

Note

这是对MIT Foundation of 3D Computer Graphics附录B的翻译，本附录讲解了2D/3D仿射函数的基础知识。本书内容仍在不断的学习中，因此本文内容会不断的改进。若有任何建议，请不吝赐教netymiles@icloud.com

注：文章中相关内容归原作者所有，翻译内容仅供学习参考。

另：Github项目[CGLearning](#)中拥有相关翻译的完整资料、内容整理、课程项目实现。

附录B（Appendix B）

仿射函数（Affine Functions）

在12章和13章中，我们讨论了变异变量的线性插值。要理解那种材料，我们首先需要学习关于仿射函数的一些简单的概念。

B.1 2D仿射（2D Affine）

只要函数 v 是下列形式，我们就将函数 v 作为变量 x 和 y 中的仿射函数

$$v(x, y) = ax + by + c \quad (\text{B.1})$$

这个函数拥有常量 a, b 和 c 。这个表达也可以被记作

$$v = [a \quad b \quad c] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

这样一种函数经常被称作“线性”，因为包含额外的附加常量项(+ c)，我们而是使用术语仿射(affine)。相同的项-仿射(affine)-被用于第三章的仿射变换，因为那些变换允许额外的常量项，在那个上下文中，其表达了平移变换。

很明显在某个 (x, y) 处评估这样一种函数 v 的方式为只是把变量 (x, y) 插入方程(B.1)中。在平面中沿着某条直线（比如说沿着三角形的一条边，或者沿着一条水平线）以平均间隔的步骤前进并且在每个步骤上快速评估函数 v 。实际上，因为函数是仿射的，每次我们沿着一个固定的单位矢量移动，函数 v 以某种固定量改变。

我们已经看到了2D仿射函数的例子。回忆一下，从第11章开始，3D投射变换映射3D中的同平面物体到3D中的同平面物体。如此，给出一个3D中的三角形和一个选定的眼睛帧和投射矩阵，一个3D中三角形上的点上的值 z_n 为那个点上的值 (x_n, y_n) 的仿射函数。

如果我们给出值 v_i ，针对 (x, y) 平面上3个点（非同线性）的函数 v 的值，比如说三角形的顶点，这确定了整个平面上的函数 v 。在这种情形中，我们说函数 v 为3个顶点上的值的线性插值体。评估3个顶点的线性插值体被称作线性插值。

计算3个顶点 $[x_i, y_i]^t$ 上的函数 v_i ，我们拥有

$$\begin{aligned} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} &= \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ 1 & 1 & 1 \end{bmatrix} \\ &:= \begin{bmatrix} a & b & c \end{bmatrix} M \end{aligned}$$

通过反转表达式，给出 v_i 和 $[x_i, y_i]^t$ ，我们可以明白怎样决定 (a, b, c) 的值

$$\begin{bmatrix} a & b & c \end{bmatrix} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} M^{-1} \quad (\text{B.2})$$

B.2 3D仿射（3D Affine）

如果一个函数 v 是下列形式的，我们就说这个函数 v 在变量 x, y 和 z 中是仿射的

$$v(x, y, z) = ax + by + cz + d \quad (\text{B.3})$$

这个函数具有常量 a, b, c 和 d 。这个函数被位于3D中四面体的4个顶点上的值所唯一确定。

B.3 向上扩展（Going Up）

给出3D中的三角形，假设我们指定了3个顶点上的函数值。可能存在很多在 (x, y, z) 中仿射的函数，它们在这3个函数值上一致。但是当被限制到三角形平面上时，所有这些函数会一致。如此，我们可以称三角形上这个被限制的函数为顶点上的线性插值体。我们仍然可以用方程(B.3)中的形

式来记录，虽然常量不再是唯一的。

在小节6.3我们的顶点着色器中，返回3D计算机图形和变异变量，当我们把一个色彩和三角形的每个顶点关联上，最自然的解释为我们希望三角形内部点的色彩被三角形上唯一的线性插值函数，这个函数在物体坐标 (x_o, y_o, z_o) 上是仿射的。

在计算机图形中，我们可以使用纹理映射把一个纹理粘贴到一个三角形上。在此处理中，我们关联两个值，被称为纹理坐标， x_t, y_t ，到三角形中的每个点上。如果我们通过把纹理坐标关联到三角形的每个顶点上来指定这种粘合动作，最自然的解读是针对三角形内部点的纹理坐标我们希望使用三角形上对应的唯一的插值函数来确定，这些插值函数在 (x_o, y_o, z_o) 中是仿射的，

作为一种相当自我参照的例子，我们甚至将三角形中一个点的三个物体坐标的每个都当作 (x_o, y_o, z_o) 中的仿射函数。例如 $x_o(x_o, y_o, z_o) = x_o$ （也就是说，当 $b = c = d = 0$ 时， $a = 1$ ）。这意味着在我们的顶点着色器中，`vPosition`中的3个顶点应该被解读为在 (x_o, y_o, z_o) 坐标值上的3个函数。

针对这个原因，OpenGL的默认语义为插值所有变异变量为三角形上的函数，这些函数在 (x_o, y_o, z_o) 中是仿射的。正如我们会在下面小节B.5中会看到的，这等价于在眼睛坐标 (x_e, y_e, z_e) 上并行的一个函数，但是并不等价于在标准化设备坐标 (x_n, y_n, z_n) 上的函数。

B.4 向下扩展 (Going Down)

当被限定到3D中的一个三角形时，如果我们拥有在变量 (x, y, z) 中并行的函数 v ，那么我们可以使用三角形是平面的事实将 v 记为只在两个变量上仿射的一个函数。例如，当被投射到 (x, y) 平面时，假设三角形拥有非零区域，那么，在3D中三角形的每个点上， z 的值本身是变量 (x, y) 上的仿射函数，比方说 $z = ex + fy + g$ 。对于 z 你可以采用这种表达式，并且你可以把它插入到针对函数 v 的仿射表达式中。这给出了我们下面形式的函数

$$\begin{aligned} v &= ax + by + c(ex + fy + g) + d \\ &= hx + iy + j \end{aligned}$$

这个表达式针对某种常量 (h, i, j) 。在这步被完成后，对 z 值的依赖消失，剩下的就是 (x, y) 变量上的某种仿射函数。理解了这些，我们可以使用方程式(8.2)直接计算合适的 (h, i, j) 因子。

B.5 旁边扩展 (Going Sideways)

假设我们拥有系列形式的矩阵表达式

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (\text{B.4})$$

这种表达式针对某种 4×4 矩阵 M (此处 M 甚至不必是仿射矩阵)。那么，只要独立地观察矩阵中的4行内容，我们就会看到 x', y', z' 和 w' 为 (x, y, z) 的并行函数。

如果我们拥有在变量 (x, y, z) 中并行的函数 v ，那么，给出方程 (B.4) 中的关系，我们可以看到函数 v 在 (x', y', z', w') 中也是仿射的（实际上是线性的）。要明白这种情形，注意：

$$v = [a \quad b \quad c \quad d] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (\text{B.5})$$

$$= [a \quad b \quad c \quad d] M^{-1} \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} \quad (\text{B.6})$$

$$= [e \quad f \quad g \quad h] \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} \quad (\text{B.7})$$

表达式针对某种值 e, f, g, h 。

我们不得不注意的唯一时刻是当除法被完成的时候。例如，给出关系

$$\begin{bmatrix} x'w' \\ y'w' \\ z'w' \\ w' \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

通常不是这种情形：函数 v 在变量 (x, y, z) 中是仿射的也会在变量 (x', y', z') 或 (x', y', z', w') 中是仿射的。在第13章中我们会看到这种非仿射函数如何可以被有效评估。