# 房间复制优化 + 螺旋梯段离散

## 房间复制优化

## 螺旋梯段离散

螺旋梯段从视觉上是由底部的螺旋面加上若干阶踏步构成的，由于其形状复杂，内部含有螺旋曲面，因此猜测离散慢与其有关。

由于螺旋梯段的每一阶踏步的形状特性一致，且侧边的螺旋线高度变化均匀，圆心和半径不变，根据此规律，重写螺旋梯段的离散算法，将螺旋梯段的离散特殊化，可以达到提升离散效率的目的。

### 螺旋梯段属性与体的关系

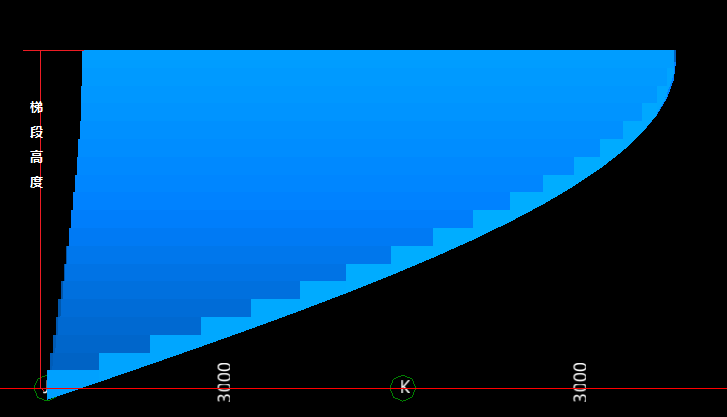
#### 属性描述

一个螺旋梯段图元中，与体形状相关的属性有以下几个：

|  |  |
| --- | --- |
| 属性 | 合法范围 |
| 踏步总高 | [50, 50000] |
| 梯段宽度 | [100, 50000] |
| 踏步高度 | [50, 500] |
| 梯板厚度 | [10, 10000] |
| 内半径 | [0, 50000] |
| 旋转角度 | [10, 180] |
| 旋转方向 | 1. 逆时针 2. 顺时针 |

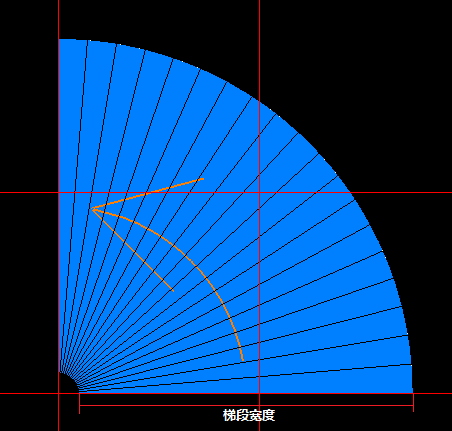
1. 踏步总高

由梯段底标高到梯段顶的高度：



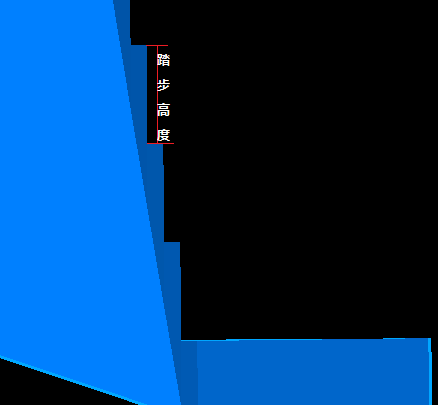
1. 梯段宽度

梯段台阶的宽度(内外同心圆的半径差):

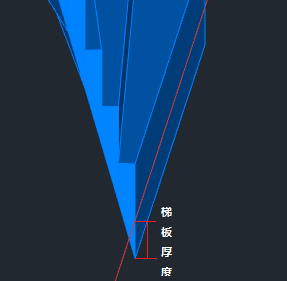


1. 踏步高度

一阶踏步的高度：



1. 梯板厚度



剩下几个参数就不描述了，比较直观。

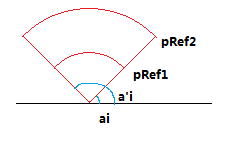
#### 单一台阶显示体的近似表示方法

每一阶台阶可以拆分为6个面：外侧面、内侧面、前端面、顶面、底面、底端面、后端面：

|  |  |
| --- | --- |
| 面 | 图形描述 |
| 外侧面 |  |
| 内侧面 |  |
| 前端面 |  |
| 顶面 |  |
| 底面 |  |
| 底端面 |  |
| 后端面 |  |

假设螺旋梯段的单个阶梯的所有边均可以近似用直线表示(离散精度足够大或者)：

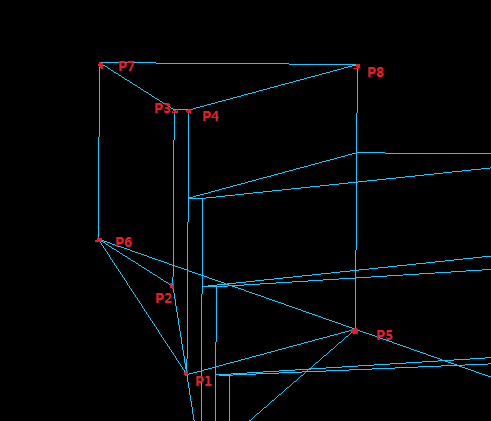
设螺旋梯段底标高为，踏步总高为，踏步高度为，梯板厚度为，踏步宽度为 ，第n阶踏步的弧度范围为[]，双侧的螺旋线在俯视状态下的弧度范围为[]，圆心为 内半径为, 外径为。首先，描述出两个参考点pRef1和pRef2：



*0 + CVector2d()*

*0 + CVector2d()*

那么对于一般台阶，可以使用8个点近似描述：



*CVector3d(, )*

*CVector3d(, )*

#### 带离散精度的单一台阶显示体近似表示方法

在实际工程中，单个的台阶踏步的弧度跨度可能比较大，此时如果依然使用直线来表示单个踏步的所有边，那么离散出来的效果将会非常差。

为了兼顾离散效果，需要我们需要根据离散参数来估算螺旋梯段螺旋边上的离散点的个数。

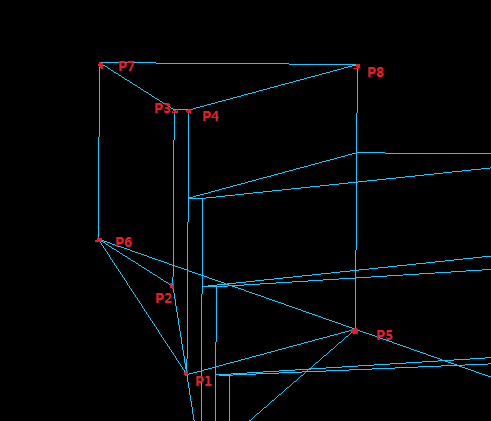
假设在当前显示品质下，螺旋体锻离散的angle为，tolerance为tol，根据李钍的分享，假设一段圆弧的圆心角为A，参数域长度为L，该圆弧的离散递增条件和离散份数*n*为：

根据上述条件可知，在停止离散的时候, 弧段的离散份数:

根据上述条件，不难推断出，之所以通用离散方法离散比较慢，是因为对于相同圆心角的圆弧线和螺旋线，螺旋线的参数阈明显更长(Z分量影响)，因此在同一tolerance下离散出来的点更多，计算量更大。

#### 离散点组成三角面片

根据2.1.3，如果一个踏步的弧边被离散为若干个线段，那么需要一种合理的方法将这些线段的端点(离散点)构造成三角面片，有以下四个面需要重组三角面片：



(S代表曲面，P代表平面)

可以构造若干个poly(比如面上，如果两条弧边各离散出两段，那么需要构造两次poly，因为这两个poly是不共面的)，然后调用图形平台的离散算法，后者的实现难度低但是要多次调用平台离散算法，在此情况下，三角面片数量可能比较多，需要验证效率。

由于已知各边上的离散点个数，因此可以构造一辅助点，用它与其它离散点来构造三角面片。的构造原则类似，这里挑来说明方法，因为这个面上两条弧边上的离散点个数可能不一样，其它面的边上的离散点个数是一致的。

假设在P3P4边上的离散点个数为N1，在P7P8边上的离散点个数为N2，其中

在顶面上取中轴线在角度上的三维点：

如果给这()个离散点排序，那么面上可以拆分为) +2 个三角面片，其中第i个三角面片可以表示为：

### 离散算法

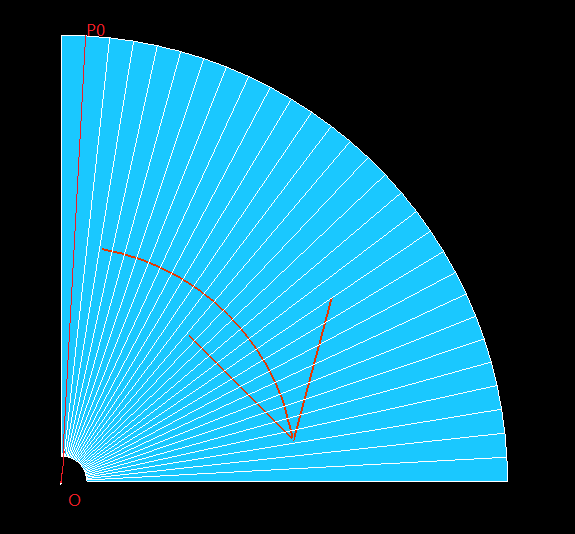
#### 中轴线的生成

#### 计算每一阶楼梯的离散份数

#### 离散点三维坐标计算

#### 构造三角面片三角面片法向及三维边框

#### 二维阶梯线的构造



原先的生成流程：

1. 根据台阶数计算出P0点
2. 构造直线 CLine2d oLine(O, P0);
3. 用oLine分别与内弧线/外弧线求交
4. 将两个交点加入borderlist

这些curve curve求交不必要，这些点可以直接计算出来。