

OpenGL 模拟摄影测量方法研究

苏国中 郑顺义 张剑清 张祖勋

(武汉大学遥感信息工程学院, 武汉 430079)

摘要 通过分析 OpenGL 透视成像过程, 建立了 OpenGL 成像矩阵与摄影测量中内外方位元素的关系。实现了静态单幅影像、立体像对的模拟以及运动目标轨迹和姿态模拟。利用摄影测量空间后方交会方法对模拟影像进行处理, 验证了模拟结果的正确性。

关键词 OpenGL 成像矩阵 摄影测量内外方位元素 相机姿态模拟 空间后交

中图分类号: P208 P273 文献标识码: A 文章编号: 1006-8961(2006)04-0540-05

How to Relate the OpenGL Imaging Process with Exterior and Interior Parameters of Photogrammetry

SU Guo-zhong, ZHENG Shun-yi, ZHANG Jian-qing, ZHANG Zu-xun

(School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079)

Abstract Based on deep analysis of the imaging mechanism of OpenGL in the sight of projective geometry, this paper have constructed the relationship between photogrammetry exterior and interior parameters and OpenGL projective matrix and proved their internal consistency. Then the transition equations of two imaging processes are derived. With above relationship and derivation, experiments have been conducted to simulate single static image, stereo image pair and motion orbit and pose in OpenGL. Experimental results prove the correctness of the theory proposed in this paper.

Keywords OpenGL, imaging matrix, exterior and interior parameters of photogrammetry, simulate camera's pose, space resection

1 引言

摄影测量主要研究自然景物影像的获取、几何处理、被摄物体大小形状测量及测量精度评定^[1]。OpenGL 主要提供 3 维模型构造、3 维场景渲染、物体表面纹理贴面、特效和相关 2 维及 3 维图形操作功能^[2]。OpenGL 可以将一个 3 维模型置于一个场景中, 加上光照和纹理用透视投影方法生成一幅逼真的 2 维影像^[3]。以影像为主要研究对象的当代数字摄影测量与计算机图形学有着天然的联系, 摄影测量是不规则物体的高精度 3 维建模的一个重要技术途径之一; 如果 OpenGL 能模拟出已知摄影参数的影像, 这可为探索新的摄影测量数据处理方法、摄影测量精度评定、摄影测量教学、空间摄影模拟创造

有利条件。众所周知 OpenGL 这一功能强大的计算机图形学程序库, 其成像过程中涉及到的参数和矩阵与摄影测量学中的内外方位元素没有直接的联系^[4], 这就为 OpenGL 在摄影测量中的应用带来很大不便。为此通过对 OpenGL 的透视投影成像过程和机理的深入剖析, 总结了二者之间的本质关系, 推导出两种成像过程的转换公式, 并就摄影测量中典型的空间单片后方交会、立体像对生成、运动目标轨迹及姿态模拟等问题进行了分析和试验。

摄影测量与计算机视觉以及计算机图形学等学科的研究内容在某些方面有很大的相似, 但由于发展背景不同, 研究方法上存在很大差异, 以致于很难交流和沟通, 本文深入探索了摄影测量与计算机视觉以及计算机图形学^[5]之间的区别及其内在联系, 对于学科之间的交流和沟通具有积极的启发作用。

收稿日期: 2004-08-05; 改回日期: 2005-07-04

第一作者简介: 苏国中(1964 ~), 男, 高级工程师。2005 年于武汉大学遥感信息工程学院获摄影测量与遥感工学博士学位。现从事遥感、摄影测量与计算机视觉、地理信息系统等方面的研究。E-mail: sgz_gh@126.com

2 OpenGL 成像过程与摄影测量方位元素分析

OpenGL 透视成像的几何过程如图 1 所示^[2]。

一个 3 维物体模型上的点经过模型矩阵 M 的旋转平移及缩放变换, 进入像方空间坐标系中, 再经过投影矩阵 P 的透视投影变换, 透视出仿射变换 F 后获得屏幕影像坐标。其成像公式为

$$[x \ y \ 1]^T = F \cdot P \cdot M \cdot [X \ Y \ Z \ 1]^T \quad (1)$$

如图 2 所示, 在 OpenGL 中可以用函数

$glFrustum(x_l, x_r, y_b, y_t, z_{map}, z_{obj})$ ^[6,7] 确定透视投影视锥体, 其中, $(x_l, y_b), (x_r, y_t)$ 分别为成像平面左下和右上角坐标, z_{map} 和 z_{obj} 是视点到像平面和物平面的距离。由该函数生成的相应投影矩阵为

$$P = \begin{bmatrix} \frac{2z_{map}}{x_r - x_l} & 0 & \frac{x_r + x_l}{x_r - x_l} & 0 \\ 0 & \frac{2z_{map}}{y_t - y_b} & \frac{y_t + y_b}{y_t - y_b} & 0 \\ 0 & 0 & -\frac{z_{obj} + z_{map}}{z_{obj} - z_{map}} & -\frac{2z_{obj}z_{map}}{z_{obj} - z_{map}} \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (2)$$

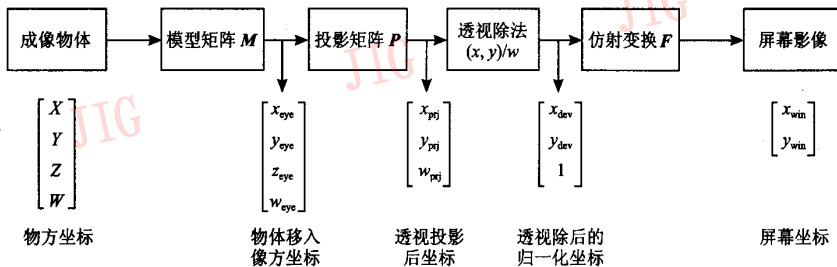


图 1 OpenGL 成像过程

Fig. 1 OpenGL imaging process

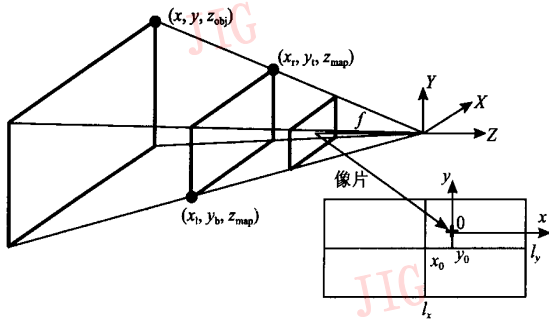


图 2 OpenGL 成像原理图

Fig. 2 OpenGL imaging theory

假设摄影底片宽为 (l_x, l_y) , 摄像机的焦距为 f , 像主点坐标为 (x_0, y_0) , 按如图 2 所示, 透视投影矩阵 P 中的参数可以重新确定为

$$\begin{cases} x_l = -\frac{z_{map}}{z_{obj}} \left(\frac{1}{2} l_x + x_0 \right) \\ x_r = \frac{z_{map}}{z_{obj}} \left(\frac{1}{2} l_x - x_0 \right) \\ y_b = -\frac{z_{map}}{z_{obj}} \left(\frac{1}{2} l_y + y_0 \right) \\ y_t = \frac{z_{map}}{z_{obj}} \left(\frac{1}{2} l_y - y_0 \right) \end{cases} \quad (3)$$

进一步设物体从原物方坐标空间系变换到像方空间坐标系的模型变换矩阵 M 为

$$M = \begin{bmatrix} a_1 & b_1 & c_1 & -a_1 X_s - b_1 Y_s - c_1 Z_s \\ a_2 & b_2 & c_2 & -a_2 X_s - b_2 Y_s - c_2 Z_s \\ a_3 & b_3 & c_3 & -a_3 X_s - b_3 Y_s - c_3 Z_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

其中, $\{a_i, b_i\}$ 为物体旋转矩阵元素, (X_s, Y_s, Z_s) 为物体中心相对于原始位置移动的距离。在不失一般性的情况下, 像主点在屏幕中心的仿射变换矩阵 F 可设置为

$$F = \begin{bmatrix} \frac{l_x}{2} & 0 & 0 & 0 \\ 0 & \frac{l_y}{2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

将式(2) ~ (5)代入式(1)得:

$$\begin{cases} x - x_0 = -f \frac{a_1(X - X_s) + b_1(Y - Y_s) + c_1(Z - Z_s)}{a_3(X - X_s) + b_3(Y - Y_s) + c_3(Z - Z_s)} \\ y - y_0 = -f \frac{a_2(X - X_s) + b_2(Y - Y_s) + c_2(Z - Z_s)}{a_3(X - X_s) + b_3(Y - Y_s) + c_3(Z - Z_s)} \end{cases} \quad (6)$$

式(6)与摄影测量共线方程^[11]完全一致, 说明

只要按式(2)、(3)、(5)将摄影测量的成像参数设置到 OpenGL 中的投影矩阵、视口矩阵和模型矩阵中,OpenGL 与摄影测量的成像过程就完全一致。否则即使获得视口矩阵、投影矩阵、模型矩阵和相应的影像,也不能确定 OpenGL 中虚拟相机的内外方位元素。

3 OpenGL 模拟摄影测量方法

3.1 摄影测量空间后方交会模拟

在计算机中构造一个 3 维实体,对 OpenGL 虚拟相机设置相应的内外方位元素和相应的视口矩阵,只要物体在虚拟相机的视场范围内,就可以获得一幅屏幕影像。

由于相应的模型坐标是已知的,在该影像上量测 $n(n > 3)$ 个影像坐标,就可以用空间后方交会的方法求出相机的内外方位元素。如果计算数据与预先设置的数据一致,可以验证以下两条推论。

(1) 如果空间后方交会程序用其他方法证明是正确的,那么就可以证明在 OpenGL 中设置摄影测量内外方位元素的方法也是正确的;

(2) 如果在 OpenGL 中设置摄影测量方位元素的方法是正确的,那么就可以证明编写的空间后方交会程序是正确的。依据前面的分析在 OpenGL 中设置一个虚拟相机和相应的方位元素是可能的。

设虚拟相机的角元素为 $(\varphi, \omega, \kappa)$, 摄影中心位置为 (X_c, Y_c, Z_c) , 焦距为 f , 其像幅大小为 (l_x, l_y) , 则可以按下列步骤设置虚拟相机的摄影测量方位元素:

(1) 用 $glMatrixMode(GL_PROJECTION)$ 函数设置 OpenGL 当前矩阵为投影矩阵;

(2) 用 $glLoadIdentity()$ 函数设置当前投影矩阵为单位阵;

(3) 依据屏幕大小按式(2)、(3)计算透视投影矩阵 P ;

(4) 用 $glMultMatrixd(*P)$ 函数将矩阵 P 加入 OpenGL 当前投影矩阵中;

(5) 调用函数 $glRotated(\varphi, 0, 1, 0)$ 、 $glRotated(\omega, 1, 0, 0)$ 、 $glRotated(\kappa, 0, 0, 1)$, 设置摄影测量角元素;

(6) 调用函数 $glTranslated(-X_s, -Y_s, -Z_s)$, 设置摄影测量线元素;

(7) 调用 $glMatrixMode(GL_MODELVIEW)$, 按需

求设置模型变换坐标 M , 其中, $GL_MODELVIEW$ 为 OpenGL 规定的设置模型矩阵常数。

(8) 依据屏幕大小调用函数 $glViewport(0, 0, scrX, scrY)$ 设置视口矩阵 F , 其中 $scrX, scrY$ 为屏幕宽高。

现以一个 10×10 的立方块为研究对象, 移动水平放置的立方块中心点到 OpenGL 全局坐标系的 $(0, 0, -15)$ 。设 OpenGL 虚拟相机的摄站点坐标为 $(20, 0, 0)$, 焦距 $f = 5.0$, 像主点坐标为 $(0.00, 0.00)$, 相机姿态: $\varphi = 0.0^\circ$ 、 $\omega = 0.0^\circ$ 、 $\kappa = -10^\circ$ 。这时可以获得一幅如图 3 所示的模拟影像。量测影像上 8 个角点的坐标, 用空间后方交会解算虚拟相机参数, 从计算结果表 1 中可以看出计算值与原虚拟相机的设置值是一致的。实际测试过程中, 使用的摄影测量空间后方交会程序是经过其他数据检验的一个比较可靠的程序。试验结果说明, 用 OpenGL 完全可以模拟出航空摄影影像的各种几何参数的。

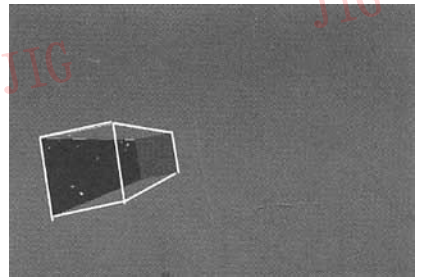


图 3 OpenGL 产生的模拟图像

Fig. 3 The simulated image produced by OpenGL

3.2 摄影测量中立体像对的模拟

OpenGL 本身支持带双缓冲区的显示卡, 只要将具有左右视差的影像置入两个缓冲区中就可以使用偏振眼镜观察立体。现在要使用 OpenGL 模拟出模型物体的一个立体像对, 利用这个像对可以进行相对定向、绝对定向、核线排队、相关匹配等程序调试工作。只要依据设定的相片重叠度, 定义虚拟相机摄站点位置和参数就可以获得立体像对, 图 4 是 OpenGL 模拟的一个飞机模型的立体像对。

3.3 摄影测量中运动轨迹及物体姿态模拟

用 OpenGL 模拟运动轨迹比较简单, 依据物体在某一时刻 t 的位置坐标 (x_t, y_t, z_t) , 直接调用 OpenGL 库函数 $glTranslated(x_t, y_t, z_t)$ 即可设置其模型矩阵, 从而实现物体的运动轨迹模拟。然而, 为了保证被模拟目标(例如飞机)自身方向(机头所指方

表 1 摄影测量空间后方交会结果
Tab.1 The result of space resection

序号	物体原始坐标			影像量测坐标		方位元素	计算值	原设置值
	X	Y	Z	x	y	f		
1	-5.00	-5.00	-20.0	230	280	x_0	5.004 038 8	5.00
2	5.00	-5.00	-20.0	291	269	y_0	0.000 000 00	0.00
3	5.00	5.00	-20.0	280	208	φ	0.000 003 61°	0.00°
4	-5.00	5.00	-20.0	219	219	ω	-0.000 001 79°	0.00°
5	-5.00	-5.00	-10.0	81	338	κ	-9.936 987 56°	-10.0°
6	5.00	-5.00	-10.0	204	316	X_s	20.009 091 50	20.00
7	5.00	5.00	-10.0	182	193	Y_s	-0.000 127 84	0.00
8	-5.00	5.00	-10.0	59	215	Z_s	0.000 760 32	0.00

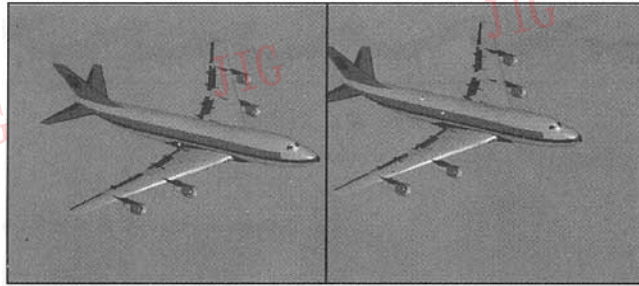


图 4 OpenGL 模拟立体像对

Fig.4 The simulated stereo image pair produced by OpenGL

向)与前进方向基本一致,需要解决矢量转角分解这样一个问题。在摄影测量中使用的都是独立转角系统,即绕三坐标轴转角 φ, ω, κ 是相互无关的量;物体每向前移动一个位置相对于前进方向的矢量 V 旋转一个角度 α 。如何将转角 α 分解为 φ, ω, κ 再设置到 OpenGL 的模型矩阵中达到模拟运动物体姿态的目的,这是运动物体轨迹和姿态模拟的关键问题之一。

如图 5, 设矢量 $V_1(x_1, y_1, z_1)$ 与 XZ 平面的夹角为 α_1 , 与 XY 平面的夹角为 β_1 ; 矢量 $V_2(x_2, y_2, z_2)$ 是

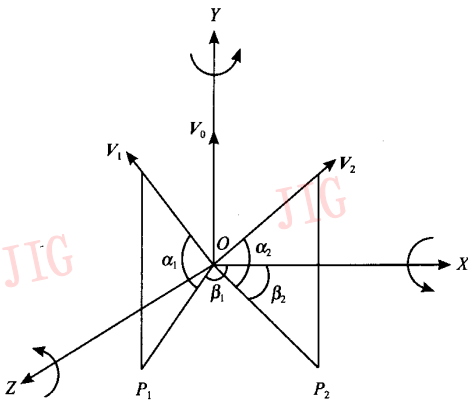


图 5 在坐标系中矢量旋转示意图

Fig.5 The sketch map of vector rotation in coordinate system

矢量 V_1 旋转的结果,它与 XZ, XY 二平面的夹角分别为 α_2, β_2 ;再设矢量 $V_0(0, s, 0)$ 是矢量 V_1 旋转到与 Y 轴重合的结果。于是有:

$$V_1 = V_0 \cdot R_{\kappa(\alpha_1-90^\circ)} \cdot R_{\varphi(-\beta_1)} \quad (7)$$

$$V_2 = V_0 \cdot R_{\kappa(\alpha_2-90^\circ)} \cdot R_{\varphi(-\beta_2)} \quad (8)$$

其中, $R_\varphi, R_\omega, R_\kappa$ 为相应坐标轴旋转矩阵。由式(7)、(8)可得:

$$\begin{aligned} V_2 &= V_1 (R_{\kappa(\alpha_1-90^\circ)} R_{\varphi(-\beta_1)})^{-1} R_{\kappa(\alpha_2-90^\circ)} R_{\varphi(-\beta_2)} \\ &= V_1 (R_{\varphi(-\beta_1)})^T (R_{\kappa(\alpha_1-90^\circ)})^T R_{\kappa(\alpha_2-90^\circ)} R_{\varphi(-\beta_2)} \\ &= V_1 \begin{bmatrix} \cos(\beta_1) & 0 & -\sin(\beta_1) \\ 0 & 1 & 0 \\ \sin(\beta_1) & 0 & \cos(\beta_1) \end{bmatrix} \begin{bmatrix} \sin(\alpha_1) & \cos(\alpha_1) & 0 \\ -\cos(\alpha_1) & \sin(\alpha_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= V_1 \begin{bmatrix} \sin(\alpha_2) & -\cos(\alpha_2) & 0 \\ \cos(\alpha_2) & \sin(\alpha_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\beta_2) & 0 & \sin(\beta_2) \\ 0 & 1 & 0 \\ -\sin(\beta_2) & 0 & \cos(\beta_2) \end{bmatrix} \quad (9) \end{aligned}$$

如果设定

$$\begin{cases} s = x_1^2 + y_1^2 + z_1^2 = x_2^2 + y_2^2 + z_2^2 \\ d_1 = \sqrt{x_1^2 + y_1^2} \\ d_2 = \sqrt{x_2^2 + y_2^2} \\ d = d_1 \times d_2 \end{cases} \quad (10)$$

及

$$(11) \quad \begin{cases} \sin(\alpha_1) = \frac{y_1}{\sqrt{s}} & \cos(\alpha_1) = \frac{d_1}{\sqrt{s}} \\ \sin(\alpha_2) = \frac{y_2}{\sqrt{s}} & \cos(\alpha_2) = \frac{d_2}{\sqrt{s}} \\ \sin(\beta_1) = \frac{z_1}{d_1} & \cos(\beta_1) = \frac{z_2}{d_2} \\ \sin(\beta_2) = \frac{x_1}{\sqrt{s}} & \cos(\beta_2) = \frac{x_2}{\sqrt{s}} \end{cases}$$

考虑到计算角度前将矢量单位化,则可以推导出:

$$(12) \quad \begin{cases} \varphi = \arctan 2(z_1 x_2 s - x_1 y_1 y_2 z_2 - x_1 z_2 d, \\ \quad y_1 z_1 y_2 z_2 + z_1 z_2 d + x_1 x_2 r) \\ \omega = \arcsin(-d_1 y_2 z_2 / s / d_2 + y_1 z_2 / s) \\ \kappa = \arctan 2(x_2 y_2 d_1 - x_2 y_1 d_2, d d_2 + y_1 y_2 d_2) \end{cases}$$

为此,只要为模拟的运动物体设计一条平稳的飞行路线,按式(12)计算每一位置点的姿态角,再加上一些随机噪声,在 OpenGL 程序中设置时间事件的响应就可以完成运动物体轨迹和姿态的模拟,图 6 是一个已知真实轨迹的姿态模拟序列每隔 10 幅取一幅的结果。



图 6 模拟飞行序列

Fig. 6 The simulated image sequence

4 结 论

通过本文分析可知,OpenGL 的成像过程完全可以用摄影测量的方法进行定量控制;用 OpenGL 可以模拟各种摄影测量过程,把 OpenGL 强大的虚拟现实能力和摄影测量精确的空间定位和测量能力结合起来,可以在单片摄影测量、相机参数设置、摄影测量软件算法研究及质量评估、移动目标识别和姿态测量等方面发挥作用。

参考文献 (References)

- 1 Zhang Zu-xun, Zhang Jian-qing. Digital photogrammetry [M]. Wuhan: Publishing house of Wuhan University, 1996. [张祖勋,张剑清著. 数字摄影测量学[M]. 武汉:武汉测绘科技大学出版社,1996.]
- 2 Richard S Wright, Jr Michael Sweet. OpenGL Bible [M]. Postal Publishing House of People, 2001. [Richard S Wright, Jr Michael

Sweet 著. OpenGL 超级宝典 [M]. 北京:人民邮电出版社, 2001.]

- 3 Sun Jia-guang, Yang Chang-gui. Computer Graphics [M]. Beijing: Tsinghua University Publishing House, 1997. [孙家广,杨长贵著. 计算机图形学[M]. 北京:清华大学出版社,1997.]
- 4 Wang Zhi-zhuo. Principle of Photogrammetry [M]. Beijing: Publishing House of Surveying and Mapping, 1980. [王之卓著. 摄影测量原理[M]. 北京:测绘出版社,1980.]
- 5 Rogers D F. Algorithm Foundation of Computer Graphics [M]. Translated by Liang You-dong, Shi Jiao-ying, Peng Qun-sheng. Beijing: Tsinghua University Publishing House, 1986. [Rogers D F 著. 计算机图形学的算法基础[M]. 梁友栋,石教英,彭群生译,北京:清华大学出版社,1986.]
- 6 Qiao Ling, Fei Guang-zheng. OpenGL Program Design [M]. Beijing: Tsinghua University Publishing House, 2000. [乔林,费广正著. OpenGL 程序设计[M]. 北京:清华大学出版社,2000.]
- 7 Li Ying, Xue Hai-bin. OpenGL Function and Sample Analysis Manual [M]. Beijing: National Defence Industry Publishing House, 2002. [李颖,薛海斌著. OpenGL 函数与范例分析手册[M]. 北京:国防工业出版社,2002.]