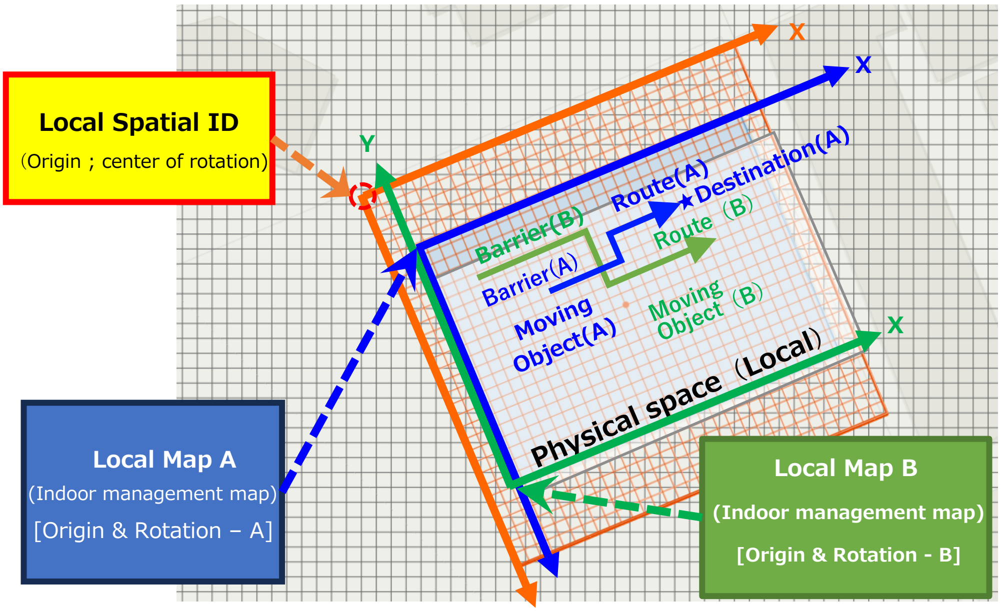


Figure 1-16 Local spatial IDs overlaid on global spatial IDs based on coordinate definition information

(5) Zoom level and voxel size of local spatial ID

The relationship between zoom level and voxel size in local spatial IDs is as follows.

Zoom level	Entire range	Voxel size[m]	Zoom level	Entire range	Voxel size[m]
0	32	**32**	**0**	25.6	**25.6**
1	32	16	1	25.6	12.8
2	32	8	2	25.6	6.4
3	32	4	3	25.6	3.2
4	32	2	4	25.6	1.6
5	32	**1**	5	25.6	0.8
6	32	0.5	6	25.6	0.4
7	32	0.25	7	25.6	0.2
8	32	0.125	**8**	25.6	**0.1**

Figure 1-17 Zoom level and voxel size of local spatial ID

In the example on the left above, when the entire range is defined to have 32-m sides, zoom level [5] creates 1-m voxels, which can be used as the operational reference unit. To set the reference unit to '0.1 m (10 cm),' set the length of one side of the entire range to 25.6 m so that zoom level [8] creates 10-cm voxels. In other words, to arbitrarily define the operational reference unit, you should specify the size of the entire range as 'the reference unit multiplied by the powers of 2.'

(6) Handling of areas outside the range of local spatial IDs

The handling of areas outside the entire range of local spatial IDs is as follows.

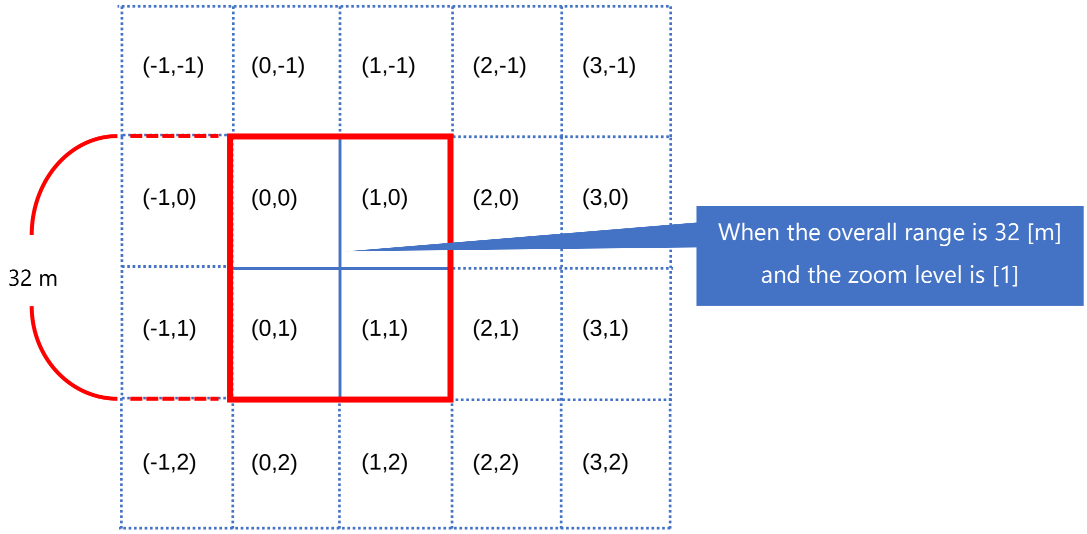


Figure 1-18 Handling of areas outside the entire range of local spatial IDs

Areas outside the defined entire range (including negative directions) can also be indexed. However, considering the influence on tools such as libraries, IDs outside the entire range cannot be used.

Regarding the connection processing of the left and right boundaries of the entire range, when the adjacency of voxels is determined based on the boundary in the left-right direction (definition range: 180° west longitude to 180° east longitude) of global spatial IDs, the two ends are connected as being adjacent. However, in local spatial IDs, the left and right ends are not connected (similar to the vertical direction).

(7) Correspondence of local spatial IDs and global spatial IDs

The following is an example of corresponding local spatial IDs with global spatial IDs in a 3D viewer.

Local spatial ID/common library

The following is an example of display in a 3D viewer based on the definition information of local spatial ID coordinate using a demo site.

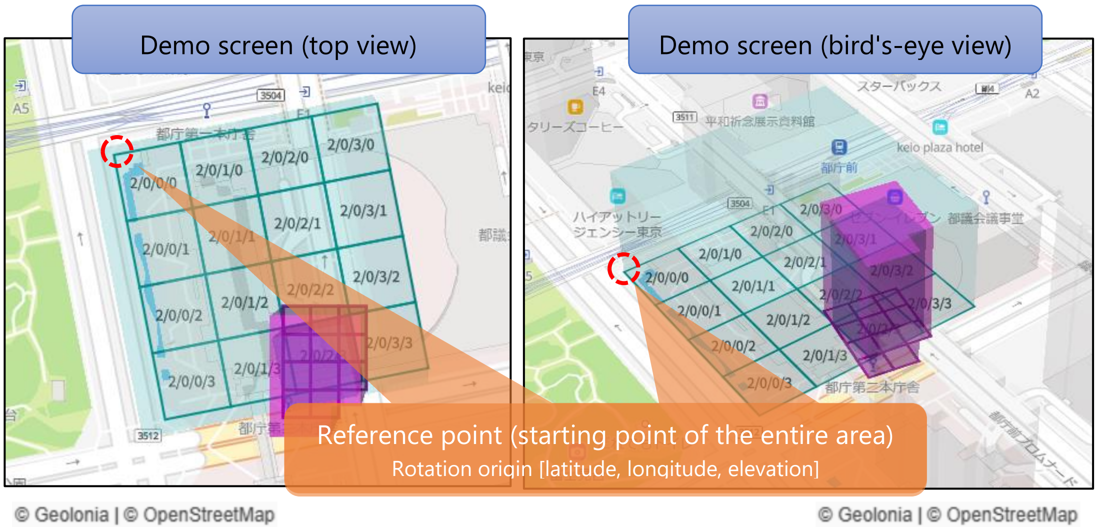


Figure 1-19 Example display of local spatial IDs using a 3D viewer

The following settings and operations are available on the demo site. Selecting a part of local spatial IDs enables to obtain the voxels of corresponding global spatial IDs (purple voxels in the figure).

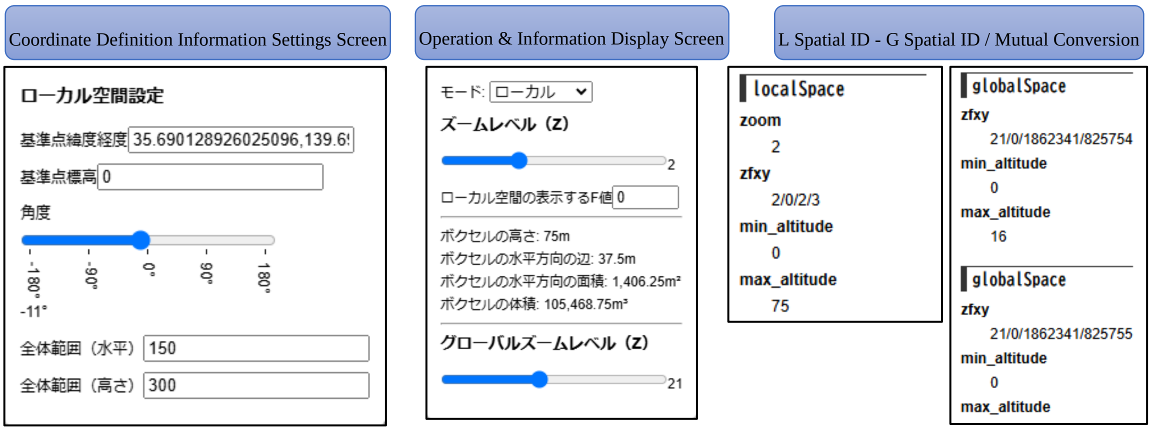


Figure 1-10 Demo site settings screen and information confirmation screen

In the above settings screen, the following information is set as the coordinate definition information shown in Figure 2-19, and the related information can be checked on the screen.

- Coordinate definition information: [Latitude and longitude of the reference point][Elevation][Angle][Entire range (horizontal)][Entire range (vertical)]
  Example: [35.690128926025096,139.69097558834432][0][-11][150][300]
- Mode selection (local or global)
- Zoom level [local, global]

1.5.2. Spatio-temporal ID

A Spatio-temporal ID is defined so that its role as a search key for searching and integrating spatial information of spatial IDs can be applied to information that changes and moves over time. By linking and integrating information on moving objects and events to Spatio-temporal ID created by dividing the time axis into equal intervals, it is possible to identify the general situation (number of moving objects, pedestrian traffic, weather conditions, etc.) at a specific time and space. This can be used to identify Spatio-temporal points with high congestion or risk levels.

(1) Structure of Spatio-temporal ID

A Spatio-temporal ID consists of a spatial ID and a time ID, with the Temporal ID being optional.

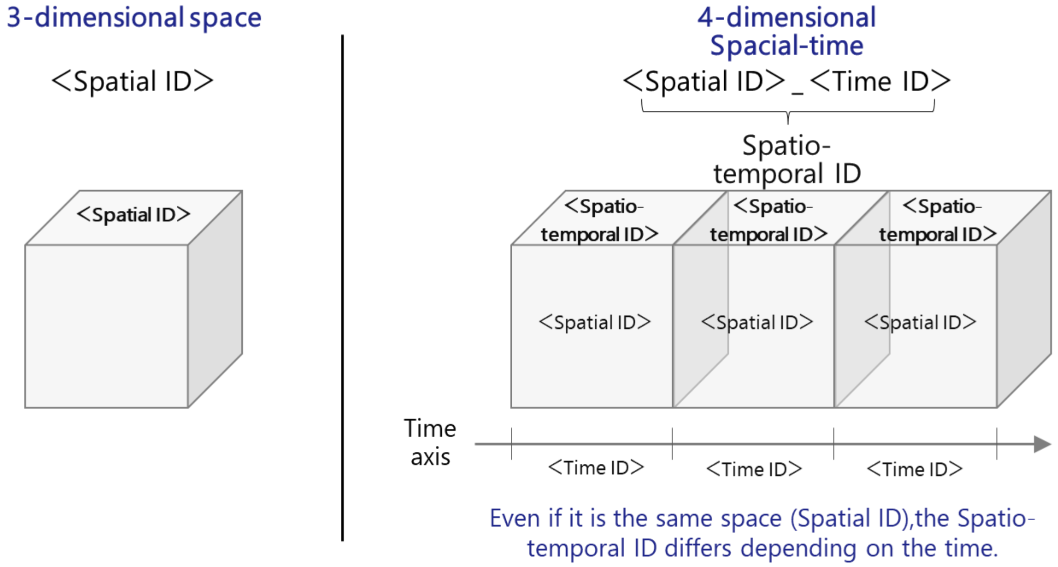


Figure 1-11 Image of comparison of spatial ID and Spatio-temporal ID

(2) Time axis partitioning method

The time axis is divided into equal intervals [arbitrarily specified intervals (unit: seconds)] starting from the starting point of the time axis [1970/1/1, 0:00], and a unique identifier is assigned to each time interval. The defined items and their description for time axis partitioning are as follows.

Starting point of time axis: 1970/1/1, 0:00 (Coordinated Universal Time (UTC))
Time unit: seconds

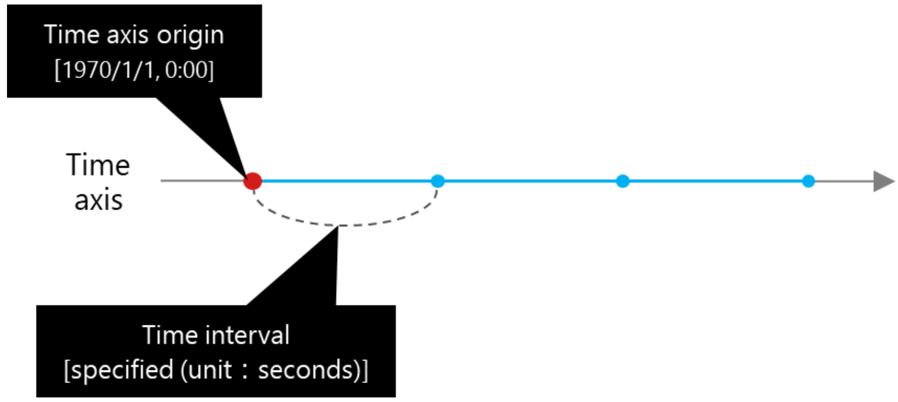


Figure 1-12 Time axis partitioning

The starting point of the time axis is 1970/1/1 0:00 (Coordinated Universal Time (UTC)), which is the starting point of UNIX time, and the number of seconds elapsed from the starting point is equivalent to UNIX time. UNIX time is not affected by time differences between countries and regions or daylight-saving time, so globally unified time IDs can be defined. Because UNIX time is based on Coordinated Universal Time (UTC), the time used to calculate time IDs is unified to Coordinated Universal Time (UTC).

Time intervals can be specified as any number of seconds. Multiple levels can be defined, and they are independent of the zoom level of spatial IDs.

(3) Calculation method for time index

The following formula is used to calculate the time index, which is the identifier for each time interval, from UNIX time and the time interval (seconds).

---

t : Time index(The identifier for each time interval)
u : UNIX time (The number of seconds elapsed from 1970/1/1 0:00)
i : The time interval (seconds)

$$t = floor(u / i)$$

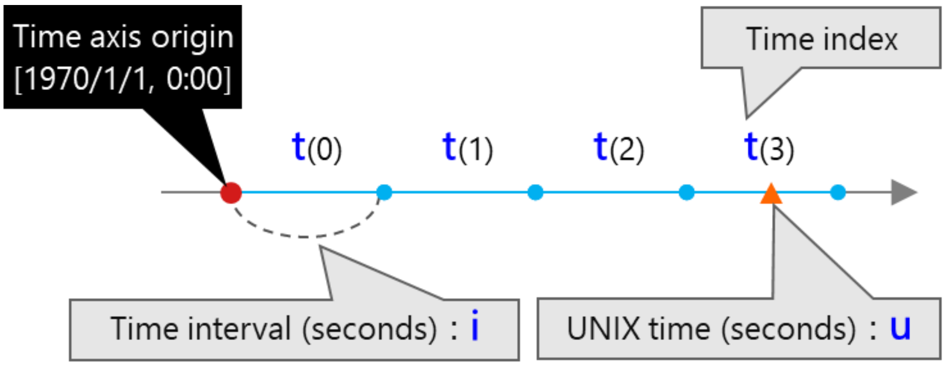


Figure 1-23 Components in a divided time axis

(4) Format of Spatio-temporal ID

Spatio-temporal ID is created by adding a time ID (consisting of a time interval and a time index) to a spatial ID ({z}/{f}/{x}/{y}).

The character used to link the spatial ID and time ID is an underscore (_). (Although a slash (/) is used as the linking character between indexes for spatial IDs, the underscore is employed to avoid confusion with spatial Ids, which use spatial IDs that correspond to data without height information ({z}/{x}/{y}).)

Components of time ID

{i}: Time interval (seconds)

{T}: Time index

Array of Spatio-temporal ID

{z}/{f}/{x}/{y}_{i}/{t}

Example: 12/0/3638/1614_1800/809712

(5) Identification of the relationship between time IDs with different time interval definitions

Calculating the time range in seconds from the time ID components enables the identification of the relationship between time IDs with different time interval definitions (such as overlap and inclusion).

Formula for calculating the start point (seconds) of the time range of time ID

i * t [Time interval * Time index]

Formula for calculating the end point (seconds) of the time range of time ID

(i * t) + t [(Time interval * Time index) + Time interval]

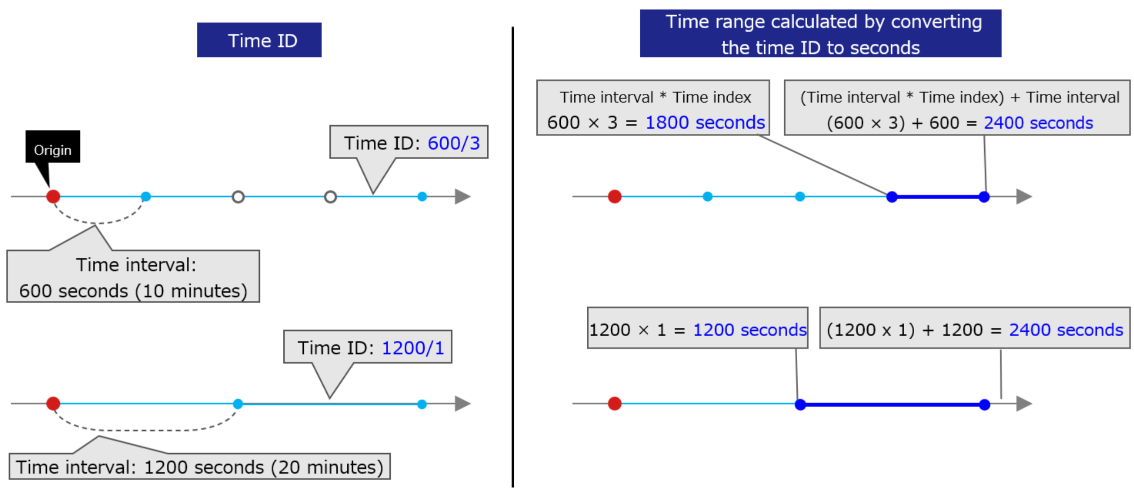


Figure 1-24 Conversion from time ID to time range in seconds