

# 建筑 CAD 及其实时漫游技术\*

童格亮 秦开怀

(清华大学计算机科学与技术系 北京 100084)

**摘要** 一个完整的建筑造型通常都包含 10 几万乃至上百万个多边形,若要在这样的环境中作实时漫游,则必须进行可见性预处理.通常当视点位于建筑物的内部时,只有少数面是可以看见的,而绝大多数面都被遮挡住了.如果能够只对这一小部分面进行消隐或渲染处理,就可以大大提高计算速度.本文根据建筑 CAD 的特点探索了一种基于“房间”结构的可见性预处理方法,这种方法速度较快,而且容易实现.在实时漫游中,只需根据预处理结果从视点所在房间的数据结构中提取相关的可见面进行计算,而不需处理整个建筑模型,从而大大提高了实时计算的速度.实践证明,采用可见性预处理技术,在微机上实现快速动态消隐,在速度上可以满足实时漫游的需要.

**关键词** 建筑 CAD,可见性预处理,模型空间划分,可见面超集.

近 2 年来,用计算机对建筑模型进行虚拟现实的实时漫游一直是一个令人感兴趣的话题,它用一种简单的方式使建筑师和客户能够直观地了解一个建筑设计方案.复杂的建筑模型通常都包含 10 几个以上的面,即便是高性能的工作站也无法在至多 1/10s 内对这么多的面进行消隐和渲染处理,但这对于实时漫游来说是必须的.

可以想象的是,在这样的场景中一般只能看到建筑物的一小部分,即视点所在的房间及相邻几个房间的物体,如墙、门窗和家具等等,建筑物中其它大多数面是看不到的.通过预处理,可见面的数量通常可以降低 2 个数量级.另外,在视点的移动过程中,帧与帧之间也有较大的相关性,例如只有当观察者从一个房间走进另一个房间时,场景才会有较大的变化.

在进行建筑实时漫游的探索中,有一些学者曾对可见性预处理问题进行过研究.<sup>[1,2]</sup>其中一部分把精力集中在可见性的精确计算问题上<sup>[3,4]</sup>,也就是对每个区域中的视点所能看到的的面进行精确的计算.这样完整的计算方法一般都是非常复杂而难于实现的. Sequin 等人提出了一个可见性算法<sup>[5]</sup>,用于计算可见面的超集(Superset),其中包含了所有可见面,但它只局限于轴向的建筑模型,而且仍然比较复杂.本文阐述了一种简单而又易于实现的可见性预处理方法,它是基于令人较易理解的“房间”结构基础上的.

本文第 1 节将对建筑造型中同可见性预处理有关的特点进行描述,第 2 节具体阐述可

\* 作者童格亮,1972年生,硕士生,主要研究领域为计算机辅助设计,计算机图形学.秦开怀,1958年生,博士,副教授,主要研究领域为计算机图形学,可视化,几何造型,计算机辅助建筑设计,小波分析,CAD/CAM,VR.

本文通讯联系人:童格亮,北京 100084,清华大学计算机科学与技术系

本文 1996-03-04 收到修改稿

见性预处理的方法,而实时漫游的一些技术问题将在第3节说明.

### 1 建筑造型及其特点

在目前的大多数建筑 CAD 系统中,建筑模型的各种构件如墙壁、门窗、阳台、楼梯等都是由一些简单的平面来表示的,这样的表示方式在用于实时漫游时是非常不方便的,因为这种模型中的面与面之间是孤立的,并没有体现出各种构件之间的连接关系,不利于空间的划分.事实上,各构件元素之间的联系非常紧密,例如门、窗等构件是依附于墙的,门是连接相邻 2 个房间的通路等等.下面我们描述一种建筑模型的表达方式,它可以把建筑物中各元素紧密地联系在一起而组成一个整体.清华大学 CAD 中心开发的建筑辅助设计软件就使用了这种方法.

在建筑设计系统中,主要的工作在于建筑施工图的设计.每张建筑平面施工图对应 1 个建筑标准层,即平面布局相同的楼层模型,它可以为许多楼层所共享.各种建筑构件都可以布置在建筑标准层的平面图中.

为了在数据结构中充分体现构件之间的二维平面拓扑关系,我们在造型过程中对各种构件建立了图 1 所示的聚合关系.

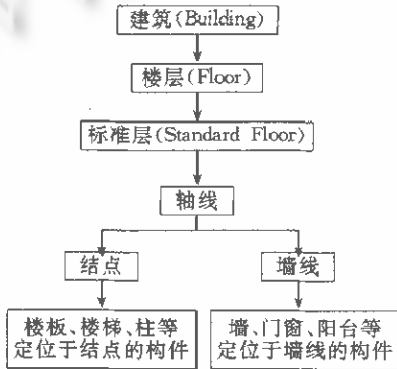


图1 建筑模型中各种构件的聚合关系

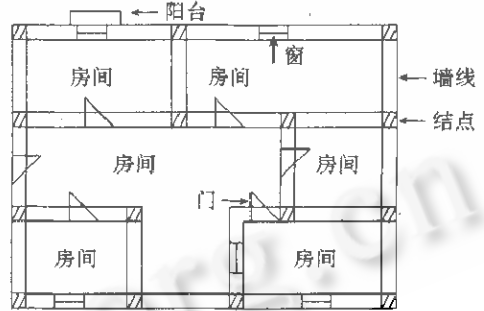


图2 建筑标准层平面布局示意图

建筑标准层中(见图 2),轴线是定位基准,轴线与轴线的交点就是结点,在结点上可以布置柱、楼梯等构件;轴线上被结点分开的线段称为墙线,墙线上可以布置墙、门窗、阳台等构件.在数据结构中,彼此关联的构件用指针来建立联系.此外,为了进行室内设计以及实时漫游中的可见性预处理,还特别设立了房间(Room)的结构,它是一个四周由墙线封闭起来的区域,在逻辑上也反映了一般意义上的房间概念.为了在实时计算时能快速地得到可见面数据,房间的数据结构中应当储存该房间的墙线指针以及房间中的孤立结点指针.而某些如门、窗这样的构件连接着相邻的 2 个房间,观察者可以通过这些元素看到其它房间的场景,因此 2 个相邻房间的指针也应当在这些构件的数据中记录下来.

### 2 可见性预处理

#### 2.1 空间划分

在可见性预处理过程中,首先要进行模型空间划分,也就是把建筑标准层的平面图划分为若干个房间.根据上 1 节所叙述的模型,我们把空间划分为 2 个步骤:建立连接图和搜索

房间.

### 2.1.1 建立连接图

房间的划分主要沿着墙线进行,因此首先由模型数据建立连接图  $G$ ,其节点  $G\_NODE$  对应模型中的结点,其边  $G\_EDGE$  对应模型中的墙线.图2对应的连接图见图3.在连接图中每个  $G\_EDGE$  连接2个  $G\_NODE$ ,而每个  $G\_NODE$  可连接多个  $G\_EDGE$ .可以看出,从造型数据生成连接图是很容易的,只需作一些简单的转换.需要注意的是,生成的连接图  $G$  可能包含了几个连通子图.

### 2.1.2 搜索房间

房间的划分主要是一个深度优先的搜索过程.当搜索出一个简单回路时,该回路上的边  $G\_EDGE$  就组成了一个房间的墙线.搜索策略如下:

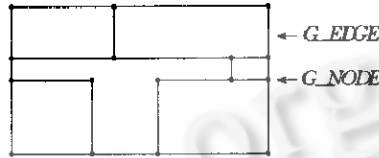


图3 连接图

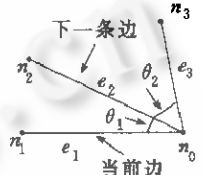


图4 搜索策略

假设当前边为  $e_1$ ,搜索方向由节点  $n_1$  到  $n_0$ ,在下一步搜索中,总是选择与顺时针夹角最小的那条边作为下一条边.例如在图4中,即选  $e_2$  作为下一条边.如果同节点  $n_0$  相连的边只有  $e_1$ ,则下一条边仍然选  $e_1$ ,沿原路返回.由图4可以看出,在搜索过程中,所有墙线都要沿不同方向被搜索2次.

根据上述策略对连接图  $G$  进行搜索,当所有墙线都被搜索了2次之后,就得到了一系列由封闭的折线段组成的房间结构.同时,该标准层的外围墙线也能被搜索出来.由这些外围墙线构成的房间是一个特殊的房间,它表示的区域是建筑物外部的空间,我们将其记为外围房间.

如果连接图  $G$  由若干个连通子图组成,则每个连通子图都有一个外围房间,某些外围房间是包含在其它房间内部的(如图5),这时就应当把这2个房间合并为1个.

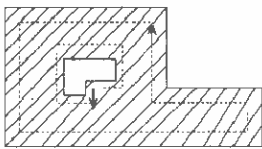


图5 房间的合并

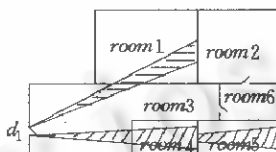


图6 房间对门窗的可见性

当平面图正确划分为房间结构之后,就可以把每个房间同所有位于其内部的构件元素用指针联系起来,以便在实时漫游中能方便地提取数据.而象门窗这样的构件,应当把相邻的2个房间

指针记录下来.

## 2.2 房间的可见性预处理

在实时漫游中,关键的问题在于如何快速地得到视点可能看到的面的数据.从上面介绍的内容可知,只要知道了视点能看到的房间,就可以很快地得到可见面的超集.因此在预处理中要计算的是在任何一个房间内能看到周围的哪些房间,这实际上就是该房间内在所有门窗处能看到的房间的并集.所以问题就转化为计算任一门窗处所能看到的房间.

下述递归过程可以计算出任何一个门窗看到的房间,并将其保存在该构件的可见房间队列中.假设第1次调用这个过程时Room是Element相邻的房间,并且Element的可见房间队列为空.

### 算法1.

Room Visibility(门或窗 Element, 房间 Room)

```

{
  if (Room 不在 Element 的可见房间队列中)
  {
    把 Room 加入到 Element 的可见房间队列中;
    for (Room 中的每一个门窗构件 TmpElement)
      if (Element 与 TmpElement 之间的连线不与任何墙线相交, 或者相交的交点处均为门窗类构件)
      {
        Room Visibility(Element, TmpElement 的相邻房间 1);
        Room Visibility(Element, TmpElement 的相邻房间 2);
      }
  }
}

```

如图 6 所示, 在门  $d_1$  处可以看到的房间有  $room1, room3, room4$  和  $room5$ , 而看不到  $room2$  和  $room6$ .

### 3 实时漫游

在漫游时, 视点的位置在不断地变化. 实时处理中首先要确定视点在哪个房间内, 把这个房间所有的门窗结构中保存的可见房间队列合并起来, 就是在当前房间的任何位置所能看到的房间全集. 只要视点不离开这个房间, 该集合就不会改变.

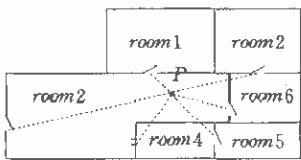


图7 视点P看不到room4和room5

实时处理过程中, 可以把这个房间全集中每个房间的面全部提取出来进行处理, 显然该面集合就是可见面的超集. 这种方法比较简洁, 而且视点在一个房间内移动时该集合不变, 只需提取 1 次. 但是在某些情况下, 可见房间全集中的一些房间是从视点完全看不到的. 例如图 7 中视点  $P$  位于房间  $room3$  中, 其可见房间全集为  $room1 \sim 6$ , 而  $room2, room4$  和  $room5$  是  $P$  点看不见的, 不应参与计算. 有时这种情况会非常严重, 以致影响到实时漫游的速度.

为了进一步减少实时计算的数据量, 我们用下面的算法对可见房间全集进行筛选, 得到一个可见面的超集.

#### 算法 2.

```

可见面超集 = NULL;
for (可见房间全集中的每一个房间 R)
  for (R 中的每一个门窗 E)
    if (视点 P 和 E 的连线不与任何墙线相交, 或者交点处均为门窗类构件)
      {将 R 中的所有面提取出来加入可见面超集}

```

这个算法保证了只有当从视点能够看到一个房间的门窗时, 才把这个房间的数据参与计算. 在进行  $P$  与  $E$  连线和墙线求交时, 我们只要求出  $PE$  与可见房间全集中的墙线的交点即可, 而不必把  $PE$  与所有的墙线都求交.

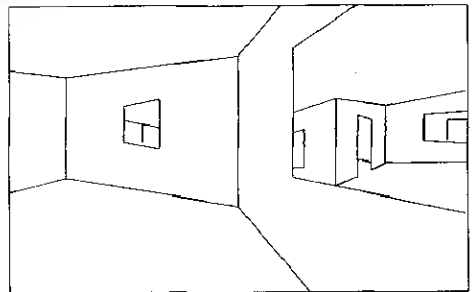


图8 实时线框消隐场景实例

在上述算法的基础上, 我们在微机上实现了线框消隐的交互式实时漫游系统. 图 8 是漫游中的一个场景实例. 从表 1 可以看出进行预处理后需要计算的面的数量降低了很多.

表 1 预处理前后计算量的比较

标准层实例	总面数	平均看到的面数	所占百分比
标准层实例 1	1 592	37.0	2.3%
标准层实例 2	576	35.2	6.1%
标准层实例 3	228	30.2	13.2%

## 4 结 论

利用本文提出的可见性预处理算法可以有效地处理建筑模型的实时漫游和动态巡视,可对建筑物进行实时消隐处理.利用本文介绍的方法,在 586(Pentium 75MHz)微机,每秒钟可生成并显示 20~30 幅以上的消隐图形,基本上可以满足实时漫游的要求.

作为下一步的研究内容,我们将进一步研究在微机上的具有透明、阴影、纹理等真实感图形效果的实时漫游问题.

### 参 考 文 献

- 1 Sequin C H, Bukowski R W. Interactive virtual building environments. In: Shin S Y, Kunii T L eds., Computer Graphics and Applications, Proc. of the Third Pacific Conference on Computer and Applications, Pacific Graphics'95, Singapore: World Scientific, 1995. 159~179.
- 2 Airey J M, Rohlf J H, Brooks F P Jr. Towards image realism with interactive update rates in complex virtual building environments. ACM SIGGRAPH, 1990, 24(2):41~50.
- 3 Chazelle B, Guibas L J. Visibility and intersection problems in plane geometry. Proc. 1st ACM Symposium on Computational Geometry, 1985. 135~146.
- 4 Vegter G. The visibility diagram: a data structure for visibility problems and motion planning. Proc. 2nd Scandinavian Workshop on Algorithm Theory, 1990. 97~110.
- 5 Teller S J, Sequin C H. Visibility preprocessing for interactive walkthroughs. Proc. of SIGGRAPH'91, 1991. 61~69.

## VISIBILITY PREPROCESSING AND INTERACTIVE WALKTHROUGHS FOR ARCHITECTURAL CAD

Tong Geliang Qin Kaihuai

(Department of Computer Science and Technology Tsinghua University Beijing 100084)

**Abstract** Usually, a whole architectural model contains millions of polygons. Visibility preprocessing is necessary for real time walkthroughs in such an environment. When the viewpoint is in the building, only a small part of polygons can be seen by the viewer, the others are all hidden. The display will be speeded up if people just compute this small part of polygons. In this paper, an efficient algorithm of visibility preprocessing based on the "room" structure is proposed. During the real time walkthroughs, the visible polygons instead of those of the whole building are retrieved from the database and are sent to the hide or render pipeline.

**Key words** Architectural CAD, visibility preprocessing, model-space subdivision, superset of visible faces.