FME 在二维视图下面要素拔高的应用

摘要

本次实验的目的是基于简单面要素在 FME 中实现二维视图下的拔高。基本思 路是通过在平面横纵坐标上的平移,得到构建 2.5D 面要素的侧面线、底面线及 顶面线要素。侧面线通过提取面折点并对这些折点编号后平移折点,连接相同编 号的折点,从而得到侧面线。底面线通过对角线与多边形边的相交关系,对相交 线打断并筛选出多边形边从而得到底面线。顶面线则由底面线平移得到。最后以 侧面线、底面线及顶面线构建 2.5D 面。

1 实验及数据简介

1.1 实验简介

在本实验前,我接触了一个电子地图制图类的项目。用户向开发者提出了对 场景内建筑物进行拔高的要求,并要求数据量小、显示速度快。开发者基于用户 的要求对建筑物进行了简易的 2.5D 拔高,而非通过三维场景中搭建三维模型来 实现,既满足了建筑物的三维即视感也满足数据量小、显示速度快的要求。作为 FME 爱好者,我设想了如何在 FME 实现上述功能,以下是对于本实验的介绍与 解释。

1.2 数据简介

为了使实验过程简洁,只需要准备数个面要素即可(可直接在 ArcGIS 中随机绘制面要素)。

2 程序设计及实施

2.1 程序设计

1) 将准备的面要素作为底面,根据底面的折点生成点要素。

2)将上述的点要素在水平方向及竖直方向各平移一定的距离,将平移后点要素与原点要素两两连线,构成侧面线。

3) 将准备的面要素根据折点生成线要素并在水平方向及竖直方向各平移一定 的距离,生成底面线要素和顶面线要素。

4) 根据侧面线要素、底面线要素及顶面线要素构面,生成 2.5D 的面要素。

2.2 程序实施

2.2.1 面要素折点转点

将面要素折点转为点要素,首先需要得到面要素所有折点的坐标信息,再将 坐标信息转为点要素,从而得到面的各个折点。

1) 获取面要素上所有点的坐标:

使用转换器 CoordinateConcatenator,由于程序只在平面上执行,故只获取 X 坐标与 Y 坐标。如图 2-1 是该转换器的基本设置。

😤 CoordinateConcatenator Paramete	ers X
Transformer	
Transformer Name:	CoordinateConcatenator
Parameters	
Coordinate Delimiter:	1
Coordinate Element Delimiter:	, 🔍
Coordinate Element(s) To Include:	х ү \cdots 💌
Full Precision:	[¥₀ ▼] ▼
Coordinate Attribute:	_coordinates
Help Defaults V	OK Cancel

图 2-1

通过转换器 CoordinateConcatenator 得到了面要素所有折点的坐标,但是坐标 是以","分割 X 坐标与 Y 坐标,以"|"分割不同点之间的坐标,储存在同一 字段下(字段名为_coordinates),故需要将该字段裂开并且将 X 坐标,Y 坐标分 别储存到不同字段中。

	Id	图号 ▲	_coordinates
1	. 0	1	200001.69333672151,199993.64998730272 200008.14918296412,199995.02582338639 200007.30251460522,
2	0	2	$200020.53170773014, 199998.62416391633 200027.62255524471, 200000.95250190422 200029.95089323632, {1, 1, 1, 2, 2, 2, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,$
5	0	3	200008.25501650944,200005.7150114309 200012.38252476417,200005.82084497437 200011.43002285995,20

图 2-2

2) 将不同点的坐标裂开:

裂开属性使用转换器 AttributeSplitter。裂开后的不同坐标仅是存储在 FME 列表(列表名为_list)中,为了方便后续操作,应将列表中"_list"中的坐标信息暴露出。暴露属性用转换器 ListExploder。

😤 AttributeSplitter Parameters	23
Transformer	
Transformer Name: AttributeS	plitter
Parameters	
Attribute to Split: 🔷 _coordi	nates 🔻 💌
Delimiter or Format String:	•
Trim Whitespace: Both	• •
List Name: _list	
Drop Empty Parts: No	•
Help Defaults V OK	Cancel



🙊 ListExploder Parameters	×
Transformer	
Transformer Name:	ListExploder
Parameters	
List Attribute:	↓ _list{}
Element Index Attribute:	_element_index
Attribute Accumulation	
Accumulation Mode:	Merge Incoming List 🔻
Conflict Resolution:	Use Incoming List 💌 💌
Prefix:	
Help Defaults -	OK Cancel

图 2-4

在 "_list" 列表中, 面要素的点坐标信息以 "_list{1}"、 "_list{2}" 等分别储存, 而暴露出列表信息的要素中名为 "_element_index" 的字段正是 "_list{0}"、 "_list{1}" 中的 0、1, 可理解为同一面的折点的编号。

在暴露出"_list"列表后,生成了一些新要素,该新要素中储存有面的各个 折点的坐标信息,则该新要素可视为每个折点与其原始面要素的复合要素。对这 些要素编号,可视为对各折点编号,编号使用转换器 Counter,起始编号设为1。

3) 将相同点的 X、Y 坐标裂开:

与 2) 中裂开不同点的坐标的思路相似,但分割符号应设置为","。得到的 X、Y 坐标分别储存到新列表中,列表名为别为"_list1{0}"、"_list1{1}"。那么, 经过"_list1"列表暴露后生成的新要素中"_element_index"为0的是带有 X 坐标信息的要素,"_element_index"为1 的是是带有 Y 坐标信息的要素。

🙊 AttributeSplitter Parameters	×
Transformer	
Transformer Name:	AttributeSplitter_2
Parameters	
Attribute to Split:	🌗 _list 💌 💌
Delimiter or Format String:	,
Trim Whitespace:	Both
List Name:	_list1
Drop Empty Parts:	No •
Help Defaults -	OK Cancel

图 2-5

_element_index	_count 2 <u>≜</u>	_list1
0	1	200001.69333672151
1	1	199993.64998730272
0	2	200008.14918296412
1	2	199995.02582338639
0	3	200007.30251460522
1	3	199990.58081449755
0	4	200000.6350012701
1	4	199989.41664550081
0	5	200001.69333672151

图 2-6

4) 分别将 X、Y 坐标赋到新字段中:

如图 2-6, X、Y 坐标都储存在"list1"字段中,但通过"_element_index" 可进行区分,故为了将 X、Y 坐标分别赋值到新字段中,首先要通过 "_element_index"筛选X坐标与Y坐标的信息。筛选使用转换器 Tester。

	Transformer	Name: Test	er		
	Test Descrip	tion:			
Pas	ss Criteria				
	Pass Crit	eria: One 1	Test (OR)		
on	posite Expres	sion.			
Ces	nposite Expres	s10h.			
Con Tes	st Clauses	Operator	Right Value	Negate	Mode
Fes 1	st Clauses Left Value _eleme	Operator	Right Value	Negate	Mode Automatic
Fes	st Clauses Left Value	Operator	Right Value	Negate	Mode Automatic

图 2-7

筛选之后的数据通过转换器 AttributeCreator,将筛选成功的要素的 list1 赋值 到"x"字段,将筛选失败的要素的 list1 赋值到"y"字段

Iransformer		
	Transfor	mer Name: AttributeCreator
Advanced: Attrib	ute Value Ha	andling
Attributes To Crea	ate	
New Attribute		Attribute Value
x		🔷 _list1
-		x' Innext
+ - * * *	≍ Filte	Import



5) 将同一点的 X、Y 坐标合并:

经过上述步骤的操作,已经将同一点的 X、Y 坐标信息存储到了两个要素中, 而这两个要素的"_count"字段的值应该相同,故可以通过"_count"将要素分 组,以组为单位进行合并,合并使用转换器 Dissolver。

🙊 Dissolver Parameters	X
Transformer	
Transformer Name:	Dissolver
Group By:	_count 💌 💌
Parallel Processing Level:	No Parallelism 💌 💌
Input is Ordered by Group:	No v
Parameters	
Dissolve Count Attribute:	
Connect Z Mode:	First Wins 💌 💌
Attribute Accumulation	
Accumulation Mode:	Merge Incoming Attributes
Attributes to Sum:	No items selected.
Attributes to Average:	No items selected.
Attributes to Average, Weighted by Area:	No items selected.
▷ 🥅 Generate List	
Advanced	
Assume Input is Clean:	No •
Help Defaults V	OK Cancel

图 2-9

6) 通过坐标转点和去除重复点:

坐标转点使用转换器 VertexCreator,因不需要保留面状要素,则在转换模式 选择中选择以点替换原要素。

📯 VertexCreator Para	ameters X
Transformer	
Transformer Name:	VertexCreator
Parameters	
Mode:	Replace with Point 🔹
X Value:	🔷 x 💌
Y Value:	Ф у
Z Value:	
Index:	▼
Help Defau	lts V OK Cancel



本次实验的基础数据是两个四边形和一个三角形,显而易见总共应该有 11 个点,但是经过上述操作生成了 14 个点,是因为在统计面要素的全部点坐标时, 有一个点被算法当做起点和终点被统计了两次,故需要进行点的去重。去重使用 转换器 Matcher,输出数据的出口选择 SingleMatched 与 NotMatched。

在完成上述操作后,可以对要素的属性表进行清理,使用转换器 AttributeKeeper。

🙊 AttributeKeeper Para	ameters		23
Transformer			
Transformer Name:	AttributeKeeper		
Parameters			
Attributes to Keep:	x, y, _count, 图号		
Lists to Keep:	No Attributes Available		
Help Default	5 🔻	ОК	Cancel

图 2-11

2.2.2 侧面线生成

1) 平移点要素:

平移点要素用转换器 Offsetter。

♀ Offsetter 参数
转换器
转换器名: Offsetter
参数
模式: 笛卡尔坐标 ▼ ▼
▲ 笛卡尔坐标参数
X偏移: -1
Ÿ偏移: 5 ▼
Z偏移: 0
▷ 极坐标参数
▷ 球面坐标参数
▷点云
帮助(且) 默认(卫) ▼ 确定 取消

图 2-12

2) 连接点要素:

连接原点要素与平移后点要素即可构成侧面线,使用转换器 LineBuilder。选择"_count"字段作为点要素之间的识别字段。

	×
转换器	
转换器名: LineBuilder	
连接打断的属性: _count	
聚合模式: 从一个要素来使用属性	▼ ▼
▷ 生成列表	
帮助(∐) 默认(型) ▼	确定取消

图 2-13

图 2-14 是生成侧面线要素。



图 2-14

2.2.3 底面线与顶面线生成

1) 底面线生成:

首先需要从面要素提取线要素,但是提取到线要素是闭合的连续线要素,而 闭合的连续线要素并不利于后面构面的使用,故需要将提取的线要素打断为不连 续的线要素。

提取线要素使用转换器 GeometryCoercer。

♀ GeometryCoercer 参数		
转换器		
转换器名: GeometryCoercer		
几何对象类型输出		
几何对象类型: fme_line ▼ ▼		
▷ 几何对象部分选项		
帮助(<u>H</u>) 默认(<u>D</u>) ▼ 确定 取消		

图 2-15

打断上述的线要素的思路是在闭合的连续线要素形成的面内找到与外部线相交的线,利用线相交的关系打断外部线要素,而面的对角线正好满足这个条件。 在本实验中利用面的直骨架线代替对角线的功能。提取面的直骨架线使用转换器 CenterLineReplacer。

Ŕ	CenterLin	eReplacer 参数
	转换器	
	转换器名:	CenterLineReplacer
	参数	
	模式:	直骨架
	帮助(<u>H</u>)	〕 默认 (⊉) ▼

图 2-16

为了区分面的外部线与直骨架线,先给外部线赋予特有的属性,对其编号是 实现这一目的最简捷的办法。

😤 Counter 参数	No.	X
~ 转换器		
转换器名:	Counter_2	
参数		
计数输出属性:	_countline	
计数器名称:	counter	
计数起始于:	1	
计数范围:	全局	•
帮助(出)	默认(型) ▼ 确定	取消

图 2-17

在完成上述步骤后,可使用转换器 Intersector 打断相交线,得到不连续的外部线与直骨架线。再使用 Tester 筛选出外部线。

😤 测试参数			X	
转换器 转换器名: Tester_ 测试说明:	_2			
 通过条件 通过条件: 一个测试(或) ▼ 复合表达式: 测试条件 				
左边值 1 	运算符 =	右边值	否 模式	
+ - ▲ ▼ ≖ ェ 重复 帮助(<u>u</u>) 确定 取消				

图 2-18

2) 顶面线生成:

顶面线的生成通过底面线的平移即可实现。

♀ Offsetter 参数			X
● 转换器			
转换器名: Offs	setter_2		
●			
模式: 笛卡	尔坐标		•
▲ 笛卡尔坐标参	≥娄灯		
X偏移: -1			•
⊻偏移: 5			
Z偏移: 0			
▷ 极坐标参数			
▷ 球面坐标参数	ý ———		
▷ 点云			
帮助(出)	默认(型) ▼	确定	取消

图 2-19

图 2-20,是顶面线、底面线、侧面线的展示。



图 2-20

2.2.4 2.5D 面要素生成

由于顶面线、侧面线与底面线直接构面的效果不理想,故将三组线要素中所 有的相交线打断。使用转换器 Intersector_2。

😤 Intersector 参数	X
转换器	
转换器名:	Intersector_2
分组:	没有条目被选择。
并行处理:	无并行处理 🔹 💌
按顺序输入:	否 • •
参数	
在每个高程上复制节点:	否 ▼
压盖计数属性:	_overlaps
分段计数属性:	_segments
分离共线分段:	否
分离自交叉要素:	否 • •
聚合处理:	打散聚合 ▼ ▼
把量测当作:	[连续 ▼ ▼
属性聚合	
聚合模式:	从一个要素来使用属性 ▼ ▼
前缀:	
▶ 生成列表	
帮助(出) 默认(D)	 → 确定 取消

对打断的线要素进行构面,即可得到 2.5D 面要素。使用转换器 AreaBuilder。

😤 AreaBuilder 参数		X		
──转换器──				
转换器名:	AreaBuilder			
分组:	没有条目被选择。			
并行处理:	无并行处理	▼ ▼		
按顺序输入:	否	•		
⊿ 多边形参数				
考虑节点高程:	否	•		
连接 Z 模式:	第一个胜出	•		
保留线作为路径部分:	否	▼ ▼		
构建内部边(高级):	否	▼ ▼		
检查曲线方向:	否	•		
⊿☑ 创建环				
丢弃洞	: 否	▼ ▼		
⊿ 属性聚合				
聚合模式: 从一个要素来使用属性 ▼ ▼				
▶ □ 生成列表				
▷ 捕捉预处理参数				
帮助(出) 默认(D		确定 取消		

图 2-22

图 2-23 是 2.5D 面的展示:



图 2-23

3 总结

本实验实现了二维视图下的面要素拔高。在实现面要素拔高的过程中实现了 面折点坐标提取、面折点点要素提取、要素平移、点分组并连线、面外部线提取、 面骨架线提取、相交线打断、闭合线构面等功能。二维视图下的面要素拔高主要 用于电子地图中建筑物的简易拔高,这种假三维的建筑物拔高虽然不及真实三维 视图下的建筑物拔高(真实三维视图下的建筑物一般具有三维分析功能),但有 助于缩小电子地图的数据量以及无需三维场景。

本实验不足在于没有单独输出侧面、顶面及底面。