# 1.4 叉积

叉积（cross product）是向量数学定义的第二种乘法形式。它与点积不同，点积的计算结果是一个标量，而叉积的计算结果是一个向量；另外，叉积只能用于3D向量（2D向量没有叉积）。通过对两个3D向量**u**和**v**计算叉积，可以得到第3个向量**w**，该向量同时垂直于**u**和**v**。也就是说，**w**即垂直于**u**，**w**也垂直于**v**（参见图1.13）。设**u**=（*u*x,*u*y,*u*z），**v**=（*v*x,*v*y,*v*z），则叉积为：

 （1.5）

****

**图1.13 通过为两个3D向量u和v计算叉积，可以得到第3个向量w，该向量同时垂直于u和v。如果读者抬起左手，将拇指之外的其他4个手指指向第一个向量u的方向，然后朝着v的方向沿角度0≤*θ*≤π弯曲手指，此时拇指所指的方向即为w=u×v的方向；这叫做左手拇指定则（left-hand-thumb rule）。**

注意：如果你处理的是一个右手坐标系，则需要使用右手拇指定则（right-hand-thumb rule）：如果抬起右手，将拇指之外的其他4个手指指向第一个向量**u**的方向，然后朝着**v**的方向沿角度0≤*θ*≤π弯曲手指，此时拇指所指的方向即为**w**=**u**×**v**的方向。

【例1.6】

设u= (2, 1,3)、v= (2, 0,0)。计算**w**=**u**×**v**和**z**=**v**×**u**，并验证**w**既垂直于**u**，也垂直于**v**。由公式1.5可得，

**w**=**u**×**v**

= (2, 1,3)×(2, 0,0)

= (1 ∙ 0 − 3 ∙ 0,3 ∙ 2 − 2 ∙ 0,2 ∙ 0 − 1 ∙ 2)

= (0, 6, −2)

和

**z**=**v**×**u**

= (2, 0,0) × (2, 1,3)

= (0 ∙ 3 − 0 ∙ 1,0 ∙ 2 − 2 ∙ 3,2 ∙ 1 − 0 ∙ 2)

= (0, −6,2)

该结果说明**u**×**v**≠**v**×**u**。也就是，叉积不支持交换律。实际上，它可以表达为**u**×**v**= −**v**×**u**。读者可以通过左手拇指定则来判断由这个叉积得出的向量。如果你从第1个向量朝着第2个向量的方向卷曲手指时（通常选择角度最小的路径），你的拇指会指向最终得到的向量的方向，如图1.13所示。

为了说明**w**既垂直于**u**，也垂直于**v**，我们回顾1.3节的内容：如果**u**∙**v**=0，则**u**⊥**v**（即，向量相互垂直）。因为

**w**∙**u**= (0, 6, −2) ∙ (2, 1,3) =0 ∙ 2 +6 ∙ 1+ (−2) ∙ 3 =0

和

**w**∙**v**= (0, 6, −2) ∙ (2, 0,0) =0 ∙ 2 +6 ∙ 0+ (−2) ∙ 0 =0

我们得出结论：**w**既垂直于**u**，也垂直于**v**。

## 1.4.1 2D伪叉积

叉积可以计算垂直于给定两个3D向量的向量。在2D的情况中，并不存在这种情况，但我们常常要求出垂直于给定2D向量**u** = (*u*x,*u*y)的向量**v**。图1.14展示了这种操作的几何图景，从图中可以看出**v** = (-*u*y, *u*x)。数学证明很简单：

**u**·**v**= (*u*x,*u*y)·(-*u*y,*u*x)= -*u*x*u*y+ *u*y*u*x= 0

所以**u**⊥**v**。而**u**·-**v**=*u*x*u*y+ *u*y(-*u*x) =0，也为零，所以还能得出结论：**u**⊥-**v**。

****

**图1.14 u向量的2D伪叉积为一个垂直于它的向量v**

## 1.4.2 使用叉积进行正交规范化

在1.3.1节中，我们介绍了一种正交化一组向量的处理方法。对3D的情况来说，还可以使用叉积对一组向量（这些向量近似正交，但由于数值累积精度的误差，会变得不再正交）进行正交规范化操作。可参见图1.15理解这个过程的几何图景：

1．令

2．令

3．令。由后面的练习14可知，因为**w**2⊥**w**0且||**w**2||=||**w**0||=1，所以||**w**2×**w**0||=1，这样我们就无需进行规范化操作了。

****

**图1.15 使用叉积进行3D正交化**

至此，完成了向量集{**w**0,**w**1,**w**2}的正交规范化处理。

**注意**：在前面的示例中，我们首先令，表示从**v**0变化到**w**0并没有改变向量的方向，只是改变了大小。但是，**w**1和**w**2的方向与**v**1和**v**2的方向并不相同。根据应用程序的需要，选择哪个向量不改变方向可能会很重要。例如，本书的后面我们将会使用三个正交向量{**v**0,**v**1,**v**2}代表相机的朝向，其中第三个向量**v**2表示相机的观察方向。当正交规范化这三个向量时，我们常常不想改变观察的方向，因此，我们会首先使用上面的算法处理**v**2，然后修改**v**0和**v**1生成正交向量。