LCC Data Organization Format White Paper

(1) Terms and Concepts

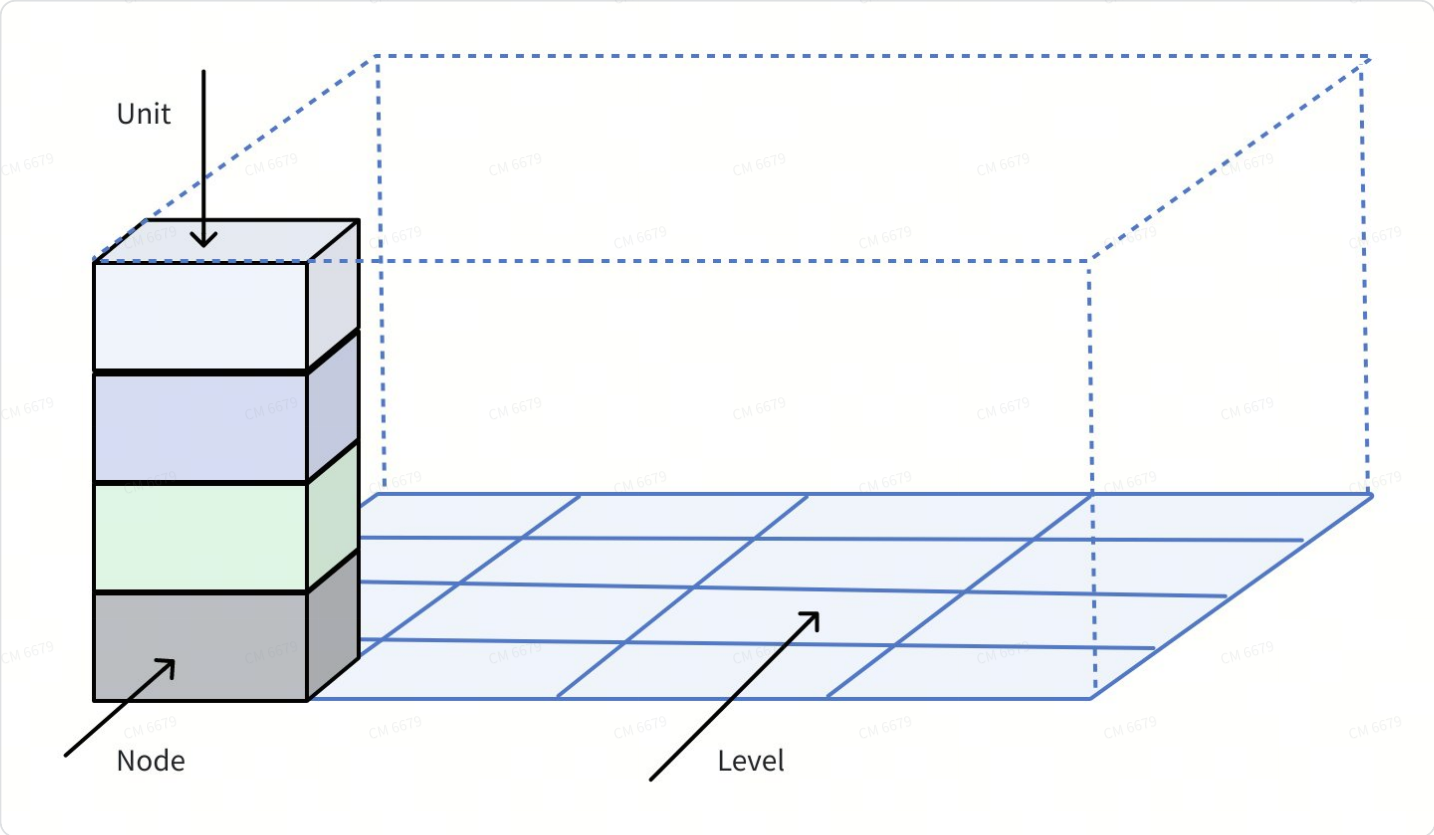

Figure 1 Schematic Diagram of Data Organizational Structure

Node: A chunk at a specified LOD level

Index: Used to index the x and y index values of a partition chunk, represented by a Uint32 value, with the x value in the lower 16 bits and the y value in the upper 16 bits

Unit: refers to a group of Nodes corresponding to the same Index

Level: One LOD level

The coordinate system where the data is located is shown in the figure below:

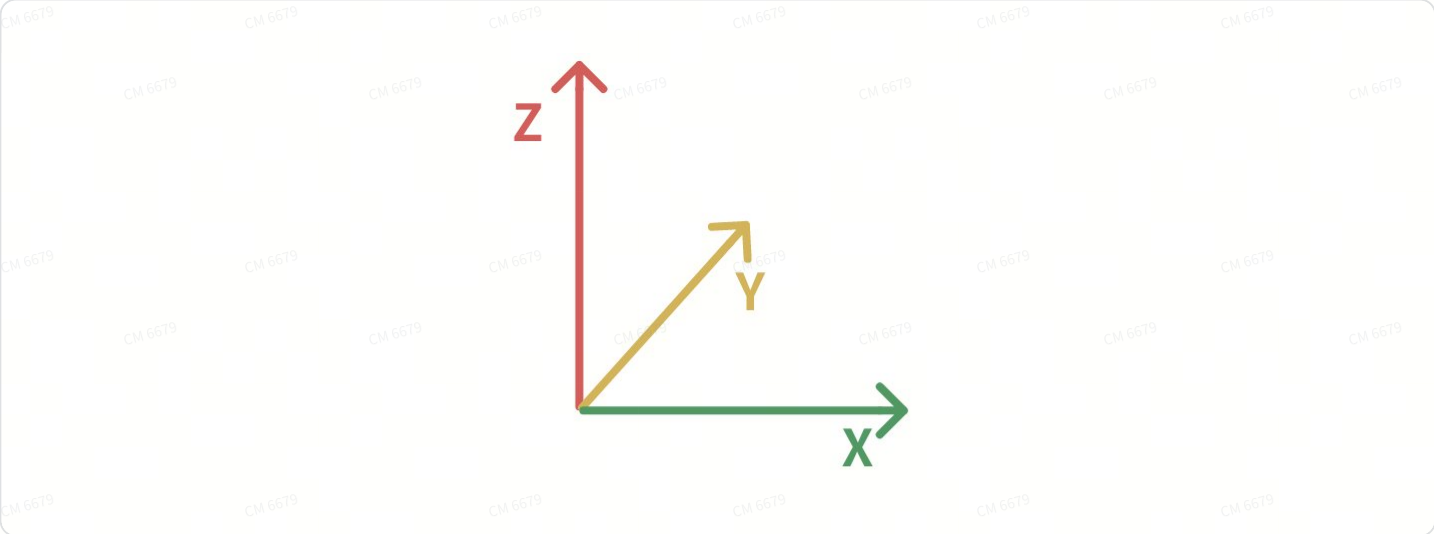

Figure 2 Data Coordinate System

(2) Data Organization

The complete LCC data consists of the following files: meta.lcc, Index.bin, Data.bin, Shcoef.bin, Collision.lci, Environment.bin

File Name	Description	Optional	Remarks
meta.lcc	Metadata description file, providing an overall description of the scenario data	Required	The file name of this file can be modified at will
Index.bin	Node Index File	Required	
Data.bin	LCC Basic Data File	Required	
Shcoef.bin	Spherical Harmonics Data File	Optional	
Environment.bin	Environmental data file	Optional	
Collision.lci	collision mesh file	Optional	

LCC data supports 0th and 3rd order spherical harmonics, which are specified by the **fileType** attribute in the meta.lcc file. If this attribute is **Portable**, it represents 0th order spherical harmonics, and there is no Shcoef.bin file in this case. If this attribute is **Quality**, it represents 3rd order spherical harmonics, and there is a Shcoef.bin file in this case.

Environment.bin is background environment data, optional.

Collision.lci is a collision mesh file that stores the original Mesh vertex and face information, as well as serialized data of the Mesh BVH acceleration structure partitioned in the same way as the scene data, which is used for chunked loading.

(a) Meta.lcc File

Metadata describes data such as the total number of Splats, LOD levels, index block size, etc. Specific examples are as follows:

代码块

```
{
        "version": "5.0",                       //version
        "guid":"jfdskli45sfdsfdsf22d22fdd",
        "name": "XGrids Splats",
        "description": "XGrids all right reserved",
        "source":"lcc",                         //source: lcc、ply、splat、none
        "dataType":"L1",                        //device: L1、L2、K1、Drone、Drone_L2、PortCam、None
        "totalSplats": 3678719,                 //total Splat count ,include all lods
        "totalLevel": 9,                        //lod count
        "cellLengthX": 15.33,                   //Node x length, unit: metre
        "cellLengthY": 14.33,                   //Node y length, unit: metre
        "indexDataSize": 86,                    //Unit index bytes size
        "offset": [0, 0, 0],                    //global offset
        "epsg":0                                //global epsg
        "shift": [0, 0, 0],                     //global shift
        "scale": [1, 1, 1],                     //global scale
        "splats": [2890271,1343679,632205,300751,144410,70149,34299,16759,8142], //each lod splats count, from left to right :LOD0、LOD1、LOD2 ...
        "boundingBox": {                        //BoundingBox without environment data
                "min": [-172.99162292480469, -135.55508422851562, -10.660738945007324],
                "max": [59.935356140136719, 97.371894836425781, 222.26624011993408]
        },
        "encoding": "COMPRESS",                 //data compress, current version is COMPRESS
        "fileType":"Portable",                  //Portable: only RGB; Quality:RGB+SH
        "attributes": [
                {
                        "name": "position",     //BoundingBox with environment data
                        "min": [-172.99162292480469, -135.55508422851562, -10.660738945007324],
                        "max": [59.935356140136719, 68.578681945800781, 18.81260871887207]
                },
```

```
                {
                    "name": "normal",           //normal min max
                     "min": [-172.99162292480469, -135.55508422851562,
-10.660738945007324],
                     "max": [59.935356140136719, 68.578681945800781,
18.81260871887207]
                },
                {
                    "name": "color",            //color min max
                     "min": [-172.99162292480469, -135.55508422851562,
-10.660738945007324],
                     "max": [59.935356140136719, 68.578681945800781,
18.81260871887207]
                },
                {
                    "name": "shcoef",           //sh min max
                     "min": [-172.99162292480469, -135.55508422851562,
-10.660738945007324],
                     "max": [59.935356140136719, 68.578681945800781,
18.81260871887207]
                },
                {
                    "name": "opacity",          //opacity min max
                    "min": [-172.99162292480469],
                    "max": [59.935356140136719]
                },
                {
                    "name": "scale",            //scale min max
                    "min": [-172.99162292480469, -135.55508422851562,
-10.660738945007324],
                    "max": [59.935356140136719, 68.578681945800781,
18.81260871887207]
                },      //environment data
                {
                    "name": "envnormal",        //normal min max
                     "min": [-172.99162292480469, -135.55508422851562,
-10.660738945007324],
                     "max": [59.935356140136719, 68.578681945800781,
18.81260871887207]
                },
                {
                    "name": "envshcoef",         //sh min max
                     "min": [-172.99162292480469, -135.55508422851562,
-10.660738945007324],
                     "max": [59.935356140136719, 68.578681945800781,
18.81260871887207]
                },
```

```
                {
                    "name": "envscale",          //scale min max
                    "min": [-172.99162292480469, -135.55508422851562,
-10.660738945007324],
                    "max": [59.935356140136719, 68.578681945800781,
18.81260871887207]
                }
        ]
}

```

(b) Index.bin File

The index data Index is the index for each **Node** chunk. The length of this part in a specific scene data is fixed (described by the **indexDataSize** attribute in the Meta.lcc file), but the lengths of different scene data may vary (related to LOD division), using little-endian storage, as follows:

Attribute	ByteDance	Type	Remarks
index	4	uint32	Index of Unit
PointsCount0	4	uint32	Total number of splats in LOD0
LOD0Offset	8	uint64	Offset of LOD0 data stored in the **Data.bin** file
LOD0Size	4	uint32	Total ByteSize of LOD0 Data
PointsCount1	4	uint32	Total number of splats in LOD1
LOD1Offset	8	uint64	Offset of LOD1 data stored in the **Data.bin** file
LOD1Size	4	uint32	Total ByteSize of LOD1 Data
...			

(c) Data.bin Data Section

The file of **Data.bin** is binary data, which sequentially stores the basic data, using little-endian storage. The data storage structure is as follows:

Attribute	Component	ByteDance	Type	Remarks
Postion	X	4	float	

	Y	4	float	
	Z	4	float	
Color	RGBA	4	uint32	R: uint8, G: uint8, B: uint8, A: uint8
Scale	X	2	uint16	
	Y	2	uint16	
	Z	2	uint16	
Rotation	xyzw	4	uint32	
Normal	X	2	uint16	
	Y	2	uint16	
	Z	2	uint16	
...				

(d) Shcoef.bin Data Section

The file of **Shcoef.bin** is spherical harmonic coefficients binary data, using little-endian storage, and the data storage structure is as follows:

Attribute	Component	ByteDance	Type	Remarks
SH	xyzxyzxyz...	64	uint32uint32...	
	xyzxyzxyz...	64	uint32uint32...	
...				

(e) Environment.bin Data Section

Since the amount of environment data is relatively small, its basic data and spherical harmonic coefficients are combined, with only one **environment.bin** file, using little-endian storage:

Attribute	Component	ByteDance	Type	Remarks
Postion	X	4	float	
	Y	4	float	

	Z	4	float	
Color	RGBA	4	uint32	R: uint8, G: uint8, B: uint8, A: uint8
Scale	X	2	uint16	
	Y	2	uint16	
	Z	2	uint16	
Rotaion	xyzw	4	uint32	
Normal	X	2	uint16	
	Y	2	uint16	
	Z	2	uint16	
SHcoef	xyzxyzxyz...	64	uint32uint32...	
...				

(f) Collision.lci Data Section

Collision.lci is a collision file, which consists of two parts: `header + data`.

- The header, total length is controlled by **headerLen**, contains metadata information and several meshHeaders;
- data consists of several mesh data, each of which contains vertex and triangular face index data;

Data is stored in little-endian format, and the complete file format is:

Field		Data Format	Description	bytesOffset
header	magic	uint32	File identifier, fixed value 0x6c6c6f63	0
	version	uint32	Version number, the current version number is 2 (for protocol extension)	4
	headerLen	uint32		8

					Total length of file header data (number of bytes)	
	min		x	float32	Scene Bounding Box Min Value	12
			y	float32		16
			z	float32		20
	max		x	float32	Scene Bounding Box Max Value	24
			y	float32		28
			z	float32		32
	cellLengthX			float32	Length of the chunk in the x-direction	36
	cellLengthY			float32	Length of the chunk in the y-direction	40
	meshNum			uint32	Mesh Count	44
	meshHeader0 (40)	indexX		uint32	X Index	48 - 0
		indexY		uint32	Y Index	4
		offset		uint64	Mesh Data Offset	8
		bytesSize（包括BVH）		uint64	Mesh Data Length	16
		vertexNum		uint32	Number of Vertices	24
		faceNum		uint32	Triangle Count	28
		bvhSize		uint32	BVH Data Length (This field is 0 in Version 1)	32
		reserved1		uint32	Reserved (default value is 0)	36
	...			...	...	
data	mesh0	vertexes	x	float32	Vertex data, where each vertex contains three values: x, y, and z	
			y	float32		
			z	float32		

			...	...		
		faces	f0	uint32	Triangular face data, where each face contains three index values (index value starts from 0)	
			f1	uint32		
			f2	uint32		
			...	...		
		bvh	reserved0	uint32	Reserved (default value is 0)	
			reserved1	uint32	Reserved (default value is 0)	
			reserved2	uint32	Reserved (default value is 0)	
			reserved3	uint32	Reserved (default value is 0)	
			data	uint8[]	BVH Acceleration Structure Data	
	...			...	...	

The Collision.lci file mainly includes two types of data: one is the **segmented Mesh data,** where the Mesh segmentation is consistent with that of the lcc scene Data; the other is the preprocessed BVH acceleration structure of the segmented data, which is mainly used for fast collision testing. The Mesh data can be read for rendering and display, or it can be left unread.

BVH acceleration structure data is the serialized result of the "preorder traversal" of a binary tree, with each node occupying 32 bytes. Depending on the use case, there are two types of nodes, namely internal nodes and leaf nodes. Internal nodes store information such as bounding box information and position for retrieval; leaf nodes store the starting index of triangular faces, the number of triangular faces, and bounding box information for obtaining triangular face data.

Internal Node Data Format:

Field	Data Format	Description	Byte Offset
boundingBox	float32 * 6	Boundingbox minx,miny,minz,maxx,maxy,maxz	0
right	uint32		24

		The starting position of the right child node, where the position information is relative to the current BVH acceleration structure data, with the address aligned to 4 bytes; for example, 32 indicates that the byte offset of the right child node is 128	
splitAxis	uint16	0: Divide the x-axis 1: Divide the y-axis 2: Divide the z-axis	28
flag	uint16	0xFFFF: Leaf node, other values: Internal node	30

Leaf node data format:

Field	Data Format	Description	Byte Offset
boundingBox	float32 * 6	Boundingbox minx,miny,minz,maxx,maxy,maxz	0
faceOffset	uint32	Starting triangular face number, e.g., 0, 1, 2,...	24
faceCount	uint16	Number of triangular faces contained in the node	28
flag	uint16	0xFFFF: Leaf node, other values: Intermediate node	30

Example:

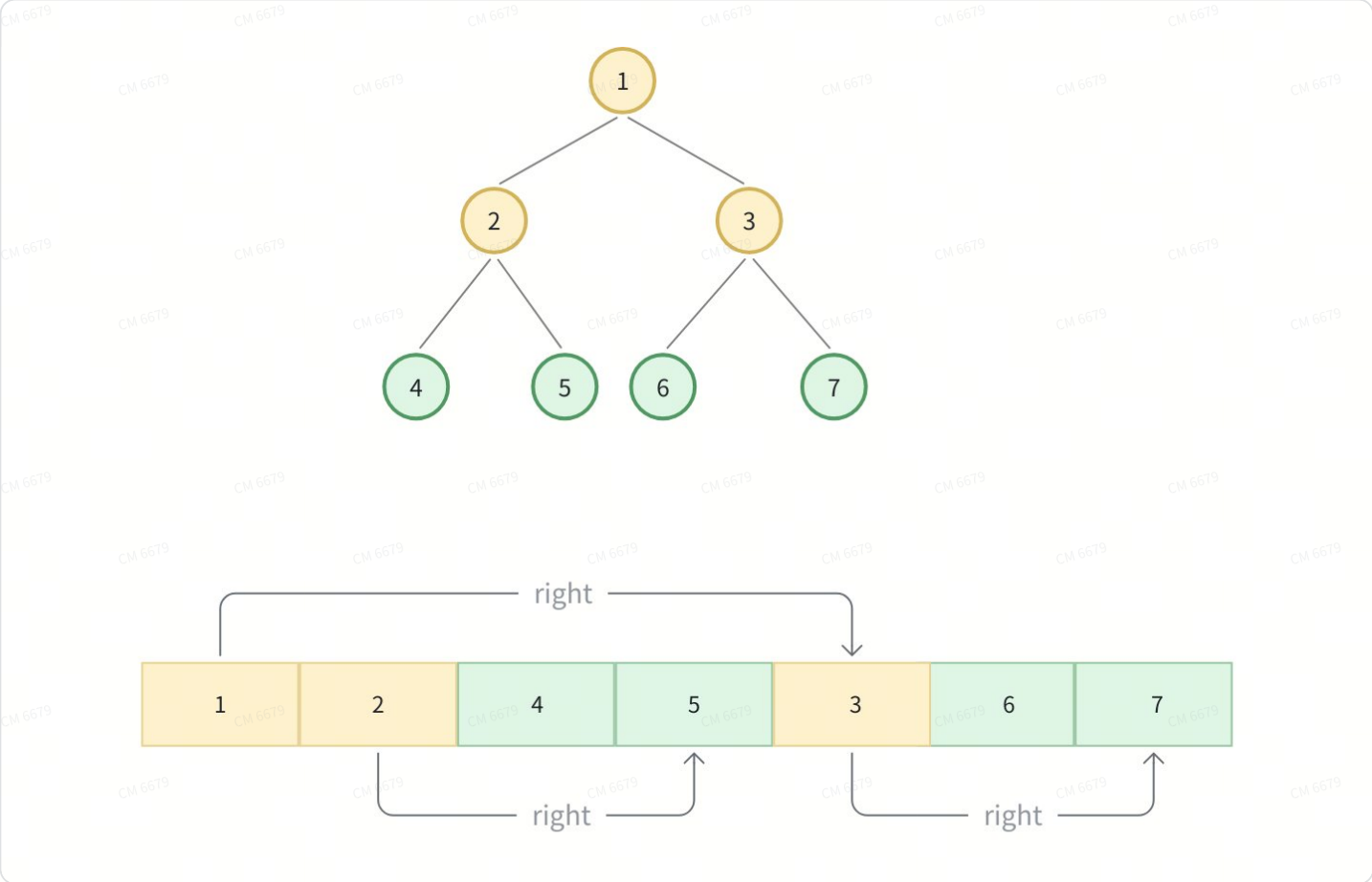

Figure 4 Node Data Organizational Structure

(3) Data Reading and Parsing

To utilize memory and video memory bandwidth more efficiently, it is not recommended to parse data in memory. Instead, pass the data through to video memory for parsing. Data parsing is not complex, and GPUs are more efficient at parsing data while consuming less memory and video memory.

(a) Reading and parsing of Meta files and Index files

Meta.lcc is in JSON format and can be directly read;

Index.bin is a binary file with a small amount of data. It is recommended to read it all at once, then parse and create the LOD and Node node structures.

After reading the **Index.bin** file once, obtain the total file size, divide it by **indexDataSize** to get the total number of Units, iterate through all Units to initialize each level of Level and Node. The **BoundingBox** of a Unit starts from **boundingBox.min** in the meta.lcc file, and the Unit size is (**cellLengthX**, **cellLengthY**).

(b) Position Data Parsing

Each component of Position XYZ is a float, requiring no processing and can be used directly.

(c) Scale Data Parsing

Each **scale** occupies a total of 6 Bytesize. First, extract the xyz component data and convert it to Int32, then interpolate the available **scale** value through the min and max of the **Scale** item in the **attributes** of the meta.lcc file.

(d) Rotation Data Parsing

Rotation is of type Uint32, and this data uses a special compression method. The parsing method is as follows:

代码块

```
static const int QLut[16] = { 3, 0, 1, 2,  0, 3, 1, 2, 0, 1, 3, 2, 0, 1, 2, 3};
static const float sqrt2 = 1.414213562373095;
static const float rsqrt2 = 0.7071067811865475;

float4 DecodeRotation(uint enc)
{
    float4 pq = float4(
        (enc & 1023) / 1023.0,
        ((enc >> 10) & 1023) / 1023.0,
        ((enc >> 20) & 1023) / 1023.0,
        ((enc >> 30) & 3) / 3.0);
    uint idx = (uint) round(pq.w * 3.0);
    float4 q;
    q.xyz = pq.xyz * sqrt2 - rsqrt2;
    q.w = sqrt(1.0 - saturate(dot(q.xyz, q.xyz)));
    float4 p = float4(q[QLut[idx * 4]], q[QLut[idx * 4 + 1]], q[QLut[idx * 4 + 2]], q[QLut[idx * 4 + 3]]);
    return p;
}
```

(e) Color Data Parsing

Color is of type Uint32, stored in RGBA format. After separating each channel, the values need to be divided by 255 to convert them to the range [0, 1].

(f) Shcoef Spherical Harmonic Data Parsing

Whether there is spherical harmonic data is specified by the **fileType** attribute in the **Meta.lcc** file. If it is **Portable**, there is no spherical harmonic data (no Shcoef.bin file); if it is **Quality**, there is spherical harmonic data. The spherical harmonic offset corresponding to the certain Node and is 2 times its **data.bin** offset, and the **dataSize** is also 2 times the Node

Size recorded in the **index.bin** file, i.e., each Splat primitive includes 32 Bytesize of basic data and 64 Bytesoze of spherical harmonic data.

Each spherical harmonic data is 64 Bytesize in total. After decompression, the spherical harmonic data is interpolated using the shcoef min max in the Meta.lcc file. The pseudocode is as follows:

代码块

```
half3 DecodePacked11(uint enc)
{
    return half3(
        (enc & 2047) / 2047.0,
        ((enc >> 11) & 1023) / 1023.0,
        ((enc >> 21) & 2047) / 2047.0);
}

void DecodeSH(int idx)
{
    uint _shOffset = idx * 64;
    uint4 shRaw0 = _SplatSH.Load4(_shOffset + 0);
    uint4 shRaw1 = _SplatSH.Load4(_shOffset + 16);
    uint4 shRaw2 = _SplatSH.Load4(_shOffset + 32);
    uint3 shRaw3 = _SplatSH.Load3(_shOffset + 48);
    s.sh1 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz, DecodePacked11(shRaw0.x));
    s.sh2 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz, DecodePacked11(shRaw0.y));
    s.sh3 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz, DecodePacked11(shRaw0.z));
    s.sh4 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz, DecodePacked11(shRaw0.w));
    s.sh5 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz, DecodePacked11(shRaw1.x));
    s.sh6 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz, DecodePacked11(shRaw1.y));
    s.sh7 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz, DecodePacked11(shRaw1.z));
    s.sh8 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz, DecodePacked11(shRaw1.w));
    s.sh9 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz, DecodePacked11(shRaw2.x));
    s.sh10 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz, DecodePacked11(shRaw2.y));
    s.sh11 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz, DecodePacked11(shRaw2.z));
```

```
27      s.sh12 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz,
    DecodePacked11(shRaw2.w));
28      s.sh13 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz,
    DecodePacked11(shRaw3.x));
29      s.sh14 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz,
    DecodePacked11(shRaw3.y));
30      s.sh15 = lerp(_AttrInfo[0].xyz, _AttrInfo[1].xyz,
    DecodePacked11(shRaw3.z));
31  }
32
```

_AttrInfo[0].xyz represents the upper and lower limits of **shcoef min max** in the attributes property of the meta.lcc file.

(g) Environment Data Parsing

The basic data and spherical harmonic data of the environment are stored together. Whether there is spherical harmonic data is specified by the **fileType** attribute in the **Meta.lcc** file. If it is **Portable**, there is no spherical harmonic data part; if it is **Quality**, there is spherical harmonic data.

The decompression method for Envrionment basic data and spherical harmonic data is the same as described above, with the only thing to note being that the upper and lower limits of the interpolation values are the environmental data section of the attributes part in **Meta.lcc**.

(4) License and Restrictions

(a) Subject to your full compliance with this White Paper, we/XGRIDS (meaning XGRIDS LIMITED and its Affiliates, where “Affiliate” refers to any entity that directly or indirectly controls, is controlled by, or is under common control with a Party, and which maintains a direct or indirect interest relationship therewith) hereby grant you a non-exclusive, non-transferable, and royalty-free limited license to use, reproduce, modify, distribute (including provision to third parties, but excluding assignment), and create derivative data organization formats or other works based on the LCC Data Organization Format (hereinafter the “**Data Organization Format**”), provided that you satisfy all of the following conditions:

(i) You shall display a clear and prominent attribution stating “Data Organization Format originated from XGRIDS” within your application, website, or other visible interface of the product developed using the Data Organization Format.

(ii) You shall include a conspicuous notice in all modified data organization formats or derivative content stating that you have made modifications to the original Data Organization Format.

(iii) You shall provide an accessible electronic link or a copy of this White Paper to all third parties receiving the Data Organization Format or any derivative thereof.

(iv) Any redistribution to third parties must include a notice stating that the individual or organization uses and distributes the Data Organization Format under authorization from XGRIDS pursuant to this White Paper, and that XGRIDS or its Affiliates own and retain all intellectual property and other rights in the Data Organization Format.

(v) Without the prior written consent of XGRIDS, you shall not use the Data Organization Format for training or fine-tuning any artificial intelligence model that competes, directly or indirectly, with XGRIDS' products or services. You shall inform all third parties to whom you distribute the Data Organization Format of this restriction in writing and shall incorporate this clause into any applicable agreement (including but not limited to license agreements or terms of use) governing the use and/or distribution of the Data Organization Format.

(vi) You shall comply with this White Paper and all applicable laws and regulations.

(b) Subject to your compliance with this White Paper, any modification, extension, reprocessing, or derivative data organization format or content created based on the Data Organization Format shall be made publicly available under terms no less open than those of this White Paper License, and you shall indicate the source and licensing information of the Data Organization Format.

(c) Failure to perform or a breach of the foregoing open-source obligations shall automatically terminate all rights granted to you under this White Paper as of the date of such failure or breach.

(d) Subject to your compliance with this White Paper, you may use, reproduce, or distribute the Data Organization Format and its derivative forms in accordance with this White Paper; provided, however, that such use, reproduction, or distribution shall not be deemed a waiver, assignment, or limitation of any existing rights of XGRIDS.

(e) This White Paper does not grant any trademark license. Licensees shall not use any name or logo owned by or associated with XGRIDS or its Affiliates, except to the extent reasonably and customarily necessary for the description and distribution of the Data Organization Format.

(f) You shall not use the Data Organization Format or its derivatives in any of the following ways:

(i) In violation of any applicable international, national/federal, local, or other applicable laws or regulations;

(ii) To develop, train, test, or support any model or system that directly or indirectly causes harm, discrimination, misinformation, or any unlawful purpose;

(iii) To develop, train, test, or support any model, system, or application intended to exploit, harm, or potentially harm minors;

(iv) To support high-risk automated decision-making systems (including but not limited to those involving personal safety, health, employment, credit, justice, or education), or for any unlicensed professional use;

(v) In a manner that violates generally accepted social ethics or public order;

(vi) To implement, support, or promote violence, extremism, or terrorism;

(vii) For any discriminatory purpose based on race, gender, religion, nationality, disability, age, or any other legally protected characteristic;

(viii) For any military or weapons development purpose;

(ix) To identify, de-anonymize, or recover any personal data, confidential information, or sensitive content that may be present in the Data Organization Format;

(x) In any manner that damages or may damage the rights or interests of XGRIDS.

(g) If you initiate or participate in any lawsuit, arbitration, or other legal proceeding against XGRIDS or any party, alleging that XGRIDS has infringed upon your rights or interests, all rights granted to you under this White Paper shall automatically terminate as of the date such legal action is initiated.

(5) Miscellaneous

(a) XGRIDS shall have no obligation to provide any support, maintenance, updates, training, or to develop any subsequent versions of the Data Organization Format, nor shall XGRIDS be obligated to grant any further licenses related thereto. Unless and only to the extent required by applicable law, the Data Organization Format and any outputs or related results are provided on an "AS IS" basis, without any express or implied warranties, including but not limited to warranties of title, merchantability, non-infringement, fitness for a particular purpose, or those arising from a course of dealing, usage, or trade practice. You are solely responsible for determining the appropriateness of using, reproducing, modifying, performing, displaying, or distributing the Data Organization Format or any output derived therefrom, and you assume all risks associated with your and any third party's use, distribution, or exercise of rights and licenses under this White Paper.

(b) Any implementation, tool, model, or service developed, released, or distributed by any third party based on or derived from the Data Organization Format or its structure shall be deemed the independent action of such third party. Such actions shall not represent the views of XGRIDS, nor shall they constitute an official version, authorization, or endorsement by XGRIDS. XGRIDS makes no representations or warranties, and assumes no liability, regarding the performance, compatibility, legality, or fitness for any purpose of any such third-party implementations.

(c) In the event that any third party makes a claim, initiates litigation, arbitration, or other legal proceedings against XGRIDS arising out of or in connection with your or your authorized third party's use, modification, redistribution, or derivative application of the Data Organization Format, you shall provide necessary assistance (or cause your authorized third parties to assist) in the defense of such proceedings and shall hold XGRIDS harmless from and against any and all liabilities, losses, damages, or expenses arising therefrom.

(d) To the maximum extent permitted by applicable law and regulation, and regardless of the theory of liability (including contract, tort, negligence, product liability, or otherwise), XGRIDS shall not be liable for any damages arising out of or in connection with this White Paper or the Data Organization Format, including without limitation any direct, indirect, special, incidental, punitive, or consequential damages, or any loss of profits, revenues, data, or goodwill.

(e) This White Paper, and any dispute arising out of or in connection with it, shall be governed by and construed in accordance with the laws of the People’ s Republic of China (Mainland), without regard to its conflict of laws principles. Any dispute arising out of or relating to this White Paper shall be submitted to the Shenzhen Court of International Arbitration (SCIA) for arbitration in Shenzhen, China, in the Chinese language. The arbitral award shall be final and binding upon the parties.

(f) XGRIDS reserves the right to update, revise, or interpret this White Paper at any time. Any updated version shall take effect upon its publication on the official XGRIDS website or other official channels, or upon written notice (including by email or other accessible means) provided to you by XGRIDS.