

# 机器人仿真系统中动态图形的快速生成和处理

李大生 戴学耕 高理基

(精密机械与仪器系)

**摘要** 本文提出一种用于三维动态图形快速生成与处理的算法,能适应机器人实时仿真的要求。算法包括几何体定义、变换、隐藏线消除和动画技术等内容。本算法使用 Turbo-Pascal 语言在 IBM-PC 机上运行通过。

**关键词** 机器人仿真, 计算机图形学, 几何造型

## 1. 引言

机器人仿真技术是机器人学的一个重要分支,运用仿真手段,可大大加快机器人研究的步伐<sup>[1]</sup>。借助于机器人的实体模型仿真,可以实现机器人机构和控制器的优化设计,规划出最优的运动轨迹和进行碰撞、干涉测试,从而对机器人的学习、研究、设计、试验及其应用具有重要的指导意义。近年来,国外已开发了不少图形仿真系统,这些系统大都是利用通用的图形软件在功能强大的中小型机上实现的。由于仿真任务数据量大,占内存多,在一般微机(如 IBM-PC)上执行效率甚低,无法适应实时控制的需要。同时,现有的适用于微机的通用软件,如 AUTOCAD、CADKEY 等,往往缺乏三维图形处理能力,或者三维图形功能较弱,很难应用于动态图形的处理。因此,开发一种面向微机的快速动态图形处理技术就成为当务之急。

## 2. 图形生成和变换

### 1) 机器人几何实体模型的特点

一般来说,机器人的实际形状和结构是很复杂的,建立实体模型需要经过一定简化。我们可以将其分解为底部、腰部、臂以及手腕等部件。而这些部件又可以视为由一些简单的形体组合而成。例如图 1 所示的斯坦福机器人的底部,它是给定位置上的三个圆柱体的相交体。描绘其三维图形有两种方法:一种方法是直接地描绘立体图,把它看成一个整体,不进行分解;二是间接地描绘这个零件的立体图,把它看成是三个圆柱体的相交体。很明显,前者只针对这个零件绘图,缺乏通用性。后者把它看成三个圆柱体

的相交体,从本质上分析解决问题,如果改变三个圆柱体的相对位置仍可以绘制其立体图,所以通用性很好。在立体造型中,后者一般是用三个圆柱体的“并”来表示这个物体的,因而需要进行复杂的“交”、“并”、“差”、“补”等图形集合运算,势必增加图形数据结构及其处理的复杂程度。如果将其看成由三个具有固定相对位置的圆柱体装配而成,则不需要做这种运算。机器人实体模型中的每一部件一般都可以用圆柱体、长方体、圆锥、圆台,多面体等几种简单形体拼装而成,对于不能用这些简单形体拼装的部件,我们采用同简单形体一样直接定义的办法来描述。这样,既能达到目的又免去了费时的集合运算。

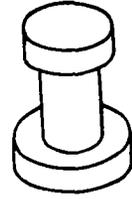


图 1

2) 图形数据结构与几何体生成

图形数据结构所表示的信息主要包括三个方面,即几何信息、拓扑信息和形体的属性方面的信息[2]。本文描述几何体是采用离散化的思想,对体素中的曲面,将其离散为平面,用一系列的平面来拟合[2]。由于没有涉及集合运算,数据结构简单,程序量及运算量都很小。每个几何体的体记录含有下列内容:

- a. 体名或体号;
- b. 体的顶点坐标;
- c. 体的外接盒参数或位置姿态矩阵。

几何体生成指建立描述几何体的数据结构。下面以圆柱体为例说明建立数据结构的方法。

如图3所示圆柱体是以一个多面体来逼近它的,其局部坐标和顶点如图2所示。

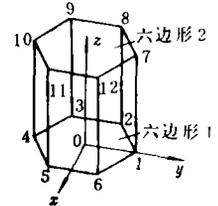


图 2

已知:  $n$ —底面等分数(图3中 $n=6$ );  $R$ —圆柱体的直径,  $H$ —圆柱体的高。

分析: 该几何体可由两个  $n$  多边形和  $n$  个矩形构成。

$n$  多边形 1 由  $n$  个顶点来描述, 设其第  $k$  个顶点在局部坐标系中的坐标为  $[x_k, y_k, z_k]$ , 则

$$x_k = -R \sin\left[(k-1) \cdot \frac{2\pi}{n}\right]; \quad y_k = R \cos\left[(k-1) \cdot \frac{2\pi}{n}\right]; \quad z_k = 0; \quad k=1, \dots, n.$$

同理得到  $n$  多边形 2 的顶点坐标

$$x_k = x_{k-n}; \quad y_k = y_{k-n}; \quad z_k = H, \quad k=n+1, \dots, 2n.$$

$n$  个矩形中每个矩形是由四个顶点来描述的。设第  $i$  个矩形的四个顶点分别为  $P_1, P_2, P_3, P_4$ , 则这些顶点可以根据几何关系由上述两个  $n$  多边形的  $2n$  个顶点中对应的四个顶点来表示:

$$P_1 \rightarrow \text{顶点 } i; \quad P_2 \rightarrow \text{顶点 } n+i; \quad P_3 \rightarrow \text{顶点 } n+i+1; \quad P_4 \rightarrow \text{顶点 } i+1.$$

这里  $i=1, \dots, n-1$ . 对于第  $n$  个矩形为:

$$P_1 \rightarrow \text{顶点 } n; \quad P_2 \rightarrow \text{顶点 } 2n; \quad P_3 \rightarrow \text{顶点 } n+1; \quad P_4 \rightarrow \text{顶点 } 1.$$

在顶点排号中, 始终保证相邻两个顶点的走向又乘指向立方体内, 这样每个面的法矢量就可以由两个走向的叉乘来表示, 而不需要存贮圆柱体各面的法矢量信息, 其余形体可以依此类推。

### 3) 几何变换

几何变换在模型化中表示物体的相对位置,而在三维图形生成中用它来表示从不同位置和方向进行观察的效果<sup>[3]</sup>。常用的变换有平移、旋转、定比例、逆变换,取景和视区变换。将两个或两个以上的变换矩阵相乘在一起则可以级联成最终的变换矩阵。在机器人几何图形生成过程中,每一几何形体,都可以先在关节坐标系中描述,然后变换到基础坐标系,最后再变换到屏坐标系,这一过程可由下式表示:

$$[X' Y' Z' 1] = [X Y Z 1]VT_iN$$

其中 $X, Y, Z$ 为几何体上一点在第 $i$ 个关节坐标系中的坐标; $X', Y', Z'$ 为几何体上一点在屏坐标系中的坐标; $V$ 为把物坐标系中的点转换为目坐标系中的点的 $4 \times 4$ 阶变换矩阵,即取景变换; $N$ 为把目坐标系中的点变换到屏坐标系中的 $4 \times 4$ 阶变换矩阵,即视区变换; $T_i$ 为描述机器人第 $i$ 个关节坐标系在基础坐标系中的位置和姿态的齐次变换矩阵,它又可以由下式表示:

$$T_i = A_1 A_2 \cdots A_i$$

其中 $A_i$ 为描述机器人杆件与杆件之间相对位置关系的齐次变换矩阵,称为 $A$ 矩阵<sup>[5]</sup>。

这样,我们已把 $A$ 矩阵引入了几何变换,因而建立了机器人实体模型中各几何体的相互联系,充分利用了信息,为进行图形处理带来了方便。

## 3. 隐藏线的快速处理

隐藏线和面的消除是计算机图形显示中引人注目的问题之一。在实际生活中,不透明的物体挡住了隐蔽部分发出的光线,以致于人们看不见这些部分。当计算机生成图形时,在把物体投影到屏坐标系的过程中,是不会自动产生这种消除隐蔽部分的作用的。为了保证计算机绘出或显示出的图形能真实地反映实际情况,避免二义性以及提高图形的清晰度,必须把这些隐线从由计算机生成的图形中清除。

为了有效地消除隐线,已有多种算法,如Roberts, Apple, Loutrel, Galimberti and montanari及warnock算法等等。这些算法基本上可分为两大类:一类是物空间算法,另一类是象空间算法。物空间算法是把注意力集中于景中各物体之间的几何关系,以便确定哪些物体是可见的,哪些物体是不可见的。而象空间算法则集中于最终形成的图形,从中确定哪些光栅象素点是可见的。在机器人仿真系统的图形处理中,使用前一种算法数据结构复杂,程序工作量大,且占据较大的内存空间,运行时间长,难以满足实时仿真的要求。使用第二种算法,因对显示硬件有一定的要求,在普通的微机上无法实现。因此,本文针对机器人仿真系统的特点,提出了一种新的算法,虽然也是在象空间内处理图形的,但对显示硬件却没有特殊要求。

早在七十年代,国外就有人提出对显示屏上某一区域进行填色,则可将这一区域的原有图形复盖。这一方法因为它与绘制油画的方式相同而叫做画家算法。画家作画从背景开始,他可以把整张帆布涂成同一色彩的背景,然后再画出物体的景色,没有必要擦除背景部分,画家只需要在背景上简单地地上色就行。新的颜色盖住了原来的,所以只有最新的颜色层是可见的。帧缓冲存储器具有同样的性质。例如填充多边形,通常按照多边形内部情况改变帧缓冲器中相应的象素值,从而将被填充的多边形输入到帧缓冲

器内<sup>[4]</sup>。此后，如果我们把第二个多边形输入到第一个的上面，那么两个多边形共有的那一部分就会按第二个多边形内部的填充属性而改变，只要第二个多边形一放上去，第一个多边形的像素设定值就被清除，于是第二个多边形盖住了第一个，如图3所示。

在中小型机上，这种算法对处理几何体之间的隐线非常实用。在处理几何体之间的隐线时，只要按一定顺序将各个几何体显示并填上不同颜色，几何体之间的隐藏线便自动被覆盖。但是这种算法在微机上实现遇到困难，因为微机上实际上只有三种颜色可用于填充。本文提出的算法成功地解决了这一问题。我们的思路很简单，即在填色的基础上，再清除填色（即填上背景色），如此循环就可以消除任意数量的几何体之间的隐线，这在微机上是能很方便和迅速实现的。

如前所述，机器人的实体模型是由多种几何体组成的组合体，对于其隐线处理可以分两步进行。第一步用法向量法<sup>[4]</sup>判别并消除那些在单个多面体中被自身遮蔽的隐面；第二步再消去那些被其它位于前方（指按投影方向）的多面体遮蔽掉的隐线，这可以采用上述的算法来完成。

### 1) 单个几何体的隐线消除

对于凸多面体，如图2所示的多面体，我们在定义时，规定组成该体的各平面法向量指向该体的体内，法向量可由定义平面的几个顶点的走向来得到，即平面的法向量可由两个走向的叉乘来表示。当一个平面的法向量在屏坐标系中的 $z$ 轴方向的投影不小于零，则这个面是可见的，反之则不可见，该面的所有边界都为隐线，必须将其在图形数据结构中删除。对于凹多面体，除了经过上述过程外，还需采用前述算法消除几何体上凸面与凹面之间的隐线。我们可以先显示凹多面体的可见凹面，然后再显示可见凸面，对可见凸面进行填充、清色，如图4所示。

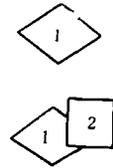
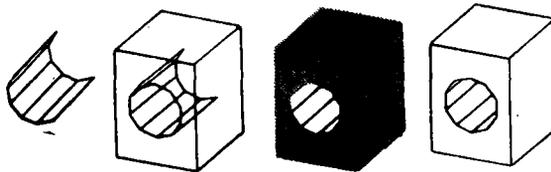


图 3



a) 显示可见凹面

b) 显示可见凸面

c) 对凸面填充

d) 清除填色

图 4

### 2) 几何体之间的隐线处理

观察图5上的三个圆柱体组成的组合体的图形，设三个圆柱体已经过单体消隐。显然，体2与体3之间，体1与体2之间构成遮挡和被遮挡关系。因此，这种关系只发生在体与体之间，下一步的隐线处理就可以以体为单位进行。我们可以按下述三个过程来消除组合体之间的隐线。

- (1) 显示不遮挡其它几何体的几何体，如体3；
- (2) 确定各体的前后位置关系，按投影方向由后到前排成一根链，如体2 → 体1；
- (3) 按链的顺序依次分别显示链中各几何体，并依次填充和清除填充。

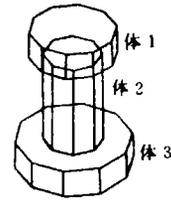


图 5

显然,按照上述步骤,图5中几何体之间的隐线即被消除。由此可知,对组合体中各几何体按投影方向的前后相对位置来排序,是实现我们的算法的关键。常用的方法是构造每个几何体的最小外接盒,然后按最小最大原则来排序<sup>[2]</sup>。但在机器人的几何模型中,已建立了每个关节坐标系相对于基础坐标系的位置姿态矩阵以及反映机器人相邻杆件之间相对位置关系的齐次变换矩阵,同时各几何体的外形尺寸也已知。这就完全确定了各几何体之间的位置关系,不需要通过建立每个体的外接盒来排序,从而减少了需要处理的数据量,加快了消去隐线的速度。

### 4. 机器人动画

在动态图形仿真中,可以将机器人运动过程中的每一个位置的图形存贮起来,然后再象动画片那样一幅一幅地推出,从而产生动的效果,生动地演示出机器人运动的全过程。在微机中,存贮一幅全屏大小的画面是很占内存的,如在 IBM-PC 机上每幅画面约占 15K 左右。由于动态仿真中需要显示很多的画面,这些画面用普通的数据结构存贮在内存中是不可能的,因而,通常利用外存来存贮,这样在动态显示中,每幅画面都必须从外存中调用,影响了显示速度。本文为了提高显示动画的速度,采用动态数据结构指针类型来存贮画面,充分利用了内存,提高了显示速度。另一方面,由于每幅画面是利用指针将其链接在一条线性链表中的,因而可以很方便地对动画片进行修改、插入、剪辑等处理。有关此技术笔者将另文详细介绍。

### 5. 结束语

本文提出的算法,已在 IBM-PC 机上用 Turbo-Pascal 语言实现了多种常用机器人的动态图形仿真,可以在几秒钟内生成和处理图6中的图形。该算法原理简单,程序量少,实用性好且速度快,图形处理效果较好,除了用于机器人仿真系统外,还可用于其它方面的图形快速生成和处理。

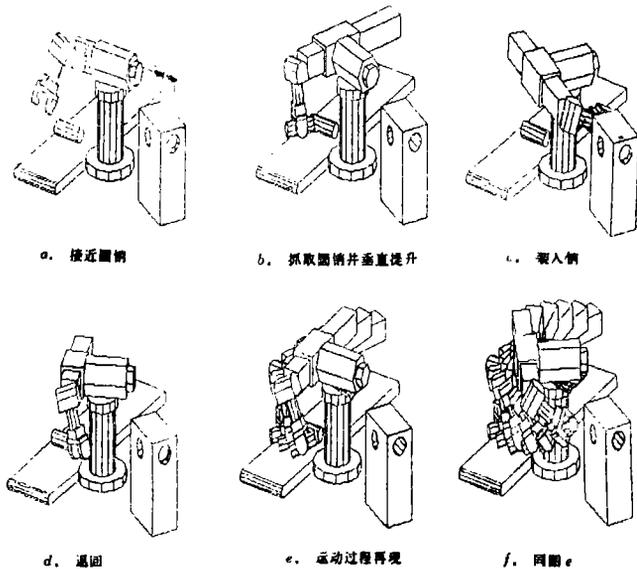


图6 斯坦福装配机器人运动仿真(IBM-PC机屏幕硬拷贝)

## 参 考 文 献

- [1] Vukbratovic M., Potkonjak V.: Kinematics and Trajectory Synthesis of Manipulation Robots, Springer-Verleg, 1982, pp,214-225
- [2] Applicon Inc: Computerized Graphic Processing Systems: System User Manual, Applicon Inc. 154 Middlesex Turnpike, Burlington, mass 01803. september 1977
- [3] W. M. 纽曼, R. F. 斯普劳尔著, 易晓东译; 对话式计算机图形显示原理, 科学出版社, 1984, pp.297-302
- [4] S. Harrington 著, 高福文译: 计算机图形学, 北师大出版社, 1985.12, pp229-240,
- [5] Paul R, Robots manipulators: Mathematics, Programing and Control, The MIT Press, 1981, pp.41-50

## An Efficient Algorithm of 3-D Dynamic Graphic Generation and Processing for Robot Manipulators Simulation System

Li Dasheng    Dai Xuegeng    Gao Liji

### Abstract

This paper presents a fast and efficient algorithm for generation and processing of 3-D dynamic graphics. It is appropriate to the purpose for the real time graphics simulation of robot manipulators. A procedure is described by which combinative geometry bodies are determined, the geometry transformation and hidden lines elimination are also introduced. The program is written in Turbo-Pascal and implemented on IBM-PC.

**Key Words** Robot Simulation, Computer Graphics, Geometry modelling