

CesiumLab 地理信息基础数据处理平台

V3.0.4

使用手册

CesiumLab

目录

1	软件安装以及注册	6
1.1	下载安装	6
1.2	注册登录	7
1.2.1	登录	7
1.2.2	注册	8
1.2.3	合作伙伴	9
1.3	修改密码	9
1.4	授权类型	10
1.4.1	本机授权信息	10
1.5	申请测试授权	11
1.6	购买授权	12
1.6.1	单机授权安装	12
1.6.2	USB 授权	13
1.7	软件设置	14
1.7.1	端口设置	14
2	数据处理	15
2.1	数据处理基础和处理流程	15
2.1.1	切片格式	16
2.1.1	输入数据	18
2.1.2	传统 GIS 数据	19
2.1.3	人工模型	22
2.1.4	BIM 模型	24
2.1.5	倾斜实景数据	25
2.1.6	点云数据	26
2.2	通用功能	26
2.2.1	文件以及目录选择	26
2.2.2	空间参考	28
2.2.3	数据存储	33
2.3	地形切片	40
2.3.1	输入文件	41
2.3.2	处理参数	44
2.3.3	输出文件	46
2.3.4	增量融合	46
2.4	影像切片	46
2.4.1	输入文件	47
2.4.2	处理参数	51
2.4.3	输出文件	56
2.4.4	增量融合	56
2.5	点云切片	56
2.5.1	输入文件	57
2.5.2	资源库	58
2.5.3	处理参数	59

2.5.4	Draco 压缩	60
2.5.5	属性字段	61
2.5.6	输出文件	62
2.6	实例模型切片	62
2.6.1	输入文件	63
2.6.2	位置姿态	68
2.6.3	高程	69
2.6.4	字段列表	70
2.6.5	存储类型	70
2.7	倾斜模型切片	70
2.7.1	输入文件	71
2.7.2	重建顶层	73
2.7.3	效果参数	77
2.7.4	压缩参数	77
2.7.5	存储类型	78
2.7.6	倾斜处理时间	79
2.7.7	合并多块索引	79
2.8	通用模型切片	79
2.8.1	输入文件	80
2.8.2	Obj 参数设置	81
2.8.3	Fbx 参数设置	83
2.8.4	Shp 参数设置	84
2.8.5	Ifc 参数设置	90
2.8.6	Clm 参数设置	90
2.8.7	资源库	96
2.8.8	小场景处理器	96
2.8.9	八叉树处理器	99
2.8.10	压缩参数	102
2.8.11	属性字段	103
2.8.12	数据存储	104
2.8.13	模型结构树	105
2.9	Revit 导出 clm	108
2.9.1	安装插件	108
2.9.2	操作过程	109
2.9.3	导出参数	109
2.10	Microstation 导出 clm	111
2.10.1	安装插件	111
2.10.2	操作过程	111
2.10.3	导出参数	113
2.11	散列紧凑互转	115
2.11.1	输入文件	115
2.11.2	输出文件	115
2.12	数据库配置	116
2.13	工具使用统计	116

2.14	正在运行	117
2.15	处理日志	118
2.16	原始数据管理	119
2.16.1	添加	120
2.16.2	删除	120
2.16.3	修改	121
2.16.4	处理	121
2.16.5	查询	122
3	分发服务	123
3.1	首页	123
3.1.1	服务统计	124
3.1.2	从 cesiumlab2 导入服务	124
3.2	常规影像服务	124
3.2.1	添加	124
3.2.2	删除	125
3.2.3	修改	125
3.2.4	查询	126
3.2.5	服务操作	126
3.3	实时缓存影像服务	126
3.3.1	添加	127
3.3.2	删除	128
3.3.3	修改	128
3.3.4	查询	128
3.3.5	服务操作	129
3.4	影像集合服务	129
3.4.1	添加	129
3.4.2	删除	129
3.4.3	修改	130
3.4.4	查询	130
3.4.5	服务操作	130
3.5	常规地形服务	130
3.5.1	添加	131
3.5.2	删除	131
3.5.3	修改	132
3.5.4	查询	132
3.5.5	服务操作	132
3.6	实时缓存地形服务	132
3.6.1	添加	133
3.6.2	删除	133
3.6.3	修改	134
3.6.4	查询	134
3.6.5	服务操作	134
3.7	地形集合服务	134
3.7.1	添加	135

3.7.2	删除	135
3.7.3	修改	136
3.7.4	查询	136
3.7.5	服务操作	136
3.8	3dtiles 服务	136
3.8.1	添加	137
3.8.2	删除	137
3.8.3	修改	137
3.8.4	查询	137
3.8.5	服务操作	138
3.9	场景服务	138
3.9.1	添加	138
3.9.2	删除	139
3.9.3	修改	139
3.9.4	查询	139
3.9.5	预览、复制路径	139
3.9.6	复制场景	139
3.10	lod 模型库服务	140
3.10.1	添加	140
3.10.2	删除	140
3.10.3	查询	140
3.10.4	打开文件、复制路径	141
3.11	资源服务	141
3.11.1	添加	141
3.11.2	删除	142
3.11.3	修改	142
3.11.4	查询	142
3.11.5	预览、复制路径	142
4	三维可视	142

当前技术架构下，为了最大化提升三维数据可视化效率，几乎所有的三维数据都要进行预处理过程，常见的流程如下：



数据预处理过程把多源异构的 GIS 数据存储为切片格式。通过标准的 HTTP 数据分发服务送给数据可视化端软件展示。

开源的三维可视化端目前有 Cesium 平台，Unreal Engine 平台，OsgEarth 平台等。

开源的 HTTP 数据服务有 tomcat, nginx 等。

但是数据预处理部分基本没有完整的开源产品，CesiumLab 就是定位围绕 Cesium 平台的一套完整易用的数据预处理工具集，并且已经内置了一个启动可用的轻量高性能 http 分发服务，以及集成了我们另外一个产品 EarthSDK 做为数据预览工具，所以整个 CesiumLab 的功能三大部分：

- 1，数据预处理，包括地形、影像、模型、矢量、倾斜、BIM、点云等相关切片工具。
- 2，分发服务，包括影像切片服务，地形切片服务、3dtiles 切片服务等。
- 3，EarthSDK 预览页面，用来基于 EarthSDK 整合三维场景。

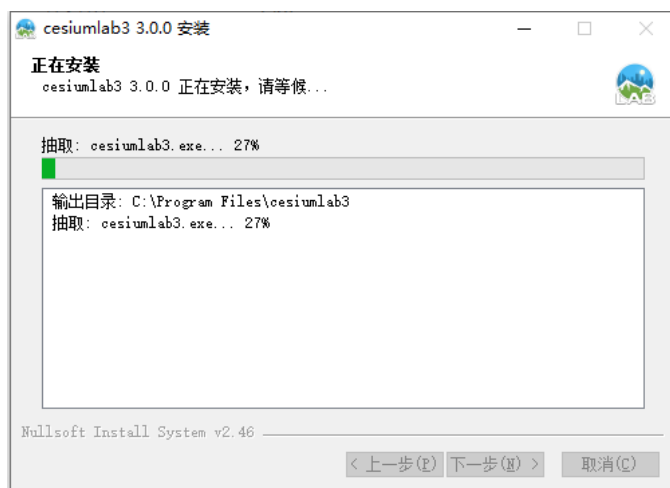
1 软件安装以及注册

1.1 下载安装

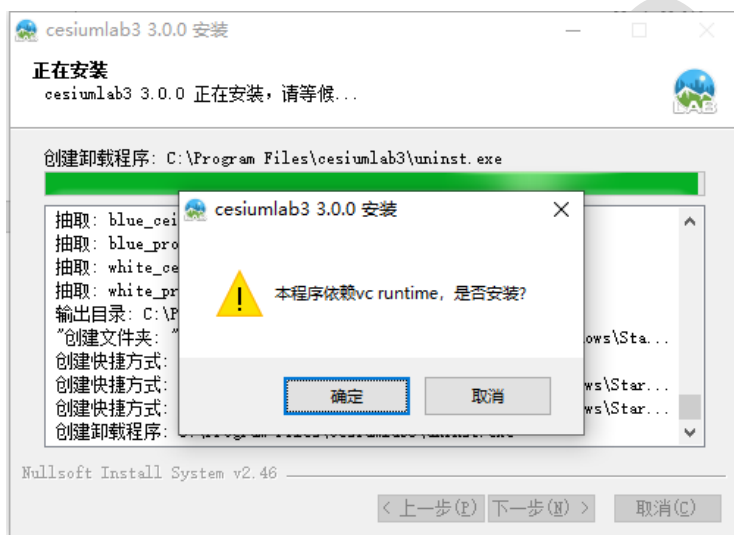
<http://cesiumlab.com/#/update> 官网下载安装文件。

 cesiumlab3_3.0.0.exe	2021/11/13 16:49	应用程序	179,334 KB
--	------------------	------	------------

双击 exe 安装。



安装过程会提示 本程序依赖 VC2019 实时运行库, 如本机已安装了 VC2019 实时运行库, 可以点击【取消】, 否则请点击【确定】。



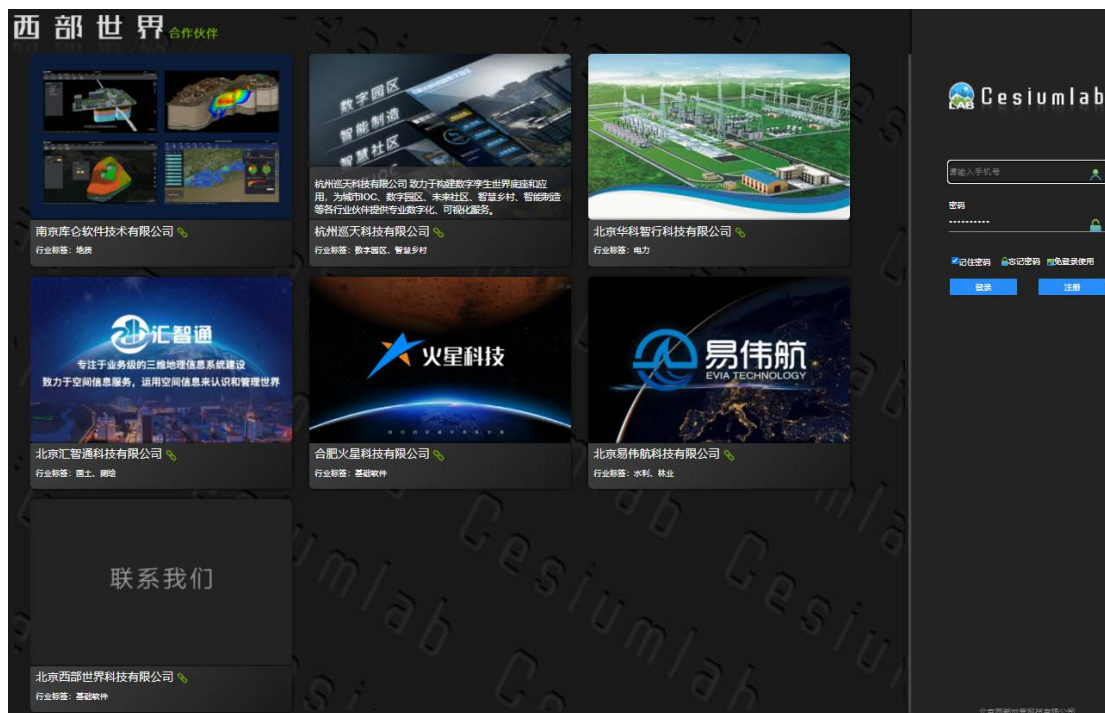
☆ 建议使用 win10 系统。

1.2 注册登录

1.2.1 登录

在 CesiumLab3 登录界面, 直接输入账号密码(同 CesiumLab2 账号密码), 点击【登录】, 提示登录成功。

☆ 单机授权和 USB 授权用户无需登录。



CesiumLab 登录界面

1.2.2 注册

若没有 CesiumLab 账号，在登录界面点击【注册】，跳转到如下注册界面，填写信息后，点击【免费注册】。

CesiumLab 注册界面

1.2.3 合作伙伴

CesiumLab 本身是一款地理信息基础数据处理平台软件,旨在为在地理信息领域里奋斗的伙伴们提供力所能及的帮助,您的成功是我们最大的动力。为此,我们在登录界面把最容易被用户看到的空间让给了合作伙伴。如果需要在此展示您的风采,请备好资料联系我们,期待您的加入。



1.3 修改密码

进入主页后, 点击软件右上角的用户按钮, 下拉选择修改密码。输入新密码, 确认完成修改。



1.4 授权类型

CesiumLab 授权类型分为：在线授权、单机授权、USB 授权。

a) **在线授权（即免费用户）**：必须连接互联网，登录后才可使用，仅可用基础功能，高级功能无法使用（软件中所有金色字体 **重建顶层** 的功能即为高级功能）。

b) **单机授权**：无需连接互联网，无需登录，绑定一台电脑，可用全部功能。

c) **USB 授权**：无需连接互联网，无需登录，随 USB 插拔使用任意电脑，可用全部功能。

1.4.1 本机授权信息

软件内，一次点击【授权详情】，【本机授权信息】查看本机授权



授权有效期显示当前授权过期时间，注意在线授权的有效期为登录过期时间。

1.5 申请测试授权

在购买正式授权前，可以申请测试授权。测试授权使用期限为 3 天。

如果是联网机器，申请流程如下：

点击授权详情>>我的授权>>单机，点击【申请授权】，弹出申请授权界面，填写申请信息。

☆**请务必填写单位名称全称，同时上传名片、工牌等能证明您就职于该单位的图片，有助于通过审核！**



待审核通过，点击【刷新】，下方授权列表中会出现授权记录。点击【安装】，提示安装成功，用户便可以使用 CesiumLab 所有功能了。

本机授权信息

我的授权

单机

USB

授权购买

申请授权

刷新

有效

全部

机器码: [redacted] 1

序号	机器码	用户名	开始时间	结束时间	过期状态	操作
1	106[redacted]a9 4ee161	Cesiumlab	2021-11-05 00:00:00	2021-12-31 23:00:00	有效	安装
2	8[redacted]6b 7025dc	Cesiumlab	2021-11-02 00:00:00	2021-12-02 23:00:00	有效	下载
3	[redacted]b8 2b1684	Cesiumlab	2021-11-02 00:00:00	2021-12-02 23:00:00	有效	下载
4	[redacted]5 0e294U	Cesiumlab	2020-07-10 00:00:00	2021-08-31 23:00:00	过期	

如果是离线机器申请，在登录界面点击【免登录使用】，会显示离线机器码。将该机器码记录下来，再用联网机器申请（步骤同上），注意申请时机器码处填写离线机器的机器码。

待审核通过，在授权列表中点击【下载】。将下载的 CesiumLab.lic 放到离线机器的 C:\Users\用户名\cesiumlab3 目录下。

在登录界面点击【刷新授权】，此时授权状态更新为单机授权。

点击【进入主页】。



1.6 购买授权

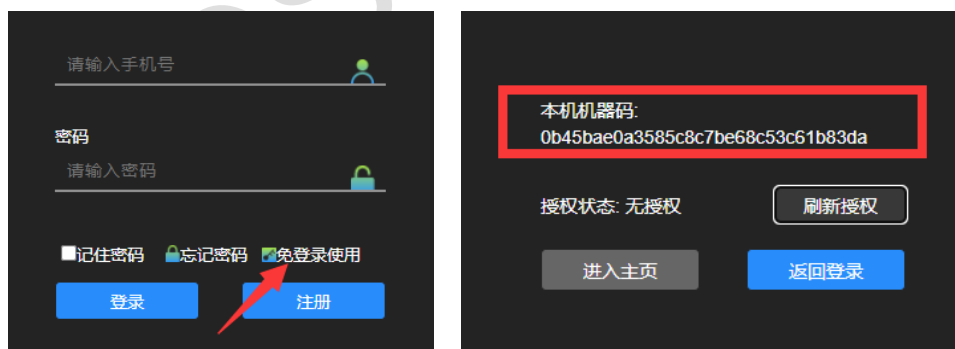
如需购买授权，请登录 <http://cesiumlab.com/#/service>，可查看授权类型、对应价格、开票信息和销售联系电话等。

付款后我们会发放授权。对于单机授权，我们会添加到购买用户的手机号注册账户里。对于 usb 授权，我们会通过顺丰发送到客户指定位置。

1.6.1 单机授权安装

单机授权为硬件绑定机制：更换硬盘（包括插拔 usb 硬盘）、重装系统、切换网卡等任何的硬件改动都可能导致机器码改变，从而导致授权失效。所以购买单机授权用户要确保保持授权机器在有效期内不会做任何改动。因为我方无法判定您是更换机器还是硬件改动，所以授权失效后，我们也不承诺重新发放授权!!!

请先在您待授权的机器安装最新版的 CesiumLab3，安装之后登录界面，点击【免登录使用】等待片刻，会显示本机的机器码，请把机器码拷贝（不要截图或拍照，请拷贝）发给销售。



1.6.1.1 在线安装授权

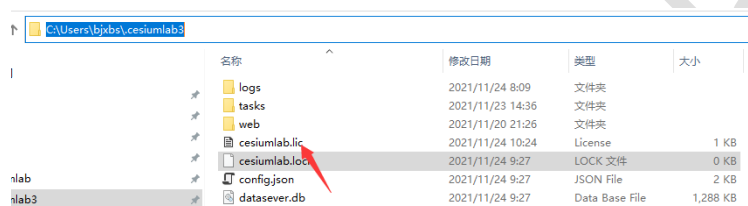
如果待授权的机器为可上网机器，和销售联系确认已经发放授权之后，如下操作。登录后，点击授权详情>>我的授权>>单机，刷新，再点【安装】即可。



1.6.1.2 离线安装授权

如果待授权的机器不能连接互联网，那么联系销售，销售会发送授权文件。
将收到的 cesiumlab.lic 文件放到：

C:\Users\【您的 windows 用户名】\.cesiumlab3 目录



在登录界面点击【免登录使用】，刷新授权，授权状态更新为单机授权，点【进入主页】即可。



1.6.2 USB 授权

将 USB 插入当前 CesiumLab3 的安装机器，点击【免登录使用】，再刷新授权，授权状态更新为 usb 授权，点击【进入主页】即可。



1.7 软件设置

软件右上角找到设置按钮。



1.7.1 端口设置

在下拉菜单中点击【端口设置】按钮，弹出设置界面，可以在此修改 CesiumLab 端口号，点【应用】即可。

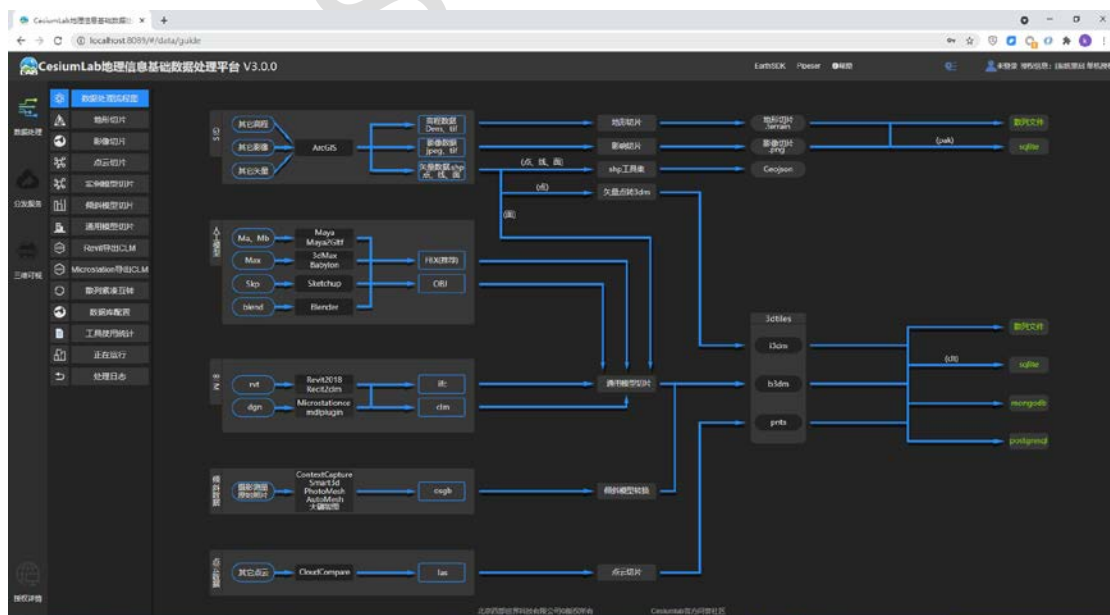


2 数据处理

数据处理包括下述工具和模块：



2.1 数据处理基础和流程



数据处理流程图

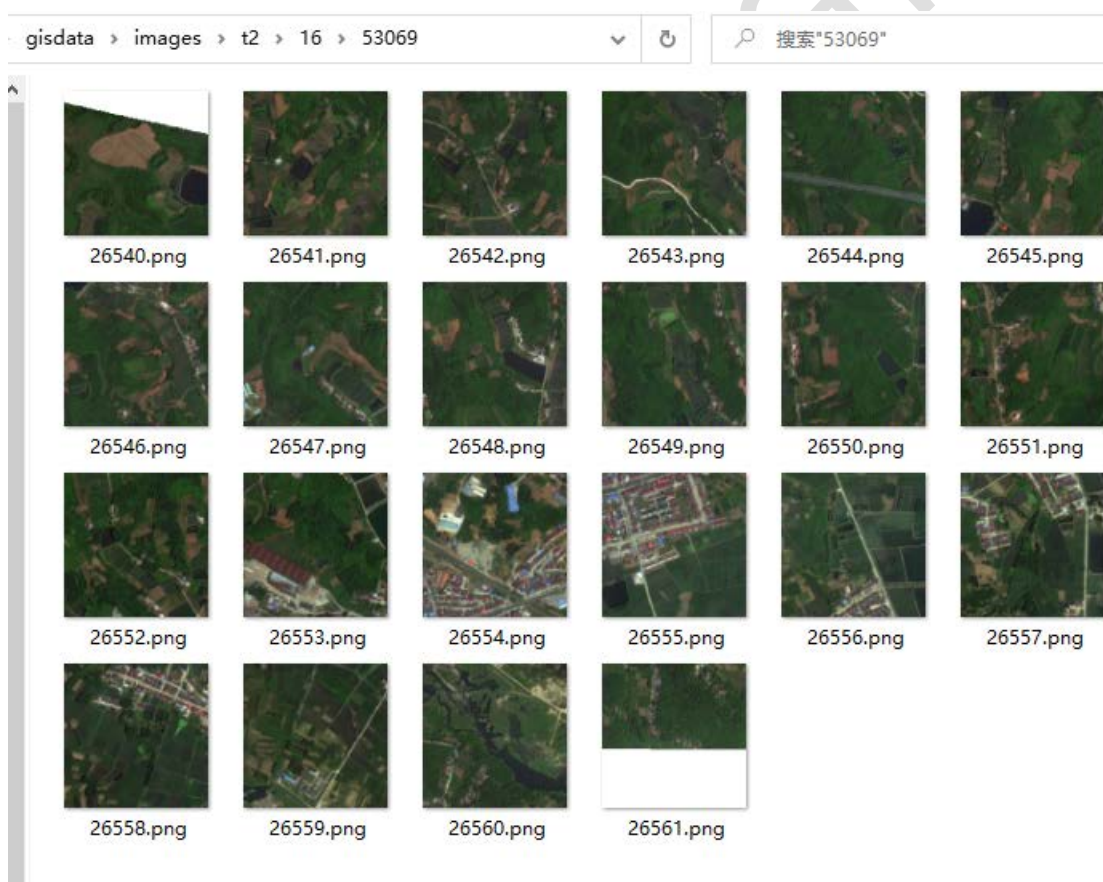
数据处理的目的是把各种来源、各种格式的输入数据，处理为三维 GIS 上能够渐进加载的切片文件格式。整个过程是一个输入到输出的过程，我们先讨论输出的切片格式。

2.1.1 切片格式

根据可视化的效果需求，目前至少以下几种切片形式：

2.1.1.1 影像切片

和传统二维 GIS 上的切片类似，它的切片文件格式是常规图片，比如 png 或者 jpg，可以直接在二维引擎上浏览查看。Cesium 目前支持的影像切片规范有两种形式的，一种是经纬度直拉切片方式（GeographicTilingScheme），一种是 web 墨卡托切片方式（WebMercatorTilingScheme）。CesiumLab3 已经都支持了，后续影像切片会详述。



影像切片

2.1.1.2 地形切片

地形是通过三角网来描述地表起伏的一类数据。一般来说地形是一个三维引擎的基础数据，很多其他数据的显示都要依赖地形，比如影像数据需要贴合在地形上展示，POI 点需要采样地形高程，鼠标交互需要获取鼠标交互位置点。

Cesium 的地形切片格式以及标准，请参考 Cesium 官方的文档：
<https://github.com/CesiumGS/quantized-mesh>

data > dem > t1 > 12 > 6671

名称	修改日期	类型	大小
2548.terrain	2021/6/28 10:07	TERRAIN 文件	4 KB
2549.terrain	2021/6/28 10:07	TERRAIN 文件	3 KB
2550.terrain	2021/6/28 10:07	TERRAIN 文件	1 KB
2551.terrain	2021/6/28 10:07	TERRAIN 文件	17 KB
2552.terrain	2021/6/28 10:07	TERRAIN 文件	15 KB
2553.terrain	2021/6/28 10:07	TERRAIN 文件	8 KB
2554.terrain	2021/6/28 10:07	TERRAIN 文件	6 KB
2555.terrain	2021/6/28 10:07	TERRAIN 文件	6 KB
2556.terrain	2021/6/28 10:07	TERRAIN 文件	4 KB
2557.terrain	2021/6/28 10:07	TERRAIN 文件	5 KB
2558.terrain	2021/6/28 10:07	TERRAIN 文件	12 KB
2559.terrain	2021/6/28 10:07	TERRAIN 文件	4 KB
2560.terrain	2021/6/28 10:07	TERRAIN 文件	2 KB
2561.terrain	2021/6/28 10:07	TERRAIN 文件	3 KB
2562.terrain	2021/6/28 10:07	TERRAIN 文件	3 KB
2563.terrain	2021/6/28 10:07	TERRAIN 文件	2 KB
2564.terrain	2021/6/28 10:07	TERRAIN 文件	3 KB
2565.terrain	2021/6/28 10:07	TERRAIN 文件	2 KB
2566.terrain	2021/6/28 10:07	TERRAIN 文件	2 KB
2567.terrain	2021/6/28 10:07	TERRAIN 文件	2 KB
2568.terrain	2021/6/28 10:07	TERRAIN 文件	4 KB
2569.terrain	2021/6/28 10:07	TERRAIN 文件	2 KB
2570.terrain	2021/6/28 10:07	TERRAIN 文件	1 KB
2571.terrain	2021/6/28 10:07	TERRAIN 文件	1 KB

地形切片

2.1.1.3 模型切片

对于 Cesium 引擎 和 CesiumLab 工具，模型切片就是 3dtiles 格式。3dtiles 的切片规范如下：

<https://github.com/CesiumGS/3d-tiles/tree/main/specification>

3dtiles = 索引 + 内容

索引是 json 格式的元数据定义，包括数据范围、几何误差等，索引可以嵌套引用其他索引。

内容可以进一步分为 3 种内容格式 和 1 个集合格式，具体如下：

格式	用途
Batched 3D Model (b3dm)	多种三维模型格式，例如有纹理的地形数据，包含内外结构的三维建筑，大范围的模型数据

格式	用途
Instanced 3D Model (i3dm)	实例化三维模型，例如森林、路灯和垃圾桶等城市附属物、设备零部件等。
Point Cloud (pnts)	海量点数据
Composite (cmpt)	仅仅用来把上述多个不同格式的文件组织为一个文件。

3dtiles 内容格式

名称	修改日期	类型	大小
 Tile_+000_+003.b3dm	2021/9/15 23:45	B3DM 文件	72 KB
 Tile_+000_+003.json	2021/9/16 9:22	JSON File	3 KB
 Tile_+000_+003_L19_0010t3.b3dm	2021/9/15 23:45	B3DM 文件	8 KB
 Tile_+000_+003_L19_0020t3.b3dm	2021/9/15 23:45	B3DM 文件	102 KB
 Tile_+000_+003_L19_0020t3.json	2021/9/16 9:22	JSON File	20 KB
 Tile_+000_+003_L19_0030t3.b3dm	2021/9/15 23:49	B3DM 文件	125 KB
 Tile_+000_+003_L19_0030t3.json	2021/9/16 9:22	JSON File	43 KB
 Tile_+000_+003_L20_00130t3.b3dm	2021/9/15 23:45	B3DM 文件	8 KB
 Tile_+000_+003_L20_00210t3.b3dm	2021/9/15 23:45	B3DM 文件	42 KB
 Tile_+000_+003_L20_00220t3.b3dm	2021/9/15 23:45	B3DM 文件	107 KB
 Tile_+000_+003_L20_00230t3.b3dm	2021/9/15 23:46	B3DM 文件	189 KB
 Tile_+000_+003_L20_00300t3.b3dm	2021/9/15 23:49	B3DM 文件	81 KB
 Tile_+000_+003_L20_00310t4.b3dm	2021/9/15 23:50	B3DM 文件	68 KB
 Tile_+000_+003_L20_00320t3.b3dm	2021/9/15 23:51	B3DM 文件	168 KB
 Tile_+000_+003_L20_00330t4.b3dm	2021/9/15 23:54	B3DM 文件	94 KB
 Tile_+000_+003_L21_001330t3.b3dm	2021/9/15 23:45	B3DM 文件	8 KB
 Tile_+000_+003_L21_002200t3.b3dm	2021/9/15 23:45	B3DM 文件	10 KB
 Tile_+000_+003_L21_002210t3.b3dm	2021/9/15 23:45	B3DM 文件	56 KB
 Tile_+000_+003_L21_002220t3.b3dm	2021/9/15 23:45	B3DM 文件	284 KB
 Tile_+000_+003_L21_002230t3.b3dm	2021/9/15 23:46	B3DM 文件	409 KB
 Tile_+000_+003_L21_002300t3.b3dm	2021/9/15 23:47	B3DM 文件	336 KB
 Tile_+000_+003_L21_002310t3.b3dm	2021/9/15 23:47	B3DM 文件	360 KB
 Tile_+000_+003_L21_002320t3.b3dm	2021/9/15 23:48	B3DM 文件	376 KB
 Tile_+000_+003_L21_002330t3.b3dm	2021/9/15 23:48	B3DM 文件	179 KB
 Tile_+000_+003_L21_003010t5.b3dm	2021/9/15 23:49	B3DM 文件	35 KB
 Tile_+000_+003_L21_003020t4.b3dm	2021/9/15 23:49	B3DM 文件	243 KB
 Tile_+000_+003_L21_003030t4.b3dm	2021/9/15 23:50	B3DM 文件	376 KB
 Tile_+000_+003_L21_003100t4.b3dm	2021/9/15 23:50	B3DM 文件	36 KB
 Tile_+000_+003_L21_003110t3.b3dm	2021/9/15 23:50	B3DM 文件	41 KB

3dtiles 切片示例

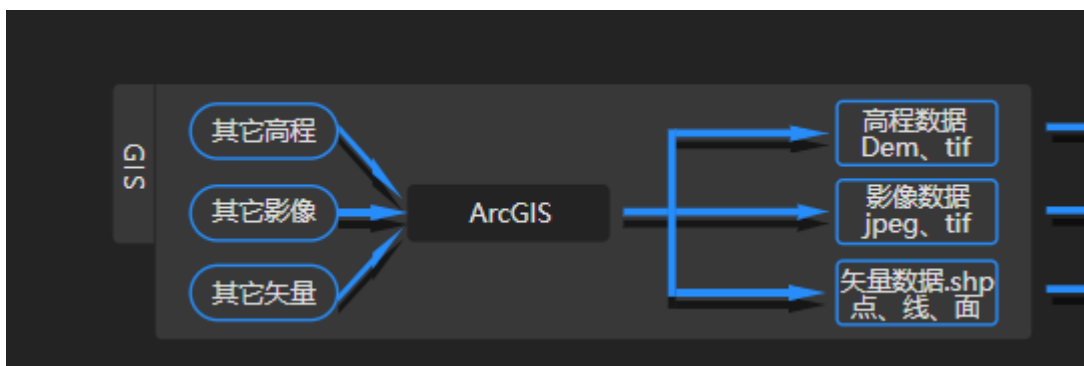
此文档编写过程中，Cesium 刚刚发布了 3dtiles next，关于 3dtiles next 我们也会持续跟进。

2.1.1 输入数据

说清楚了输出格式，我们再来讨论输入数据。用一个流行的词语来描述输入：多源异构，多种来源，多种差异化数据结构，所以实际输入是比较复杂的，比如有常规 GIS 的栅格矢量，有现在流行的实景倾斜模型，有传统的手工模型，也有当下火热的 BIM 或者 CIM 模型，还有一些特殊，比如点云数据、地质数据、体数据、场数据、流数据等等。CesiumLab 作为一个基础软件，目前无法涵盖所有数据类型，它定位解决各个行业或者各种三维项目都会用到

的数据类型的切片化。所以我们把基础数据输入分为如下五大类：

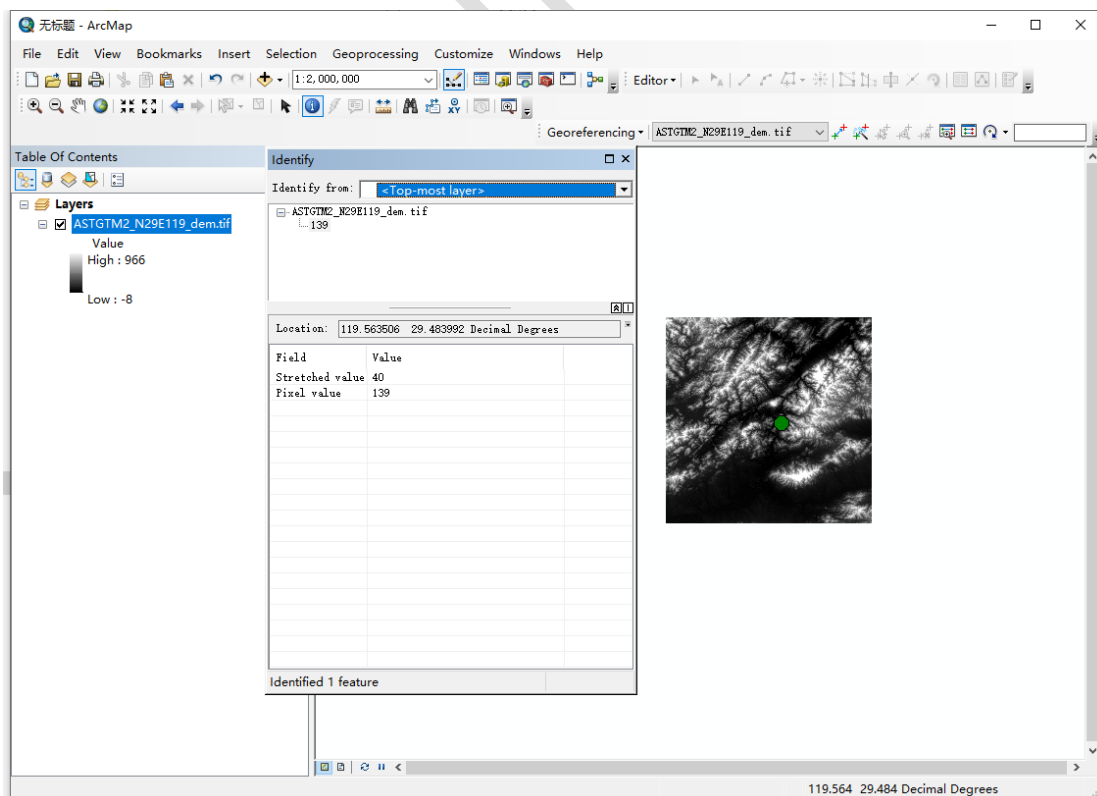
2.1.2 传统 GIS 数据



传统 GIS 数据，按照组织分为两大类：栅格和矢量，栅格数据通过逐点的属性值来描述数据，比如地形高程栅格，或者遥感影像栅格，对于栅格三维中有两种方式去可视化。

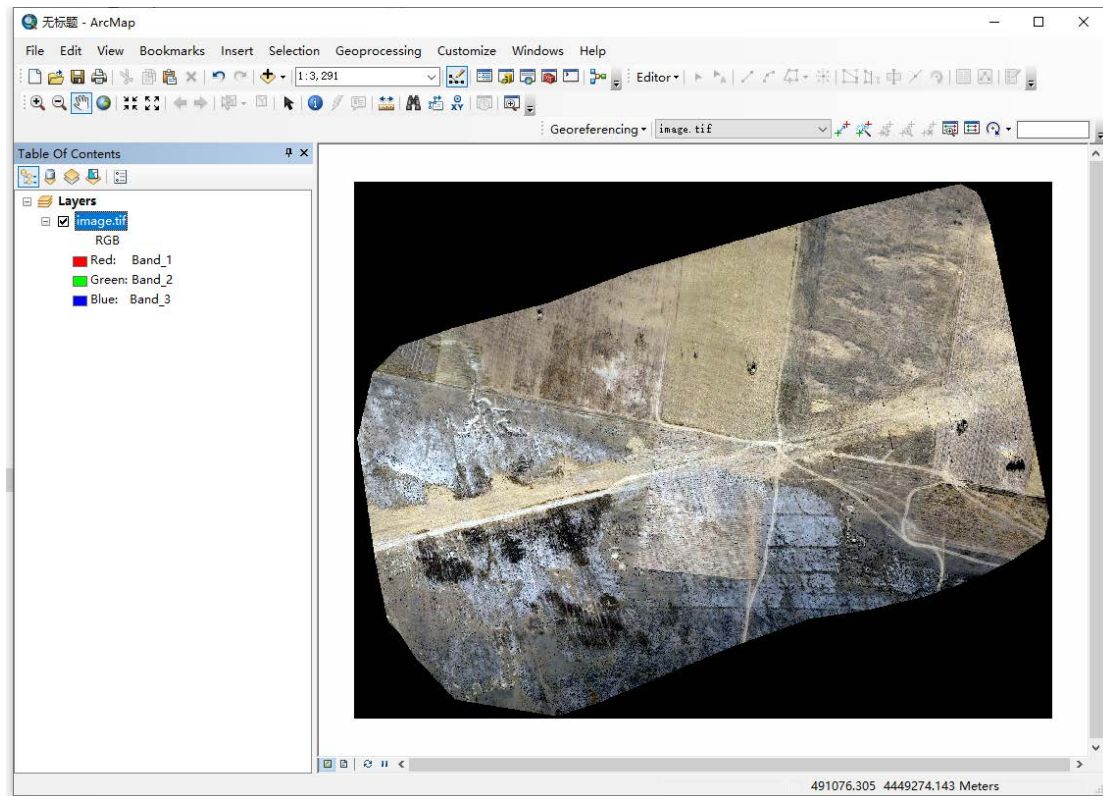
2.1.2.1 栅格数据

第一种栅格就是地形高程数据（dem），此类数据三维可视化用来显示地形起伏高低，地形数据的常见格式有 dem，tif 等。其他非常见格式的高程数据可以使用 arcgis 进行格式转换，转到 dem 或者 tif 来进行地形切片。



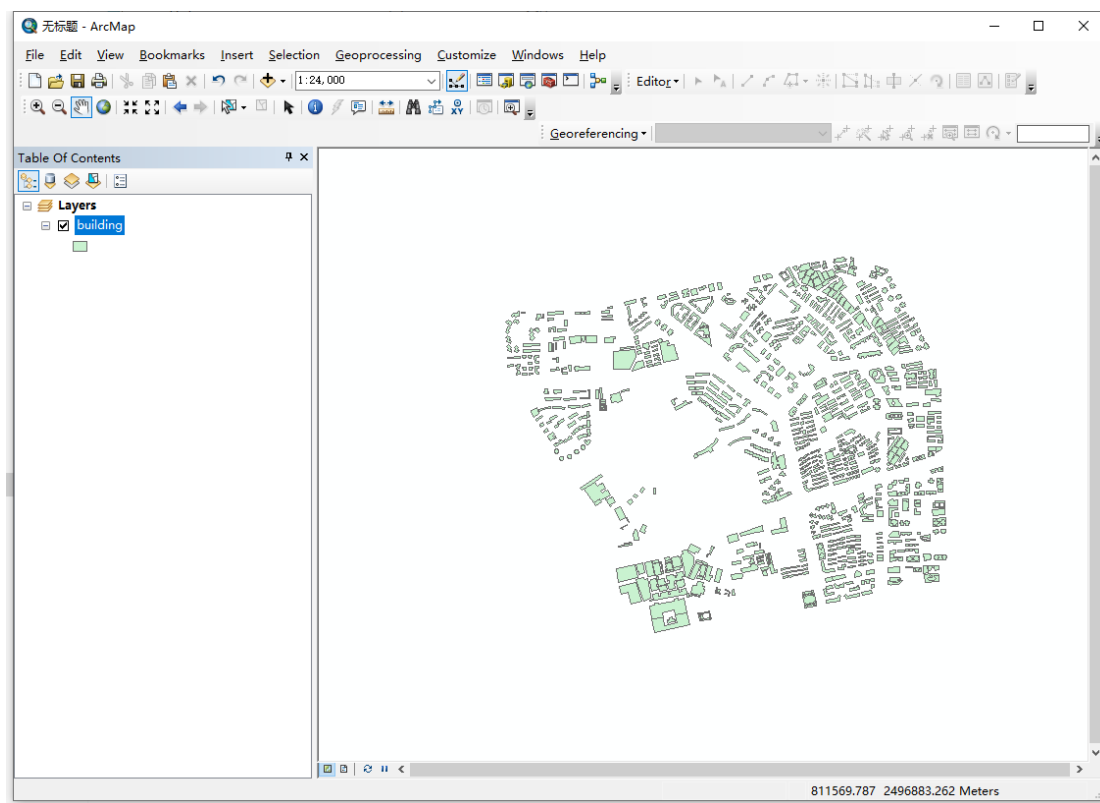
Arcmap 打开高程数据

除了高程之外，其他的栅格数据一般都是处理成影像切片来可视化。



Arcmap 打开影像数据

2.1.2.2 矢量数据



Arcmap 打开矢量数据

除了栅格之外，还有一类传统 GIS 数据，通过点、线、面等几何对象来描述地理实体，这就是矢量。矢量的三维可视化问题比较复杂，根据三维的渲染方式可以分为以下几种：

第一种是栅格化展示和绘制，借助二维 GIS 服务器的矢量渲染出图能力，矢量可以在服务端生成图片，三维上加载方式就类似栅格的影像切片，二维 GIS 相关的标准服务，比如 WMTS, WMS 等可以直接接入到三维中。这块的产品也比较多，比如 Esri 的 arcgis server, 超图的 iserver, 开源的 geoserver 等。

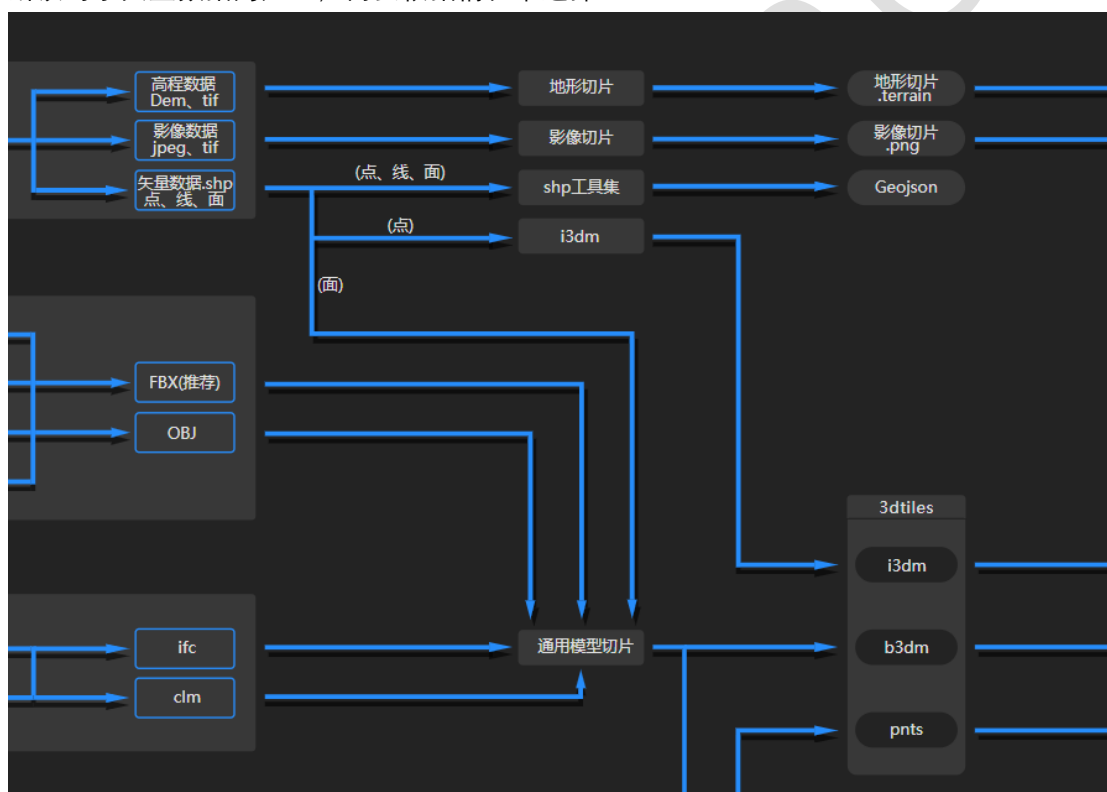
第二种是矢量化展示和绘制，比如 Cesium 里的 geojson 、czml 等绘制的图标、文字、折线、多边形等。

第三种是模型化展示和绘制，比如建筑物轮廓，我们可以根据建筑高度拉成白模去可视化，比如用来描述森林位置或者城市附属物（路灯，垃圾桶）等点位的可以用实例化模型去展示。

这三种方式各有利弊，我三年前整理的一张图，到今天依然适用：

， 矢量数据加载方案				
方案	Geojson/kml	wms	czml	3dtiles
数据量	一次性载入，数据量必须小	无上限限制	支持流式载入	无上限限制
灵活性	强：样式完全自定义	差：需要通过wms服务才可以实现样式更新	强：支持自定义样式	强：通过3dtiles style配置
显示效果	好：矢量化显示	差：图片平面显示	好：矢量化显示	好：矢量化显示
技术成熟度	好：非常多的工具可以生成	好：完全标准化，服务器也比较多	差：目前没有现成工具，需要自己组织	差：常规矢量切片的3dtiles定义还未完成。（建筑物矢量面可以处理为模型切片）
时间序列	部分支持	支持(非实时)	支持	不支持
适应性	少量临时性用户标注数据	海量地图以及专题图数据	专业科学模型数据，如卫星轨道等	海量地图以及专题图数据

所以对于矢量数据的处理， 需要根据情况来选择：

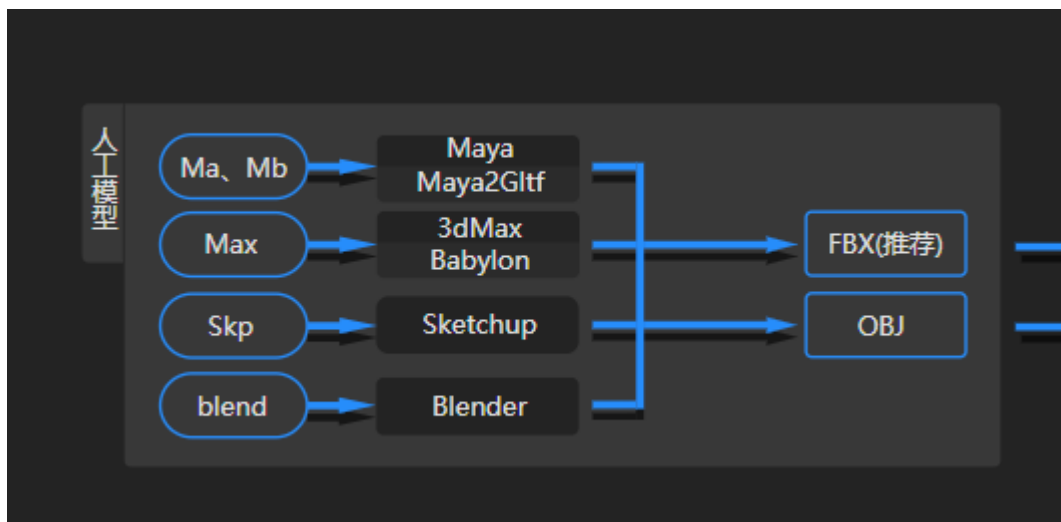


CesiumLab 矢量数据处理

一般的矢量都可以通过 lab 自带的开源 shp 工具集，把 shp 转 geojson 格式去加载。
 建筑矢量面可以通过通用模型切片处理为白模。
 描述实体点的数据，比如树木点位、灯杆位置等可以用 i3dm 工具处理为实例化模型。

2.1.3 人工模型

人工模型数据（或者手工模型数据）是三维 GIS 行业发展的最早的需求来源，通过 3dsmax, maya 等建模工具人工建模的数据。



人工模型

我们这里只罗列上面四个建模工具，其实建模工具远多于四个，手工模型格式可以多达 50 多种。这里我给大家解释几个概念。

2.1.3.1 原始格式

比如 3dsmax 的 .max 格式等，这些格式存在的意义在于是各种建模工具存储自定义信息的，一般来说格式都不开放，我们无法直接去读取和处理这些格式。所以有这些格式，一定要在建模工具内导出交换格式。

2.1.3.2 交换格式

比如比较古老的 obj 格式，以前流行的 3ds 格式，曾经的标准 collada (dae) 格式，现在事实上的交换格式标准 fbx 格式。这些格式存在的意义是为了各个软件之间能够相互导入导出模型成果。一般这些格式是开放的。所以我们模型切片的大部分输入格式，应该是交换格式。

这里我们推荐优先考虑 fbx，原因是 fbx 是 autodesk 相关产品 (3dsmax, revit, maya 等) 的官方交换格式，定义的比较完善，而且 autodesk 提供了官方的 fbxsdk，这样更稳定。

另外一个 obj，使用 obj 是退而求其次的选择，我们实践中发现一些大的模型，在 3dsmax 里导出 fbx 经常崩溃出错，但是导出 obj 反倒很稳定。虽然是 obj，但是我们也推荐使用 3dsmax 导出的 obj，obj 格式太灵活太简单，反倒对于数据标准化处理来说是缺陷，比如它没有单位信息（处理后模型大小不对），没有向上轴信息（处理后模型是垂直的），没有几何体共享（无法做实例化优化）。

如果你有其他交换格式的数据（比如 dae 或者 3ds）可以用 max 导出 fbx 再处理，如果量比较大，可以详询我们的付费服务。

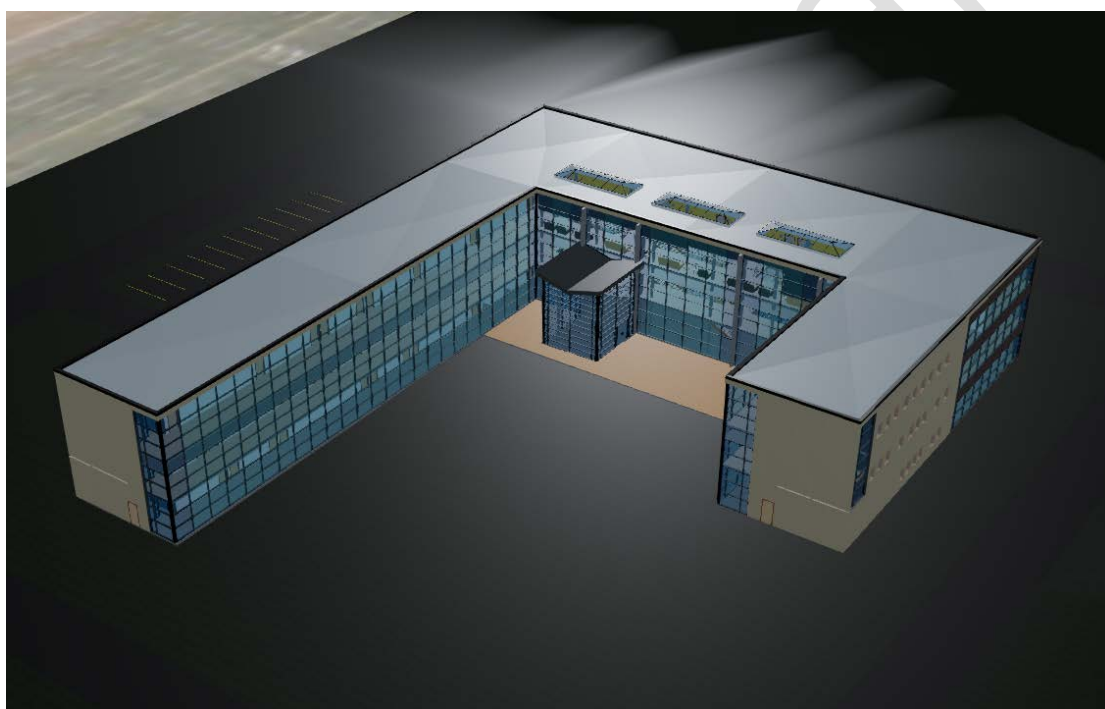
2.1.3.3 渲染格式

很多人跟我们提过需求或者问过我们，为什么不支持 gltf 进行切片。虽然对于 gltf 我们

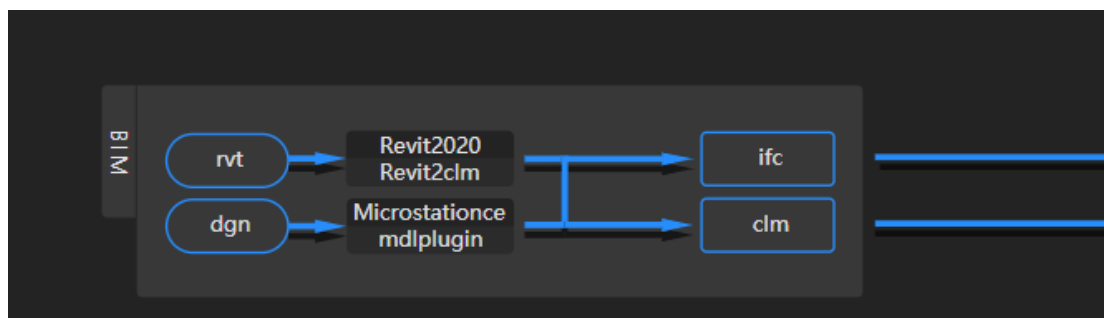
可以说是很了解的（3dtiles 的切片大部分都是构造 gltf）。但是几个原因：1，gltf 格式标准还比较新，目前已有的导出 gltf 插件或多或少有缺陷，并不推荐大家使用。2，gltf 格式定义的目的基本不是为了在各个软件之间交换三维模型，而是为了让引擎去高效渲染。所以我们这里称之为渲染格式，所以它的结构组织相对简单，目前来说也不是很适合做为切片的输入格式。

除了数据格式之外，我们做模型切片，其实应该考虑数据的“异构”，比如说建筑物或者一个机械设备，这两个都可能是 3dsmax 建模的，但是模型精度以及贴图方式都不同，那么切片方案上应该有区别。但是目前做为 Cesiumlab 软件本身来说，它是没有行业属性的，所以我们的处理需要通用一些，无论哪种“构”的模型都可以处理，所以我们做了通用模型切片工具，这个工具可以让您把各种模型导出为 3dtiles，但是可能对于特定数据来说不是最优化的方案，如果您有特殊的数据，请联系我们。

2.1.4 BIM 模型



BIM 模型



BIM 模型和手工模型最大的区别在于几点：1，建模目标不同，手工模型的目的是为了可视化，就是为了看的见，看不见的东西能省则省。BIM 完全是按照一些工程标准去创建的，

比如路面可能有多层代表了不同的物理层。手工模型的纹理贴图量可能很大，BIM 的贴图相对简单。2，手工模型一般没有属性信息，但是 BIM 模型有大量属性。

基于这些差异，虽然有些 BIM 工具可以导出 fbx 等模型的交换格式，但是还是推荐按照我们的流程来。

第一种是导出 clm 格式([clm 格式定义](#))，clm 是我们自定义的中间格式，它内部存储了我们切片需要的所有可视化数据（几何体、材质纹理、属性信息、模型结构）等。目前我们做了 revit2020 的插件和 microstation connect edition 的插件。以前的 catia v5-2016 插件因为我们实在难以找到更新的 catia 开发资料，也没有技术支持，我们无法再更新，只能暂停，并且从 lab3 里剔除。

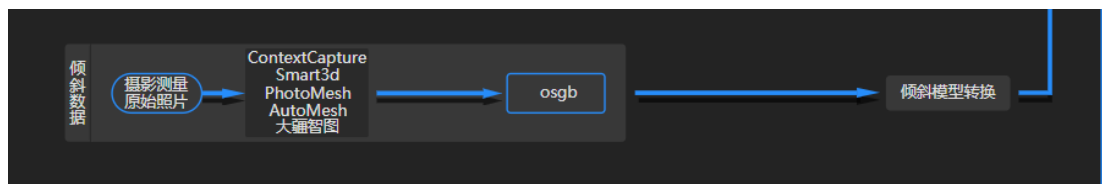
第二种格式 ifc 格式，ifc 号称是 bim 的标准格式，但事实上 ifc 的标准也是古老而复杂，各家 bim 工具对于 ifc 的支持都不是尽善尽美。另外我们读取和解析 ifc 也是一个比较头疼的问题，容易产生构件丢失的问题，一般最好不要优先考虑 ifc 的方式，而应该是 clm 方式。

对于 BIM 模型切片，目前也是通过我们的通用模型切片进行的。

2.1.5 倾斜实景数据



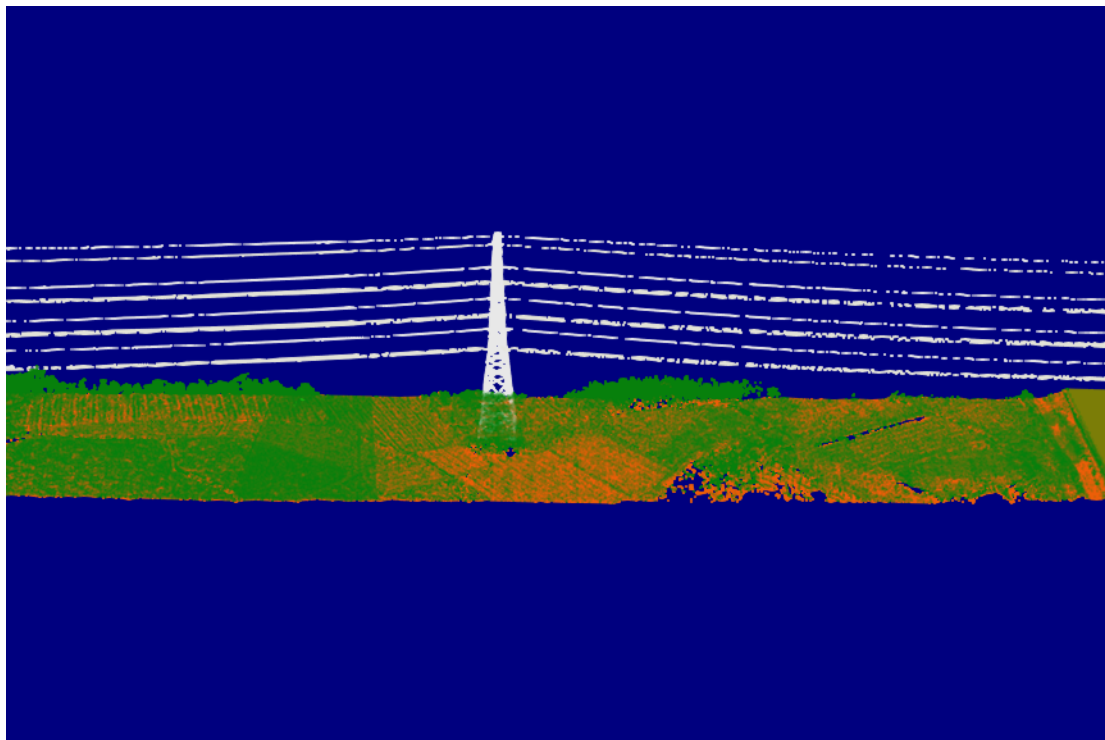
倾斜数据



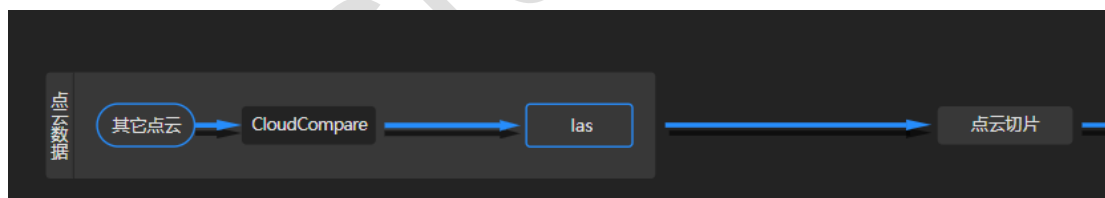
倾斜实景数据一般是来自照片进行三维重建得到的，目前能做这个格式的软件也比较多

了，例如 contextcapture 以及国内的 smart3d，大疆等。这些软件产生的结果有很多格式，由于历史原因吧，目前大部分产生的结果依然是 osgb，倾斜数据的特点是总量大，自带 LOD，这类数据我们通过倾斜模型切片工具进行处理。非 osgb 格式的数据目前不支持处理。

2.1.6 点云数据



点云数据



点云数据的格式有很多种，比如 xyz，las，csv 等，但是 las 是相对标准的一种格式。点云数据最初是来自激光扫描设备，受限于采集方式，只能用来展示一些小型数据。随着技术发展，现在倾斜摄影也可以直接获取点云，点云的数据可以呈现大范围的。就我们实践来看，点云主要还是应用在电力领域，虽然它有行业属性，但是点云数据本身是不带行业属性的，所以我们依然是放到 Cesiumlab 来完成切片过程。目前仅支持 las，其他类型的，可以尝试用 cloudcompare 这个开源软件转化下格式。

2.2 通用功能

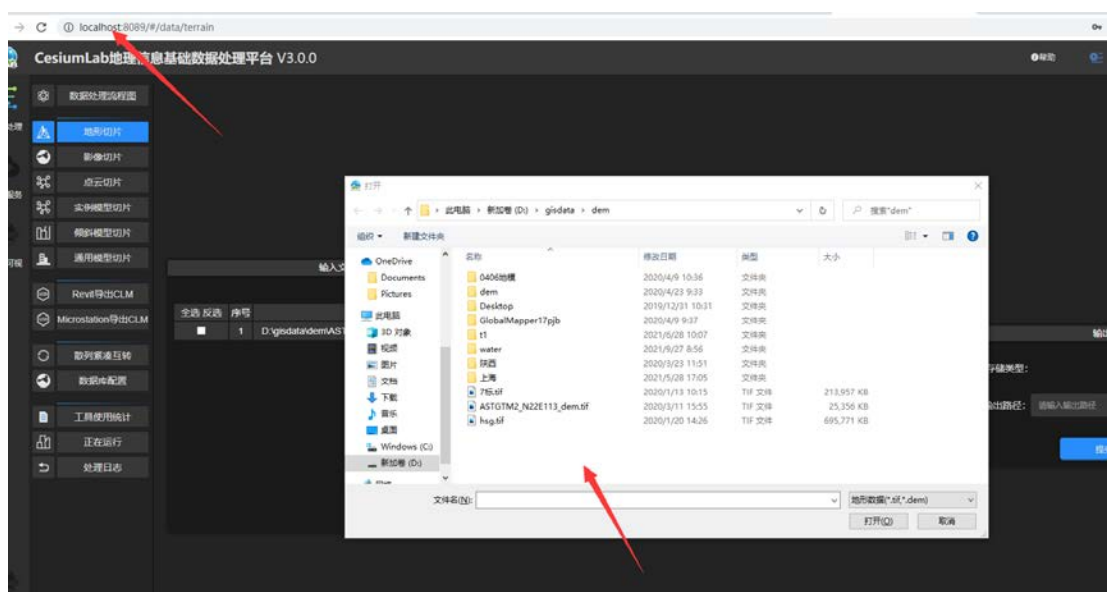
2.2.1 文件以及目录选择

CesiumLab 此版本已经完全是一个 BS 架构的产品，它的数据处理运行在 CesiumLab 产

品部署的机器上（服务端），用户可以通过局域网使用浏览器来操作它。所以注意一点，数据处理的所有路径都是服务器上的路径，而不是本机路径。当然有一个特殊情况，就是通过 localhost 访问，也就是 CesiumLab3 部署在当前用户机器上，我们称之为本机操作。

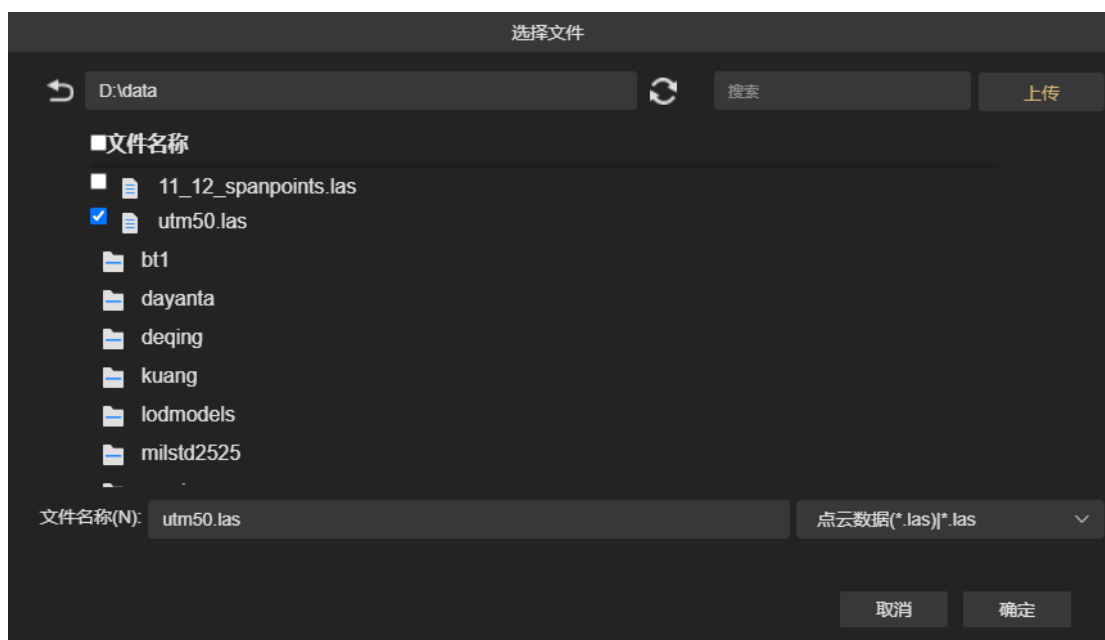
2.2.1.1 本机操作

为了便于本机操作，我们使用了一些技术手段，点击 Lab3 的页面按钮的时候，会弹出 windows 系统自带的文件和目录选择器，选择完毕后会得到实际的路径。和我们常规操作 C 端软件没有任何区别。



2.2.1.2 远程操作

对于非本机的操作，Lab3 会弹出如下面板，类似于 Windows 的文件浏览器，是对服务器上的文件进行操作，可以选择服务器上的文件，选择或创建服务器上的输出路径以及上传本地文件到服务器。



左上角返回：可以返回上一级目录；

地址栏：显示当前所在文件夹，可手动输入；

刷新：重新加载地址栏所指文件夹下的内容；

搜索：输入关键字，回车，从地址栏所指文件夹下搜索包含该关键字的文件，如果该文件夹下内容比较多，则搜索很慢，请慎用；

上传：可以上传本地文件（目前只支持 zip 文件）到服务器，自动解压并定位到指定文件夹下（默认为 C:\Users\用户\cesiumlab3\upload）；

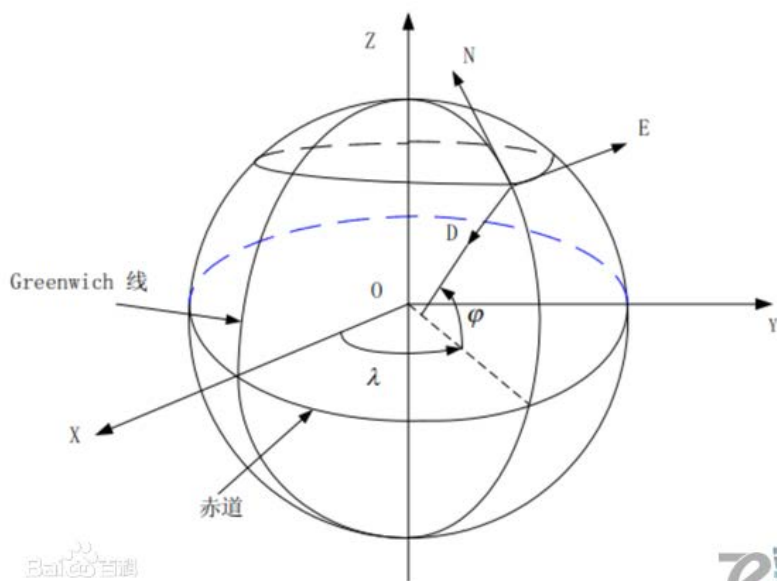
文件名称列表：双击文件夹进入该文件夹，点选文件表示选中该文件，若文件名前存在复选框，则表示可多选；

文件名称：选中文件时，自动填写该文件文件名，当需要保存文件时也可以输入文件名。

2.2.2 空间参考

2.2.2.1 输出数据的空间参考

和格式部分类似，我们先搞清楚我们主要切片格式（3dtiles）的空间参考问题，3dtiles 是 Cesium 定义的三维模型渲染格式，但是 Cesium 是一个地球开发包，想当然的很多非业内人士说，地球坐标不就是经纬度吗，那么 3dtiles 就是经纬度了。但是事实上 3dtiles 坐标不能直接存储为经纬度方式。原因是现代图形学都是构建在直角坐标系上的，经纬度本质是球坐标，不适合做渲染。Cesium 引擎包括所有的地球开发包，对外可能提供了经纬度二次开发接口，但是其内部的渲染依然是直角坐标系，Cesium 采用了 WGS84 的地心坐标系作为世界坐标。地球质心为 0,0,0 点，地心和北极连线为正 z 轴，地心和 本初子午线（0°经线）与赤道交点连线为正 x 轴。正交得到 y 轴。



Wgs84 世界坐标

考虑到地球半径是 6378137 米, 那么这个坐标系比较大, 涉及另一个图形学古老问题, 渲染都是单浮点精度表示的。为了避免渲染精度的问题, 一般 3dtiles 里的坐标都是经过 tileset.json 里的 transform 变换矩阵 变换的相对坐标, 有如下等式。

$$\text{3dtiles 里的坐标} * \text{3dtiles 里的 transform 矩阵} = \text{wgs84 世界坐标}$$

2.2.2.2 输入数据的空间参考

空间参考 (spatial reference) 是一个 GIS 的概念, 具体概念我就不照本宣科了。说下我的理解, 比如我们读取了一个文件, 里面有个坐标是 x,y,z 分别是 1,1,1, 那么请问你知道这个坐标在地球上的位置在哪里吗? 显然答案是否定的, 我们不知道这个(1,1,1)该放到地球的何处。如果有人告诉你, 你的数据是 EPSG:4549 下的 1,1,1 那么这个位置信息就完备了。

EPSG:4549 是 ogc 的标准编码, 它的定义是 CGCS2000 / 3-degree Gauss-Kruger CM 120E, 椭球定义是 CGCS2000, 采用了中央经线东经 120°, 采用了 3 度带高斯克吕格投影方式的坐标系。有了这些信息, 我们就可以把 1, 1, 1 换成称 WGS84 下的世界坐标。所以空间参考是一个数据的元属性, 很多用户总是问我们说该设置什么, 这个有时候难以回复。

目前我们支持几种形式的空间参考设置:

第一种就是上述提到的 EPSG:{code} 这种形式的, OGC 对所有已知的标准的空间参考都编上了唯一码, 这样方便使用。推荐一个网站 <http://epsg.io/>, 可以去上面搜索, 有非常多。最常见几种编码, 大家可以了解:

EPSG:4326 这就是 WGS84 椭球定义的官方编码

EPSG:3857 这是流行的电子地图 web 墨卡托, 比如谷歌地图, 腾讯地图、高德地图等都是这种方式。

EPSG:4490 这是我国的椭球定义标准 CGCS2000

EPSG:4546 等紧挨着的一系列, 都是基于 EPSG:4490 椭球定义, 采用 3 度带分割高斯克吕格投影的投影坐标系, 也是目前国内数据常用的相关坐标系。

第二种是采用 wtk 方式描述的空间参考，在工程领域，由于一些精度需要，会自定义中央经线，这样就没有对应的 EPSG 编码，那么这种方式下，会使用如下类似的格式来编写：GEOGCS["WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.017453292519943295]]

这两种都是精确的地理空间投影。

第三种是站心坐标系 ENU (east north up)

实践中，经常有一些模型本来也就没有考虑和地球位置匹配的问题，这样的数据我们可以采用一种站心坐标系的方式来处理。ENU 是以一个地球表面的位置点做为 0,0,0 点，以垂直地表向上为 z 轴，以正东向为 x 轴，正北向为 y 轴的一个局部空间来定义位置。

lab 上是 ENU:{维度},{经度} 的方式来设置

除了空间参考之外，lab 界面上还有一个零点坐标的概念。零点坐标的出现还是数据精度的问题。假设你要建城市的三维模型，采用流行的 2000 某个投影坐标系，你会发现你的模型坐标非常大：

EPSG:4549

CGCS2000 / 3-degree Gauss-Kruger CM 120E

Transform

Get position on a map

Attributes

Unit: metre

Geodetic CRS: China Geodetic Coordinate System 2000

Datum: China 2000

Ellipsoid: CGCS2000

Prime meridian: Greenwich

Data source: OGP

Information source: OGP

Revision date: 2009-11-04

Scope: Large scale topographic mapping, cadastral and engineering survey.

Remarks: Truncated form of CGCS2000 / 3-degree Gauss-Kruger zone 40 (code 4528).

Area of use: China - onshore between 118°30'E and 121°30'E.

Coordinate system: Cartesian 2D CS. Axes: northing, easting (X,Y). Orientations: north, east. UoM: m.

Covered area powered by MapTiler



Center coordinates

500000.00 4305182.26

Projected bounds:

347872.25 2703739.74

599933.05 5912395.20

WGS84 bounds:

118.5 24.43

121.5 53.33

China - onshore between 118°30'E and 121°30'E.

4549 的坐标范围

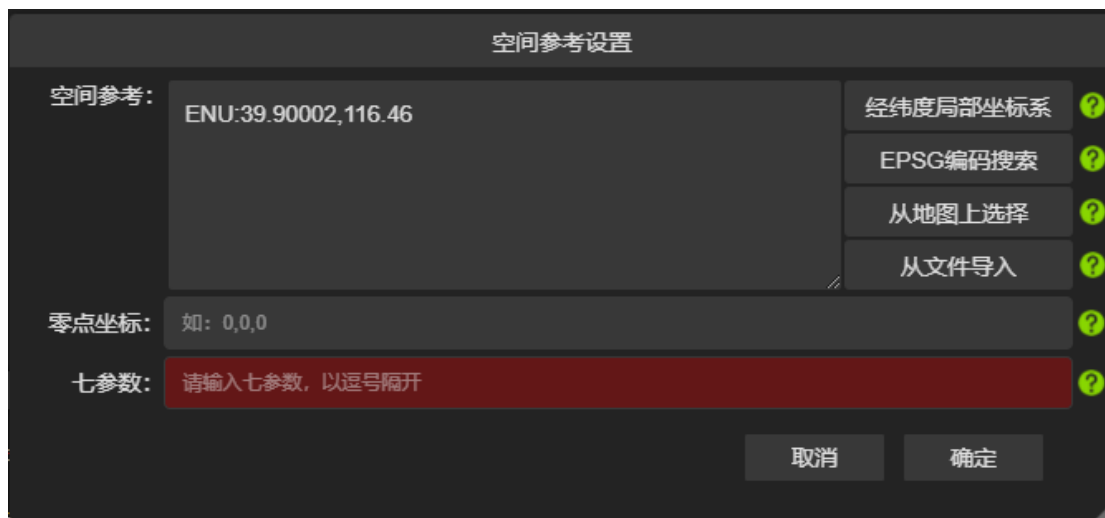
这个坐标范围导致在 max 里几乎难以操作，实践中，经常假定我所有数据都有一个偏移位置，建模的坐标 + 这个偏移位置 = EPSG:4549 下投影坐标，这样就可以解决建模工具的精度问题。那么这个偏移位置就应该设置在 lab 的零点坐标上。

Lab 的处理逻辑也简单：

做空间变换（模型坐标+零点坐标）= WGS84 世界坐标

这样你的数据才能精确放置到地球上。

2.2.2.3 空间参考界面操作



空间参考设置界面

空间参考包括三个参数，空间参考，零点坐标以及七参数。

其中空间参考的选择支持 5 种方式：

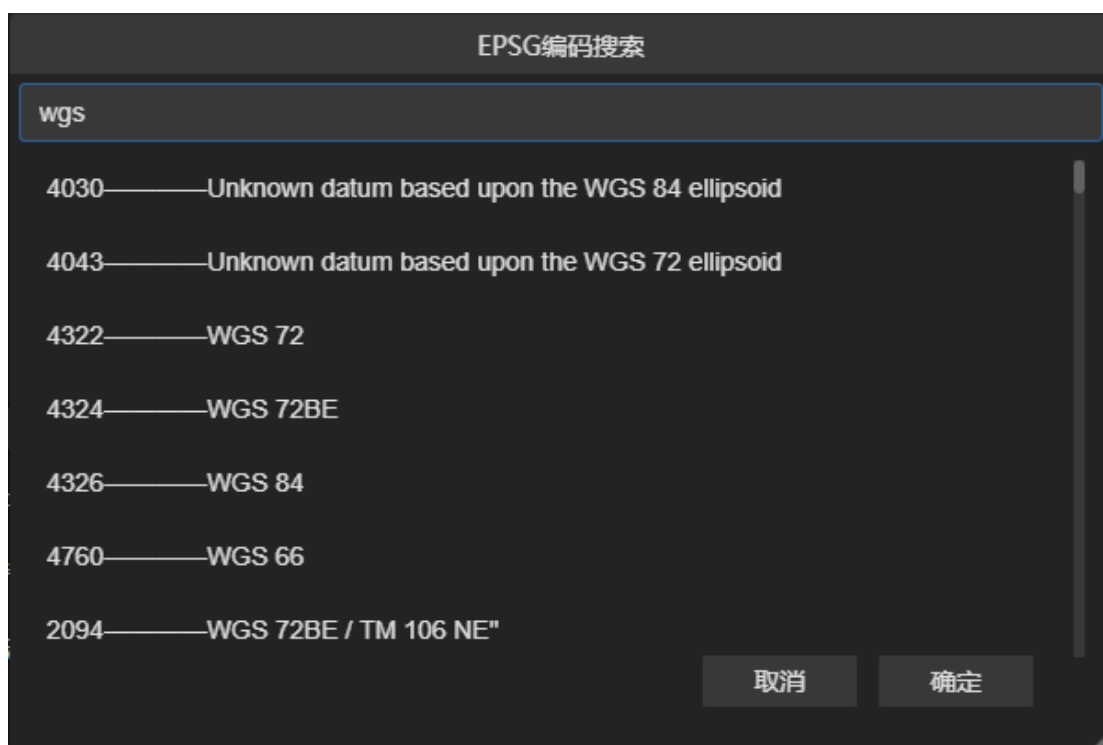
手动填写：空间参考支持多种格式，最常用的比如：ENU 字符串（例 ENU:39.90002,116.46，纬度在前，经度在后，中间用逗号隔开，默认经纬度是天安门所在位置），EPSG 字符串（例 EPSG:32650），proj4 和 [wkt](#) 等；

经纬度局部坐标系：填写纬度，经度，点击确定，自动构造 ENU 字符串；



经纬度局部坐标系设置界面

EPSG 编码搜索：通过输入关键字搜索 EPSG 编码，点击确定，自动构造 EPSG 字符串；



EPSG 编码搜索界面

从地图选择：可以在右上角输入框输入经纬度，也可以在地图上点击拾取经纬度，自动计算并列出该经纬度位置相关的 ENU 和 EPSG 空间参考，然后选择需要的点确定；



从地图选择空间参考

从文件导入：支持从 prj(shp 数据的空间参考文件)文件中导入空间参考。

2.2.3 数据存储

2.2.3.1 散列存储

其实就是每个切片独立存一个硬盘文件，这种方式适合大部分小型项目，因为散列可以直接使用现成的静态服务器（tomcat, nginx, iis 等）分发，所以使用相对方便。但是散列也有一个缺陷，当数据量大了之后，碎文件过多，硬盘存取效率降低，不仅是数据处理过程缓慢，最终分发读取效率也不高，更致命的是，基本上没办法做数据迁移。

当输出部分选择是散列的时候，我们点击【选择】需要选择一个目录。



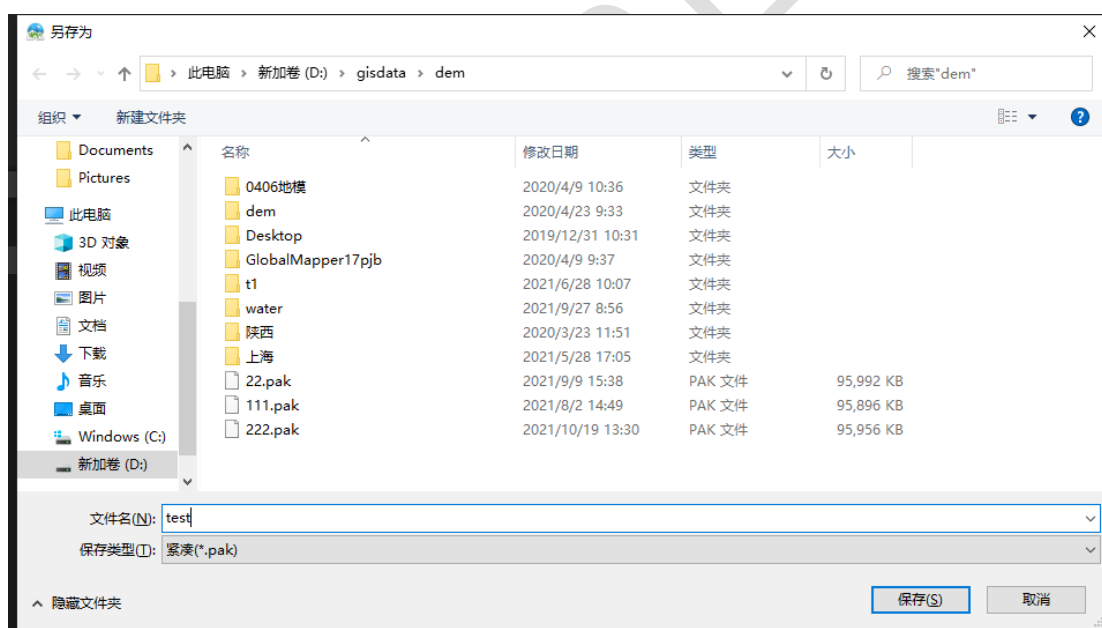
The image shows a dark-themed dialog box titled "输出文件" (Output File). Inside, there are two main sections. The first section is labeled "存储类型:" (Storage Type:) and has a dropdown menu currently showing "散列" (Hash). The second section is labeled "输出路径:" (Output Path:) and contains a text input field with the placeholder "请输入输出路径" (Please enter output path). To the right of this input field is a blue button labeled "选择" (Select). A blue arrow points to this "选择" button. Below these sections is a large blue button labeled "提交处理" (Submit Processing).

【输入路径】也可以手动输入一个路径，如果这个路径不存在，那么我们切片程序会尝试去创建这个路径，如果这个路径没有权限或者无法创建成功，那么切片程序会处理失败。

2.2.3.2 PAK 地形影像紧凑存储



输出部分，存储类型选择【紧凑(pak)】，点击【选择】，在文件选择器中，输入一个文件名，点【保存】具体参见本文文件及目录选择操作。



PAK 是我们自定义的一种用来存储常规瓦片的文件格式，它本身是一个 sqlite 文件，使用 sqlite expert 软件可以直接打开看到它的结构。其内部有两种表

infos 表:

记录这个数据图层的一些基础信息，只有一个。

minx,miny,maxx,maxy 存储这个数据的经纬度范围最小经度，最小纬度，最大经度，最大纬度。

minlevel 最小级别

maxlevel 最大级别

source: 保留字段

type: terrain 或者 image 分别表示地形或者影像切片

tiletrans: 保留字段

zip: 0 表示数据未做 zip 压缩, 1 表示 zip 压缩

cur_level,cur_x,cur_y: 保留字段

layerjson: 存储的 terrain 图层的 layer.json 信息, 影像图层请忽略

contenttype: quantizedmesh 表示地形切片, image/png 表示影像切片

RecNo	Column Name	SQL Type	Size	Precision	PK	Default Value	Not Null	Not Null Conflict Clause	Unique	Unique Conflict Clause	Collate	Generated Always As	Storage	FK
1	minx	double			☐		☐		☐					
2	miny	double			☐		☐		☐					
3	maxx	double			☐		☐		☐					
4	maxy	double			☐		☐		☐					
5	minlevel	int			☐		☐		☐					
6	maxlevel	int			☐		☐		☐					
7	source	VARCHAR	255		☐		☐		☐					
8	type	VARCHAR	20		☐		☐		☐					
9	tiletrans	VARCHAR	20		☐		☐		☐					
10	zip	int			☐		☐		☐					
11	cur_level	int			☐		☐		☐					
12	cur_x	int			☐		☐		☐					
13	cur_y	int			☐		☐		☐					
14	layerjson	blob			☐		☐		☐					
15	contenttype	VARCHAR	20		☒		☒		☒					

infos 表结构

rowid	minx	miny	maxx	maxy	minlevel	maxlevel	source	type	tiletrans	zip	cur_level	cur_x	cur_y	layerjson	contenttype
1	118.999700546265	38.999662399292	120.000143051147	40.0001049041748	0	14	(null)	terrain	tms	0	(null)	(null)	(null)	(blob)	quantizedmesh

infos 表示例

block 系列表:

用来存储切片数据, 可能不止一个。

RecNo	Column Name	SQL Type	Size	Precision	PK	Default Value	Not Null	Not Null Conflict Clause	Unique	Unique Conflict Clause	Collate	Generated Always As	Storage	FK
1	z	int			☐		☐		☐					
2	x	long			☐		☐		☐					
3	y	long			☐		☐		☐					
4	tile	blob			☐		☐		☐					

block 表结构

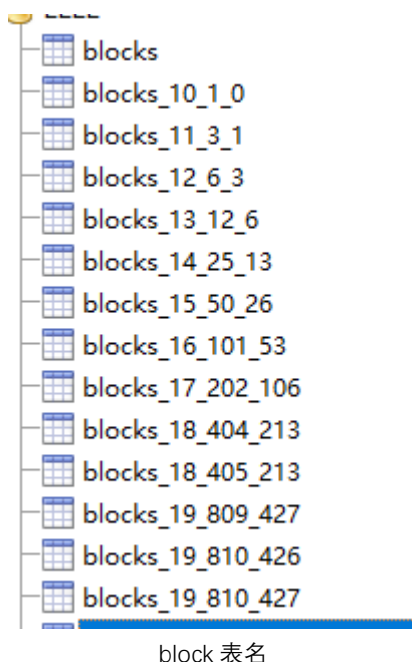
rowid	z	x	y	tile
2	10 809	427	(blob)	
3	10 810	427	(blob)	
7	10 810	426	(blob)	

block 表示例

z, x, y 表示切片的索引。

tile 字段存储了切片内容, 对于影像切片, 就是 png 图片内容, 对于地形切片是 terrain 格式。如果 infos.zip = true 那么这里存储是做了 gzip 压缩后的数据。

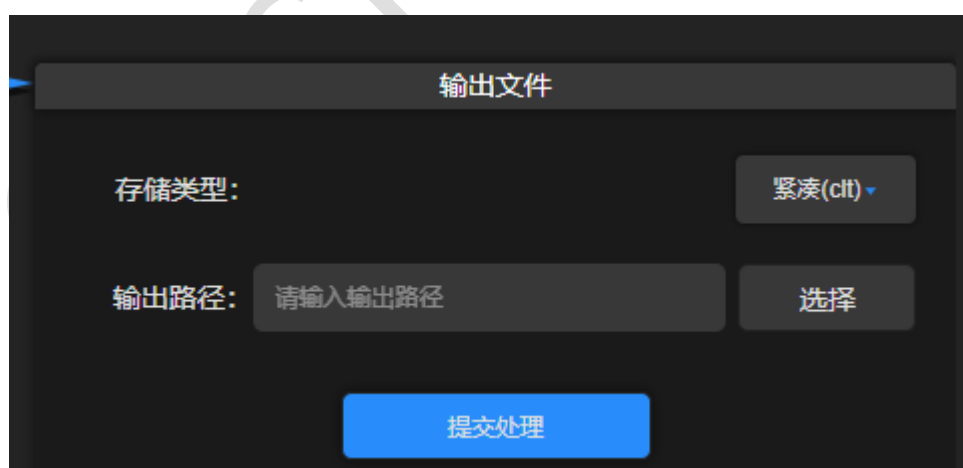
blocks 可能会有多个表, 主要原因是避免一个表里的块个数太多带来的请求性能降低



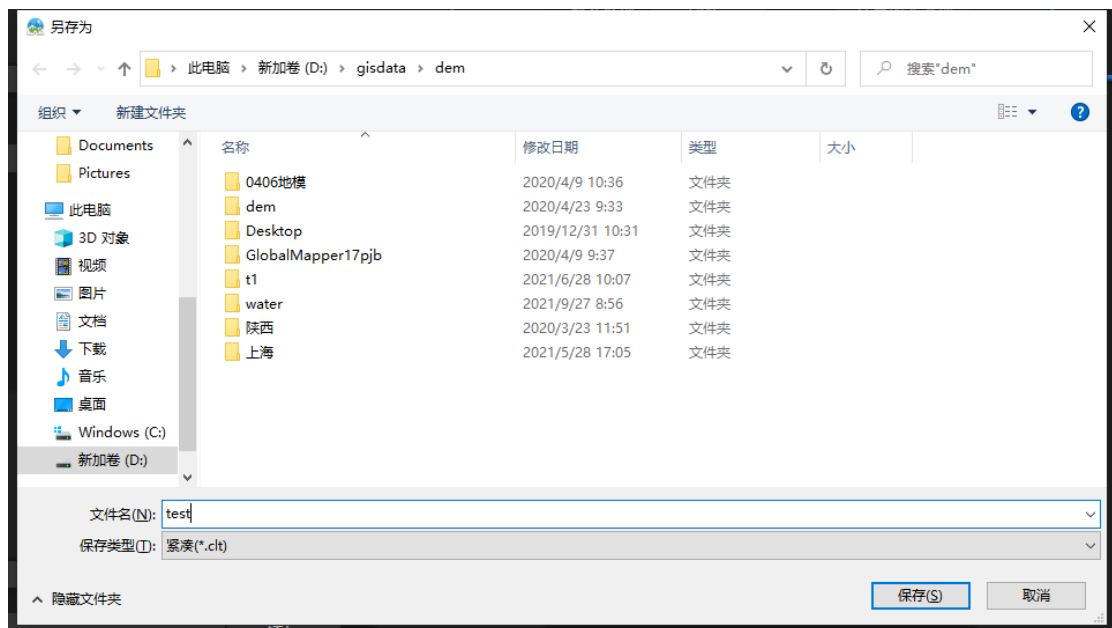
根据一个块的 x, y, z 索引可以计算得到它实际存储的 block 表名
如果 $z < 10$, 那么表名是 “blocks”。
否则 表名 = “blocks_” + tostring(z) + “_” + tostring(x / 512) + “_” + tostring(y / 512)

2.2.3.3 CLT 瓦片紧凑存储

除地形切片和影像切片工具之外, 其他工具切片输出的紧凑都是 clt 格式的紧凑。这种方式不直观, 而且需要动态服务支持, 但是它的优势是对于大量散列切片能够加速存储和迁移。



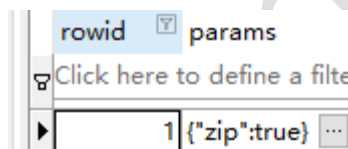
输出部分, 存储类型选择【紧凑(clt)】, 点击【选择】, 在文件选择器中, 输入一个文件名, 点【确定】, 具体参见本文文件及目录选择操作。



clt 文件里有两个表：

infos 表：

存储了数据元信息，目前仅一个字段 params



params 是个 text 类型的字段，实际是 json 结构，里面的 zip 属性如果为 true，表示切片数据经过 gzip 压缩。

tiles 表：

RecNo	Column Name	SQL Type	Size	Precision	PK	Default Value	Not Null	Not Null Conflict Clause	Unique	Unique Conflict Clause	Collate	Generated Always As	Storage	FK
1	md5	VARCHAR	16		☐		☐		☐					
2	path	text			☐		☐		☐					
3	tile	blob			☐		☐		☐					
4	type	VARCHAR	20		☐		☐		☐					

tiles 表结构

rowid	md5	path	tile	type
Click here to define a filter				
1	25533111c970f9abf5cda67f2f19d3dd	1615111115753.json	(blob)	file
2	684ed9287c806e2410e1620cb266fed8	1615111115753.b3dm	(blob)	file
3	5412d635d2d5984fc131c2c6e7f6e950	16151111157533.json	(blob)	file
4	698a6177b0e85f35942fa9ba77e1c0fd	16151111157533.b3dm	(blob)	file
5	ba3718ddd5d42970676ebd3d5ae995ae	161511111575333.json	(blob)	file
6	8b2476f31fc66f567eca08d5555ecf5c	161511111575333.b3dm	(blob)	file
7	15b3103a8217e25de78376c4602e6ba4	1615111115753337.json	(blob)	file
8	b9da2b912b13ba73504bfb9ce3860c36	1615111115753337.b3dm	(blob)	file
9	2acbd508e09e2d3461735c4ecb817684	16151111157533371.json	(blob)	file
10	b99397584fe7c10523d2577a0c24bd92	16151111157533371.b3dm	(blob)	file
11	ab30d63039631f4f868d8ef4e3b970de	161511111575333713.json	(blob)	file
12	8b4ef08de5f68ab0e92bf89935cbc51c	161511111575333713.b3dm	(blob)	file
13	e5d95700d38dc976e0454341273f38f6	1615111115753337131.json	(blob)	file
14	30859b5c35af3de5ba4ce8032cfada4e	1615111115753337131.b3dm	(blob)	file
15	b3e4463e236415a5fbd2b0d5d9ea457a	16151111157533371313.json	(blob)	file
16	f7c59eb09b26a3e5ba841453417a5f72	16151111157533371313.b3dm	(blob)	file

tiles 表示例

md5字段：对 path 字段的 md5 加密

path字段：当前切片文件路径（相对于 tileset.json）

tile 字段：瓦片内容，默认进行了 gzip 压缩

2.2.3.4 Mongodb 瓦片存储

除地形切片和影像切片工具之外，其他工具可将切片输出到 mongodb 数据库中：

存储类型

存储类型：

mongo

选择：

名称：

请输入名称

主机：

请输入主机IP

端口：

请输入端口号

数据库：

请输入数据库名

用户名：

请输入用户名

密码：

请输入密码

测试连接

保存

提交处理

Mongo 存储类型设置

输出部分，存储类型选择【mongo】，点击【选择】，从下拉列表中选择已配置好的数据库名称（参见 [1.12 数据库配置](#)），或者填写数据库连接参数，点【提交处理】即可。这里也可以通过【测试连接】验证参数正确性，也可点击【保存】保存到数据配置中，以便下次使用。

3dtiles 切片存储在所选数据库的名为 tiles 的 GridFS 中。

名	ID	长度	数据块大小	上传日期	MD5
t3jvb9o6t/tiles.json	611f64abb95000003d006142	661 bytes	255 KB	2021-08-20 08:15:39	2411c8db4c5cecbcb8787756372d1b144
t3jvb9o6t/15_53589_20513_16384.pnts	611f64b6b95000003d006143	29.73 KB	255 KB	2021-08-20 08:15:50	81a98c942ccc58ddcd99454539389e7
t3jvb9o6t/15_53589_20513_16384.json	611f64b6b95000003d006144	509 bytes	255 KB	2021-08-20 08:15:50	eb1f0c3a0e98e5be9e0333e6ec9744f7
t3jvb9o6t/15_53589_20514_16384.pnts	611f64b6b95000003d006145	106.17 KB	255 KB	2021-08-20 08:15:50	54d1f5d214d82542163818d6cf90f02a
t3jvb9o6t/15_53589_20514_16384.json	611f64b6b95000003d006146	512 bytes	255 KB	2021-08-20 08:15:50	bbabe5cc5002b140e41bc1d00c9c498c
t3jvb9o6t/15_53590_20514_16384.pnts	611f64b6b95000003d006147	406.46 KB	255 KB	2021-08-20 08:15:50	3fefa8b3177da5d3629f96ffc4d44cb3
t3jvb9o6t/15_53590_20514_16384.json	611f64b6b95000003d006148	526 bytes	255 KB	2021-08-20 08:15:50	1a016e0ed428df1c0f93c44b8ee0c87b
t3jvb9o6t/15_53590_20513_16384.pnts	611f64b6b95000003d006149	1.31 MB	255 KB	2021-08-20 08:15:50	3d9c75022945e10919c989e42eefa006
t3jvb9o6t/15_53590_20513_16384.json	611f64b6b95000003d00614a	589 bytes	255 KB	2021-08-20 08:15:50	b78c02c593a42cb881e7128adff79c2
t3jvb9o6t/16_107180_41026_32768.pnts	611f64b6b95000003d00614b	510.84 KB	255 KB	2021-08-20 08:15:50	27cb1729731339f73f0c7f1722810dd5
t3jvb9o6t/16_107180_41026_32768.json	611f64b6b95000003d00614c	518 bytes	255 KB	2021-08-20 08:15:50	ff50de798af5989337a637268a466345
t3jvb9o6t/16_107181_41026_32768.pnts	611f64b6b95000003d00614d	602.48 KB	255 KB	2021-08-20 08:15:50	3586d5d2043decf82c37da0861fca805
t3jvb9o6t/16_107181_41026_32768.json	611f64b6b95000003d00614e	617 bytes	255 KB	2021-08-20 08:15:50	1fa27e6032d21d323df9e52b42ccb608
t3jvb9o6t/16_107179_41027_32768.pnts	611f64b6b95000003d00614f	109.88 KB	255 KB	2021-08-20 08:15:50	ee681b31b9e23f0c237e6b9844e50a8a
t3jvb9o6t/16_107179_41027_32768.json	611f64b6b95000003d006150	505 bytes	255 KB	2021-08-20 08:15:50	76fd40a02328021eb5fa61d386dbc1b0
t3jvb9o6t/16_107180_41027_32768.pnts	611f64b6b95000003d006151	2.14 MB	255 KB	2021-08-20 08:15:50	3f992ec6e5ec7608cb7d9d92ad8700e
t3jvb9o6t/16_107180_41027_32768.json	611f64b6b95000003d006152	619 bytes	255 KB	2021-08-20 08:15:50	059d1e799c423ee8463ef8d9c56006ef
t3jvb9o6t/16_107179_41028_32768.pnts	611f64b7b95000003d006153	381.01 KB	255 KB	2021-08-20 08:15:51	917502fb8d740c4212f230b80c748ede
t3jvb9o6t/16_107179_41028_32768.json	611f64b7b95000003d006154	501 bytes	255 KB	2021-08-20 08:15:51	c89701f3e727e45aee39040b915fd19b
t3jvb9o6t/16_107180_41028_32768.pnts	611f64b7b95000003d006155	1.39 MB	255 KB	2021-08-20 08:15:51	0c533ea4e582925d2c9ba4fa6f634624

tiles 示例

名：当前切片，服务 id + 文件路径（相对于 tileset.json），服务 id 即分发服务中的服务 id

md5：对“名”字段的 md5 加密

data：切片数据，默认进行了 gzip 压缩

2.2.3.5 Postgresql 瓦片存储

除地形切片和影像切片工具之外，其他工具可将切片输出到 Postgresql 数据库中：

输出文件

存储类型：

postgres

选择：

postgres

名称：

postgres

主机：

localhost

端口：

5432

数据库：

请输入数据库名

用户名：

postgres

密码：

测试连接

保存

提交处理

postgres 存储类型设置

输出部分，存储类型选择【postgres】，点击【选择】，从下拉列表中选择已配置好的数据库名称（参见 [1.12 数据库配置](#)），或者填写数据库连接参数，点【提交处理】即可。这里也可以通过【测试连接】验证参数正确性，也可点击【保存】保存到数据配置中，以便下次使用。

3dtiles 切片存储在所选数据库的 public 模式下，涉及两个表，infos 和 tiles_[服务 id]：

infos 表存储了数据元信息，目前有两个字段 name 和 params，name 存储服务 id，即分发服务中的服务 id，params 是个 text 类型的字段，实际是 json 结构，里面的 zip 属性如果为 true，表示切片数据经过 gzip 压缩。对于 postgres 存储类型，默认 zip 压缩。

name	params
tzUWVf0m	{"zip":true}
tXLVUO4m9	{"zip":true}
tZQaem1qT	{"zip":true}
trtDq06yU	{"zip":true}

infos 示例

tiles 表：表名为 tiles_[服务 id]

RecNo	Column Name	SQL Type	Size	Precision	PK	Default Value	Not Null	Not Null Conflict Clause	Unique	Unique Conflict Clause
1	md5	VARCHAR	16		☐		☐		☐	
2	path	text			☐		☐		☐	
3	tile	blob			☐		☐		☐	
4	type	VARCHAR	20		☐		☐		☐	

tiles 表结构

rowid	md5	path	tile	type
Click here to define a filter				
1	25533111c970f9abf5cda67f2f19d3dd	1615111115753.json	(blob)	file
2	684ed9287c806e2410e1620cb266fed8	1615111115753.b3dm	(blob)	file
3	5412d635d2d5984fc131c2c6e7f6e950	16151111157533.json	(blob)	file
4	698a6177b0e85f35942fa9ba77e1c0fd	16151111157533.b3dm	(blob)	file
5	ba3718ddd5d42970676ebd3dfe995ae	161511111575333.json	(blob)	file
6	8b2476f31fc66f567eca08d5555ecf5c	161511111575333.b3dm	(blob)	file
7	15b3103a8217e25de78376c4602e6ba4	1615111115753337.json	(blob)	file
8	b9da2b912b13ba73504bfb9ce3860c36	1615111115753337.b3dm	(blob)	file
9	2acbd508e09e2d3461735c4ecb817684	16151111157533371.json	(blob)	file
10	b99397584fe7c10523d2577a0c24bd92	16151111157533371.b3dm	(blob)	file
11	ab30d63039631f4f868d8ef4e3b970de	161511111575333713.json	(blob)	file
12	8b4ef08de5f68ab0e92bf89935cbc51c	161511111575333713.b3dm	(blob)	file
13	e5d95700d38dc976e0454341273f38f6	1615111115753337131.json	(blob)	file
14	30859b5c35af3de5ba4ce8032cfada4e	1615111115753337131.b3dm	(blob)	file
15	b3e4463e236415a5fbd2b0d5d9ea457a	16151111157533371313.json	(blob)	file
16	f7c59eb09b26a3e5ba841453417a5f72	16151111157533371313.b3dm	(blob)	file

tiles 表示例

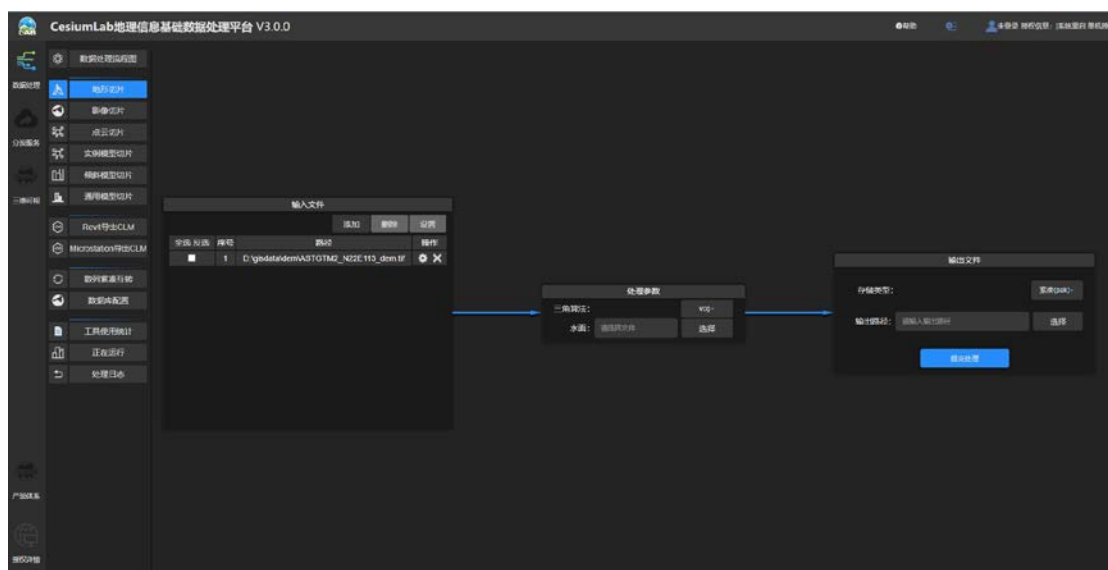
md5字段：对 path 字段的 md5 加密

path字段：当前切片文件路径（相对于 tileset.json）

tile 字段：瓦片内容，默认进行了 gzip 压缩

2.3 地形切片

主页上依次点击【数据处理】、【地形切片】进入地形切片工具。

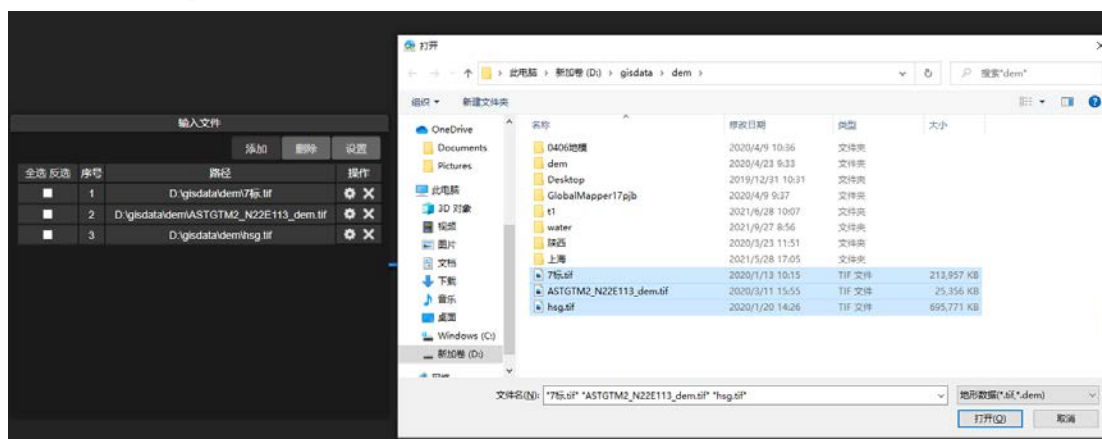


2.3.1 输入文件



2.3.1.1 添加输入

点击【添加】按钮，弹出选择文件对话框，一次可以选择多个 tif 文件，不限制文件大小和文件个数。



2.3.1.2 删除输入

选择一个或者多个输入后，点击【删除】按钮，选中的 tif 被从输入框删除。
也可以直接点击操作栏里【X】按钮，删除对应这一条输入。

2.3.1.3 设置输入参数

选择一个或者多个输入后，点击【设置】按钮。
或者点击操作栏里【⚙️】按钮。
弹出输入的属性设置面板

参数配置

空间参考: 经纬度局部坐标系 ?
EPSG编码搜索 ?
从地图上选择 ?
从文件导入 ?

零点坐标: 如: 0,0,0 ?

七参数: 请输入七参数, 以逗号隔开 ?

最大级别: 10 无数据值: 请输入值

取消 确定

2.3.1.4 空间参考

空间参考，零点坐标、七参数设置请参考本文【空间参考】部分。

注意如果这里没有设置，那么处理程序会尝试从输入 tif 的元信息里读取空间参考，如果没有读取到，该 tif 处理失败。

如果这里设置了，那么处理程序会使用设置的空间参考，但是如果此空间参考和 tif 元信息里的空间范围有冲突，也会导致处理失败或者出现结果异常。

所以注意，这里空间参考设置要么就保留空白，要么操作者明确知道数据正确空间参考，否则不要去浪费时间尝试。

2.3.1.5 最大级别

最大级别默认选中【自动计算】，处理程序会根据栅格数据的分辨率来计算切片级别，具体如下表：

级别	一个tile的经度范围	格网精度(经度/像素)	格网(米/像素)
0	180.0000000000	2.8125000000	313079.3428843080
1	90.0000000000	1.4062500000	156539.6714421540
2	45.0000000000	0.7031250000	78269.8357210770
3	22.5000000000	0.3515625000	39134.9178605385
4	11.2500000000	0.1757812500	19567.4589302692
5	5.6250000000	0.0878906250	9783.7294651346
6	2.8125000000	0.0439453125	4891.8647325673
7	1.4062500000	0.0219726563	2445.9323662837
8	0.7031250000	0.0109863281	1222.9661831418
9	0.3515625000	0.0054931641	611.4830915709
10	0.1757812500	0.0027465820	305.7415457855
11	0.0878906250	0.0013732910	152.8707728927
12	0.0439453125	0.0006866455	76.4353864464
13	0.0219726563	0.0003433228	38.2176932232
14	0.0109863281	0.0001716614	19.1088466116
15	0.0054931641	0.0000858307	9.5544233058
16	0.0027465820	0.0000429153	4.7772116529
17	0.0013732910	0.0000214577	2.3886058264
18	0.0006866455	0.0000107288	1.1943029132
19	0.0003433228	0.0000053644	0.5971514566
20	0.0001716614	0.0000026822	0.2985757283
21	0.0000858307	0.0000013411	0.1492878642
22	0.0000429153	0.0000006706	0.0746439321

比如 30 米格网精度的栅格高程数据，查表，应该是在 13 和 14 级之间，我们会向下一个级别，也就是取 14 做为最大级别。

一般来说【自动计算】是合适的，一些情况下可能需要自定义最大级别：

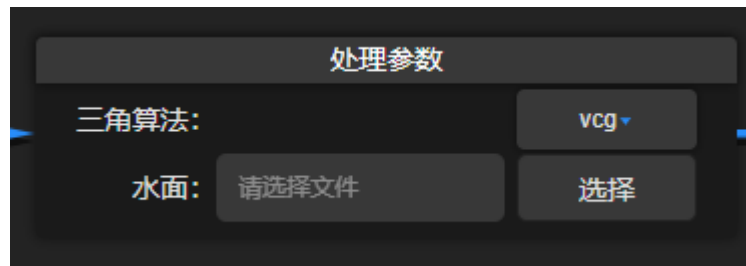
- 1, 数据精度很高，比如一些倾斜软件输出的 dsm，按【自动计算】都可能超过 22 级别，切到 22 级 dem 实际意义也不大，为了加速处理，可以设置到 17 左右。
- 2, 数据格网精度和高程差精度不匹配，常见在一些手动编辑过的 dem 数据，比如一些河道，深度也就 5 米左右，但是输出的 dem 格网也是 5 米，导致自动计算的最大级别，这个高程差体现不出来，可以设置一个更高的处理级别。

2.3.1.6 无数据值 (nodata)

栅格数据通过点位值来描述信息，但是由于一些原因（比如非矩形的裁剪），某些点位的高程值没有，但是栅格里该点位存储的就是无数据值，该信息一般是包含在 tif 头信息里，处理程序可以自己读取。但是也不排除有一些数据 tif 头信息里没有该信息，那么就需要用户设定该数值，至于设置多少，建议用 arcgis 打开高程数据，点击查询下无效范围的数值是多少，然后再填写。

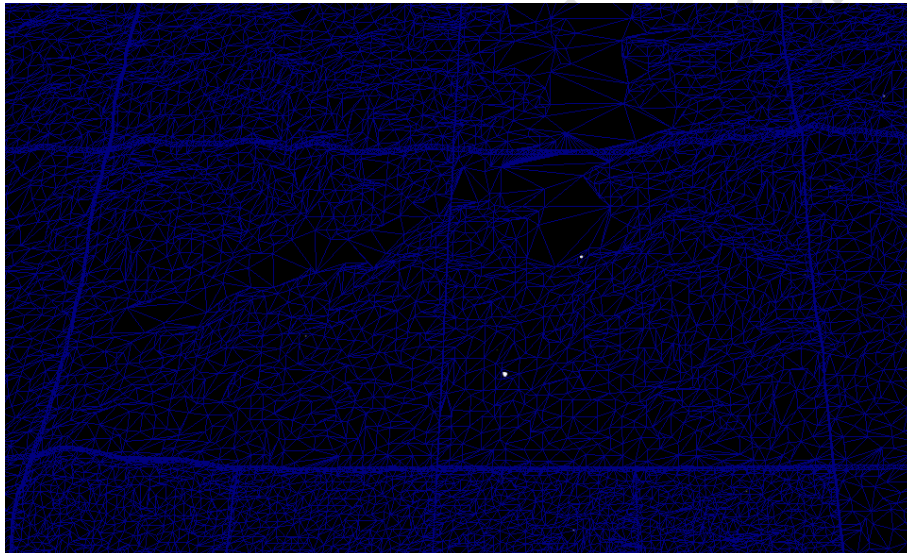
对于无数据值，处理程序会把该点位的高程设置为 0。

2.3.2 处理参数



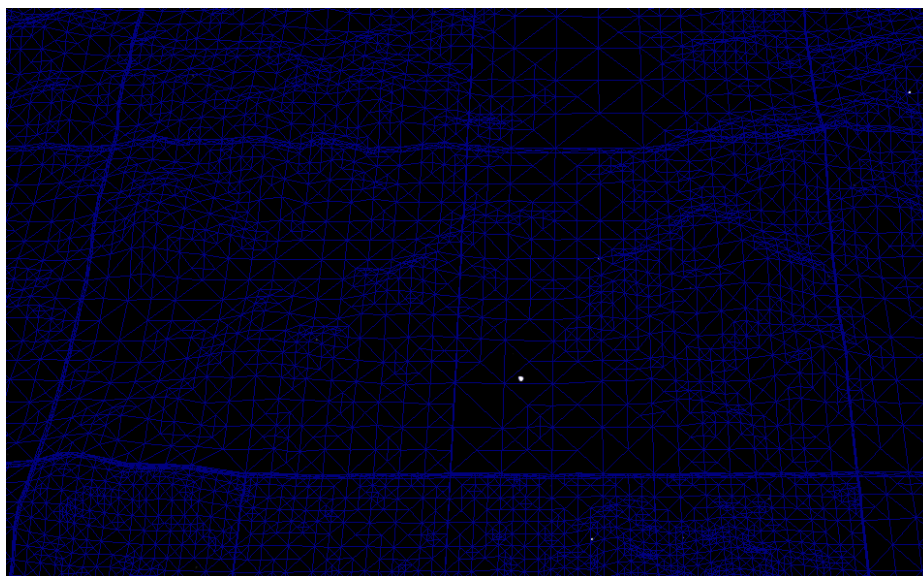
2.3.2.1 三角算法

vcg 是基于经典的边塌陷算法的三角网简化算法，该算法网格结构质量较好，误差更小，但是缺陷是处理速度较慢。



vcg 地形网格

ctb 是一种基于网格的快速三角网简化算法，算法计算速度快，缺陷是误差较大。



vcb 的时间消耗大约是 ctb 的 60 倍，所以根据您数据选择合理的构造算法。

2.3.2.2 水面文件

Cesium 的地形切片上可以附加一层水面蒙版 (watermask)，lab 的地形切片程序可以根据水面数据来生成对应地形的 watermask。

Lab 的水面文件支持两种形式的水面，一种是水面矢量数据，必须是面(polygon)数据，用来描述水域范围。但是当地形较大或者分辨率较高，矢量水面生成速度会变慢。

另一种是水面栅格数据，该栅格数据应满足如下要求：

- 1、水面栅格分辨率应高于 dem 分辨率 4 倍以上；
- 2、水面栅格坐标系应是 EPSG:4326；
- 3、水面栅格数据类型应是 8 bit unsigned integer；
- 4、水面栅格中有水的地方像素值为 255，无水的地方是 NoData(且 NoData 值为 0)。

注，cesium 地形上 watermask 效果比较差，仅仅是一个波浪纹理，而且这个效果会和底图混色，容易受到底图影响，水面效果不明显，更没有任何反射折射效果，做个大范围水域示意下可以，但是不要过高期待。

2.3.3 输出文件



地形输出

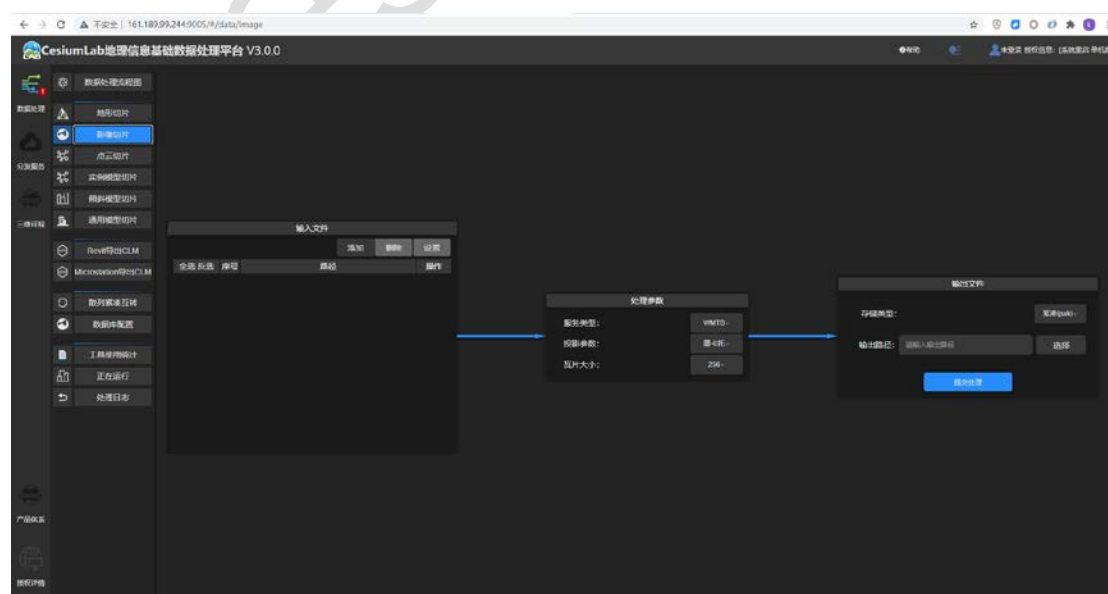
【存储类型】见本文【通用功能】部分【数据存储】介绍。

2.3.4 增量融合

在输入文件时，一次选择多个文件，当范围有重叠时，会自动融合。当多次处理，输出文件选择同一路径时，如果有重叠区域，也不会直接覆盖，而是采用增量处理的方式进行融合。

2.4 影像切片

主页上依次点击【数据处理】、【影像切片】进入影像切片工具



2.4.1 输入文件



2.4.1.1 添加输入

点击【添加】按钮，弹出选择文件对话框，一次可以选择多个文件 tif 文件，多个文件会列到列表里。不限制文件大小和文件个数。

2.4.1.2 删除输入

选择一个或者多个输入后，点击【删除】按钮，选中的 tif 被从输入列表删除。也可以直接点击操作栏里【X】按钮，删除对应这一条输入。

2.4.1.3 设置参数

选择一个或者多个输入后，点击【设置】按钮。
或者点击操作栏里【⚙️】按钮。
弹出输入的属性设置面板

参数配置

空间参考：

经纬度局部坐标系

EPSG编码搜索

从地图上选择

从文件导入

零点坐标：

如：0,0,0

七参数：

请输入七参数，以逗号隔开

最大级别：

自动计算

无数据值：

请输入值

颜色拉伸：

偏移缩放

最值

百分比

偏移：

0

缩放：

0

背景透明：

自动

第四波段

透明色

取消

确定

影像参数设置

2.4.1.4 空间参考

空间参考，零点坐标，七参数请参考本文通用功能，空间参考部分的文档。

注意如果这里没有设置，那么处理程序会尝试从输入 tif 的元信息里读取空间参考，如果没有读取到，该 tif 处理失败。

如果这里设置了，那么处理程序会使用设置的空间参考，但是如果此空间参考和 tif 元信息里的空间范围有冲突，也会导致处理失败 或者出现结果异常。

所以注意，这里空间参考设置要么就保留空白，要么操作者明确知道数据正确空间参考，否则不要去浪费时间尝试。

2.4.1.5 最大级别

最大级别默认选中【自动计算】，处理程序会根据栅格数据的分辨率来计算切片级别。因为影像切片支持两种投影的切片形式，一种是墨卡托，一种是经纬度，在本章【处理参数】【投影参数】里详细探讨。

如果设置了【最大级别】，我们内部会判定：

如果【设置的最大级别】< 【自动计算的最大级别】那么按照【设置的最大级别】来切片。

否则该设置值无效，依然是按照【自动计算的最大级别】来切片。这么做主要是避免无效的数据处理。

2.4.1.6 无数据值 (nodata)

栅格数据通过点位值来描述信息，但是由于一些原因（比如非矩形的裁剪），某些点位的高程值没有，栅格里该点位存储的就是无数据值，该信息一般是包含在 tif 头信息里，处理程序可以自己读取。但是也不排除有一些数据 tif 头信息里没有该信息，那么就需要用户设定该数值，至于设置多少，建议用 arcgis 打开影像数据，点击查询下无效范围的数值是多少，然后再填写。

对于无数据值，处理程序会把该点位的颜色设置为纯透明色 (0, 0, 0, 0)。

2.4.1.7 颜色拉伸

本文数据处理基础部分也阐述了栅格数据的定义，栅格数据每个位置有一个值（或者多个波段值），这个值不一定是颜色值，但是除了地形高程栅格外，其他类型的数据如果要做三维可视化都需要转为图片切片，那么切片里就说常规颜色值。所以就需要一个栅格值到颜色值的映射算法。当然对于绝大多数非专业的栅格（尤其是下载得到的）里面的值就是 RGBA，那么这个映射就很简单，每个点位最终颜色=栅格里存储的值。

几种设置方法的实际意义：

1，选中了【偏移缩放】



颜色拉伸: **偏移缩放** 最值 百分比

偏移: 0 缩放: 0

如果【偏移】是 0，【缩放】是 0，这种表示我们根据栅格的波段数值类型来进行自动处理。如果栅格的每个波段是 8bit，那么根据如下计算：

$$\text{切片颜色} = \text{栅格里存储的值}$$

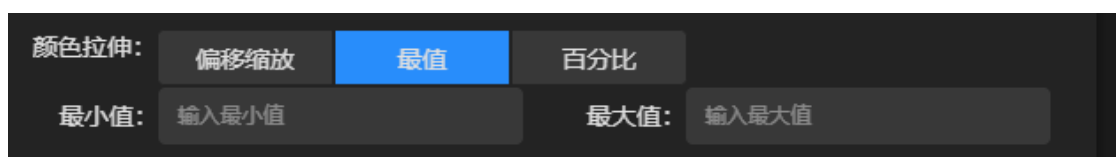
如果栅格的每个波段是 16bit，那么如下计算：

$$\text{切片颜色} = \text{栅格里存储的值} * 255 / 65535$$

如果【缩放】不是 0 值，那么按照下述计算

$$\text{切片颜色} = (\text{栅格里存储的值} + \text{偏移}) * \text{缩放}$$

3，选中了【最值】



颜色拉伸: 偏移缩放 **最值** 百分比

最小值: 输入最小值 最大值: 输入最大值

则按照下述计算

$$\text{切片的像素颜色} = (\text{栅格里存储的值} - \text{最小值}) * (255 / (\text{最大值} - \text{最小值}))$$

4, 选中了【百分比】



百分比裁剪拉伸首先根据最小百分比和最大百分比计算需要裁剪的最小像素个数和最大像素个数，然后把所有像素值进行排序，并统计每个像素值的个数，最小像素个数和最大像素个数进行裁剪，最后将剩余部分进行线性拉伸。

例如最小最大百分比都设置为 0.25 时，影像大小为 100*100 像素，首先把像素值从小到大排序，然后找到正序第 100*100*0.25 个像素的像素值 minvalue 和倒序第 100*100*0.25 个像素的像素值 maxvalue，最后进行线性拉伸（同最小最大值拉伸），如下述计算

$$\text{切片的像素颜色} = (\text{栅格里存储的值} - \text{minvalue}) * (255 / (\text{maxvalue} - \text{minvalue}))$$

2.4.1.8 背景透明

影像切片碰到的很多问题都是，切片之后的数据有白边或者黑边。



1, 选中了【自动】

处理程序内部自动根据栅格的波段元数据判定哪个波段是透明色。适合 tif 元数据信息完整的数据。

2, 选中了【第四波段】



如果栅格的波段数大于 3 个，那么强制使用第四波段是透明波段。适合实际数据带第四波段，并且明确知道该波段是透明值的数据。

3, 选中了【透明色】



无论栅格的波段数是多少，只要读取到的颜色等于设定的 RGB 值，认为该像素是纯透明的。适合透明信息不完整的数据。这种方式有个缺陷，比如我们在 arcgis 看到数据有白边，我们把透明色设置 (255, 255, 255)，处理完后，白边可以透明，但是有效范围内可能一些亮点也是该颜色，导致该点就会纯透明。

2.4.1.9 【无数据值】和【背景透明】的区别

我们切片的流程如下：

读取原始栅格值 -> 颜色计算 -> 透明计算

【无数据值】作用在前半部分，比如有一些索引色数据，我们读取到的原始栅格值只是一个表索引，如果该值=【无数据值】，我们直接认为该位置透明。

【背景透明】作用在后半部分，颜色计算前面也讨论了，读取到的栅格值可能需要做一些额外计算，并不一定等于原始栅格值。比如这时候选择【透明色】设置，那么判定的是计算后的颜色和【透明色】来对比相等。

2.4.2 处理参数



2.4.2.1 服务类型

实际两种不同的切片索引组织方式，具体如下：

1, WMTS

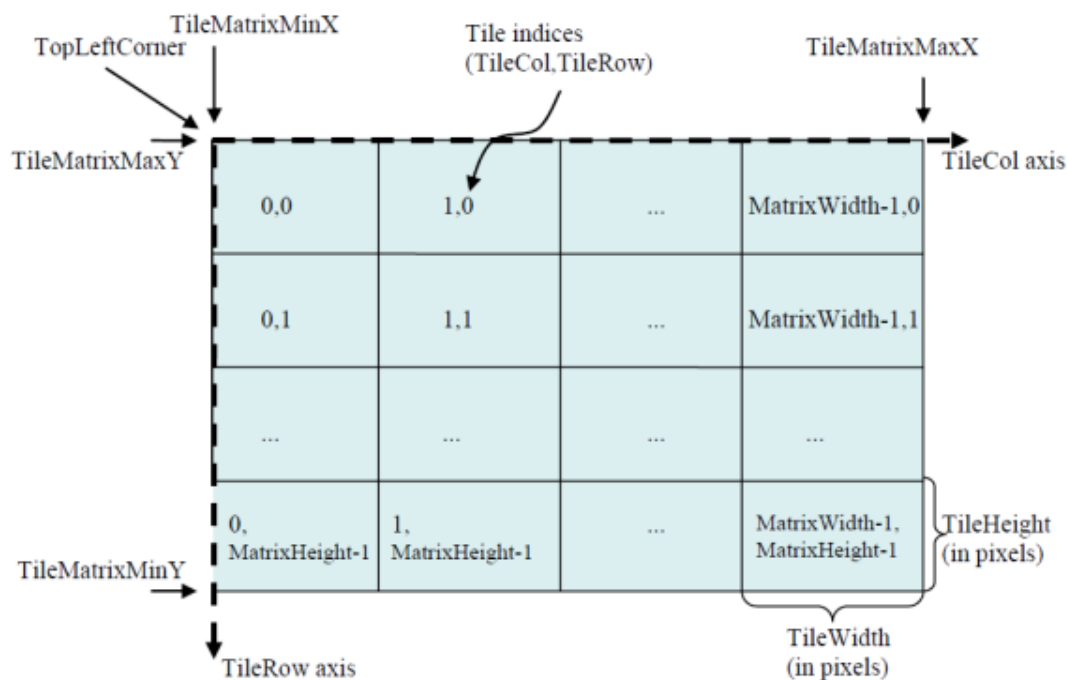
WMTS 是 OGC 提出的一种瓦片地图服务标准，包含了很多内容和一系列接口，CesiumLab 并没有完全兼容 WMTS 服务标准，只是借鉴其切片规则，WMTS 从 0 级开始，以地图左上角为原点，从左到右为横轴正方向，从上到下为纵轴正方向。CesiumLab 发布的 wmts 服务切片索引为{z}/{x}/{y}。

其中 z 为级别；x 是横轴方向的索引，y 是纵轴方向的索引。假设某点坐标为(x0,y0)，x 和 y 索引有如下计算公式：

$$x = \text{Math.floor}((x0 + r) / \text{tilesize})$$

$$y = \text{Math.floor}((r - y0) / \text{tilesize})$$

其中，r 为地球半周长，tilesize 为瓦片分辨率，详细参见下一节[投影参数和瓦片大小](#)。



关于 WMTS 更详细内容可以参考: https://portal.ogc.org/files/?artifact_id=35326

2, TMS

TMS 也是种标准的瓦片地图服务, 其定义非常灵活, 但默认情况下其与 WMTS 非常相似, 只是以地图左下角为原点, 纵轴从下到上为正方向。那么对于某点坐标 (x_0, y_0) , x 和 y 索引有如下计算公式:

$$x = \text{Math.floor}((x_0 + r) / \text{tilesize})$$

$$y = \text{Math.floor}((y_0 + r) / \text{tilesize})$$

其中, r 为地球半周长, tilesize 为瓦片分辨率, 详细参见下一节[投影参数和瓦片大小](#)。

关于 TMS 更详细内容可以参考:

https://wiki.osgeo.org/wiki/Tile_Map_Service_Specification

2.4.2.2 投影参数和瓦片大小

根据常规传统, 瓦片大小都是 256, 但是事实上瓦片大小可以设置非 256, 比如 512, 这样做的目的可以在大屏上减少瓦片的请求次数。

投影参数有两种。

墨卡托的投影级别和精度如下表:

赤道半径	6378137	球周长	40075016.69
	瓦片大小	256	
级别	一个tile的web墨卡托范围	格网精度(米/像素)	
0	40075016.6855785000	156543.0339280410	
1	20037508.3427892000	78271.5169640205	
2	10018754.1713946000	39135.7584820102	
3	5009377.0856973100	19567.8792410051	
4	2504688.5428486600	9783.9396205026	
5	1252344.2714243300	4891.9698102513	
6	626172.1357121640	2445.9849051256	
7	313086.0678560820	1222.9924525628	
8	156543.0339280410	611.4962262814	
9	78271.5169640205	305.7481131407	
10	39135.7584820102	152.8740565704	
11	19567.8792410051	76.4370282852	
12	9783.9396205026	38.2185141426	
13	4891.9698102513	19.1092570713	
14	2445.9849051256	9.5546285356	
15	1222.9924525628	4.7773142678	
16	611.4962262814	2.3886571339	
17	305.7481131407	1.1943285670	
18	152.8740565704	0.5971642835	
19	76.4370282852	0.2985821417	
20	38.2185141426	0.1492910709	
21	19.1092570713	0.0746455354	
22	9.5546285356	0.0373227677	

256 瓦片大小的 web 墨卡托切片

赤道半径	6378137	球周长	40075016.69
	瓦片大小	512	
级别	一个tile的web墨卡托范围	格网精度(米/像素)	
0	40075016.6855785000	78271.5169640205	
1	20037508.3427892000	39135.7584820102	
2	10018754.1713946000	19567.8792410051	
3	5009377.0856973100	9783.9396205026	
4	2504688.5428486600	4891.9698102513	
5	1252344.2714243300	2445.9849051256	
6	626172.1357121640	1222.9924525628	
7	313086.0678560820	611.4962262814	
8	156543.0339280410	305.7481131407	
9	78271.5169640205	152.8740565704	
10	39135.7584820102	76.4370282852	
11	19567.8792410051	38.2185141426	
12	9783.9396205026	19.1092570713	
13	4891.9698102513	9.5546285356	
14	2445.9849051256	4.7773142678	
15	1222.9924525628	2.3886571339	
16	611.4962262814	1.1943285670	
17	305.7481131407	0.5971642835	
18	152.8740565704	0.2985821417	
19	76.4370282852	0.1492910709	
20	38.2185141426	0.0746455354	
21	19.1092570713	0.0373227677	
22	9.5546285356	0.0186613839	

512 瓦片大小的 web 墨卡托切片

赤道半径	6378137	球周长	40075016.69
	瓦片大小	1024	
级别	一个tile的web墨卡托范围	格网精度(米/像素)	
0	40075016.6855785000	39135.7584820102	
1	20037508.3427892000	19567.8792410051	
2	10018754.1713946000	9783.9396205026	
3	5009377.0856973100	4891.9698102513	
4	2504688.5428486600	2445.9849051256	
5	1252344.2714243300	1222.9924525628	
6	626172.1357121640	611.4962262814	
7	313086.0678560820	305.7481131407	
8	156543.0339280410	152.8740565704	
9	78271.5169640205	76.4370282852	
10	39135.7584820102	38.2185141426	
11	19567.8792410051	19.1092570713	
12	9783.9396205026	9.5546285356	
13	4891.9698102513	4.7773142678	
14	2445.9849051256	2.3886571339	
15	1222.9924525628	1.1943285670	
16	611.4962262814	0.5971642835	
17	305.7481131407	0.2985821417	
18	152.8740565704	0.1492910709	
19	76.4370282852	0.0746455354	
20	38.2185141426	0.0373227677	
21	19.1092570713	0.0186613839	
22	9.5546285356	0.0093306919	

1024 瓦片大小的 web 墨卡托切片

经纬度的投影级别和精度如下表：

C	D	E	F
	瓦片大小	256	
级别	一个tile的经度范围	格网精度(经度/像素)	格网(米/像素)
0	180.0000000000	0.7031250000	78269.8357210770
1	90.0000000000	0.3515625000	39134.9178605385
2	45.0000000000	0.1757812500	19567.4589302692
3	22.5000000000	0.0878906250	9783.7294651346
4	11.2500000000	0.0439453125	4891.8647325673
5	5.6250000000	0.0219726563	2445.9323662837
6	2.8125000000	0.0109863281	1222.9661831418
7	1.4062500000	0.0054931641	611.4830915709
8	0.7031250000	0.0027465820	305.7415457855
9	0.3515625000	0.0013732910	152.8707728927
10	0.1757812500	0.0006866455	76.4353864464
11	0.0878906250	0.0003433228	38.2176932232
12	0.0439453125	0.0001716614	19.1088466116
13	0.0219726563	0.0000858307	9.5544233058
14	0.0109863281	0.0000429153	4.7772116529
15	0.0054931641	0.0000214577	2.3886058264
16	0.0027465820	0.0000107288	1.1943029132
17	0.0013732910	0.0000053644	0.5971514566
18	0.0006866455	0.0000026822	0.2985757283
19	0.0003433228	0.0000013411	0.1492878642
20	0.0001716614	0.0000006706	0.0746439321
21	0.0000858307	0.0000003353	0.0373219660
22	0.0000429153	0.0000001676	0.0186609830

256 瓦片大小经纬度切片

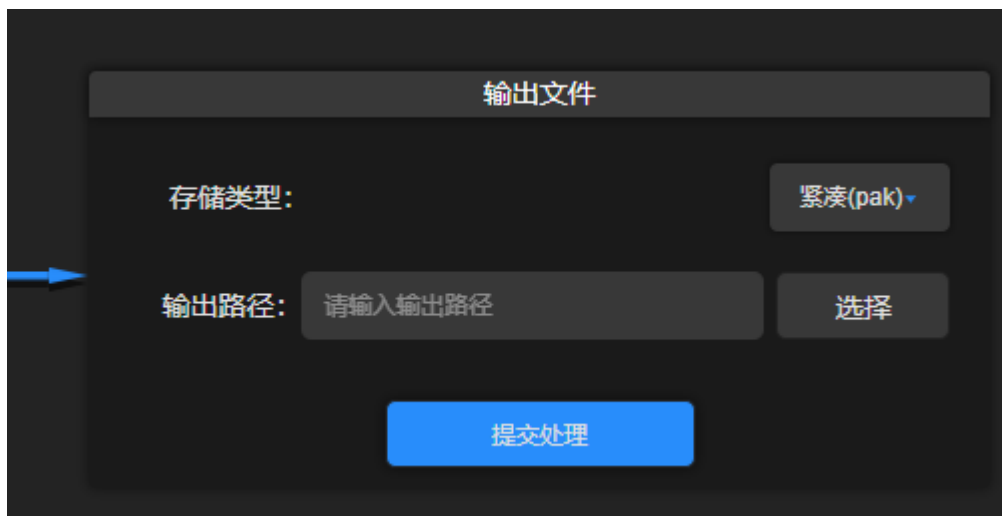
级别	瓦片大小 一个tile的经度范围	512		经度/米	
		格网精度(经度/像素)	格网(米/像素)		
0	180.0000000000	0.3515625000	39134.9178605385		0.0000089833
1	90.0000000000	0.1757812500	19567.4589302692		0.0000089833
2	45.0000000000	0.0878906250	9783.7294651346		0.0000089833
3	22.5000000000	0.0439453125	4891.8647325673		0.0000089833
4	11.2500000000	0.0219726563	2445.9323662837		0.0000089833
5	5.6250000000	0.0109863281	1222.9661831418		0.0000089833
6	2.8125000000	0.0054931641	611.4830915709		0.0000089833
7	1.4062500000	0.0027465820	305.7415457855		0.0000089833
8	0.7031250000	0.0013732910	152.8707728927		0.0000089833
9	0.3515625000	0.0006866455	76.4353864464		0.0000089833
10	0.1757812500	0.0003433228	38.2176932232		0.0000089833
11	0.0878906250	0.0001716614	19.1088466116		0.0000089833
12	0.0439453125	0.0000858307	9.5544233058		0.0000089833
13	0.0219726563	0.0000429153	4.7772116529		0.0000089833
14	0.0109863281	0.0000214577	2.3886058264		0.0000089833
15	0.0054931641	0.0000107288	1.1943029132		0.0000089833
16	0.0027465820	0.0000053644	0.5971514566		0.0000089833
17	0.0013732910	0.0000026822	0.2985757283		0.0000089833
18	0.0006866455	0.0000013411	0.1492878642		0.0000089833
19	0.0003433228	0.0000006706	0.0746439321		0.0000089833
20	0.0001716614	0.0000003353	0.0373219660		0.0000089833
21	0.0000858307	0.0000001676	0.0186609830		0.0000089833
22	0.0000429153	0.0000000838	0.0093304915		0.0000089833

512 瓦片大小经纬度切片

级别	瓦片大小 一个tile的经度范围	1024		经度/米	
		格网精度(经度/像素)	格网(米/像素)		
0	180.0000000000	0.1757812500	19567.4589302692		0.0000089833
1	90.0000000000	0.0878906250	9783.7294651346		0.0000089833
2	45.0000000000	0.0439453125	4891.8647325673		0.0000089833
3	22.5000000000	0.0219726563	2445.9323662837		0.0000089833
4	11.2500000000	0.0109863281	1222.9661831418		0.0000089833
5	5.6250000000	0.0054931641	611.4830915709		0.0000089833
6	2.8125000000	0.0027465820	305.7415457855		0.0000089833
7	1.4062500000	0.0013732910	152.8707728927		0.0000089833
8	0.7031250000	0.0006866455	76.4353864464		0.0000089833
9	0.3515625000	0.0003433228	38.2176932232		0.0000089833
10	0.1757812500	0.0001716614	19.1088466116		0.0000089833
11	0.0878906250	0.0000858307	9.5544233058		0.0000089833
12	0.0439453125	0.0000429153	4.7772116529		0.0000089833
13	0.0219726563	0.0000214577	2.3886058264		0.0000089833
14	0.0109863281	0.0000107288	1.1943029132		0.0000089833
15	0.0054931641	0.0000053644	0.5971514566		0.0000089833
16	0.0027465820	0.0000026822	0.2985757283		0.0000089833
17	0.0013732910	0.0000013411	0.1492878642		0.0000089833
18	0.0006866455	0.0000006706	0.0746439321		0.0000089833
19	0.0003433228	0.0000003353	0.0373219660		0.0000089833
20	0.0001716614	0.0000001676	0.0186609830		0.0000089833
21	0.0000858307	0.0000000838	0.0093304915		0.0000089833
22	0.0000429153	0.0000000419	0.0046652458		0.0000089833

1024 瓦片大小经纬度切片

2.4.3 输出文件



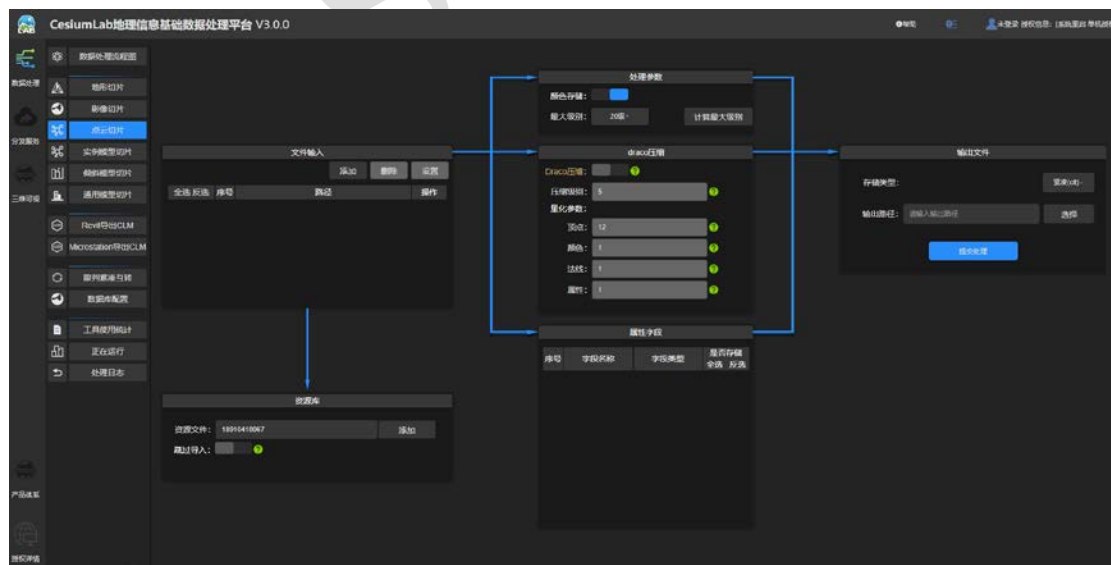
【存储类型】见本文【通用功能】部分【数据存储】介绍。

2.4.4 增量融合

在输入文件时，一次选择多个文件，当范围有重叠时，会自动融合。当多次处理，输出文件选择同一路径时，如果有重叠区域，也不会直接覆盖，而是采用增量处理的方式进行融合。

2.5 点云切片

主页上依次点击【数据处理】、【点云切片】进入点云切片工具



2.5.1 输入文件

2.5.1.1 添加输入

点击【添加】按钮，弹出选择文件对话框，一次可以选择多个文件 las 文件。

2.5.1.2 删除输入

选择一个或者多个输入后，点击【删除】按钮，选中的 las 被从输入列表删除。
也可以直接点击操作栏里【X】按钮，删除对应这一条输入。

2.5.1.3 设置参数

选择一个或者多个输入后，点击【设置】按钮。
或者点击操作栏里【⚙️】按钮。
弹出输入的属性设置面板

The screenshot shows a 'Parameter Configuration' (参数配置) dialog box with the following fields and controls:

- 空间参考:** A large text input field. To its right are four buttons: '经纬度局部坐标系' (Geographic Local Coordinate System), 'EPSG编码搜索' (EPSG Code Search), '从地图上选择' (Select from Map), and '从文件导入' (Import from File). Each button has a green question mark icon.
- 零点坐标:** A text input field containing '0,0,0' with a green question mark icon to its right.
- 七参数:** A red text input field with the placeholder text '请输入七参数，以逗号隔开' (Please enter seven parameters, separated by commas) and a green question mark icon to its right.
- 颜色计算:** A formula input field showing '(color+ 0)* 255 / 65535'.

2.5.1.4 空间参考

空间参考，零点坐标，七参数请参考本文通用功能，空间参考部分的文档。

注意如果这里没有设置，那么处理程序会尝试从输入 las 的头信息里读取空间参考，如果没有读取到，会提示未添加空间参考。

2.5.1.5 颜色计算

las 文件本身是可以存储颜色信息的, 但是和栅格数据类似, 里面可能并非直接存储的最大为 255 的 RGB 值, 最常见的存储为最大值为 65535 的颜色, 如图所示的填写方式, 我们会根据此公式来计算颜色

颜色计算: (color+ 0) * 255 / 65535

2.5.2 资源库



2.5.2.1 资源文件

点云切片分为两大过程: 数据导入, 数据切片

在数据导入过程我们会把点云数据暂存。如果这里点击【添加】设置了一个本地文件路径, 那么数据暂存到该文件内。

如果这里没有设置, 那么实际暂存在内存中。

如果您切片的数据量超级大, 比如几十 G 的点云数据, 我们推荐您设置一个本地缓存路径。比如今天我们拿到了一个 las 文件, 我们设置了本地缓存进行切片。明天我们又拿到另一个 las 文件, 需要把两个文件合并在一起, 那么可以直接选择昨天的本地缓存。这样就完成了点云切片的增量处理和局部更新。

2.5.2.2 跳过导入

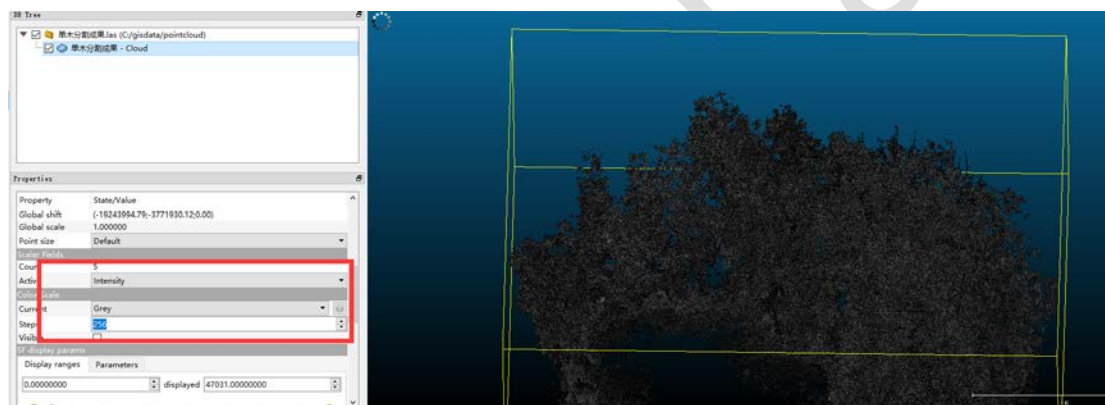
该选项是个【调试选项】, 对于大数据, 导入和切片过程都是很耗时的, 而且可能会出错, 为了方便解决 bug, 如果打开了此选项, 并且设置了本地缓存, 那么不进行数据导入过程, 直接从本地缓存进行切片过程。

2.5.3 处理参数



2.5.3.1 颜色存储

如果打开了该选项，那么我们会在 3dtiles 存储点云的颜色。注意一点，部分激光扫描的点云一般都不带颜色值，而倾斜处理生成的点云是带颜色的，所以根据您的数据来开启该选项，如果不开启会减小 3dtiles 的量。需要注意一点，数据在 cloudcompare 软件下看到的颜色，不一定是点云的颜色，可能是根据某个字段值着色得到的：



CloudCompare 下的点云颜色

如果想做到这个效果，请参考本章【属性字段】。

2.5.3.2 最大级别

和前面提到的栅格切片不同地方，las 文件信息里基本没有分辨率或者格网精度的信息。主要原因是 las 是一种散列空间点位结构，并没有强制约束必须均匀分布。但是对于我们切片程序和可视化需求中，我们还是需要有一个最大级别，这个最大级别也必须根据点云的精度来计算。我们依据下表来计算点云的切片级别：

地球半周长	20037508.34	
块精简参数	512	
级别	级别大小(米)	点云精度
0	20037508.34	39135.75848
1	10018754.17	19567.87924
2	5009377.086	9783.939621
3	2504688.543	4891.96981
4	1252344.271	2445.984905
5	626172.1357	1222.992453
6	313086.0679	611.4962263
7	156543.0339	305.7481131
8	78271.51696	152.8740566
9	39135.75848	76.43702829
10	19567.87924	38.21851414
11	9783.939621	19.10925707
12	4891.96981	9.554628536
13	2445.984905	4.777314268
14	1222.992453	2.388657134
15	611.4962263	1.194328567
16	305.7481131	0.597164283
17	152.8740566	0.298582142
18	76.43702829	0.149291071
19	38.21851414	0.074645535
20	19.10925707	0.037322768
21	9.554628536	0.018661384
22	4.777314268	0.009330692
23	2.388657134	0.004665346
24	1.194328567	0.002332673
25	0.597164283	0.001166336

剩下的问题就是如何估算点云精度，我们内部是根据点云的平均距离来计算点云精度，但是这个平均距离随着点的数量增加会很缓慢。所以我们把这个过程单列出来。对于同一批点云数据，其实最大级别应该是一致，没必要每次切片前都计算，可以直接选择。对于陌生的数据，我们提供了按钮【计算最大级别】，点击后，等待片刻，我们会根据当前选择第一个输入 las 文件去计算点云距离，然后按照上表查询得到最大级别。

2.5.4 Draco 压缩

CesiumLab3 支持点云的 draco 压缩，可以极大减小 3dtiles 的数据量。

<https://cesium.com/blog/2019/02/26/draco-point-clouds/>

draco压缩

Draco压缩: ☐ ?

压缩级别: 5 ?

量化参数:

顶点: 12 ?

颜色: 1 ?

属性: 1 ?

Draco 压缩：是否开启 Draco 压缩；

压缩级别：分为 1-10 级，越大压缩率越高，压缩速度越慢；

量化参数：可以是 1-31 中的任意值，量化参数越小压缩率越高，但数据精度损失越严重。实际处理时，如果结果有明显失真，则可以调大量化参数，相反可以调小。Draco 压缩不仅仅是对顶点的压缩，颜色，属性都可以压缩。

2.5.5 属性字段

属性字段			
序号	字段名称	字段类型	是否存储 全选 反选
1	x	double	☐
2	y	double	☐
3	z	double	☐
4	Intensity	uint8_t	☐
5	ReturnNumber	uint8_t	☐
6	NumberOfReturns	uint8_t	☐
7	ClassificationFl...	uint8_t	☐
8	ScanDirectionFlag	uint8_t	☐
9	EdgeofFlightLine	uint8_t	☐
10	Classification	uint8_t	☐

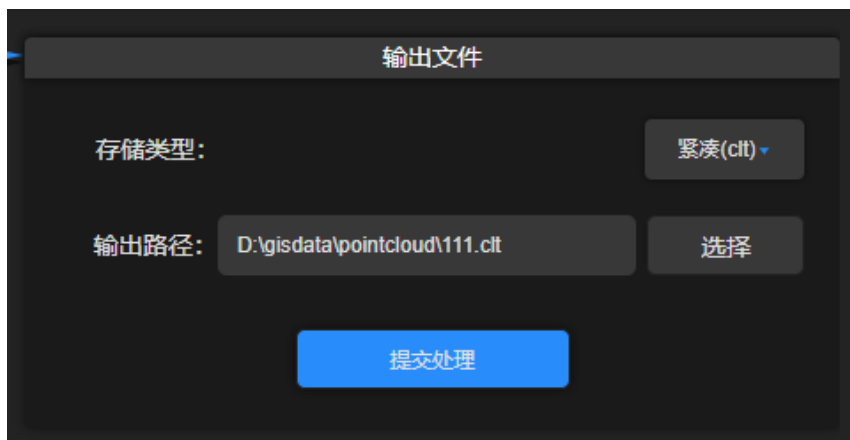
选中了一个 las 之后，我们会列出该 las 的属性字段，注意目前还不支持 las 的自定义字段。这里需要注意下，x, y, z 字段，las 并没有该属性，但是有一些需求，比如需要根据点云进行高度着色，有一些简单做法，直接在 shader 里根据渲染数据的顶点 z 值进行处理得到颜色，但是本文【空间参考】的【输出数据的空间参考】里已经讨论过，切片后的 z 值可能是一个相对坐标，并不直接代表高程值。为了实现准确的高度着色，我们可以在此部分勾选 z 字段，我们会把原始点云里的 z 坐标当作一个属性存入 3dtiles，使用 3dtilesstyle 可以根据此 z 直接进行颜色设置即可。

对于其他属性字段也是类似的，根据数据可视化的需要，选择需要的属性字段，还需要注意点，比如 Intensity 字段只要您选了 las，我们都会列出该字段，但是对于很多数据来说，该字段里的值都是 0 或者一个统一值，那么即便您勾选了，存储下来了，最终也无法做到一个好的着色效果。综述，属性的选择需要根据您数据的实际情况和可视化的需求来选择。

一个根据 Intensity 灰度着色的 3dtilestyle 示例

```
var style = {
  color:
    "vec4(${Intensity}/4600,${Intensity}/4600,${Intensity}/4600,1)"
}
```

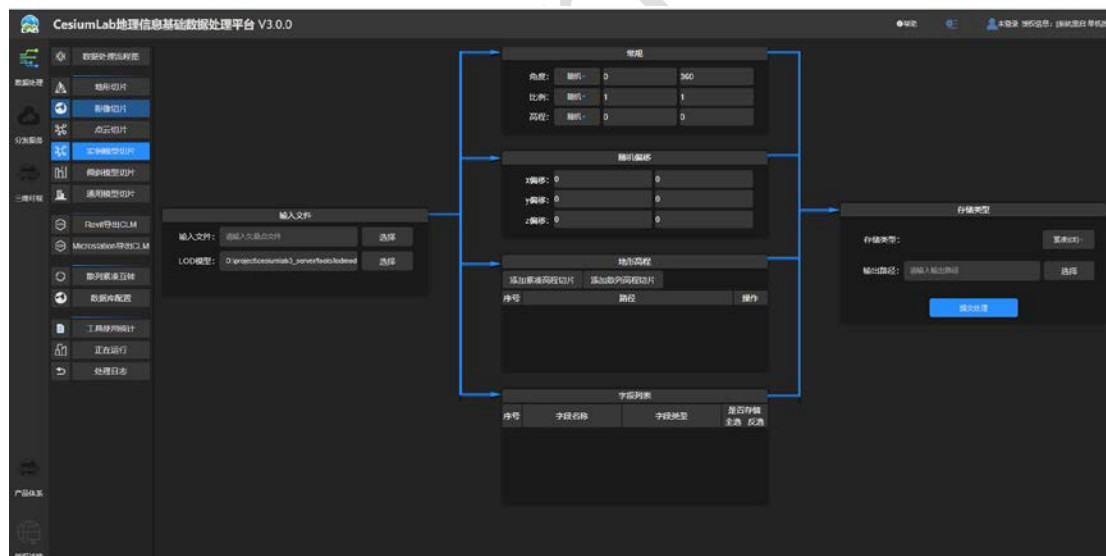

2.5.6 输出文件



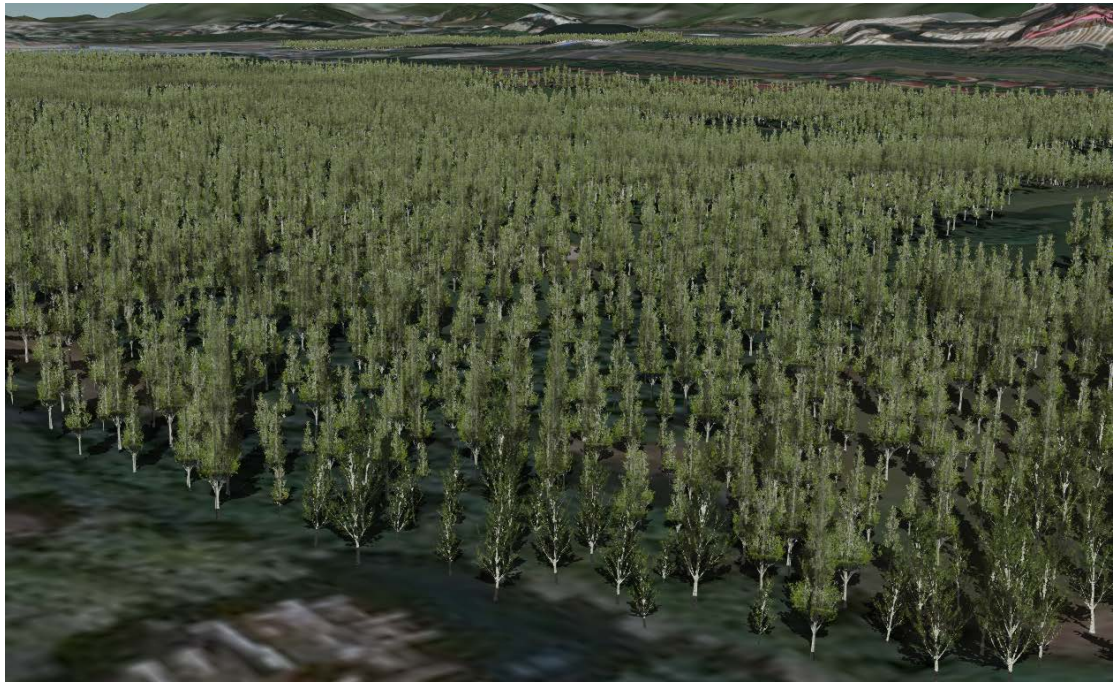
输出文件的选择在本文【数据存储】部分已经讨论过，这里注意点，对于点云来说，一般量很大，我们强烈推荐选择紧凑存储方式。

2.6 实例模型切片

主顶上依次点击【数据处理】、【实例模型切片】进入实例模型切片工具



我们先解释下实例模型，实例模型使用 GPU instance 技术来渲染的模型，通常用来绘制大量几何体一致，但是位置姿态不同的对象，比如说森林场景，大量路灯，井盖等，如下图：

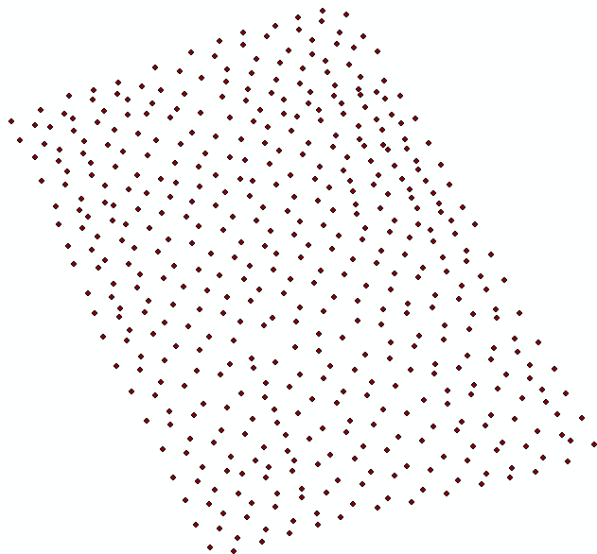


2.6.1 输入文件



2.6.1.1 选择输入文件

目前输入文件仅支持 shp 点数据，该数据用 arcgis 打开应该如下所示，其他格式或者 shp 不是点数据，将无法处理。

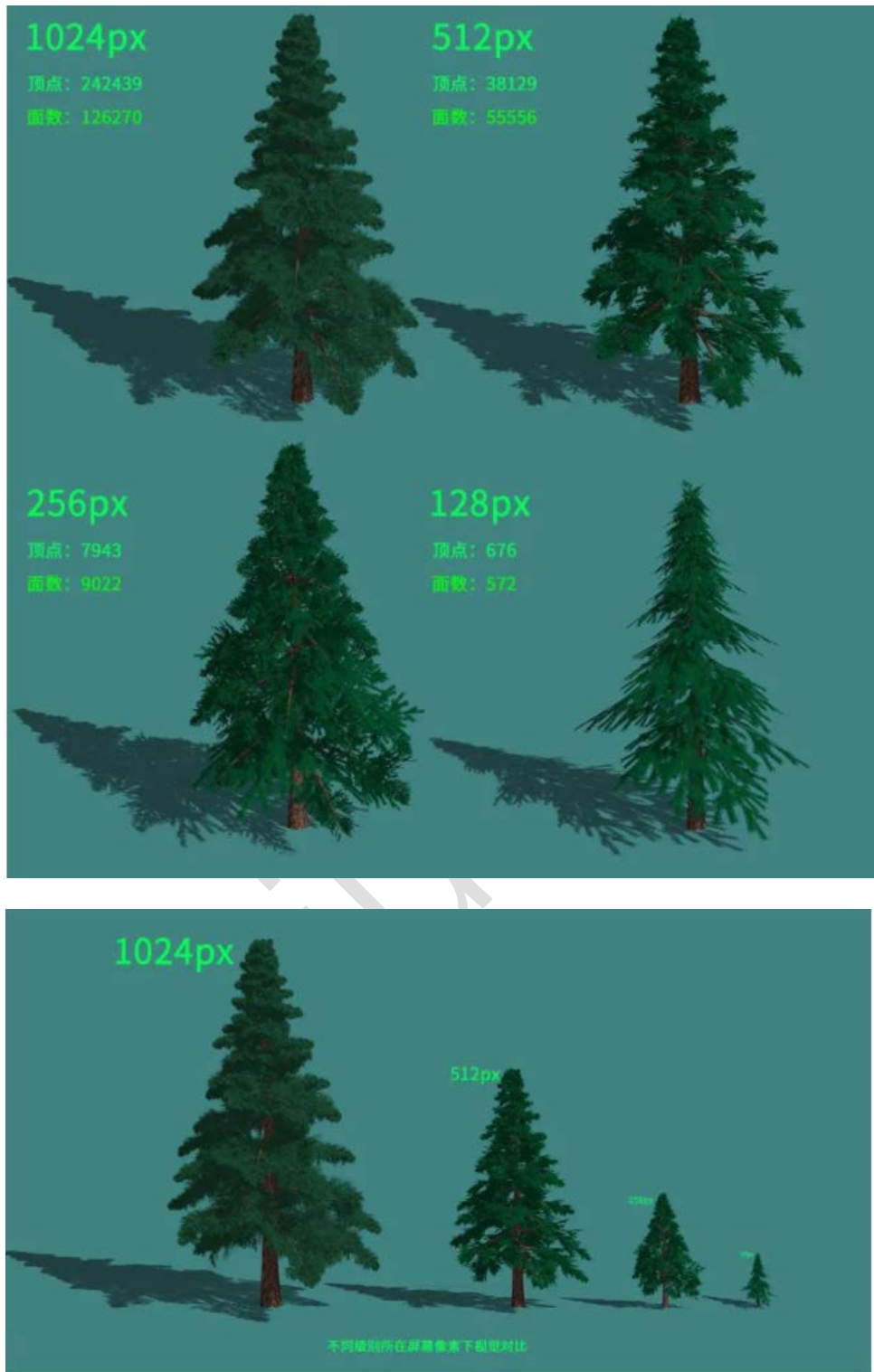


2.6.1.2 LOD 模型

LOD 模型是我们自定义的一组模型资源，它由一个 meta.json 和 一些 glb（或者 gltf 以及相关资源）来定义。随着 lab3 发布我们默认指定了 lab3 自带的一个树的 lod 模型，给大家做测试使用，此字段的路径默认就是该路径。

在不影响对应屏幕投影像素下级别的模型画面视觉效果的情况下，通过逐次简化模型的细节与贴图的大小来减少模型的几何复杂性和贴图大小，从而提高效率。

最大模型精度定为 1024px，最小定为 2px。1024px 精度指的是模型在 1024px 屏幕像素下不影响视觉效果的精度，之所以定 1024px 为最大精度，因为在一般显示器的屏幕分辨率下，1024px 已经近乎占满全屏或者超过全屏。最小为 2px，即使在屏幕像素 2px 下时模型已经很小，近乎于一个像素点，但此时模型依然存在高度，所以不能使用颜色数值代替。一棵在 1024、512、256、128 屏幕像素下各级别的松树模型，随着投影像素的变化，模型的表面复杂程度在逐次简化，顶点与面数在大幅减少。



不同级别所在屏幕像素下视觉对比

LOD 模型所有资源放在一个目录下，模型命名，直接以投影像素值命名，例如

名称	修改日期	类型	大小
2.bin	2020/3/11 14:15	BIN 文件	1 KB
2.gltf	2020/3/11 14:15	3D Object	4 KB
2_0.png	2020/3/11 14:15	PNG 文件	1 KB
4.bin	2020/3/11 14:15	BIN 文件	1 KB
4.gltf	2020/3/11 14:15	3D Object	4 KB
4_0.png	2020/3/11 14:15	PNG 文件	1 KB
8.bin	2020/3/11 14:15	BIN 文件	1 KB
8.gltf	2020/3/11 14:15	3D Object	4 KB
8_0.png	2020/3/11 14:15	PNG 文件	1 KB
16.bin	2020/3/11 14:15	BIN 文件	1 KB
16.gltf	2020/3/11 14:15	3D Object	4 KB
16_0.png	2020/3/11 14:15	PNG 文件	1 KB
32.bin	2020/3/11 14:15	BIN 文件	1 KB
32.gltf	2020/3/11 14:15	3D Object	4 KB
32_0.png	2020/3/11 14:15	PNG 文件	4 KB
64.bin	2020/3/11 14:15	BIN 文件	1 KB
64.gltf	2020/3/11 14:15	3D Object	4 KB
64_0.png	2020/3/11 14:15	PNG 文件	244 KB
128.bin	2020/3/11 14:15	BIN 文件	144 KB
128.gltf	2020/3/11 14:15	3D Object	6 KB
128_0.png	2020/3/11 14:15	PNG 文件	2 KB
128_1.jpg	2020/3/11 14:15	JPG 文件	4 KB
256.bin	2020/3/11 14:15	BIN 文件	144 KB
256.gltf	2020/3/11 14:15	3D Object	6 KB
256_0.png	2020/3/11 14:15	PNG 文件	4 KB
256_1.jpg	2020/3/11 14:15	JPG 文件	13 KB
512.bin	2020/3/11 14:15	BIN 文件	2,401 KB
512.gltf	2020/3/11 14:15	3D Object	6 KB
512_0.png	2020/3/11 14:15	PNG 文件	20 KB
512_1.jpg	2020/3/11 14:15	JPG 文件	49 KB
1024.bin	2020/3/11 14:15	BIN 文件	4,811 KB
1024.gltf	2020/3/11 14:15	3D Object	9 KB
1024_0.png	2020/3/11 14:15	PNG 文件	66 KB
1024_1.jpg	2020/3/11 14:15	JPG 文件	167 KB
meta.json	2020/3/11 14:15	JSON File	1 KB

引用纹理图片命名，例：32.gltf 引用的图片为 32_0.png，32_1.png 其它依次累积
meta.json 定义了模型的 名称、缩略图、基础颜色、中心点、大小，内容如下图。这里需要注意的一点是，lod0pixel 指定了第一级 lod 模型的像素大小，比如是 2，那么在“lod”

```
{
  "url": "16.gltf"
},
{
  "url": "16.gltf"
},
{
  "url": "16.gltf"
},
{
  "url": "16.gltf"
},
{
  "url": "32.gltf"
```

指定的模型文件里，前 4 个 16.gltf，分别表示投影像素是 2，4，8，16 的时候使用该模型。

```
1  {
2    "thumbnail": "2_0.png",
3    "name": "杨树",
4    "color": [
5      94,
6      104,
7      62,
8      1
9    ],
10   "center": [
11     0.19,
12     0.208,
13     3.632
14   ],
15   "size": [
16     2.995,
17     2.913,
18     7.264
19   ],
20   "lod0Pixel": 2,
21   "lod": [
22     {
23       "url": "16.gltf"
24     },
25     {
26       "url": "16.gltf"
27     },
28     {
29       "url": "16.gltf"
30     },
31     {
32       "url": "16.gltf"
33     },
34     {
35       "url": "32.gltf"
36     },
37     {
38       "url": "64.gltf"
39     },
40     {
41       "url": "128.gltf"
42     },
43     {
44       "url": "256.gltf"
45     },
46     {
47       "url": "512.gltf"
48     },
49     {
50       "url": "1024.gltf"
51     }
52   ]
53 }
```

2.6.2 位置姿态

2.6.2.1 位置

x偏移:	0	0
y偏移:	0	0
z偏移:	0	0

x,y,z 偏移默认都是 0,0, 表示位置没有任何偏移, 位置就使用 shp 里的点对象位置。如果是非 0 值, 比如

x偏移:	-1	10
------	----	----

表示 x 位置偏移会在 -1 ~ 10 之间随机产生
y,z 类似, 不再赘述。

2.6.2.2 角度

角度:	随机▼	0	360
-----	-----	---	-----

目前角度仅表示水平方向的旋转。

默认选中【随机】, 范围 0~360, 表示随机旋转角度。

如果选中【字段】, 并且选择了 shp 的某个字段, 那么角度会根据该字段来计算。

角度:	字段▼	size▼	
比例:	随机▼	Id	1
高程:	随机▼	angle	0
		size	
		height	

2.6.2.3 大小比例

比例:	随机▼	1	1
-----	-----	---	---

比例默认选择【随机】, 范围 1~1, 表示默认大小, 所有模型都是原始 glb 大小。如果如下

设置：

比例: 随机 0.1 10

表示比例会在 0.1~10 之间随机。

如果比例选择【字段】，并且选择了对应字段后，比例值使用该字段值。

比例: 字段 size
高程: 随机 Id 0
angle
size
height

2.6.3 高程

高程计算分为两部分：点位高程 和 地表高程，两者是叠加关系：

$$\text{最终高程} = \text{点位高程} + \text{地表高程}$$

2.6.3.1 点位高程

高程: 随机 0 0

如果这么选择，那么点位高程就是 0，

如果设定了非 0 的范围，那么高程会在最小最大之间随机。

如果选择【字段】

高程: 字段 height

那么点位高程依据该字段值定义。

2.6.3.2 地表高程

添加紧凑高程切片		添加散列高程切片
序号	路径	操作
1	D:\testdata\dem\tiles	×
2	C:\gisdata\dem\12.pak	×

地表高程我们支持从地形切片的输出数据来采样，而且支持多个地形数据。

点击【添加紧凑高程切片】选择 pak 地形切片，会列在此表格里。

点击【添加散列高程切片】选择散列地形切片，散列地形会列在此表格里。
处理程序会按照列表顺序来依次采样，只要正确采样到对应点位高度，就停止采样下一个。

2.6.4 字段列表

字段列表			
序号	字段名称	字段类型	是否存储 全选 反选
1	Id	Integer	☐
2	angle	Real	☐
3	size	Real	☐
4	height	Real	☐

选择输入文件后，这里会列出 shp 里所有字段，根据实际需要选择哪些字段要存储在 3dtiles 里，存储在 3dtiles 里的字段可以依据 3dtilestyle 语言进行着色和可见性控制。字段存储会增加 3dtiles 的量，所以按需选则。

2.6.5 存储类型

存储类型

存储类型:

紧凑(cit) ▾

输出路径:

请输入输出路径

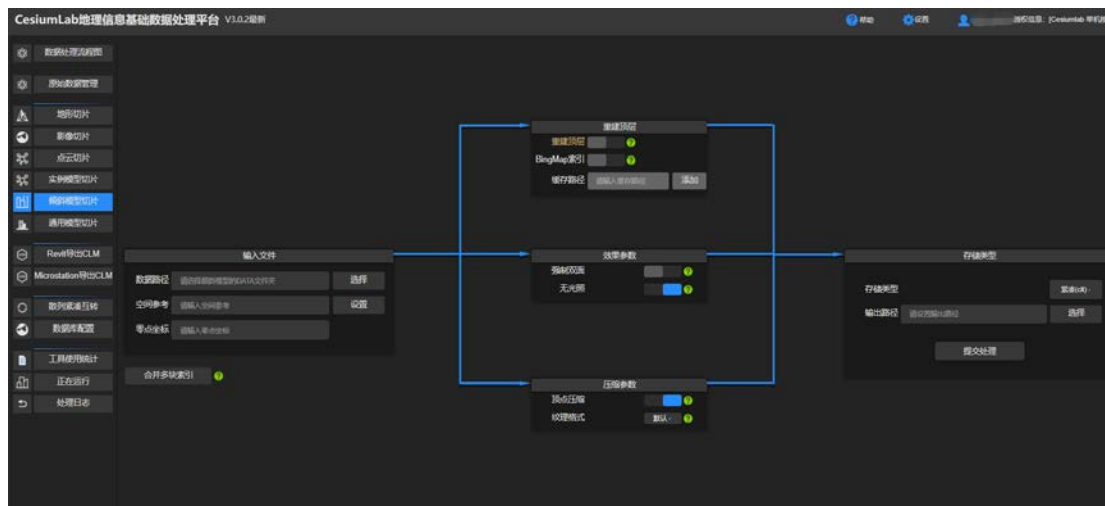
选择

提交处理

输出文件的选择在本文【数据存储】部分已经讨论过，这里注意点，对于实例模型切片来说，一般量很大，文件个数很多，我们强烈推荐选择紧凑存储方式。

2.7 倾斜模型切片

主站上依次点击【数据处理】、【倾斜模型切片】进入倾斜模型切片工具



2.7.1 输入文件



输入文件我们要求必须是带 lod 的 osgb 形式的数据，比如如下结构：

	名称	修改日期	类型	大小
dayanta				
Data				
Tile_+000_+000	Tile_+000_+004.osgb	2018/10/15 10:38	OSGB 文件	
Tile_+000_+001	Tile_+000_+004_L16_0.osgb	2018/10/15 10:38	OSGB 文件	
Tile_+000_+002	Tile_+000_+004_L17_00.osgb	2018/10/15 10:38	OSGB 文件	
Tile_+000_+003				
Tile_+000_+004				
Tile_+000_+005				
Tile_+000_+006				
Tile_+000_+007				
Tile_+000_+008				
Tile_+001_+000				
Tile_+001_+001				

单一 osgb 文件，或者非 osgb 格式的文件即便也是类似组织，也无法处理。

我们需要选择的是 Data 目录，选择之后，如果和 Data 平级的有 Metadata.xml 文件，我们会自动从该文件解析空间参考和零点坐标填写进来：

名称	修改日期	类型	大小
Data	2020/1/13 9:47	文件夹	
metadata.xml	2018/10/15 10:38	XML 文档	1 KB

```
D: > gisdata > modelos > dayanta > metadata.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <ModelMetadata version="1">
3      <!--Spatial Reference System-->
4      <SRS>ENU:34.2196,108.9594</SRS>
5      <!--Origin in Spatial Reference System-->
6      <SRSOrigin>0,0,0</SRSOrigin>
7  </ModelMetadata>
8
```

Metadata.xml 里 SRS 即空间参考。SRSOrigin 为零点坐标。

如果此信息未正确填写，那么你又明确知道数据的空间参考，可以点击【设置】按钮，弹出空间参考详细设置面板，关于此设置方法，请参考本文【通用功能】里的【空间参考】部分。

空间参考设置

空间参考:

EPSG:32648

经纬度局部坐标系 ?

EPSG编码搜索 ?

从地图上选择 ?

从文件导入 ?

零点坐标:

421447,3370768,0 ?

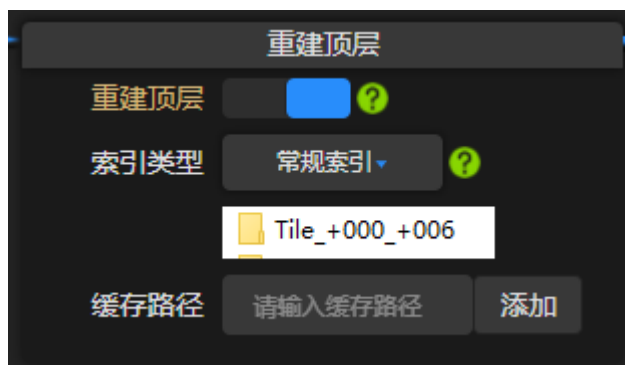
七参数:

请输入七参数，以逗号隔开 ?

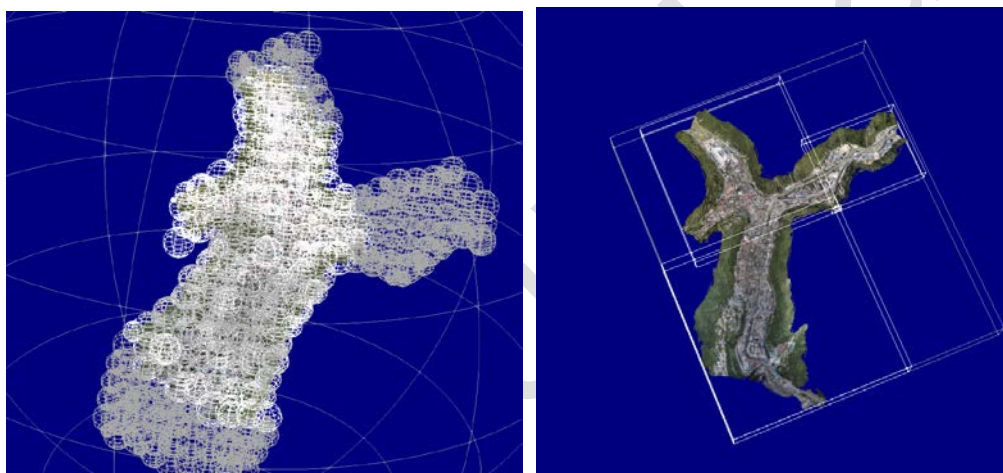
取消

确定

2.7.2 重建顶层



参考上述倾斜的组织方式，一个 Data 目录下的 Tile 可能会成千上万，非重建顶层，那么是下图左侧的效果（每个白色包围球代表了一个 tile，即一次请求）。重建顶层后如下图右侧（每个矩形包围盒代表一个 tile，即一次请求）很明显，请求数大量减少，加速了渲染过程。



2.7.2.1 支持重建顶层的索引形式

目前来说，由于倾斜生产软件过多，或者参数设置，倾斜数据的 Tile 索引有各种形式，目前我们支持的索引形式有如下两种，我们依据后面的数字序号来排列 Tile 的位置，所以请注意，不能手动去修改或者设置此序号，比如一些人为精修的数据设置的 Tile_99_99，这里设置的 99 只是为了目录不重名，形式上满足，但是不表示具体 Tile 位置，此类也是不符合重建顶层要求的。

Tile_+033_+035	2016/12/2 10:47	文件夹
Tile_+033_+036	2016/12/2 10:47	文件夹
Tile_+033_+037	2016/12/2 10:47	文件夹
Tile_+033_+038	2016/12/2 10:47	文件夹
Tile_+034_+034	2016/12/2 10:47	文件夹
Tile_+034_+035	2016/12/2 10:24	文件夹
Tile_+034_+036	2016/12/2 10:24	文件夹
Tile_+034_+037	2016/12/2 10:24	文件夹
Tile_+034_+038	2016/12/2 10:24	文件夹
Tile_+034_+039	2016/12/2 10:47	文件夹
Tile_+035_+033	2016/12/2 10:47	文件夹
Tile_+035_+034	2016/12/2 10:47	文件夹
Tile_+035_+035	2016/12/2 10:24	文件夹
Tile_+035_+036	2016/12/2 10:24	文件夹
Tile_+035_+037	2016/12/2 10:24	文件夹
Tile_+035_+038	2016/12/2 10:24	文件夹
Tile_+035_+039	2016/12/2 10:24	文件夹
Tile_+036_+031	2016/12/2 10:53	文件夹
Tile_+036_+032	2016/12/2 10:53	文件夹

默认形式的索引

Tile_132100133210312033	文件夹
Tile_132100133210312102	文件夹
Tile_132100133210312103	文件夹
Tile_132100133210312112	文件夹
Tile_132100133210312113	文件夹
Tile_132100133210312120	文件夹
Tile_132100133210312121	文件夹
Tile_132100133210312122	文件夹
Tile_132100133210312123	文件夹
Tile_132100133210312130	文件夹
Tile_132100133210312131	文件夹
Tile_132100133210312132	文件夹
Tile_132100133210312133	文件夹
Tile_132100133210312211	文件夹
Tile_132100133210312213	文件夹
Tile_132100133210312230	文件夹
Tile_132100133210312231	文件夹
Tile_132100133210312232	文件夹
Tile_132100133210312233	文件夹
Tile_132100133210312300	文件夹
Tile_132100133210312301	文件夹
Tile_132100133210312302	文件夹
Tile_132100133210312303	文件夹
Tile_132100133210312310	文件夹
Tile_132100133210312311	文件夹
Tile_132100133210312312	文件夹
Tile_132100133210312313	文件夹
Tile_132100133210312320	文件夹
Tile_132100133210312321	文件夹
Tile_132100133210312322	文件夹
Tile_132100133210312323	文件夹
Tile_132100133210312330	文件夹
Tile_132100133210312331	文件夹
Tile_132100133210312332	文件夹
Tile_132100133210312333	文件夹
Tile_132100133210313020	文件夹

Bingmap 型索引

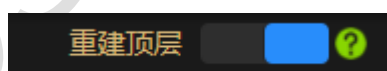
Bingmap 索引需要打开 Bingmap 索引开关。

2.7.2.2 不支持重建顶层的索引形式

只要不是上述提到的两种支持的形式，其他都是不支持的。为了更有对照性，罗列了一些：

		名称	修改日
Tile_1 Tile_2 Tile_3 Tile_4 Tile_5 Tile_6 Tile_7 Tile_8 Tile_9	2021/9/4 18:05	0_modelTile_0	2020/
		0_modelTile_1	2020/
		0_modelTile_2	2020/
		0_modelTile_3	2020/
		0_modelTile_4	2020/
		0_modelTile_5	2020/
		0_modelTile_6	2020/
		0_modelTile_7	2020/
		0_modelTile_8	2020/
		1_modelTile_0	2020/
3313.375-362.375 3313.375-362.500 3313.375-363.250 3313.500-361.750 3313.500-361.875 3313.500-362.000 3313.500-362.125 3313.500-362.250 3313.500-362.375 3313.500-362.375_1 3313.500-362.500 3313.500-362.500_1 3313.500-362.625 3313.500-363.250	2021/2/18 10:01	pyramid	
		Tile-0-0-1-1	
		Tile-0-1-1-1	
		Tile-0-2-1-1	
		Tile-0-3-1-1	
		Tile-0-4-1-1	
		Tile-0-5-1-1	
		Tile-0-6-1-1	
		Tile-1-0-1-1	
		Tile-1-1-1-1	
		Tile-1-2-1-1	
		Tile-1-3-1-1	
		Tile-1-4-1-1	
		Tile-1-5-1-1	
		Tile-1-6-1-1	
		Tile-2-0-1-1	
		BlockB	
		BlockY	
		metadata.xml	
		Model.osgb	

2.7.2.3 重建顶层开关



只有授权购买用户才可以开启该开关。

用户开启该开关，需要自行确认数据的索引形式符合本章的【支持重建顶层的索引形式】。

请注意，对于【不支持重建顶层的索引型式】的索引形式倾斜数据重建后，可能会导致很多浏览效果异常。

一般来说，数据不重建顶层也能浏览，尤其当 Tile 个数不多时候（小于 100）那么浏览效率并不低，也可以不重建顶层

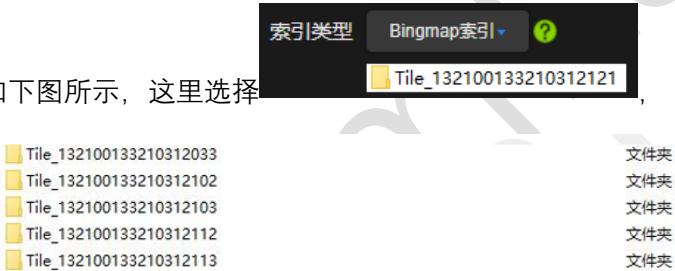
2.7.2.4 索引类型



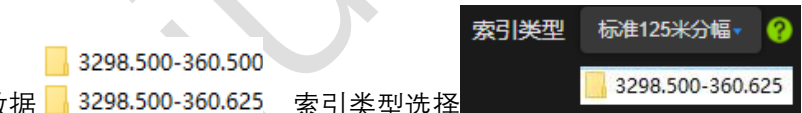
对于如下图默认形式的索引，索引类型选择



当 Tile 索引形式如下图所示，这里选择



对于标准化分幅数据 3298.500-360.500，索引类型选择



2.7.2.5 缓存路径

重建顶层使用了一些复杂的算法来根据用户数据生成粗精度数据，涉及到数据合并和分割，计算过程较慢，为了我们测试调试程序方便，设置了一个缓存路径，可以把重建的中间结果缓存到硬盘上，一般用户不需要设置该选项。除非您在处理的 Tile 个数多达上万甚至更多，那么可以设置下该选项，节省内存。

2.7.3 效果参数



2.7.3.1 强制双面

一般来说，强制双面开关无需开启，但是对一些异常数据，可能有正反面裁剪引起的漏洞之类的，可以尝试开启该选项，达到双面渲染的目的。需要具体分析，如果不确定可以联系我们来判定是否要开启该选项。

2.7.3.2 无光照

倾斜数据的来源是实景照片，无光照选项开启后，倾斜模型的渲染部分不参与 Cesium 引擎的默认光照计算，显示的更加真实一些。而且生成的 3dtiles 里不包含法向信息，节省了空间。一般来说无需关闭，但是一些引擎对于不含法向信息的模型渲染是黑色的，为了避免这个问题可以尝试关闭该选项。

2.7.4 压缩参数

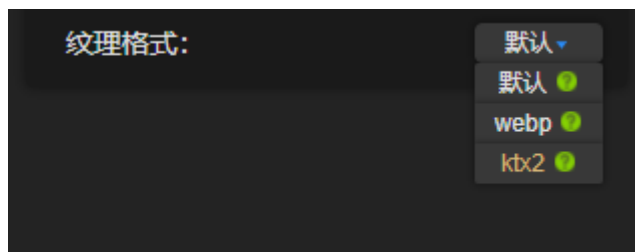


2.7.4.1 顶点压缩

顶点压缩采用了 Draco 压缩算法来减少顶点数据的存储量，Draco 是比较高效的一种压缩方式，目前我们内部的配置是顶点采用 14bit，纹理坐标采用 10bit，总计 24bit 存储一个顶点。未压缩的数据顶点是 96bit (3 个 float)，纹理坐标 64bit (2 个 float)，总计 160bit。

那么压缩率是： $24/160 = 3:20$ ，大约 $1/7$ ，这个压缩率很强大的，大量节省传输数据量，此部分我们建议开启。

2.7.4.2 纹理格式



选择【默认】使用 jpg 格式的纹理，也就是原始 osgb 里的格式。

选择【Webp】使用 webp 格式的纹理，webp 存储量大约是默认 jpg 的 70%，可以加快网络传输，但是它只能在 chrome 下使用，而且不会减少显存占用。

选择【ktx2】使用 ktx2 格式的纹理，存储量大约是默认 jpg 的 90%，可以固定减少显存消耗 $5/6$ 左右，比如一张 1024×1024 分辨率纹理，默认 jpg 在显存里实际占用是固定 3MB 显存，处理成 ktx2 只占 $3MB / 6 = 512KB$ ，这个对于显卡来说极度友好。不过目前处理过程还相对缓慢。

目前 ktx2 需要付费才可以选择。

我们建议对陌生数据分两次处理，第一次默认纹理处理一遍，检查有无效果异常。如果没问题，第二次再进行 ktx2 压缩处理提升显示效率。

2.7.5 存储类型



存储类型的选择在本文【数据存储】部分已经讨论过，不再赘述。

2.7.6 倾斜处理时间

处理过程的时间问题

处理过程分两个大阶段

A, 原始 osgb 优化阶段, 默认参数下, 一般这个阶段 15G 每小时

如果选了 ktx2 格式压缩, 那么这块可能是 0.5G / 每小时

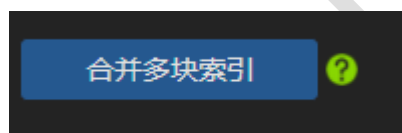
B, 顶层重建过程, 这个时间消耗, 是根据 Tile 的个数多少来定的

需要重建的次数大约 = Tile 的个数 * 0.6

单次重建的次数大约是 0.5 分钟

比如一个 Data 下有 1000 块, 那么重建顶层的时间消耗 = $1000 \times 0.6 \times 0.5 = 300$ 分钟, 也就是 5 个小时

2.7.7 合并多块索引



常见的需求是用户有很多块数据 (一个 Data 算一个块), 最终要求作为一个统一图层来发布, 那么可以根据下述步骤来进行:

第一步, 使用倾斜模型切片, 对一个 Data 使用倾斜模型切片得到一个 3dtiles

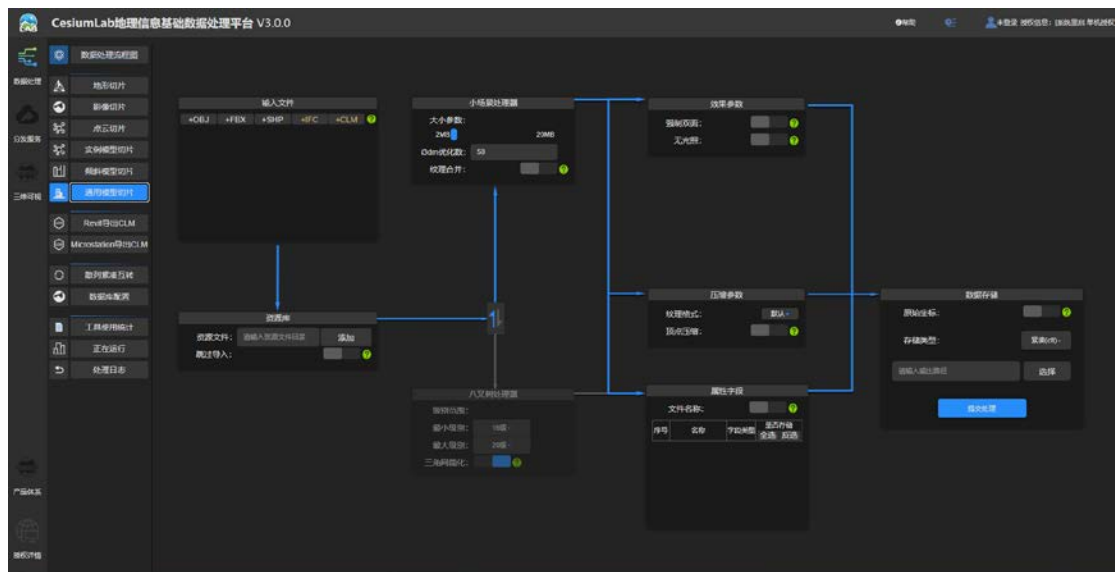
第二步, 使用索引合并工具对 你项目里需要一次性引入的, 并且是第一步得到的多个 3dtiles, 放在同一级目录下。点击【合并 3dtiles】按钮, 得到一个总览的 tileset.json 文件, 并在你项目里引用这个总览的 tileset.json

第二步并没有做任何数据上的额外优化, 仅仅是把多个倾斜索引合在一起, 这么做也是有意义的, 因为 Cesium 的 3dtiles 最大缓存 (maximumScreenSpaceError) 是对 每个 tileset 设置的。这种引用一个总览 tileset.json, 比引用多个 tileset.json 来说, 缓存的调度合适一些, 也可以加速淘汰看不见的 倾斜块。

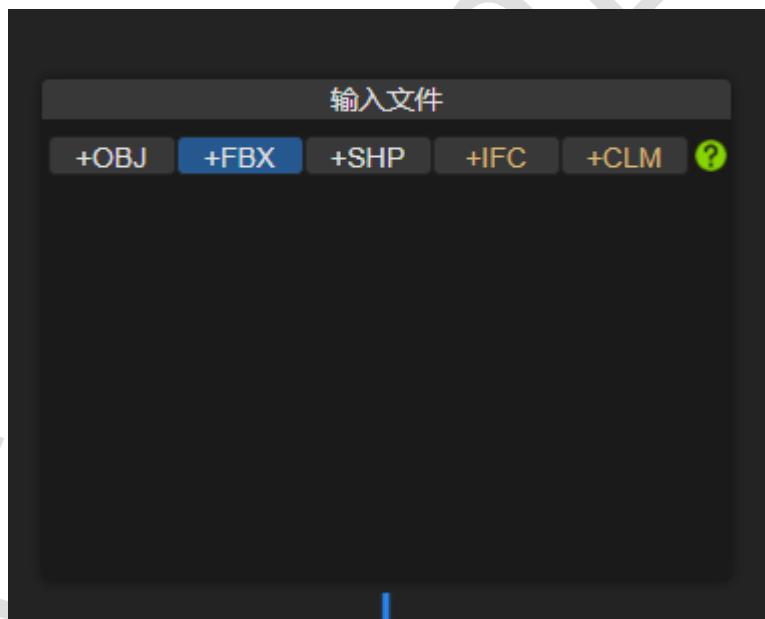
注意此功能仅支持合并【倾斜模型切片】输出的 3dtiles, 对其他工具输出的 3dtiles 不一定适用。

2.8 通用模型切片

主页上依次点击【数据处理】、【通用模型切片】进入通用模型切片工具。



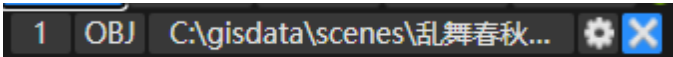
2.8.1 输入文件



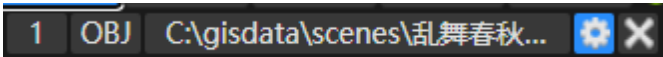
2.8.1.1 添加输入

点击 **+OBJ** **+FBX** **+SHP** **+IFC** **+CLM**，弹出选择输入文件的对话框，这里需要注意的，如果是一次选择了多个输入文件，在列表只会显示一条，但是实际这些文件都会被处理。这样是为了方便统一设置参数，和前面的地形、影像、点云的多选逻辑稍有不同。

2.8.1.2 删除输入

点击  每条后的【X】删除该条

2.8.1.3 设置参数

点击  每条后的【⚙️】设置该输入的参数，具体参数面板，根据输入类型不同而不同。详见下面章节。

2.8.2 Obj 参数设置



2.8.2.1 空间参考

空间参考的概念以及基本设置方法参考本文【通用功能】里的【空间参考】部分，对于通用模型，大部分的数据都没有自带空间参考，为了便于处理，所以我们使用了一个默认值：ENU:39.90691,116.39123 此位置在天安门附近

模型坐标的 (0,0,0)点会放置到该位置，如果你的模型坐标非常大，距离零点很远，可能漂浮在半空中。默认坐标仅仅是为了测试，需要根据你数据具体设置。

2.8.2.2 属性文件

通用模型处理处理的数据，都会默认带两个属性：id, name

对于不同的数据 id 和 name 来源不同。

Obj: name 来自 node name, id 是随机生成

Fbx: name 来自 node name, id 是根据 fbxsdk 的 uniqueid 计算 md5 得到。

Clm: name 来自 构件名称, id 根据导出插件, revit 来自构件的 uuid, microstation 来自构件 id。

Ifc: name 来自构件名称, id 是随机生成。

Shp: name 来自 name 字段, 如果没有来自 FID, id 是来自 FID。

除了这两个属性之外，在本章【属性字段】部分可以选择是否携带可选属性。对于 obj、fbx 等模型来说，由于本身不携带属性数据，我们支持通过属性文件来外挂属性字段。对于属性文件的要求：

- (1) 必须是 csv 格式，utf8 编码，注意 csv 是通过英文逗号分割的。
- (2) 第一列必须是【唯一名称】。
- (3) 第一列的值和模型场景里的名称关联匹配。

	A	B	C	D
1	唯一名称	地层代码	岩性	mpPerimete
2	WF_DZfz_M_sutiantu_jz001	1	杂填土	6282.657165

2.8.2.3 透明模式

用户可以强制设置透贴的形式，可以指定是 mask 或者 blend。

【Mask】：适合树叶类的透明。

【blend】：适合玻璃类透明。

【自动】：表示让程序自动判断。如果你透明通道数据里的 alpha 值，除了 0，就是 255，那就用 mask，否则就用 blend。

2.8.2.4 强制双面

由于数据制作或者转格式中的一些缺陷，一些三角面绕向是反向的。或者还有一些类似护栏、纱网等对象，虽然在 3dmax 里正面背面观察都可以，但是导出后可能只在一个面能观察到，为了避免此种问题。这里可以开启 ，不过该选项会导致一定的渲染性能损失。

2.8.2.5 无光照

对于烘焙后的 3dsmax 数据，光照阴影已经烘焙到贴图上了，无需通过 Cesium 的渲染光照计算或者说也不能通过 Cesium 的光照计算，那么开启此选项后，保证模型颜色和烘焙效果一致。

2.8.2.6 翻转 YZ

Obj 没有信息存储向上轴的问题, 比如 gltf 是 y 向上, 但是很多三维引擎尤其地理引擎都是 z 向上, 我们默认认为 obj 也是 z 向上。这个开关需要和你 3dsmax 导出时候的选项一致, 向上轴就不会错。

2.8.2.7 gbk 编码

Obj 是个纯文本文件, 所有文本文件都涉及到文字编码问题, 如果 obj 内容都是纯英文, 这里可以忽略。如果处理结果点查属性, 发现有乱码, 那么这个开关切片下再尝试。默认 3dsmax 导出的 obj 都是 gbk 编码的。

2.8.2.8 翻转绕向

Obj 格式简单, 但是来源多种, 一般来说 3dsmax 导出的数据没有绕向问题, 但是发现有背面裁剪情况, 尝试开启此选项, 重新处理。

2.8.3 Fbx 参数设置



2.8.3.1 空间参考

详见 Obj 空间参考设置。

2.8.3.2 属性文件

详见 Obj 属性文件设置。

2.8.3.3 透明模式

详见 Obj 透明模式设置。

2.8.3.4 强制双面

详见 Obj 强制双面参数说明。

2.8.3.5 无光照

详见 Obj 无光照参数说明。

2.8.4 Shp 参数设置



2.8.4.1 空间参考

一般来说 shp 的空间参考信息完备，这里不需要设置。如果缺失，请根据本文【通用功能】
【空间参考】设置。

2.8.4.2 造型参数

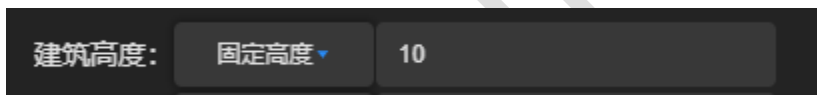


造型参数，处理程序根据 shp 的轮廓信息来生成三维几何体需要的。各项参数详述：

2.8.4.3 造型参数.建筑高度

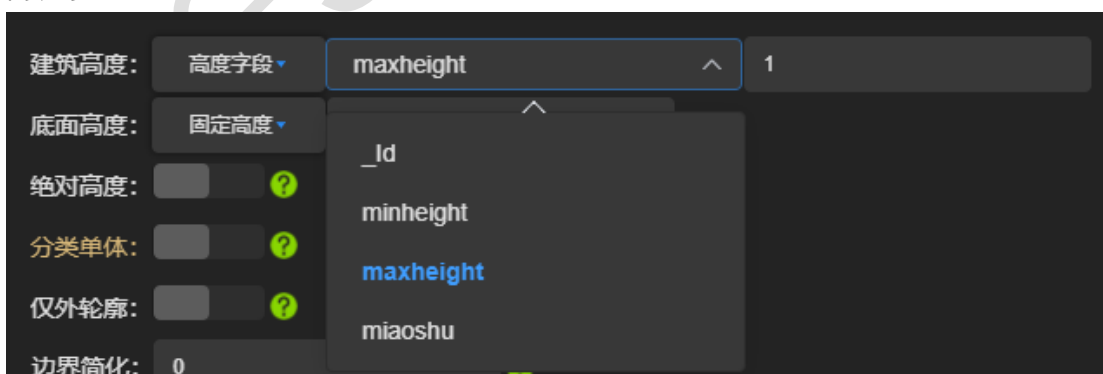
建筑高度有两种方式：

1， 固定高度方式：



根据给定值来设置建筑模型的高度。

2， 高度字段方式：



建筑高度 = 选择的字段值 * 设定的倍率

比如 shp 里有楼层个数字段 floor，那么这里可以选择 floor，倍率设置 3，表示一层三米，总高度 = floor 的值 * 3

2.8.4.4 造型参数.底面高度

和【建筑高度】字段类似，固定或者字段两种方式来设置建筑底面的高度。设置不赘述，下一个参数会统一说明建筑的高度计算方式

2.8.4.5 造型参数.绝对高度

如果【绝对高度】开关关闭，那么

建筑下底高 = 【底面高度】 + 【地面中心点的地形高程】

建筑上顶高 = 建筑下底高 + 【建筑高度】

如果【绝对高度】开关打开，那么

建筑下底高 = 【底面高度】

建筑上顶高 = 【建筑高度】

2.8.4.6 造型参数.分类单体

【分类单体】打开之后，生成的 3dtiles 比较特殊，是一种支持 Cesium 分类渲染的几何体，可以用来做倾斜单体化，具体见这篇文章 <https://bbs.cesiumlab.com/thread?topicId=22>

2.8.4.7 造型参数.仅外轮廓

建筑矢量面可能是带孔洞的复杂 Polygon，如果此开关打开，那么会忽略内部孔洞，只按外轮廓生成建筑几何体。

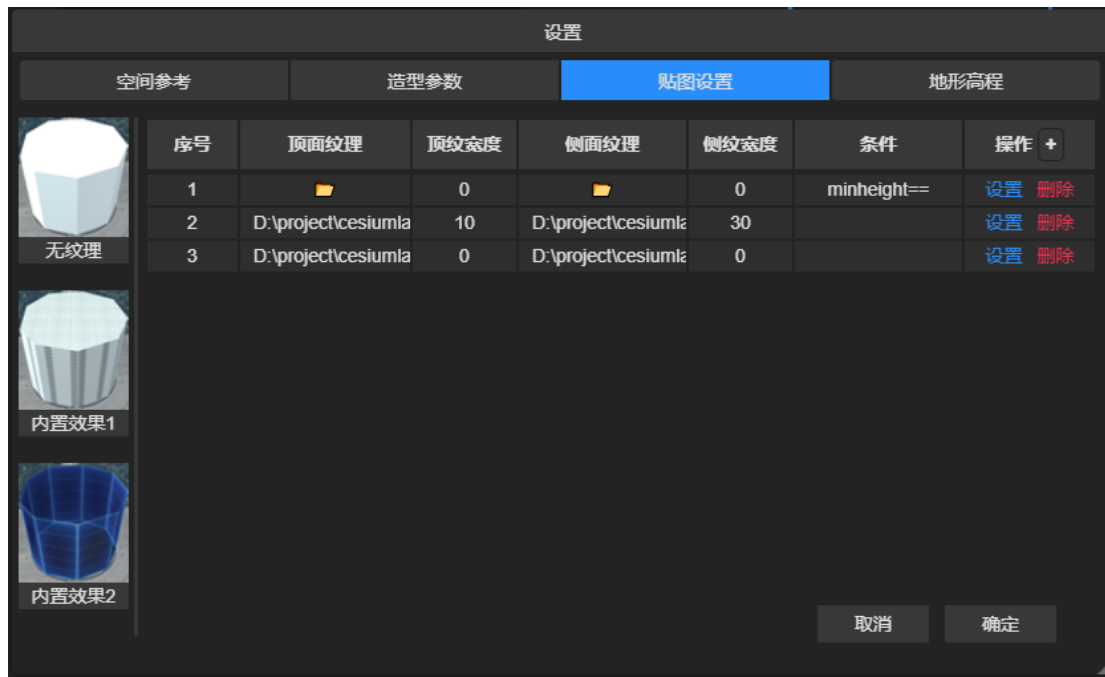
2.8.4.8 造型参数.边界简化

一般来说，建筑矢量面的外轮廓边界并不复杂，我们不需要开启此参数（保持默认值 0），但是如果有一些非建筑轮廓的矢量面，希望也能拉体，如果边界过于复杂，可以常设设置此参数来简化边界，比如设置了 1，那就表示以 1 米误差来简化边界。这种使用是非常规方式，请自行测试。

2.8.4.9 造型参数.缓冲范围

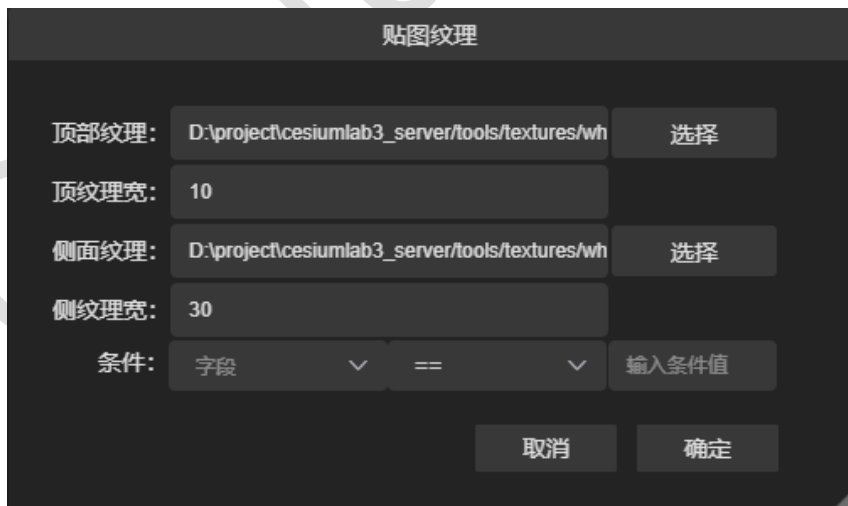
这个主要需求在于，很多单体化矢量是基于正射来勾勒得单体化轮廓，但是如果此轮廓勾勒得过于紧密，单体化不足以完全包裹倾斜得模型范围，导致部分裸漏，效果不好，可以尝试设置缓冲范围，我们对矢量轮廓向外做一个缓冲区，然后再进行生成几何体操作

2.8.4.10 贴图设置



对于 shp 生成的白模，我们可以依据条件来进行模型贴图。条件是一个列表，可以设置多个。

点击操作里 **操作 +** 弹出添加条件界面：



2.8.4.11 贴图设置.顶部纹理

顶部纹理：需要设置一个 jpg 图片的路径，可以点击【选择】按钮来选择该路径。

2.8.4.12 贴图设置.顶纹理宽

顶纹理宽：默认为 0，表示一张纹理覆盖整个顶面。如果不为 0，比如设置了 1，表示这张纹理每隔 1 米重复一次，那处理程序会计算重复次数，设置每个顶点的 UV 坐标。

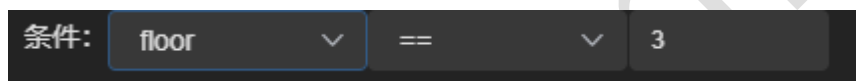
2.8.4.13 贴图设置.侧面纹理

侧面纹理：需要设置一个 jpg 图片的路径，可以点击【选择】按钮来选择该路径。

2.8.4.14 贴图设置.侧纹理宽

侧纹理宽：默认是 0，表示一张纹理图片贴满整个侧面。如果不是 0，比如设置 1，表示这张贴图每隔 1 米重复一次，那处理程序会计算重复次数，设置每个顶点的 UV 坐标。

2.8.4.15 贴图设置.条件



需要选择一个【字段】，选择【操作符】，输入【条件值】

比如 字段选择的“floor”，操作符“==”，条件是“3”

那么表示这个贴图条件只作用在 floor==3 的建筑物上，其他条件类似。

多个条件的判定逻辑：

对每个矢量对象，所有条件按照添加顺序顺次判定，只要对象满足了某个条件，立即采用该条件的贴图设置，不再做后续判定。

2.8.4.16 贴图设置.内置贴图



我们自带了三种贴图效果，单击该图片，会添加一个配置条件，内置的贴图已经配置好了，只需要设置下条件即可。

2.8.4.17 地形高程

添加紧凑高程切片		添加散列高程切片
序号	路径	操作
1	D:\testdata\dem\tiles	×
2	C:\gisdata\dem\12.pak	×

地表高程我们支持从地形切片的输出数据来采样，而且支持多个地形数据。
点击【添加紧凑高程切片】选择 pak 地形切片，会列在此表格里。
点击【添加散列高程切片】选择散列地形切片，散列地形会列在此表格里。
处理程序会按照列表顺序来依次采样，只要正确采样到对应点位高度，就停止采样下一个。
建筑的高程，我们只采样轮廓的中心点位置。

2.8.5 Ifc 参数设置



2.8.5.1 空间参考

详见 Obj 空间参考设置。

2.8.6 Clm 参数设置

2.8.6.1 Clm 格式定义

Clm (CesiumLab Model 简称 clm) 格式是 CesiumLab 自定义的 BIM 数据中间格式，基于 sqlite 文件数据库，单文件涵盖了可视化需要的所有 BIM 数据信息，包括属性、几何体、构件、纹理、结构等等。总共包括下述 6 个表。

1. Models

定义了模型信息表

字段名	类型	说明
id	Text	必须，唯一性标识符，sqlite 下使用 guid 生成
path	Text	必须，源 bim 数据路径
date	Date	必须，模型信息导出完成时间
elementcount	Int	必须，模型中的构件个数
geometrycount	Int	必须，模型中的几何体个数
Vertexcount	Int	必须，模型中顶点个数
trianglecount	Int	必须，模型中三角形个数
propertycount	Int	必须，模型中属性个数

rowid	id	path	elementcount	geometrycount	propertycount	vertexcount	trianglecount	date
Click here to define a filter								
1	461d8ea9-4a4a-472a-b2f0-d364af5bba2c	D:\gisdata\bim\bentley\01 道路.dgn	933	933	151	1379298	1210895	2018-12-16 04:55:58

2. Properties

定义了模型属性表

字段名	类型	说明
id	Text	必须，唯一性标识符，sqlite 下使用 guid 生成
name	Text	必须，字段名称
type	Text	必须，字段类型，string、integer、double、float、bool
group	Text	可选，属性分组
unitttype	Text	可选，单位类型，可选 length，volume 等

rowid	id	name	type	group	unitttype
Click here to define a filter					
1	e4974852-dc7c-48c4-b328-73f3709de58a	Is3dElement	boolean		
2	b6e09256-0dd5-48da-b754-b9d212f103d7	IsGraphicalElement	boolean		
3	24265588-2cb2-4a35-8e94-e5e8230b0f25	ElementDescription	string	General	
4	b5d81ad9-14a4-4775-92b8-0f8a3faee774	Level	int	General	
5	89caacb6-ff84-451f-adee-ed93c6890b23	ElementID	long	RawData	
6	8a570905-934f-4f32-896d-3d0d93249a2d	Model	string	Extended	
7	fc50454-73a1-4423-b0f3-f19e1b7ef910	ModifiedTime	string	Extended	
8	9d763430-ea8b-42d6-8a86-d91d185f696f	ElementSize	int	RawData	
9	7f42de9b-5b0d-420f-87bd-aa8a92a70cb2	Linkages	int	RawData	
10	99d2f682-eeb-446a-85a8-64c0d0dc64d4	XAttributes	int	RawData	
11	731e4f55-6217-4695-9f6f-5048487f7349	Links	string	PropertyCategory	
12	0d41aa9a-4671-447c-a911-9ac65f06c139	DwgCustomObjectPropertyList	string	PropertyCategory	
13	998300e8-7944-492a-9c9a-d8df0a853391	ElementType	string	PropertyCategory	
14	25fa907a-ce80-45a3-a1d8-aa704e9a5df7	Color	int	General	

3. Geometries

定义了模型的几何信息

字段名	类型	说明
id	Text	必须，唯一性标识符，sqlite 下使用 guid 生成
gltf	Blob	必须，几何体信息，使用 glb2.0 格式存储，gltf 和 buffer 存在一起。并用 zip 压缩。glb 中纹理采用 url 形式，为 modelresources 表的 id。
lod	Int	可选，表示几何体的 lod 级别
vertexcount	Int	几何体顶点个数
trianglecount	Int	三角形个数
x	Double	包围球中心点 x 值
y	Double	包围球中心点 y 值
z	Double	包围球中心点 z 值
r	Double	包围球半径

rowid	id	gltf	lod	vertexcount	trianglecount	x	y	z	r
Click here to define a filter									
1	eb8b9ca-f0e32-44ad-a975-5d7070807394	(blob)	0	175	173	50.6554358680947	61.5281180622533	0.00316617286446516	0.00899493648698688
2	4fe1e5e0-8e35-4895-8ca6-c55a1b0114dd	(blob)	0	582	580	50.6217687788736	61.5027201630586	0.003188	0.0384345591434692
3	2bb759a3-3985-4b7c-b26c-aa0d3d7c93e2	(blob)	0	508	506	50.6856989315907	61.5508641209952	0.0031953	0.0338569281935578
4	2e72af83-bc25-4841-80f0-9f9a61634fc5	(blob)	0	412	190	50.6536.292579723	61.5244.923086458	31.6450220407391	21.4264428304206
5	e9892cc1-69d2-4e90-bda2-a73a1a9a46de	(blob)	0	3456	2304	50.6536.292485947	61.5244.923303052	31.5996773734743	21.4290166754974
6	64acc906-baee-4171-b5bc-844402651e82	(blob)	0	346	155	50.6516.966079042	61.5277.763571259	32.0545977411157	18.7832989252996
7	77fe755d-0f98-4108-8a80-c0514f192c54	(blob)	0	377	183	50.6573.473334603	61.5317.499944447	32.0640519442309	19.204527911255
8	1537716e-a6b8-4546-b6d5-17ab011a99d0	(blob)	0	291	166	50.6592.640137203	61.5286.526236826	31.7547290585388	17.026440542446
9	b0b84759-de70-4b8c-b0ca-f86daf20db21	(blob)	0	956	954	50.6694328787453	61.5037757837122	0.00250876642697894	0.0251090240615955
10	2cbe675b-372a-49e9-a9fc-de4e53f425e4	(blob)	0	298	296	50.6462278502473	61.5402778485856	0.0031481891899653	0.0130028008170381
11	a048e752-d673-48fe-b852-8bed15676571	(blob)	0	2826	1884	50.6516.966225375	61.5277.76345121	32.0303854144687	18.7857847679646
12	259a1947-8df9-4339-9e81-123016449a45	(blob)	0	3042	2028	50.6573.473420684	61.5317.499863175	32.0510969612731	19.2068525185891
13	032df751-fb0d-40da-8935-71b3af07f413	(blob)	0	2772	1848	50.6592.640109951	61.5286.52634097	31.7146853751009	17.0292301259807
14	b2eabc59-7d8e-47f7-83bd-5d2f2789ce3e	(blob)	0	84	82	50.657347220942	61.5317500152922	0.00322760969612731	0.00189334748040491
15	48c1b9be-359f-4fc9-a949-fc1f2c94027e	(blob)	0	4656	2324	50.6217.687788736	61.5027.201630586	31.705	384.345644999764
16	45f10036-2b8b-49e6-9a8f-553a9ec40a62	(blob)	0	1400	696	50.6554.358680947	61.5281.180622533	31.4867286446516	89.9498369436838
17	a6f62ab7-c510-48b8-9e03-eb4b990c0eca	(blob)	0	7648	3820	50.6694.328787453	61.5037.757837122	24.9126642697894	251.092103821028
18	84f851c9-7779-4030-adbb-df13a25b6c87	(blob)	0	7648	3820	50.6694.328787453	61.5037.757837122	24.7626642697894	251.092103821028
19	d7fb4117-0731-4dfa-a82a-1c3f5cc5c17e	(blob)	0	4064	2028	50.6856.989315907	61.5508.641209952	31.778	338.569332123868
20	75f6e2d3-f86f-4daa-85c0-a9c0b8623afc	(blob)	0	2384	1188	50.6462.278502473	61.5402.778485856	31.156891899653	130.028438209404
21	ca6b0a56-efa9-415e-8d35-7840e73cb89f	(blob)	0	1004	656	50.6541.769877049	61.5223.423835338	31.3636080723133	23.224212891846
22	ffc13bdb-2597-4d61-99ae-f2780e631eb70	(blob)	0	110	108	50.6540.999872181	61.5218.234006557	30.0886080723133	18.938762870696
23	c3a15f12-5bce-4936-b33e-62e596b612ec	(blob)	0	5251	3488	50.6232.383759463	61.5007.113153446	31.618	360.630088622938
24	9b7d70ea-86b9-4586-beea-3b0132c2f23c	(blob)	0	582	580	50.6236.81997612	61.5001.252917665	30.343	361.980759991237
25	b4d41b70-aa0b-4b11-073b-767f48a7b124	(blob)	0	12308	8102	50.66671.80555087	61.5010.674734738	24.0852151306036	224.325685100338

4. Elements

定义了 BIM 构件属性

字段名	类型	说明
id	Text	必须，唯一性标识符，目前 id 虽然是字符串形式，但是必须是整形格式。
name	Text	必须，构件名称
modelid	Text	必须，构件属于哪个模型
properties	Blob	必须，构件的属性信息，如果是 sqlite，这块直接存入 json 文本，并使用 zip 压缩。
geometries	Blob	必须，构件的几何体信息，如果是 sqlite，存入 json 文本。
geometrysize	Int	所有几何体的 glb 未压缩字节大小之和
vertexcount	Int	顶点个数
trianglecount	Int	三角形个数
x	Double	包围球中心点 x 值，注意这里因为 element 对 geometry 的引用是有偏移矩阵的，所以这里的包围球是几何体坐标乘以偏移矩阵得到的。
y	Double	包围球中心点 y 值
z	Double	包围球中心点 z 值
r	Double	包围球半径

Properties 的 json 部分示例如下：

```
"props": [
  {
    "index": 0,
    "value": 852.50170500341039
  },
  {
    "index": 1,
    "value": 279.69216043419016
  },
  {
    "index": 2,
    "value": 0
  },
  {
    "index": 3,
    "value": 0.0
  },
  {
    "index": 4,
    "value": 0.0
  },
  {
    "index": 5,
    "value": 0
  },
  {
    "index": 6
```

该信息为一个数组。“index”为当前 BIM 模型的属性列表里序号。Value 为当前字段的值。

Geometries 的部分，参考如下：

```
Geometries:[{
  "id":"aaaaaaaaaaaaa",
  "matrix":[0,...]
},{
  "id":"aaaaaaaaaaaaa",
  "matrix":[0,...]
}]
```

该信息也是一个数组，gid 表示引用 geometries 库下的几何体 id，matrix 表示偏移矩阵，默认为单位矩阵。

rowid	id	name	properties	geometries	geometriesize	vertexcount	trianglecount	x	y	z	r
Click here to define a filter											
1	6868	Design_Triangles	(blob)	(blob)	6378	175	173	506554.358680947	615281.180622533	31.6617286446516	89.9493648698644
2	6886	Design_Triangles1	(blob)	(blob)	18588	582	580	506217.687788736	615027.201630586	31.88	384.345591434694
3	6905	Design_Triangles2	(blob)	(blob)	16352	508	506	506856.989315907	615508.641209952	31.953	338.569281935562
4	6918		(blob)	(blob)	12184	412	190	506536.292579723	615244.923086458	31.6450220407391	21.4264428304206
5	7696		(blob)	(blob)	97892	3456	2304	506536.292485947	615244.923303052	31.5996773734743	21.4290166754974
6	7716		(blob)	(blob)	10382	346	155	506516.966079042	615277.763571259	32.0545977411157	18.7832989252996
7	7728		(blob)	(blob)	11294	377	183	506573.473334603	615317.499944447	32.0640519442309	19.204527911255
8	7740		(blob)	(blob)	9128	291	166	506592.640137203	615286.526236826	31.7547290585388	17.026440542446
9	7793	Design_Triangles3	(blob)	(blob)	29816	956	954	506694.328787453	615037.757837122	25.0876642697894	251.09024061597
10	7824	Design_Triangles4	(blob)	(blob)	10068	298	296	506462.278502473	615402.778485856	31.481891899653	130.028008170362
11	7901		(blob)	(blob)	80252	2826	1884	506516.966225375	615277.76345121	32.0303854144687	18.7857847679646
12	7985		(blob)	(blob)	86292	3042	2028	506573.473420684	615317.499863175	32.0510969612731	19.2068525185891
13	8073		(blob)	(blob)	78740	2772	1848	506592.640109951	615286.52634097	31.7146853751009	17.0292301259807
14	9590	Existing_Triangles2	(blob)	(blob)	3656	84	82	506573.47220942	615317.500152922	32.2760969612731	18.9334748040277
15	9658		(blob)	(blob)	126876	4656	2324	506217.687788736	615027.201630586	31.705	384.345644999764
16	9692		(blob)	(blob)	38924	1400	696	506554.358680947	615281.180622533	31.4867286446516	89.949369436838
17	9711		(blob)	(blob)	207652	7648	3820	506694.328787453	615037.757837122	24.9126642697894	251.092103821028
18	9717		(blob)	(blob)	207652	7648	3820	506694.328787453	615037.757837122	24.7626642697894	251.092103821028
19	9730		(blob)	(blob)	110884	4064	2028	506856.989315907	615508.641209952	31.778	338.569331223868
20	9755		(blob)	(blob)	65524	2384	1188	506462.278502473	615402.778485856	31.156891899653	130.028438209404
21	9882		(blob)	(blob)	29180	1004	656	506541.769877049	615223.423835338	31.3636080723133	23.224212891846
22	9894		(blob)	(blob)	4444	110	108	506540.999872181	615218.234006557	30.0886080723133	18.938762870696

5. modelresources

定义了几何体 gltf 中引用的外部资源，一般都是纹理图片

字段名	类型	说明
id	Text	必须，唯一性标识符，sqlite 下使用 guid 生成
name	Text	必须，资源名称
data	Blob	必须，资源的二进制数据

6. Scenetree

定义了模型的结构树中的节点 (node)

字段名	类型	说明
id	Text	必须，唯一性标识符，sqlite 下使用 guid 生成，节点如果是构件，那么必须和构件 id 相同
name	Text	必须，节点名称
type	Text	可选，节点类型，对于构件，必须是 element
props	Text	可选，JSON 格式，附加到本 node 的额外属性，比如
parent	Text	可选，节点的父节点，为空表示根节点

rowid	id	name	type	props	parent
Click here to define a filter					
1	461d8ea9-4a4a-472a-b2f0-d364af5bba2c		model		
2	1	Terrain_Triangle	level	('ByCell':false,'Color':64,'Description':'Prop Triangles','Display':true,	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
3	2	Terrain_Contour_Minor	level	('ByCell':false,'Color':36,'Description':'Prop Minor Contours','Displ	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
4	3	Terrain_Contour_Major	level	('ByCell':false,'Color':10,'Description':'Prop Major Contours','Displ	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
5	4	Terrain_Triangle_Vertices	level	('ByCell':false,'Color':48,'Description':'','Display':true,'DisplayPriori	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
6	5	Terrain_Points_Random	level	('ByCell':false,'Color':67,'Description':'DTM-Random Points','Displ	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
7	6	Terrain_Flow_Arrows	level	('ByCell':false,'Color':7,'Description':'','Display':true,'DisplayPriority	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
8	7	Terrain_Points_Low	level	('ByCell':false,'Color':0,'Description':'','Display':true,'DisplayPriority	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
9	8	Terrain_Points_High	level	('ByCell':false,'Color':0,'Description':'','Display':true,'DisplayPriority	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
10	9	Terrain_Breakline	level	('ByCell':false,'Color':32,'Description':'DTM-Breaklines','Display':trn	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
11	10	Terrain_Hole	level	('ByCell':false,'Color':99,'Description':'','Display':true,'DisplayPriori	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
12	11	Terrain_Island	level	('ByCell':false,'Color':97,'Description':'','Display':true,'DisplayPriori	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
13	12	Terrain_Exterior	level	('ByCell':false,'Color':53,'Description':'DTM-Exterior Boundary','Dis	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
14	13	Terrain_Contour_Imported	level	('ByCell':false,'Color':80,'Description':'','Display':true,'DisplayPriori	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
15	14	Terrain_Void	level	('ByCell':false,'Color':49,'Description':'DTM-Void or Interior Bound	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
16	15	0	level	('ByCell':true,'Color':0,'Description':'','Display':true,'DisplayPriority	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
17	16	Geom_Centerline	level	('ByCell':false,'Color':3,'Description':'Prop Centerline','Display':true	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
18	17	Geom_Text	level	('ByCell':false,'Color':0,'Description':'Prop Horizontal Alignment Te	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
19	18	Geom_Control	level	('ByCell':false,'Color':36,'Description':'Survey Control','Display':trux	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
20	19	_CIVIL_CONSTRUCTION	level	('ByCell':false,'Color':0,'Description':'','Display':true,'DisplayPriority	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
21	20	道路偏置线	level	('ByCell':false,'Color':1541,'Description':'','Display':true,'DisplayPri	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
22	21	边线	level	('ByCell':false,'Color':2,'Description':'','Display':true,'DisplayPriority	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
23	22	渠化岛	level	('ByCell':false,'Color':7,'Description':'','Display':true,'DisplayPriority	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
24	23	下部结构	level	('ByCell':false,'Color':7,'Description':'','Display':true,'DisplayPriority	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
25	24	点划线	level	('ByCell':false,'Color':5,'Description':'','Display':true,'DisplayPriority	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
26	25	桥梁布置	level	('ByCell':false,'Color':2,'Description':'','Display':true,'DisplayPriority	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
27	26	Terrain_Design	level	('ByCell':false,'Color':55,'Description':'Prop Ground','Display':true,	461d8ea9-4a4a-472a-b2f0-d364af5bba2c
28	6868	6868	element		26
29	6886	6886	element		26
30	6905	6905	element		26
31	7793	7793	element		26

2.8.6.2 空间参考

设置

空间参考:

ENU:39.90691,116.39123

经纬度局部坐标系

EPSG编码搜索

从地图上选择

从文件导入

零点坐标:

0,0,0

七参数:

请输入七参数,以逗号隔开

取消

确定

详见 Obj 空间参考设置。

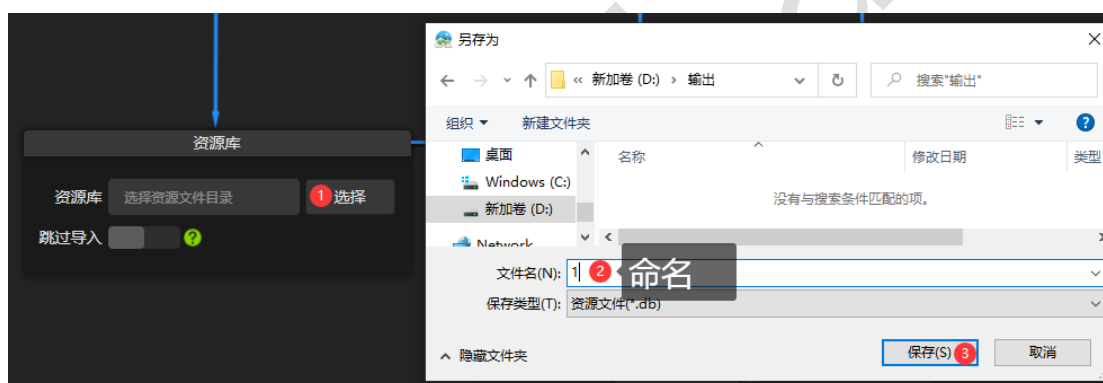
2.8.7 资源库



通用模型处理分为两个过程：数据导入 和 模型切片

数据导入过程会让各种格式的数据归一化，模型切片过程不再考虑格式的差异。

2.8.7.1 资源文件

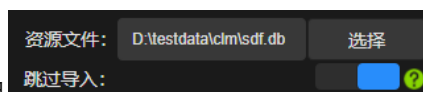


资源文件如果没有设置，我们会在内存中创建一个资源文件。如果设置了本地一个文件，那么会创建在本地。

设置本地资源路径的两种可能：1，数据量过大，内存不够用。2，为了方便进行【跳过导入】的调试。

2.8.7.2 跳过导入

模型切片过程是个慢过程，数据容易处理失败，为了方便调试，我们可以直接选择一个



已经存在硬盘上的资源文件，并且，那么就跳过数据导入过程，直接进行模型切片过程。

2.8.8 小场景处理器

根据数据特点和用户可视化需求，我们把通用模型处理分为两种处理策略，一种是小场景，一种是八叉树。

界面上如下选择：



小场景处理器不对输入数据做任何简化和缩放，直接使用的原始精度。但是会做合并、压缩（注意简化和压缩的区别，简化是生成粗数据，顶点个数可能会改变，压缩是减少硬盘、内存、显存的消耗，顶点个数不变）、分割等。在保证显示质量不损失的前提下，最大化提升数据加载效率。

2.8.8.1 大小参数

小参数控制产生的最小切片数据量，比如默认是 2MB，比如我们处理过程中有一些 b3dm 或者 i3dm 小于 2MB，我们会进行合并生成 cmpt，这么做目的是减少请求数量。

大参数控制产生的最大切片数据量，比如默认是 20MB，我们处理过程中如果有一些 b3dm 超过 20MB，我们会尝试进行分割。需要注意，这个分割是在空间进行的，我们碰到过一些特例，一个肉眼不可见的小构件竟然有几十万顶点（BIM 数据本身的问题），这种无法在空间上分割，导致最终产生的单个 b3dm 有 200 多 MB。

这两个参数判断未做顶点压缩之前的数据，所以开启了顶点压缩后，实际产生的 b3dm 要比设置的更小一些。

2.8.8.2 i3dm 优化数

通用模型处理工具里已经实现了对于具有同样几何体数据的对象做实例化渲染优化。这种优化对于模型复用率较高的数据有很大优化，比如 BIM 中的道路附属物，建筑内部的一些小型设备等等。

这里默认是 50，表示如果一个几何体复用次数超过了 50 次，那么我们就把该几何体使用 i3dm 方式渲染。反之使用 b3dm。

模型复用率越高，i3dm 优化意义越大。有人尝试把这个值设置为 2，只要复用超过 2 次，就使用 i3dm，就会发现 i3dm 的个数会很多，反倒导致最终的渲染效率降低。所以这个阈值最好和你的数据来匹配调整更合适一些，达到一个渲染批次和数据量的平衡。

2.8.8.3 纹理合并

纹理合并开关打开之后，我们会尝试对 repeat 方式的纹理贴图进行合并，合并之后的数据在官方 Cesium 版本上无法浏览，需要在 EarthSDK 下浏览。如果场景中 repeat 纹理过多，通过此参数可以提升渲染效率。

2.8.8.4 中心线

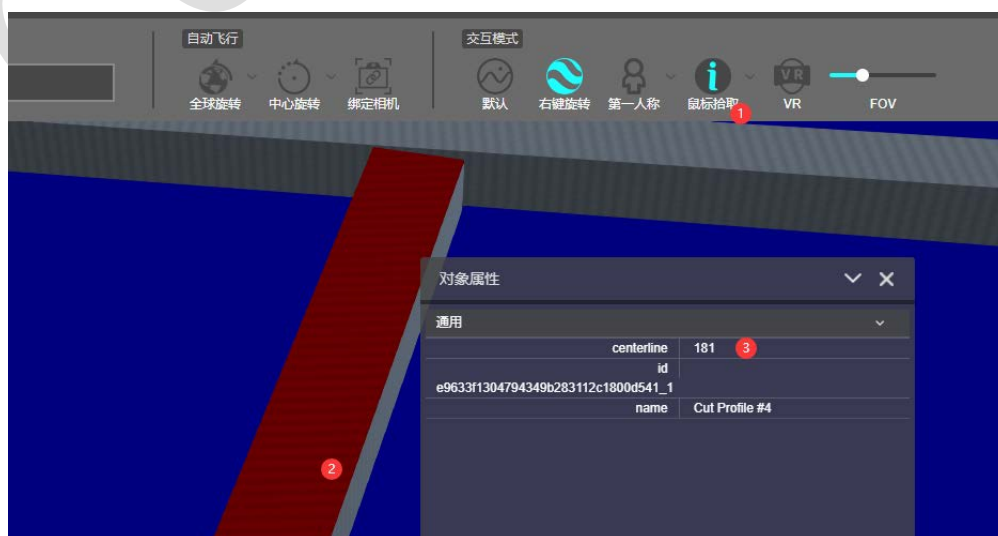
我们在一些项目实践中，经常要做一些针对线性数据（管道、巷道、隧道）做沿着线性方向的着色等控制，如下图：



我们设置中心线之后，内部处理模型的时候，我们会增加模型的逐顶点属性 (attribute)，三个 float 值，分别对应了：

- 1， 相对中心线起点的距离比例（0~1） 0 表示起点位置，1 表示终点位置。
- 2， 中心线序号，从 0 开始，是中心线 shp 的 fid。
- 3， 相对中心线起点的绝对距离，单位是米。

另外我们在 3dtiles 的属性里增加了 centerline 属性，通过 lab 的点击工具可以看到，这个值表示构件所属的中心线序号：



为了效果正确，这里必须对中心线有要求：

- 1，必须是 shp 格式，而且要信息完整，包括 prj，dbf 等文件。
- 2，里面矢量元素必须是折线（linestring）而且坐标点带 z 值，不支持其他类型的中心线格式。
- 3，中心线的位置必须和模型位置匹配，中心线落在线状模型的内部。
- 4，每个构件只能属于一个中心线，所以构件的长度需要小于等于中心线，而且不能跨越中心线。

最后说明，处理的数据携带上述的 attribute 之后，不影响在 cesium 里的加载和使用，但是如果要做类似上述热力图效果，需要自己写 shader 实现，未来可能会在 EarthSDK 上做一些封装来支持。

2.8.9 八叉树处理器

点击中间的  按钮，切换到八叉树处理器，界面如下：



关于八叉树 LOD，我们需要澄清一个误区：

数据不是带了 LOD 就更优，加载更平滑，渲染更高效。尤其你数据总量就 1G 左右，那其实更应该选择小场景，小场景不带 LOD，可以通过其他的优化手段来最快速加载数据，展示效果和效率基本和建模的精度相当。对于小数据建了 LOD，可能大多情况下都是多此一举。

当数据实在是无法用小场景加载，再考虑采用我们的八叉树处理器来创建 LOD，创建 LOD 是个算法要求很高的过程，直到当前的 Cesiumlab3，我们也无法宣称这个算法很完美，它依然不能应对不限量的数据加载。

2.8.9.1 LOD 策略

创建 LOD 有很多策略，比如以前做数字城市的时候根据建筑对象，每个建筑去生成 lod，

然后每个建筑根据远近单独调度。但是我们既然这个工具叫通用模型处理，那么其实也就表示，我们的 LOD 策略需要更通用一些。我们无法预先知道用户的数据是建筑还是道路或者是很精细的 BIM 模型，甚至是某种设备模型。目前我们有两种 LOD 策略：

1，重建化简：

严格按照八叉树分块边界对数据分块，然后每块内尝试简化。3dtiles 在调度中是子块完全替换父块 (refine:replace)

这种 LOD 策略简单暴力适应性强，适合传统的数字城市的城市建筑模型，目前主要缺陷是：简化速度过慢，而且有个别表面法向量不正确。

2，尺寸过滤：

根据每个对象的大小和当前八叉树级别来计算对象的像素大小

对象当前的像素大小 = 对象 AABB 包围盒 / 当前八叉树块大小 * 512

当这个像素大于最小像素参数的时候，那么该对象会存在于当前八叉树节点中

3dtiles 调度中是增量方式，子块和父块同时渲染 (refine:add)

这种 LOD 策略下，每个对象实际并未精简，只是每个对象根据大小加载的优先级不同，适合构件个数非常多的复杂 BIM 模型。

即便有两种策略，但是对于每一类特定数据可能都不是最好方案，如果你们有搞不定的已知数据，可以拿给我们来定制 LOD 生成策略。

2.8.9.2 最小级别

我们的级别和精度 是如下计算的

地球半径(米)	6378137	
地球半周长 (米)	20037508.34	
级别	块大小(米)	块精度米
0	20037508.34	39135.76
1	10018754.17	19567.88
2	5009377.09	9783.94
3	2504688.54	4891.97
4	1252344.27	2445.985
5	626172.14	1222.992
6	313086.07	611.4962
7	156543.03	305.7481
8	78271.52	152.8741
9	39135.76	76.43703
10	19567.88	38.21851
11	9783.94	19.10926
12	4891.97	9.554629
13	2445.98	4.777314
14	1222.99	2.388657
15	611.50	1.194329
16	305.75	0.597164
17	152.87	0.298582
18	76.44	0.149291
19	38.22	0.074646
20	19.11	0.037323
21	9.55	0.018661
22	4.78	0.009331
23	2.39	0.004665
24	1.19	0.002333
25	0.60	0.001166

默认的最小级别 16 级，对应的精度 0.59，也就是说这种尺度下最多能看清楚半米左右的区别。最大级别是 20 级，对应的精度是 0.037，也就是这种尺度下最多能看清 4 厘米左右的区别。

这个尺度对于我们一般的建筑模型完全足够了。

八叉树计算是个极度缓慢的过程，一个块需要的时间大约是在分钟级别。1000 米*1000 米范围的大约有 9 个 16 级块，可能有 50 或者 60 个 17 级块，500 个 18 级块，4000 个 19 级块，32000 个 20 级。这一平方公里要处理需要的时间很夸张，如果数据已知，那就降低下最大级别。

最小级别按照 数据总范围来设定，如果还是 1000 米*1000 米的模型数据，我们可能设置到 16 或者 17 级比较合适，保证顶层的块个数小于 100 即可。

2.8.9.3 最大级别

一般最大级别 = 最小级别 + 3

这样设置的目的是为了保证数据能够处理完毕

如果等的起，可以设置的更大一些。

另外也不是所有位置我们都会处理到最大级别：

如果某个八叉树块内已经没有包含任何数据或者我们统计出来包含的数据量最高精度也不超过我们的阈值，那么这个块就不再继续细分。

有一类特殊数据，比如对于建筑物白模，可以设置 最大级别=最小级别，因为它的几何体简单而且几乎没有纹理。

2.8.9.4 三角网简化

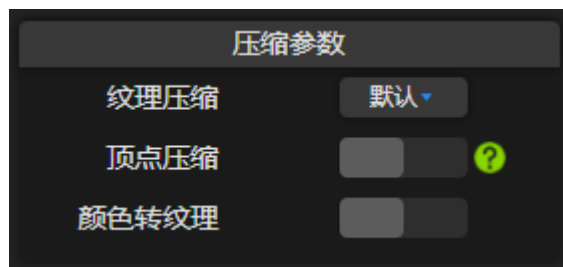
如果场景里几何体复杂密度很高，那么可以开启。但是我们见过很多模型，他的几何体其实很简单，纹理非常大，比如 20MB 的 obj，但是纹理竟然有 8000 多张，这种模型就无需开启三角网进化，我们内部仅对纹理进行简化处理。

在 LOD 策略选择了重建化简得时候有效，如果不开启，那么不对几何体重建，但是会对纹理重建。

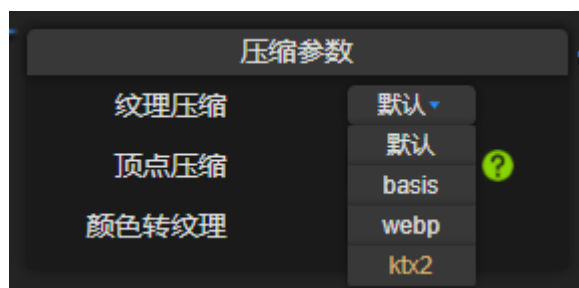
2.8.9.5 可见大小

单位是像素，在 LOD 策略是尺寸筛选的时候有效，具体见 LOD 策略的解释。

2.8.10 压缩参数



2.8.10.1 纹理格式



选择【默认】使用 jpg（非透明）或者 png（透明）格式。

选择【basis】比原来的综合优化（crn）的压缩速度更快，且支持 pc，android，iphone 多平台使用，只能在 EarthSDK 下使用。

选择【Webp】使用 webp 格式的纹理，webp 存储量大约是默认 jpg 的 70%，可以加快网络传输，但是它只能在 chrome 下使用，而且不会减少显存占用。

选择【ktx2】使用 ktx2 格式的纹理，存储量大约是默认 jpg 的 90%，可以固定减少显存消耗 5/6 左右，比如一张 1024*1024 分辨率纹理，默认 jpg 在显存里实际占用是固定 3MB 显存，处理程 ktx2 只占 $3MB / 6 = 512KB$ ，这个对于显卡来说极度友好。不过目前处理过程来相对缓慢。

目前 ktx2 需要付费才可以选择。

我们建议对陌生数据分两次处理，第一次默认纹理处理一遍，检查有无效果异常。如果没问题，第二次再进行 ktx2 压缩处理提升显示效率。

2.8.10.2 顶点压缩

顶点压缩采用了 Draco 压缩算法来减少顶点数据的存储量，Draco 是比较高效的一种压缩方式，目前我们内部的配置是顶点采用 14bit，法向量采用 10bit，纹理坐标采用 10bit，总计 34bit 存储一个顶点。未压缩的数据顶点是 96bit（3 个 float），法向量 96bit（3 个 float），纹理坐标 64bit（2 个 float），总计 256bit。那么压缩率是：大约 1/7.5，这个压缩率很强大的，大量节省传输数据量。如果你的模型观察视角都是较大范围，那么可以开启这个，如果你的模型包含一些细小的零件，那么最好不要开启，会导致模型扭曲变形。

2.8.10.3 颜色转纹理

这个开关选项，如果开启，如果模型 Mesh 的材质存在基础颜色(baseColorFactor)，且不等于纯白色，会做如下操作：

如果此 Mesh 材质也存在基础色纹理，那么该纹理的每个像素值会乘以这个基础颜色。

如果此 Mesh 材质不存在基础色纹理，那么会构造一个 4x4 的小纹理设置到这个材质上，并且把材质的基础色设置为白色。

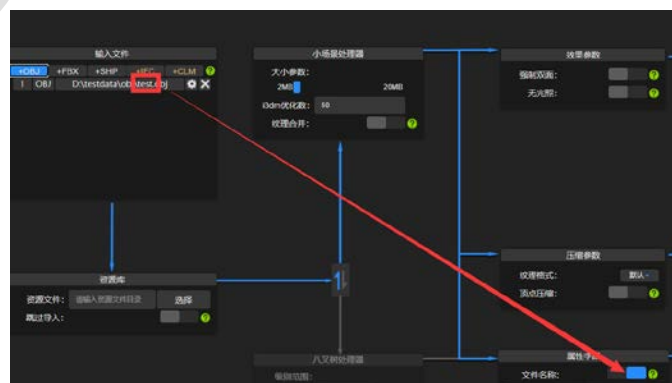
这么做的意义在于，对于一些 BIM 构件，内部几乎没有纹理，绝大部分构件都是不同颜色，为了降低渲染批次，而进行此操作。

2.8.11 属性字段



2.8.11.1 文件名称

当 **文件名称:** 开启之后，我们生成的 3dtiles，会多出一个 file 的属性，该属性保存的是该对象的导入文件名，在 3dtiles 预览的时候点击可以看到该属性。主要用来区分模型对象的所属文件，方便做业务。



2.8.11.2 属性存储

当添加 clm、shp、ifc 这些自带属性字段的文件，或者添加有属性文件配置的 fbx、obj 的时候，属性字段列表会列出所有字段

序号	名称	字段类型	是否存储	
			全选	反选
1	sid	int32_t	☐	☐
2	类别	string	☐	☐
3	族	string	☐	☐
4	族与类型	string	☐	☐
5	名称	string	☐	☐
6	材质	string	☐	☐
7	图像	string	☐	☐

是否存储列通过勾选来选择需要存储到 3dtiles 里的属性。3dtiles 属性可以用来做着色分析和控制可见性，同时也会增加数据量，根据你可视化业务的需要酌情选择。

2.8.12 数据存储

数据存储

原始坐标:

☐

?

存储类型:

紧凑(alt)

请输入输出路径

选择

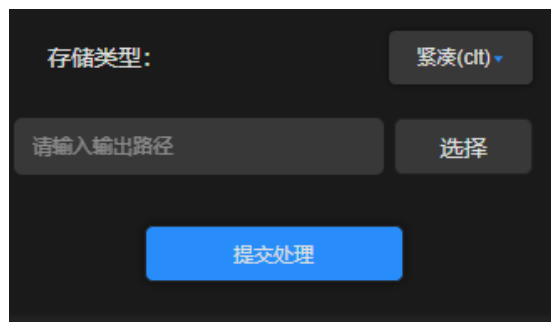
提交处理

2.8.12.1 原始坐标

这是一个特殊需求选项，在本文的【通用功能】【空间参考】【输出数据的空间参考】部分我们讨论过 3dtiles 里存储的坐标和输入文件里的坐标并不一致，但是有一些需求会要求保证 3dtiles 里坐标和输入一致，那么可以打开这个选项。

有一些限制条件，要求输入数据的空间参考必须是 ENU:{纬度},{经度}形式，EPSG 形式的即便勾选了此选项也没有作用。如果有多个输入文件，只能保证和第一个输入文件的原始坐标一致（除非所有文件的空间参考是一致的，那么所有文件都可以保证）。

2.8.12.2 存储类型



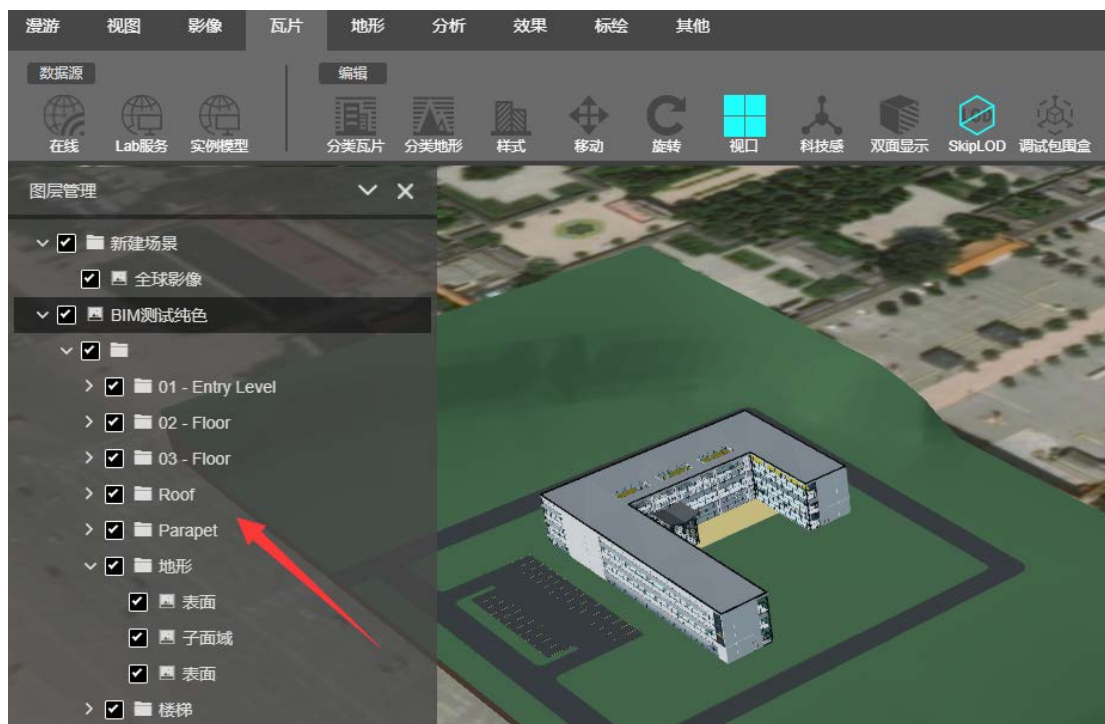
存储类型的选择在本文【数据存储】部分已经讨论过，不再赘述。

2.8.13 模型结构树

通用模型处理的时候默认会生成一个 scenetree.json 文件和 tileset.json 同时存放：

> gisdata > scenes > test > t1		
名称	修改日期	类型
16_1_0_0.json	2021/11/23 15:40	JSON File
17_3_0_0.json	2021/11/23 15:40	JSON File
18_7_1_0.b3dm	2021/11/23 15:40	B3DM 文件
18_7_1_0_0.json	2021/11/23 15:40	JSON File
scenetree.json	2021/11/23 15:40	JSON File
tileset.json	2021/11/23 15:40	JSON File
18_7_0_0.b3dm	2021/11/23 15:40	B3DM 文件
18_7_0_0.json	2021/11/23 15:40	JSON File
18_6_1_0.b3dm	2021/11/23 15:40	B3DM 文件

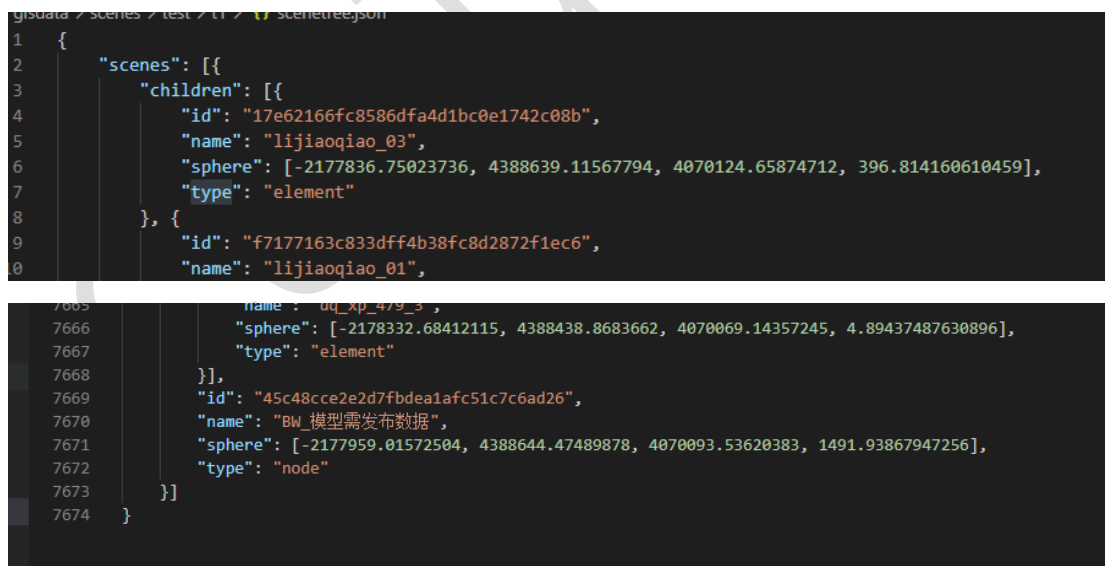
此文件不是 3dtiles 的标准，仅仅是我们自定义的一个文件，但是这个文件在实际使用的时候非常有用，CesiumLab 自带的三维可视页面上，添加一个新的瓦片图层（3dtile）的时候，会尝试自动请求该文件，如果请求成功，会在图层管理上构造模型结构树：



通过模型结构树可以定位构件，以及控制可见性。

2.8.13.1 Scenetree.json 文件说明

scenetree.json 是 utf8 编码的 json 格式文件。



最根上是 scenes，它是一个数组，表示 json 里有几个场景，此处场景的个数实际对应了你处理数据时候的输入文件个数。

除 scenes 之外，其他都是节点对象，对象有以下属性：

type:

如果为“node”表示节点对象是一个文件夹，node 在三维数据上并没有对应的渲染内容，仅仅是为了组织这个树结构。

如果为"element"表示节点对象是一个构件, element 在三维数据上一定有对应的渲染内容, 可以通过 3dtilestyle 去根据 id 控制可见性和着色。

id:

节点对象的 id, 具体来源参考 id 属性的来源。

name:

节点对象的名称, 这个名称是为了 UI 上显示方便, 具体来源参考 name 属性的来源。

sphere:

wgs84 世界坐标系下包围球, 前三个值表示 x,y,z 坐标, 最后一个表示半径。对于"node"对象来说它的包围球是它下面所有"node"和"element"的并集包围球。

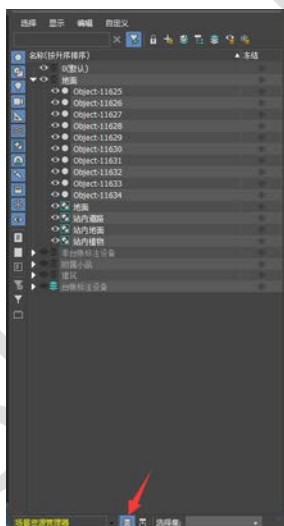
children:

"node"对象下包含此属性, 表示在这个对象下的子对象, 用来构造树状结构。

2.8.13.2 Scenetree.json 生成方式

Scenetree 的树结构对于不同的输入文件按照如下方式组织:

对于 3dsmax 导出的 fbx 格式, 该场景组织是和 3dmax 里的结构组织是完全一致的



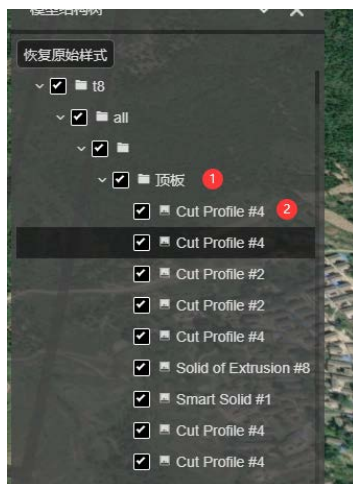
对于 revit 导出的 clm, 我们按照:

标高 (Level), 类别 (Category), 构件 (Element) 三级来构造此树



对于 microstation 导出的 clm, 我们按照:

层 (Layer), 构件 (element) 两级构造此树。

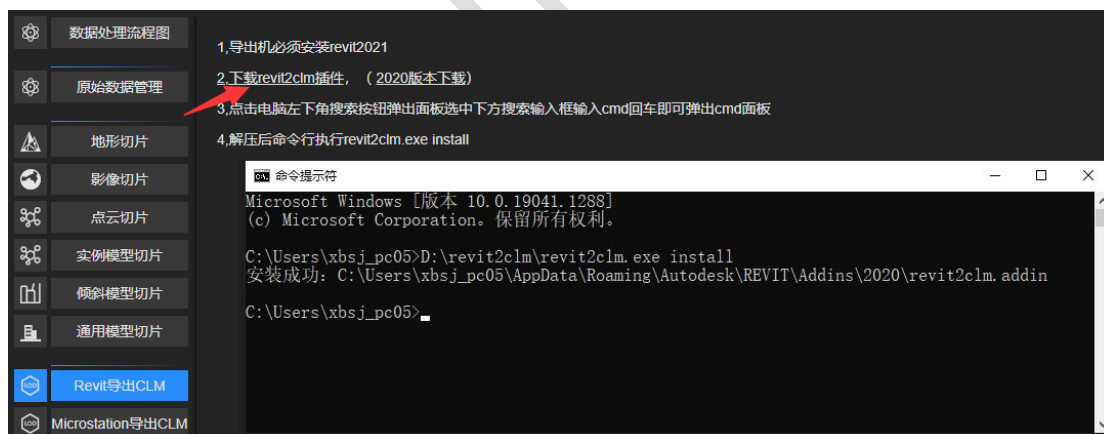


2.9 Revit 导出 clm

注意 Revit 导出 clm 是运行在安装了 Revit2021 的机器上，而不是服务器上，所以在需要导出的地方要进行独立安装。

2.9.1 安装插件

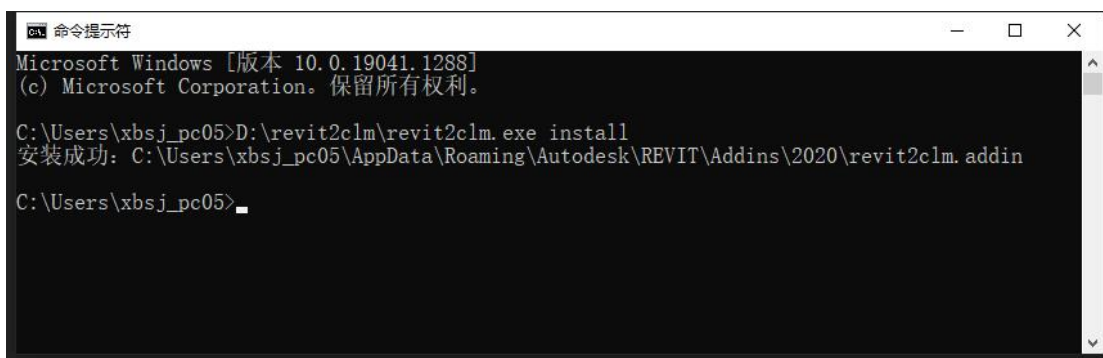
1, 点击程序上这个下载链接



2, 把下载的 revit2clm.zip 解压缩到一个目录

3, 在解压缩目录打开 cmd 窗口

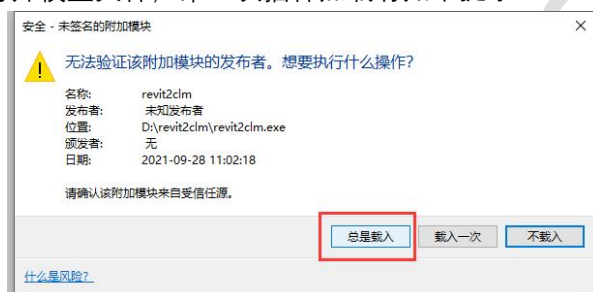
4, 执行 revit2clm.exe install 命令



注意观察提示信息，有**安装成功**字样。

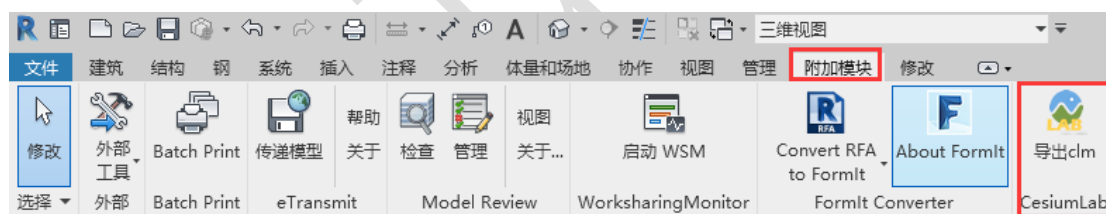
2.9.2 操作过程

启动 revit2020，并打开模型文件，第一次插件加载有如下提示：



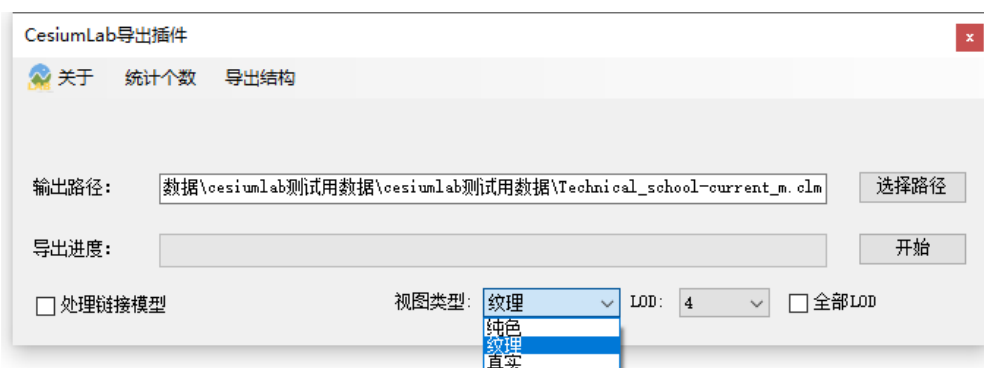
点击【总是载入】

revit2020 菜单栏 附加模块 内 寻找 导出 clm 按钮



弹出导出 clm 面板，选择导出 clm 的路径 以及参数，开始导出

2.9.3 导出参数



2.9.3.1 统计个数

一个调试功能，仅弹出当前打开的 revit 文件里构件个数。

2.9.3.2 导出结构

一个调试功能，导出当前打开的 revit 文件的回调结构，生成一个 labc 文件，主要是辅助开发者解决 bug 使用。

2.9.3.3 输出路径

输入或者点击【选择路径】按钮，输入一个文件名，做为 clm 生成的路径。
默认的路径已经填写为和 rvt 同路径，同名称，仅后缀不同的 clm。

2.9.3.4 处理链接模型

rvt 文件是支持挂接其他 rvt 文件，勾选此选项，那么对于其他挂接的 rvt 也会进行导出处理，否则仅处理当前 rvt 文件。

2.9.3.5 视图类型

纯色：表示导出的材质里忽略纹理信息，仅使用材质中的基础色。

纹理：表示导出的材质里优先使用纹理信息做为基础颜色，而忽略材质颜色，如果没有纹理，那么使用材质颜色。

混色：表示导出的材质里纹理和材质颜色都存储

revit 本身的材质系统也是非常复杂的，材质颜色和纹理之间并非简单混色关系，所以我们通常推荐仅使用【纹理】模式即可。

2.9.3.6 LOD

Revit 把构件生成三角网的级别设定了 0~15 总计 16 个级别，数值越大，几何体越精细，我们通常推荐 7 或者 8 即可，如果数据量很大，可以降低选择 4 甚至更小。

2.9.3.7 全部 LOD

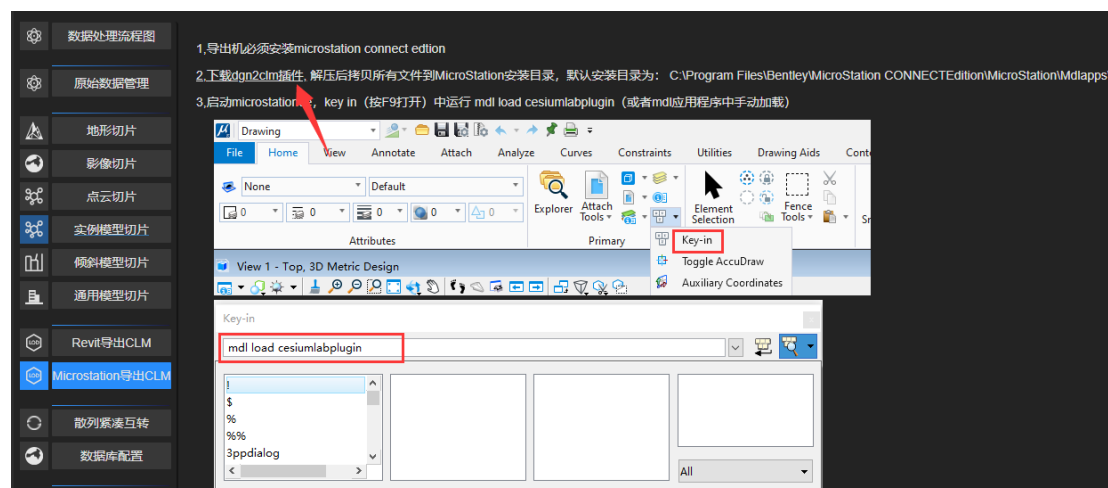
一个调试选项，目前不要勾选。

2.10 Microstation 导出 clm

注意 Microstation 导出 clm 是运行在安装了 Microstation Connect Edition 的机器上，而不是服务器上，所以在需要导出的地方要进行独立安装。

2.10.1 安装插件

1, 点击程序上这个下载链接



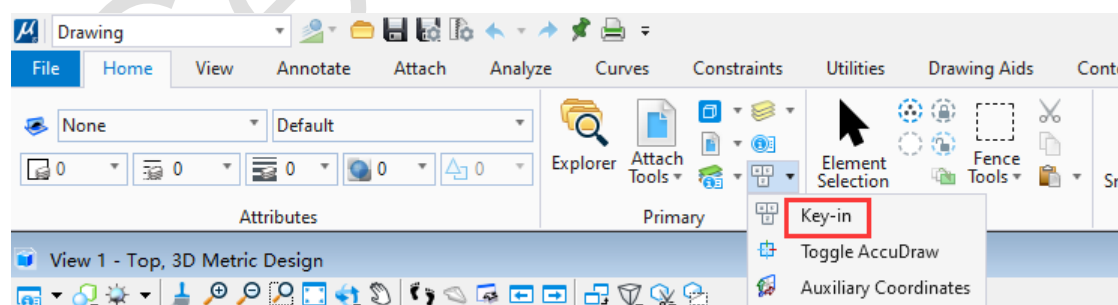
2, 解压缩下载到的 dgn2clm.zip

3, 把解压后的文件拷贝到:

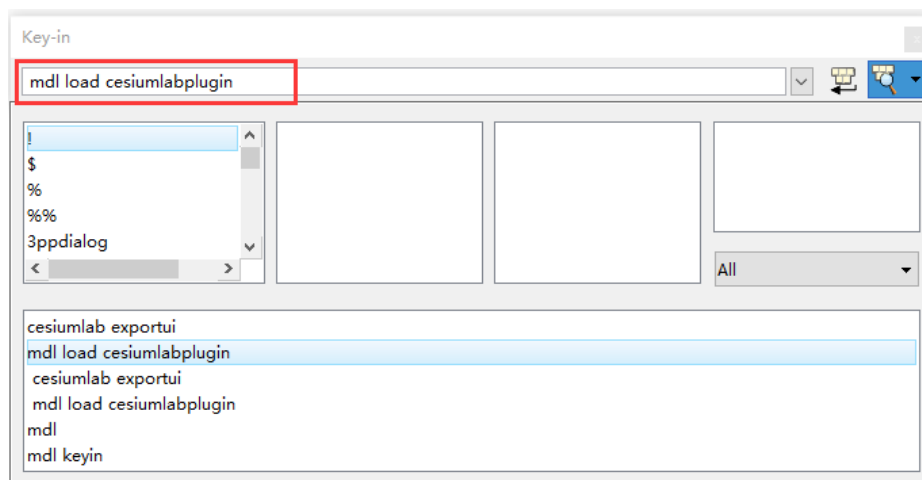
C:\Program Files\Bentley\MicroStation CONNECT Edition\MicroStation\Mdlapps\

2.10.2 操作过程

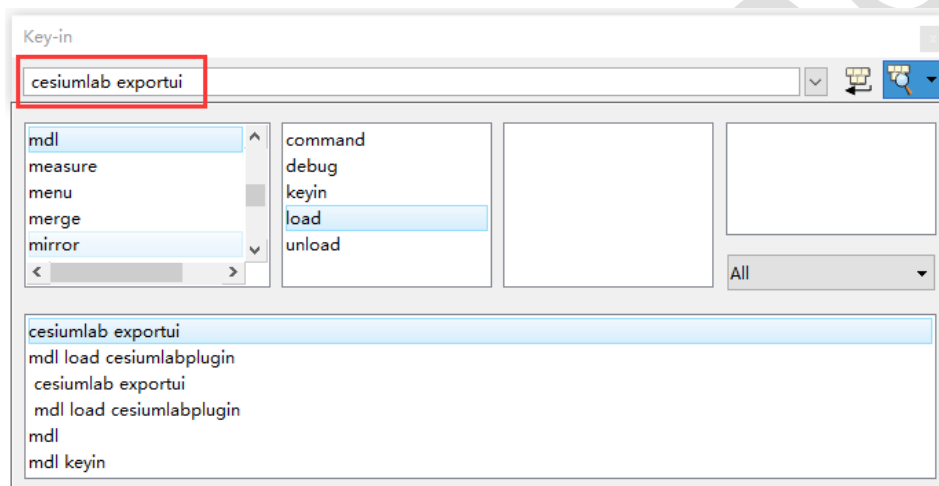
启动 microstationnce 并打开 dgn 模型, 打开 key-in (按 F9 打开)



Key-in 中运行 mdl load cesiumlabplugin

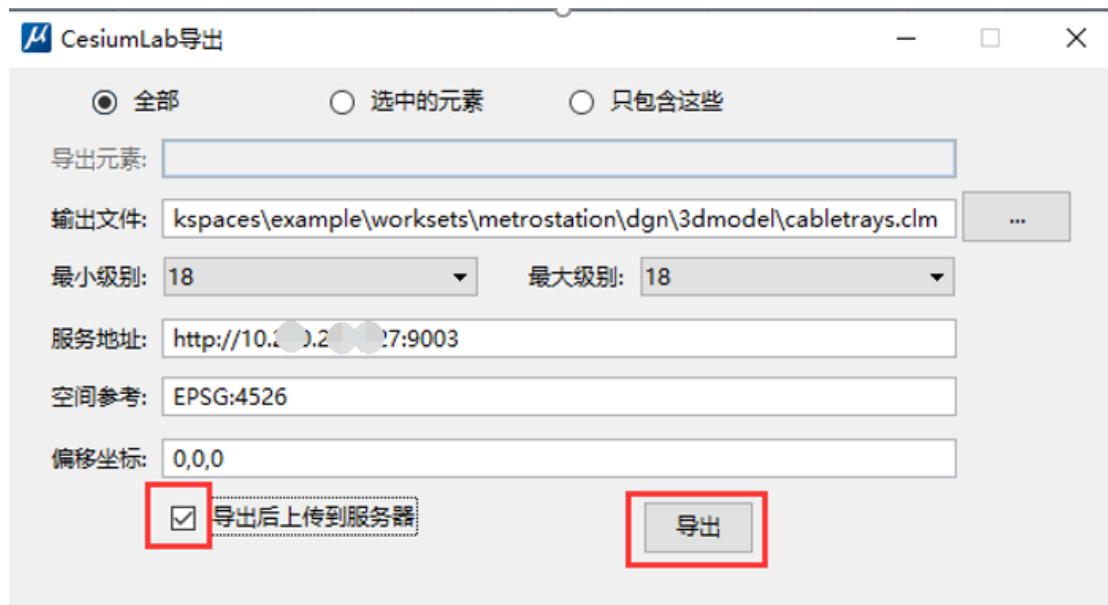


Key-in 中运行 cesiumlab exportui



弹出导出界面

2.10.3 导出参数



勾选【导出后上传到服务器】，输入服务地址、空间参考等参数，点【导出】，完成后提示 cesiumlab 导出成功，并且上传服务器成功。

2.10.3.1 全部

默认选中全部，表示所有构件都导出。

2.10.3.2 选中的元素

选中此选项，那么只导出选中的元素。

2.10.3.3 只包含这些

选中此选项，只导出【导出元素】输入框里列出的元素，需要根据 id 指定，如果多个对象，那么逗号分割。注意这是个程序调试选项，一般不使用。

2.10.3.4 输出文件

直接输入或者点击【...】按钮输入一个文件名，做为 clm 导出的路径。
默认的路径和 dgn 是同目录，同名，后缀为 clm 的文件。

2.10.3.5 最小级别

我们会根据级别设定来导出对应精度的三角网，精度按照下表计算：

地球半径(米)	6378137	
地球半周长 (米)	20037508.34	
级别	块大小(米)	块精度米
0	20037508.34	39135.76
1	10018754.17	19567.88
2	5009377.09	9783.94
3	2504688.54	4891.97
4	1252344.27	2445.985
5	626172.14	1222.992
6	313086.07	611.4962
7	156543.03	305.7481
8	78271.52	152.8741
9	39135.76	76.43703
10	19567.88	38.21851
11	9783.94	19.10926
12	4891.97	9.554629
13	2445.98	4.777314
14	1222.99	2.388657
15	611.50	1.194329
16	305.75	0.597164
17	152.87	0.298582
18	76.44	0.149291
19	38.22	0.074646
20	19.11	0.037323
21	9.55	0.018661
22	4.78	0.009331
23	2.39	0.004665
24	1.19	0.002333
25	0.60	0.001166

2.10.3.6 最大级别

如果最大级别不等于最小级别，那么 clm 会保留多个精度的几何数据，暂时保留，处理阶段还没用上。

2.10.3.7 服务地址

CesiumLab3 服务地址

2.10.3.8 空间参考

输入数据的空间参考

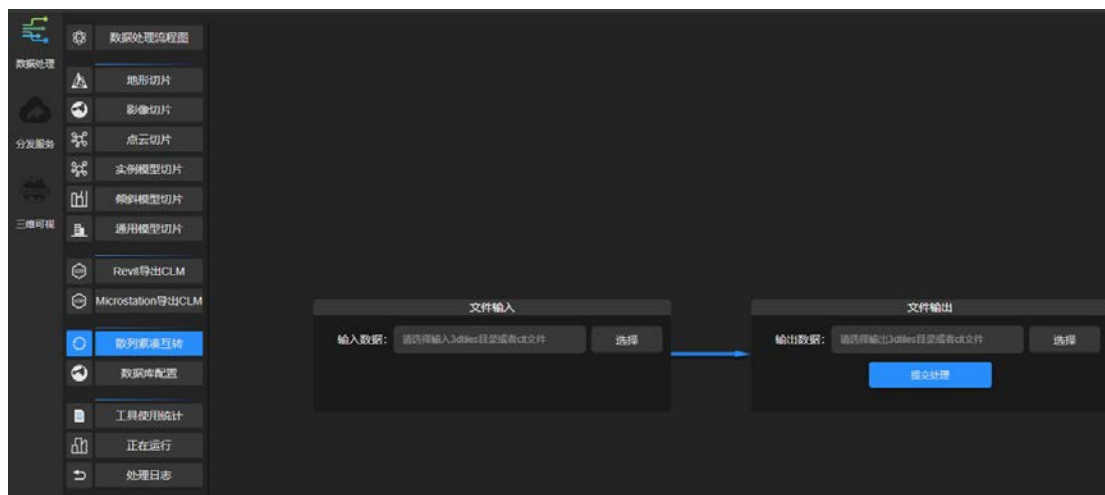
2.10.3.9 偏移坐标

即零点坐标

2.11 散列紧凑互转

用于散列 3dtiles 和 紧凑 (clt) 之间的相互转换。

主页上依次点击【数据处理】、【散列紧凑互转】进入散列紧凑互转工具。

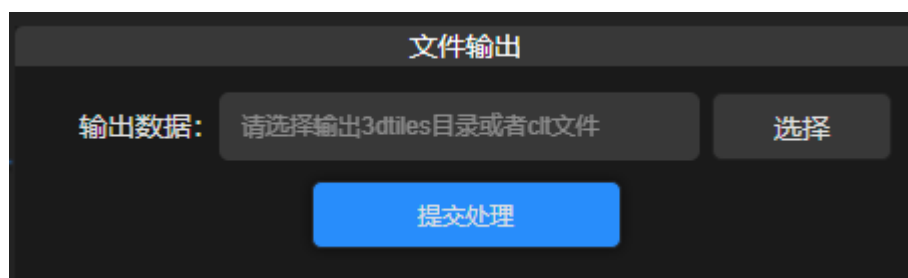


2.11.1 输入文件



点击【选择】按钮，弹出选择文件对话框，可以选择 clt 或者散列 3dtiles 的 tileset.json 文件，如果选择 clt，则表示是 clt 转散列 3dtiles，否则是散列 3dtiles 转 clt。

2.11.2 输出文件



点击【选择】按钮，弹出选择文件对话框，当输入文件为 clt 时，这里只能选择或创建文件夹，当输入文件为散列 3dtiles 时，这里只能选择或创建 clt 文件。

点击【提交处理】跳转到【正在运行】，处理完成后可在分发服务->3dtiles 服务列表中查看和浏览该数据服务。

2.12数据库配置

Lab3 支持将 3dtiles 切片存储到 postgres 或者 mongodb 数据库中，可参考 [2.2.3.4Mongodb 瓦片存储](#)和 [2.2.3.5Postgresql 瓦片存储](#)。在数据库配置中可以添加常用数据库连接。

主页上依次点击【数据处理】、【数据库配置】进入数据库配置。

全部类型

请输入名称及关键字

查询

添加

序号	名称	数据库名	主机	端口	用户名	密码	操作
1	postgres	postgres	localhost	5432	postgres	123456	设置 测试连接 删除
2	a	admin	localhost	27017	cesiumlab	123	设置 测试连接 删除

点击【添加】按钮，弹出添加数据库参数设置窗口，选择数据库类型（postgres/mongodb），名称可以自定义，数据库名称、主机、端口、用户名和密码必须为有效的数据库连接参数，点击【确定】。



设置

数据库类型: postgres

名称: 请输入名称

数据库名称: 请输入数据库名称

主机: 请输入主机IP

端口: 请输入端口号

用户名: 请输入用户名称

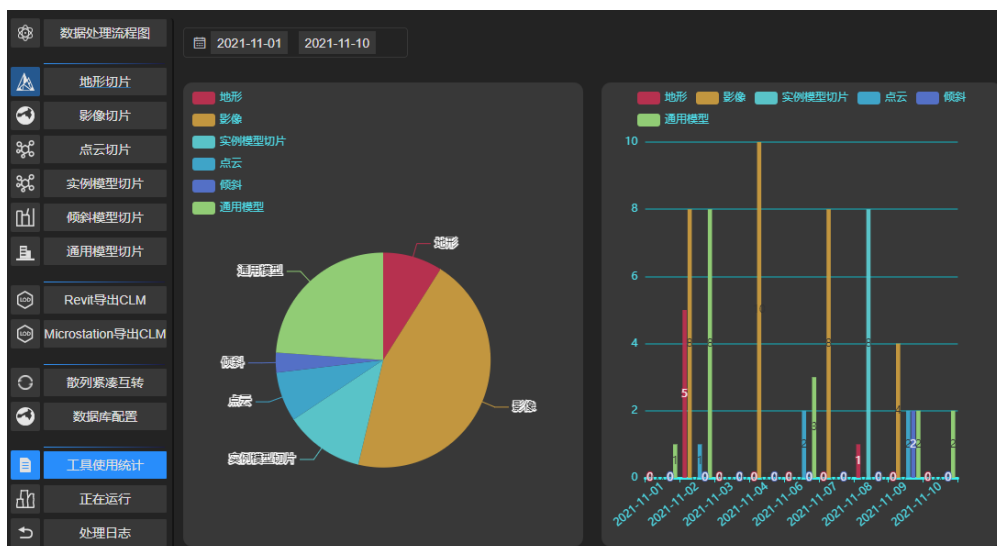
密码: 请输入密码

取消 确定

每条配置记录可以进行设置、测试连接、删除操作，其中测试连接用于测试是否能够连接成功，当连接失败时可能需要等待较长时间才会弹出提示。

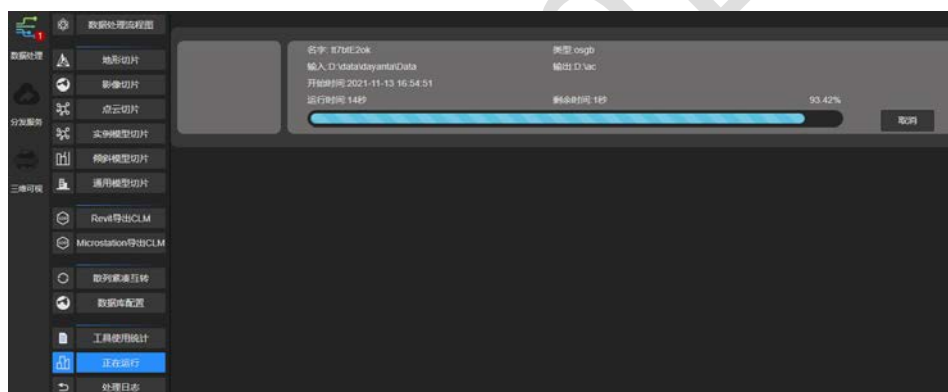
2.13工具使用统计

用于统计某一时间段内的工具使用次数频率等。



2.14正在运行

用于显示正在运行的任务，以及每个任务的进度，



正在运行参数说明



名字：任务的 id

输入：添加处理的切片的输入文件路径；

输出：添加处理的切片的输出文件路径；

开始时间：添加切片的时间；

运行时间：添加开始到添加结束的这一段时间，运行时间=当前时间-开始时间；

剩余时间：剩余时间=运行时间/进度-运行时间；

进度：添加的切片的处理进度；

取消：可以取消正在运行的任务，注意，不支持暂停任务和继续任务；

如果存在正在运行的任务，则在左上角【数据处理】菜单上显示正在运行的任务数。此时点击【数据处理】则会跳转到正在运行界面，否则跳转到数据处理流程图。



2.15处理日志

本页面列出了历史处理任务的所有相关信息。例如：任务类型、输入、输出、开始时间、结束时间以及状态等。

自	选择日期	全部任务类型	全部状态				
序号	任务类型	状态	开始时间	结束时间	输入	输出	操作
1	倾斜		2021-11-13 16:54:51		D:\data\dayanlaData	D:\ac	重新处理
2	点云	完成	2021-11-13 11:51:34	2021-11-13 11:51:37	D:\data\utm50.las	mongodb://cesiumlab:123@loc...	重新处理
3	点云	完成	2021-11-13 11:50:25	2021-11-13 11:50:27	D:\data\bt1\白腾一线6-10\cros...	mongodb://cesiumlab:123@loc...	重新处理
4	点云	完成	2021-11-13 11:49:23	2021-11-13 11:49:24	D:\data\bt1\白腾一线6-10\cros...	mongodb://cesiumlab:123@loc...	重新处理
5	点云	完成	2021-11-13 11:48:14	2021-11-13 11:48:15	D:\data\bt1\白腾一线6-10\cros...	mongodb://cesiumlab:123@loc...	重新处理
6	点云	完成	2021-11-13 11:47:53	2021-11-13 11:47:54	D:\data\bt1\白腾一线6-10\cros...	mongodb://cesiumlab:123@loc...	重新处理
7	点云	完成	2021-11-13 11:45:14	2021-11-13 11:45:15	D:\data\bt1\白腾一线6-10\cros...	mongodb://cesiumlab:123@loc...	重新处理
8	点云	失败	2021-11-13 11:36:42	2021-11-13 11:36:42	D:\data\utm50.las	D:\aa	重新处理
9	点云	失败	2021-11-13 11:36:18	2021-11-13 11:36:18	D:\data\bt1\白腾一线6-10\cros...	D:\aa	重新处理
10	点云	失败	2021-11-13 11:35:34	2021-11-13 11:35:34	D:\data\bt1\白腾一线6-10\cros...	mongodb://cesiumlab:123@loc...	重新处理

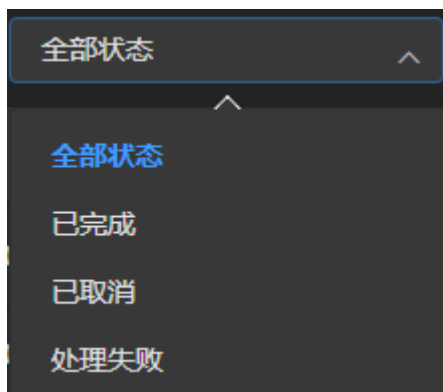
选择日期：点击这个选择日期输入框可以选择要查询的日期，再点击查询按钮查询按钮截图查看这个时间段的所有的处理日志，也可以手动输入日期进行查询。



全部任务类型：点击这个选择框会弹出一个下拉框，下拉框里展示了处理日志的所有类型，可以根据类型来进行筛选查询处理日志。



全部状态：点击这个选择框会弹出一个下拉框，下拉框里展示了处理日志的处理状态，可以根据处理状态来进行筛选查询处理日志。



2.16原始数据管理

原始数据管理功能方便用户上传和管理已上传的数据。

主页上依次点击【数据处理】、【原始数据管理】，进入到原始数据管理工具页面。

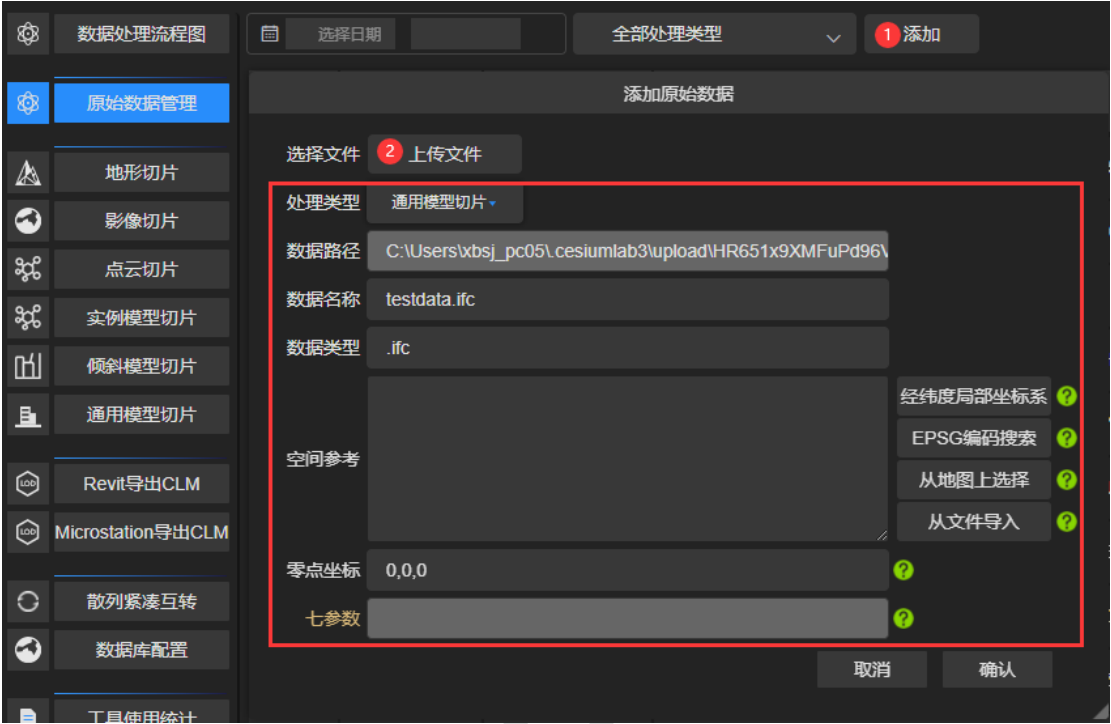
功能包括：①添加数据 ②删除数据 ③修改数据 ④处理数据 ⑤查询数据。

数据管理	序号	ID	数据名称	数据源存储路径	空间参考	上传时间	数据来源	数据类型	处理类型	操作
地形切片	1	LEQlabGN	point.shp	C:\Users\lab\Documents\upload\点云数据_y5F...		2022-01-21 15:39:54	...	.shp	实例模型切片	删除 修改 处理
影像切片	2	h7u4h3	shp_city.shp	C:\Users\lab\Documents\upload\点云数据_y5F...		2022-01-21 15:39:21	...	.shp	通用模型切片	删除 修改 处理
点云切片	3	p4h3w8	metadata.xml	C:\Users\lab\Documents\upload\点云数据_y5F...		2022-01-21 15:21:19	...	.xml	倾斜模型切片	删除 修改 处理
通用模型切片	4	4nu4h3	xc.FBX	C:\Users\lab\Documents\upload\点云数据_y5F...		2022-01-21 15:17:10	...	.fbx	通用模型切片	删除 修改 处理
倾斜模型切片	5	rUedJ7L	Map_v1.unl...	C:\Users\lab\Documents\upload\点云数据_y5F...		2022-01-21 15:17:02	...	.obj	通用模型切片	删除 修改 处理
通用模型切片	6	Og4rV25	BOEING.doc	C:\Users\lab\Documents\upload\点云数据_y5F...		2022-01-21 15:16:43	...	.doc	通用模型切片	删除 修改 处理
倾斜模型切片	7	ou4n242	point_cloud...	C:\Users\lab\Documents\upload\点云数据_y5F...		2022-01-21 15:16:33	...	.las	点云切片	删除 修改 处理
通用模型切片	8	sShcy24k	1224.tif	C:\Users\lab\Documents\upload\点云数据_y5F...		2022-01-21 15:16:19	...	.tif	影像切片	删除 修改 处理
通用模型切片	9	EF7zAvy	ASTGTM2_N3...	C:\Users\lab\Documents\upload\点云数据_y5F...		2022-01-21 15:16:03	...	.tif	地形切片	删除 修改 处理
通用模型切片	10	s7e5dwL				2022-01-21 10:21:43	...		数据源交互	删除 修改 处理

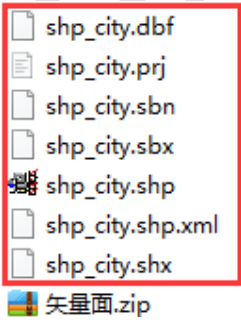
使用方法（功能）如下：

2.16.1 添加

依次点击【添加】、【上传文件】按钮，选择文件，设置[空间参考](#)等参数，点【确认】，完成添加。



注意：务必上传完整文件。例如：矢量面由.shp、.dbf 等文件组成，需将所有文件压缩，上传时选择该压缩文件。



2.16.2 删除

序号	ID	数据名称	服务器存储路径	空间参考	上传时间	数据来源	数据类型	处理类型	操作
1	LBQlabGN	point.shp	C:\Users\xbj_pc05\cesiumlab3\upload\矢量点\3dm_y5f...		2022-01-21 15:39:54	..1	shp	实例模型切片	删除 修改 处理
2	h7uJzhC3	shp_city.shp	C:\Users\xbj_pc05\cesiumlab3\upload\矢量面_5hwSh0VIL...		2022-01-21 15:39:21	..1	shp	通用模型切片	删除 修改 处理
3	jR5TWk8	metadata.xml	C:\Users\xbj_pc05\cesiumlab3\upload\osgb_gBepLHNY...		2022-01-21 15:21:19	..1	xml	倾斜模型切片	删除 修改 处理
4	4wuf8ndY	xc.FBX	C:\Users\xbj_pc05\cesiumlab3\upload\gYK-4gtMvCH9F...		2022-01-21 15:17:10	..1	fbx	通用模型切片	删除 修改 处理
5	rUedJ7L	Map_v1.unlty...	C:\Users\xbj_pc05\cesiumlab3\upload\LR32Ygaw2mYK...		2022-01-21 15:17:02	..1	obj	通用模型切片	删除 修改 处理

点击【删除】，删除单个记录。

2.16.3 修改

空间参考	上传时间	数据来源	数据类型	处理类型	操作
	2022-01-21 15:39:54	..1	.shp	实例模型切片	删除 修改 处理
修改原始数据					
处理类型	实例模型切片				删除 修改 处理
数据路径	C:\Users\xbsj_pc05\cesiumlab3\upload\矢量点\3dm_y5fBY2				删除 修改 处理
数据名称	point.shp				删除 修改 处理
数据类型	.shp				删除 修改 处理
空间参考	经纬度局部坐标系 EPSG编码搜索 从地图上选择 从文件导入				删除 修改 处理
零点坐标	0,0,0				删除 修改 处理
七参数					删除 修改 处理
取消 确认					删除 修改 处理

点击【修改】，可修改处理类型、数据名称、空间参考等。点【确认】，完成修改。

2.16.4 处理

序号	id	数据名称	服务器存储路径	空间参考	上传时间	数据来源	数据类型	处理类型	操作
1	LBQabGN	point.shp	C:\Users\xbsj_pc05\cesiumlab3\upload\矢量点\3dm_y5fBY2		2022-01-21 15:39:54	..1	.shp	实例模型切片	删除 修改 处理
2	h7uJthC3	shp_city.shp	C:\Users\xbsj_pc05\cesiumlab3\upload\矢量点\5hwSNOVh...		2022-01-21 15:39:21	..1	.shp	通用模型切片	删除 修改 处理
3	f#R5Twk8	metadata.xml	C:\Users\xbsj_pc05\cesiumlab3\upload\osgb_g8epLHNY...		2022-01-21 15:21:19	..1	.xml	倾斜模型切片	删除 修改 处理
4	4wuf8edY	xc.FBX	C:\Users\xbsj_pc05\cesiumlab3\upload\gyK_4gbMvO8G...		2022-01-21 15:17:10	..1	.fbx	通用模型切片	删除 修改 处理
5	rUedU7L	Map_v1.unify...	C:\Users\xbsj_pc05\cesiumlab3\upload\R3cygac2mYK...		2022-01-21 15:17:02	..1	.obj	通用模型切片	删除 修改 处理
6	Og4zEy25	BOEING.fic	C:\Users\xbsj_pc05\cesiumlab3\upload\584Qda1daDUL...		2022-01-21 15:16:43	..1	.fic	通用模型切片	删除 修改 处理
7	o44nJ24Q	point_cloud_te...	C:\Users\xbsj_pc05\cesiumlab3\upload\Fb5QE0owlzECn...		2022-01-21 15:16:33	..1	.las	点云切片	删除 修改 处理
8	sShcy24x	1224.tif	C:\Users\xbsj_pc05\cesiumlab3\upload\YlpO_Xpl_86ZOT...		2022-01-21 15:16:19	..1	.tif	影像切片	删除 修改 处理
9	EF2zAwy	ASTGTM2_N3...	C:\Users\xbsj_pc05\cesiumlab3\upload\wYg5BmqgZ2xv...		2022-01-21 15:16:03	..1	.tif	地形切片	删除 修改 处理
10	s7e5kwL				2022-01-21 10:21:43	..1		数据要素互转	删除 修改 处理

点【处理】，跳转到切片工具界面。例如：处理类型为实例模型切片，则跳转到实例模型切片界面。

如果勾选多个数据，点【批量处理】，可将这些数据同时添加到处理任务中。



2.16.5 查询

可通过添加日期、处理类型进行查询。



3 分发服务

分发服务包含 1.首页（点击分发服务后默认显示该页）、2.常规影像、3.实时缓存影像、4、影像集合、5.常规地形、6.实时缓存地形、7.地形集合、8.3dtiles 服务、9.场景服务、10.lod 模型库服务、11.资源服务。



3.1 首页



3.1.1 服务统计

- ①该处选择时间后自动查询对应时间段的服务。
- ②该处可以点击选择需要查看的服务。
- ③该处显示鼠标悬停处的服务数量和占比。

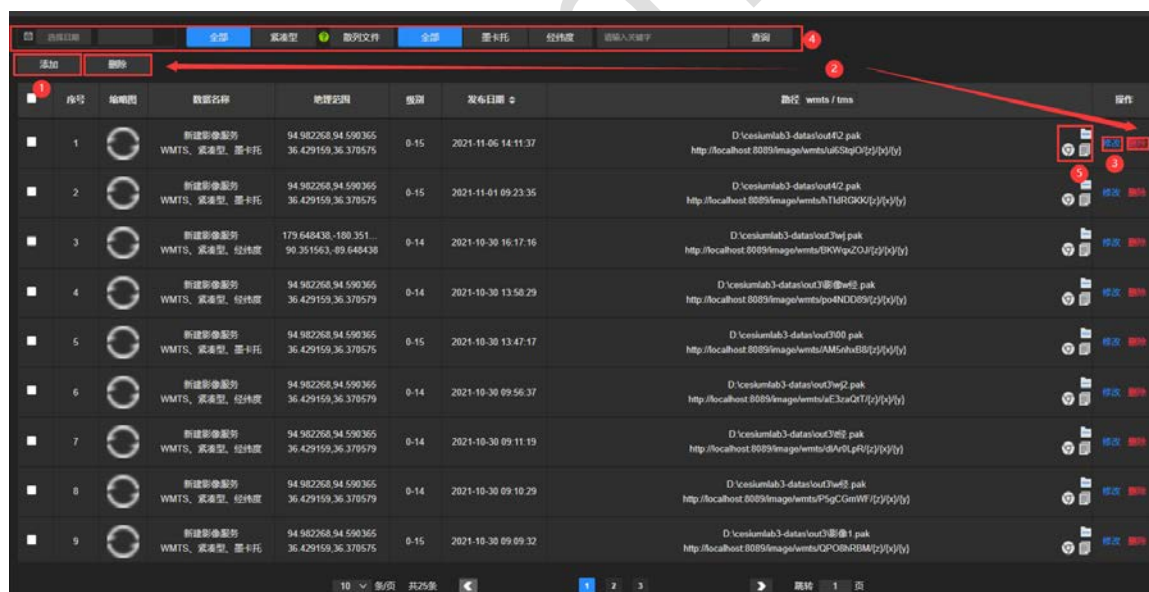
3.1.2 从 cesiumlab2 导入服务

点击 **从cesiumlab2导入服务** 按钮

Cesiumlab3 在服务能力方面做了很多改进，不能直接兼容 cesiumlab2 发布的服务，因此如果需要，可以手动从 cesiumlab2 中导入原有服务。

3.2 常规影像服务

功能简介：cesiumlab 影像切片工具生成的影像切片都可以在这里发布。功能包括：①添加常规影像。②删除常规影像。③修改常规影像。④查询常规影像。⑤打开文件、预览、复制路径。



序号	缩略图	数据名称	地理范围	格式	发布日期	路径: wmts / tms	操作
1		新建影像服务 WMTS、瓦片型、墨卡托	94.982268,94.590365 36.429159,36.370575	0-15	2021-11-06 14:11:37	D:\cesiumlab3-data\out42 pak http://localhost:8089/image/wmts/u65kgO(pj)(v)(y)	预览 删除
2		新建影像服务 WMTS、瓦片型、墨卡托	94.982268,94.590365 36.429159,36.370575	0-15	2021-11-01 09:23:35	D:\cesiumlab3-data\out42 pak http://localhost:8089/image/wmts/h18RQ00(pj)(v)(y)	预览 删除
3		新建影像服务 WMTS、瓦片型、经纬度	179.648438,-100.351... 90.351563,89.648438	0-14	2021-10-30 16:17:16	D:\cesiumlab3-data\out3waj pak http://localhost:8089/image/wmts/BKQWzOJ(pj)(v)(y)	预览 删除
4		新建影像服务 WMTS、瓦片型、经纬度	94.982268,94.590365 36.429159,36.370579	0-14	2021-10-30 13:58:29	D:\cesiumlab3-data\out3waj pak http://localhost:8089/image/wmts/p04N0089(pj)(v)(y)	预览 删除
5		新建影像服务 WMTS、瓦片型、墨卡托	94.982268,94.590365 36.429159,36.370575	0-15	2021-10-30 13:47:17	D:\cesiumlab3-data\out300 pak http://localhost:8089/image/wmts/AM5kuB8(pj)(v)(y)	预览 删除
6		新建影像服务 WMTS、瓦片型、经纬度	94.982268,94.590365 36.429159,36.370579	0-14	2021-10-30 09:56:37	D:\cesiumlab3-data\out3waj pak http://localhost:8089/image/wmts/uE3uQzT(pj)(v)(y)	预览 删除
7		新建影像服务 WMTS、瓦片型、经纬度	94.982268,94.590365 36.429159,36.370579	0-14	2021-10-30 09:11:19	D:\cesiumlab3-data\out3waj pak http://localhost:8089/image/wmts/dA9LpR(pj)(v)(y)	预览 删除
8		新建影像服务 WMTS、瓦片型、经纬度	94.982268,94.590365 36.429159,36.370579	0-14	2021-10-30 09:10:29	D:\cesiumlab3-data\out3waj pak http://localhost:8089/image/wmts/PSgCmW(pj)(v)(y)	预览 删除
9		新建影像服务 WMTS、瓦片型、墨卡托	94.982268,94.590365 36.429159,36.370575	0-15	2021-10-30 09:09:32	D:\cesiumlab3-data\out3waj pak http://localhost:8089/image/wmts/QPOBhR8M(pj)(v)(y)	预览 删除

使用方法（功能）如下：

3.2.1 添加

点击添加按钮，输入数据名称，添加数据路径，可选添加文件夹（散列文件）或添加 pak，点击确定，在服务列表中增加了该服务，完成发布。注：当该影像数据路径已存在服务列表时，将会覆盖原有影像服务。



3.2.2 删除

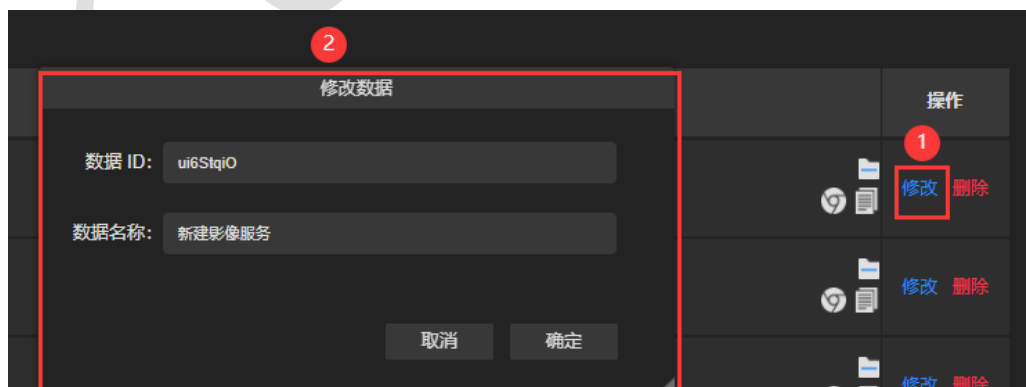
批量删除：对应①②，选中要删除的影像，点击删除。

单个删除：对应③，点击删除。



3.2.3 修改

点击修改，修改数据 ID 或者数据名称，点击确定完成修改。



3.2.4 查询

可通过发布日期、影像存储类型、影像投影类型以及关键字进行查询。



3.2.5 服务操作

对于每个服务，可进行以下操作：打开文件、预览、复制路径

3.2.5.1 打开本地存储目录

点击可打开该影像数据所在路径。

3.2.5.2 预览

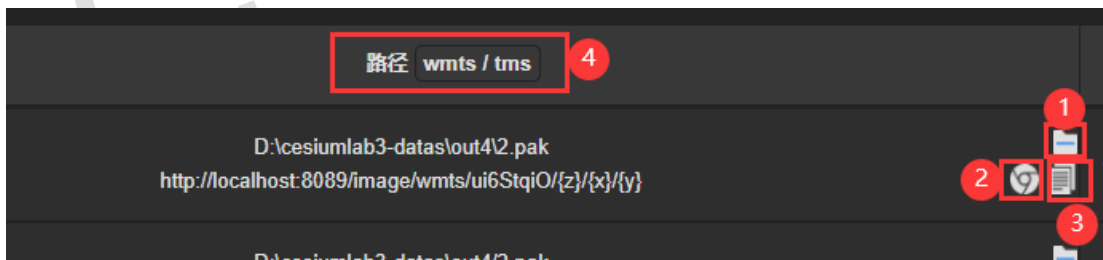
点击跳转到三维可视，自动加载该影像服务，进行浏览。

3.2.5.3 复制路径

点击可复制该影像服务地址。

3.2.5.4 服务类型实时切换

对于同一个影像数据, cesiumlab3 可以支持 wmts 和 tms 两种服务, 点击“wmts/tms”, 可切换服务地址。



3.3 实时缓存影像服务

实时缓存影像服务是对其他影像服务的代理和缓存 (pak)，当请求此服务的切片时，首先从缓存中查找影像切片，若不存在则从其他影像服务下载该切片，返回给客户端并缓存到 pak 中。

实时缓存影像服务包含以下操作：①添加实时缓存影像。②删除实时缓存影像。③修改实时缓存影像。④查询实时缓存影像。⑤打开文件、预览、复制路径。



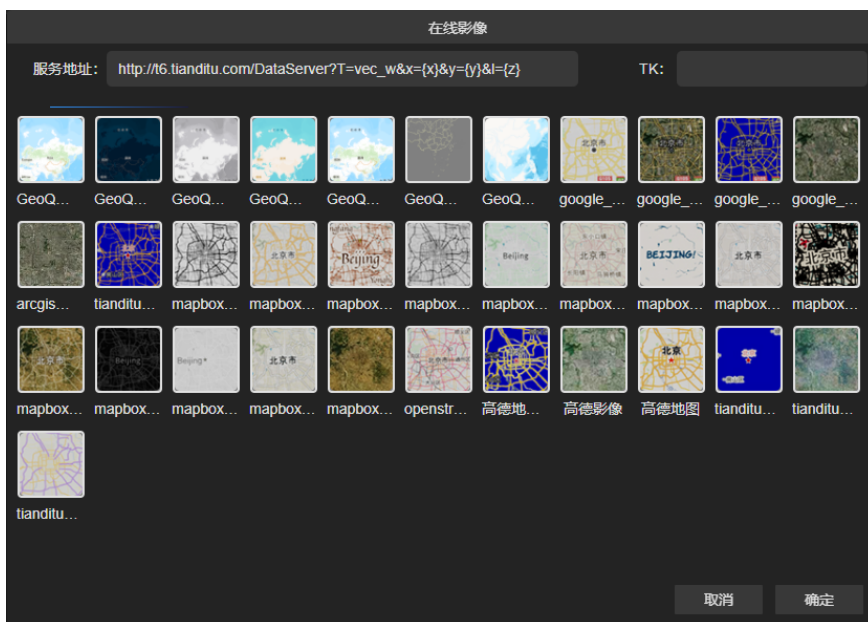
使用方法（功能）如下：

3.3.1 添加

点击添加按钮，输入数据名称，选择在线影像（即需要代理的影像服务地址），选择数据路径，目前只支持紧凑 pak，可以添加一个不存在的 pak 文件，系统将自动创建，输入需要缓存的地理范围、级别范围；也可以添加一个已存在的 pak 文件，此时不可修改地理范围和级别范围，点击确定完成添加。



注：如果代理的是天地图服务，则需要添加 tk 参数，详情可参考：
<http://lbs.tianditu.gov.cn/>。

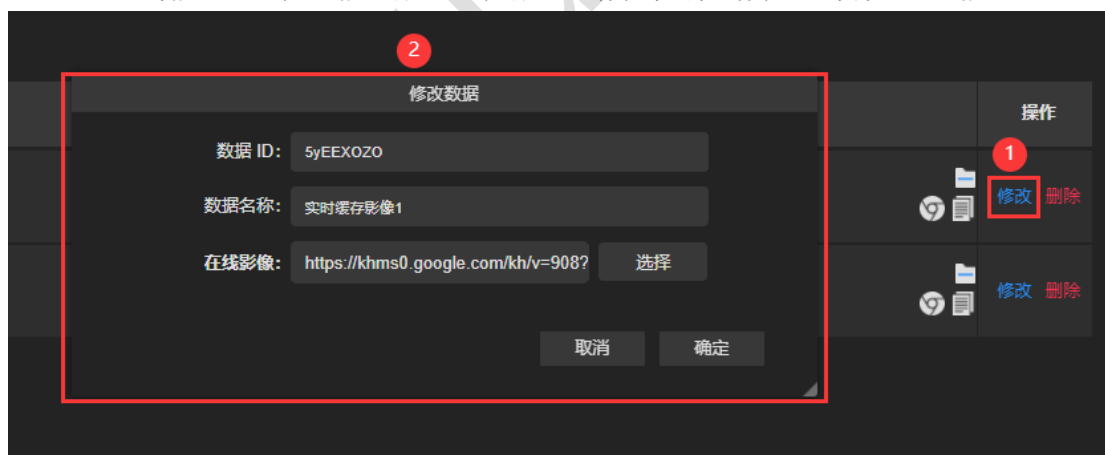


3.3.2 删除

参见“常规影像”中②删除常规影像。

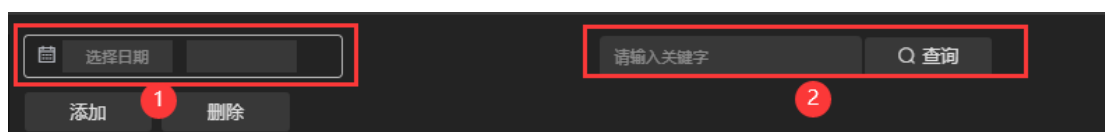
3.3.3 修改

点击修改按钮，可修改数据 ID，数据名称，在线影像，点击确定完成修改。



3.3.4 查询

①处可选择影响发布日期。②处可输入关键字进行查询。



3.3.5 服务操作

和本文【常规影像服务】服务操作类似，不再赘述。

3.4 影像集合服务

影像集合可以将多个影像服务合并成一个服务，当请求此服务的切片时，按顺序从影像集合中查找切片，若存在则返回，若不存在则查找下一个。类似于图层组。

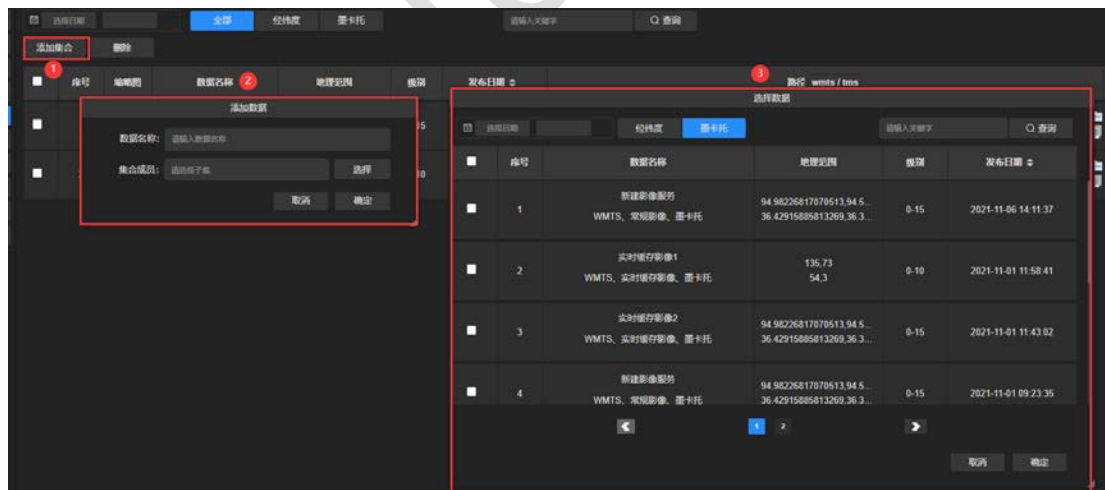
影像集合服务包含以下操作：①添加影像集合。②删除影像集合。③修改影像集合。④查询影像集合。⑤打开文件、预览、复制路径。



3.4.1 添加

点击添加集合按钮，输入数据名称，选择集合成员：对应③处，集合成员从所有的常规影像和实时缓存影像中勾选，勾选后点击确定即可选择集合成员。点击②处的确定即可完成集合的添加。添加后在修改里面调整影像成员的顺序。

注：一个集合中所有影像的投影类型必须一致。

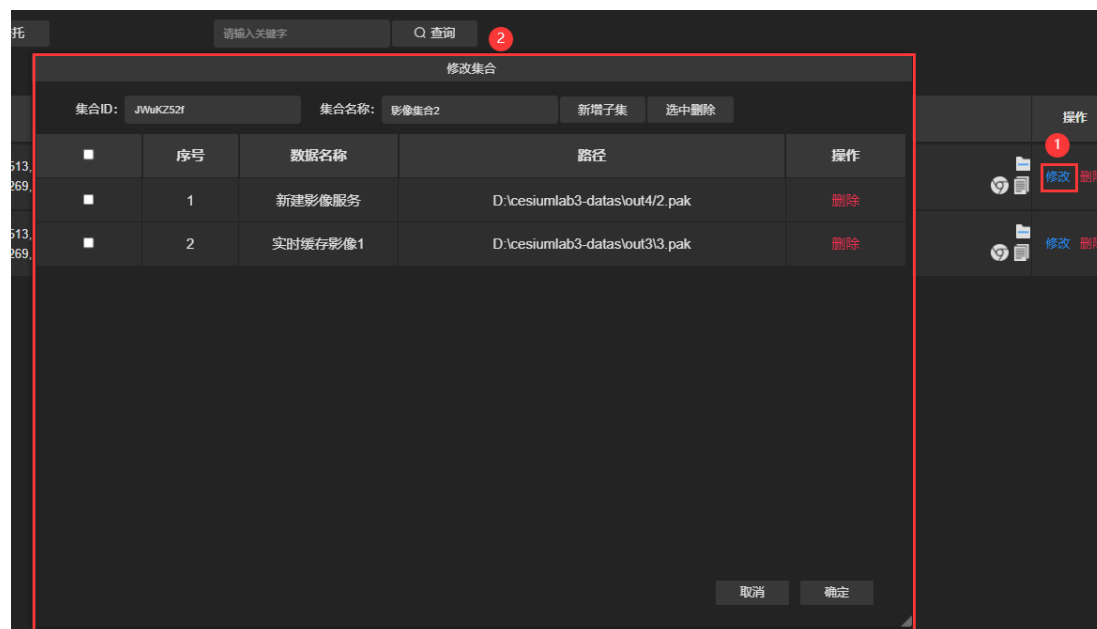


3.4.2 删除

参见“常规影像”中②删除常规影像。

3.4.3 修改

点击修改按钮后,可对集合进行新增子集、删除集合内元素、对集合内的元素进行排序。
新增子集: 同 1.添加影像集合。删除集合内元素: 可以选中删除,也可单个删除。对集合内的元素进行排序: 长按某一元素,上下拖动即可改变顺序。



3.4.4 查询

可通过发布日期、影像投影类型以及关键字进行查询。

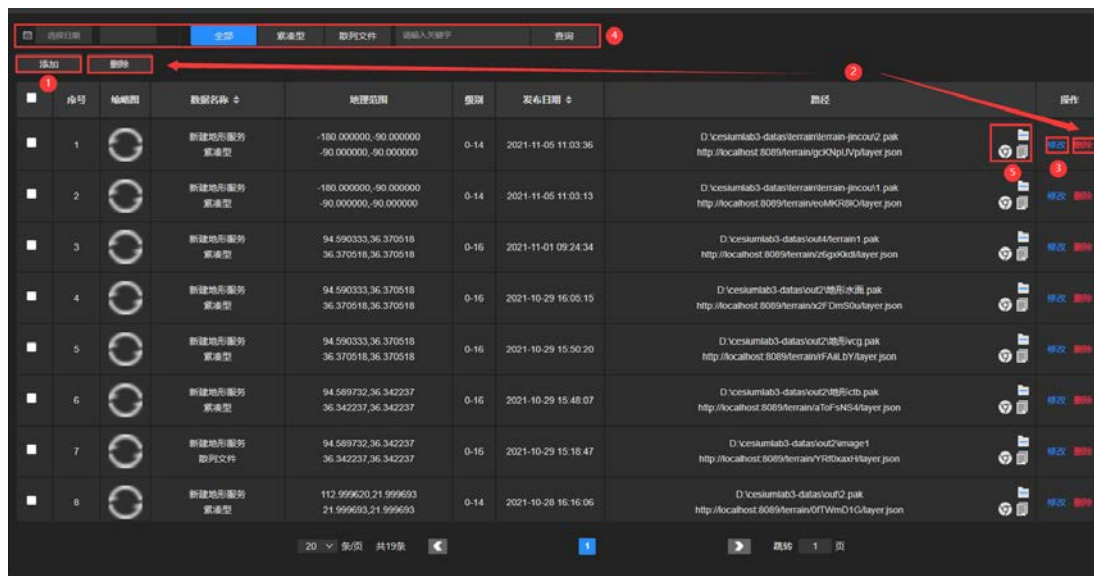


3.4.5 服务操作

类似本文常规影像服务的服务操作不再赘述

3.5 常规地形服务

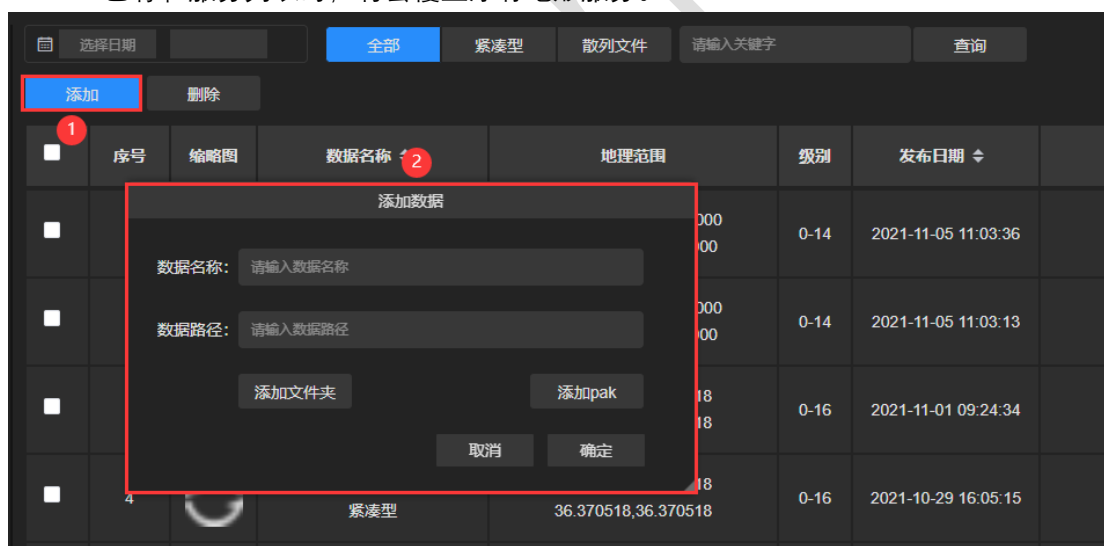
cesiumlab 地形切片工具生成的地形切片都可以在这里发布。功能包括: ①添加常规地形。②删除常规地形。③修改常规地形。④查询常规地形。⑤打开文件、预览、复制路径。



使用方法（功能）如下：

3.5.1 添加

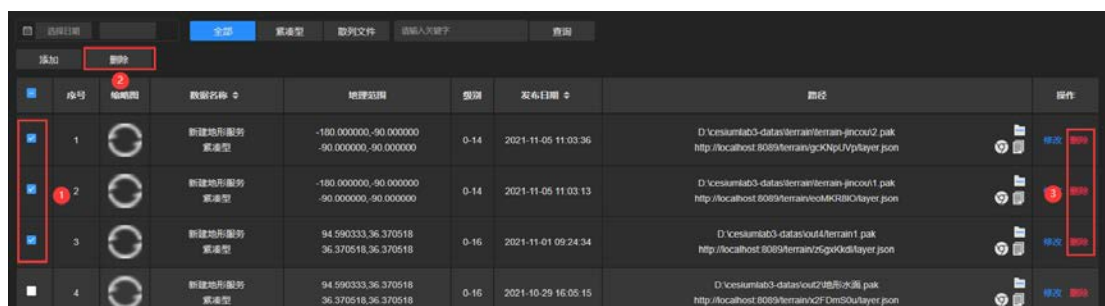
点击添加按钮，输入数据名称，添加数据路径，可选添加文件夹（散列文件）或添加 pak，点击确定，在服务列表中增加了该服务，完成发布。注：当该地形数据路径已存在服务列表时，将会覆盖原有地形服务。



3.5.2 删除

批量删除：对应①②，选中要删除的地形，点击删除。

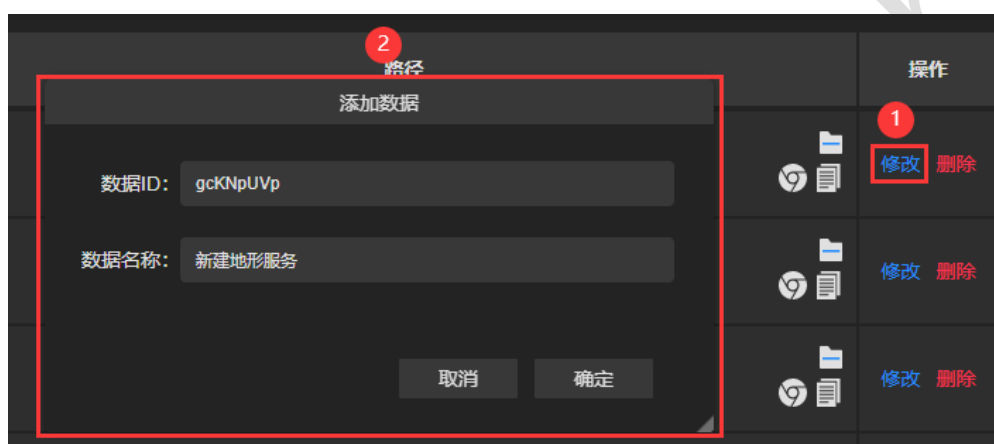
单个删除：对应③，点击删除。



序号	数据名称	地理范围	数据源	发布日期	路径	操作
1	新建地形服务	-180.000000,-90.000000 -90.000000,90.000000	0-14	2021-11-05 11:03:36	D:\cesiumlab3-data\terrain-jncou2.pak http://localhost:8089/terrain/gcKNpUVp/layer.json	修改 删除
2	新建地形服务	-180.000000,-90.000000 -90.000000,90.000000	0-14	2021-11-05 11:03:13	D:\cesiumlab3-data\terrain-jncou1.pak http://localhost:8089/terrain/colMKR8C/layer.json	修改 删除
3	新建地形服务	94.590333,36.370518 36.370518,36.370518	0-16	2021-11-01 09:24:34	D:\cesiumlab3-data\out44\terrain1.pak http://localhost:8089/terrain/z5G6kd/layer.json	修改 删除
4	新建地形服务	94.590333,36.370518 36.370518,36.370518	0-16	2021-10-29 16:05:15	D:\cesiumlab3-data\out2\地形水面.pak http://localhost:8089/terrain/k2F7Den8/layer.json	修改 删除

3.5.3 修改

点击修改，修改数据 ID 或者数据名称，点击确定完成修改。



数据ID: gcKNpUVp

数据名称: 新建地形服务

取消 确定

操作

修改 删除

修改 删除

修改 删除

3.5.4 查询

可通过发布日期、地形存储类型以及关键字进行查询。



选择日期

全部 紧凑型 散列文件

请输入关键字

查询

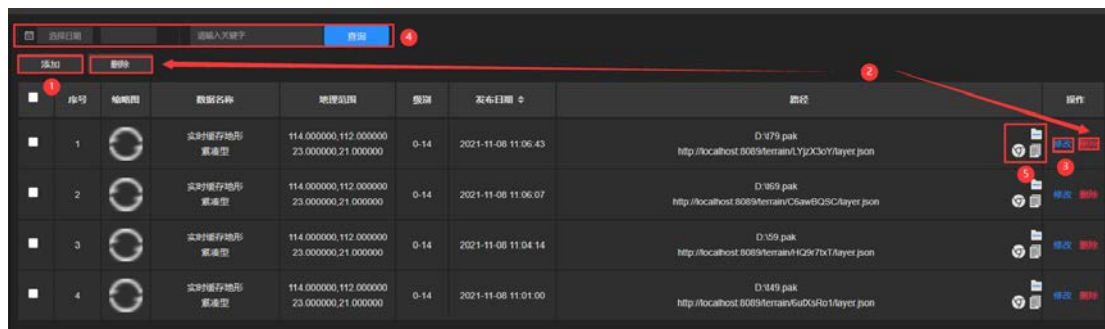
3.5.5 服务操作

类似本文常规影像服务的服务操作不再赘述

3.6 实时缓存地形服务

功能简介：实时缓存地形服务是对 Cesium 官方的地形服务进行代理和缓存（pak），当请求此服务的切片时，首先从缓存中查找地形切片，若不存在则从 Cesium 地形服务下载该切片，返回给客户端并缓存到 pak 中。

实时缓存地形服务包含以下操作：①添加实时缓存地形。②删除实时缓存地形。③修改实时缓存地形。④查询实时缓存地形。⑤打开文件、预览、复制路径。



使用方法（功能）如下：

3.6.1 添加

点击添加按钮后，输入数据名称、CesiumToken(注册地址：<https://cesium.com/ion/signin/tokens>)、数据路径，目前只支持紧凑 pak，点击确定完成添加。



3.6.2 删除

批量删除：对应①②，选中要删除的地形，点击删除。

单个删除：对应③，点击删除。

添加	删除	序号	数据名称	地理范围	类型	发布日期	路径	操作
		1	实时缓存地形 景类型	114.000000,112.000000 23.000000,21.000000	0-14	2021-11-08 11:06:43	D:\793.pak http://localhost:8089/terrain/LYj2X3oY/layer.json	修改 删除
		2	实时缓存地形 景类型	114.000000,112.000000 23.000000,21.000000	0-14	2021-11-08 11:06:07	D:\859.pak http://localhost:8089/terrain/C6aw9QSC/layer.json	修改 删除
		3	实时缓存地形 景类型	114.000000,112.000000 23.000000,21.000000	0-14	2021-11-08 11:04:14	D:\859.pak http://localhost:8089/terrain/HQ29r7bT/layer.json	修改 删除
		4	实时缓存地形 景类型	114.000000,112.000000 23.000000,21.000000	0-14	2021-11-08 11:01:00	D:\849.pak http://localhost:8089/terrain/6ufXsRo1/layer.json	修改 删除

3.6.3 修改

点击修改，可以修改数据 ID、数据名称、CesiumToken，点击确定完成修改。

2 路径

修改数据

数据ID: LYj2X3oY

数据名称: 实时缓存地形

CesiumToken: 请输入Cesium Token

取消 确定

操作

1 修改 删除

修改 删除

修改 删除

修改 删除

3.6.4 查询

①处可选择地形发布日期。②处可输入关键字进行查询。

选择日期

1

请输入关键字

2

查询

添加

删除

3.6.5 服务操作

和常规影像服务类似，不再赘述。

3.7 地形集合服务

地形集合可以将多个地形服务合并成一个服务，当请求此服务的切片时，按顺序从地形

集合中查找切片，若存在则返回，若不存在则查找下一个。类似于图层组。

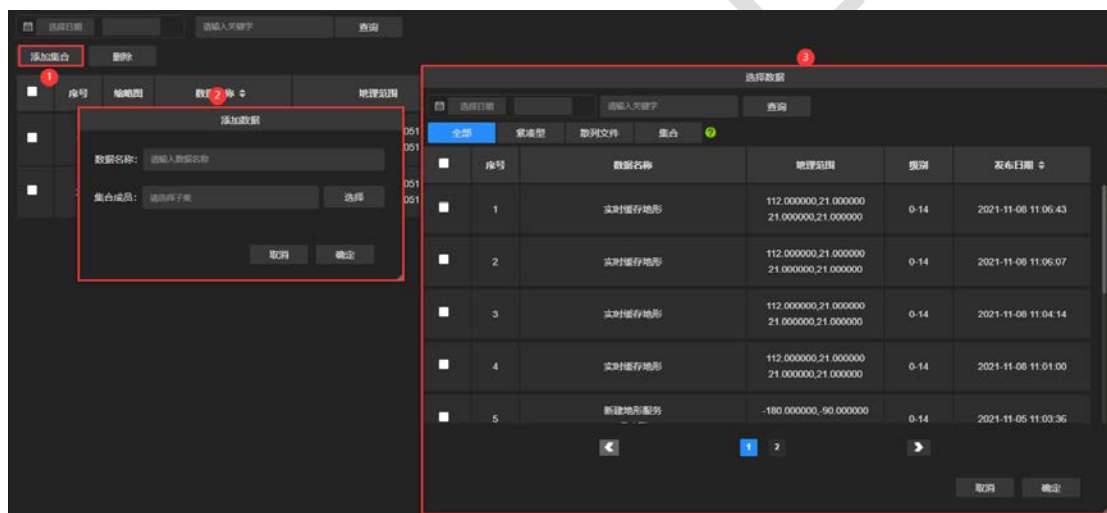
地形集合服务包含以下操作：①添加地形集合。②删除地形集合。③修改地形集合。④查询地形集合。⑤打开文件、预览、复制路径。



使用方法（功能）如下：

3.7.1 添加

点击添加集合按钮，输入数据名称，选择集合成员：对应③处，集合成员从所有的常规地形和实时缓存地形中勾选（内置查询功能），勾选后点击确定即可选择集合成员。点击②处的确定即可完成集合的添加。



3.7.2 删除

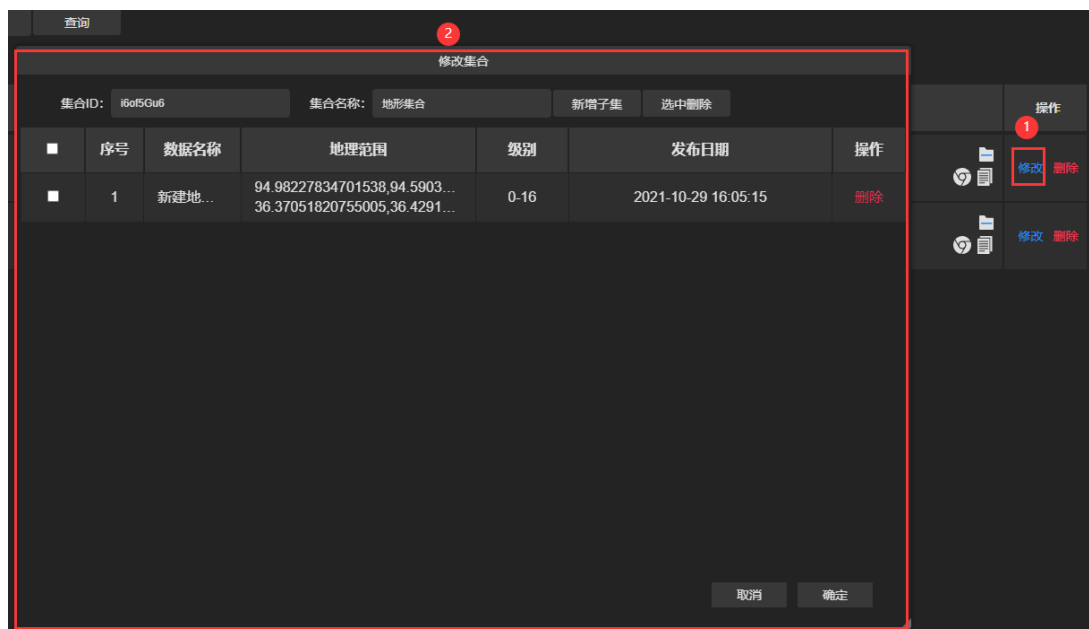
批量删除：对应①②，选中要删除的地形，点击删除。

单个删除：对应③，点击删除。



3.7.3 修改

点击修改按钮后，可对集合进行新增子集、删除集合内元素、对集合内的元素进行排序。新增子集：同 1.添加地形集合。删除集合内元素：可以选中删除，也可单个删除。对集合内的元素进行排序：长按某一元素，上下拖动即可改变顺序。



3.7.4 查询

可通过发布日期以及关键字进行查询。



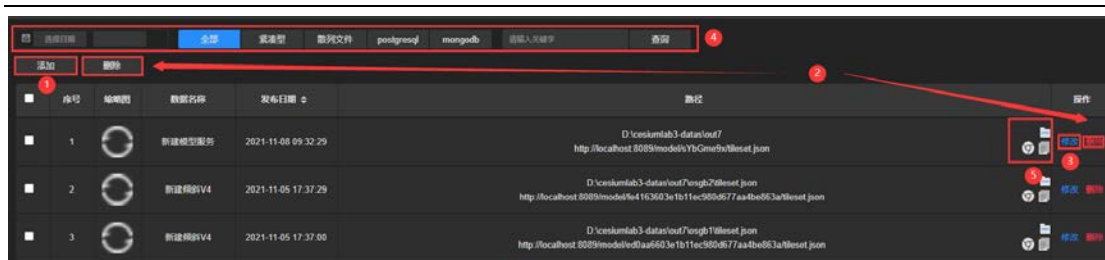
3.7.5 服务操作

和常规影像服务类似，不再赘述。

3.8 3dtiles 服务

用于发布 3dtiles 数据。在数据处理中，点云切片、实例模型切片、倾斜模型切片和通用模型切片的成果数据都在这里发布。

3dtiles 服务包含以下操作：①添加 3dtiles。②删除 3dtiles。③修改 3dtiles。④查询 3dtiles。⑤打开文件、预览、复制路径。



3.8.1 添加

点击添加按钮,输入数据名称,选择添加的 3dtiles 文件,可选择三种类型进行添加:紧凑型切片 clt、散列切片中的 tileset.json 以及下载的谷歌数据 glt(目前只支持内部使用), 点击确定完成添加。



3.8.2 删除

参见“常规影像”中②删除常规影像。

3.8.3 修改

参见“常规影像”中③修改常规影像。

3.8.4 查询

可通过发布日期、3dtiles 存储类型以及关键字进行查询。



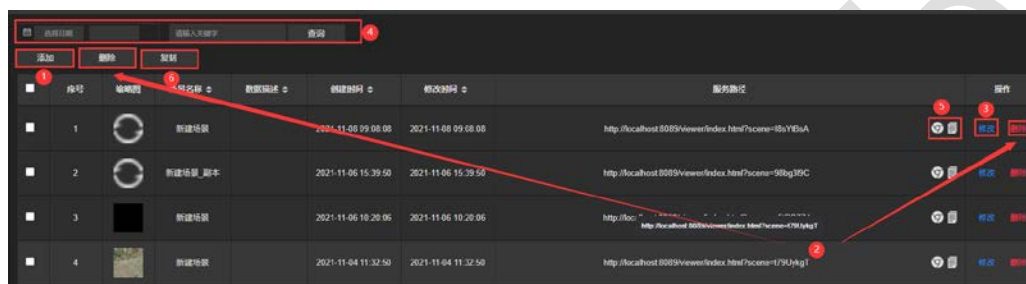
3.8.5 服务操作

参见“影像服务”中⑤打开文件、预览、复制路径。

3.9 场景服务

cesiumlab 支持将三维可视页面配置的场景内容保存下来，并发布成场景服务，以方便保存，编辑和分享。

场景服务包含以下操作：①添加场景。②删除场景。③修改场景。④查询场景。⑤预览、复制路径。⑥复制场景。



3.9.1 添加

点击添加按钮后，跳转至三维可视页面，可点击保存按钮进行场景添加。





3.9.2 删除

参见“常规影像”中②删除常规影像。

3.9.3 修改

参见“常规影像”中③修改常规影像。

3.9.4 查询

参见“实时缓存影像”中的④查询实时缓存影像。

3.9.5 预览、复制路径

参见“影像服务”中⑤打开文件、预览、复制路径。

3.9.6 复制场景

选中某一个要复制的场景，点击复制即可。

日期范围

请输入关键字

查询

添加

删除

复制

	序号	缩略图	场景名称	数据描述	创建时间	修改时间	服务路径
	1		新建场景_副本		2021-11-06 15:39:50	2021-11-06 15:39:50	http://localhost:8089/viewer/index.html?scene=98bg39C
	2		新建场景		2021-11-06 10:20:06	2021-11-06 10:20:06	http://localhost:8089/viewer/index.html?scene=z5DB7ZJ

3.10lod 模型库服务

lod 模型库用于制作实例模型（大量且相同的模型），参考本文【数据处理】【实例模型切片】【lod 模型】中的资料。

包含以下操作：①添加 lod 模型。②删除 lod 模型。③查询 lod 模型。④打开文件、复制路径。⑤帮助。



3.10.1 添加

点击添加按钮，选择数据路径，点击确定完成添加。



3.10.2 删除

参见“常规影像”中②删除常规影像。

3.10.3 查询

①处可选择模型发布日期。②处可输入模型 id。③处可输入关键字进行查询。④处为模型 id。



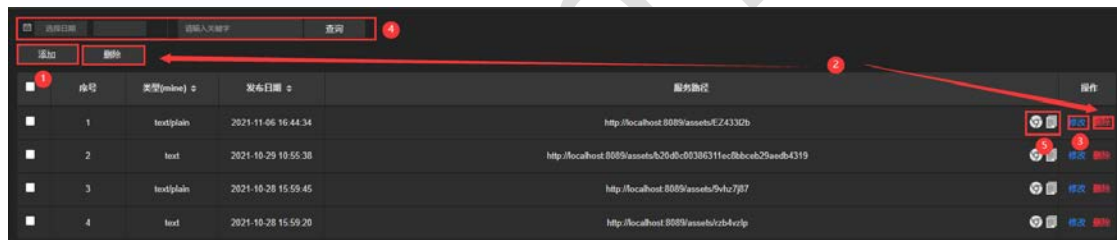
3.10.4 打开文件、复制路径

参见“影像服务”中⑤打开文件、预览、复制路径。

3.11 资源服务

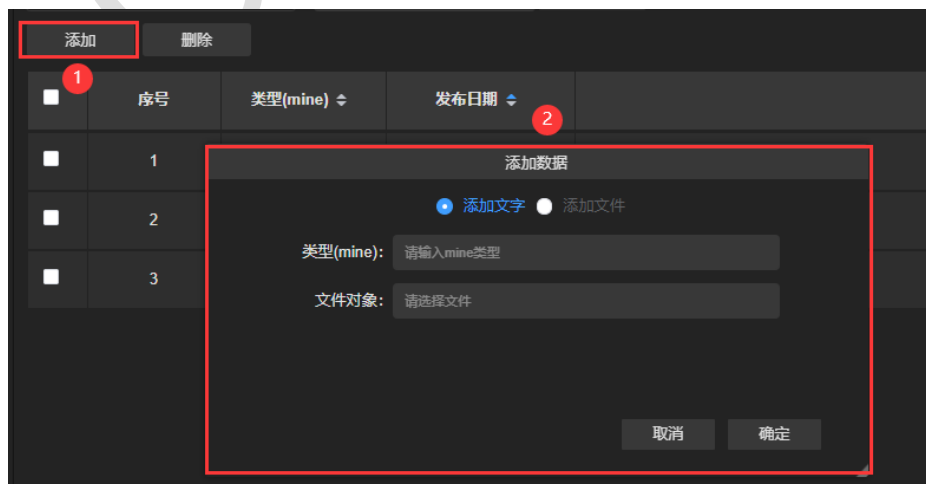
资源服务可以很方便的把一段文字或者本地文件，图片，视频等发布为服务，方便在三维场景中使用。

包含以下操作：①添加资源。②删除资源。③修改资源。④查询资源。⑤预览、复制路径。



3.11.1 添加

点击添加按钮，选择添加文字或添加文件。输入 mine 类型，选择文件对象，点击确定完成添加。

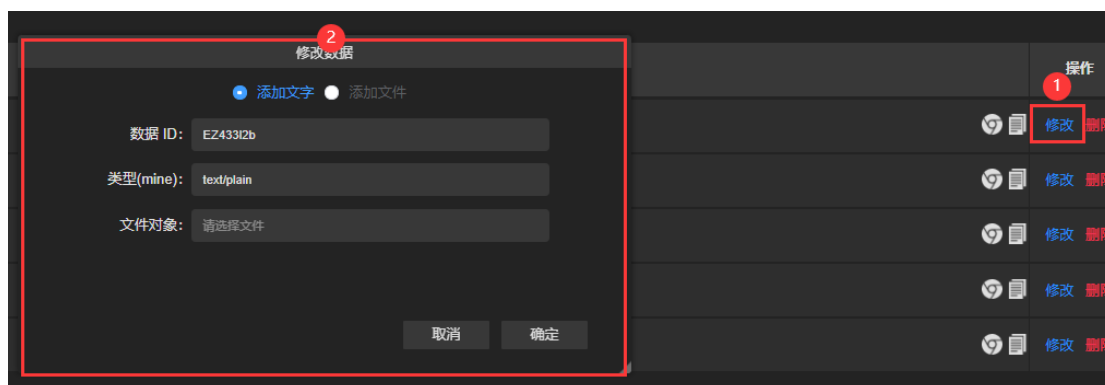


3.11.2 删除

参见“常规影像”中②删除常规影像。

3.11.3 修改

可修改数据 ID，类型，文件对象。



3.11.4 查询

可通过发布日期以及关键字进行查询。



3.11.5 预览、复制路径

参见“影像服务”中⑤打开文件、预览、复制路径。

4 三维可视

三维可视基于我们另一个产品 EarthSDK，详见 EarthSDK 的用户手册。