F4D Reference Manual

F4D v0.0.1 Data Format Specification
10 JAN 2019

V 1	2019-1-10	F4D v0.0.1

Contents

1.Format structure

F4D is composed of 1 header file, 3 folders for indoor data structure, 3 binary files for outdoor data structure and 4 image files for textures used in LOD 2~5. Different with LOD systems of other 3D graphics fields, LOD system of F4D has reverse LOD order. The lower number LOD a model has, the more detailed data it has. It is for, in future, supporting flexible LOD based on an idea that bigger buildings can be seen from further distance. The name of the folder which contains one F4D dataset is 'F4D_[dataKey]' where 'dataKey' is an unique key for an F4D dataset.

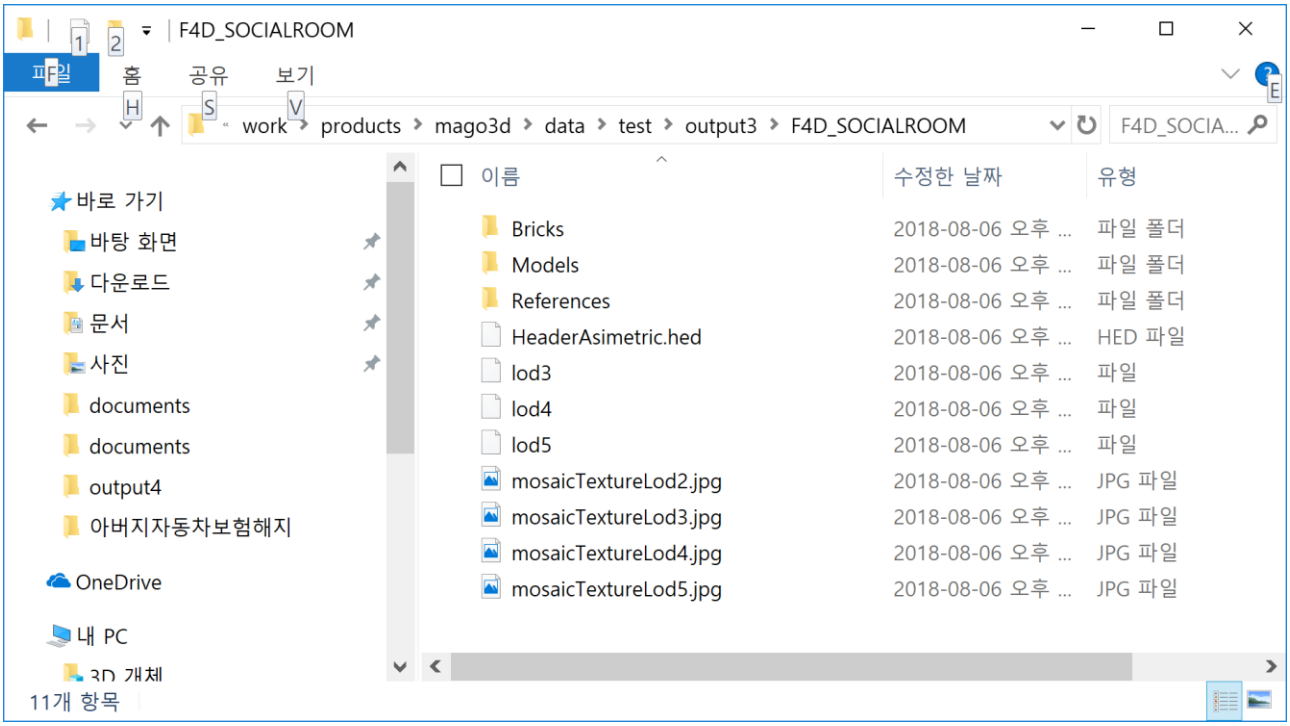

Figure 1. Overall Structure of F4D Format

objectIndexFile.ihe is necessary to publish F4D data through web services. This file is a temporary metadata, which is going to nothing. (The role of this file is being implemented in other parts mago3D and this file is being removed. not removed completely yet.) This file should be in F4D data folder for web service at the same depth as each F4D data folders.

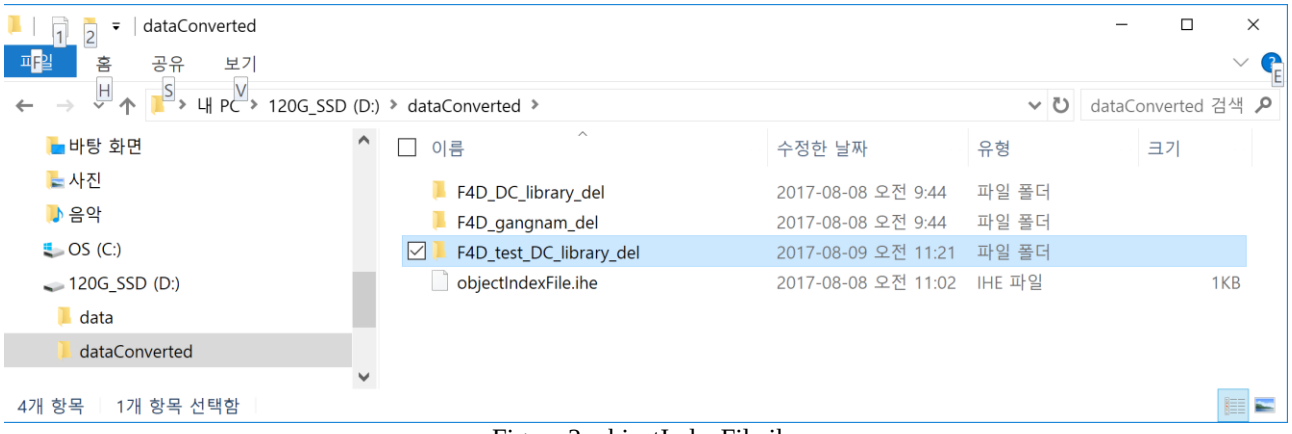

Figure 2. objectIndexFile.ihe

* "Length of Bytes" in all following table contents are described by C++ style.

1.1 HeaderAsimetric.hed

It contains the metadata for 3D GIS of datasets in an F4D folder and serves as a header file of datasets. Overall structure of HeaderAsimetric.hed is explained in Table 1, 1-1, and 1-2.

Table 1 Structure of HeaderAsimetric.hed

Name of the Field	Length of Bytes	Remarks
version	5 × sizeof(char) bytes	F4D version info. Latest verison is 0.0.1
guidLength	sizeof(int) bytes	Length of guid
guid	[guidLength] × sizeof(char) bytes	Guid. Reserved. Not used now.
longitude latitude	2 × sizeof(double) bytes	Longitude, latitude(degree) of datasets. Not used now.
altitude	sizeof(float) bytes	Altitude (unit: m) of datasets. Not used now.
bbox	6 × sizeof(float) bytes	Bounding box of minX, minY, minZ, maxX, maxY, maxZ in meter
octree dimension	flexible	Hierarchy info of octree structure
textureCount	sizeof(unsigned int) bytes	Count of textures used in the F4D
textureInfo	$\sum_{i=1}^{\text{textureCount}} \text{textureInfoSize}_i$ bytes	all texture information
lodCount	sizeof(unsigned char) bytes	Count of all supported LOD
lodInfo	$\sum_{i=1}^{\text{lodCount}} \text{lodInfoSize}_i$ bytes	all supported LOD info

Table 1-1 Detailed Structure of “octree dimension” Specified in Table 1

Name of the Field	Length of Bytes	Remarks
depth	sizeof(unsigned int) bytes	Depth of this octree
bbox	6 × sizeof(float) bytes	Bounding box of minX, minY, minZ, maxX, maxY, MaxZ. Only record when depth == 0
childCount	sizeof(unsigned char) bytes	No. of child octree
triangleCount	sizeof(unsigned int) bytes	No. of triangle in this octree
chidrenDimensions	flexible	Recursively record the child octree dimensions

Table 1-2 Detailed Structure of “textureInfo” Specified in Table 1

Name of the Field	Length of Bytes	Remarks
textureTypeLength	sizeof(unsigned int) bytes	Length of string for texture type
textureType	[textureTypeLength] × sizeof(char) bytes	Texture type. Only “diffuse” is used now.
textureFileNameLength	sizeof(unsigned int) bytes	Length of string for texture file name
textureFileName	[textureFileNameLength] × sizeof(char) bytes	Texture file name

Table 1-3 Detailed Structure of “lodInfo” Specified in Table 1

Name of the Field	Length of Bytes	Remarks
lod	sizeof(unsigned char) bytes	LOD number
bDividedBySpatialOctree	sizeof(bool) bytes	whether this LOD is divided by octree
lodFileNameLength	sizeof(unsigned char) bytes	Length of string for LOD data file name
lodFileName	[lodFileNameLength] × sizeof(char) bytes	LOD data file name. Only available when lod >= 3.(that is, only outdoor)
lodTextureFileNameLength	sizeof(unsigned char) bytes	Length of string for LOD texture file name
lodTextureFileName	[lodTextureFileNameLength] × sizeof(char) bytes	LOD texture file name. Only available when LOD >=2.(that is, only data made by NSM method)

1.2 Bricks Folder

This folder contains binary files of LOD2 geometric data. F4D converter splits an original 3D model data into smaller-sized models divided by octrees, and applies NSM(Net Surface Mesh) method on each octree to reduce data size and creates rougher data. The name of each files is "[octreeNumber]_brick". The count of digit of octreeNumber represents the depth of octree system and non-existing octreeNumber means that there is no intersection between original model and the octree of that octreeNumber.

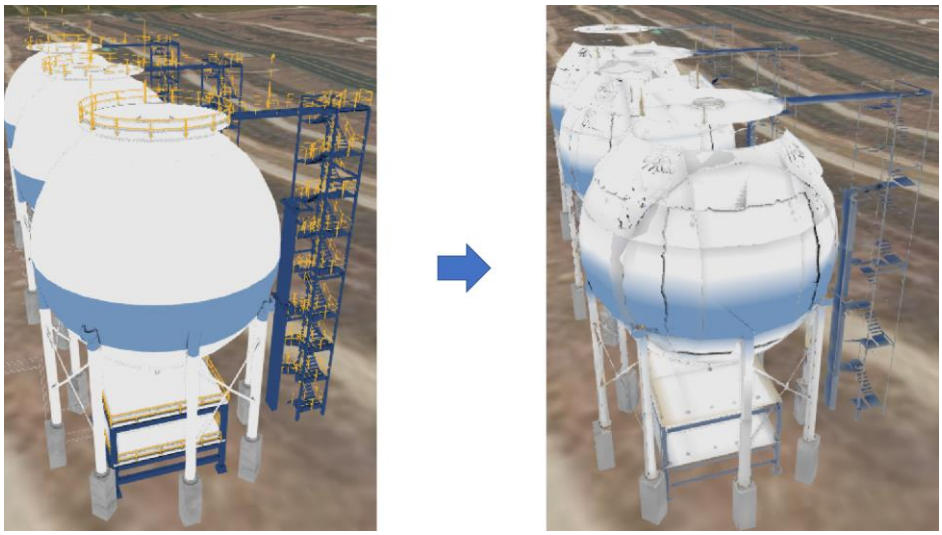

Figure 3 before/after NSM is applied.

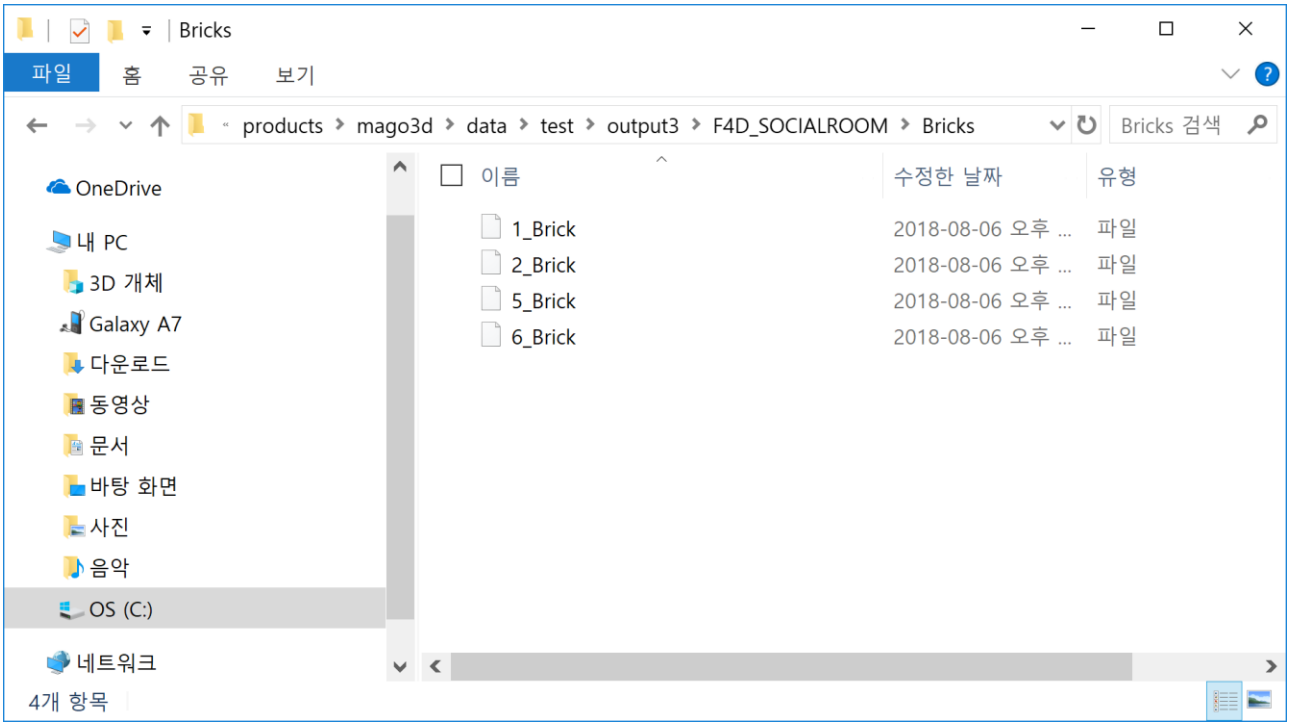

Figure 4 Sample "Bricks" folder of depth 1 octree. Octree 4, 7, and 8 are removed.

Overall structure of LOD2 files in “Bricks” folder is explained in Table 2.

Table 2 Overall Structure of LOD2 file

Name of the Field	Length of Bytes	Remarks
bbox	6 × sizeof(float) bytes	Bounding box of minX, minY, minZ, maxX, maxY, MaxZ
vertexCount	sizeof(unsigned int) bytes	No. of vertex in a Brick
vertexPositions	[vertexCount] × 3 × sizeof(float) bytes	x, y, z coordinates of each vertex (unit: m)
bNormal	sizeof(bool) bytes	Whether normal vectors exist or not
normalCount	sizeof(unsigned int) bytes	No. of normal vectors (same as No. of vertex count) Only record when bNormal == true
normalVectors	[normalCount] × 3 × sizeof(char) bytes	x, y, z components of normal vectors. Only record when bNormal == true
bColor	sizeof(bool) bytes	Record whether colour array is existing or not
colorCount	sizeof(unsigned int) bytes	No. of colours (same as No. of vertex count) Only record when bColour == true
vertexColors	[colorCount]×4×sizeof(unsigned char) bytes	Colour values (red, green, blue and alpha) of each vertex. Only record when bColour == true
bTextureCoord	sizeof(bool) bytes	Record whether texture coordinate array is existing or not
texgtureCoordType	sizeof(unsigned short) bytes	GL enumeration value for type of texture coordinate description. Now only 5126(float) type available. Only record when bTextureCoord == true
vertexCount	sizeof(unsigned int) bytes	No. of vertex count. Only record when bTextureCoord == true
textureCoordinates	[vertexCount] × 2 × sizeof(float)	u, v coordinate of each vertex. Only record when bTextureCoord == true

1.3 Models Folder

This folder contains original 3D objects information called Models that constitutes F4D itself. Models will be made to be an instance physically in the 3D space by pairing Reference information in Reference folder. Structure of F4D's Model information is shown in Table 3, 3-1, 3-2.

Table 3 Overall Structure of Models File

Name of the Field	Length of Bytes	Remarks
version	5 × sizeof(char) bytes	Version of F4D. latest version is 0.0.1
modelCount	sizeof(unsigned int) bytes	No. of Models that References in octree refers
modelData	$\sum_{i=1}^{modelCount} modelSize_i$ bytes	Data of each Model

Table 3-1 Detailed Structure of "modelData" in Table 3

Name of the Field	Length of Bytes	Remarks
modelIndex	sizeof(unsigned int) bytes	Unique ID of Model. This ID is unique number throughout the datasets
bbox	3 × sizeof(float) bytes	Bounding box of minX, minY, minZ, maxX, maxY, MaxZ
vboCount	sizeof(unsigned int) bytes	No. of VBO(Vertex Buffer Object) that constitutes Model
vboData	$\sum_{i=1}^{vboCount} vboDataSize_i$ bytes	Data of each VBO
dummy boolean	sizeof(bool) bytes	just false.

Table 3-2 Detailed Structure of "vboData" in Table 3-1

Name of the Field	Length of Bytes	Remarks
vertexCount	sizeof(unsigned int) bytes	No. of vertex that constitutes VBO
vertexPositions	[vertexCount] × 3 × sizeof(float) bytes	x, y, z coordinates of each vertex(unit: m)
normalCount	sizeof(unsigned int) bytes	No. of normal vectors(same as vertexCount)
normalVectors	[normalCount] × 3 × sizeof(char) bytes	Normal vectors of x, y, z
indexCount	sizeof(unsigned int) bytes	No. of index of vertex that constitutes triangle
sizeLevels	sizeof(unsigned char) bytes	No. of levels to be used when storing index of triangles in order of size
sizeThresholds	[sizeLevels] × sizeof(float) bytes	Delimiter of each segment
indexMarkers	[sizeLevels] × sizeof(unsigned int) bytes	The positions of the indexes whose size is first changed in the sorted triangle index array.
indexArray	[indexCount] × sizeof(unsigned short) bytes	Array that records indexes of vertex of each triangle

1.4 Reference Folder

This folder contains binary files of reference information such as position, ID, indexes of 3D objects that is constituting F4D. By combining Reference information and Models information, real 3D objects will be physically rendered on the described place with attributes (- called as “3D objects are instantiated.”). Reference files are created along spatial octrees so that the name of a reference file is “[octreeNumber]_ref”.

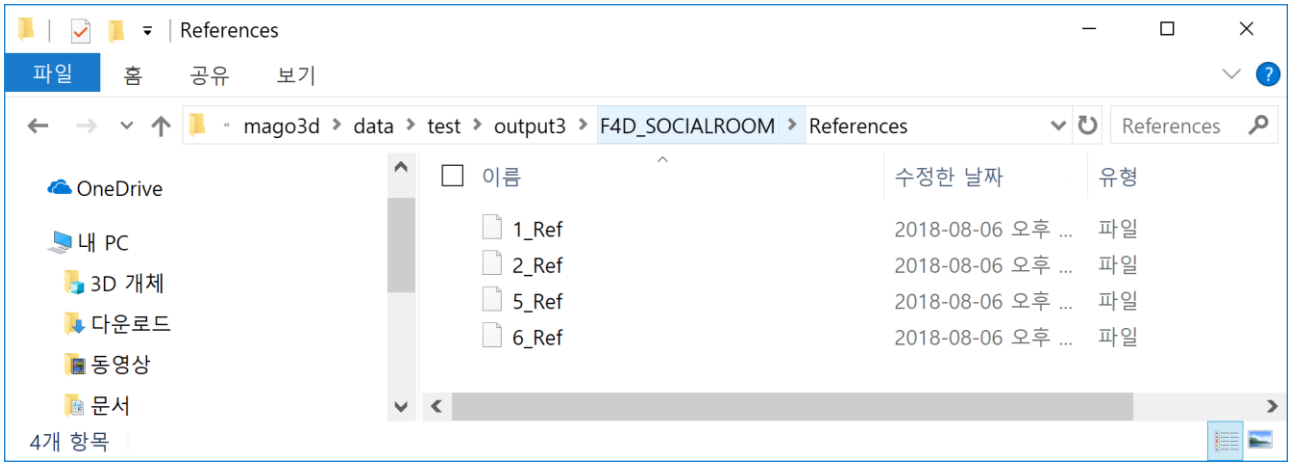

Figure 5 Sample “Reference” folder of depth 1 octree. Octree 4, 7, and 8 are removed.

Structure of F4D’s Reference information is shown in Table 4, 4-1, 4-2, 4-3.

Table 4 Overall Structure of Reference File

Name of the Field	Length of Bytes	Remarks
version	5 × sizeof(char) bytes	Version of F4D. latest version is 0.0.1
referenceCount	sizeof(unsigned int) bytes	No. of reference in the octree
referenceData	$\sum_{i=0}^{referenceCount} referenceDataSize_i$ bytes	Data of each reference
triangleCount	sizeof(unsigned int) bytes	Total triangle count in this reference
exteriorVisibilityIndices	flexible	Visibility indices for exterior occlusion culling. Written recursively for all vision octrees
interiorVisibilityIndices	flexible	Visibility indices for interior occlusion culling. Written recursively for all vision octrees

Table 4-1 Detailed Structure of “referenceData” in Table 4

Name of the Field	Length of Bytes	Remarks
referenceIndex	sizeof(unsigned int) bytes	Unique ID of reference. This ID is unique number throughout the datasets
objectIdLength	sizeof(unsigned char) bytes	Length of object ID
objectId	[objectIdLength] ×sizeof(char) bytes	Object ID
modelIndex	sizeof(unsigned int) bytes	Unique ID of original model referred by reference information
transformMatrixType	sizeof(unsigned char) bytes	Transform matrix type. 0 is for identity, 1 is for translation so that only 3 floating numbers are written for transform matrix, and 2 is for translation & rotation so that full 16 floating numbers are written for transform matrix.
transformMatrix	- transformMatrixType == 0	- transformMatrixType == 0

	Not write anything - transformMatrixType == 1 3 × sizeof(float) bytes - transformMatrixType == 2 16 × sizeof(float) bytes	Identity matrix - transformMatrixType == 1 3 elements of transform matrix - transformMatrixType == 2 16 elements of transform matrix
bSingleColor	sizeof(bool) bytes	whether a representative color is available
singleColorValueType	sizeof(unsigned short) bytes	Default colour value type in GL enumeration for color value. Now only 5121(unsigned byte) supported. Only record when bSingleColor == true
singleColorDimension	sizeof(unsigned char) bytes	Channel count of default color. Now fixed as 4(RGBA). Only record when bSingleColor == true
singleColor	[colorDimension1] × sizeof(unsigned char) bytes	default colour value of this reference
bVertexColor	sizeof(bool) bytes	Record whether vertex colours are available
bTextureCoord	sizeof(bool) bytes	Record whether texture coordinates are available
vboCount	sizeof(unsigned int) bytes	Count of VBO(vertex buffer object).
vertexColorAndTextureCoordinateInfo	∑ sizeOfvertexColor&textureCoord	vertex colours and texture coordinates of this reference on each vbo
textureIndex	sizeof(int) bytes	index to a texture file name used by this reference in the list written in HeaderAsimetric.hed. -1 means not using any texture file.

Table 4-2 Detailed Structure of “vertexColorAndTextureCoordinateInfo” in Table 4-1

Name of the Field	Length of Bytes	Remarks
vertexColorType	sizeof(unsigned short) bytes	GL enumeration value for color. Now only 5121 supported. Only record when bVertexColor in Table 4-1 == true
vertexColorDimension	sizeof(unsigned char) bytes	Channel count of vertex colour. Now fixed as 3. Only record when bVertexColor in Table 4-1 == true
vertexCount	sizeof(unsigned int) bytes	Vertex count in this vbo. Only record when bVertexColor in Table 4-1 == true
vertexColors	[vertexCount] × vertexColorDimension × sizeof(unsigned char) bytes	Vertex color of each vertex. Only record when bVertexColor in Table 4-1 == true
textureCoordinateValueType	sizeof(unsigned short) bytes	GL enumeration value for texture coordinate. Now firxed as 5126. Only record when bTextureCoord in Table 4-1 == true
vertexCount	sizeof(unsigned int) bytes	Vertex count in this vbo. Only record when bTextureCoord in Table 4-1 == true
textureCoordinates	[vertexCount] × 2 × sizeof(float) bytes	u, v coordinate of each vertex. Only record when bTextureCoord in Table 4-1 == true

Table 4-3 Detailed Structure of “exteriorVisibilityIndices” and “interiorVisibilityIndices” in Table 4

Name of the Field	Length of Bytes	Remarks
bRoot	sizeof(bool) bytes	Whether this vision octree is root
bbox	sizeof(float) × 6 bytes	minX, maxX, minY, maxY, minZ, maxZ of this vision octree. Only record when bRoot == true (Be cautious of the order of saving. Different with that of bounding box in the file “HeaderAsimetric.hed”)
childCount	sizeof(unsigned int) bytes	Count of available children of this vision

		octree
referenceCount	sizeof(unsigned int) bytes	No. of references which can be seen from this vision octree. Only record when childCount == 0(that is, this octree is a leaf octree)
referenceIndices	[referenceCount] × sizeof(unsigned int) bytes	Indices of references which can be seen from this vision octree. Only record when childCount == 0(that is, this octree is a leaf octree)
childVisionOctreeInfo	flexible	Record contents of this table recursively. Only record when childCount > 0

1.5 Images_Resized folder

This folder is for image files for texture mapping. Files in this folder are created only when raw data has images for texture. Because WebGL accepts texture images only with width and height of power of 2 pixels, resized texture images are created if necessary.

1.6 lod[lod] (lod : 3, 4, 5)

These files are geometric data for LOD 3~5. Data structure of these files are exactly same as files in “Bricks” folder, that is, LOD2 data file.

1.7 MosaicTextureLOD[lod].png (lod : 2, 3, 4, 5)

This files are for texture mapping of LOD 2~5.

1.8 objectIndexFile.ihe

As described in “1. Format structure”, this file is metadata of full set of all F4D data on web service. Structure of objectIndexFile.ihe is shown in Table 10 and 11.

Table 5 Overall Structure of Reference File

Name of the Field	Length of Bytes	Remarks
dataFolderCount	sizeof(unsigned int) bytes	Count of F4D folders on service
eachDataFolderInfo	∑ dataFolderInfoSize	Simple information of each F4D data folder.

Table 5-1 Detailed Structure of “eachDataFolderInfo” in Table 5

Name of the Field	Length of Bytes	Remarks
dataFolderNameLength	sizeof(unsigned int) bytes	Length of this F4D data folder name
dataFolderName	[dataFolderNameLength] × sizeof(char) bytes	Name of this F4D data folder.
lonLatAlt	sizeof(double) + sizeof(double) + sizeof(float) bytes	Representative longitude, latitude, and altitude of this F4D(Deprecated)
boundingBox	sizeof(float) × 6 bytes	minX, minY, minZ, maxX, maxY, maxZ of bounding box of this F4D data