

团体标准

T/CAGIS ×-××××

三维地理空间数据格式

3D Geospatial Data Format Specification

(征求意见稿)

在提交反馈意见时，
请将您知道的相关专利连同支持性文件一并附上。

××××-××-××发布

××××-××-××实施

中国地理信息产业协会 发布

目 次

前 言.....	IV
引 言.....	V
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 符号.....	2
4.2 缩略语.....	2
4.3 UML 图示符号.....	3
4.4 UML 多样性描述.....	3
4.5 数据流描述.....	3
5 数据组织结构.....	4
5.1 数据的逻辑组织结构.....	4
5.2 数据的文件组织.....	4
5.3 树形结构描述.....	5
6 数据的物理存储.....	6
6.1 描述文件.....	6
6.2 索引树文件.....	8
6.3 数据文件.....	9
6.4 属性文件.....	19
附录 A （参考性附录） 数据示例.....	21
A.1 描述文件示例.....	21
A.2 索引树文件示例.....	22
A.3 材质内容示例.....	24
A.4 属性描述文件示例.....	24
A.5 属性数据文件示例.....	25
图 1 瓦片数据组织 UML 图.....	4
图 2 TileTree 的树形结构.....	5
图 3 树形结构 UML 图.....	6
图 4 描述文件的 UML 图.....	6
图 5 S3MB 文件存储对象 UML 图.....	9
图 6 骨架对象 UML 图.....	11
图 7 材质对象 UML 图.....	13
图 8 纹理对象 UML 图.....	15
图 9 S3MB 文件二进制数据包.....	16
表 1 符号对照表.....	2
表 2 缩写对照表.....	2

表 3	UML 图符号对照表	3
表 4	UML 图多样性描述	3
表 5	数值数据类型描述	3
表 6	对象存储的文件组织形式	4
表 7	描述文件各标签含义	7
表 8	Rect 对象各标签含义	7
表 9	Range 对象各标签含义	7
表 10	WDescript 对象各标签含义	7
表 11	Position 对象各标签含义	8
表 12	Point3D 对象各标签含义	8
表 13	TileTreeInfo 对象各标签含义	8
表 14	BoundingBox 对象各标签含义	8
表 15	索引树文件各标签含义	8
表 16	TileInfo 对象各标签含义	8
表 17	Status 对象各标签含义	9
表 18	材质对象各标签含义	13
表 19	ColorValue 对象各标签含义	13
表 20	TextureUnitState 对象	14
表 21	TexureAddressingMode 对象各标签含义	14
表 22	FilterOptions 对象各标签含义	14
表 23	属性描述文件各标签含义	19
表 24	LayerInfo 对象各标签含义	19
表 25	IDRange 对象各标签含义	20
表 26	FieldInfo 对象各标签含义	20
表 27	属性数据文件各标签含义	20
表 28	LayerInfo 对象各标签含义	20
表 29	Record 对象各标签含义	20
表 30	Value 对象各标签含义	21

前 言

本标准按照GB/T 1.1—2009给出的规则起草。

本标准由北京超图软件股份有限公司提出。

本标准由中国地理信息产业协会团体标准化管理委员会归口。

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，中国建筑标准设计研究院有限公司，中国建筑科学研究院有限公司，中国城市规划设计研究院，中建工程研究院有限公司，中设数字技术股份有限公司，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。。

本标准起草人：XXXXXXXXXXXX。

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

引 言

近年来，倾斜摄影、激光点云等数据采集技术的发展，有效降低了三维地理空间数据的获取成本和时间周期，提高了三维地理空间数据的精度。三维地理空间数据获取方式的变革，使大量三维地理空间数据的获取成为可能。伴随着大规模的三维地理空间数据不断积累，三维地理空间数据的高效发布、数据共享和数据标准，成为三维GIS应用热点之一。

本标准定义了一种开放式可扩展的数据规范，对倾斜摄影模型、人工建模数据、BIM、点云、三维管线、二维/三维点线面等各类数据进行整合，形成了适用于多源异构、海量三维地理空间数据的格式规范，很好的解决了大规模三维地理空间数据在Web环境下传输与解析这两大问题，为多源三维地理空间数据在不同终端（移动设备，浏览器，桌面电脑）地理信息平台中的存储、融合、共享与互操作等问题提供解决方案，大力推动了我国三维地理空间数据资产安全可控、开放与共享的建设进程。

三维地理空间数据格式

1 范围

本标准规定了三维地理空间数据格式的逻辑结构及物理存储格式,适用于网络环境和离线环境下海量、多源三维地理空间数据的数据传输、交换及高性能可视化,适用于不同终端(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

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

GB/T 23707—2009 地理信息 空间模式

GB/T 30170—2013 地理信息 基于坐标的空间参照

GB/T 30320—2013 地理空间数据库访问接口

ISO 19101 地理信息 参考模型 (Geographic information — reference model)

3 术语和定义

下列术语和定义适用于本标准。

3.1

矢量数据 vector data

由几何元素所表示的数据。

[GB/T 30320-2013]

3.2

数据集 dataset

可以标识的数据集合。

[ISO 19101]

注:数据集是由同种类型数据组成的数据集合,也就是一组数据对象的集合。

3.3

记录 record

有限的、有名称的相关项(对象或值)的集合。

[GB/T 23707--2009]

3.4

纹理贴图金字塔 Mipmap

由一系列被预先计算和优化过的图片组成的文件,每一个层级的小图都是主图的一个特定比例的缩小细节的复制品。

3.5

瓦片数据 tile data

指定地理范围内的空间数据和属性数据。

3.6

字段 field

表示属性信息的集合。

[GB/T 30320-2013]

3.7

字段信息 field info

字段的相关信息，如名称、类型、长度等。

[GB/T 30320-2013]

3.8

坐标 coordinate

用来指示N维空间中点的位置的数值序列。

[GB/T 30170—2013]

3.9

EPSG编码 EPSG Code

The European Petroleum Survey Group发布并维护的一套公用的坐标参考系统，其坐标参考系统的编码称为 EPSG Code。

4 符号和缩略语

4.1 符号

本标准采用的符号描述见表1。

表1 符号对照表

符号	含义
~	表示在符号前后对象之间取值
	表示符号前后项任选其一
[]	表示括号内的对象为可选
n	表示数字
(,)/[,]	表示在符号左端数值到右端数值的区间中任取其一，()表示开区间，[]表示闭区间

4.2 缩略语

本标准采用的缩略语见表2。

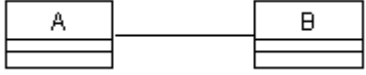
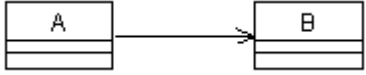
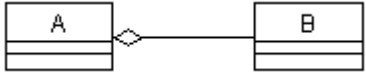
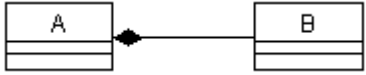
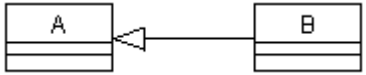
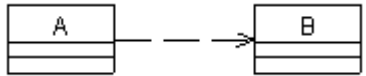
表2 缩写对照表

描述符	中文名称	英文名称
LOD	层次细节	Level of Detail
SRID	空间参照系唯一标识	Spatial Reference System Identifier
UML	统一建模语言	Unified Modelling Language
UUID	通用唯一识别码	Universally Unique Identifier
XML	可扩展标记语言	eXtensible Markup Language

4.3 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3。

表3 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
	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

4.4 UML 多样性描述

本标准涉及的UML图中多样性描述含义见表4。

表4 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

4.5 数据流描述

4.5.1. 字节序规定

本标准涉及的二进制流存储，字节序规定为Little-Endian，即低位字节排放在内存的低地址端。

4.5.2. 基本数据类型定义

本标准涉及的二进制流数据中基本数据类型及其描述见表5。主要用于6.3.2节。

表5 数值数据类型描述

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	[-32768, 32767]	短整型
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

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

4.5.3. 字符串类型

本标准涉及的字符数据类型用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]; //二进制内容
}
```

5 数据组织结构

5.1 数据的逻辑组织结构

本标准基于地理范围对三维地理空间数据按照瓦片方式进行组织，用TileTreeSet对象表示；TileTreeSetInfo是其描述信息，是对数据的整体描述；TileTreeSet如果是基于点、线、面或模型数据集进行构建，则可以有AttributeInfos，表示每个数据集的属性描述信息。

对指定地理范围内的三维地理数据进行空间划分，每个空间划分对应一个采用树结构进行组织的瓦片集合，用TileTree对象表示；IndexTree是其树结构的索引信息；TileTree可以有属性数据AttributeData，记录TileTree中各对象的属性数据。

每个TileTree自上而下逐步细分，每个空间划分对应一个瓦片，用Tile对象表示。

对象之间的关系见图1。

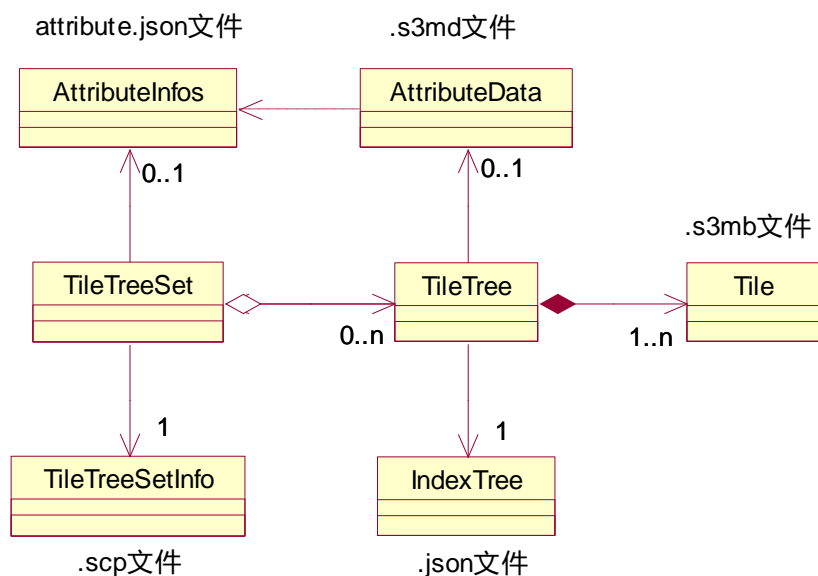


图1 瓦片数据组织 UML 图

5.2 数据的文件组织

本标准规定的数据的主要包括：描述文件、索引树文件、数据文件、属性文件。各对象存储的文件组织形式见表6。

表6 对象存储的文件组织形式

对象	存储形式	文件类型	描述
TileTreeSetInfo	.scp文件	描述文件	整个数据的描述信息
AttributeInfos	attribute.xml	属性描述文件	TileTreeSet中各数据集属性描述信息
TileTree	文件夹	数据文件夹	存放瓦片范围内所有数据
AttributeData	.xml文件	属性数据文件	该Tile下所有对象的属性数据
IndexTree	.json文件	索引树文件	该Tile下所有PagedLOD信息
Tile	.s3mb文件	数据文件	一个S3MB文件存储了该LOD层一个空间划分的数据

描述文件(.scp)和数据文件夹是基础组成部分；描述文件包含每个TileTree的索引文件(.json)路径名；索引文件是对该瓦片数据的树形结构的描述，可以在不加载实际数据的情况下，获取每层的每个瓦片文件的包围盒、LOD的切换信息、挂接的子节点文件等，主要作用是加速瓦片文件检索的效率；属性数据包括属性描述文件(attribute.xml)和每个TileTree中存储各瓦片属性数据的.xml文件，可选。

5.3 树形结构描述

TileTree中所有的Tile构成树形逻辑结构，由PagedLOD描述父子关系，自上而下由粗糙层逐步过渡到精细层。以四叉树为例，其划分结构见图2。

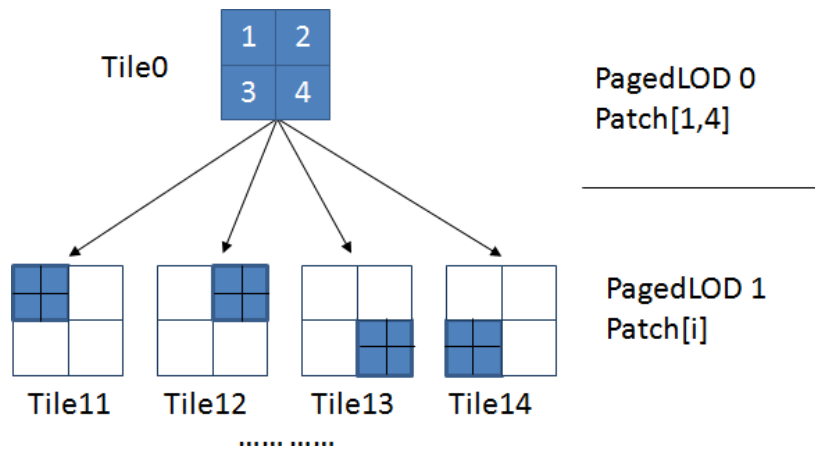


图2 TileTree 的树形结构

每个LOD层级用PagedLOD表示,标识了该Tile在TileTree中所处的LOD层级。

每个PagedLOD在树形结构的横向上又有一个或多个划分，每个划分用Patch表示。每个Patch有零个或一个父Patch，有零个或多个子Patch，每个子Patch是父Patch的一个空间划分；Patch的父子关系构建成树形结构。

每个Patch由零个或多个数据包构成，用Geode表示；每个Geode包含一个或多个实体对象(ModelEntity)，并且有一个矩阵作用于Geode中所有的骨架对象。

ModelEntity分为Skeleton、Material、Texture三种实体类型，分别对应了骨架、材质、纹理三种实体对象；Geode记录了数据包包含的实体对象名字，从而实现同一个实体对象可以被不同Geode引用。

相关对象的UML图见图3。

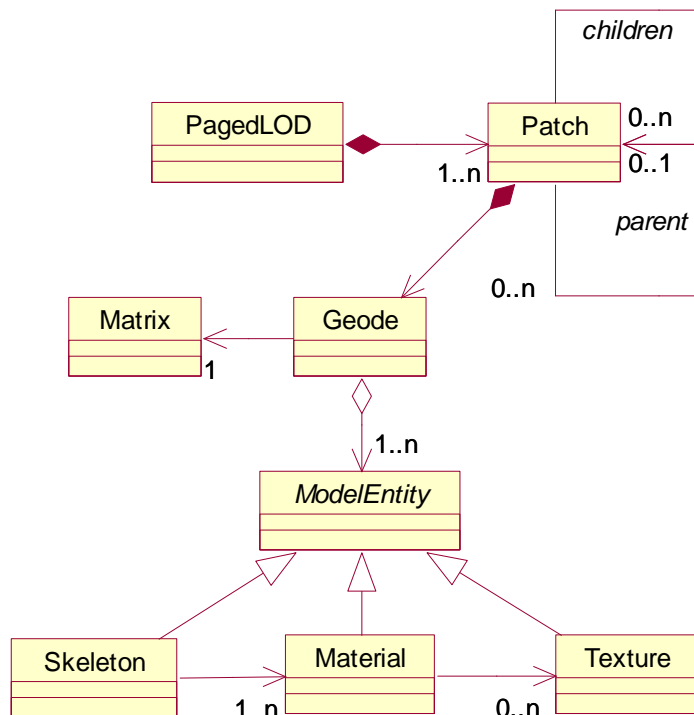


图3 树形结构 UML 图

6 数据的物理存储

6.1 描述文件

6.1.1 描述文件概述

描述文件存储TileTreeSetInfo对象，用于描述数据的基本信息，关联对象的UML图见图4。

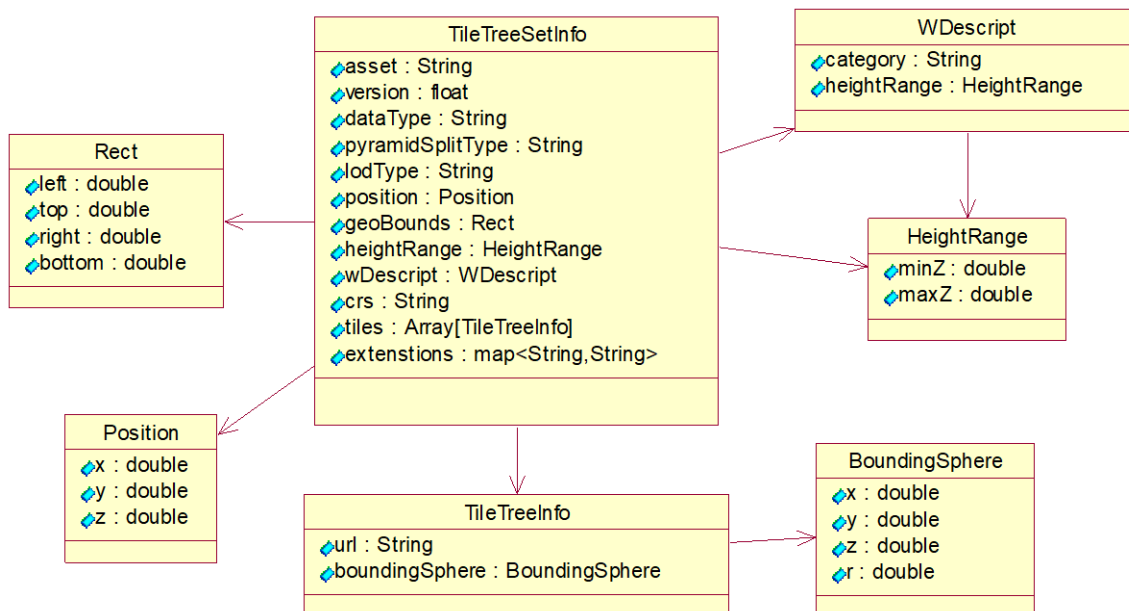


图4 描述文件的 UML 图

6.1.2 描述文件标签信息

描述文件扩展名为. scp (Spatial Cache PagedLOD)，采用JSON文件存储 (UTF8编码，不带BOM头)，各标签含义见表7。

表7 描述文件各标签含义

标签名	类型	描述
asset	String[0..1]	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, BIM, PointCloud, Pipeline} 分别对应：矢量数据，倾斜摄影模型，BIM，点云和管线数据类型。
pyramidSplitType	String	数据的空间剖分类型。 取值范围：{Octree, QuadTree} 分别对应：四叉树、八叉树
lodType	String	LOD类型 取值范围：{Add、Replace} 分别对应：添加、替换
geoBounds	Rect	数据的地理范围，用Rect对象表示，见表8。
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9。
wDescript	WDescript	w位的描述，用WDescript对象表示，见表10。
position	Position	整个TileTreeSet放置的空间点坐标位置，用Position对象表示，包含空间坐标位置和及其单位，见表11。
crs	String	坐标系信息，支持EPSG编码。 表述格式：crs: {‘epsg:4326’}
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13。
extensions	用户自定义	用户扩展信息。

表8 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9 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10 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 位最小值最大值，参见表 9。

表11 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。参见表 12。
unit	String	空间坐标系的单位。 取值范围：{Degree、Meter} 分别对应：度、米

表12 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13 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 14。

表14 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，参见表 12。
min	Point3D	数据包围盒最小角点，用 Point3D 对象表示，参见表 12。

6.2 索引树文件

索引树文件扩展名为.json，采用JSON文件存储（UTF8编码，不带BOM头）。各标签含义见表15。

表15 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，参见表16。
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。参见表17。

表16 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，自顶向下递减，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，参见表 14。
children	Array[TileInfo]	各子节点信息。

表17 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。

6.3 数据文件

6.3.1 S3MB 文件逻辑结构

6.3.1.1 主要结构

数据文件是数据主要组成部分，由.s3mb（Spatial 3D Model Binary）文件组成；每个S3MB文件存储该PagedLOD层一个空间划分范围内的三维地理空间数据。对象UML图见图5。

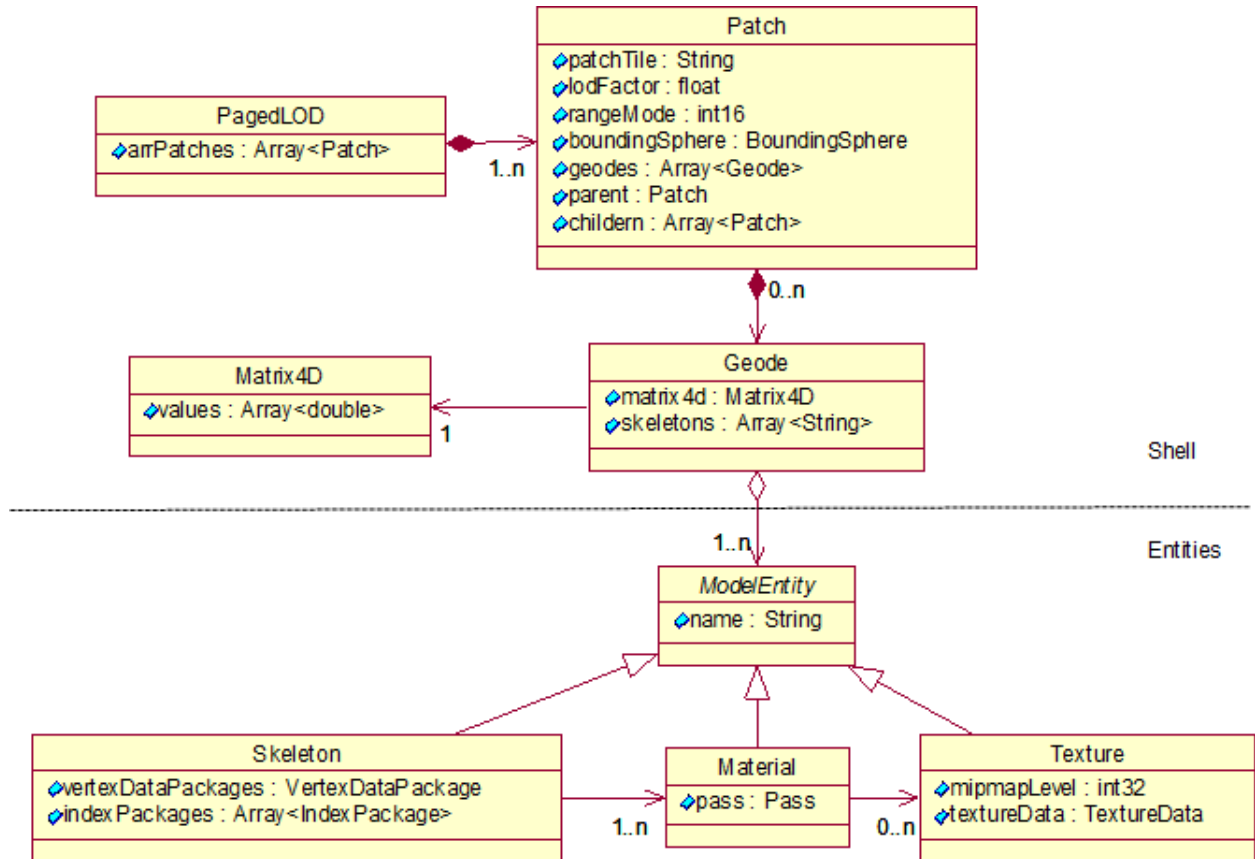


图5 S3MB 文件存储对象 UML 图

- PagedLOD：由 Patch 构成
 - arrPatches：该层 LOD 的所有 Patch 集合
- Patch：包含指定空间范围内的数据
 - patchTile：Patch 所在的 S3MB 文件名
 - lodFactor：切换因子，即 LOD 切换的阈值，与切换模式配合使用
 - rangeMode：切换模式，见 6.3.2.2。
 - boundingShpere：包围球
 - childTile：挂接的子文件的相对路径
 - geodes：空间数据包
 - parent：父节点
 - children：子节点数组，即挂接的子文件中的所有 Patch
- Geode：数据包
 - matrix4d：作用于骨架姿态的矩阵

- skeletons: 骨架名字数组
- Matrix 4D: 4×4 矩阵, 行主序
 - values: 由 16 个 double 值组成的数组
- ModelEntity: 实体对象基类
 - name: 实体名字, 是实体对象在 TileTree 中的唯一标识
- Skeleton: 骨架对象, 继承自 ModelEntity
 - vertexDataPackages: 顶点数据包
 - indexPackages: 顶点索引数据包
- Material: 材质对象, 继承自 ModelEntity
 - pass: 渲染通道
- Texture: 纹理对象, 继承自 ModelEntity
 - mipmaplevel: 纹理对象 MipMap 的层级
 - textureData: 纹理数据

6.3.1.2 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 (IndexPacakge) 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6。

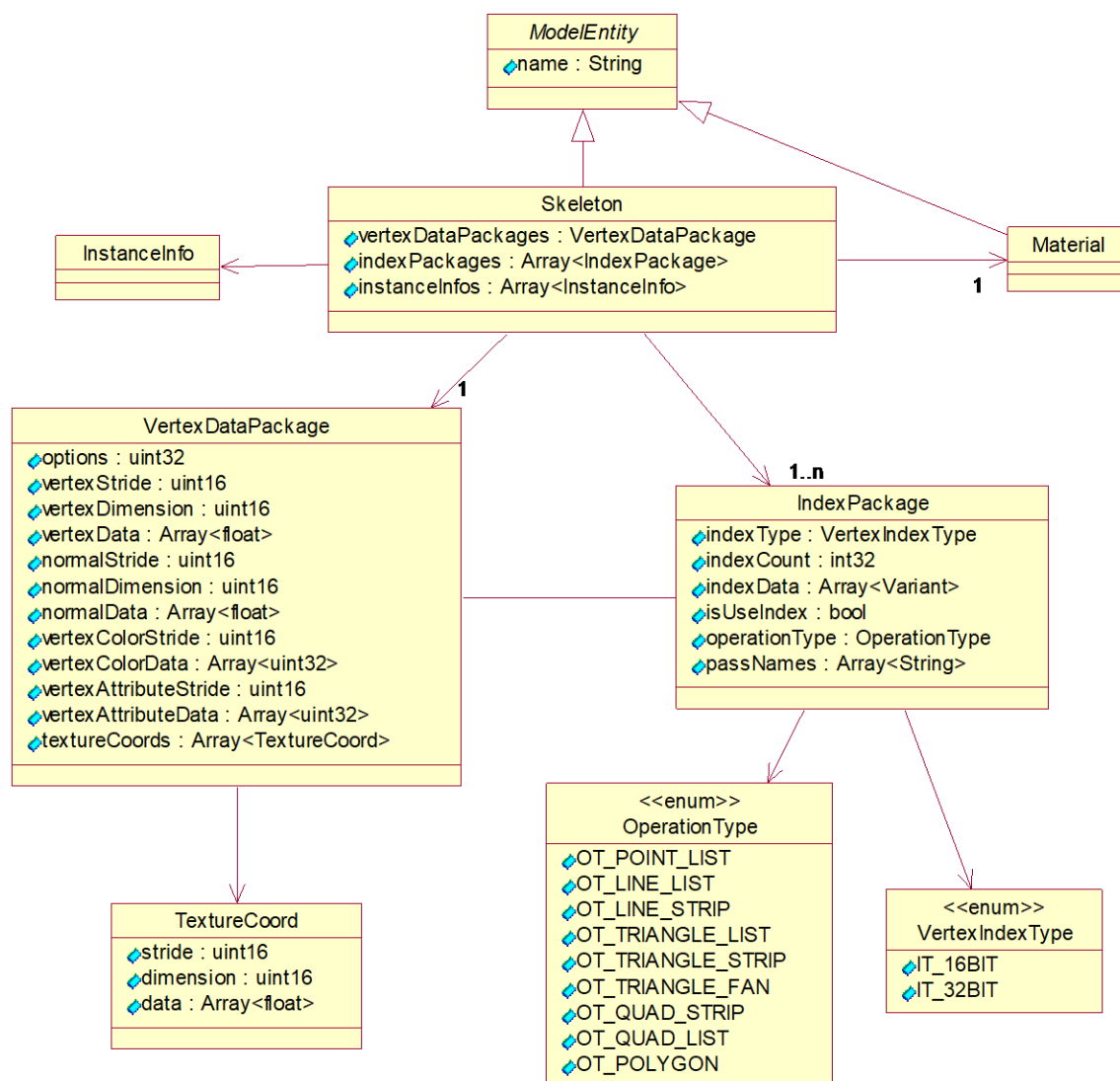


图6 骨架对象 UML 图

——VertexDataPackage: 顶点数据包

- Options: 数据选项信息, 存储扩展数据
- vertexStride: 顶点坐标在数组中的偏移量
- vertexDimension: 顶点维度
- vertexData: 顶点坐标值数组
- normalStride: 法向量在数组中的偏移量
- normalDimension: 法向量的维度
- normalData: 法向量分量值构成的数组
- vertexColorStride: 顶点颜色在数组中的偏移量
- vertexColorData: 顶点颜色数组
- vertexAttributeStride: 顶点属性在数组中的偏移量
- vertexAttributeData: 顶点属性数组 (可以存储骨架所属对象的 ID 等信息)
- textureCoords: 纹理坐标数组
- instanceInfos: 实例化信息数组

——InstanceInfo: 实例化信息

- matrixValues: 4×4 矩阵值

- objectID: 对象 ID
- TextureCoord: 纹理坐标
 - stride: 偏移量
 - dimension: 纹理坐标维度
 - data: 纹理坐标值数组
- IndexPackage: 顶点索引包
 - indexType: 顶点索引类型
 - indexCount: 顶点索引个数
 - indexData: 顶点数据, 根据顶点索引类型, 可能是 Short 数组或 Integer 数组
 - isUseIndex: 是否使用索引
 - operationType: 索引的组织方式
 - passNames: 渲染该对象时使用的 Pass 对象名称
- OperationType: 索引的组织方式
 - OT_POINT_LIST: 单个点
 - OT_LINE_LIST: 两点线
 - OT_LINE_STRIP: 线串
 - OT_TRIANGLE_LIST: 三角形
 - OT_TRIANGLE_STRIP: 条带三角形
 - OT_TRIANGLE_FAN: 扇面三角形构成
 - OT_QUAD_STRIP: 条带四边形
 - OT_QUAD_LIST: 四边形串, 不共享边
 - OT_POLYGON: 多边形
- VertexIndexType: 顶点索引的数据类型
 - IT_16BIT: 16 位无符号整型
 - IT_32BIT: 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 65535 则采用 IT_32BIT, 否则采用 IT_16BIT。

6.3.1.3 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 (UTF8 编码)。材质相关对象的 UML 图见图 7, 各标签含义见表 18。

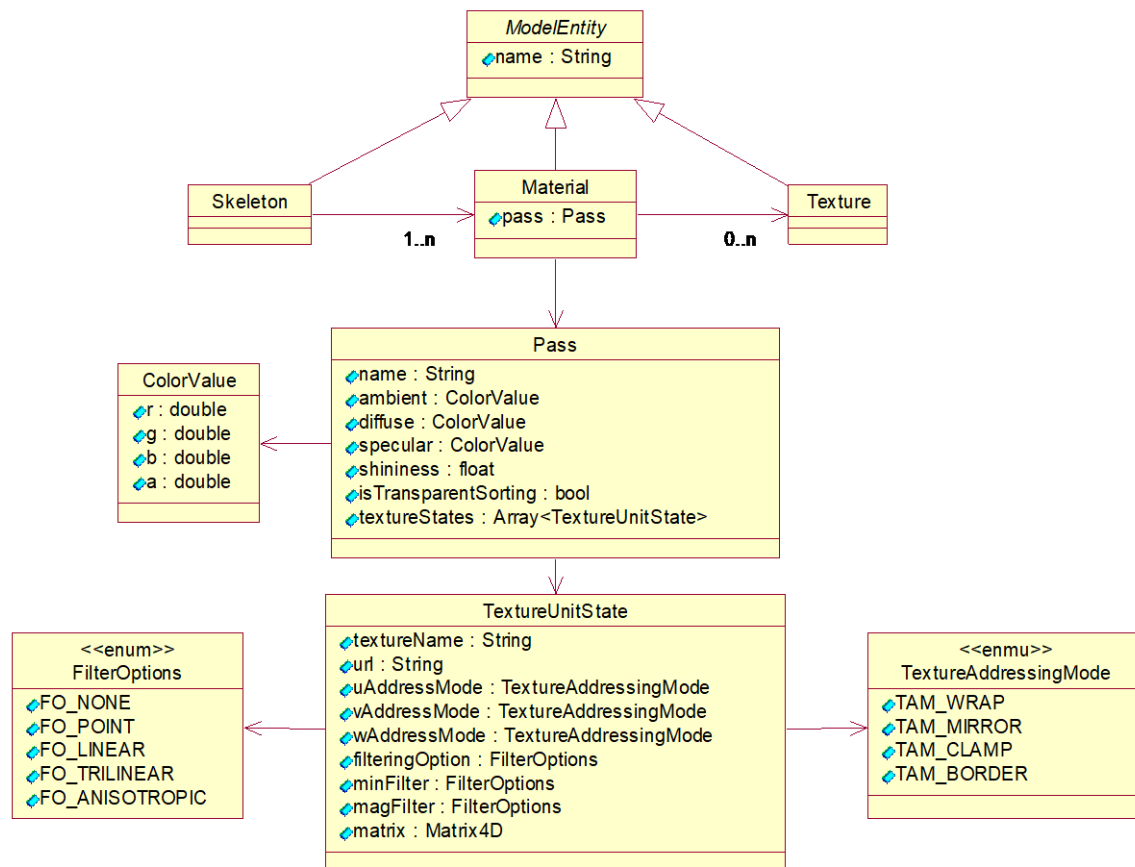


图7 材质对象 UML 图

表18 材质对象各标签含义

标签名	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。详细描述请参见表 19。
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。详细描述请参见表 19。
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。详细描述请参见表 19。
shininess	float	反射光颜色。
isTransparentSorting	boolean	是否使用透明排序。 取值范围: { 'True' , 'False' } 分别对应: 是、否
textureStates	Array[TextureUnitState]	纹理信息,用纹理信息 TextureUnitState 对象表示。包含 textureName 、 url 、 uAddressMode 、 vAddressMode 、 vAddressMode、filteringOption、minFilter、magFilter、matrix 等元素。详细描述请参见表 20。

表19 ColorValue 对象各标签含义

标签名	类型	描述
-----	----	----

r	int	红色分量值
g	int	绿色分量值
b	int	蓝色分量值
a	int	透明度分量值

表20 TextureUnitState 对象

标签名	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 方向上贴图模式，用纹理处理模式 TextureAddressingMode 对象表示，参见表 21。
vAddressMode	TextureAddressingMode	纹理坐标 v 方向上贴图模式。用纹理处理模式 TextureAddressingMode 对象表示，参见表 21。
wAddressMode	TextureAddressingMode	纹理坐标 w 方向上贴图模式。用纹理处理模式 TextureAddressingMode 对象表示，参见表 21。
filteringOption	FilterOptions	纹理插值模式，用 FilterOptions 对象表示，参见表 22。
minFilter	FilterOptions	纹理缩小时采用的插值模式，用 FilterOptions 对象表示，参见表 22。
magFilter	FilterOptions	纹理放大时采用的插值模式，用 FilterOptions 对象表示，参见表 22。
matrix	double[16]	4*4 纹理矩阵，16 个 double 值表示，行主序。

表21 TextureAddressingMode 对象各标签含义

标签名	类型	描述
tAM_WRAP	int	复制整张贴图
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22 FilterOptions 对象各标签含义

标签名	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

6.3.1.4 纹理对象

纹理（Texture）对象UML图见图8。

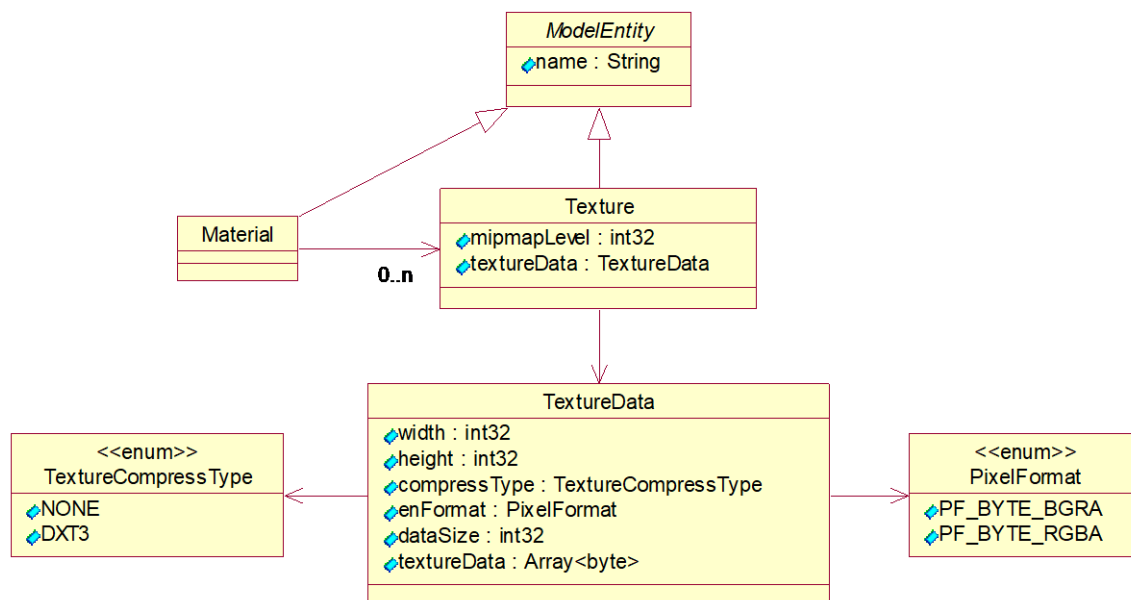


图8 纹理对象 UML 图

- Texture: 纹理对象
 - mipmapLevel: MipMap 层数
 - textureData: 纹理数据
- TextureData: 纹理数据
 - width: 横向像素个数
 - height: 纵向像素格式
 - compressType: 纹理压缩方式
 - enFormat: 纹理的像素格式
 - dataSize: 纹理的二进制流大小
 - textureData: 纹理数据二进制流
- PixelFormat: 像素格式
 - PF_BYTE_BGRA: BGRA 格式
 - PF_BYTE_RGBA: RGBA 格式
- TextureCompressType
 - None: 无压缩格式
 - DXT3: DXT3 压缩格式

6.3.2 S3MB 文件的二进制流描述

6.3.2.1 S3MB 文件的主要组成部分

S3MB文件采用二进制流形式存储，包含Header、Shell和Entities三大部分，其中Shell和Entities采用zip压缩存储，见图9。

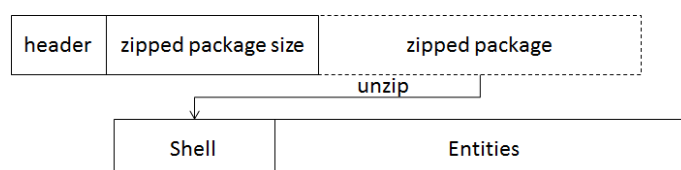


图9 S3MB 文件二进制数据包

```

S3MBFile{
    float header;           //S3MB文件版本号
    uint32 nZippedSize;      // zippedPackage的字节数
    byte* zippedPackage;     //Shell和Entities压缩后的字节流
};

```

Shell存储PagedLOD、Patch、Geode对象；Entities即实体数据，包括骨架(Skeleton)、材质(Material)、纹理(Texture)。Shell和Entities的对象结构图参见图5；

6.3.2.2 Shell 的二进制流描述

```

Shell{
    int32 lodCount;
    PagedLOD pagedLods[lodCount];
};

PagedLOD{
    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];
};

Enum 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];
};

```

```
Matrix4d{                                // 4×4矩阵，行主序
    double values[16];
};
```

6.3.2.3 Entities 的二进制流描述

```
Entities{
    int32 skeletonCount;                //骨架
    Skeleton skeletons[skeletonCount];
    int32 materialCount;                //材质
    Material materials[materialCount];
    int32 textureCount;                //纹理
    Texture textures[textureCount];
};
```

```
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
```

```

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;
    IndexType enIndexType; //存储为byte
    byte reserved;
    OperationType opType; //存储为byte
    byte reserved;
    variant indexData[indexCount]; //索引值，说明：若IndexType为IT_16BIT，则variant类型为uint16；若IndexType为IT_32BIT，则variant类型为uint32
    int32 passCount;
    String passNames[passCount];
};

IndexType{
    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
};

```

```
Material{
    String strMaterial;    //材质采用json描述，用String类型存储，UTF8编码
};

Texture{
    String strName;
    int32 mipMapLevel;
    TextureData texData;
};

TextureData{
    int32 width;
    int32 height;
    TextureCompressType compressType;    //存储为uint32
    int32 datasize;
    PixelFormat enPixelFormat;           //存储为uint32
    byte data[datasize];
};

TextureCompressType{
    TC_NONE = 0,
    TC_DXT3 = 14,
};

PixelFormat{
    PF_BYTE_BGRA = 12,
    PF_BYTE_RGBA = 13,
};
```

6.4 属性文件

6.4.1 属性文件的构成

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；属性数据文件名与瓦片数据根节点文件名相同，扩展名为.s3md，与数据文件（.s3mb）处于同级目录。

6.4.2 属性描述文件

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（UTF8编码，不带BOM头）。各标签含义见表23。

表23 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。详细描述请参见表24。

表24 Layer Info 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。详细描述请参见表25。
fieldInfos	Array[FieldInfo]	字段信息集合,用FieldInfo对象表示,参见表26。

表25 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26 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
type	String	字段类型。 取值范围: {'bool','int16','uint16','int32','uint32','int64','float','double','wchar','String' } 对应的含义及取值范围见表5及4.5.3小节。
size	int32	字段长度
isRequired	boolean	是否是必填字段 取值范围: {'True','False'} 分别对应: 是、否

6.4.3 属性数据文件

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格式文件存储（UTF8编码，不带BOM头），并采用zip压缩，见图10。

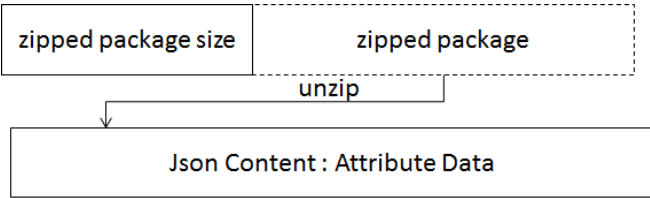


图10 属性数据文件二进制流结构

二进制流结构如下：

```
AttributeData{
    uint32 nZippedSize;      // zippedPackage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27。

表27 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。详细描述请参见表 28。

表28 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 25 。
fieldInfos	Array[FieldInfo]	字段信息集合，用 FieldInfo 对象表示，见表 26 。
records	Array[Record]	各对象的属性值记录信息集合，用 Record 对象表示，见表 29。

表29 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 对象表示，见表 30。

表30 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
(参考性附录)
数据示例

A.1 描述文件示例

```
{
  "asset": "SuperMap",
  "version": 1.0,
  "dataType": "BIM",
  "pyramidSplitType": "QuadTree",
  "lodType": "Replace",
  "position":
    {
      "x": 116.36,
      "y": 39.99,
      "z": 0.0,
      "units": "Degree"
    },
  "geoBounds":
    {
      "left": 116.3635,
      "top": 40.0018,
      "right": 116.3755,
      "bottom": 39.9932
    },
  "heightRange":
    {
      "min": 9.4875,
      "max": 119.9612
    },
  "wDescript":
    {
      "category": "",
      "range":
        {
          "min": 0.0,
          "max": 0.0
        }
    },
  "tiles":
    [
      {
        "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
        }
    },
    {
        "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

```
{
  "lodTreeExport":
  {
    "name": "Tile_-7281_21185_0000",
    "tileInfo":
    {
      "lodNum": 0,
      "modelPath": "Tile_-7281_21185_0000.s3mb",
      "rangeMode": "pixelSizeOnScreen",
      "rangeValue": 1.0,
      "boundingBox":
      {
        "min":
        {
          "x": -68.02222442626953,
          "y": -43.73067092895508,
          "z": 9.495752334594727
        },
        "max":
        {
          "x": 68.02222442626953,
          "y": 43.73127746582031,
          "z": 119.96125030517578
        }
      }
    },
    "children":
    [
      {
        "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":[...]
  }
],
"status":
{
  "lodCount":5,
  "tilesCount":6
}
}
```

A. 3 材质内容示例

```
{
  "materials":
  [
    {
      "material":
      {
        "id": "0_10710_Sec_0005_-7281_21185_0000_0000_0",
        "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,
        "transparentsorting": false,
        "textureunitstates":
        [
          {
            "textureunitstate":
            {
              "id": "0_10710_Sec_0005_-7281_21185_0000_0000",
              "url": "",
              "addressmode": { "u": 0, "v": 0, "w": 0 },
              "filteringoption": -842150451,
              "filtermin": 2,
              "filtermax": 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

```
{
  "layerInfos":
  [
    {
      "layerName": "Building_Sub",
      "idRange": { "minID": 1, "maxID": 10 },
      "fieldInfos":
      [
        {
          "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

```
{
  "layerInfos":
  [
    {
      "idRange":
      {
        "minID": 1,
        "maxID": 1
      },
      "fieldInfos":
      [
        {
          "name": "SmID",
          "alias": "SmID",
          "type": "int32",
          "size": 4,
```



```
        "isRequired":true
      },
      {
        "name":"MODELNAME",
        "alias":"ModelName",
        "type":”String”,
        "size":30,
        "isRequired":false
      },
    ]
  ]
  "records":
  [
    {
      "id":1,
      "values":
      [
        {
          "name":"SmID",
          "value":1
        },
        {
          "name":" ModelName",
          "value ":"鸟巢"
        },
      ]
    }
  ]
}
}
```
