**第3章 变换**

**1，线性变换**

我们称函数t为线性变换，当且仅当此函数具有下列性质：

t(u+v) = t(u) +t(v)

t(ku) = kt(u)

缩放矩阵：

S =

旋转矩阵(绕x轴)：

Rx =

**2，仿射变换**

仿射变换：由一个线性变换与一个平移变换组合而成。

平移操作只能应用于点，不应作用于向量。

齐次坐标：在采用其次坐标表示法时，我们将坐标扩充为四元组(x,y,z,0)表示向量，(x,y,z,1)表示点。

w=1能使点被正确的平移，w=0则可以防止向量坐标受到平移操作的影响。

仿射变换的矩阵表示：

复合变换的变换顺序：先缩放，后旋转再平移。

**3，坐标系变换**

向量的坐标变换

p= xu + yv + zw

向量u,v,w分别是指向标架A中x轴，y轴和z轴正方向上的单位向量。

点的坐标变换

p = xu + yv +zw + Q

向量u,v,w分别是指向标架A中x轴，y轴和z轴正方向上的单位向量，Q为标架A中的原点。

坐标变换的矩阵表示

**4，DirectXMath库提供的变换函数**

//构建缩放矩阵

inline XMMATRIX XM\_CALLCONV XMMatrixScaling(float ScaleX, float ScaleY, float ScaleZ)

inline XMMATRIX XM\_CALLCONV XMMatrixScalingFromVector(FXMVECTOR Scale)

//构建一个绕X轴旋转的矩阵Rx

inline XMMATRIX XM\_CALLCONV XMMatrixRotationX(float Angle)

//构建一个绕任意轴旋转的矩阵Rx

inline XMMATRIX XM\_CALLCONV XMMatrixRotationAxis(FXMVECTOR Axis, float Angle)

//构建平移矩阵

inline XMMATRIX XM\_CALLCONV XMMatrixTranslation(float OffsetX, float OffsetY, float OffsetZ)

inline XMMATRIX XM\_CALLCONV XMMatrixTranslationFromVector(FXMVECTOR Offset)

//计算向量与矩阵的乘积vM，此函数为针对点的变换，即总是默认令Vw为1

inline XMVECTOR XM\_CALLCONV XMVector3TransformCoord(FXMVECTOR V, FXMMATRIX M)

//计算向量与矩阵的乘积vM，此函数为针对向量的变换，即总是默认令Vw为0

inline XMVECTOR XM\_CALLCONV XMVector3TransformNormal(FXMVECTOR V, FXMMATRIX M)