

三维地理设计技术新进展

三维地理设计支撑更多行业应用

(部分图片来源于网络)



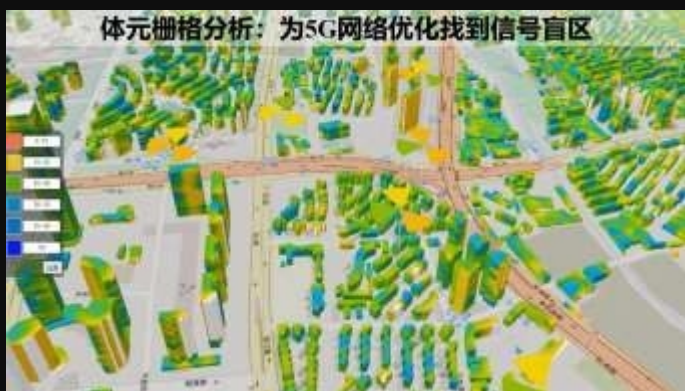
城市设计



交通设计



输电线路设计



通信行业

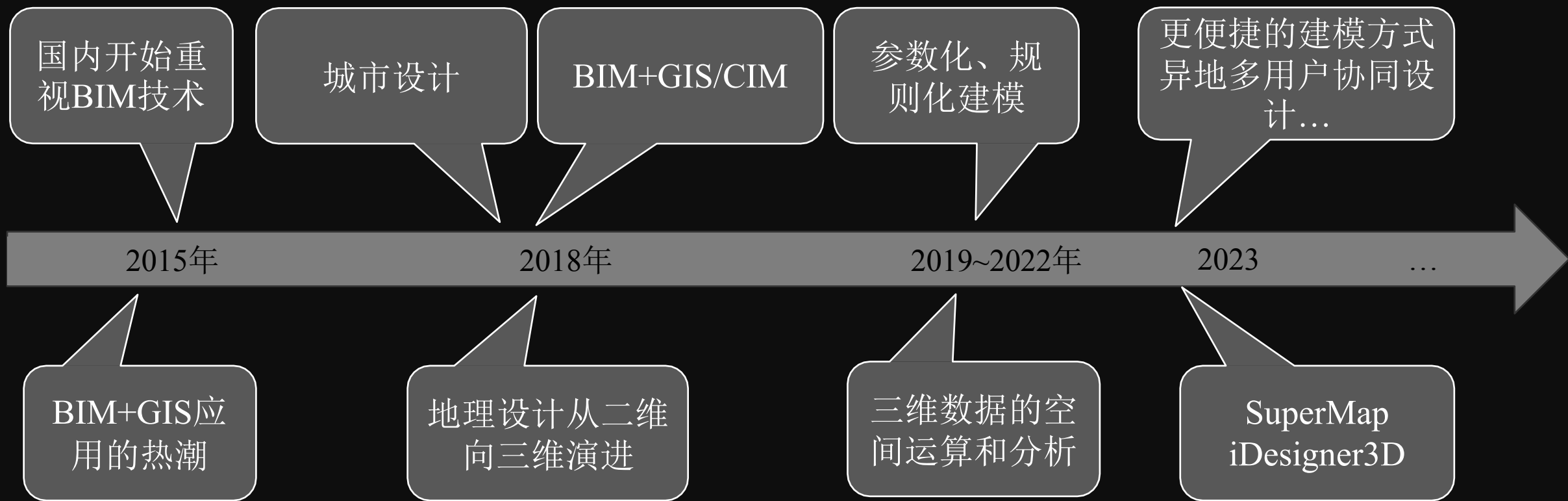


管网/管廊/巷道



水利设计

三维地理设计技术演进



三维地理设计产品演进



三维地理设计相关能力

历史版本

- 支持常见三维模型格式及BIM数据直接导入
 - DAE、FBX、GLTF、MOD、OBJ、OFF、STL
 - 3DXML、RVM、IFC、GIM、Geo3DML
- 规则化建模
 - 参数化构建三维体模型（CSG）
 - 根据三维点集生成产权体、带属性的三维点集生成TIM（不规则四面网格体）
 - 模型对象的裁剪、切割、组合等
- 空间运算和分析
 - 模型对象的布尔运算、空间关系判断
 - 地形和倾斜摄影模型的裁剪、挖洞、镶嵌等
 - 在地形和倾斜摄影模型上提取高程
 - 三维模型拓扑检查
- 在线三维地理设计能力
 - ...

2024版本

- 增强三维数据接入能力
 - 直接支持RVT、DGN格式数据的高效导入，并且保留数据的结构层级关系
 - 提升GIM数据的导入性能
- 提升规则化建模能力
 - 完善参数化建模算子
 - 新增根据建筑矢量快速生成带屋顶类型的建筑模型
 - 新增根据道路中线及横截面快速生成全要素道路模型
- 空间运算和分析
 - 优化日照分析，提供一种全新的日照分析模式
- 在线三维地理设计
 - 支持更多的数据源类型
 - 支持建模成果保存到模型数据集（iserver提供数据服务）
 - 完善交互编辑操作
 - 支持添加标注、PBR材质及添加和设置更多的场景元素

更强大的规则化建模能力

主要内容



1、参数化三维体模型 *update*



2、道路方案设计 workflow (GPA) *update*



3、程序化建模工具：批量构建城市级建筑三维模型 *new*



4、程序化建模工具：快速构建城市道路及道路交叉口 *new*

1.1 参数化三维体模型



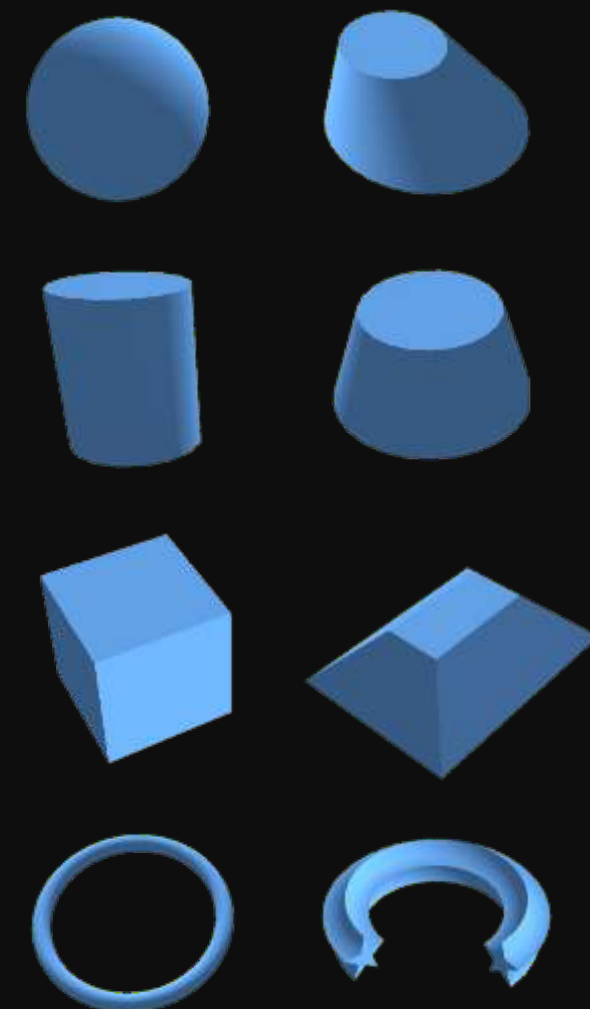
参数化三维体 → IFC、GIM标准参数化对象的对应关系图

参数化三维体	IFC	GIM
球体	IfcSphere	Sphere
立方体	IfcBlock	Cuboid
棱台体	IfcBrep	Table
楔形体	IfcBrep	OffsetRectangularTable
棱锥体	IfcBrep	Table
椭圆体	IfcBrep	/
椭圆环体	IfcBrep	EllipticalRing
矩形环体	IfcBrep	RectangularRing
弯折圆柱体	IfcBrep	BendingCylindrical
斜口圆柱	IfcBrep	/
圆锥体	IfcRightCircularCone	TruncatedCone
圆柱体	IfcRightCircularCylinder / IfcExtrudedAreaSolidTapered	Cylinder
圆环体	IfcRevolvedAreaSolid	Ring
圆台体	IfcExtrudedAreaSolidTapered	TruncatedCone
旋转体	IfcRevolvedAreaSolid	/
拉伸体	IfcExtrudedAreaSolid	StretchedBody
放样体	IfcBrep	/
...

GIS



BIM

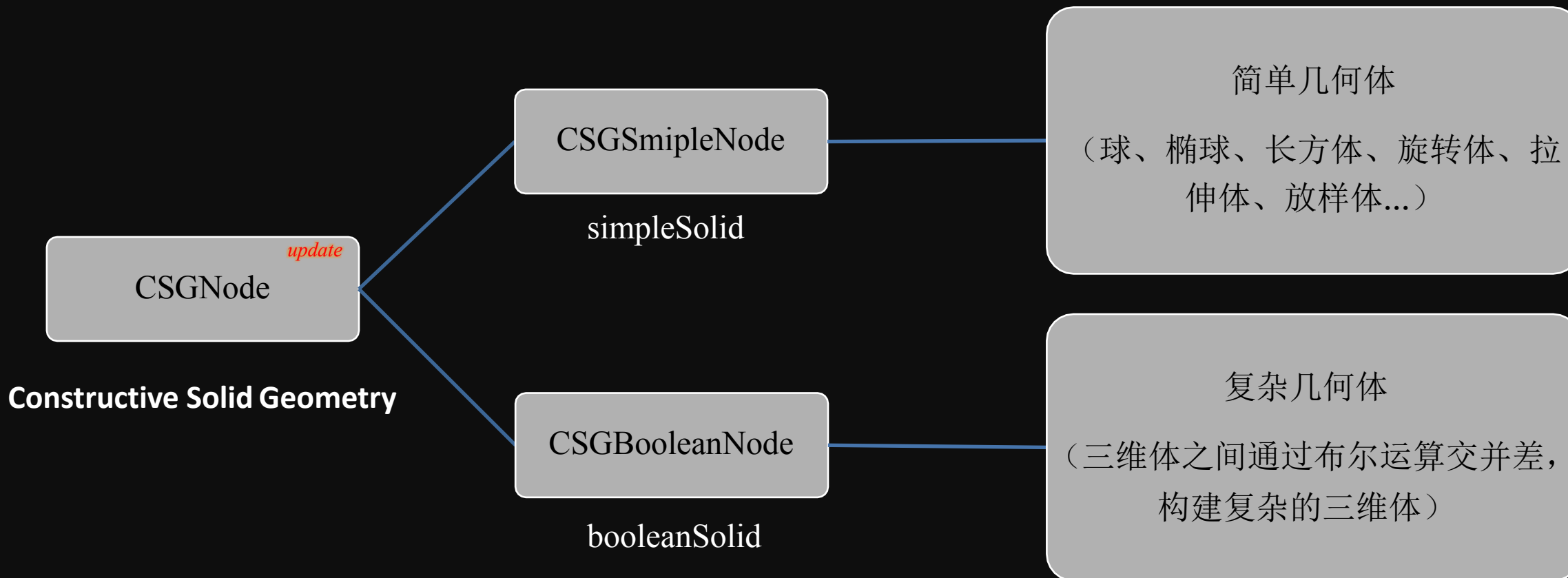


完善相关算子

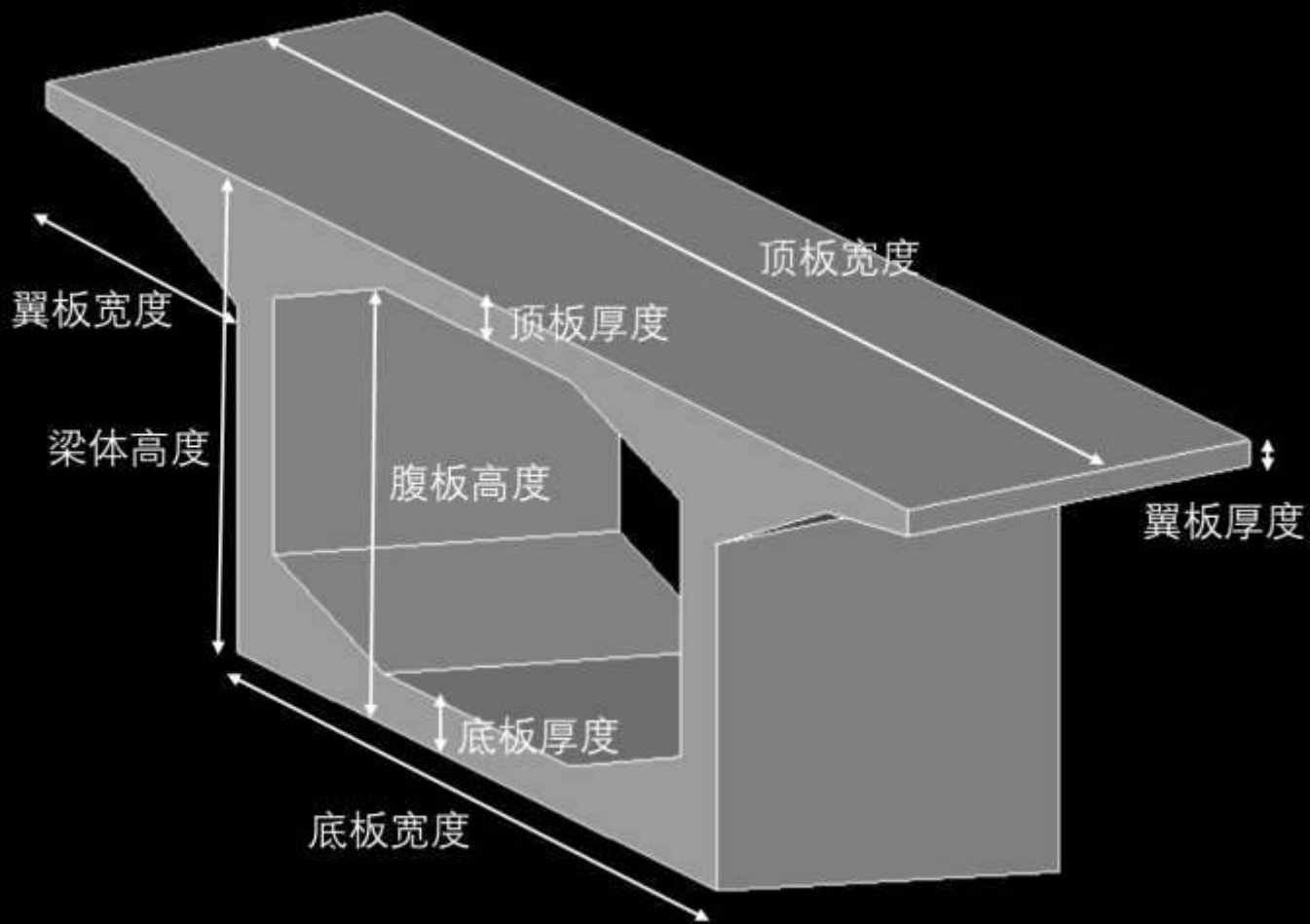
- 输入输出
 - 新增生成参数化流程模板算子
 - Revit软件中的参数化对象输出CSG文件
 - 将生成的CSG文件生成GPA流程模型
 - 新增输出模型流
 - 将生成的模型，输出成模型流传给web使用
- 运算
 - 新增创建复合对象算子
 - 将多个参数化对象合并为一个



实体几何构造法(CSG) → 构建三维体对象



如何生成桥的箱梁？



```
class Point3:
    def __init__(self, x, y, z):
        self.x = x
        self.y = y
        self.z = z

    def toString(self):
        return str(self.x) + ";" + str(self.y) + ";" + str(self.z)

import math
class BeamProfile:
    # beamVLength 梁纵向长度
    # beamHeight 梁体高度
    # flangeThickness 翼板厚度
    # flangeWidth 翼板宽度
    # floorWidth 底板宽度

    # roofThickness 顶板厚度
    # floorThickness 底板厚度

    def execute(self, keyargs):
        beamVLength = float(keyargs['beamVLength'])
        beamHeight = float(keyargs['beamHeight'])
        floorWidth = float(keyargs['floorWidth'])

        flangeWidth = float(keyargs['flangeWidth'])
        flangeThickness = float(keyargs['flangeThickness'])

        innerThickness = keyargs['innerThickness']

        #外圈和内圈点集顺序需要相反
        outPts = []
        outPts.append(Point3(-floorWidth / 2.0, 0, 0))
        outPts.append(Point3(floorWidth / 2.0, 0, 0))
        outPts.append(Point3(floorWidth / 2.0, 0, beamHeight - flangeThickness - 0.5))
        outPts.append(Point3((floorWidth / 2.0) + (flangeWidth / 2.0), 0, beamHeight - flangeThickness))
        outPts.append(Point3(beamVLength / 2.0, 0, beamHeight - flangeThickness))
        outPts.append(Point3(beamVLength / 2.0, 0, beamHeight))
        outPts.append(Point3(-beamVLength / 2.0, 0, beamHeight))
        outPts.append(Point3(-beamVLength / 2.0, 0, beamHeight - flangeThickness))
        outPts.append(Point3(-(floorWidth / 2.0) - (flangeWidth / 2.0), 0, beamHeight - flangeThickness))
        outPts.append(Point3(-floorWidth / 2.0, 0, beamHeight - flangeThickness - 0.5))
        outPts.append(Point3(-floorWidth / 2.0, 0, 0))
```

构建拉伸体三维体模型（箱梁）

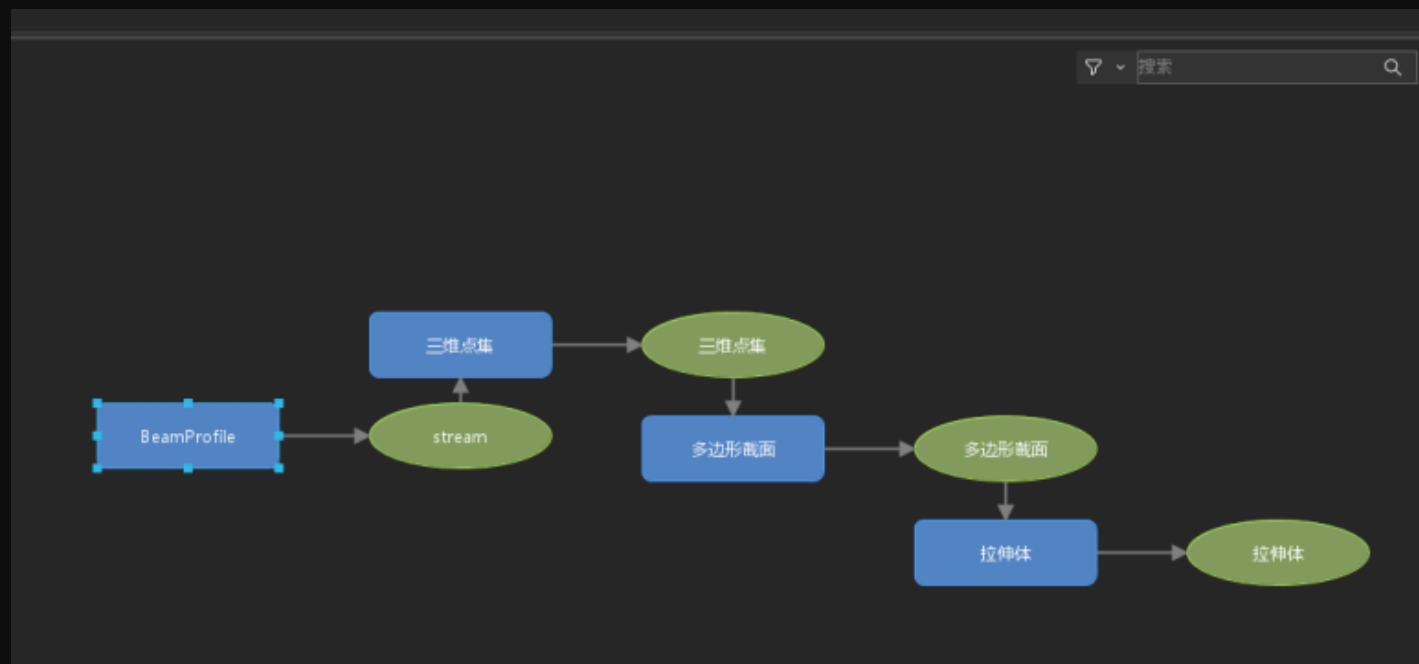
Python脚本生成
复杂截面的三维
点串

三维点串构建
多边形截面

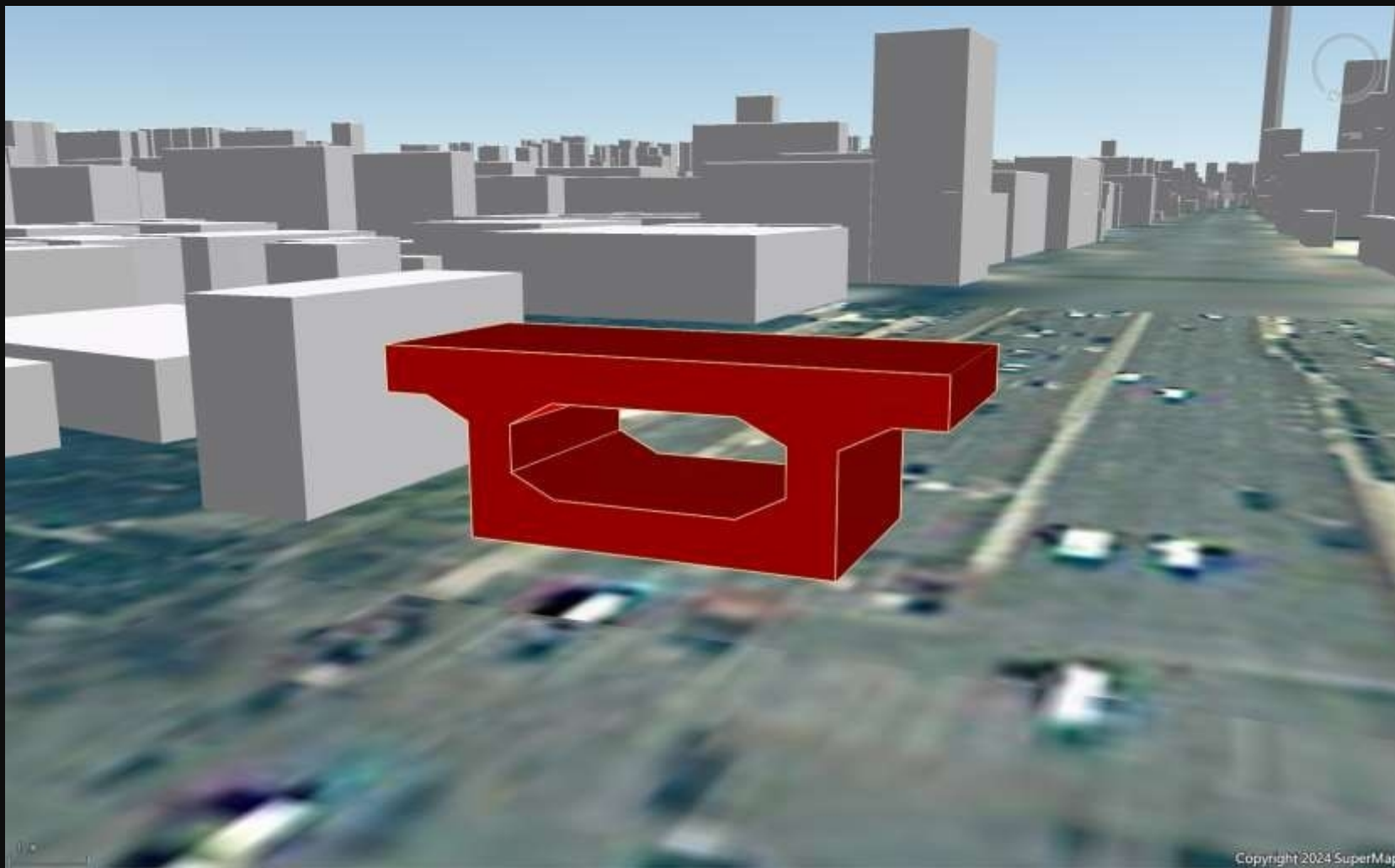
构建拉伸体



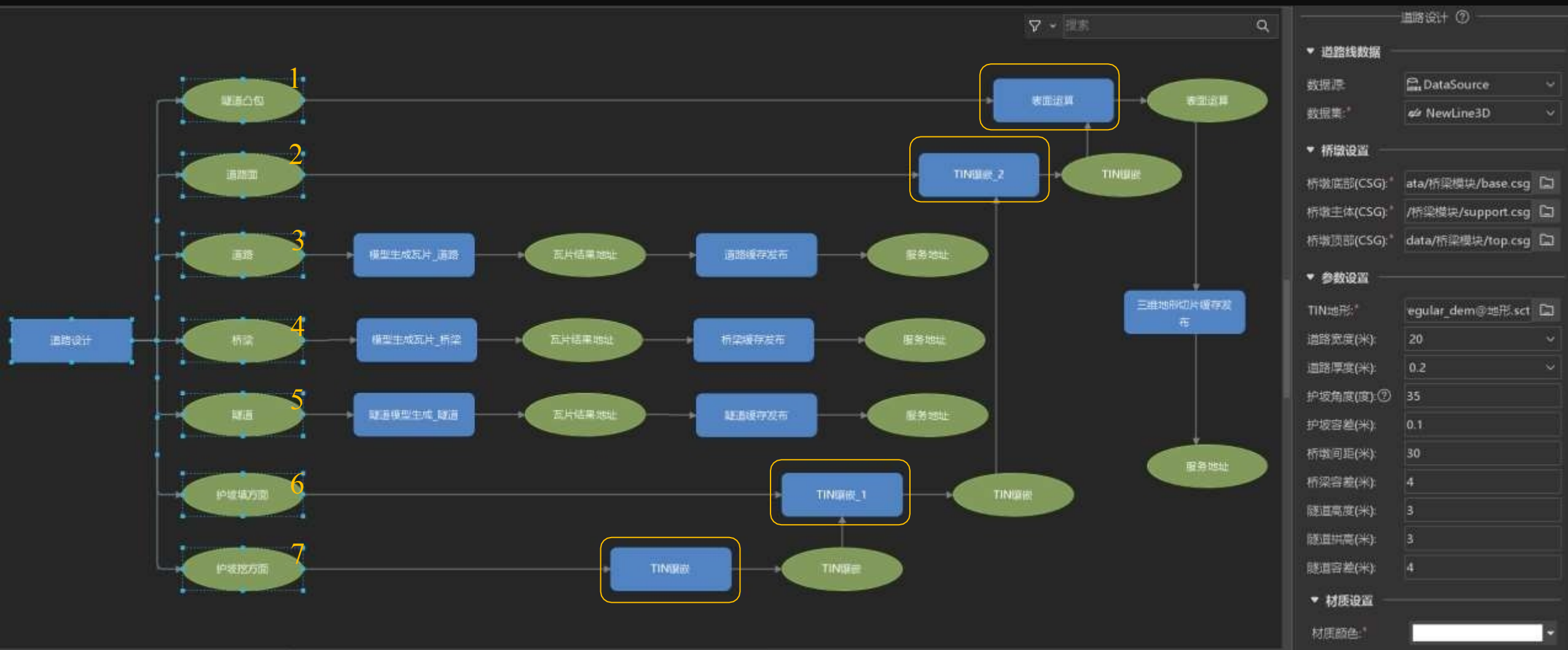
```
BeamProfile.py X
> Designer > 3D模型 > 梁
6 # flangeWidth 翼板宽度
7 # floorWidth 底板宽度
8
9 # roofThickness 顶板厚度
10 # floorThickness 底板厚度
11 # sideThickness 侧板厚度
12
13 import math
14 class BeamProfile(object):
15     def execute(self, keyargs):
16         beamLength = keyargs['beamLength']
17         beamHeight = keyargs['beamHeight']
18         floorWidth = keyargs['floorWidth']
19         flangeWidth = keyargs['flangeWidth']
20         flangeThickness = keyargs['flangeThickness']
21         roofThickness = keyargs['roofThickness']
22         floorThickness = keyargs['floorThickness']
23         sideThickness = keyargs['sideThickness']
24
25         chamfer = 30 #倒角
26         pts = [Point3D(-floorWidth / 2.0, 0, 0), Point3D(floorWidth / 2.0, 0, 0)]
27         pts.add(Point3D(floorWidth / 2.0, 0, beamHeight - flangeThickness - (flangeWidth * math.tan(chamfer) / 2.0))
28         pts.add(Point3D((floorWidth / 2.0) + (flangeWidth / 2.0), 0, beamHeight - flangeThickness))
29         pts.add(Point3D(beamLength / 2.0, 0, beamHeight - flangeThickness))
30         pts.add(Point3D(beamLength / 2.0, 0, beamHeight))
31
32         pts.add(Point3D(-beamLength / 2.0, 0, beamHeight))
33         pts.add(Point3D(-beamLength / 2.0, 0, beamHeight - flangeThickness))
34         pts.add(Point3D(-floorWidth / 2.0 - (flangeWidth / 2.0), 0, beamHeight - flangeThickness))
35         pts.add(Point3D(-floorWidth / 2.0, 0, beamHeight - flangeThickness - (flangeWidth * math.tan(chamfer) / 2.0))
36         pts.add(Point3D(-floorWidth / 2.0, 0, 0))
```



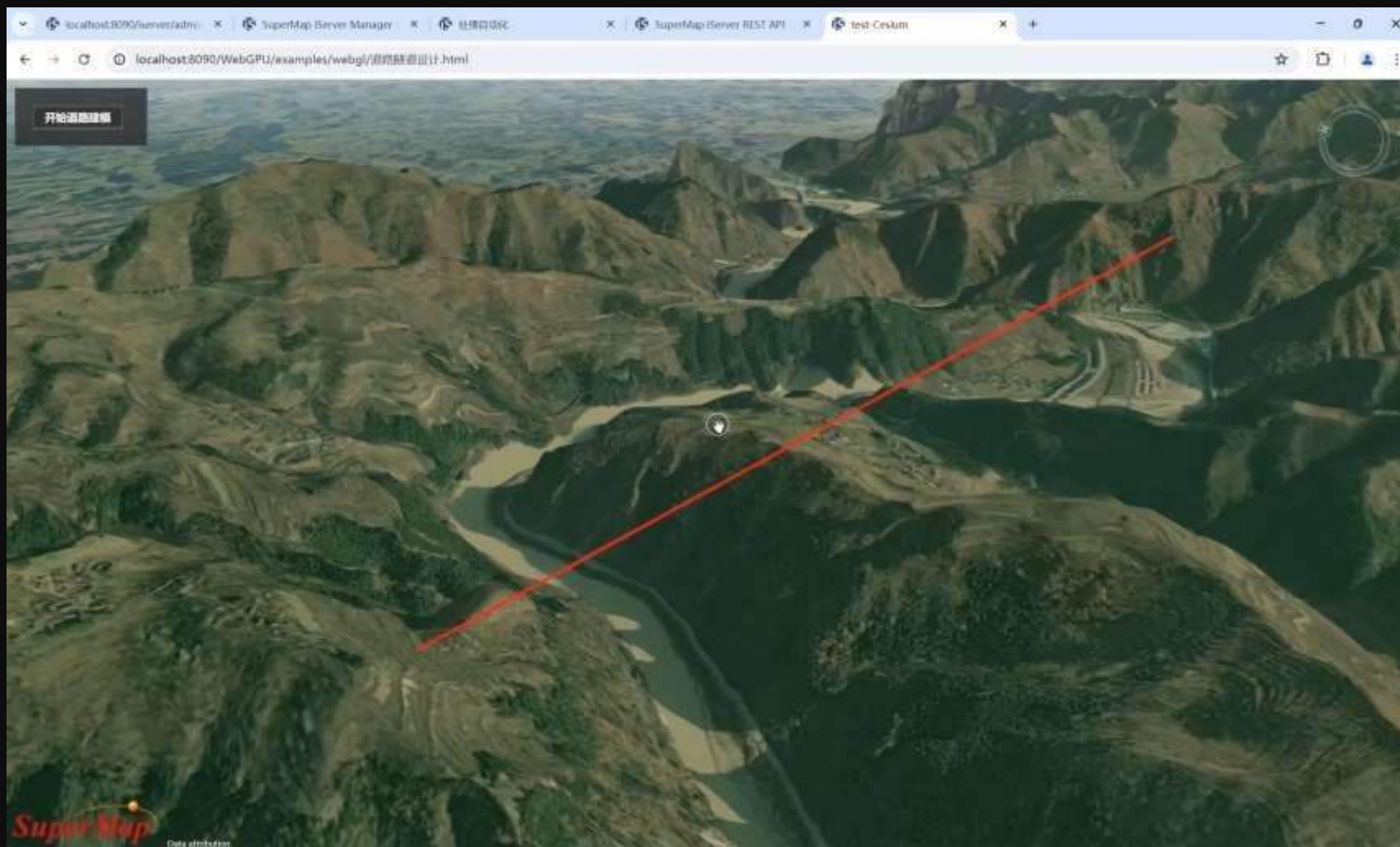
构建拉伸体三维体模型（箱梁）



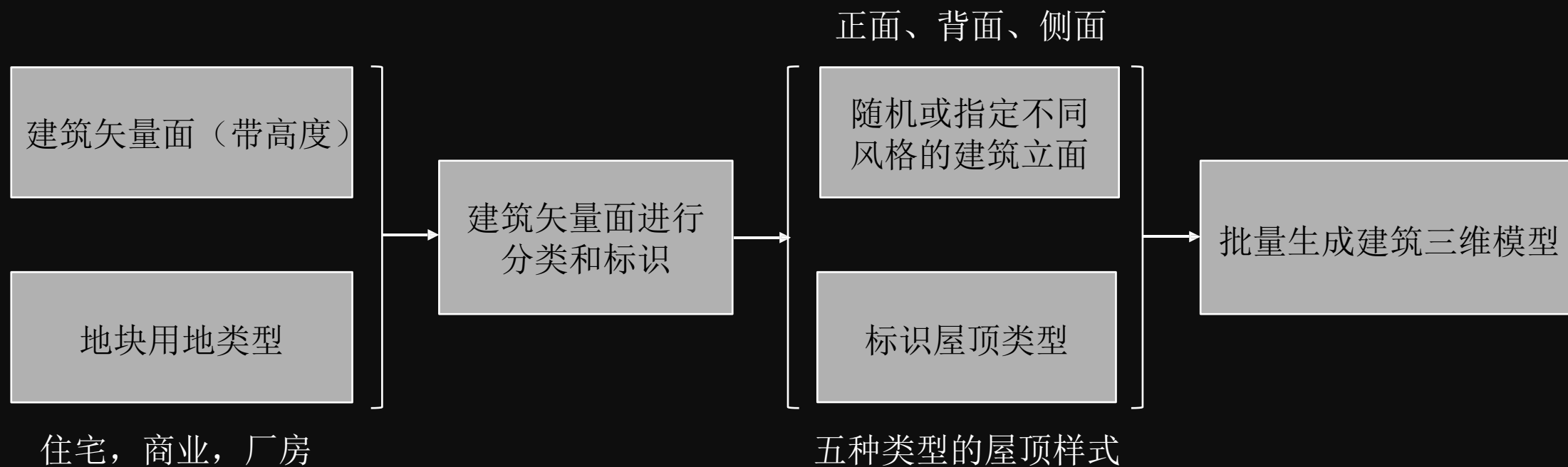
1.2 道路方案设计 workflow (GPA)



应用：公路方案设计效果展示



1.3 程序化建模工具：批量构建城市级建筑三维模型



基于矢量面批量构建城市级建筑三维模型

自动生成更多建筑结构，更多细节

- 5种屋顶类型：挑檐平屋顶（Flat），女儿墙平屋顶（Parapet），单坡坡屋顶（Shed），双坡坡屋顶（Gable），四坡坡屋顶（Hip）
- 女儿墙、挑檐、腰线等

内置多种材质种类，即拿即用

- 提供住宅，商业，厂房三大类型的材质
- 74套普通材质格式的建筑立面
- 12套PBR格式的建筑立面

提升建模性能

- 城市百万级别矢量面建模用时6小时

建筑类型与细节



住宅（Residential）精细模型



住宅（Residential）精细模型



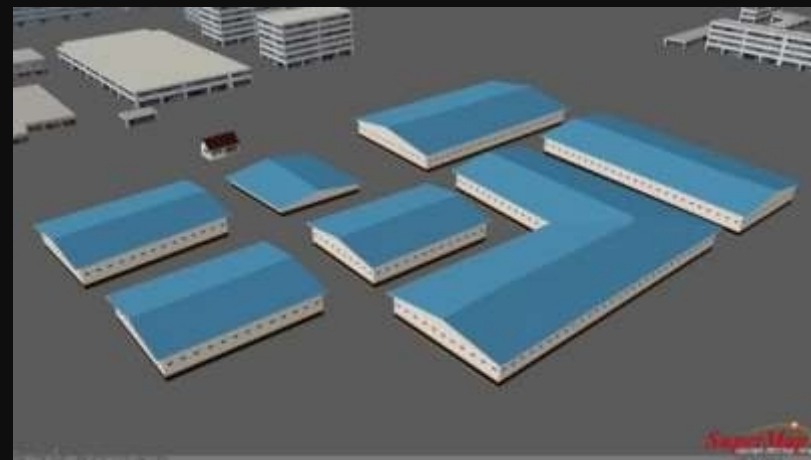
商业（Commercial）



住宅（Residential）精细模型



住宅（Residential）基础模型



厂房（Industrial）

能否在倾斜摄影三维模型上提取单体化建筑模型？



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/068025043101006124>