

团 体 标 准

T/CAGIS 1—2019

空间三维模型数据格式

Data format for spatial 3D model

2019-07-19 发布

2019-08-31 实施

中国地理信息产业协会 发布

目 次

前言	III
引言	IV
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 符号和缩略语	2
4.1 缩略语	2
4.2 UML 图示符号	2
4.3 UML 多样性描述	3
5 基本规定	3
5.1 基本数据类型	3
5.2 字符串类型	4
5.3 json 格式存储	4
6 组织结构	4
6.1 文件组织	4
6.2 树形结构描述	5
7 存储格式	5
7.1 描述文件	5
7.2 数据文件	8
7.3 索引树文件	21
7.4 属性文件	22
附录 A (资料性附录) 数据示例	25
参考文献	32

前 言

本标准按照 GB/T 1.1—2009 给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准由中国地理信息产业协会提出并归口。

本标准起草单位：自然资源部信息中心、国家基础地理信息中心、北京超图软件股份有限公司、中国城市规划设计研究院、中国建筑科学研究院有限公司、中国建筑标准设计研究院有限公司、中设数字技术股份有限公司、中建工程研究院有限公司、河北省第三测绘院、广州市红鹏直升机遥感科技有限公司、广州都市圈网络科技有限公司、成都软易达信息技术有限公司。

本标准主要起草人：钟耳顺、宋关福、冯振华、李绍俊、周芹、刘聚海、高崧、贾文珏、张敬波、徐辉、杨滔、马恩成、魏来、于洁、李云贵、徐鹏、孙伟、蔡文文、李濛、何倩。

引 言

近年来,倾斜摄影建模、激光扫描等数据采集技术的发展,有效降低了三维空间数据的获取成本和时间周期,提高了数据精度。伴随大规模三维空间数据的不断积累,三维空间数据的高效发布、数据共享和数据交换,成为三维 GIS 研究重要内容。

本标准定义了一种开放式可扩展的空间三维模型数据格式——Spatial 3D Model(S3M),适用于空间三维模型数据的传输、交换与共享,有助于解决多源空间三维模型数据在不同终端(移动设备、浏览器、桌面电脑)地理信息平台中的存储、高效可视化、共享与互操作等难题,对于推动我国三维地理空间数据的共享及深入应用具有重要作用。

目前,S3M 涵盖的数据类型包括:

- a) 传统模型:人工建模三维模型数据;
- b) 实景三维:包括倾斜摄影建模数据和点云数据;
- c) 建筑信息模型(Building Information Modeling, BIM)数据:采用 BIM 设计软件制作的三维模型数据;
- d) 矢量数据:包括二维点/线/面数据、三维点/线/面数据、三维管线数据。

空间三维模型数据格式

1 范围

本标准规定了一种空间三维模型数据格式的文件组织结构及存储格式要求。

本标准适用于网络环境和离线环境下三维空间数据的传输、交换与共享,也适用于三维空间数据在不同终端(移动设备、浏览器、桌面电脑)上的三维地理信息系统相关应用。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 7408—2005 数据元和交换格式 信息交换 日期和时间表示法

GB/T 16831—2013 基于坐标的地理点位置标准表示法

GB/T 30320—2013 地理空间数据库访问接口

GB/T 33187.1—2016 地理信息 简单要素访问 第1部分:通用架构

3 术语和定义

下列术语和定义适用于本文件。

3.1

瓦片 tile

对应一个二维矩形或三维长方体空间,邻接瓦片之间空间范围可重叠。

3.2

根节点 root node

TileTree 的根节点。

注:一个 TileTree 只有一个根节点,其空间范围是所有子节点的并集。

3.3

瓦片树 tiletree

由唯一的根节点自上而下逐级划分的、多细节层次的树形空间数据结构,树形结构的每个节点表示一个 Tile。

注:父节点 Tile 的空间范围是所有子节点 Tile 的并集。

3.4

瓦片树集合 tiletreeset

由一个或多个 TileTree 构成的集合。

3.5

LOD 分片 patchLOD

表示 TileTree 中指定 LOD 层的一个数据分片。

注:一个 LOD 可包含一个或多个 PatchLOD。

3.6

数据片 patch

PatchLOD 的一个数据分片。

注：一个 PatchLOD 可包含一个或多个 Patch，每个 Patch 有零个或一个父 Patch，有零个或多个子 Patch；父 Patch 的空间范围是各子 Patch 空间范围的并集；Patch 的父子关系构成树状结构。

3.7

数据包 geode

表示一个 Patch 对应的数据包。

注：每个 Patch 可包含零个或多个 Geode。Geode 包含了骨架、材质、纹理三种实体对象。

3.8

实体对象 model entity

Geode 的基本数据构成，包含骨架、材质、纹理三种实体类型。

3.9

骨架 skeleton

几何数据信息，包括顶点、顶点索引、纹理坐标、纹理坐标索引等。

注：骨架附带材质信息。

3.10

材质 material

模型对象表面各可视化属性的集合，包括模型对象表面的色彩、纹理、光滑度、透明度、反射率、折射率、发光度等。

3.11

纹理 texture

纹理贴图信息，包含宽、高、压缩方式及纹理二进制数据等。

4 符号和缩略语

4.1 缩略语

下列缩略语适用于本文件。

EPSG: 欧洲石油测绘组织(The European Petroleum Survey Group)

LOD: 细节层次(Level of Detail)

UML: 统一建模语言(Unified Modelling Language)

WKT: 常用文本表示(Well-known Text Representation)

4.2 UML 图示符号

本标准出现的图用 UML 静态结构表示。

所有数据模型 UML 图示中符号表示的规定见表 1。

表 1 UML 图符号对照表

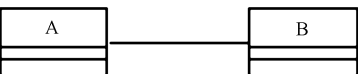
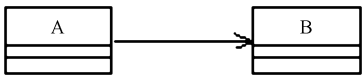
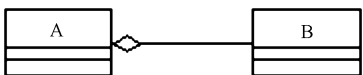
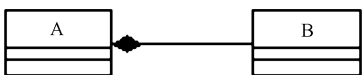
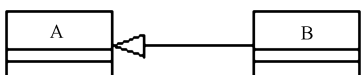
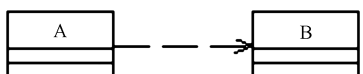
符号	名称	说明
	双向关联	表示 A、B 两个类之间的一般关系，两个类都知道另一个类的公共属性和方法

表 1 (续)

符号	名称	说明
	单向关联	表示 A、B 两个类之间的关联关系, A 类知道 B 类的公共属性和方法, 但 B 类不知道 A 类的公共属性和方法
	聚合	A 对象拥有 B 对象, A 对象可以包含 B 对象, 但 B 对象不是 A 对象的组成部分, 二者生命周期可以不同
	组合	A 对象拥有 B 对象, 是整体和部分的关系, 且生命周期一致
	泛化	B 对象继承 A 对象, 即 B 对象由 A 对象派生
	依赖	A 类依赖于 B 类, B 类的变化将影响 A 类。如果 A 类依赖 B 类, 则 B 可以体现为 A 的局部变量、方法的参数或者静态方法的调用

4.3 UML 多样性描述

本标准涉及的 UML 图中多样性描述含义见表 2。

表 2 UML 图多样性描述

多样性	意义
0..1	0 个或 1 个
1	只能 1 个
0..n	0 个或多个
1..n	1 个或多个

5 基本规定

5.1 基本数据类型

本标准涉及的基本数据类型规定见表 3。

表 3 数值数据类型规定

类型	字节数	取值范围	描述
byte	1	[0,255]	单字节
bool	1	0 1	布尔型
int16	2	[-32 768,32 767]	短整型

表 3 (续)

类型	字节数	取值范围	描述
uint16	2	[0,65535]	无符号短整型
int32	4	[-2147483648,2147483647]	整型
uint32	4	[0,4294967295]	无符号整型
int64	8	$[-2^{63}, (2^{63} - 1)]$	长整型
uint64	8	$[0, (2^{64} - 1)]$	无符号长整型
float	4	$[-3.4 \times 10^{38}, 3.4 \times 10^{38}]$	单精度浮点型
double	8	$[-1.7 \times 10^{308}, 1.7 \times 10^{308}]$	双精度浮点型
wchar	2	—	宽字符类型

5.2 字符串类型

本标准涉及的字符串数据类型用 String 对象描述,采用 Unicode 编码,字符集规定为 UTF8。

```
String{
    int32      length;      //字节数
    byte      str[length]; //数据内容
}
```

5.3 json 格式存储

本标准涉及的 json 格式存储,规定 UTF8 编码,不带 BOM 头。

6 组织结构

6.1 文件组织

本标准规定的数据的组成文件主要包括:描述文件、数据文件、索引树文件、属性文件。

描述文件和数据文件是基础组成部分。描述文件中包含一个或多个 TileTree 的根节点路径;数据文件通过 TileTree 组织,TileTree 中每个 Tile 对应一个 .s3mb 文件;索引树文件是对 TileTree 中各 Tile 的描述,可以在不加载实际数据的情况下,获取 LOD 层的各 Tile 的包围盒、LOD 切换信息、挂接的子节点文件路径等,主要作用是加速 Tile 文件检索的效率;属性文件包括属性描述文件和属性数据文件。各类文件的组织形式见表 4。

表 4 文件组织形式

文件类型	存储形式	存储规定	是否必须
描述文件	.scp 文件	整个数据的描述信息,json 格式。 文件名可自定义,扩展名限定为“.scp”	是
数据文件	.s3mb 文件	Tile 的数据信息。 文件名可自定义,扩展名限定为“.s3mb”	是

表 4 (续)

文件类型	存储形式	存储规定	是否必须
索引树文件	.json 文件	TileTree 各 LOD 层的 Tile 文件信息。 文件名限定为该切片根节点名,扩展名限定为“.json”。 与 TileTree 的根节点同级目录	否
属性描述文件	attribute.json	TileTreeSet 中各数据集属性描述信息。数据集描述见 GB/T 30320—2013 的规定。 文件全名限定为“attribute.json”。 与 .scp 文件同级目录	否
属性数据文件	.s3md 文件	TileTree 中所有对象的属性数据。 文件名限定为 TileTree 根节点名,扩展名限定为 “.s3md”。 与 TileTree 的根节点同级目录	否

6.2 树形结构描述

树形结构相关对象的 UML 图见图 1。

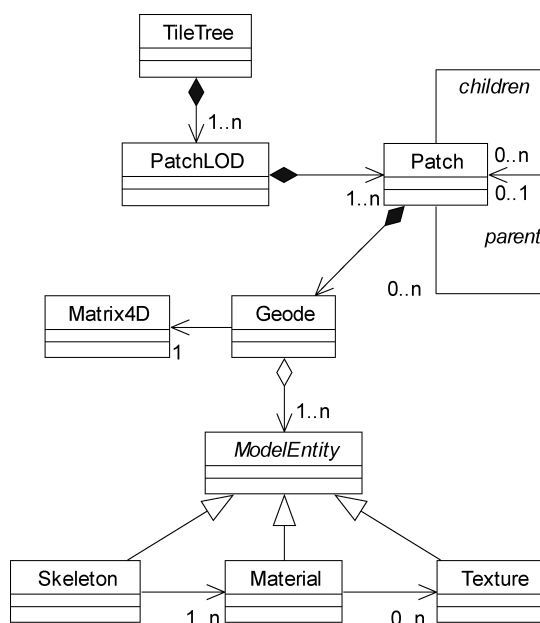


图 1 树形结构 UML 图

7 存储格式

7.1 描述文件

7.1.1 描述文件概述

描述文件(TileTreeSetInfo 对象),用于描述数据的基本信息,关联对象的 UML 图见图 2。

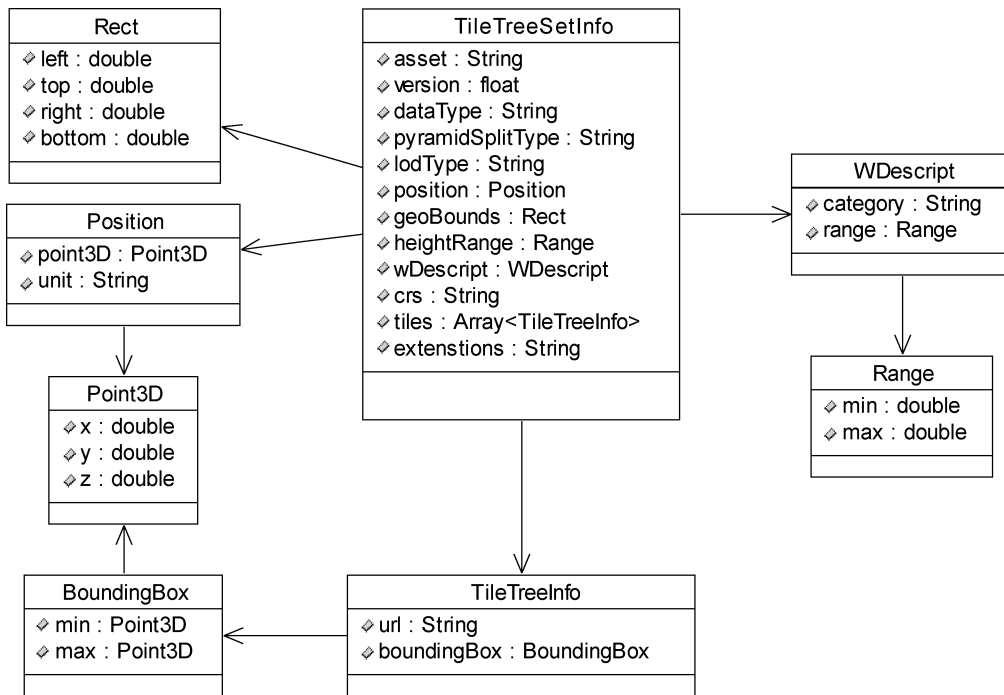


图 2 描述文件的 UML 图

7.1.2 描述文件标签信息

描述文件各标签含义见表 5。

表 5 描述文件各标签含义

标签名	类型	描述
asset	String	数据的基本信息,如生产单位等
version	float	版本号。 取值范围: {‘1.0’}
dataType	String	三维地理空间数据类型。 取值范围: {‘Vector’, ‘ObliquePhotogrammetry’, ‘ArtificialModel’, ‘BIM’, ‘PointCloud’, ‘PipeLine’}。 分别对应: 矢量数据, 倾斜摄影模型, 人工模型, BIM, 点云和管线数据类型
pyramidSplitType	String	数据的空间划分类型。 取值范围: {‘QuadTree’, ‘Octree’, ‘RTree’, ‘K-DTree’}。 分别对应: 四叉树, 八叉树, R 树, K-D 树
lodType	String	LOD 细化类型。 取值范围: {‘Add’, ‘Replace’}。 分别对应: 添加细化, 替换细化
geoBounds	Rect	数据的地理范围, 用 Rect 对象表示, 见表 6
heightRange	Range	数据的高度范围, 用 Range 对象表示, 标识数据高度最大值和最小值, 见表 7
wDescript	WDescript	W 位的描述, 用 WDescript 对象表示, 见表 8
position	Position	整个 TileTreeSet 放置的空间点坐标位置, 用 Position 对象表示, 包含空间坐标位置及其单位, 见表 9

表 5 (续)

标签名	类型	描述
crs	String	坐标系信息。 表示形式: crs: { 'type: content' }, 其中 crs 为关键字, type 可以是 wkt 或 epsg, content 是字符串内容。 epsg 形式的表述格式: crs: { 'epsg: 4326' }。 wkt 形式的表述格式: crs: { 'wkt: wktcontent' }。 wktcontent 应符合 GB/T 33187.1—2016 的规定
tiles	Array<TileTreeInfo>	瓦片数据信息, 用 TileTreeInfo 对象表示, 包括各 TileTree 的索引文件 URL 和包围球, 见表 11
extensions	String	用户扩展信息

表 6 Rect 对象各标签含义

标签名	类型	描述
left	double	数据地理范围的左值
top	double	数据地理范围的上值
right	double	数据地理范围的右值
bottom	double	数据地理范围的下值

表 7 Range 对象各标签含义

标签名	类型	描述
min	double	最小值
max	double	最大值

表 8 WDescript 对象各标签含义

标签名	类型	描述
category	String	W 位含义描述信息
range	Range	W 位数值范围, 用 Range 对象表示, 包含 W 位最小值和最大值, 见表 7

表 9 Position 对象各标签含义

标签名	类型	描述
point3D	Point3D	空间点坐标值, 用 Point3D 对象表示, 包含空间点的 x、y、z 坐标值, 见表 10。 经度和纬度的表示法应符合 GB/T 16831—2013 的规定
unit	String	空间坐标系的单位。 取值范围: { 'Degree', 'Meter' }。 分别对应: 度、米

表 10 Point3D 对象各标签含义

标签名	类型	描述
x	double	空间点的 x 坐标值
y	double	空间点的 y 坐标值
z	double	空间点的 z 坐标值

表 11 TileTreeInfo 对象各标签含义

标签名	类型	描述
url	String	瓦片所在路径
boundingBox	BoundingBox	瓦片数据范围,用 BoundingBox 对象表示,见表 12

表 12 BoundingBox 对象各标签含义

标签名	类型	描述
max	Point3D	数据包围盒最大角点,用 Point3D 对象表示,见表 10
min	Point3D	数据包围盒最小角点,用 Point3D 对象表示,见表 10

7.2 数据文件

7.2.1 s3mb 文件逻辑结构

7.2.1.1 主要结构

数据文件是数据的主要组成部分,由.s3mb(Spatial 3D Model Binary)文件组成;一个 PatchLOD 的数据对应存储为一个.s3mb 文件。对象 UML 图见图 3。s3mb 文件包含的各对象含义见表 13。

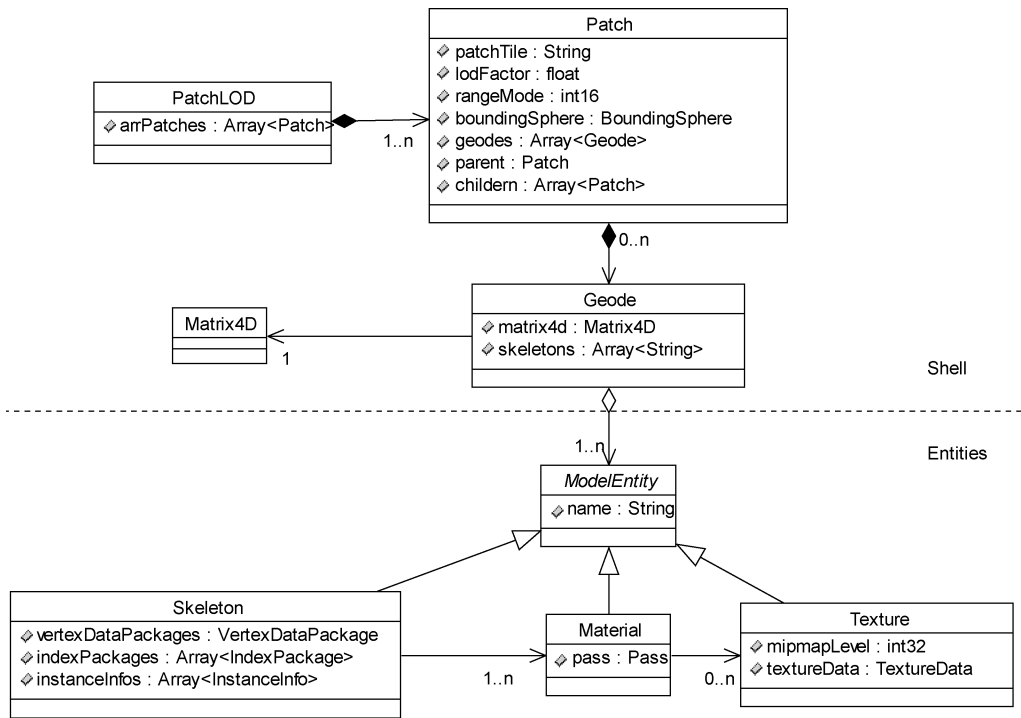


图 3 s3mb 文件存储对象 UML 图

表 13 s3mb 文件各对象的含义

对象名	类型	描述
arrPatches	Array<Patch>	该 PatchLOD 中所有 Patch 的集合。Patch 描述见表 14

表 14 Patch 对象各属性含义

属性名	类型	描述
patchTile	String	Patch 所在的 s3mb 文件名
lodFactor	float	切换因子,即 LOD 切换的阈值,与切换模式配合使用
rangeMode	int16	切换模式,见 7.2.2.2
boundingSphere	BoundingSphere	包围球,用 BoundingSphere 对象表示,见表 15
geodes	Array<Geode>	数据包集合。Geode 见表 16
parent	Patch	父节点
children	Array<Patch>	子节点数组,即挂接的所有子 Patch

表 15 BoundingSphere 对象各属性含义

属性名	类型	描述
x	double	中心点的 x 坐标值
y	double	中心点的 y 坐标值
z	double	中心点的 z 坐标值
r	double	包围球半径

表 16 Geode 对象各属性含义

属性名	类型	描述
matrix4d	Matrix4D	作用于骨架姿态的矩阵,用 Matrix4D 对象表示,见表 17
skeletons	Array<String>	骨架名字数组,见表 18

表 17 Matrix4D 对象各属性含义

属性名	类型	描述
values	double[16]	4×4 纹理矩阵,16 个 double 值表示,行主序

表 18 ModelEntity 对象各属性含义

属性名	类型	描述
name	String	实体名字,是实体对象在 TileTree 中的唯一标识,Skeleton、Material、Texture 对象继承自 ModelEntity 对象,见表 19、表 26 和表 32

7.2.1.2 骨架对象

骨架(Skeleton)对象由一个顶点数据包(VertexDataPackage)和一个或多个顶点索引包(IndexPackage)组成。顶点数据包是对各顶点的描述,包括坐标、法线、颜色、纹理坐标、模型对象 ID、实例化信息等;顶点索引包是对骨架结构构造的描述,每个顶点索引包有一个或多个 Pass,用来标识该组顶点的渲染方式。骨架相关对象 UML 图见图 4。骨架对象各属性含义见表 19。

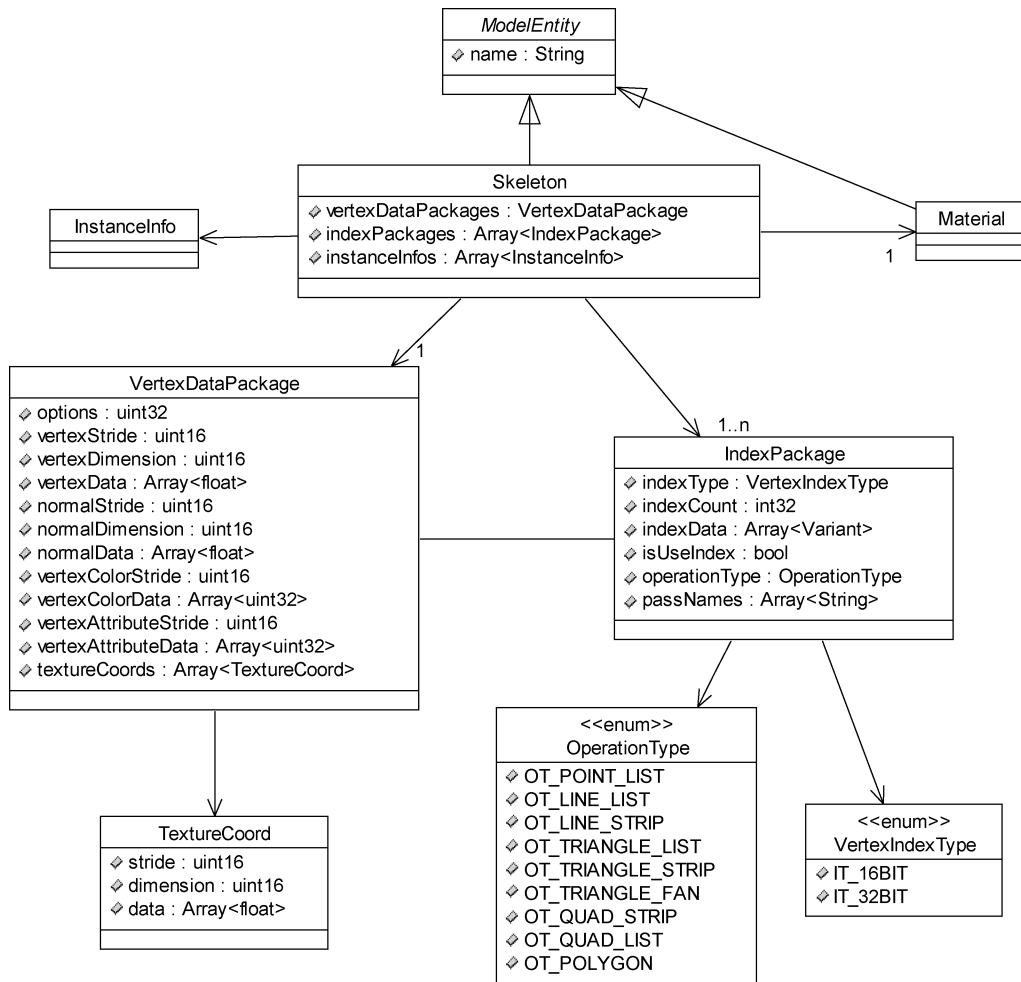


图 4 骨架对象 UML 图

表 19 Skeleton 对象各属性含义

属性名	类型	描述
vertexDataPackages	VertexDataPackage	顶点数据包,用 VertexDataPackage 对象表示,见表 20
indexPackages	Array<IndexPackage>	顶点索引数据包,用 IndexPackage 对象表示,见表 23
instanceInfos	Array<InstanceInfo>	实例化信息数组,用实例化信息对象 InstanceInfo 表示,可支持点外挂模式,见表 22

表 20 VertexDataPackage 对象各属性含义

属性名	类型	描述
options	uint32	数据选项信息,存储扩展数据
vertexStride	uint16	顶点坐标在数组中的偏移量
vertexDimension	uint16	顶点维度
vertexData	Array<float>	顶点坐标值数组

表 20 (续)

属性名	类型	描述
normalStride	uint16	法向量在数组中的偏移量
normalDimension	uint16	法向量的维度
normalData	Array<float>	法向量分量值构成的数组
vertexColorStride	uint16	顶点颜色在数组中的偏移量
vertexColorData	Array<uint32>	顶点颜色数组
vertexAttributeStride	uint16	顶点属性在数组中的偏移量
vertexAttributeData	Array<uint32>	顶点属性数组,用于存储模型对象的 ID 信息
textureCoords	Array<TextureCoord>	纹理坐标数组,用纹理坐标对象 TextureCoord 表示,见表 21

表 21 TextureCoord 对象各属性含义

属性名	类型	描述
stride	uint16	偏移量
dimension	uint16	纹理坐标维度
data	Array<float>	纹理坐标值数组

表 22 InstanceInfo 对象各属性含义

属性名	类型	描述
matrixValues	double[16]	4×4 矩阵值
objectID	uint32	对象 ID

表 23 IndexPackage 对象各属性含义

属性名	类型	描述
indexType	VertexIndexType	顶点索引类型,用 VertexIndexType 对象表示,枚举见表 24
indexCount	int32	顶点索引个数
indexData	Array<Variant>	顶点数据,根据顶点索引类型,可能是 Short 数组或 Integer 数组
isUseIndex	bool	是否使用索引
operationType	OperationType	索引的组织方式,用 OperationType 对象表示,枚举见表 25
passNames	Array<String>	渲染该对象时使用的 Pass 对象名称

表 24 VertexIndexType 对象各枚举值含义

枚举名	类型	描述
IT_16BIT	uint16	16 位无符号整型
IT_32BIT	uint32	32 位无符号整型

一般根据顶点的个数选择,顶点的个数大于 65 535 则采用 IT_32BIT,否则采用 IT_16BIT。

表 25 OperationType 对象各枚举值含义

枚举名	类型	描述
OT_POINT_LIST	int	单个点
OT_LINE_LIST	int	两点线
OT_LINE_STRIP	int	线串
OT_TRIANGLE_LIST	int	三角形
OT_TRIANGLE_STRIP	int	条带三角形
OT_TRIANGLE_FAN	int	扇面三角形构成
OT_QUAD_STRIP	int	条带四边形
OT_QUAD_LIST	int	四边形串,不共享边
OT_POLYGON	int	多边形

7.2.1.3 材质对象

材质(Material)对象由 Pass 构成,Pass 中记录了材质采用的纹理对象名称,采用 json 格式表述。材质相关对象的 UML 图见图 5,各属性含义见表 26。

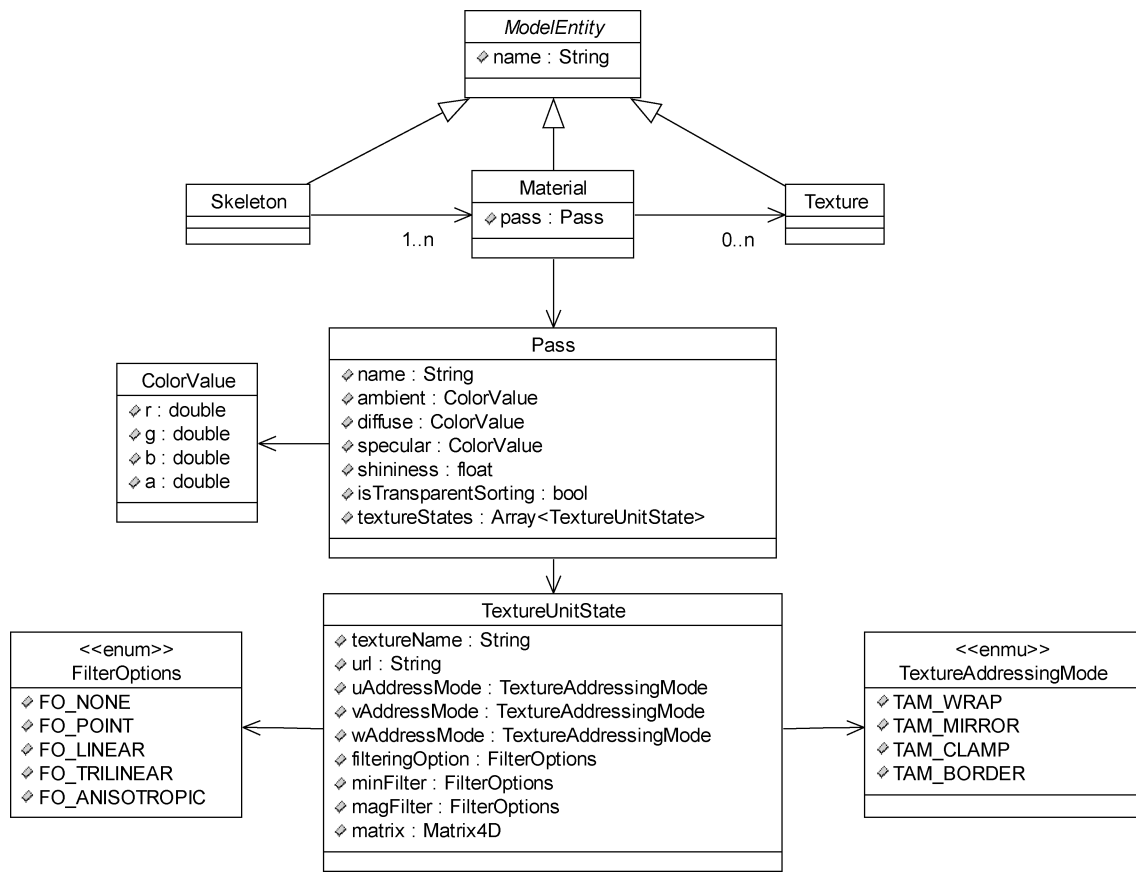


图 5 材质对象 UML 图

表 26 Material 对象各属性含义

属性名	类型	描述
pass	Pass	渲染通道,用 Pass 对象表示,见表 27

表 27 Pass 对象各属性含义

属性名	类型	描述
name	String	Pass 的名称
ambient	ColorValue	环境光颜色,用颜色值 ColorValue 对象表示。包含环境光颜色的 r、g、b、a 分量值等元素,见表 28
diffuse	ColorValue	散射光颜色,用颜色值 ColorValue 对象表示。包含散射光颜色的 r、g、b、a 分量值等元素,见表 28
specular	ColorValue	镜面光颜色,用颜色值 ColorValue 对象表示。包含镜面光颜色的 r、g、b、a 分量值等元素,见表 28
shininess	float	光泽度,影响反射光的高光部分的强度

表 27 (续)

属性名	类型	描述
isTransparentSorting	bool	是否使用透明排序。 取值范围: {'True', 'False'}。 分别对应: 是, 否
textureStates	Array<Texture UnitState>	纹理信息, 用纹理信息 TextureUnitState 对象表示。包含 textureName、url、uAddressMode、vAddressMode、wAddressMode、fil- teringOption、minFilter、magFilter、matrix 等元素, 见表 29

表 28 ColorValue 对象各属性含义

属性名	类型	描述
r	double	红色分量值, 取值范围 0.0 到 1.0
g	double	绿色分量值, 取值范围 0.0 到 1.0
b	double	蓝色分量值, 取值范围 0.0 到 1.0
a	double	透明度分量值, 取值范围 0.0 到 1.0

表 29 TextureUnitState 对象各属性含义

属性名	类型	描述
textureName	String	纹理名字
url	String	纹理引用路径
uAddressMode	TextureAddressingMode	纹理坐标 u 方向上纹理坐标寻址模式。 用纹理处理模式 TextureAddressingMode 对象表示, 见表 30
vAddressMode	TextureAddressingMode	纹理坐标 v 方向上纹理坐标寻址模式。 用纹理处理模式 TextureAddressingMode 对象表示, 见表 30
wAddressMode	TextureAddressingMode	纹理坐标 w 方向上纹理坐标寻址模式。 用纹理处理模式 TextureAddressingMode 对象表示, 见表 30
filteringOption	FilterOptions	纹理插值模式, 用 FilterOptions 对象表示, 见表 31
minFilter	FilterOptions	纹理缩小时采用的插值模式, 用 FilterOptions 对象表示, 见表 31
magFilter	FilterOptions	纹理放大时采用的插值模式, 用 FilterOptions 对象表示, 见表 31
matrix	Matrix4D	纹理矩阵, 见表 17

表 30 TextureAddressingMode 对象各枚举值含义

枚举名	类型	描述
TAM_WRAP	int	重复贴图
TAM_MIRROR	int	对称翻转
TAM_CLAMP	int	边缘像素来填充所有大于 1 的纹理坐标, 边缘拉长
TAM_BORDER	int	边框像素来填充所有大于 1 的纹理坐标, 边框拉长

表 31 FilterOptions 对象各枚举值含义

枚举名	类型	描述
FO_NONE	int	无过滤
FO_POINT	int	邻近取样
FO_LINEAR	int	双线过滤
FO_TRILINEAR	int	三线过滤
FO_ANISOTROPIC	int	各向异性过滤

7.2.1.4 纹理对象

纹理(Texture)对象 UML 图见图 6。纹理(Texture)对象各标签含义见表 32。

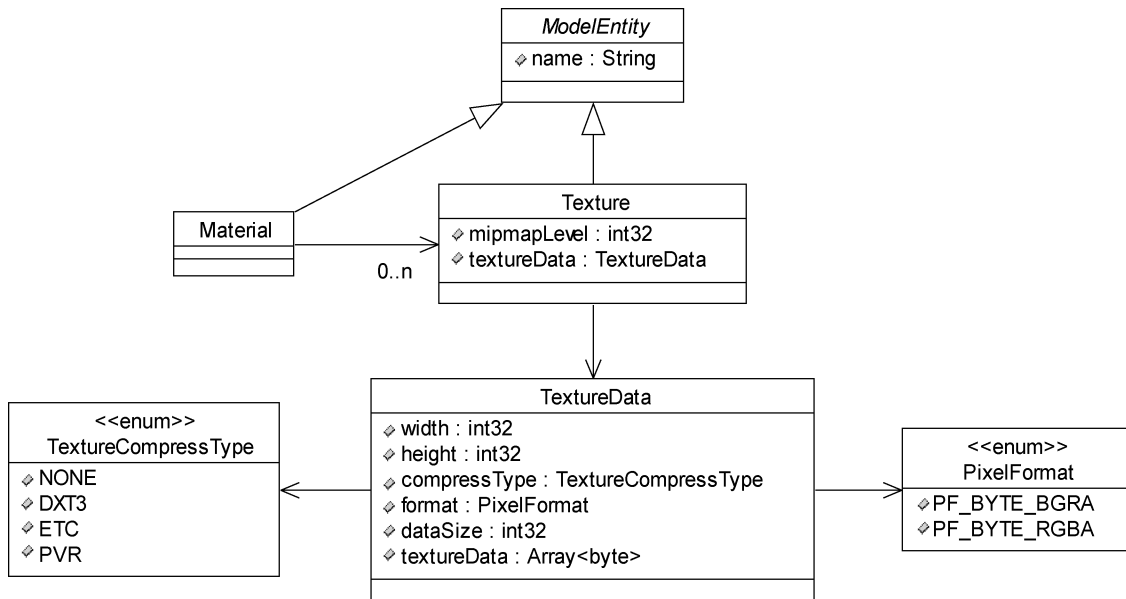


图 6 纹理对象 UML 图

表 32 Texture 对象各属性含义

属性名	类型	描述
mipmapLevel	int32	纹理对象包含的 mipmap 层数
textureData	TextureData	纹理数据,用 TextureData 对象表示,见表 33

表 33 TextureData 对象各属性含义

属性名	类型	描述
width	int32	横向像素个数
height	int32	纵向像素个数

表 33 (续)

属性名	类型	描述
compressType	TextureCompressType	纹理压缩方式,用枚举类型 TextureCompressType 表示,见表 34
format	PixelFormat	纹理的像素格式,用枚举类型 PixelFormat 表示,见表 35
dataSize	int32	纹理的二进制流大小
textureData	Array<byte>	纹理数据二进制流

表 34 TextureCompressType 各枚举值含义

枚举名	类型	描述
NONE	int	无压缩格式
DXT3	int	DXT3 纹理压缩格式,适用于 PC 端
ETC	int	ETC 纹理压缩格式,适用于 Andriod 端
PVR	int	PVR 纹理压缩格式,适用于 iOS 端

表 35 PixelFormat 各枚举值含义

枚举名	类型	描述
PF_BYTE_BGRA	int	BGRA 格式
PF_BYTE_RGBA	int	RGBA 格式

7.2.2 s3mb 文件的二进制流描述

7.2.2.1 s3mb 文件的主要组成部分

s3mb 文件采用二进制流形式存储,字节序规定为 Little-Endian,即低位字节排放在内存的低地址端。s3mb 文件流内容如下:

```
S3MBFile{
    float header;           //s3mb 文件版本号
    uint32 zippedSize;     //zipped package 的字节数
    byte * zippedPackage;  //数据压缩包,长度为 zippedSize
};
```

zippedPackage 解压缩后,包含 Reserved、Shell 和 ModelEntities 三个部分,见图 7。Reserved 是预留的四个字节;Shell 存储 PatchLOD、Patch、Geode 对象;ModelEntities 即实体数据,包括骨架(Skeleton)、材质(Material)、纹理(Texture)。

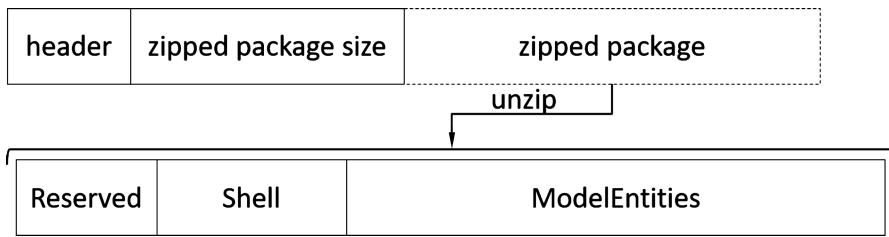


图 7 s3mb 文件二进制数据包

7.2.2.2 Shell 的二进制流描述

Shell 及相关对象的二进制流结构符合如下规定：

```

Shell{
    uint32 streamSize;           //shell 的二进制流字节数
    PatchLOD patchLod;
};
PatchLOD {
    int32 patchCount;           //Patch 对象的个数
    Patch patches[patchCount];
};
Patch{
    float lodFactor;            //LOD 切换因子
    RangeMode rMode;           //LOD 切换模式,存储为 int16
    BoundingSphere boundingSphere; //包围球
    String strChildTile;        //挂接的子文件的相对路径
    int32 geodeCount;           //包含的 Geode 个数
    Geode geodes[geodeCount];
};
RangeMode{
    Distance_From_EyePoint,     //根据到相机的距离切换
    Pixel_Size_OnScreen         //根据投影到屏幕的像素大小切换
};
BoundingSphere{
    double x;                   //中心点 x 坐标
    double y;                   //中心点 y 坐标
    double z;                   //中心点 z 坐标
    double r;                   //包围球半径
};
Geode{
    Matrix4D matrix;
    int32 skeletonCount;
    String skeletonNames[skeletonCount]; //骨架名
};
    
```



```
Matrix4D{ //4×4 矩阵,行主序
    double values[16];
};
```

Shell 的 Geode 中仅存储骨架名,对应的实体对象存储在 ModelEntities 中。

7.2.2.3 ModelEntities 的二进制流描述

ModelEntities 及相关对象的二进制流结构符合如下规定:

```
ModelEntities{
    uint32 skeletonStreamSize; //骨架数据流的二进制长度,以 byte 为单位
    int32 skeletonCount;
    Skeleton skeletons[skeletonCount];
    uint32 textureStreamSize;
    int32 textureCount; //纹理数据流的二进制长度,以 byte 为单位
    Texture textures[textureCount];
    String materials; //材质字符串(json 格式)
};
Skeleton{
    String name;
    VertexDataPackage dataPack;
    int32 indexpackCount;
    IndexPacakge indexPacks[indexpackCount];
};
VertexDataPackage{
    byte reserved[4]; //预留
    uint32 vertexCount; //顶点
    uint16 vertexDimension;
    uint16 vertexStride;
    float vertexData[vertexCount * vertexDimension];
    uint32 normalCount; //法线
    uint16 normalDimension;
    uint16 normalStride;
    float normalData[normalCount * normalDimension];
    int32 vertexcolorCount; //顶点颜色
    uint16 vertexColorStride;
    byte reserved[2];
    uint32 vertexColorData[vertexcolorCount]; //颜色采用 uint32 存储,byte[0]~byte[4]分别表示 R、G、B、A 的值
    int32 vertexAttributeCount; //顶点属性
    uint16 vertexAttributeStride;
    byte reserved[2];
    uint32 vertexAttributeData[vertexAttributeCount];
    uint16 texturecoordCount; //纹理坐标
    byte reserved[2];
```

```

TextureCoord textureCoords[textureCoordCount];
uint16 instanceInfoCount;           //实例化信息
InstanceInfo instanceInfo[instanceInfoCount];
};
TextureCoord{
uint32 coodsCount;
uint16 dimension;
uint16 stride;
float data[coodsCount * dimension];
};
InstanceInfo{                       //实例化信息
double matrixvalues[16];           //矩阵,行主序
uint32 objectID;                   //对象 ID
};
IndexPacakge{
uint32 indexCount;
VertexIndexType enIndexType;       //存储为 byte
byte reserved;
OperationType opType;              //存储为 byte
byte reserved;
variant indexData[indexCount]; //索引值,说明:若 VertexIndexType 为 IT_16BIT,则 variant
类型为 uint16;若 VertexIndexType 为 IT_32BIT,则 variant 类型为 uint32)
int32 passCount;
String passNames[passCount];
};
VertexIndexType{
IT_16BIT = 0,                       //索引值用 uint16 表示
IT_32BIT = 1                         //索引值用 uint32 表示
};
OperationType{
OT_POINT_LIST = 1,                   //单个点
OT_LINE_LIST = 2,                     //两点线
OT_LINE_STRIP = 3,                   //线串
OT_TRIANGLE_LIST = 4,                //三角形
OT_TRIANGLE_STRIP = 5,               //条带三角形
OT_TRIANGLE_FAN = 6,                 //扇面三角形构成
OT_QUAD_STRIP = 8,                   //条带四边形
OT_QUAD_LIST = 9,                    //四边形串,不共享边
OT_POLYGON = 10,                     //多边形
};
Texture{
String strName;
int32 mipMapLevel;

```

```

TextureData texData;
};
TextureData{
    int32 width;
    int32 height;
    TextureCompressType compressType; //存储为 uint32
    int32 datasize;
    PixelFormat pixelFormat; //存储为 uint32
    byte data[datasize];
};
TextureCompressType{
    TC_NONE = 0,
    TC_DXT3 = 14,
};
PixelFormat{
    PF_BYTE_BGRA = 12,
    PF_BYTE_RGBA = 13,
};

```

以上是 ModelEntities 的二进制流规定。

7.3 索引树文件

索引树文件采用 json 文件存储,扩展名为.json。各标签含义见表 36。

表 36 索引树文件各标签含义

标签名	类型	描述
name	String	Tile 的名称
tileInfo	TileInfo	Tile 的信息,见表 37
status	Status	瓦片数据的状态,用 Status 对象表示,包含瓦片 LOD 层级总数元素 lodCount 以及瓦片总数元素 tilesCount,见表 38

表 37 TileInfo 对象各标签含义

标签名	类型	描述
lodNum	int	根结点所在 LOD 层号,自顶向下递增,起始层号为 0
modelPath	String	数据文件的路径,相对于索引文件本身
rangeMode	String	距离切换模式。 取值范围: { 'distanceFromEyePoint', 'pixelSizeOnScreen' }。 分别表示:切片与视点的距离,切片投影在屏幕上的像素数
rangeValue	double	子节点切换阈值
boundingBox	BoundingBox	数据的包围盒,用 BoundingBox 对象表示,见表 12
children	Array<TileInfo>	各子节点信息

表 38 Status 对象各标签含义

标签名	类型	描述
lodCount	int	瓦片 LOD 层级总数
tilesCount	int	瓦片总数

7.4 属性文件

7.4.1 构成

属性文件包括属性描述文件和属性数据文件。属性描述文件名规定为 attribute.json,与描述文件(.scp)处于同级目录;属性数据文件名与 TileTree 的根节点文件名相同,扩展名为.s3md(Spatial 3D Model Description);一个根节点对应一个属性数据.s3md 文件,与数据文件(.s3mb)处于同级目录。

7.4.2 描述文件

属性描述文件描述各图层对象的 ID 范围及字段信息,采用 json 格式。各标签含义见表 39。

表 39 属性描述文件各标签含义

标签名	类型	描述
layerInfos	Array<LayerInfo>	各图层的属性描述信息,用 LayerInfos 对象表示,包含单个图层属性描述元素 LayerInfo,见表 40

表 40 LayerInfo 对象各标签含义

标签名	类型	描述
layerName	String	图层名
iDRange	IDRange	ID 的范围,用对象 IDRange 表示,包含某一图层对象的 ID 最小值元素 min 和某一图层对象的 ID 最大值元素 max,见表 41
fieldInfos	Array<FieldInfo>	字段信息集合,用 FieldInfo 数组表示,见表 42

表 41 IDRange 对象各标签含义

标签名	类型	描述
min	int32	该图层对象的 ID 最小值
max	int32	该图层对象的 ID 最大值

表 42 FieldInfo 对象各标签含义

标签名	类型	描述
name	String	字段名
alias	String	字段别名

表 42 (续)

标签名	类型	描述
type	String	字段类型。 取值范围: {‘bool’, ‘int16’, ‘uint16’, ‘int32’, ‘uint32’, ‘int64’, ‘uint64’, ‘float’, ‘double’, ‘wchar’, ‘text’, ‘date’, ‘time’, ‘timestamp’}。 其中, text 为 String 类型; date 为日期型; time 为时间型; timestamp 为时间日期型。日期和时间的表示法应符合 GB/T 7408—2005 的规定。 其他类型含义及取值范围见 5.1 及 5.2
size	int32	字段长度
isRequired	bool	是否是必填字段。 取值范围: {‘True’, ‘False’}。 分别对应: 是, 否

7.4.3 数据文件

属性数据文件包含了各图层的属性描述信息和每个对象的各属性值, 采用 json 文件存储, 并采用 zip 压缩, 见图 8。

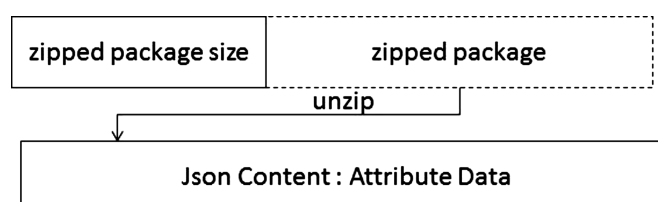


图 8 属性数据文件二进制流结构

二进制流结构如下:

```

AttributeData{
  uint32 nZippedSize;          //zipped package 的字节数
  byte * zippedPackage;       //压缩后的属性数据
}
  
```

数据解压后, 为 json 字符串。属性数据中的各标签含义见表 43。

表 43 属性数据文件各标签含义

标签名	类型	描述
layer	Array<LayerInfo>	图层信息标签, 用 LayerInfo 对象表示, 包含 ID 范围元素 IDRange、字段信息集合元素 FieldInfos、各对象属性值集合元素 Records, 见表 44

表 44 LayerInfo 对象各标签含义

标签名	类型	描述
idRange	IDRange	ID 的范围,表示对应瓦片范围内对象的最大和最小 ID,用 IDRange 对象表示,见表 41
fieldInfos	Array<FieldInfo>	字段信息集合,用 FieldInfo 数组表示,见表 42
records	Array<Record>	各对象的属性值记录信息集合,用 Record 数组表示,见表 45

表 45 Record 对象各标签含义

标签名	类型	描述
id	int32	对象的 ID
values	Array<Value>	各字段的属性值详细描述,用 Value 数组表示,见表 46

表 46 Value 对象各标签含义

标签名	类型	描述
name	String	字段名
value	变体	字段值

附 录 A
(资料性附录)
数据示例

A.1 描述文件示例

以“鸟巢.scp”为例,包含了两个 TileTree,名称分别为 Tile_-7281_21185_0000 和 Tile_-7282_21183_0000,具体内容如下:

```
{
  "asset": "SuperMap",
  "version": 1.0,
  "dataType": "BIM",
  "pyramidSplitType": "QuadTree",           //四叉树剖分
  "lodType": "Replace",                    //LOD 为替换模式
  "position":                               //插入点,单位为度
  {
    "x": 116.36,
    "y": 39.99,
    "z": 0.0,
    "units": "Degree"
  },
  "geoBounds":                             //地理空间范围
  {
    "left": 116.3635,
    "top": 40.0018,
    "right": 116.3755,
    "bottom": 39.9932
  },
  "heightRange":                           //高度范围
  {
    "min": 9.4875,
    "max": 119.9612
  },
  "wDescript":                             //w 位描述信息
  {
    "category": "",
    "range":
    {
      "min": 0.0,
      "max": 0.0
    }
  }
}
```

```

    },
    "tiles": //各 TileTree 对应的根节点文件路径及包围盒(局部坐标系)
    [
    { //第一个根节点的信息
      "url": "./Tile_-7281_21185_0000/Tile_-7281_21185_0000.s3mb",
      "boundingbox":
      {
        "min":
        {
          "x":245.36567664297159,
          "y":-534.7293082718718,
          "z":-34.66962171293413
        },
        "max":
        {
          "x":443.1873785885407,
          "y":-336.9076063263026,
          "z":163.152080232635
        }
      }
    },
    { //第二个根节点的信息
      "url": "./Tile_-7282_21183_0000/Tile_-7282_21183_0000.s3mb",
      "boundingbox":
      {
        "min":
        {
          "x":-604.2845700298257,
          "y":92.21901333930407,
          "z":-190.14669717375353
        },
        "max":
        {
          "x":-147.10063304583208,
          "y":549.4029503232977,
          "z":267.03723981024009
        }
      }
    }
  ]
}

```


A.2 索引树文件示例

对其中一个名称为 Tile_-7281_21185_0000 的 TileTree,其文件夹下有索引树文件 Tile_-7281_21185_0000.json,具体内容如下:

```
{
  "lodTreeExport":
  {
    "name": "Tile_-7281_21185_0000",
    "tileInfo":
    {
      "lodNum": 0, //LOD 层号,0 层为根节点
      "modelPath": "Tile_-7281_21185_0000.s3mb", //根节点对应的文件路径(相对于索引文件)
      "rangeMode": "pixelSizeOnScreen", //LOD 的距离切换模式
      "rangeValue": 1.0, //子节点切换阈值
      "boundingBox": //根节点包围盒(局部坐标系)
      {
        "min":
        {
          "x": -68.02222442626953,
          "y": -43.73067092895508,
          "z": 9.495752334594727
        },
        "max":
        {
          "x": 68.02222442626953,
          "y": 43.73127746582031,
          "z": 119.96125030517578
        }
      },
    },
    "children": //子节点信息
    [
      {
        "tileInfo":
        {
          "lodNum": 1,
          "modelPath": "Tile_-7281_21185_0000_0004_0000.s3mb",
          "rangeMode": "pixelSizeOnScreen",
          "rangeValue": 2.0,
          "boundingBox":
          {
            "min":
            {
```

```

        "x": -68.02222442626953,
        "y": -43.73067092895508,
        "z": 9.495752334594727
    },
    "max":
    {
        "x": 68.02222442626953,
        "y": 43.73127746582031,
        "z": 119.96125030517578
    }
},
    "children": [...] //该节点的子节点信息(递归)
}
],
"status": //TileTree 的总体描述信息
{
    "lodCount": 5,
    "tilesCount": 6
}
}
}
}
}

```

A.3 材质内容示例

材质内容包含在.s3mb文件中,见7.2.1.3和7.2.2.3中的内容,示例数据如下:

```

{
    "materials": //材质对象描述信息集合
    [
        {
            "material": //第一个材质对象
            {
                "id": "0_10710_Sec_0005_-7281_21185_0000_0000_0", //材质对象的 ID
                "ambient": {"r": 1.0, "g": 1.0, "b": 1.0, "a": 1.0},
                "diffuse": {"r": 1.0, "g": 1.0, "b": 1.0, "a": 1.0},
                "specular": {"r": 1.0, "g": 1.0, "b": 1.0, "a": 1.0},
                "shininess": 0.0,
                "transparentsoring": false,
                "textureunitstates": //包含的纹理数据信息
                [
                    {
                        "textureunitstate":

```

```

    {
      "id": "0_10710_Sec_0005_-7281_21185_0000_0000", //纹理对象的 ID
      "url": "", //URL 为空,则纹理数据通过 ID 与 s3mb 文件中存储的纹理数据关联
      "addressmode": {"u":0,"v":0,"w":0},
      "filteringoption": -842150451,
      "filtermin": 2,
      "filtermag": 2,
      "texmodmatrix": [1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0]
    }
  ]
}
],
}

```

A.4 属性描述文件示例

属性相关数据为可选。如果存在属性数据,则属性描述文件的名称限定为 attribute.json,且与“鸟巢.scp”文件在同级目录,见表 39。示例数据如下:

```

{
  "layerInfos":
  [
    {
      "layerName": "Building_Sub", //对应的原始数据集名称
      "idRange": {"minID":1,"maxID":10}, //TileTreeSet 中包含的数据集对象的 ID 范围
      "fieldInfos": //字段描述信息
      [
        {
          "name": "SmID",
          "alias": "SmID",
          "type": "int32",
          "size": 4,
          "isRequired": true
        },
        {
          "name": "MODELNAME",
          "alias": "ModelName",
          "type": "String",
          "size": 30,
          "isRequired": false
        }
      ]
    }
  ]
}

```

```

    }
  ]
}

```

A.5 属性数据文件示例

属性相关数据为可选。如果存在属性数据,则除了属性描述文件描述图层的相关属性信息,在每个 TileTree 文件夹中,存在与根节点同名,扩展名为.s3md 的文件,用于存储该 TileTree 下所有属性数据,见表 43。以 Tile_-7281_21185_0000 为例,属性数据为文件夹下 Tile_-7281_21185_0000.s3md 文件,具体内容如下:

```

{
  "layerInfos":           //属性数据集合
  [
    {
      "idRange":         //该 TileTree 中包含的对象 ID 范围
      {
        "minID":1,
        "maxID":1
      },
      "fieldInfos":      //各个字段描述的信息(SmID、MODELNAME)
      [
        {
          "name":"SmID",
          "alias":"SmID",
          "type":"int32",
          "size":4,
          "isRequired":true
        },
        {
          "name":"MODELNAME",
          "alias":"ModelName",
          "type":"String",
          "size":30,
          "isRequired":false
        },
      ]
    }
  ]
  "records":             //各个字段值集合
  [
    {
      "id":1,
      "values":          //ID=1 的对象各个字段的值(SmID、MODELNAME)
      [
        {

```

```
"name": "SmID",  
  "value": 1  
},  
{  
  "name": " ModelName",  
  "value ": "鸟巢"  
},  
]  
}  
]  
}  
]  
}
```

参 考 文 献

- [1] GB/T 23707—2009 地理信息 空间模式
 - [2] GB/T 30170—2013 地理信息 基于坐标的空间参照
 - [3] ISO 19101 Geographic information—Reference model
-

