BSP树

首先解释**二元平面分割**（**BSP**，Binary Space Partitioning）。

如果把平面视为二维平面的话，那么直线就是一维平面，而立体图形就是三维平面了，通过这样的抽象，我们可以把任意维度的问题简化为平面问题（即**超平面**）。

对这样的平面的分割，就是BSP。BSP有很多应用场景，例如在3D计算机图形学中，判断场景中物体面的前后关系。而具体如何分割，则用**BSP树**表示。

设场景中的面都是多边形，组成了一个列表，则BSP树的构建过程如下（翻译自维基百科）：

1. 在列表中选择一个多边形P。
2. 创建根节点N，将P放入N中。
3. 依次判断列表中的其他多边形P'：
   1. P'在P的前面。将P'放入列表左中，将来用于构建左子树。
   2. P'在P的后边。将P'放入列表右中，将来用于构建右子树。
   3. P'和P相交。沿交线将P'分为两个多边形，分别放入两个列表中。
   4. P'和P重叠。将P'放入节点N中
4. 对于左右列表，重复以上的步骤，直到整棵树构建完毕，确定所有多边形的先后关系。

以下是图例，实现和文字描述相比有所优化。以直线对平面的划分为例（在超平面概念中，维度并不重要）。

|  |  |  |
| --- | --- | --- |
|  | 二维场景中有若干直线（或者三维中的多边形），组成了一个列表。在图例中，圆角矩形包含了未放入节点中的直线，而圆形中包含了已经放入相应节点的直线，箭头表示对应直线的前后方向。 | [Example of BSP tree construction - step 1.svg](http://en.wikipedia.org/wiki/File:Example_of_BSP_tree_construction_-_step_1.svg) |
| **i.** | 模拟前述算法：   1. 在列表中选择直线A 2. 创建根节点，并添加A. 3. 划分剩下的直线，分别放入A前边（B2, C2, D2）和A后边对应的列表（B1, C1, D1）。 4. 先处理A前边的直线（参见步骤ii-v）。 5. 再处理A后边的直线（参见步骤vi–vii）。 | [Example of BSP tree construction - step 2.svg](http://en.wikipedia.org/wiki/File:Example_of_BSP_tree_construction_-_step_2.svg) |
| **ii.** | 处理A前的直线列表（包含B2, C2, D2）。 选择直线B2，把它加入当前子树根节点，然后分割剩下的直线，分为了直线前（D2）和直线后（C2, D3） | [Example of BSP tree construction - step 3.svg](http://en.wikipedia.org/wiki/File:Example_of_BSP_tree_construction_-_step_3.svg) |
| **iii.** | 对于直线前的列表，选择直线D2，添加到子树根节点后，没有剩余直线，直接结束。 | [Example of BSP tree construction - step 4.svg](http://en.wikipedia.org/wiki/File:Example_of_BSP_tree_construction_-_step_4.svg) |
| **iv.** | We are done with the lines in front of B2, so consider the lines behind B2 (C2 and D3). Choose one of these (C2), add it to a node, and put the other line in the list (D3) into the list of lines in front of C2. | [Example of BSP tree construction - step 5.svg](http://en.wikipedia.org/wiki/File:Example_of_BSP_tree_construction_-_step_5.svg) |
| **v.** | Now look at the list of lines in front of C2. There is only one line (D3), so add this to a node and continue. | [Example of BSP tree construction - step 6.svg](http://en.wikipedia.org/wiki/File:Example_of_BSP_tree_construction_-_step_6.svg) |
| **vi.** | We have now added all of the lines in front of A to the BSP tree, so we now start on the list of lines behind A. Choosing a line (B1) from this list, we add B1 to a node and split the remainder of the list into lines in front of B1 (i.e. D1), and lines behind B1 (i.e. C1). | [Example of BSP tree construction - step 7.svg](http://en.wikipedia.org/wiki/File:Example_of_BSP_tree_construction_-_step_7.svg) |
| **vii.** | 处理B1前的直线，只有一条直线D1，直接插入节点即可。 | [Example of BSP tree construction - step 8.svg](http://en.wikipedia.org/wiki/File:Example_of_BSP_tree_construction_-_step_8.svg) |
| **viii.** | 选择并处理唯一的直线C1。至此，BSP树构建完毕。 | [Example of BSP tree construction - step 9.svg](http://en.wikipedia.org/wiki/File:Example_of_BSP_tree_construction_-_step_9.svg) |

参考文献：[Binary space partitioning - Wikipedia, the free encyclopedia](http://en.wikipedia.org/wiki/Binary_space_partitioning)