

文章编号: 1005—0523(2008)04—0056—05

全景图生成中圆柱面正投影算法研究

杨 刚¹, 石云飞²

(1. 陕西理工学院 计算机系, 陕西 汉中 723000; 2. 西北工业大学 机电学院, 陕西 西安 710072)

摘要:在采用 IBMR 的方法构造虚拟环境的过程中, 全景图的生成是一个重要的环节. 在把用普通数码相机得到的一组相关照片生成全景图时, 需要将照片经过投影、拼接和平滑化等步骤生成一幅全景图像. 主要把全景图生成中圆柱面正投影算法进行分析、讨论、归纳、总结, 并提出一些新观点. 最后验证了此算法及其讨论的正确性.

关 键 词:全景图; 成像模型; 圆柱面正投影; 算法

中图分类号: TP391 **文献标识码:** A

虚拟全景空间系统又称为 VPS 系统. VPS 系统分为虚拟全景空间生成平台与虚拟全景空间浏览器. 生成平台又分为全景图像生成器与空间编辑器. 前者负责把多幅相互间有少量重叠区域的普通图像或连续的视频拼接为全景图像, 把众多的全景图像组织为虚拟全景空间的功能则是由空间编辑器实现的, 生成虚拟全景空间的过程如图 1 所示^[1]. 根据视点空间建模的不同, 对于固定的单视点来说, 一般将全景图模型分为立方体面、球面和圆柱面三种模型^[2-4]. 本文主要讨论圆柱面全景图模型中圆柱面正投影算法.

1 成像模型

图像是视觉信息表示的一种物理形式, 要认识图像产生及其所携带信息的内在性质, 必须用适当的数学模型去表征图像. 在实景图像的拍摄过程中, 我们使用普通数码相机, 采用小孔成像模型^[5-6].

先看一下简易照相机的成像光路图^[7] (如图 2 所示), 观察此图片可以发现, 照相机成像的原理与人眼成像的原理是一样的, 不过人像成像只是一个会自动调节的晶状体, 而照相机的关键部件则是一

组镜头组. 虽然是很复杂的镜头组, 但起主要作用的还是凸透镜, 在示意图中可用一片凸透镜来代替. 凸透镜成像公式为:

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v},$$

其中 f 为焦距 (对凸透镜而言 f 为正值), u 为物距, v 为相距

在照相机中, 由于 $u \gg f$, 所以有 $v \approx f$.

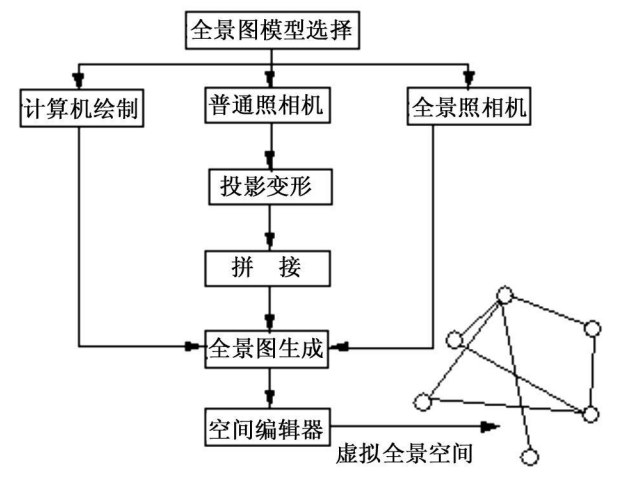


图 1 虚拟全景空间生成流程图

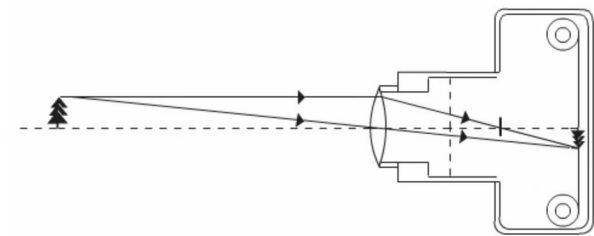


图 2 照相机成像原理图

由此可以得到如图 3 所示的小孔成像模型示意图,平面 S 为二维成像平面(即视平面), O 为小孔位置(光学中心), $P(X, Y, Z)$ 为照相机坐标系 $OXYZ$ 中一个三维空间点. 设 S 上的点 $p(x, y)$ 是 $P(X, Y, Z)$ 在视平面上的投影(成像), O 与 C 之间的距离 f 为该光学系统的焦距. 为方便起见,一般取 C 为视平面的中心点,光轴方向为照相机坐标系的 Z 轴方向,视平面与光轴垂直,且其 x 、 y 轴分别平行于照相机坐标系的 X 轴和 Y 轴.

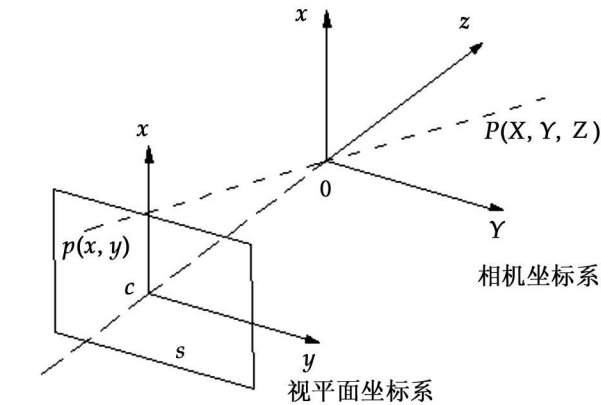


图 3 小孔成像模型示意图

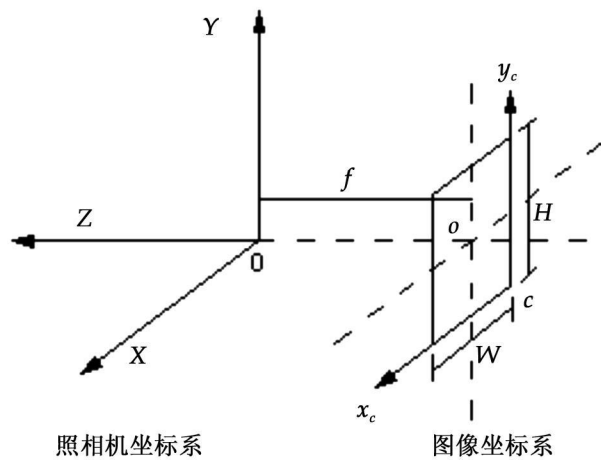


图 4 照相机坐标系中的焦距

由光学成像原理, $p(x, y)$ 与 $P(X, Y, Z)$ 之间有如下几何关系^[8]:

$$\begin{cases} x=f \frac{X}{Z} \\ y=f \frac{Y}{Z} \end{cases} \quad (1)$$

上式称为小孔成像公式. 照相机坐标系 $OXYZ$, 也就是观察坐标系.

2 像素坐标

在上面讨论的小孔成像模型中, 计算所涉及的物理量都是以实际的物理单位作为基本度量单位, 但是实景图像是以像素作为基本的度量单位, 这就需要统一基本度量单位. 我们将像素作为统一的基本度量单位. 为了实现此目的, 首先要求出实景图像上每一个像素点在照相机坐标系中的像素坐标.

在照相机坐标系 $OXYZ$ 中, 视平面的位置关系如图 4 所示, 视平面在 $Z=-f$ 的平面上. 假设照相机的镜头中心不存在偏差, 则实景图像的中心是照相机的光轴(照相机坐标系的 Z 轴)与视平面的交点, 由于照相机坐标系的 X 轴和 Y 轴分别平行于图像坐标系的横轴与纵轴, 那么, 对于实景图像上任何一个像素点 (x, y) , 由图 4 不难得到它在照相机坐标系 $OXYZ$ 下的像素坐标为 $(x-W/2, y-H/2, -f)$ ^[8]. 其中, W 和 H 分别是实景图像宽度和高度, x 、 y 、 W 、 H 以像素作为基本度量单位.

3 普通镜头的像素焦距

首先, 估算像素焦距. 如图 5 设实景图像 I 的宽度和高度分别为 W 和 H (像素), 相机水平半视角为 θ , 相机焦距为 f .

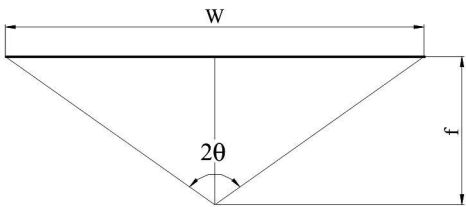


图 5 相机的像素焦距与实景图像宽度关系示意图

水平视角 2θ 决定了拍摄景物的多少, 它由水平

旋转照相机一周来估计: $2\theta = \frac{360}{n_0}$, 其中 n_0 是水平旋转相机一周, 使相邻位置拍摄的图像恰好左右衔接时的实景图像张数. 则 f (像素) 为^[8]: $f = \frac{W}{2 \tan \theta}$.

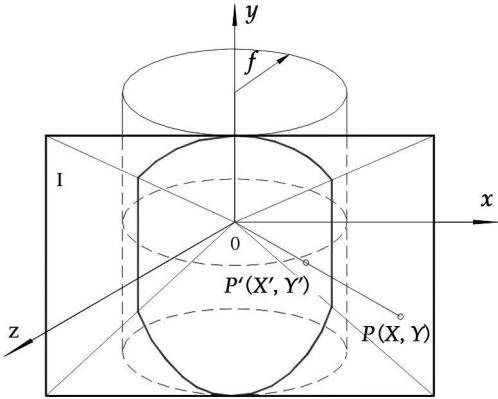


图 6 柱面正投影示意图

4 圆柱面正投影算法^[8]

实景图像的柱面正投影算法是为了将多幅实景图像投影到一个圆柱面上, 一方面消除了实景图像之间可能存在的重复景物信息; 另一方面, 也得到了每幅实景图像上像素点在视点空间中的方位信息.

4.1 圆柱面方程的三维表示

如图 6 所示, 用数码相机拍摄的一张实景图像 I , $p(x, y)$ 是实景图像 I 上的任意一个像素点. 由前面结果可以得到 $p(x, y)$ 在相机坐标系下坐标为 $(x - W/2, y - H/2, -f)$. 现在把圆柱面的中心设为照相机坐标系的原点, 并以数码相机的像素焦距 f 作为圆柱面的半径, 经过一系列推导可得圆柱面方程的三维表示式(2).

4.2 将三维圆柱面方程转变为二维图像坐标的变换公式

(u, v, w) 是像素点 $p(x, y)$ 在圆柱面上的投影点 p' 的参数坐标, 把所有这样的投影点组合起来便得到全景图像. 但是, 参数坐标是三维的, 需要把它们转变为二维的图像坐标才便于储存. 即需要将圆柱面展成平面. 这里选用公式(3)把三维的参数坐标转化为二维的图像坐标.

$$\begin{cases} u = \frac{f(x - W/2)}{\sqrt{(x - W/2)^2 + f^2}} \\ v = \frac{f(y - H/2)}{\sqrt{(x - W/2)^2 + f^2}} \\ w = \frac{f^2}{\sqrt{(x - W/2)^2 + f^2}} \end{cases} \quad (2)$$

$$\begin{cases} x' = f \cdot \arctan(u/w) + f\theta \\ y' = v + H/2 \end{cases} \quad (3)$$

联立(2)和(3)可得:

$$\begin{cases} x' = f \cdot \arctan \left[\frac{x - W/2}{f} \right] + f \cdot \arctan \left[W/(2f) \right] \\ y' = \frac{f \cdot (y - H/2)}{\sqrt{(x - W/2)^2 + f^2}} + H/2 \end{cases} \quad (4)$$

式(4)是将实景图像 I 上的任意一个像素点 $p(x, y)$ 柱面正投影到柱面全景图像上一个像素点 $p'(x', y')$ 上的投影公式. 该投影算法具有使景物在垂直方向上不发生形变的性质.

5 对圆柱面正投影算法的讨论

可以发现: 实景图像在投影后得到的柱面投影图像, 在宽度和高度方向尺寸都发生了变化, 如图 6 和图 7(a)、(b)所示. 下面我们定量讨论宽度和高度的变化, 讨论的目的一方面进一步了解投影的过程, 另一方面为全景图的拼接作好理论上的准备.

5.1 宽度的变化

由图 7(b)可知, 柱面投影图像(展开图)的宽度为 $w' = 2f\theta$.

5.2 高度的变化

已知: 实景图像 I 的宽度 W , 高度 H , I 上任意一像素点 $p(x, y)$, 则 $x \in [0, W]$, $y \in [0, H]$.

由式(4)可知, y' 是关于 x, y 的函数. 最小高度发生在 $x=0$ 或 $x=W$ 时; 最大高度发生在 $x=W/2$ 时(如图 6 和图 7(a)).

(1) 当 $x=0, y=0$ 时, 并将 $f=W/(2 \tan \theta)$ 代入式(4), 得: $y'_1 = H(1 - \cos \theta)/2$;

当 $x=0, y=H$ 时, $y'_2 = H(1 + \cos \theta)/2$;

故 $H_{\min} = y'_2 - y'_1 = H \cos \theta$. 当 $x=W$ 时可以经过上边相似的推导得到相同的结果.

(2) 当 $x=W/2, y_1=H$ 时, $y'_1=H, H_{\min}=H$.

综上所述, 可见投影后, 宽度变小, 高度由 $H_{\min}$ 逐渐变大到 H 再到 $H_{\min}$.

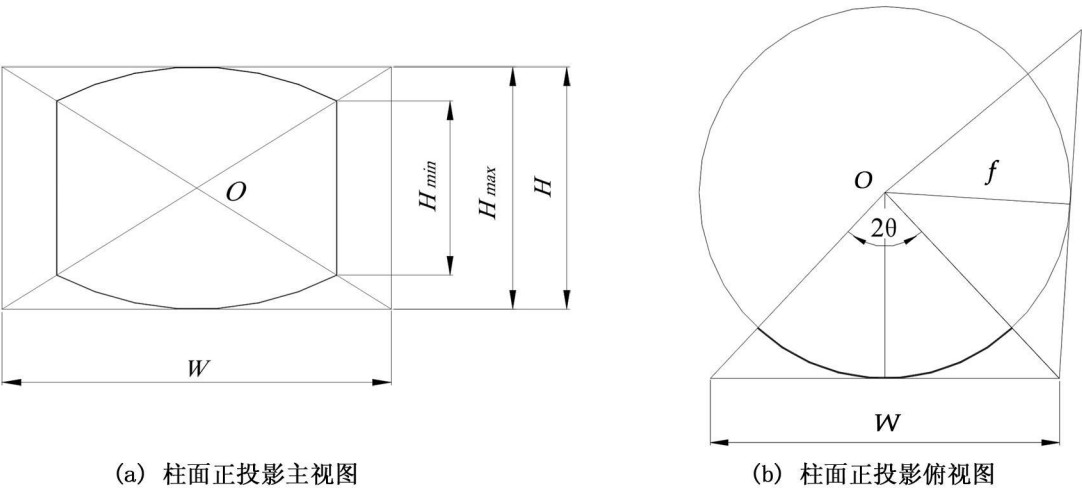


图 7 柱面正投影的主视图和俯视图

6 算法实现

实现上述算法,采用的图像是用普通数码相机和三脚架在固定地点拍摄的图片,经过圆柱面正投影程序处理前后的图片分别如图 8(a)、8(b)所示.

这里用 C++ 语言在 Visual C++ 开发环境中



图 8 圆柱面正投影结果

7 结语

虚拟全景空间技术是目前比较热门的技术之一,对许多相关知识点的理解是比较抽象的,在它实现过程中涉及许多较复杂的算法.笔者结合自己的实践和理解将全景图生成中圆柱面正投影所涉及的算法进行分析、讨论,最后对算法及其讨论的正确性进行了验证.

参考文献:

[1] 周春艳,邹峥嵘.基于全景图的虚拟校园[J].电脑与信

息技术,2002,(2):34—36.
[2] S E Chen.QuickTime VR——An image-based approach to virtual environment navigation [A]. Computer Graphics Proceedings, Annual Conference Series [C]. Los Angeles: ACM SIGGRAPH 1995. 29—38
[3] L McMilliams, G. Bishop Plenoptic modeling: An image-based rendering system [A]. Computer Graphics Proceedings, Annual Conference Series [C]. Los Angeles: ACM SIGGRAPH 1995, 39—46.
[4] R. Szeliski. Video mosaics for virtual environments [J]. IEEE Computer Graphics and Applications. 1996, (3): 22—30.
[5] 孙家广. 计算机图形学 [M]. 北京:清华大学出版社, 2000.

[6] 郑南宁. 计算机视觉与模式识别[M]. 北京: 国防工业出版社, 1998.

ID=28&ID=63&page=3.html, 2007-07-13/2008-05-05.

[7] 天地悠悠. UU 谈摄影(3)-摄影的原理与相机的构造[EB/OL], [http://www.oollt.com/photo/dispbbs.asp? board-](http://www.oollt.com/photo/dispbbs.asp?board-)

[8] 张茂军. 虚拟现实系统[M]. 北京: 科学出版社, 2005.

An Algorithm Study on the Cylinder-surface Direct Projection
in the Process of Creating Panoramic Images

YANG Gang¹, SHI Yun-fei²

(1.School of Computer Science, Shanxi University of Technology, Hanzhong 723003, China; 2.School of Mechanical and Electrical Engineering, Northwestern Polytechnical University, Xi 'an 710072, China)

Abstract: In the course of creating the virtual space based on IBMR, it is very important to create panoramic images. After getting a group of correlative photos from the ordinary numeral camera, we need to project, join and process them in order to get a panoramic image. This paper mainly analyzes, discusses, concludes, and summarizes the algorithm of the cylinder-surface direct projection in the process of creating panoramic images. And many new ideas are presented. Finally, validity of the algorithm and discussion is proved.

Key words: panoramic images; imaging model; the cylindrical surface direct projection; algorithm
(责任编辑: 周尚超)