

文章编号 1004-924X(2004)02-0210-05

# 人工立体视觉仿真系统的建立和应用

吴家麒, 袁政鹏, 杨东英  
(上海大学 机自学院, 上海 200072)

**摘要:** 为了对人工立体视觉中各个环节进行验证, 提出了开发人工立体视觉仿真系统的思路。以开放图形库 OpenGL 为工具, 架设虚拟相机, 设置虚拟物体, 拍摄虚拟图像, 在虚拟的图像上采集数据, 并根据数据的相关性进行虚拟物体重建, 最后将重建物体与原始物体进行比较, 验证整个系统或某测试模块的准确性。应用结果表明, 该仿真平台可以用于立体视觉研究中的测试和模拟。

**关键词:** 人工视觉; 立体视觉; 视觉仿真; 三维重建

**中图分类号:** TP387 **文献标识码:** A

## Establishment and application of artificial stereovision simulation system

WU Jia-qi, YUAN Zheng-peng, YANG Dong-ying

(School of Electromechanical Engineering & Automation,  
Shanghai University, Shanghai 200072, China)

**Abstract:** In order to validate the artificial stereovision modules, an artificial stereovision simulation system is established using the open graphics library OpenGL as the tool, setting up virtual cameras, placing virtual objects, taking virtual images, collecting information from the images, and reconstructing the virtual objects according to the interrelation among data collected, and finally comparing the reconstructed object with the original object, thereby checking the whole system or a particular test module for accuracy. Application results show that the simulation system can be used for stereovision research testing and simulation.

**Key words:** artificial vision; stereoscopic vision; vision simulation; 3D reconstruction

## 1 引言

人工视觉在人工智能和机器人视觉的研究中一直占有重要的地位<sup>[1]</sup>。人工视觉的实现为智能设备和机器人提供了对外在世界的一个重要感知手段,是实现智能判别和自主行动的必要基础。在人工视觉中,人工立体视觉意义更大且难度也更高。它在提供有关物体形状的平面信息的同

时,提供了深度信息,使得所获得的信息更全面、更正确。

一般的人工立体视觉系统需要设置2台或2台以上相机(或摄像机),在进行立体标定以后,可以在同一时刻摄取同一物体的不同角度的图像,然后经过一系列特征提取与特征匹配后,根据同一对象在不同图像中的相关信息,计算重建物体的三维信息<sup>[2-3]</sup>。由此可见,人工立体视觉涉及的环节相当多,任何一个环节的出错都会直接导致

最终物体三维信息的错误,而从最终错误中去确定出错环节并不是一件容易的工作。正因为如此,在人工立体视觉的研究中,对单个环节进行验证就显得十分必要。如果有一个仿真系统,可以保证在其他环节准确无误的情况下,对某一个模块的准确性进行测试,这将为人工立体视觉的研究提供很大的便利。

开发本仿真系统的目的就是为采用人工立体视觉进行目标跟踪与三维重建的研究提供开发工具,以便可以用仿真系统对实际系统的单个模块进行测试,同时可以在整个硬件系统还未搭建完成之前就对某些模块进行调试。

## 2 人工立体视觉仿真系统的思路

根据人工立体视觉仿真系统的开发目的,仿真系统的工作步骤应该和实际系统一致。所不同的是仿真系统中各个部分都是虚拟的:架设虚拟相机,设置虚拟物体,拍摄虚拟图像,在虚拟的图像上采集数据,再根据数据的相关性进行虚拟物体重建,然后可将重建物体与原始物体进行比较,验证系统或某测试模块的准确性。

这里的基础工作之一就是要建立一套虚拟的成像系统来设置虚拟的相机。实际上,现在有一些图形工具可以提供图形的透视投影功能,如 OpenGL, Direct X 等,这种透视投影功能与相机或摄像机的成像功能有类似之处,这就为用这些工具实现虚拟的成像系统提供了可能。但是,作为虚拟的摄像,必须进行一些专门的操作,相机的定位、定向、取景,物体的变换等,这些虚拟的操作都必须找出适当的方法予以实现。本仿真系统选择了 OpenGL 作为开发工具来实现上述虚拟的操作。

作为一个仿真系统,应该包括人工立体视觉的基本功能,成为一个完整的虚拟系统,从而正确运行。因为要在此系统中检测实际系统的单个模块,因此要求将整个仿真系统的各个基本功能做成可以更换的独立模块,以便在测试中方便地用实际系统的单个模块取代相应的虚拟模块,然后在运行中检验实际模块的正确性。

## 3 虚拟相机的设置

在人工立体视觉仿真系统中,最基本的要求是设置 2 台或 2 台以上的虚拟相机。虚拟相机的设置包括定位、定向和取景。

### 3.1 虚拟相机的定位

为了从不同角度拍摄到同一物体的不同图像,多台虚拟相机应该置于不同的位置,采取不同的视角。OpenGL 中所提供的透视投影矢量的定位与转向功能与相机的位置、方向的设置要求是类似的,因此可以用以下的方法来实现虚拟相机的定位<sup>[4-5]</sup>。

首先需要指定虚拟相机为变换的对象,这可以通过函数 `glMatrixMode(mode)` 来实现。参数 `mode` 取 `GL_PROJECTION`, 则以后的变换都是对透视投影矢量进行的;如果参数 `mode` 取 `GL_MODELVIEW`, 则以后的变换都是对被观测物体进行的;如果将参数 `mode` 取为 `GL_TEXTURE`, 则以后的变换都是以纹理为变换对象。当然在进行虚拟相机的定位时,参数 `mode` 应该取 `GL_PROJECTION`。

在进行具体的定位定向之前,需要指出的是在 OpenGL 中,相机位置所参照的坐标系遵循的是左手法则,相机坐标系的  $z$  坐标的正方向指向屏幕的里面。

相机位置的初始化可以通过函数 `glLoadIdentity` 来实现。从数学上讲,这个函数加载的是一个  $4 \times 4$  阶的单位阵,而从物理意义上讲,调用这个函数就是将相机位置的 3 个位移、相机方向的 3 个转角位置为 0。

相机位置的移动由函数 `glTranslate(dx, dy, dz)` 来模拟,  $dx$ 、 $dy$ 、 $dz$  分别为 3 个坐标轴方向上的移动量。按照上述坐标系的规定,  $dx$  以向左为正,  $dy$  以向下为正,  $dz$  以向里为正;而相机方向的转动由函数 `glRotate(cr, cx, cy, cz)` 来模拟。参数  $cr$  表示转动的度数,  $cx = 1$  表示绕  $x$  轴转  $cr$  角,  $cx = 0$  表示绕  $x$  轴没有转动,  $cy = 1$  或  $0$ ,  $cz = 1$  或  $0$  的含义与  $cx$  的含义相同。

### 3.2 虚拟相机的视野与焦距

真实相机的视野与镜头焦距有关,焦距的长短决定了视野的大小,焦距越短则视野越大,焦距越长则视野越小。虚拟相机当然也应该具有这一功能。虚拟相机的这一功能可用 OpenGL 中视景

体 (Frustum) 的概念来模拟。

视景物是一个以透视投影轴  $Z_c$  为中心轴的 4 棱台。4 条棱线均通过透视投影点,4 棱台在投影轴  $Z_c$  方向上的尺寸由近截面与远截面来决定,参见图 1。OpenGL 中引入视景物 Frustum 概念,实际上指定了被透视投影的物体所在的区域,凡是落在视景物之内的物体,均将在透视投影视图上予以显示;而落在视景物之外的物体均不予显示。这个概念与相机视野的功能是一致的。

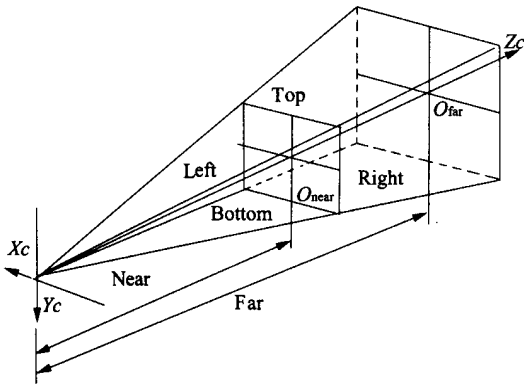


图 1 视景物示意图

Fig. 1 Sketch map of the frustum

视景物可以用函数 glFrustum (Left, Right, Top, Bottom, Near, Far) 来设置,函数中共有 6 个参数, left 与 right 决定了视景物在  $x$  方向上的大小, Top 与 Bottom 决定了视景物在  $y$  方向上的大小,参数 near 确定了近截面的位置,参数 far 确定了远截面的位置。 $O_{Near}$ 、 $O_{Far}$  分别为轴  $Z_c$  与近截面、远截面的交点,参见图 1。

需要指出的是,透视投影面实际就是 4 棱台的近截面,因此可将参数 Near 看作虚拟相机的焦距  $f$ 。

### 3.3 虚拟底片与图象坐标

由于在人工立体视觉仿真系统中,需要采集物体在虚拟底片中的坐标数据,作为进行虚拟物体三维重建的依据,因此坐标数据的正确性至关重要。为此,需要正确地确定虚拟底片的尺寸并建立图像坐标。

如上所述物体的透视投影面就是视景物的近截面。透视投影中心  $O_{Near}$  (即透视投影轴  $Z_c$  与近截面的交点) 到近截面左边的距离为参数 left, 透视投影中心到近截面上边的距离为参数 right。以透视投影中心为原点可以建立投影坐标系,其

$x$  轴过原点指向右,其  $y$  轴过原点向上,如图 2 所示。任意一个透视投影点在投影坐标系中的位置可由坐标  $x$ 、 $y$  确定。

实际上,近截面上的图像是显示在视区 (ViewPort) 上的。建立在视区上的为图像坐标系  $uv$ , 图像坐标以左上角为原点,其  $u$  轴过原点向右,其  $v$  轴过原点向下,如图 2 所示。

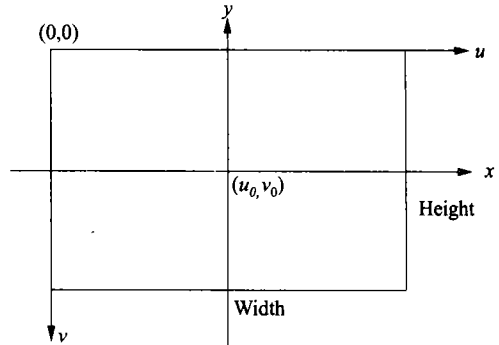


图 2 投影坐标与图像坐标

Fig. 2 Projection and image coordinates

要确定图像坐标,首先需要确定的是近截面的大小与视区的大小关系。近截面的大小由参数 Left, Right, Top, Bottom 确定;视区以像素 Pixel 为单位,其大小由宽 (Width)、高 (Height) 确定。

其次是投影坐标系的原点在图像坐标系的位置。如前所述,该原点离近截面左边为 left,离近截面上边为 Right,而图像坐标系的  $u$  轴与投影坐标系  $x$  轴同向, $v$  轴与投影坐标系  $y$  轴反向,因而可以得出投影坐标系的原点在图像坐标系的位置  $u_0$ 、 $v_0$ 。据此可以得出任一投影点在图像坐标系的位置  $u$ 、 $v$ 。

需要指明,在 Windows 环境下编程,无论是文字,还是图形,都是写在或画在视窗中的,为了在视窗中测量虚拟底片上的图像坐标,必须使视窗的大小与视区的大小相配。

## 4 虚拟物体的变换

说到物体的变换,必须先指明变换的参照坐标系。OpenGL 中规定物体的变换是相对世界坐标系的。与相机坐标系相反,世界坐标系遵循的是右手法则, $x$  的正方向朝右, $y$  的正方向朝上, $z$  的正方向指向屏幕外面。

已经定义好的物体是可以在世界坐标系中移

动和转动。当然在变换之前,必须用函数  $glMatrixMode(mode)$  规定,以后的变换将以物体为变换对象,为此参数  $mode$  应该选  $GL\_MODEVIEW$ 。

在确定参数  $mode$  之后,以后物体的变换操作就与相机的变换操作相同。物体变换矩阵的初始化由函数  $glLoadIdentity$  完成,物体的移动变换由函数  $glTranslate(dx, dy, dz)$  实现,物体的转动则由函数  $glRotate(or, ox, oy, oz)$  进行。各项参数的含义与相机的变换参数的含义一样,只不过此时的变换对象不再是相机,而是虚拟物体。此外需要注意,物体变换所参照的世界坐标系是按照右手法则定义的。

## 5 虚拟系统的标定

在使用实际的视觉系统前必须进行标定,也就是说,要建立图像坐标系和实际坐标系之间的关系:

$$zc[u, v, 1]^T = M[xw, yw, zw, 1]^T, \quad (1)$$

这一工作是经过对一系列已知点(即已知其世界坐标  $xw, yw, zw$ ) 的摄像,取得图像坐标  $u, v$ , 经过运算解算得到标定矩阵  $M$  的各个元素。

但是在虚拟的视觉系统中,这一工作必须根据  $M$  矩阵的物理意义、几何意义来确定。可以先求得图像坐标与相机坐标的关系矩阵  $M_1$ , 再求得相机坐标与世界坐标的关系矩阵  $M_2$ , 然后可以综合出世界坐标与图像坐标的关系矩阵  $M$ 。

### 5.1 内参数矩阵 $M_1$

将点的相机坐标  $[xc, yc, zc]^T$  转换到图像坐标  $[u, v, 1]^T$ , 涉及到的是相机的焦距、光轴投影点在图像中的位置、图像大小等相机的内部参数<sup>[6]</sup>, 因此相应的转换矩阵也就称为内参数矩阵。

内参数矩阵  $M_1$  是一个  $3 \times 4$  阶的矩阵, 可由下式定义:

$$M_1 = \begin{bmatrix} x & 0 & u_0 & 0 \\ 0 & y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (2)$$

式中,  $u_0, v_0$  是光轴与投影面  $O_{near}$  的交点在图像中的位置, 如前定义。  $x, y$  与相机的焦距  $f$  有关, 可由下式定义:

$$x = f/dx, \quad (3)$$

$$y = f/dy, \quad (4)$$

### 5.2 外参数矩阵 $M_2$

要将点的世界坐标转换到相机坐标, 显然这与相机相对于世界坐标系的位置有关。假设相机位置在世界坐标系中作了如下变换:

相机绕  $x$  轴转了  $rx$  角, 相应的转换矩阵为  $R_x$ ;

相机绕  $y$  轴转了  $ry$  角, 相应的转换矩阵为  $R_y$ ;

相机绕  $z$  轴转了  $rz$  角, 相应的转换矩阵为  $R_z$ ;

相机沿三轴移动了  $dx, dy, dz$ , 相应的转换矩阵为  $T$ 。

将上述 4 个变换综合起来, 就得到点的世界坐标到相机坐标的转换式:

$$[xc, yc, zc, 1]^T = M_2[xw, yw, zw, 1]^T, \quad (5)$$

式中:

$$M_2 = TR_zR_yR_x \quad (6)$$

$M_2$  中的  $R_x, R_y, R_z, T$  的表达式可以参阅计算机图形学文献[7], 不再赘述。 $R_x, R_y, R_z, T$  都是  $4 \times 4$  阶矩阵,  $M_2$  也是  $4 \times 4$  阶矩阵。由于  $M_2$  仅涉及相机的转动和移动, 与  $M_1$  相对应,  $M_2$  称为外参数矩阵。

### 5.3 标定矩阵 $M$

标定矩阵  $M$  实际就是内参数矩阵  $M_1$  与外参数矩阵  $M_2$  的乘积<sup>[1]</sup>:

$$M = M_1 \times M_2, \quad (7)$$

$M$  都是  $3 \times 4$  阶矩阵, 它反映了点的图像坐标系与世界坐标系之间的关系, 即式(1):

$$zc[u, v, 1]^T = M[xw, yw, zw, 1]^T$$

当然, 作为一个完整的系统, 系统还包括了特征图案的提取和特征图案的匹配模块。实际上这是人工立体视觉中比较困难、也比较容易出错的部分。为了系统的正确运行, 并能够对其他待测模块进行正确测试, 本系统采用人工交互的方法来完成特征图案的提取和特征图案的匹配, 以确保特征图案提取和特征图案匹配的正确性。

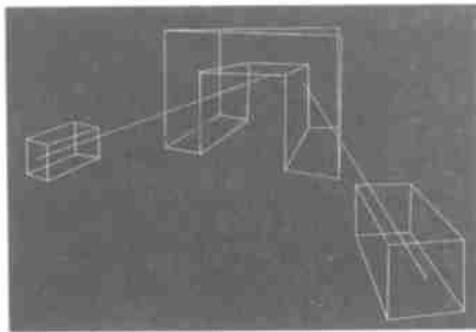
## 6 仿真系统的应用实例

建立立体视觉仿真系统, 可以用于在整个硬件系统还未搭建完成之前就可以对某些模块进行调试, 也可以对实际系统的单个模块进行单独测

试,还可以对不同硬件环境进行模拟运行,参照仿真结果选择硬件配置。

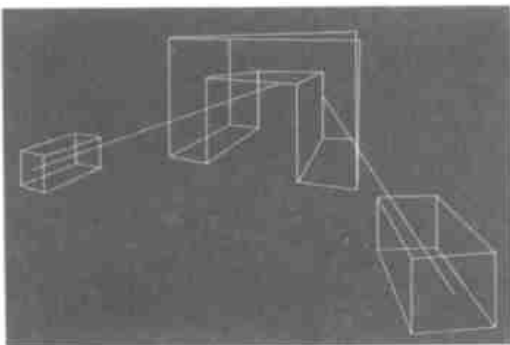
在某个人工视觉三维物体重建的研究中,用本仿真系统对其中的三维物体重建算法进行了验证。在对多个考题试算顺利通过之后,偶尔发现了重建算法中隐藏的错误,并及时加以纠正。

图 3(a) 中所示的是此次验证所用的虚拟系统的全景。包括 2 台虚拟相机以及被观测的物体,虚拟相机和虚拟物体都以线框图表示。图 3(c) 所示的是左相机所拍摄的虚拟底片;图 3(d) 所示的是右相机所拍摄的虚拟底片。在 2 张虚拟底片中采集到图像数据之后,用实际的重建算法进行了物体的重建计算,并将重建计算的结果加在系统的全景中,如图 3(b) 所示。从图 3(b) 可见,纠正后的重建算法基本正确,重建的模型与原始模型基本重合。重建的误差反映在图 3(b) 的线框图的线条有些加粗,这是因为图像数据的采集误差以及计算的舍入误差所致。



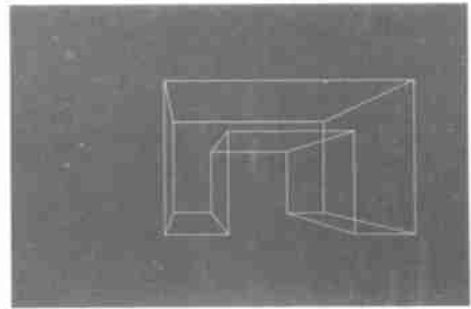
(a) 虚拟系统的全景

(a) Panorama of the virtual system



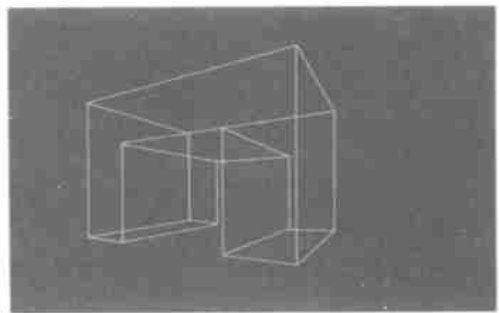
(b) 重建效果

(b) Effect of the reconstruction



(c) 虚拟左相机拍摄的图像

(c) Image of the left virtual camera



(d) 虚拟右相机拍摄的图像

(d) Image of the right virtual camera

图 3 对三维物体重建算法的验证

Fig. 3 Verification of the reconstruction arithmetic

## 7 结 论

本仿真系统是为采用人工立体视觉进行目标跟踪的研究而提供的开发工具。实际上,人工立体视觉的研究应该属于图像处理与计算机视觉的范畴,而本文所介绍的方法采用的是计算机图形学的方法,为人工立体视觉的研究提供了一个仿真平台,这是将计算机视觉研究与计算机图形学相结合的一种尝试。

利用这一仿真平台可以对立体视觉研究提供多方面的辅助,包括进行一些测试,也可以进行一些模拟。

上文所介绍的实例是这一仿真平台的一个应用。实例里重建算法中隐匿错误的顺利发现并及时纠正,说明了这一仿真平台的作用。如果仅仅根据实际系统的综合测试的最终结果,要在众多环节中去寻找这种隐匿的错误是比较困难的。这一仿真平台的成功开发与应用有助于相应的研究工作的顺利开展和进行。

**参考文献:**

- [1] 郑时雄. 机器人视觉在自动装配线中的应用[J]. 光学 精密工程, 1998, 6(1): 90-74.  
ZHEN SH X. Applications of robotics vision system in automatic assembly[J]. *Optics and Precision Engineering*, 1998, 6(1): 90-74.
- [2] 马颂德, 张正友. 计算机视觉[M]. 北京: 科学技术出版社, 1999.  
MA S D, ZHANG ZH Y. *Computer vision* [M]. Beijing: Science and Technology Press, 1999.
- [3] 董再励. 一种基于立体视觉的多视点建模方法[J]. 中国图像图形学报, 1997, 2(7): 461-467.  
DONG ZL. A multi-viewpoint modeling method based on stereo vision[J]. *Journal of Image and Graphic*, 1997, 2(7): 461-467.
- [4] 江早. OpenGL VB/VC 图形编程[M]. 北京: 科学出版社, 2001.  
JIANG Z. *OpenGL VB/VC graphic programming* [M]. Beijing: Science Press, 2001.
- [5] OpenGL API[EB/OL]. <http://airgl.myrice.com/>.
- [6] 周富强. CCD 摄像机快速标定技术[J]. 光学 精密工程, 2000, 8(1): 96-100.  
ZHOU F Q. A high speed CCD camera calibration technique[J]. *Optics and Precision Engineering*, 2000, 8(1): 96-100.
- [7] 孙家广. 计算机图形学[M]. 北京: 清华大学出版社, 1998.  
SUN J G. *Computer graphics* [M]. Beijing: Tsinghua University Press, 1998.

**作者简介:** 吴家麒(1949 - ), 男, 上海人, 上海大学机自学院副教授, 主要研究方向为精密机械和计算机图形学;  
袁振鹏(1972 - ), 男, 贵州人, 上海大学机自学院博士研究生, 主要研究方向为虚拟现实和增强现实;  
杨东英(1979 - ), 女, 上海人, 上海大学机自学院硕士研究生, 主要研究方向为机械电子工程。