

二维轮廓线提取

一、目的与应用

二维轮廓线提取的目的是当给定在二维平面内一组线段时，用户选择一个点，然后提取包含这个点的最小封闭多边形。可以应用于建筑软件上，当给定一组墙时，根据选择的点来寻找房间。

二、基本思想与实现

数据结构：

设计了两个类 CSegment 和 CNode

```
class CSegment
{
public:
    //指向左右接点
    CNode* m_LeftNode;
    CNode* m_RightNode;
    //记录所有线段的指针，指向下一个线段
    CSegment* m_pAllNext;
    //在接点中的以极角排序的链中的指针，指向下一个线段
    //L 为左接点的链表，R 为右接点的链表，都是双向循环的
    CSegment* m_pSegmentLNext;
    CSegment* m_pSegmentLprev;
    CSegment* m_pSegmentRNext;
    CSegment* m_pSegmentRprev;
    //在左接点的极角
    double m_LAngle;
    //在右接点的极角
    double m_RAngle;
    //查找状态
    int m_IsChecked;
    //查找时贡献的角度
```

```

    double m_ContibuteAngle;
    //所求轮廓线链中的指针
    CSegment* m_pNeedNext;
public:
    CSegment();
    virtual ~CSegment();
    //计算极角
    double GetAngle(CNode *Node);
    //为线段设置左右接点
    BOOL SetNode(CNode *Node1,CNode *Node2);
    //判断是否端点
    BOOL IsNode(DPOINT CrossPoint);
    //由点得到 Node
    CNode* GetNodeByPoint(DPOINT Point);
    //得到左右接点
    CNode* GetLeftNode();
    CNode* GetRightNode();
    //得到线段的与 A 不同的另一个接点
    CNode* GetOtherNode(CNode* A);
};
//为了计算准确，点的坐标使用了 double 型
typedef struct tagpoint
{
    double x;
    double y;
}DPOINT;

```

```

class CNode
{
public:
//记录点的位置
DPOINT m_Location;
//记录以该点为端点的按照极角排序的线段双向循环链表
CSegment* m_SegList;
//在所有接点中的下一个接点
CNode* m_pAllNext;
//新加入线段中的所有接点中的下一个接点
CNode* m_pNodeNext;
//所求轮廓线的所有接点链
CNode* m_pNodeNeed;
public:
CNode();
virtual ~CNode();
//向本接点加入线段主要是设置 m_SegList 链
BOOL AddSegment(CSegment* pSegment);
//设置 m_Location
BOOL SetLocation(DPOINT point);
//得到在查找过程中的下一条线段
CSegment* GetOtherSegment(CSegment* pSegment);
};

```

整个过程主要分为两个过程：

建立结构过程（划线过程）和查找过程。

建立结构：

加入一条线段时，依次与所有已经存在的线段求交，若有交点，将已存在线段一分为二，并修改接点的 `m_SegList`。将交点记下，并在求交的过程中，对这些交点进行插入排序（此处的复杂度最高，为 $O(n^2)$ ，但此过程为交互输入过程，速度影响较小）。求交完成后，建立新线段的所有线段，并调整好所有的 `m_SegList`。

寻找过程：

由用户所输入的点 `P`，向下做射线，得到一条线段，使此线段与该射线的交点离所输入的点最近，并且此线段的 `m_IsChecked` 为 `FALSE`。记下这个线段的左接点，将来用来结束循环。在查找过程中进行逆时针查找。每向前寻找一个线段，以 `P` 为极点的极角就会转过一个角度，累计这个角度，以做最后进行判断之用。在循环过程中有两个栈，一个记录线段，一个记录接点。开始时线段栈为空，接点栈中有一个元素（初始线段的左端点）。循环过程是这样的：刚开始有一条线段和它的左端点。然后得到此线段的另一个端点为新接点，累计角度，并把角度记入此线段（以便回溯），判断接点栈中是否已经有新接点，若有则弹出栈中以新接点以前的所有接点，线段栈也弹出同样个数的线段，并把所弹出的线段的角度的都从累计角度中减去，若无则无操作。判断之后，在此接点中，找到此线段的以极角排序的上一条线段为新线段，端点也更新为新线段的另一个端点。一直循环，直到新接点为初始接点为止。这时还要计算一次角度，并累计。此时根据累计角度可以判断是否存在封闭轮廓线。若为 2π ，则存在，结束程序；若为 0 ，则不存在。因为此时端点又回到了原来的位置，不会有其他的角度存在。若此时为 0 ，把初始线段的 `m_IsChecked` 置为 `TRUE`，从头开始继续如此循环，直到向下的射线与所有 `m_IsChecked` 为 `FALSE` 的线段都没有交点为止。若有轮廓线，用红线重绘。若无轮廓线，提示。