

基于仿射变换的三维到二维转换算法

刘 婧

(渭南师范学院,渭南 714000)

摘 要 根据二维仿射变换公式推导了三维仿射变换公式,给出了三维坐标系中的仿射矩阵表示公式。同时提出了一种简单的三维到二维的坐标转换公式,并且使用基于 C++ 的 QT 类库对这种算法进行了实现。这种算法可以应用于一些使用二维模拟三维的图形处理软件中,比如 flash 中的三维变换模拟和 GUI 编程中的三维图形变换等情况下。

关键词 仿射变换 三维 C++ QT

中图法分类号 TP391.75; **文献标志码** A

进行一些三维图形处理要用到三维图形的旋转、平移、拉伸等变换。同时计算机屏幕又是一个二维的平面,如何简单地将三维图形表示在二维平面上。一般的教材和网络上二维仿射变换公式和案例较多,但很少有三维仿射变换的推导和实例^[1]。于是本文根据二维仿射变换推导了三维仿射变换矩阵的表示。并借助三维图形在平面上的斜二侧画法对三维坐标进行了降维处理。使得三维图形变换和平面表示一并得到了解决。

1 坐标变换和仿射变换

在线性代数中,线性变换能够用矩阵表示。如果 T 是一个把 R_n 映射到 R_m 的线性变换,且 $\vec{x}$ 是一个具有 n 个元素的列向量,那么我们把 $m \times n$ 的矩阵 A ,称为 T 的变换矩阵,其表示形式为: $T(\vec{x}) = A\vec{x}$ 。仿射变换是线性变换的一种,经常应用于计算机图形学中^[2]。

1.1 仿射变换在二维图形中的应用

1.1.1 旋转

设 p 点坐标为 (x, y) , 将 p 点绕坐标原点逆时针旋转 θ 角度后得点 p' , 设 p' 坐标为 (x', y') 。借助几

何知识很容易推出 $x' = x\cos\theta - y\sin\theta$, $y' = x\sin\theta + y\cos\theta$ 。用矩阵表示为(1)式。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (1)$$

1.1.2 缩放

缩放公式为 $x' = s_x x$ 与 $y' = s_y y$, s_x 和 s_y 为 x 坐标和 y 坐标的缩放因子。其中用矩阵表示为式(2)。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (2)$$

1.2 仿射变换

前面研究了二维坐标的旋转和缩放。其他二维坐标变换,例如切变、反射、投影等都可以用类似的方法表示。可见常见的二维图形变换最终都可用一个 2×2 的矩阵加以表示^[4]。

前面研究的所有变换都是线性变换。某些其他变换(如平移)不是线性的,不能表示为与 2×2 矩阵相乘的形式。假定要从点 $(2, 1)$ 开始,将其旋转 90° , 在 x 方向将其平移 3 个单位, 在 y 方向将其平移 4 个单位。可通过先使用矩阵乘法再使用矩阵加法来完成此操作。如式(3)所示。

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} + \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \end{pmatrix} \quad (3)$$

其变化过程可用图 1 表示:

2011 年 6 月 28 日收到

作者简介:刘 婧(1982—), 四川成都人, 硕士, 渭南师范学院数学与信息科学学院教师, 研究方向:基础数学。

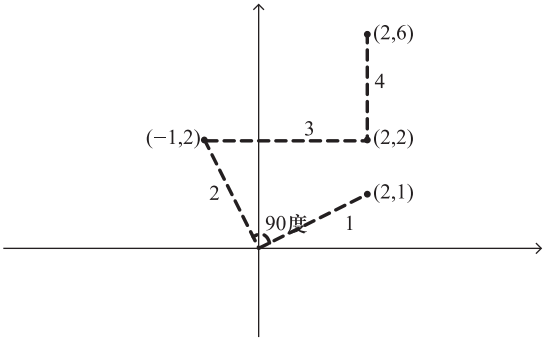


图1 旋转平移仿射

仿射变换是一个线性变换(与 2×2 矩阵相乘)外加一个平移变换(与 1×2 矩阵相加)。将仿射变换存储于一对矩阵(一个用于线性部分,一个用于平移)的替换方案是将整个变换存储于 3×3 矩阵。若要使其起作用,平面上的点必须存储于具有虚拟第三坐标的 1×3 矩阵中。通常的方法是使所有的第三坐标等于 1。例如上式可以表示为式(4)的形式。

$$\begin{pmatrix} 0 & -1 & 3 \\ 1 & 0 & 4 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \\ 1 \end{pmatrix} \quad (4)$$

式(4)中最前面的 3×3 矩阵称为仿射矩阵,其一般形式如式(5)所示。其中 $A_{11}, A_{12}, A_{21}, A_{22}$ 构成的 2×2 矩阵表示了坐标的一般线性变换的组合。 A_{13}, A_{23} 分别表示了 x 坐标和 y 坐标的平移量。其余元素固定为 0 或 1。

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ 0 & 0 & 1 \end{pmatrix} \quad (5)$$

1.2 三维仿射变换

根据上面的推导,二维仿射矩阵为一个 3×3 的矩阵。很容易地推出要同时表示一个三维坐标的线性和非线性变换。就需要一个 4×4 的矩阵。三维变换公式如式(6)所示。

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (6)$$

图样从 A_{11} 到 A_{33} 的三行三列表示了三维坐标的线性变换组合。而 A_{14}, A_{24}, A_{34} 表示了三维坐标的平移。

采用类似式(1)的推导可得出三维坐标以 z 轴为轴旋转 θ 角(无平移)的仿射矩阵为式(7)所示。

$$\begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7)$$

三维坐标以 x 和 y 轴为轴旋转 θ 角(无平移)的仿射矩阵为式(8)和式(9)所示。

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (8)$$

$$\begin{pmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9)$$

三维坐标的任意旋转变换都可以分解为以 x, y, z 为轴的旋转的组合。所以知道了以上三个变换公式。其他的变换都可以组合得来。

将式(7)、式(8)、式(9)的算法可定义一个如下的 C++ 类 Point3D 来封装:

```
class Point3D
{
public:
float x;
float y;
float z;
void rotationx(float theta)
{
y = y * cos(theta) - z * sin(theta);
z = y * sin(theta) + z * cos(theta);
}
void rotationy(float theta)
{
x = x * cos(theta) - z * sin(theta);
z = x * sin(theta) + z * cos(theta);
}
void rotationz(float theta)
{
}
```

```
x = x * cos( theta ) - y * sin( theta );
y = x * sin( theta ) + y * cos( theta );
}
Point3D( float x =0, float y =0, float z =0)
{
    this -> x = x, this -> y = y, this -> z = z;
}
void set( float x, float y, float z)
{
    this -> x = x, this -> y = y, this -> z = z;
}
};
```

2 三维坐标转化为二维坐标

2.1 单点的转化问题

基于三维图形绘制的斜二侧画法,可以给出以下的三维到二维的坐标转化算法(用 C++ 语法表示)。

```
QPoint MainWnd::point3Dto2D( Point3D p)
{
    QPoint t;
    t.setX( p.x - p.z * sin( 3.1415926/4)/2 );
    t.setY( p.y - p.z * sin( 3.1415926/4)/2 );
    return t;
}
```

该算法为图 2 所示坐标系中三维坐标的二维表示。其他方位的坐标系可依据此方法推导。考查图 2 中的 A 点和 B 点。A 点三维坐标为(5,3,0), B 点三维坐标为(5,3,8)。A 与 B 点的 x,y 坐标相同,只是 z 坐标不同。但是我们如果去掉 z 轴而将 A,B 点都看成是 xy 平面上的点,则 A,B 两点的 x,y 坐标就不相同了^[5,6]。所以上面程序中的算法就是对 z 坐标不为 0 的点进行了转化。将其转化到 xy 平面上。由于在斜二侧画法中将 z 轴画为与 x,y 轴都成 45 度的角。并且 z 轴上的距离为 x,y 轴上相同长度的一半。所以去掉 z 轴后 B 点 x 坐标计算公式为: $x' = (x - z\sin45^\circ/2)$, y 坐标计算公式为 $y' = (y - z\sin45^\circ/2)$ 。

所以利用以上算法可以很方便地使用二维绘图函数来进行三维同形的绘制。如下代码使用此转换算法后利用二维绘线函数绘出了一个棱长为

100 像素的立方体。

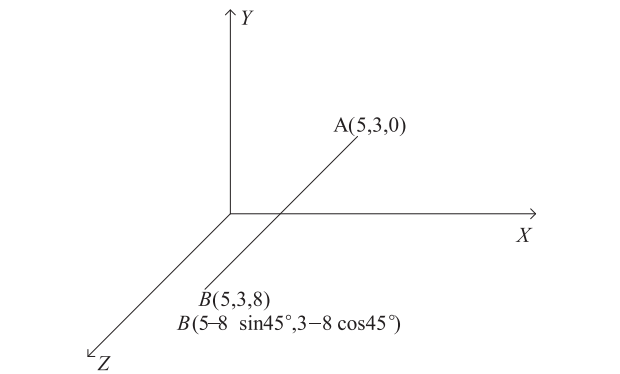


图 2 三维坐标转化为二维坐标

2.2 三维几何体的坐标转换

上面研究了单个点的转换问题,对于三维坐标系中的几何体的空间变化,可以先通过式(7)、式(8)、式(9)的仿射矩阵进行变换。然后再利用 point3Dto2D 函数将其关键点进行二维转换。最后就可以借助二维绘图函数进行空间几何体的绘制。

图 3 为使用以上的算法生成的边长为 100 的立方体的旋转动画的部分角度截图。

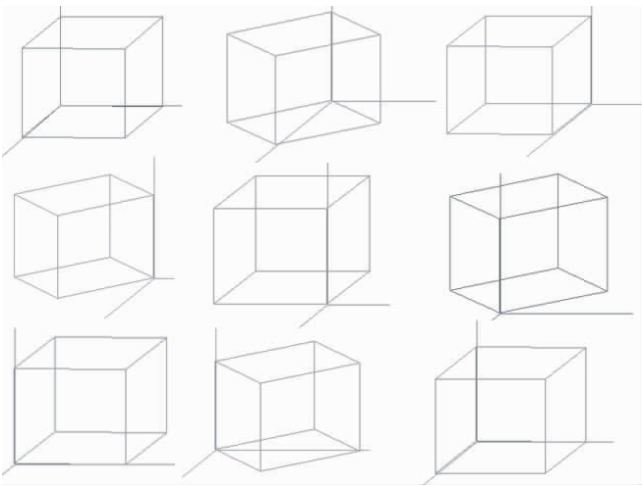


图 3 立方体旋转截图

3 总结

本文研究了二维和三维仿射变换及其应用,提出了一种简单快速的将三维坐标转变为二维坐标的算法。并通过 C++ 对算法进行了实现验证。从实验结果来看这种算法简单易行、使用方便。可以

很好地应用在一些二维动画软件中实现对三维动画的模拟。

参 考 文 献

1 Overall Contents of the Volume. Journal of Shanghai Jiaotong University, 2005;4:213—215

2 何 冰. 基于仿射变换的图像置乱改进新算法. Computer & Digital Engineering, 2011;3:105—107

3 汪文英, 张冬明, 张勇东, 等. 利用仿射变换的快速空间关系验证. 计算机辅助设计与图形学学报,2010;4:201—203

4 闻仲良, 俞胜涛. 平面仿射变换的分类及应用. 宁波大学学报(理工版),2010;1:205—201

5 饶 伟,郭业才. 基于坐标变换的常数模盲均衡新算法. Acta Electronica Sinica,2011;1:305—310

6 孙 珍. 浅谈仿射变换的应用. Journal of Yulin University,2011;2:165—166

Transform Algorithm from Three Dimension to Two Dimension
Based on Affine Transformation

LIU Jing

(Weinan Teachers University, Weinan 714000, P. R. China)

[Abstract] Three-dimensional affine transformation formula and affine array fomula in three-dimensional coordinate system, giving a simple coordinate transformation formula are deduced and its algorithm is proved by QT class libraries based on C++ which could be applied to graphic adapter that imitates three dimension by two dimension, such as three-dimensional transformation imitation in flash and three-dimensional graphic transformation in GUI programming.

[Key words] affine transformation three dimension C++ QT

(上接第 6737 页)

参 考 文 献

1 王大庆. 停车制造业渐成热门. 机电新产品导报, 2002; (08): 45—48

2 胡国宁. PLC 在多层升降横移类立体停车设备设计中的应用. 制造业自动化,2008;(01):72—73

3 程怀舟,周 霞. PLC 在立体停车库控制系统中的应用. 建筑机械,2004;(09):67—69

4 廖常初. PLC 编程及应用. 北京:机械工业出版社,2005

5 西门子 SIMATIC S7—200 可编程控制器系统手册. 西门子公司出版,2005

The Design of Control System of the Stereo Garage Based on PLC

LIU Yi-wei

(Military Representation Office of AVIC Shenyang Aircraft Corporation, Shenyang 110850, P. R. China)

[Abstract] The stereo garage has become the important way and the development direction to solve the problem of parking difficulty. It takes three-layer and ten-location of the lifting and transferring stereo garage for example, introducing its working principle and the components of the control system, focusing on the design of the control system by PLC as the core, and the corresponding program design, the stereo garage automatic control has achieved.

[Key words] stereo garage control system PLC design automatic control