ISSN1000-9825 Journal of Software 软件学报 2000,11(9):1241~1247

包含整体镜面反射的虚拟场景实时漫游算法*

华炜 彭群生

(浙江大学 CAD&CG 国家重点实验室 杭州 310027)

E-mail: peng@cad.zju.edu.cn

摘要 针对复杂多边形场景,提出了一种基于层面叠加技术并能处理整体镜面反射分量的漫游算法.此算法将整体镜面反射分量预处理和整个场景的预消隐紧密结合,把整体镜面反射的计算问题转化为求取光束 PVS (potentially visible set)的问题.算法通过预处理,建立镜面 PVS 映射 G 和可能镜面入射光束集合;在漫游阶段,则通过执行 G 映射并调用其他层面的光亮度计算过程,实时地合成整体镜面反射分量.

关键词 整体镜面反射,虚拟现实,PVS(potentially visible set),光束追踪.

中图法分类号 TP391

虚拟现实技术和仿真技术的发展,对图像生成的真实感和实时性提出了很高的要求.虽然近年来图形工作站的图形处理能力已有很大提高,但随着虚拟场景的越来越复杂以及对生成图像品质的要求不断提高,要同时获得图像的真实感和生成的实时性依然是相当困难的.

为解决图形生成的实时性问题,已经出现了许多加速图形绘制的算法,其中包括构造虚拟场景的多层次细节建模[1]方法、利用景物空间和图像空间连贯性的快速消隐的层次 Z-buffer 算法[2]、改进的 Warnock 算法[3]等.

基于图像的绘制算法则是另一类实时图形生成技术[4].它利用一系列预处理的图像数据来合成观察者所见的图像,其算法效率与场景多边形个数无关.该方法的优点在于,避免了对画面上大多数像素的实时消隐计算,但它只能处理光亮度与视点无关的虚拟场景,在其他应用场合便难以达到图像真实感方面的要求.

我们提出了一种针对虚拟场景漫游的画面绘制技术,称为 Overlay 技术.该技术着重考虑两个方面.一方面是加速消隐过程,即采用预消隐技术[5~7],在实时绘制时,对任意给定视点和视线方向,快速地进行视域裁剪和剔除隐藏多边形,以达到加速的目的.另一方面是提高光亮度的计算效率.一般景物表面的光亮度由环境分量、漫射分量、镜面高光分量、整体镜面反射分量、整体规则透射分量这 5 部分组成,在计算画面上各可见点的光亮度时,先分别计算画面光亮度的各个分量图像,然后再线性叠加.这样做的好处是,对于每一分量图像,可以根据其不同的相关性,采用专门的优化算法来加速;而对于视点无关分量(环境分量、漫射分量),则可以预先计算好,并存储起来备用.

对于基于 Overlay 技术的漫射、镜面高光分量计算问题,已经有了一些有效的算法[8,9].而本文着重研究对整体镜面反射分量的实时计算,提出了一种基于 Overlay 技术,并能得到目前通用的图形硬件支持的算法.为了验证该算法的实际效果,我们开发了一个基于 Overlay 技术的虚拟场景实时漫游原型系统,该系统实现了漫射和整体镜面反射效果,并能够达到实时显示的要求.

本文第 1 节讨论了系统采用的场景预消隐技术.第 2 节简单介绍了漫射分量的计算处理方法.第 3 节着重研究了对整体镜面反射分量的处理方法.第 4 节结合 Overlay 技术介绍了系统结构和实验结果.最后进行总结并指出了今后的研究方向.

* 本文研究得到国家自然科学基金(No.69823003)资助.作者华炜,1972 年生,博士生,主要研究领域为虚拟现实,医学数据可视化.彭群生,1947 年生,博士,教授,博士生导师,主要研究领域为真实感图形,虚拟现实,科学计算可视化,红外仿真.

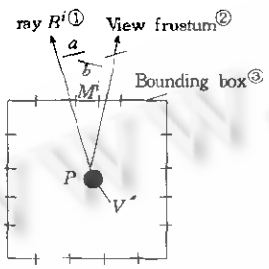
本文通讯联系人:彭群生,杭州 310027,浙江大学 CAD&CG 国家重点实验室

本文 2000-02-28 收到原稿,2000-04-17 收到修改稿

1　预消隐技术

在虚拟场景漫游过程中,观察者在每一时刻见到的只是整个场景的一部分.为了加速实时绘制时的消隐过程,我们采用预消隐来减少实际参与绘制的多边形数目.

我们将漫游过程中视点可到达的区域剖分成立体网格,称为视点网格.对每个视点网格,以其为中心构造一个包围盒,称为视域包围盒,对包围盒的6个面作二维分割,其2D示意图如图1所示.当视点在视点网格V中的任意位置P时,视点和包围盒表面上的每一方格M^i形成光束B^i,对光束B^i求取其中的可见面集合$S^V_{VS}(P,M^i)$,如图1中的a,b.这样,$S^V_{PVS}(M^i)=\bigcup_{P\in V}S^V_{VS}(P,M^i)$就是从该视点网格中的任一点$V$透过包围盒表面的方格$M$的可能的可见面集合(potentially visible set).系统不断采样V中的点P^k,然后合并相应的$S^V_{VS}(P^k,M^i)$,直到并集趋近于不变为止,其并集即为近似的$S^V_{PVS}(M^i)$.这样,对每一个视点网格即可构造一个PVS映射F,将包围盒各表面方格映射成相应光束的PVS.



①光束B^i,②视域,③包围盒.

Fig. 1　A sketch in 2D, a and b represent two patches in the scene; M^i represents one grid on the bounding box

图1　2D示意图,a,b表示场景面片;M^i表示包围盒上的一个网格

为了能快速求取光束的可见面集$S^V_{VS}(P,M^i)$,以P为视点,包围盒的每一个侧面作为投影平面,用硬件Z-buffer绘制场景.一般的硬件Z-buffer都是在帧存中直接记录离视点最近面片的光亮度值,为了取得可见面号,我们在帧存的每一像素上记录该像素上可见的面片号.将帧存图像按照包围盒表面的剖分方式分割成子图像,然后取子图像内各像素上记录的可见面号集合作为光束的PVS.具体算法可参见第3节中的算法1(在算法1中,对镜面情况进行了特别处理).

在漫游时,需要按当前视线方向快速求取位于当前视域四棱锥中的视域包围盒的表面方格,为此,我们将包围盒表面的方格表示成层次树结构.如果父方格位于视域四棱锥外,则不必再对其子方格进行判断;如果父方格完全位于视域内,则说明其子方格必在视域内;只有在父方格与当前视域四棱锥的边界面相交时,才需要对子方格作进一步的判断.

在得到位于视域器棱锥内的包围盒表面方格之后,我们就可以基于前述的PVS映射F建立该视点网格朝当前视线方向的可能的可见面集合,然后再进行消隐.

2　漫射分量的计算

对于漫射分量,最常用的光照模型是Lambert模型.漫射分量具有与视点、视线方向无关,在较平坦的表面上变化较缓慢的特点.

利用漫射分量变化缓慢的性质,可以将物体表面重新剖分成小面片,使得在小面片上,漫射分量的分布近似为双线性函数.这样,只要计算出了小面片顶点的漫射分量,面片内部各点的光亮度漫射分量即可用双线性插值来得到.双线性插值可由硬件图形加速器来实现[9].

3　整体镜面反射分量的计算

由于景物表面各点的整体镜面反射分量与视点有关,所以很难采用基于图像的绘制算法来绘制反射分量图像;此外,镜面反射分量沿表面变化非常迅速,相邻像素间几乎没有相关性.

过去处理镜面高光的方法有3类:

(1) 光线跟踪算法.它可以精确地计算整体镜面反射,对于多次镜面反射也可以处理,但计算量较大,很难实现实时绘制.

(2) 基于图像的绘制算法.对镜面表面的每一个可见点建立一个环境映照,这样,计算整体镜面反射光亮度

时只需通过环境映照找出沿视线反射方向的场景光亮度即可,避免了反射光线与场景求交的过程.但这一方法有个缺点,即在建立镜面上可见点的环境映照时,如果对面片采样密,则存储量大;采样疏,则反射分量的失真较大.事实上,对于大的镜面,要获得较真实的反射效果,所需的存储量是极大的.

(3) 基于镜面虚像的绘制算法[10].对于平面镜面,可以通过一个反射变换求出场景相对于该镜面的虚像,然后将虚像适当地进行裁剪,最后再进行绘制.该方法的优点是,于可由图形硬件支持,易于实现.但其主要问题在于,因引入虚像而使场景的复杂度增加,每增加一个镜面便使场景中待绘制多边形数增加一倍.这样,在多镜面的复杂场景下,达到实时绘制是困难的.

本文提出的方法基于镜面虚像原理,并结合了光束追踪[11]的思想.为了避免上述场景复杂度倍增的问题,我们采用了一个类似预消隐的预处理过程,并使其处理的方式能够和整个场景的预消隐过程结合起来.目的在于当漫游时避免绘制根本不可能见的虚像.

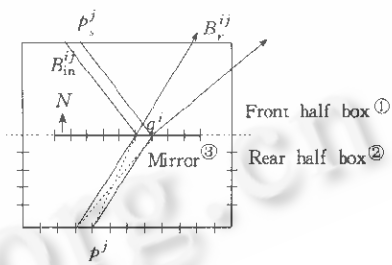
具体思路如下:首先构造一个映射 G,它将镜面的入射光束映射到其镜面反射光束可能可见景物的面片集合(PVS),并将该映射预先保存起来;在漫游时,快速根据视点位置求出视点向镜面发出的入射光束,这样通过映射 G,可得到相应的 PVS;最后对得到的 PVS 进行反射变换,用前面提出的方法绘制它们的漫射分量,这样便得到镜面的整体反射分量.

3.1 构造镜面 PVS 映射 G

就平面镜面而言,对每一个镜面构造一个包围盒,镜面恰将包围盒切分成两个相等的半包围盒,分别称为前半包围盒和后半包围盒.如图 2 所示,镜面和后半包围盒分别被剖分成若干小块,分别用 q^i 和 p^j 标示.前半包围盒上的小块 p_s^i 和 p^j 关于镜面对称.图中的 B_{in}^{ij} 是内光束,B_r^{ij} 是外光束,N 是镜面法向.设想在后半包围盒的 5 个面上的每一点处都有一个点光源,我们只考察这些点光源发出的与镜面相交的光线,可知这些光线形成一个 4D 空间 L_r.进一步地,我们将 L_r 剖分成若干子空间,即若干光束.具体做法如下:

(1) 对镜面作剖分.将整个镜面分割成若干小镜面,编号并记为 $q^i(i=0,\ldots,n)$.对后包围盒的 5 个面也分别作分割,给得到的小面片编号并记为 $p^j(j=0,m)$.

(2) 定义光束 $B_r^{ij}=\{$任意的光线 l,起点位于后半包围盒 p^j 内,并穿过镜面上区域 $q^i\}$.可以证明,$L_r=\bigcup_{0\le i\le n,0\le j\le m} B_r^{ij}$.



①前半包围盒,②后半包围盒,③镜面.

Fig. 2 The sketch of specular beam in 2D
图2 镜面光束的2D示意图

如图 2 所示,光束 B_r^{ij} 被镜面分成两部分,位于后半包围盒内的部分称为内光束,另一部分称为外光束.对于外光束 B_r^{ij},求出位于该外光束之内的可能的可见面集合,记为 S_{PVS}^{ij},具体算法参见场景预消隐中的求光束 PVS 的算法.这样就可以构造一个映射 F,满足 $S_{PVS}^{ij}=F(B_r^{ij})$.

进一步地,按照镜面对称关系,将前半包围盒也作剖分,即前半包围盒的 5 个面被细分成若干小区域 p_s^i,且每一个 p_s^i 都和后半包围盒的 p^j 是镜面对称关系.按照 B_r^{ij} 的定义方式,定义光束 $B_{in}^{ij}=\{$任意的光线 l,起点位于 p_s^i 内,交于 $q^i\}$.再定义 B_{in}^{ij} 的内光束为位于前半包围盒的部分,显然,B_{in}^{ij} 内的光束正是形成 B_r^{ij} 外光束的镜面入射光束.如果记此映射为 T,即 $B_r^{ij}=T(B_{in}^{ij})$.这样,我们就得到复合映射,即任意给出一个镜面入射光束,通过映射 G 即可得到经镜面反射后该光束内的可能的可见面集合.

G 映射具体说来是这样实现的:

(1) 映射 F 存储成一个 2 维数组,每个数组元素 $F[i,j]$ 记录光束 B_r^{ij} 所对应的 PVS 集合.

(2) 对任意一个入射光束 B_{in},转化为 B_{in}^{ij} 的形式,即计算出对应的 p_s^i 和 q^i(见 3.2 节).

(3) $F[i,j]$ 即为 $G(B_{in})$.

3.2 计算镜面入射光束

由于我们采用分别位于前半包围盒和镜面上的两区域(p_s^i 和 q^i)来定义一个光束,对于可见镜面,以当前视点为投影中心,将其前半包围盒的小块 $p_s^j(j=0,\ldots,n)$ 投影在成像平面上,取各自的编号值对各投影区域作颜

色填充,用硬件 Z-Buffer 绘制成一幅编号图像,再采用相同的取景变换和透视变换,将镜面上各小块 q^i 投影到成像平面上,计算出其边界所对应的位置并在编号图像上作出标记,则对于编号图像上位于标记 q^i 内的区域,用种子填充算法遍历其所含像素,即可得若干前半包围盒的小块编号 $p_s^j(j=k,\ldots,k+n)$,所对应的光束 B_{in}^j 则为从视点发出,交前半包围盒于区域 p_s^j,且交镜面于区域 q^i 的光束.对所有的镜面小块都进行如此操作,则可得到所有的镜面入射光束.

事实上,我们可以在整个场景预消隐过程中,同时进行镜面入射光束的计算,即预先计算视点在某一空间区域活动时,它对镜面可能发出的入射光束.由于在场景预消隐中对每一个视点网格均做了一个 PVS 映射,即将由视域包围盒表面方格确定的光束映射到它的可能可见面集.那么在处理计算每一光束的 PVS 时,可以进一步判断光束的可能可见面集中是否有镜面存在.如果有,则对该光束做特殊标记,并把它转化为镜面 PVS 映射 G 所需的光束形式,即用前半包围盒网格和镜面网格来定义,就可完成对镜面入射光束的预处理.

在进行预消隐过程中,直接对镜面剖分成的小镜面进行可见性判断,为了避免在计算镜面入射光束时再计算每个小镜面的边界,我们改造了预消隐算法以使之能同时计算镜面入射光束.其算法如下:

算法 1.

设视点网格为 V_{ijk},$SB_{ijk}[m,n]$为对应 V_{ijk}的视域包围盒表面方格数组,m 为包围盒表面的编号(0,...,5),n 为每一表面的方格编号.

```
(1) FOR(视点网格 V_ijk的每一采样点)
(2)    以视域包围盒的每一面作为投影平面,取面片号作为场景中各面片颜色,绘制场景多边形,得到图像
       Pic1
(3)    对 Pic1按视域包围盒相应表面上的剖分方式进行分割
(4)    FOR(每一分割后的子区域 sub_pic1[n])
(5)       FOR (该子区域内的每一像素 Pic1[i,j])
(6)          IF(Pic1[i,j].color<>0)
(7)             将 Id=Pic1[i,j].color(即面片号)记录到 SB_ijk[m,n]的 S_PVS中.
(8)             IF(Id 对应的是镜面)
(9)                取该镜面的前半包围盒上表面方格的编号为其颜色值
(10)               在成像平面上,绘制各方格并保存为 Pic2(若前半包围盒已绘制,则取出)
(11)               用种子填充算法遍历 sub_pic1[n]子区域中颜色为 Id 的像素 Pic1[ii,jj],并且使
(12)                  Pic1[ii,jj].color=0
(13)                  Id_enb=Pic2[ii,jj].color(即前半包围盒的表面方格编号),
(14)                    则二元组[Id,Id_enb]对应于一个镜面入射光束,
(15)                    记录在 SB_ijk[m,n]的 B_in中.
(16)                  Pic2[ii,jj].color=0
(17)            ELSE
(18)               用种子填充算法遍历 sub_pic1[n]子区域中颜色为 Id 的像素 Pic1[ii,jj],并且使
(19)                  Pic1[ii,jj].color=0
```

其中步骤(8)~(16)是用来计算镜面入射光束的.从算法上来看,这种方法和预消隐结合得很好.另一方面,正如第3.1节中指出的那样,本文所提出的计算镜面 PVS 的方法与场景预消隐中的 PVS 算法相同.在算法1中,如果将视点网格 V_{ijk}视为镜面的后半包围盒网格 p^j,$SB_{ijk}[m,n]$视为镜面网格数组 $q^i(i=0,\ldots,n)$,并且不执行(8)~(16),则执行的结果为由 p^j 和镜面网格 q^i 所形成光束的 PVS.那么只要再让 p^j 取遍所有后半包围盒网格,就能得到所有光束的 PVS.由于系统中3个主要的部分都使用了算法1,这使得整个系统的实现显得很简洁.

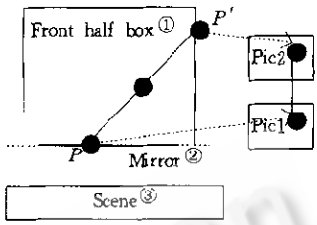
需要指出的是,在步骤(11)中,如果采样视点处于镜面前半包围盒内部,应沿原视线方向的反向来绘制前半包围盒,并将投影坐标系作适当的变换,以使任一通过视点的直线交前半包围盒和镜面所得点在 Pic1和 Pic2上的对应像素的位置是相同的.如图3所示,在绘制前半包围盒时,将绘制场景的投影坐标系作关于视点的对称变

换,这样使得 pic1和 pic2上的同位置的像素所对应的空间点关于视点对称,如图中的 p 和 p' 点.

由于算法只处理可见的镜面网格,这样对于被遮挡的镜面网格,就不必计算它的入射光束,从而避免了绘制那些不对最终画面的镜面反射分量起作用的 PVS.

3.3 漫游时计算镜面的整体反射分量

经过上述镜面入射光束预处理后,在漫游时得到普通面 PVS 的同时,可直接得到可能的镜面入射光束.



①前半包围盒,②镜面,③场景.

Fig. 3 The viewpoint is located within the front half of the bounding box

图3 视点位于前半包围盒内

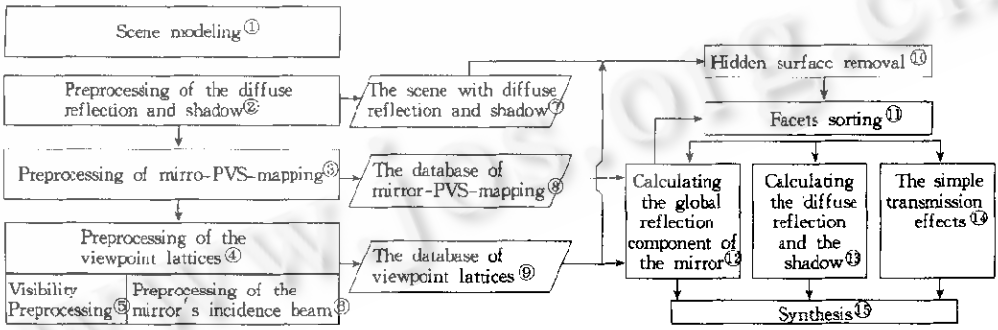
在得到入射光束之后,根据镜面 PVS 映射 G 得到所对应的可能的可见面 PVS,将所得 PVS 对镜面作反射变换,然后用镜面对 PVS 进行三维裁剪.如果镜面是三角形或四边形,则用镜面的边界和视点构成的三棱锥或四棱锥对 PVS 进行裁剪;如果镜面是 N 边形($N>4$),则采用 OpenGL 或 GL 模版测试的方法来保证 PVS 在投影平面上的投影位于镜面投影之内.再用前两节的方法来绘制这些面,这样就可以得到整体反射分量.

目前我们的原型系统只实现一次反射,但对于多次反射的情况,本文提出的方法具有可扩展性.即在视域网格中存放一个光束追踪树集合.在漫游时,以后序遍历访问结点,取出光束集合,按前述方法依次绘制即可.构造光束追踪树的方法类似于光线追踪算法.

设处理的第1个镜面为 M_1,首先用算法1计算出入射光束,然后检查这些入射光束所对应的 PVS 中是否有镜面存在,若有,则设其为镜面 M_i,然后将当前视点变换到 M_1的对称位置,再用算法1计算出对应 M_i 的入射光束,...,如此递归进行,直到镜面 PVS 中不再出现镜面为止或反射分量的贡献小于阈值时停止.

4 系统实现和测试结果

我们开发的基于 Overlay 技术的实时漫游系统包括预处理和漫游两个部分.其软件的框架如图4所示.在合成中,对场景中镜面,按照距当前视点的由前向后的顺序依次处理.先绘制出整体镜面反射分量图像,然后采用硬件的 α 混合方法,用镜面的反射率设置适当的混合方程,将镜面的漫射分量叠加上去.



①场景造型,②漫射分量及阴影预处理,③镜面PVS映射预处理,④视点网格预处理,⑤预消隐,⑥镜面入射光束预处理,⑦含漫射和阴影信息的场景,⑧镜面PVS映射数据库,⑨视点网格数据库,⑩消隐,⑪面片分类,⑫镜面整体分量计算,⑬漫射分量及阴影处理,⑭简单透射处理,⑮合成.

Fig. 4 Sketch of the structure of prototype system

图4 原型系统结构示意图

我们的原型系统是以 SGI Octane MXI 作为硬件平台,其性能指标如下:CPU 类型:R10000;主频:195MHz;内存:256M;Color&α 位面:64位;多边形:2.24M;纹理内存:4M.测试场景的多边形数为23872个,多边形均为四边形.在预处理时,视点区域的分割取10×10,视域包围盒表面的分割为10×10;镜面分割成10×10,半包围盒的5个面分别分成5×5的网格.场景中有两个镜面地板和墙上的镜子,反射系数分别为0.18和0.85.测试结果见表1.

Table 1 The test result

表1 测试结果

Number of directly visible polygons①	2 369	5 119	4 581	3 260	5 292	7 701
Number of polygons visible through the mirror②	0	439	1 328	2 888	3 015	3 038
Total number of rendered polygons③	2 369	5 558	5 909	6 148	8 307	10 739
Rendering time④(s)	0.05	0.07	0.09	0.11(Fig. 5)	0.13(Fig. 6)	0.15

①直接可见的场景面片数,②镜面可见场景面片数,③实际绘制的多边形数,④绘制时间.

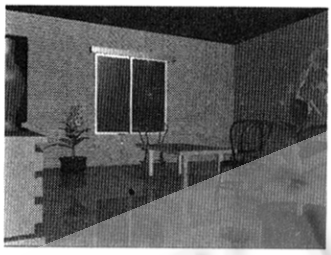


Fig. 5 The floor in the scene produces mirror reflection and its reflection coefficient is 0.25

图5 此场景中地板为镜面.镜面反射系数为0.25

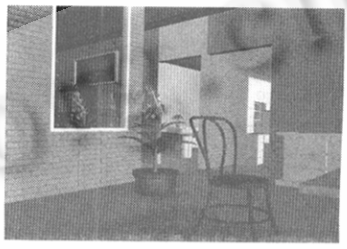


Fig. 6 In the scene, the reflection coefficients of the floor and the mirror on the wall are 0.19 and 0.8 respectively.

图6 此场景中地板、墙上的镜子为镜面.地板的镜面反射系数为0.19,墙上镜子的反射系数为0.85

5 结 论

Overlay 技术是一种真实感图像的实时生成技术.它实质上是一种方法论,将光亮度各分量的合成由原先的逐像素合成变成先计算各光亮度分量的图像,然后整块图像进行合成.其优点是,可以有效地利用各光亮度分量图像的相关性,有针对性地进行优化,最终达到整体优化的效果.注意到对于整体镜面反射分量而言,如何确定在当前视点处镜面上可见的虚像面片的集合是本质的问题,而如何最优地计算这些面的光亮度问题则是其他相应光亮度计算算法的任务.本文的思路正是基于此点,以寻找可见虚像面片最少的集合来对整体镜面反射分量的计算进行优化.另一方面,对系统而言,实用性是重要的.所以,我们不仅考虑理论上的优化,还考虑对目前通用的图形加速卡的优化.从系统原型运行的效果来看,它已经能够实时地绘制出具有漫射、阴影、整体镜面反射的图像,而且存储量不大.

将来的工作:

(1) 系统目前只能对平面镜面作处理.而对于曲面镜面,尚只能使用环境映照的方法来实现.Eyal Ofek[12]提出了一种在交互情况下的近似计算曲面镜面整体反射分量的算法,但在速度上还达不到实时.

(2) 我们将对求取光束 PVS 这一核心算法进行进一步的优化,以加速整个预处理过程的速度.

(3) 考虑到实用性,将一次性预处理过程变成可迭代的过程是十分有益的.特别是当用户需要不断修改场景时,每次预处理可以都可以从上次的处理结果中提取出依然有效的信息,作为本次处理过程初始状态,以加速整个预处理的速度.

参考文献

1 Founkhouser T A, Sequin C H. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In: Kajiya J T ed. Proceedings of the SIGGRAPH'93. New York: ACM Press, 1993. 247～254

2 Greene N, Kass M, Miller G. Hierachical Z-buffer visibility. In: Kajiya J ed. Proceedings of the SIGGRAPH'93. New York: ACM Press, 1993. 231～238

3 Greene N. Hierarchical polygon tilling with coverage masks. In: Rushmeier H ed. Proceedings of the SIGGRAPH'96.

New York: ACM Press, 1996. 65~74

4 Chen S E, Williams L. View interpolation for image synthesis. In: Kajiya J ed. Proceedings of the SIGGRAPH'93. New York: ACM Press, 1993. 279~288

5 Teller S J, Sequin C H. Visibility preprocessing for interactive walkthroughs. Computer Graphics, 1991,25(1):61~69

6 Fu Sheng, Peng Qun-sheng. The design and implementation of a destop real time walkthrough system in a virtual architecture environment. Chinese Journal of Computers, 1998,21(9):793~799
(傅晟,彭群生.一个桌面型虚拟建筑环境实时漫游系统的设计与实现.计算机学报,1998,21(9):793~799)

7 Wang Yi-gang, Bao Hu-jun, Peng Qun-sheng. Accelerated walkthroughs of virtual environments based on visibility preprocessing and simplification. Chinese Journal of Computers, 1998,21(9):787~792
(王毅刚,鲍虎军,彭群生.基于可见性预处理和细节简化的虚拟环境快速漫游算法.计算机学报,1998,21(9):787~792)

8 Wang Tian-yang, Zheng Wen-ting, Bao Hu-jun *et al*. A fast highlight algorithm based on intensity overlay scheme. Chinese Journal of Computers, 1998,21(6):449~505
(王天扬,郑文庭,鲍虎军等.面向 Overlay 技术的镜面高光快速算法研究.计算机学报,1998,21(6):449~505)

9 Fu Sheng, Bao Hu-jun, Peng Qun-sheng. A real time walkthrough algorithm based on overlay techniques for virtual reality. Chinese Journal of Computers, 1998,21(supplement):315~322
(傅晟,鲍虎军,彭群生.一种基于层面叠加绘制技术的虚拟环境实时漫游算法.计算机学报,1998,21(增刊):315~322)

10 McReynolds T, Blythe D, Grantham B *et al*. Advanced graphics programming techniques using openGL. In: Cohen M ed. Proceedings of the SIGGRAPH'98. New York: ACM Press, 1998

11 Heckbert P S, Hanrahan P. Beam tracing polygonal objects. Computer Graphics, 1984,18(3):119~127

12 Ofek E, Rappoport A. Interactive reflections on curved objects. In: Cohen M ed. Proceedings of the SIGGRAPH'98. New York: ACM Press, 1998. 333~342

Real-Time Walkthrough Algorithm for Virual Environments with Global Specular Reflections

HUA Wei PENG Qun-sheng

(*State Key Laboratory of CAD & CG Zhejiang University Hangzhou* 310027)

Abstract Based on the overlay technique, the authors propose a novel algorithm for interactive walkthrough of complex polygonal virtual environments with global specular reflection effects in this paper. This algorithm integrates the global specular reflection preprocessing with visibility preprocessing. The problem of computing global specular reflection is then converted to that of determining the potential visible polygon set (PVS) through each mirror. A mirror-PVS-mapping G is constructed, associating the potential incidence beams with their respective PVS. At the stage of walkthrough, the proposed algorithm can synthesize the global specular reflection in real time by G mapping while calculating the other intensity components of the environment.

Key words Global specular reflection, virtual reality, PVS (potentially visible set), beam tracing.