

一组新的柱面全景图投影公式及其交互浏览器的 Java Applet 实现算法

潘 伟¹, 姚晓静²

(1. 厦门大学自动化系, 2. 厦门大学南洋研究院, 福建 厦门 361005)

摘要: 事先将在同一视点环绕拍摄的多幅部分重叠的图像正投影合成为一幅无缝的图像文件——柱面全景图, 它代表在拍摄点看到的周围 360 度视角的三维场景。在浏览时将用户观察的相应部分实时进行反投影为平面图像, 然后投影到屏幕上。随着视角的改变, 就可实现虚拟场景的漫游。柱面全景图的环视浏览具有很好的真实感和沉浸感, 对计算机的硬件要求不高。本文详细介绍了这种全景图浏览器的实现原理, 给出了一组全新且简便的投影与反投影公式, 并在此基础上开发了交互式的 Java Applet 柱面全景图浏览器。最后给出了关键的 JAVA 代码。

关键词: 全景图; 虚拟现实; 交互浏览; Java Applet; 投影公式

中图分类号: TP 391. 41

文献标识码: A

文章编号: 0438-0479 (2005) 04-0479-05

实现交互式的虚拟全景空间技术总体可分为基于建模与基于图像两种模式。

基于建模方式的全景空间可以利用三维仿真开发包, 如 WTK, VEAG 等, 也可以利用通用的三维建模软件如 3Dmax 建模, 然后输出为中间格式, 再利用 cult 3D 或 VRML 等技术开发成交互式的全景空间^[1]。基于建模方式的全景空间可以在任意视点全方位地交互浏览全景空间, 但建模时间长和要求较高的专业软件使用技巧, 交互浏览时计算量很大, 对自然景观 (如树木, 花草) 等达不到真实的视觉效果。

Apple 公司推出 QuickTime VR, 第一次向大众展示了全景图像技术的视觉魅力。该系统采用与传统基于建模模式截然不同的思路, 利用一系列相关的图像或连续的视频为原始数据, 经过处理把多幅图像环绕组织成虚拟全景空间。基于图像的虚拟场景的构建主要有三种方法: 柱面全景、立方体全景和球面全景^[3~8]。目前真正得到广泛使用的是柱面全景图。柱面投影模式具有采集手段简单、拼接速度快、均匀采样, 可方便展开为平面模式等优点。虽然在实际使用中在垂直视域方面存在限制, 但是对室外大场景来说这不是大问题。

已出现一些商业软件, 比如 QuickTime VR、Cool 360、HotMedia、Panorama Maker、Realviz Stitcher 等。这些软件都能将在同一视点环绕拍摄的数码照片无缝连接成一幅长度很宽的柱面全景图, 然后利用自带的全

景图播放器进行不交互或交互式的浏览。这些浏览器一般功能都比较完整, 但文件尺寸过大, 都是可执行文件, 不能嵌入到用户自己开发的程序中。

本文在文 [9, 10] 的基础上, 推导出了一组全新且简明的柱面全景图投影与反投影公式, 在这组公式的基础上对柱面全景图浏览器的实现原理和关键技术进行讨论。由于 Java 语言具有平台无关性和良好的网络功能, 所以采用 JAVA 语言来描述其实现。程序代码行数不多, 可以实现交互浏览功能, 还能嵌入其它应用程序中。

1 柱面全景图的正投影与反投影

1.1 柱面全景图与数码照片的投影关系

柱面全景图的生成是在同一视点 O (图 1-a) 环绕拍摄一系列数码照片 (一般为 15 ~ 20 张), 相邻之间的照片要有比较大的重叠, 将照片利用正投影公式 (1) 投影到一个圆柱面上, 相邻的两张投影图相交地方相互融合。所有照片都投影到圆柱面后得到一圈圆柱形的 360 视角无缝的图像。沿柱面长度方向将圆柱面图像剖开, 形成一幅矩形的可以保存到磁盘上的平面图像 (见图 2)。

图 1 (a) 为圆柱面投影模型的几何表示俯视图, 虚线圆为投影柱面的水平截面, 圆外的每一条直线代表一幅数码照片, 圆心 O 处的长方形代表数码照相机。图 1 (b) 为一幅照片 $ABCD$ 及其在图柱面上投影 $A'B'C'D'$ 的关系, 图 1 (c) 为图 1 (b) 的局部放大图, P 为数码相片上 P 点在柱面上的投影点; O 为视点 (数

收稿日期: 2004-12-31

作者简介: 潘伟 (1958 -), 男, 副教授, wpan@xmu.edu.cn

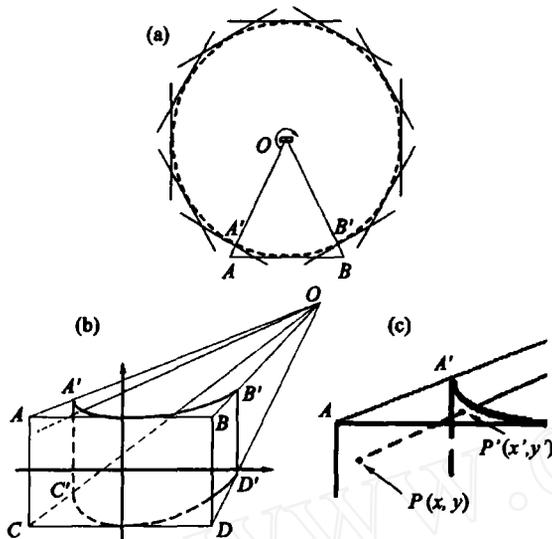


图 1 柱面全景图与数码照片的投影关系
Fig 1 The project relationship of cylinder panorama and digital images

码相机位置), $OA \sim OD$, OP 为投影线.

1.2 正投影与反投影公式

柱面全景图的正投影是指将平面照片投影到柱面形成全景图的过程;反投影是将柱面全景图在某个特定的观察区域投影到柱面的切平面上供屏幕显示的过程.有很多文献都给出了正投影与反投影的变换公式^[9,10].但这些公式比本文提出的公式复杂,计算环节多.其原因是采用的计算参数不合理,其后果则是在进行连续反投影时增加了算法的复杂度.

本文提出一种新的参数设计方法,它采用的参数合理,公式简便而且非常适合于显示不断旋转着的柱面全景图,给观察者感觉好像站在视点处旋转观察周围景物的效果.

反投影时利用由柱面全景图展开的平面图像(图2中的长矩形),假定某时刻从视点 O 观察到的柱面部分为 $ABCD$ (图1~3),假定全景展开图全长为3000像素, $ABCD$ 的宽为300像素,高为300像素;反投影到平面后的图像大小也采用 300×300 像素;定义两个长度为 300×300 的一维数组: $\text{Pixel1}[]$ 用于保存从柱面全景图的截取的一个矩形区域的每一个像素(图4上), $\text{Pixel2}[]$ 用于保存反投影到平面图像的每一像素(图4下). (x, y) 与 (x', y') 分别代表在同一投影仪上平面的坐标与柱面上的坐标.图3(上)显示了 P 与 P' 投影到水平面后的 x 与 x' 的关系;图3(下)显示了沿观察点 O 到 P 与 P' 投影线与柱面长度方向的切面上 y 与 y' 的几何关系:

设圆柱面的半径 $ON = r$ (可以由圆柱面展开图的长度除以2得到),从图3(上)知,

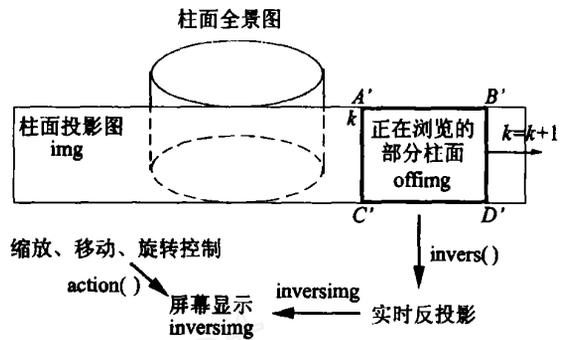


图 2 从柱面全景图变换为屏幕显示图的过程及交互操作
Fig 2 The process and interactive manipulate from cylinder Panorama to screen display

$$= \arctg(x/r).$$

由于弧长等于半径乘以弧度,得:

$$x = r \times \alpha = r \times \arctg(x/r).$$

由图3(下) OPP_1 相似于 $OP'P_1$, $OP_1 = r$ 从图3(下)的 OP_1 等于图3(上)的 $OP = r/\cos$,故:

$$y'/y = r/OP.$$

得:

$$y = y' \times r / (r/\cos) = y' \times \cos.$$

综合上述推导,柱面上的坐标 (x, y) 与平面上的坐标 (x', y') 的关系为:

$$x = r \times \arctg(x'/r) \tag{1}$$

$$x = r \times \alpha \tag{2}$$

$$y = y' \times \cos \tag{3}$$

公式(1)~(3)通常称为正投影公式,如果从 (x, y) 出发,可由式(1)~(3)导出反投影公式:

$$x = r \times \text{tg}(x'/r) \tag{4}$$

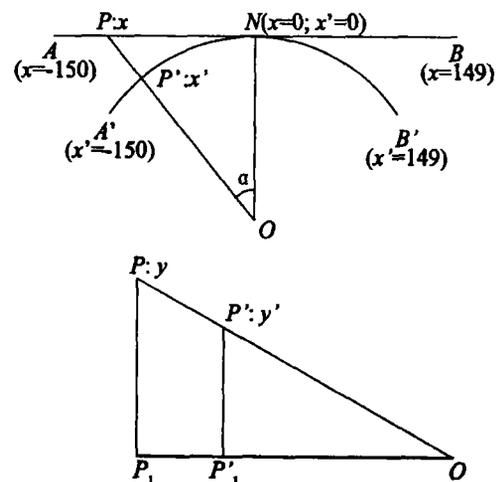


图 3 柱面投影图与平面图之间的几何关系
Fig 3 The geometry relationship of cylinder panorama and ichnography

$$y = y / \cos \tag{5}$$

$$= \arctg(x/r). \tag{6}$$

公式 (1) ~ (6) 比现有可查到的文献^[9,10]的正投影与反投影公式都简便。

1.3 像素之间的变换公式

在程序设计时,要实现柱面图像与平面图像之间的变换,根本上是图像像素之间的变换.以图 3、4 为例,就是如何将柱面 P 点的像素信息传递给平面上的 P 点.在 Java 程序中实现像素之间的信息传递主要方法是先将图像全部像素逐点读入到一个像素数组中 (Pix1[]),进行相应的位置变化后得到另一个像素数组 (Pix2[]),最后用第二个数组生成反投影图像.利用本文提出的投影公式 (1) ~ (5),两个数组像素之间的变化公式如下:

$$\begin{cases} s = (100 - y) + (x + 150); \\ t = (100 - y) + (x + 150) \\ \text{pix2}[t] = \text{pix1}[s] \end{cases} \tag{7}$$

2 交互式全景图浏览器的设计

在浏览柱面全景图时,观察者要求视场中反映出来的是一幅幅随观察角度连续变化的平面图像.可以在全景图中截取相应的区域图像,利用反投影算法反算出屏幕上应显示的图像(图 4).这里有一个问题是要特殊注意的,由于柱面投影图像是无缝连接后展开为矩形的图像,所以当窗口扫描到柱面全景展开图像的右边缘时,如何实现全景图像的一侧与另一侧的无

缝隙连接(全景图像的一侧是代表某一观测角度的起始的位置,另一侧自然是其结束位置),便成为能否实现循环的连续无缝隙浏览的关键。

这个问题可以这样解决:在图 2 中用一个参数 k 记录取景窗口 $ABCD$ 的扫描位置左上角,当 k 与右边缘 W ($=3000$) 距离小于扫描窗口的宽度时(剩余的像素数少于窗口相应方向视场所需像素数大小时,本文为 300 像素),实时从展开图的左边取宽为 $W - k - 1$ 宽度区域补充到右边剩余区域就可以得到应该观察到的图像,当 k 增大到 W 后令 $k = 1$,取景重新开始一次循环。

2.1 柱面全景图浏览程序的设计思路

本文采用 JAVA 小程序实现柱面全景图的浏览,可以嵌入到网页中,在 Internet WWW 服务上供用户下载访问.设计过程分为下列 3 个步骤:

(i) 数码照片拍摄:在场景中央将数码相机(300 万像素)放三角架上,调节三角架上的两个水平仪,使相机转动 360 度时大致保持在同一水平面上,每旋转 18 度左右拍摄一张照片,共拍摄 20 张.将数码照片从相机导出到计算机,得到 20 张 jpg 格式的平面图像。

(ii) 柱面全景图生成:将这 20 张照片导入 Cool 360(也可以利用本文的正投影公式完成),生成一幅柱面全景图(大小为 6~7 Mb),利用 Photoshop 软件将图像大小适当缩小,本文缩小为 500 kb 左右,分辨率为 72 像素/英寸

(iii) 全景图浏览器设计:利用 JDK 工具生成 class 文件,嵌入到网页中供浏览,效果截图见图 5,程序的关键代码见下一节。

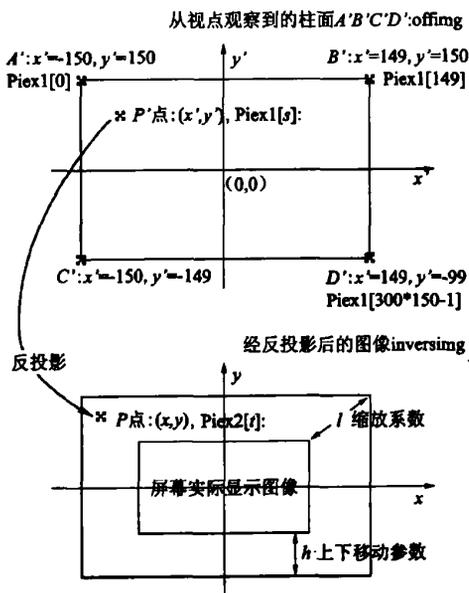


图 4 投影与反投影的像素关系及显示交互控制参数

Fig 4 The pixel relationship of project unproject and display interactive control parameter

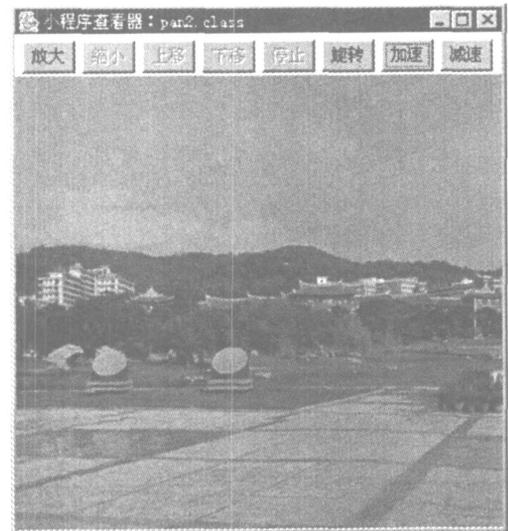


图 5 交互式柱面全景图浏览器

Fig 5 Interactive cylinder panorama browser

3 程序的关键代码

因为环视的需要,场景要不断旋转,Java小程序要实现 Runnable接口,利用线程来实现动画,同时采用双缓冲技术,使动画不会闪烁.下面为主要的代码:

```
public class view extends Applet implements Runnable
{
//定义三个图像对象,具体含义见图 2
Image img, offimg, inversimg;
int pix1[] = new int[p * q]; //存放 offimg图像的数组
int pix2[] = new int[p * q]; //存放 inversimg图像的数组
.....
public void init() //applet初始化 {
txt = getParameter(" txt "); //由网页文件取得全景图文件名
img = getImage(getCodeBase(), txt); //取得图片
offimg = createImage(300, 300); //建立 Image对象
offgra = offimg.getGraphics(); //取得绘图对象
.....
}
public void start() //applet线程开始,代码省略
public void run() //线程启动后,执行的方法
{
while(true) //线程不断循环
{
if(rotate) { //环视浏览时
offgra.clearRect(0, 0, 300, 300); //清除 offimg
if(k <= w - 300) //当还没有到图的右边时
{
//从柱面全景图上截取一块 300 x300的图像
offgra.drawImage(img, 0, 0, 300, 300, k, 0, k + 300, 300, this);
k = k + 1; //显示右移 1像素,相当于旋转 1个像素
}
else
{ if(k < w) //当接近图的右端,但没有出去时
{
//全景图右边剩余部分绘制到 offimg
offgra.drawImage(img, 0, 0, w - k, 300, k, 0, w, 300, this);
//从柱面图左边补上不足部分
offgra.drawImage(img, w - k, 0, 300, 300, 0, 0, 300 - (w - k), 300, this);
k = k + 1;
}
else //当显示全部移出全景图右边瞬间
{
//从柱面图左边从头开始
offgra.drawImage(img, 0, 0, 300, 200, 0, 0, 300, 200, this);
k = 1;
}
}
invers(); //调用反投影计算
```

```
repaint(); //旋转时调用重绘方法
}
else{ //当旋转停止时
repaint(); //放缩,上下移动后的重绘
}
//停止 40毫秒后再执行一次 run()方法,形成动画
try {
Thread.sleep(40); //暂停 40毫秒
}
}
}
void invers() //反投影方法
{ //建立取像素对象
PixelGrabber pg = new PixelGrabber(offimg, 0, 0, 300, 300, pix1, 0, 300);
pg.grabPixels(); //提取像素
//从平面图像的左上角开始逐像素反投影计算
for(int y=0; y<200; y++){ //逐行
for(int x=0; x<300; x++){ //在同一行中逐像素
a = Math.atan(x/r); //反投影计算
x1 = (int)(r * a); //反投影计算
y1 = (int)(y * Math.cos(a)); //反投影计算
t = (150 - y) * 300 + (x + 150);
s = (150 - y1) * 300 + (x1 + 150);
//传递数组 1的像素信息到数组 2的的相应位置
pix2[t] = pix1[s];
}
}
ImageProducer producer = new MemoryImageSource(p, q, pix2, 0, p);
//建立图像对象
inversimg = createImage(producer); //产生反投影后的图像
}
public void update(Graphics g) //绘图
{ //屏幕显示只需同一绘图命令
g.drawImage(inversimg, 0, 0, 300, 300, 0 + 3 * l, 0 + 2 * l + h, 300 - 3 * l, 300 - 2 * l + h, this); //将 offimg贴出来
} //e为放大,缩小系数, h为上下移动系数
//鼠标点击按钮的响应
public boolean action(Event e, Object o)
{
String temp = alpha.toString(); //取得按钮上的文字
if(temp == "放大")
{l = l + 1; //缩放
}
.....
}
}
```

4 讨论

柱面全景图在虚拟场景展示方面有它独到的优

点,但在制作过程中应考虑到以下几点,才能做出满意的柱面全景图:

(i) 在拍摄数码相片,应在相邻两张留出 15% 以上的重叠空间(甚至 25%),选择好的天气和好的投影软件,才能得到好的柱面全景图;

(ii) 如果把全景图放在 Internet 供用户下载访问,因为全景图的文件相当大(本文中约为 500 kb),必须考虑到用户的网络带宽和浏览时的等待时间,进一步的研究是利用小波图像压缩算法,把柱面全景图进行三级小波图像压缩,并利用零树编码.在程序中先将第三级小波压缩的图像先下载到用户端,形成稍有失真的浏览(不会影响视觉效果).等全部编码全部到后,再给用户一个完全质量的图像.

(iii) 编辑程序时,可以放置一个变量 txt,用于接收从网页引用 Class 文件时的图像文件名.只要生成一个类文件,就可以实现多个全景图的播放:

```
<APPLET code = "view.class" width = 350 height = 350 >
<param name = "txt" value = "jg.jpg" >
</APPLET >
```

(iv) 由于美国微软公司发行的 windows 系列操作系统不支持 JAVA,用户在浏览嵌有 Java Applet 的网页时,只看到该位置为一片灰色的区域,看不到效果.浏览用户可到 SUN 公司的网站下载 JAVA 运行环境 jre,安装后就能看到 Java Applet 的效果了,下载 jre 的

地址为: <http://java.sun.com/j2se/1.4.2/download.html>

参考文献:

- [1] 石巧珍,刘晓,蒋刚毅.交互式三维虚拟校园的设计[J]. 电脑与信息技术,2003,(4):1-5.
- [2] 杨记,陈孝威.基于全景图像的 VR 建模研究及实现[J]. 计算机应用研究,2004,(1):249-254.
- [3] 孙立峰,钟力,李云浩,等.拟实景空间的实时漫游[J]. 中国图像图形学报,1999,(6):507-513.
- [4] 韦群,高丽,龚雪晶.基于立方体全景图的虚拟场景浏览技术研究及实现[J]. 中国图像图形学报,2003,(9):1061-1066.
- [5] 张辉,崔杜武.全景图像生成算法的研究与实现[J]. 计算机工程,2003,(4):95-97.
- [6] 唐进等.柱面全景图像的一种实现方式[J]. 小型微型计算机系统,2002,(11):1363-1365.
- [7] 刘建威,崔杜武.基于全景图的多视点虚拟空间漫游技术[J]. 计算机工程,2004,(1):153-154.
- [8] 潘立公,杨晓安,郝建新,等.一种 360 度柱面全景图快速生成算法[J]. 微型电脑应用,2001,17(4):19-23.
- [9] 宋利,周源华,周军.一种全景图浏览器的 JAVA 实现算法[J]. 计算机工程与科学,2003,(12):1-2,96.
- [10] 潘华伟,邹北骥.一种圆柱形全景图生成新算法及其实现[J]. 计算机工程与科学,2003,25(6):13-16.

A Group of New Cylinder Panorama Projection Formulas and the Realization Arithmetic of Its Interactive Browser in JAVA Applet

PAN Wei¹, YAO Xiao-jing²

(1. Department of Automation, Xiamen Univ., 2. Research School of Southeast Asia Studies, Xiamen Univ., Xiamen 361005, China)

Abstract: Panorama image is a new implementation of interactive virtual scene on the internet lately. The cylinder panorama is in common use at present, beforehand, it synthesizes in positive project a few partly clinker-built images which are taken in circle at one eye point to a seamless image file, it represents the view of 360 degree three-dimensional scene around the shot point. When you browse, it will unti-project the corresponding part of the panorama to an ichnography in real time, then project it on the screen, which represents the view of the three-dimensional scene by the user at the eye point. Along with the change of the visual angle, you can realize roaming in the virtual scene. The sweep browser of the cylinder panorama has all-right third dimension and immersion, it requires not too much to the computer, and combine to GIS, it can fleetly empolder three-dimensional GIS system ground on the images. This article gives a detailed introduction of the realization elements of this kind of panorama browser, and presents a group of bran-new also simple formulas of projection and unti-projection, furthermore, it also empolder the interactive cylinder panorama browser in Java Applet based on these. Finally, it imparts the key JAVA code.

Key words: panorama; virtual realism; interactive browser; java applet; projection formula