

基于图象绘制的虚拟现实系统环境*

蔡勇 刘学慧 吴恩华

(中国科学院软件研究所计算机科学实验室 北京 100080)

摘要 虚拟现实的一个重要目标是使用计算机生成逼真的视觉世界,使用户可以对虚拟世界的客体进行交互式考察。虚拟现实的实现有两种方法。传统上,用三维图形学的方法实行建模和绘制。此方法需要繁琐的建模和昂贵的专用绘制硬件,而且图形绘制的质量和场景的复杂性受到很大的限制。基于图象的绘制是实现虚拟现实系统的一种新方法。它克服了三维图形方法的缺点。本文在总结已有技术的基础上,提出一个实现基于图象绘制的虚拟现实系统模型,并据此模型实现一个实验系统及相应的图象工具。

关键词 虚拟现实, 图象绘制, 全景图, 图象拼接。

中图法分类号 TP391

虚拟现实技术就是利用计算机生成逼真的三维视觉、听觉、触觉等感觉,使得装备适当的用户可以通过使用专用设备,用人类自然技能对虚拟环境中的实体进行交互考察与控制。^[1,2]虚拟现实具有两个基本特征:交互性和身临其境。其最重要的性能是操作者可以感受到场景的动态变化并加以实时控制,因此虚拟现实系统需要紧随人的活动(位置、方向的变化)实时生成相应的图象画面。有两个重要指标衡量人所沉浸于虚拟环境的效果和程度。其一是动态特性,其二是交互延迟特性。自然的动态特性要求每秒生成和显示30帧图象画面,至少不能少于10帧/s。交互延迟的时间不应大于0.1s,最多不能大于1/4s。显而易见,图象生成速度成为虚拟现实的重要瓶颈。

传统上,一个虚拟环境是由各类3D几何体合成的。在虚拟环境中漫游是通过实时绘制3D几何体实现的。此方法有3个主要问题。首先,几何体的建模是一个非常繁琐的问题。其次,由于漫游需要实时显示,因此限制了场景的复杂性和绘制的质量。最后,加速绘制所需要的专门硬件极其昂贵,因此限制了虚拟现实的发展。

基于图象的绘制是实现虚拟现实系统的一种新方法。它用全景图集合来构成虚拟环境,在虚拟环境中漫游相当于选择不同的全景图。全景图可以用计算机生成,也可以用全景照相机拍摄,或用普通照相机拍摄后再加拼接。此方法的优越性是生成图象的质量独立于场景的复杂性,因而可以不用专门的硬件加速就能获得真实感很强的图象和实时的交互速度。

* 本文研究得到国家自然科学基金、国家863高科技项目基金和国家“八五”攻关项目基金资助。作者蔡勇,1971年生,硕士,主要研究领域为图形学。刘学慧,女,1968年生,博士生,主要研究领域为虚拟现实。吴恩华,1947年生,研究员,博士生导师,主要研究领域为计算机图形学。

本文通讯联系人:刘学慧,北京100080,中国科学院软件研究所计算机科学实验室

本文1996-11-29收到修改稿

LIPPMAN 制作的电影图系统^[3]是最早的基于图象的绘制系统之一. 在每一个点上, 它前后左右各拍摄一幅照片, 这 4 幅照片形成了这个点的电影片段. 使用者能在每个点上朝 4 个方向看. 当人在环境中走动时, 系统根据人的位置就近选取相应的视点.

如果视线的方向是变化的, 而视点的位置是固定的, 就象固定三角架、旋转照相机一样, 这时环视四周, 所有的帧构成的图叫全景图或环境图.^[4]这个图可以看作一个与视点无关的场景. 一旦生成了环境图, 只要视点不变, 任意视线方向场景的生成过程就是一个从环境图到新的视平面的投影过程. 在“NAVIGATION”系统中, 用环境图的网格来仿真航海.^[5]环境图的实时重投影技术成为可视化周围场景和交互式漫游的关键技术. GREENE 和 HECKBERT^[6]描述了一个把多幅照片合成环境图的技术.

视点插值法是利用存储好的相邻视点的图象生成它们中间的图象. 用视点插值法可以大大减小所需的存储空间. CHEN 和 WILLWMA^[7]的方法是利用照相机的变换和图象的“范围数据”自动决定两幅图象间的对应关系. 这种对应关系以一种“前向映射”的方式存起来. 中间图象每个象素的 3D 屏幕坐标是由内插生成的. 源图象的象素根据对应关系移到内插的位置形成插值图象. 对于移到同一位置的不同象素来说, 它们的深度值决定了其相互间的遮挡关系.

APPLE 公司的 QUICKTIME VR^[8]是一个基于图象绘制方法的虚拟现实实用系统. 它包括两个部分: ① 交互式显示环境. 用软件实现了空间实时导航过程. ② 控制环境. 它用来生成 VR 电影. 交互式显示环境包括两个显示器. “全景电影显示器”允许用户环视、渐远、渐近和在场景中导航. 它还包括一个“热点”拾取功能. 所谓“热点”就是图象中有一块供用户交互控制的区域. “物体电影显示器”允许用户旋转一个物体或从不同方向观察一个物体.

1 基于图象绘制的虚拟现实系统模型

我们分析总结以上各种基于图象的虚拟现实系统, 发现它们都要经过诸如全景图模式选择等过程. 我们得出以下基于图象的虚拟现实系统模型, 如图 1 所示.

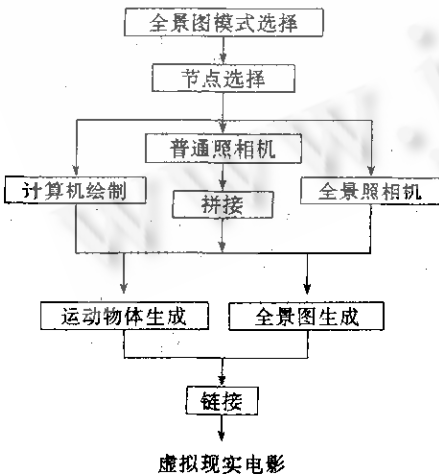


图1

全景图一般有立方体形、圆柱形和球形 3 种模式. 不同的全景图模式的获取难易程度和控制难易程度都有很大不同. 节点选择实际上是确定拍摄方案, 以多大的间隔拍摄, 每个节点如何拍摄等等. 全景图的生成有 3 种途径: ① 计算机绘制, 该方法用计算机图形学的技术给场景建模, 然后再绘制成全景图. 在此方法中, 绘制全景图并不需要实时控制, 因此可以绘制比较复杂的场景和采用真实感较强的光照模型. 此方法的好处是可以容易地得到任意形状的全景图且不需要任何附加步骤. 缺点是建模相当费时和繁琐. ② 用全景照相机拍摄. 它的优点是易于得到全景图且不需要复杂的建模. 缺点是需要昂贵的专门设备. ③ 用普通照相机拍出一系列

相互重叠的照片,然后把它们拼接成全景图.它的优点是可以做成任何形状的全景图且不需要专门的设备.它的另一个好处是可以对每幅照片进行曝光控制,因此得出的全景图光强分布比较平衡.此方法的关键是做一个很好的无缝拼接工具.

运动物体生成和全景图生成一样,同样可以用以上 3 种方法生成.在用照相机拍摄物体时,如果对物体的水平方向和垂直方向各拍摄一圈,就可以对物体进行二维的交互控制.链接是指:①把得到的全景图按一定方式组织起来,供交互式显示用;②把运动物体嵌入到全景图中去,成为“热点”,使用户可以对它进行交互式控制.

2 虚拟现实实验系统 QUICK SHOW

基于以上模型,本文实现了一个虚拟现实实验系统 QUICK SHOW 及其相应的图象工具 STITCHER 和 PICKER. QUICK SHOW 系统采用圆柱型全景图构成场景^[8],选择的实验场景是清华大学图书馆外景.我们选择若干景点,在两个端点上生成全景图,中间的节点只拍摄了一张前向照片.“热点”物体图象是由计算机 3D 方法绘制的飞机和足球模型.目前,此系统实现的功能有:连续照相机环视、渐远、渐近、漫游和物体转动.系统包括两个部分:①交互式显示环境,它用软件实现了空间实时导航过程.其核心算法是圆柱面到视平面的实时投影变换.②控制环境,它用来生成全景图及其“热点”区域.其关键部分是图象拼接工具 STITCHER,实现了图象自动无缝拼接.

2.1 交互式环境

交互式环境的体系结构如图 2 所示.全景图 1、2 和中间节点图分为 3 组,构成了整个漫游环境.用户用鼠标和键盘发出外部命令,控制系统漫游.“热点”足球和飞机图象是挂在全景图 1 上的,在显示全景图 1 时,如果“热点”区域是可见的,则在显示的全景图部分用“热点”图象覆盖.

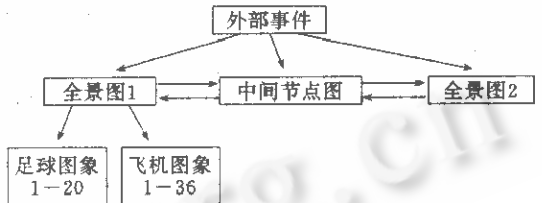


图2

存储调度的目的是协调在硬盘、内存和显示缓存中的图象文件,满足系统限时延迟要求.一般认为延迟应小于 0.1s,最多不大于 0.25s.本文在实践中发现,人的视觉对延迟的敏感程度依环境而定.当视点固定时,延迟应小于 0.1s,但从一个视点跳到另一个视点的过程中,延迟可达 1~2s.据此,本文以节点包为单位在视点跳动时进行内存调度,每个节点包包括此节点所能看到的所有图象,当视点移动到某个节点上,该节点包图象一次全部调入内存.当视点固定时,所有操作均在内存中进行,较好地满足了虚拟现实的延迟性要求,使系统做到实时显示.

从全景图到视平面的投影变换算法是交互式显示环境的核心算法.算法效率直接关系到交互速度.现有的国外文献要么对此避而不谈,要么算法受到专利保护.本文提出了存储矩阵算法^[9]来完成圆柱面到视平面的实时投影变换.其思想是以空间换时间.根据显示视区大小的不变性和视区到窗口映射关系的不变性,预先计算好投影变换的结果,存储在一个与显示视区同样大小的矩阵中.交互式显示时,根据此矩阵查找,就可更新显示视区.用软件实现此算法,达到 25 帧/s 的显示速度,实现了空间实时导航.完全满足了虚拟现实的实时性

要求.

2.2 控制环境

控制环境包括图象工具 STITCHER 和 PICKER. STITCHER 的目的是图象自动无缝拼接,使一系列相互重叠的照片拼在一起,形成全景图. 图象自动无缝拼接是一个涉及到模式识别、优化等多个领域的问题. 本文用图象登录技术结合基因算法进行模式匹配和优化. 由于被拼接的两幅图象一般存在亮度差,本文调整了模式匹配中的评价函数,并对合成后的图象进行色彩拟合,完全实现了图象自动无缝拼接. 图象嵌入工具 PICKER 把物体(如飞机模型)从原来的单一颜色背景中拾取出来,加入到全景图背景中.

2.2.1 图象拼接工具 STITCHER

两张相邻的照片,大约有 50%是重叠的. 由于三角架不可能做到完全水平,照相机在拍摄过程中会出现倾斜、仰翻等现象. 因此 STITCHER 的任务之一是精确定位相邻两张照片间的重叠位置. 拍摄过程中,随着照相机拍摄角度不同,同一场景在不同照片中的光强也会有很大的差异,故 STITCHER 的任务之二是在拼接过程中,调和两幅照片的亮度.

自动寻找图象的重叠位置,实际上可以归结为如下的模式识别问题:有两个矩形区域 A 和 B ,已知 B 中包含一个区域 A_2 , A 与 A_2 是相同的模块,求 B 中 A_2 的位置. 典型的算法是从 B 的左下角起,把每一块与 A 大小相同的区域 C 与 A 相比较,得出评价函数,其值最小的区域就是 A_2 .

评价函数一般定义为:对于 A 和 C 中每一对对应点 p 和 q ,

$$f = \sum (value(p) - value(q))^2 \tag{1}$$

在 STITCHER 中,考虑到两幅图象的亮度不一样,即使是同一模块,由于亮度不同,评价函数的值也很大,我们把评价函数定义为:

$$f = \sum [(dg - dr)^2 + (db - dr)^2] \tag{2}$$

其中

$$\begin{aligned} dr &= value(p(r)) - value(q(r)) \\ dg &= value(p(g)) - value(q(g)) \\ db &= value(p(b)) - value(q(b)) \end{aligned}$$

评价函数值最小的区域就是 A_2 .

在第 1 幅图中,如果确定了区域 A ,根据 50%的重叠原理,在第 2 幅图中,加上误差范围,即可得区域 B . 区域 A 必须包括足够多的物体特征,否则容易导致算法失败. 我们注意到,如果一个区域包括的物体特征越多,则它和周围区域的差别越大. 因此我们可以用和公式(2)类似的评价函数来寻找区域 A . 给定一个区域 S ,把它和上下左右各相隔一个象素的同样大小区域相比,用公式(2)的评价函数得到 4 个值 f_1, f_2, f_3, f_4 ,令区域 S 的评价函数

$$F = f_1 + f_2 + f_3 + f_4 \tag{3}$$

对于第 1 幅图的右半区中的任一 S ,评价函数 F 越大,则越可能是要找的区域 A .

我们注意到,此过程和寻找 A_2 的匹配过程一样,都是一个模式识别问题. 不同点在于寻找 A_2 的过程是个最优问题,必须找到最小的评价函数值,因此它必须在全部问题空间内进行搜索,而区域 A 的寻找过程是个较优问题,只要找到满意解就可以了,因此可以用一些概率算法来加速求解过程. 我们用基因算法^[10]来选择区域 A .

第 1 步:在第 1 幅图的右半区中随机挑选 20 个区域作为初始群体 $Z(0)$,令 $I=0$. 用这

些区域的左下角坐标来标记它们,分别为 $a_1(X_1, Y_1), a_2(X_2, Y_2), \dots, a_{20}(X_{20}, Y_{20})$.

第 2 步:重复

(1) 评价 $Z(I)$ 中每个个体的适应值,适应值取公式(3)中的评价函数 F 值.

(2) 如果 $Z(I)$ 中有个体 a_i ,它的评价函数值 F 大于给定的阈值,则此个体被选中,得到区域 A . 算法结束.

(3) 用 roulette wheel 选择法选择 10 个“父母”.

(4) 把 10 个“父母”按 X 坐标大小顺序排列.

(5) 把相邻的“父母”两两配对,进行重组.重组方法为:对于“父母”个体对 $a_i(X_i, Y_i), a_j(X_j, Y_j)$,在以点 $(X_i, Y_i), (X_i, Y_j), (X_j, Y_i), (X_j, Y_j)$ 组成的矩形中,随机取 4 个点作为子代区域的左下角坐标,由此生成 4 个子代个体.

(6) $I=I+1$,重复第 2 步直到得到满意解.

roulette wheel 选择法是:设 W_1, W_2, \dots, W_n 是个体的适应值,给每个个体 i 一个概率值 P_i .

$$P_i = \frac{w_i}{\sum_{j=1}^n w_j} \quad (4)$$

按此概率值来选择个体.

图象定位以后,如果简单地把两幅图象拼接起来,由于亮度差的存在,拼接处会有明显的一条缝. STITCHER 用颜色拟合的方法来调和相邻图象的亮度,生成无缝的合成图象.以合成图象的拼接线为中心,两边各取 10 个象素点的距离,此区域为矫正区.在原来的两幅图象中,分别计算矫正区 RGB 值的累加和.令第 1 幅图中为 R_1, G_1, B_1 . 第 2 幅图中为 R_2, G_2, B_2 . 计算它们的差值和差值的百分比:

$$DR = (R_1 - R_2) / 2, \quad DG = (G_1 - G_2) / 2, \quad DB = (B_1 - B_2) / 2;$$

$$DR1 = DR / R_1, \quad DG1 = DG / G_1, \quad DB1 = DB / B_1;$$

$$DR2 = DR / R_2, \quad DG2 = DG / G_2, \quad DB2 = DB / B_2$$

以拼接线为中心,两边各取 100 个象素距离为颜色渐变区.令 D 为象素到拼接线的象素距离.在第 1 幅图的颜色渐变区域内,每个象素的 RGB 值更改为:

$$r = [r * D + r * (1 - DR1) * (100 - D)] / 100$$

$$g = [g * D + g * (1 - DG1) * (100 - D)] / 100$$

$$b = [b * D + b * (1 - DB1) * (100 - D)] / 100$$

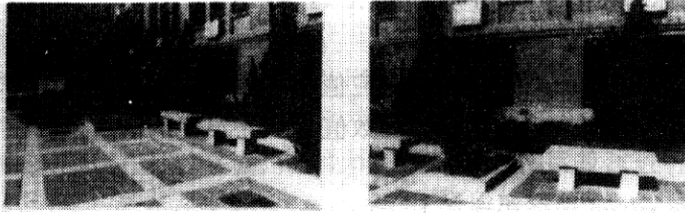
在第 2 幅图中

$$r = [r * D + r * (1 + DG2) * (100 - D)] / 100$$

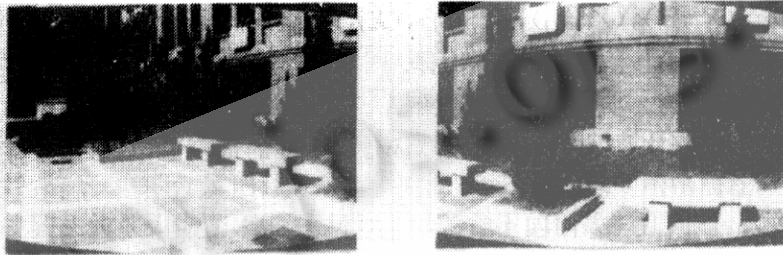
$$g = [g * D + g * (1 + DG2) * (100 - D)] / 100$$

$$b = [b * D + b * (1 + DB2) * (100 - D)] / 100$$

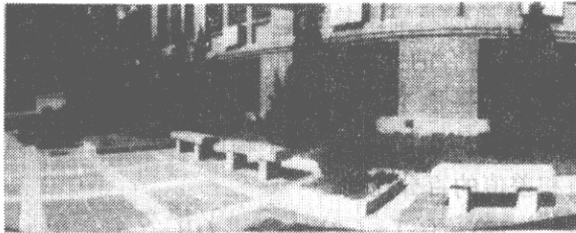
经过颜色拟合后的合成图象达到无缝拼接的效果,见图 3.



(A) 原始图象



(B) 投影到圆柱面上的图象



(C) 拼接后的图象

图 3 图象的无缝拼接

2.2.2 图象嵌入工具 PICKER

图象嵌入工具 PICKER 把物体(如飞机模型)从原来的单一颜色背景中拾取出来,加入到全景图背景中. 自动识别背景的过程如下:

取原始图象边缘一圈的点, 计算其 RGB 值的累加和, 并求得平均值 r, g, b .

对原始图象上每一点 P , 它的 RGB 值分别为 $P(r), P(g), P(b)$.

如果 $|P(r)-r| < \epsilon$ 且 $|P(g)-g| < \epsilon$ 且 $|P(b)-b| < \epsilon$

其中 ϵ 为常整数. 系统中 ϵ 取 10. 则 P 为背景上的一点, 否则, P 为物体上的点, 把 P 写入全景图背景中.

“热点”挂入后的全景图见图 4.



图 4 “热点”挂入后的全景图

3 应用与展望

基于图象绘制的方法利用全景图,特别是圆柱形全景图来实现场景的建构.全景图是一个与视线无关的图象.通过实时的图象处理过程,用户可以从任意方向进行观察.多幅全景图链接起来构成一个场景,用户可以在此场景中漫游.另外,用户可以通过一个二维帧矩阵从不同角度观察一个物体.

基于图象绘制方法的应用范围有虚拟漫游、虚拟房地产观察、建筑物可视化、虚拟博物馆、虚拟商店和虚拟现实游戏.随着压缩技术的提高与网络技术的提高,在网络上进行实时的全景图漫游也将成为现实.这为可视电话会议和交互式电视打开了通道.基于图象绘制的方法还解决了大多数 3D 虚拟现实系统中存在的物体细节尺度问题,它自动提供了物体合适的细节层次.显示的图象都是在一定位置范围内得到的场景图象,当视点变化时,图象也随着变化,这样,显示的场景永远有合适的细节层次.

此方法的主要限制是场景必须是静态的,运动限制在特定的点上.下一步的工作将是建立随着时间变化的全景图(如从白天到夜晚同一场景的全景图)和包含运动的全景图(如打开一扇门).随着时间变化的全景图可以由时间延迟拍摄法得到.解决静态环境限制的方法是结合图象绘制和 3D 绘制.大多数背景是静态的,可以用全景图来绘制.运动的物体用 3D 绘制.绘制好的物体用 alpha masking 或 Z-buffering 的方法实时合成在全景图背景上.

在场景中平滑移动非常困难.对于用计算机绘制的场景,可以用视点插值的方法来解决.它需要场景的深度信息.如何获取实拍图象的深度信息是下一步工作之一.圆柱形全景图的限制是圆柱的上底和下底与侧面不一致,如果用球形全景图则能解决此问题.但对于实拍场景,球形全景图较难获取,这也是要解决的问题之一.

虚拟现实的终极目标是让用户分不清什么是真实,什么是虚拟.由于基于图象绘制的方法能用现实中的图象来构成虚拟导航中的场景,因此成为最有希望的虚拟现实方法之一.

参考文献

- 1 Carolina Cruz-Neira. Virtual reality overview, overview of virtual reality systems. Course Notes 23 of SIGGRAPH'93, 1993.
- 2 吴恩华,刘学慧.虚拟现实与真实感图形生成.中国图象图形学报,1997,2(4):205~212.
- 3 Lippman A. Movie maps: an application of the optical video disc to computer graphics. Computer Graphics (Proc. SIGGRAPH' 80), 1980. 32~43.
- 4 Greene N. Environment mapping and other application of world projections. Computer Graphics and Applications, November 1986, 69110:21~29.
- 5 Hodges M, Sasnett R. Multimedia computing——case studies from MIT project Athena. Addison-Wesley, 1993.

- 89~102.
- 6 Greene N. Creating raster omnimax images from multiple perspective views using the elliptical weighted average filter. *IEEE Computer Graphics and Applications*, 1986, 6(6), 21~27.
 - 7 Chen S E, Williams L. View interpolation for image synthesis. *Computer Graphics (Proc. SIGGRAPH'93)*, 1993. 279~288.
 - 8 Chen S E. QuickTime VR— an image-based approach to virtual environment navigation. *Proc. SIGGRAPH'95*, 1995. 25~38.
 - 9 蔡勇. 基于图象绘制的虚拟现实系统技术研究[硕士论文]. 中国科学院软件研究所, 1996.
 - 10 Yao Xin. An overview of evolutionary computation. *Chinese Journal of Advanced Software Research*, 1996, 3(1), 12~29.

IMAGE-BASED RENDERING: A TECHNOLOGY FOR VIRTUAL REALITY SYSTEM

CAI Yong LIU Xuehui WU Enhua

(Laboratory of Computer Science Institute of Software The Chinese Academy of Sciences Beijing 100080)

Abstract The most important goal of the virtual reality is to create a virtual world by computers to allow users control the virtual objects interactively. Traditionally, the virtual reality systems use 3D computer graphics to model and render a virtual environment in real time. However, this approach usually requires laborious modeling and expensive special purpose rendering hardware. The rendering quality and scene complexity are often restricted due to the real-time constraint. Image-based rendering is a new approach in composing a virtual environment in which a set of panoramic images are used to compose the virtual environment and walk in the space is accomplished by “hopping” to different panoramic points. This paper provides a model for creating an image-based virtual reality system. By using this model, an experience system referred to as QUICK SHOW is accomplished with some additional image tools implemented.

Key words Virtual reality, image-based rendering, panoramic image, images stitching.

Class number TP391