# 1.3 点积

点积（dot product）是向量乘法的一种形式，它的计算结果是一个标量值；由于这一原因，有时也将点积称为标量积（scalar product）。设**u**=（*u*x,*u*y,*u*z），**v**=（*v*x,*v*y,*v*z），则点积定义如下：

（1.3）

简言之，点积等于两个向量对应分量的乘积之和。

点积的定义不存在任何明显的几何含义。但是，使用余弦定理可以发现存在如下关系：

（1.4）

其中，*θ*表示向量**u**和**v**之间的夹角，且0≤*θ*≤π（参见图1.9）。公式1.4说明这两个向量的点积等于向量夹角的余弦值和向量模之间的乘积。在特殊情况下，如果**u**和**v**都是单位向量，那么**u**∙**v**就等于它们之间夹角的余弦值（即，**u**∙**v**=cos*θ*）。

****

**图1.9 在左图中，u、v之间的夹角*θ*为锐角。在右图中，u、v之间的夹角*θ*为钝角。记住，当我们提及两个向量之间的夹角时，通常指的是最小的角，也就是角度*θ*，且0≤*θ*≤π。**

公式1.4提供了一些有用的点积的几何性质：

1．如果**u**∙**v**=0，则**u**⊥**v**（即，向量相互垂直）。

2．如果**u**∙**v**＞0，则两个向量之间的夹角*θ*小于90º（即，向量形成一个锐角）。

3．如果**u**∙**v**＜0，则两个向量之间的夹角*θ*大于90º（即，向量形成一个钝角）。

**注意**：“相互垂直”也可称为“互成直角”。

**【例1.4】**

设**u**=（1, 2,3）、**v**=（−4, 0, −1）。求**u**和**v**之间的夹角。首先，我们要做如下计算：



现在，由公式1.4解得*θ*为：



**【例 1.5】**

考虑图1.10。给出**v**和单位向量**n**，推导出一个使用点积求解向量**p**的公式。

****

**图1.10 v在n上的正交投影。**

首先，从该图中可以看到标量*k*可以使**p**=*k***n**；而且，由于我们已知‖**n**‖=1，所以有‖**p**‖=‖*k***n**‖=| *k*|‖**n**‖=|*k*|。（注意，当且仅当**p**与**n**的方向相反时，*k*为负数。）我们使用三角函数，可以得出*k*=‖**v**‖cos*θ*；由此，**p**=*k***n**=（‖**v**‖cos*θ*）**n**。不过，因为**n**是一个单位向量，所以我们可以用另一种方式进行表达：



请注意，这里的*k*=**v**∙**n**，它说明了当**n**为单位向量时**v**∙**n**的几何含义。我们将**p**称为**v**在**n**上的正交投影（orthogonal projection），并记为



如果我们把**v**理解为一个作用力，那么**p**可以被认为是**v**在方向**n**上的分力。同理，向量**w**=perp**n**(**v**)=**v**−**p**是与**n**垂直方向上的分力。可以看到**v**=**p**+**w**，这说明我们已经将向量**v**分解成了两个相互垂直的向量**p**和**w**。

如果n不是一个单位向量，那我们可以对它进行规范化，使其保持单位长度。通过用单位向量来代替**n**，可以得到一个更通用的投影公式：



## 1.3.1 正交化

若一个向量集合{**v**0，……，**v**n-1 }中的向量相互正交（即集合中的每一个向量与其他向量正交）并具有单位长度，我们将这个集合称之为规范化正交集。有时一组向量几乎是正交的，但又不完全是，一个常见的任务就是使其正交。在三维计算机图形中，开始时通常是一个规范化正交的向量集合，但由于计算精度问题，这个集合就会逐渐成为非规范化的了。我们主要关心的是2D和3D的情况下任何处理这个问题（即，含有两个和三个向量的情况）。

首先讨论简单的2D情况。假设有一组向量为{**v**0,**v**1}，我们要将它们正交到一个规范正交集{**w**0,**w**1}中，如图1.11所示。首先令**w**0=**v**0，然后修改**v**1使它与**w**0垂直；这需要减去**v**1向量在**w**0上的投影：

**w**1= **v**1- proj**w0**(**v**1)

现在就有了一组互相垂直的向量{**w**0,**w**1}；最后需要规范化**w**0和**w**1才能构建规范化的正交集。

****

**图1.11 2D正交化**

处理3D情况的原理与2D相同，但需要更多的步骤。假设有一组向量{**v**0,**v**1,**v**2}需要正交规范化到{**w**0,**w**1,**w**2}，如图1.12所示。首先令**w**0=**v**0，然后修改**v**1使之垂直于**w**0；这需要从**v**1中减去**v**1在**w**0方向上的投影：

**w**1= **v**1- proj**w0**(**v**1)

下一步需要令**v**2同时垂直于**w**0和**w**1，这需要从**v**2中减去**v**2在**w**0上的投影再减去**v**2在**w**1上的投影：

**w**2= **v**2- proj**w0**(**v**2) - proj**w1**(**v**2)

现在就有了一组互相垂直的向量{**w**0,**w**1,**w**2}；最后需要规范化**w**0、**w**1和**w**2才能构建规范化的正交集。

**图1.12 3D正交化**

要规范正交化任意数量的向量集{**v**0,…,**v**n-1}，我们需要按照通常叫做[Gram-Schmidt正交化](http://zh.wikipedia.org/wiki/%E6%A0%BC%E6%8B%89%E5%A7%86-%E6%96%BD%E5%AF%86%E7%89%B9%E6%AD%A3%E4%BA%A4%E5%8C%96)的处理过程进行：

基本步骤：令**w**0=**v**0

for 1 ≤ *i* ≤ *n*-1，令

规范化步骤：令

原理与上面是类似的，当选取一个向量**v**i将它添加到规范化的正交集时，我们需要减去这个向量在正交集中其他向量（**w**0,**w**1,…,**w**i-1）上的投影，这样可以确保这个新添的向量与正交集中的其他向量垂直。