

# 目录

第一章	简介 .....	1
第二章	图像形成 .....	3
2.1	几何基元与变换 .....	4
2.1.1	二维变换 .....	8
2.1.2	3D 转换 .....	11
2.1.3	三维旋转 .....	12
2.1.4	3D 到 2D 投影 .....	18
2.1.5	镜头变形 .....	28
2.2	光度成像 .....	30
2.2.1	光源 .....	30
2.2.2	反射和阴影 .....	31
2.2.3	光学 .....	36
2.3	数码相机 .....	40
2.3.1	采样和混叠 .....	44
2.3.2	颜色 .....	46
2.3.3	压缩 .....	54
2.4	更多的阅读 .....	56
2.5	练习题 .....	56



# 第一章 简介



## 第二章 图像形成

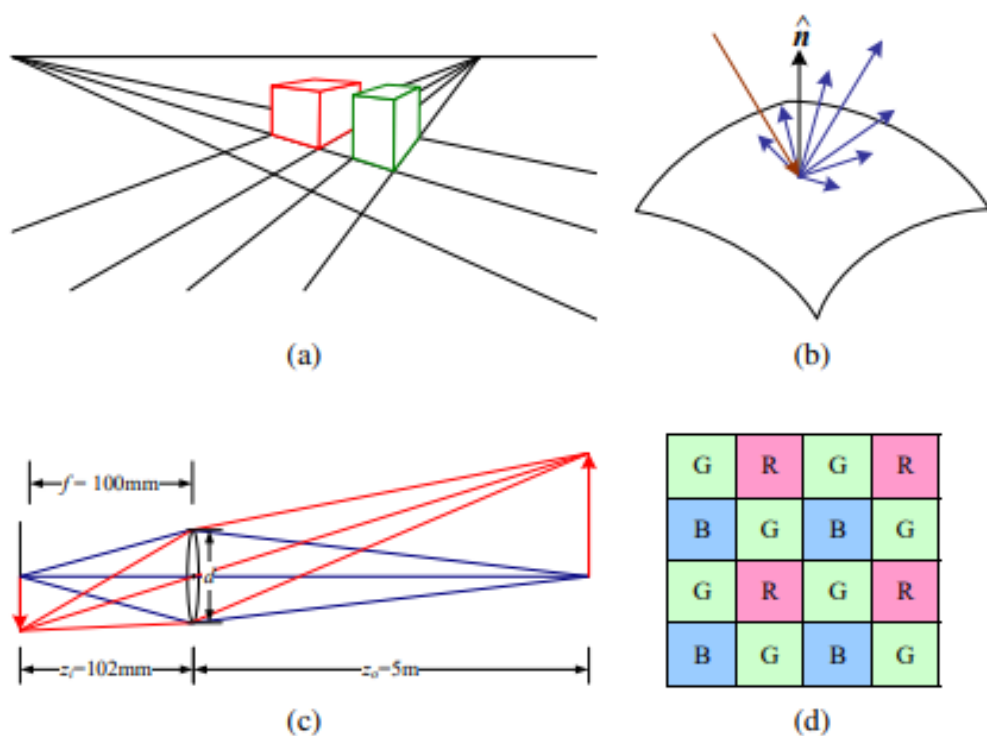


图 2-1 图像形成过程的几个组成部分: (a) 透视投影; (b) 撞击表面时的光散射; (c) 透镜光学器件; (d) 拜耳色滤光片阵列。

3 【新文本: 本章更新不多, 大多是更新的参考文献, 例如, 在 2.3 节的开头。】

4 在我们能够智能地分析和操作图像之前, 我们需要建立一个词汇表来描述场景的几何形状。我  
5 们还需要理解在给定一组照明条件、场景几何形状、表面特性和照相机光学时产生特定图像的图像  
6 形成过程。在本章中, 我们提出了这种图像形成过程的一个简化模型。

第 2.1 节介绍了整个书中使用的基本几何基语 (点、线和平面), 以及将这些三维量投影到二维图像特征中的几何转换 (图 2.1a)。第 2.2 节描述了照明、表面特性 (图 2.1b) 和相机光学器件 (图 2.1c) 如何相互作用, 以产生落到图像传感器上的颜色值。第 2.3 节描述了如何将连续彩色图像变成图像传感器内的离散数字样本 (图 2.1d), 以及如何避免 (或至少表征) 采样缺陷, 如混叠。

本章所涵盖的材料只是对一套非常丰富和深入的主题的简要总结, 传统上涵盖在一些单独的领域。关于点、线、平面和投影的几何图形的更彻底的介绍可以在关于多视图几何的教科书中找到 (Hartley 和 Zisserman, 2004 年; Faugeras 和 Luong, 2001 年) 和计算机图形学 (Hughes, vanDam 等人。2013)。图像形成 (合成) 过程传统上是作为计算机图形学课程的一部分来教授的 (格拉斯纳 1995; 瓦特 1995; 休斯, 范达姆等。2013 年; 但它也被研究在基于物理学的计算机视觉领域 (沃尔夫、谢弗和希利 1992a)。在光学中研究了相机镜头系统的行为 (Möller, 1988; Ray, 2002; Hecht, 2015)。一些关于颜色理论的好书是 (Healey 和 Shafer, 1992 年; Wandell, 1995 年; Wyszecki 和 Stiles, 2000 年), (Livingstone, 2008 年) 提供了一个更有趣和非正式的介绍颜色感知的主题。有关抽样和混叠有关的主题载于关于信号和图像处理的教科书 (Crane 1997 年; J'Ahne 1997 年; Oppenheim 和 Schaffer 1996 年; Oppenheim、Schaffer 和 Buck 1999 年; Pratt 2007 年; Russ 2007 年; e2008; Gonzalez 和 Woods 2017)。

**学生须知:** 如果你已经学习过计算机图形学, 你可能想浏览 2.1 节中的材料, 尽管 2.1.4 节末尾关于投影深度和以物体为中心的投影的部分对你来说可能是新的。同样, 物理学生 (以及计算机图形学学生) 将大多熟悉 2.2 节。最后, 具有良好图像处理背景的学生将已经熟悉采样问题 (第 2.3 节) 以及第 3 章中的一些材料。

## 2.1 几何基元与变换

在本节中, 我们将介绍本教材中使用的基本 2D 和 3D 基元, 即点、线和平面。我们还描述了如何将三维特征投影到二维特征。有关这些主题的更详细描述 (以及更温和、更直观的介绍) 可以在多视图几何教科书中找到 (《Hartley and Zisserman 2004》; 《Faugeras and Luong 2001》)。

几何基元构成用于描述三维形状的基本构造块。在本节中, 我们将介绍点、线和平面。本书后面的章节将讨论曲线 (7.3 节和 12.2 节)、曲面 (13.3 节) 和体积 (13.5 节)。

**2D 点:** 二维点 (图像中的像素坐标) 可以用一对值表示, 或者用公式 2.1 代替:

$$X = \begin{bmatrix} x \\ y \end{bmatrix}, \quad (2.1)$$

(如引言中所述, 我们使用  $(x_1, x_2, \dots)$  表示法来表示列向量。)

二维点也可以用齐次坐标表示,  $\tilde{X} = (\tilde{x}, \tilde{y}, \tilde{w}) \in P^2$ , 其中仅按比例不同的向量被认为是等价的。  $P^2 = R^3 - (0, 0, 0)$ , 被称为二维射影空间。

齐次向量  $\tilde{\mathbf{X}}$ ，可通过除以最后一个元素  $\tilde{\omega}$ ，转换回非齐次向量  $\mathbf{X}$ ，即：

$$\tilde{\mathbf{X}} = (\tilde{x}, \tilde{y}, \tilde{\omega}) = \tilde{\omega} (x, y, 1) = \tilde{\omega} \bar{\mathbf{X}}, \quad (2.2)$$

其中  $\bar{\mathbf{X}} = (x, y, 1)$ ，是增广向量。最后一个元素为  $\tilde{\omega} = 0$  的齐次点称为理想点或无穷远点，没有等价的非齐次表示。

**2D 线：**二维线也可以用齐次坐标表示  $\tilde{\mathbf{l}} = (a, b, c)$ 。相应的直线方程为

$$\bar{\mathbf{x}} \cdot \tilde{\mathbf{l}} = ax + by + c = 0. \quad (2.3)$$

我们可以规范化直线方程向量，使  $\mathbf{l} = (\hat{n}_x, \hat{n}_y, d) = (\hat{\mathbf{n}}, d)$ ，其中  $\|\hat{\mathbf{n}}\| = 1$ 。 $\hat{\mathbf{n}}$  在这种情况下，是垂直于直线的法向量， $d$  是它到原点的距离（图 2.2）。（这种标准化的一个例外是无穷远处的直线  $\tilde{\mathbf{l}} = (0, 0, 1)$ ，它包括无穷远处的所有（理想）点。）

我们也可以将  $\hat{\mathbf{n}}$  表示为旋转角  $\theta$  的函数， $\hat{\mathbf{n}} = (\hat{n}_x, \hat{n}_y) = (\cos \theta, \sin \theta)$ （图 2.2a）。这种表示法通常用于霍夫变换寻线算法，相关内容在 7.4.2 节中讨论。线性组合  $(\theta, d)$  也称为极坐标。

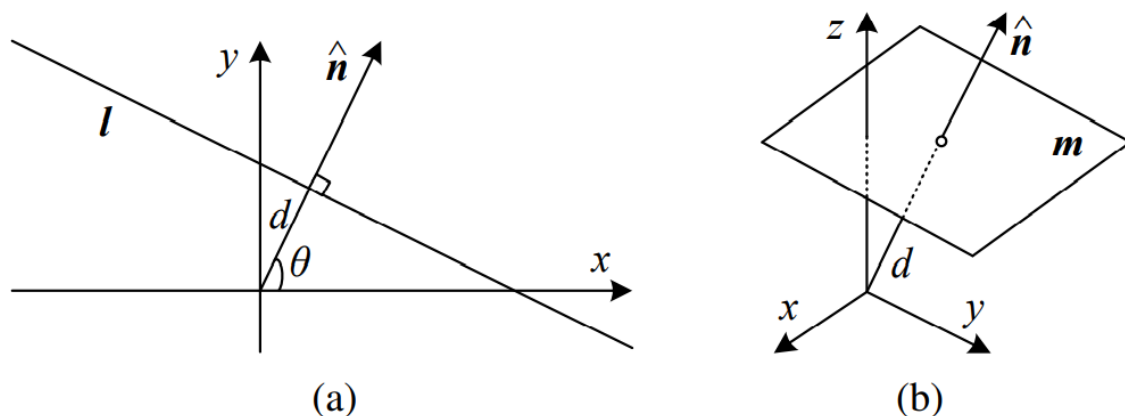


图 2-2 图像形成过程的几个组成部分:(a) 透视投影；(b) 撞击表面时的光散射；(c) 透镜光学；(d) 拜耳彩色滤光片阵列。

当使用齐次坐标时，我们可以将两条直线的交点计算为

$$\tilde{\mathbf{l}} = \tilde{\mathbf{x}}_1 \times \tilde{\mathbf{x}}_2, \quad (2.4)$$

其中  $\times$  是叉积运算符。类似地，连接两点的线可以写成

$$\tilde{\mathbf{x}} = \tilde{\mathbf{l}}_1 \times \tilde{\mathbf{l}}_2. \quad (2.5)$$

当试图将交点拟合到多条直线或相反地，将直线拟合到多个点时，可以使用最小二乘法（第 8.1.1 节和附录 a.2），如练习 2.1 所述。

48 **二维曲线**: 还有一些代数曲线可以用简单的多项式表示齐次方程。例如, 圆锥曲线 (之所以称为圆  
49 锥曲线, 是因为它们是平面和三维圆锥体的交点) 可以用二次方程来表示:

$$\tilde{\mathbf{x}}^T \mathbf{Q} \tilde{\mathbf{x}} = 0. \quad (2.6)$$

50 二次方程在多视点几何研究和摄像机标定中起着重要的作用, 但在《Hartley and Zisserman 2004; Faugeras  
51 and Luong 2001》中并未广泛应用。

52 **三维点**: 三维点坐标可以用非齐次坐标  $\mathbf{x} = (x, y, z) \in R^3$  或者齐次坐标  $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{\omega}) \in P^3$  表示。  
53 和以前一样, 有时用增广向量  $\bar{\mathbf{x}} = (x, y, z, 1)$  来表示 3D 点是有用的, 此时  $\tilde{\mathbf{x}} = \tilde{\omega} \bar{\mathbf{x}}$ 。

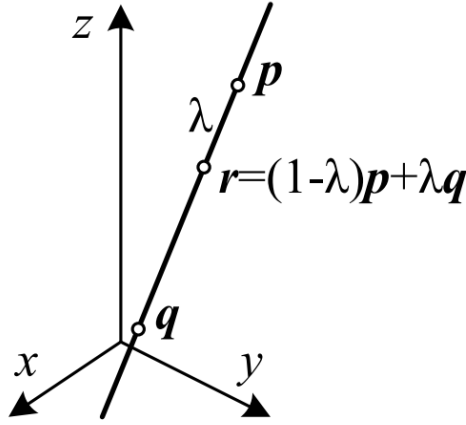


图 2-3 三维线方程,  $\mathbf{r} = (1 - \lambda)\mathbf{p} + \lambda\mathbf{q}$ .

54 **三维平面** 三维平面也可以用相应的平面方程表示为齐次坐标  $\tilde{\mathbf{m}} = (a, b, c, d)$ , 相应的平面方程为

$$\bar{\mathbf{x}} \cdot \tilde{\mathbf{m}} = ax + by + cz + d = 0. \quad (2.7)$$

55 我们也可以将平面方程正规化为  $\mathbf{m} = (\hat{n}_x, \hat{n}_y, \hat{n}_z, d) = (\hat{\mathbf{n}}, d)$ , 其中  $\|\hat{\mathbf{n}}\| = 1$ 。在这种情况下,  $\hat{\mathbf{n}}$  是垂  
56 直于平面的法向量,  $d$  是它到原点的距离 (图 2.2b)。与二维线的情况一样, 无穷远  $\tilde{\mathbf{m}} = (0, 0, 0, 1)$   
57 处的平面包含无穷远处的所有点, 不能标准化 (例如, 它没有唯一的法线或有限距离)。

58 我们可以把  $\hat{\mathbf{n}}$  表示为两个角  $(\theta, \phi)$  的函数:

$$\hat{\mathbf{n}} = (\cos \theta \cos \phi, \sin \theta \cos \phi, \sin \phi), \quad (2.8)$$

59 例如, 使用球坐标, 但它们比极坐标更不常用, 因为它们不能对可能的法向量空间进行均匀采样。



60 **三维线：**三维中的线比二维或三维平面中的任何一条线都不那么优雅。一种可能的表示方式是在直  
61 线上使用两个点  $(p, q)$ 。线上的任何其他点都可以表示为这两个点的线性组合

$$r = (1 - \lambda)p + \lambda q, \quad (2.9)$$

62 如图 2.3 所示。如果我们限制  $0 \leq \lambda \leq 1$ ，我们得到连接  $p$  和  $q$  的线段。

63 如果我们使用齐次坐标，我们可以把直线写成

$$\tilde{r} = \mu \tilde{p} + \lambda \tilde{q}. \quad (2.10)$$

64 这种情况的一个特例是当第二个点在无穷远处时，即  $\tilde{q} = (\hat{d}_x, \hat{d}_y, \hat{d}_z, 0) = (\hat{d}, 0)$ 。这里，我们看到  
65  $\hat{d}$  是这条线的方向。然后我们可以将不均匀的三维直线方程改写为

$$r = p + \lambda \hat{d}. \quad (2.11)$$

66 三维线端点表示的一个缺点是它具有太多的自由度，即六（每个端点三个），而不是三维线真正拥  
67 有的四个度。但是，如果我们将线上的两点固定在特定平面上，则得到一个四自由度的表示。例如，  
68 如果我们表示接近垂直的线，那么  $z = 0$  和  $z = 0$  形成两个合适的平面，即两个平面中的  $(x, y)$  坐  
69 标提供描述该线的四个坐标。这种双平面参数化用于第 14 章描述的基于光场和 Lumigraph 图像的  
70 渲染系统中，以表示摄影机在对象前面移动时所看到的光线集合。双端点表示法对于表示线段也很  
71 有用，即使无法看到它们的确切端点（只能猜测）。

72 如果我们希望代表所有可能的线而不偏向任何特定的方向，我们可以使用 Plücker coordinates (Hart-  
73 ley and Zisserman 2004, 第 2 章; Faugeras and Luong 2001, 第 3 章)。这些坐标是  $4 \times 4$  斜对称矩阵中的六个  
74 独立的非零项

$$L = \tilde{p}\tilde{q}^T - \tilde{q}\tilde{p}^T, \quad (2.12)$$

75 其中  $\tilde{p}$ ,  $\tilde{q}$  和是直线上的任意两个（不相同）点。这种表示只有四个自由度，因为  $L$  是齐次的，并且  
76 满足  $\det(L) = 0$ ，这导致了对 Plücker 坐标的二次约束。

77 实际上，对于大多数应用程序来说，最小表示不是必需的。三维线的适当模型可以通过估计其  
78 方向（例如，对于建筑而言，可以提前知道方向）和线的可见部分内的某个点（见第 11.4.8 节）或  
79 通过使用两个端点获得，因为线通常作为有限线段可见。但是，如果您对关于最小线参数化主题的  
80 更多细节感兴趣，Förstner (2005) 讨论了在射影几何中推导和建模三维直线的各种方法，以及如何估  
81 计这些拟合模型中的不确定性。

82 **三维二次曲面：**圆锥曲线的三维模拟是二次曲面

$$\bar{x}^T Q \bar{x} = 0 \quad (2.13)$$

(Hartley and Zisserman 2004, 第 2 章)。同样, 虽然二次曲面在多视图几何的研究中很有用, 也可以作为有用的建模基元 (球体、椭球体、圆柱体), 但在本书中我们并没有详细研究它们。

### 2.1.1 2D 变换

定义了我们的基本原语之后, 我们现在可以将注意力转向如何对它们进行转换了。最简单的变换发生在 2D 平面, 如图 2.4 所示

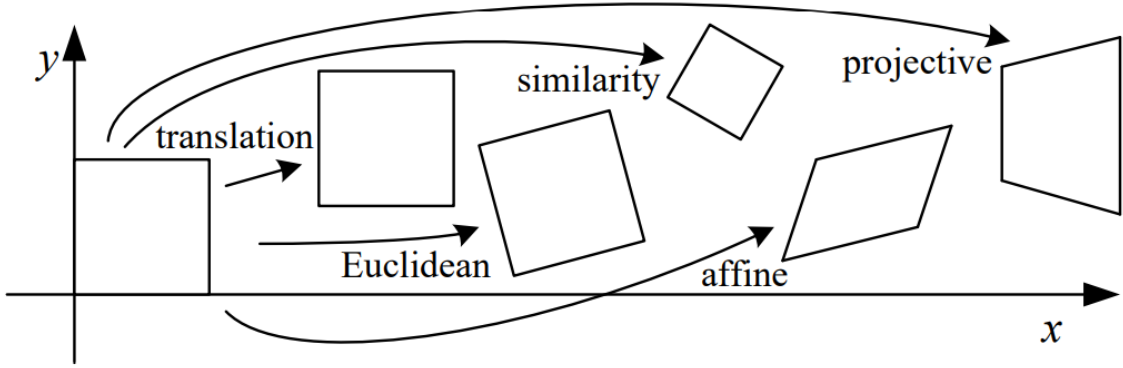


图 2-4 2D 平面变换的基本集合。

**变换:** 二维变换可以写成  $x' = x + t$  或者

$$x' = \begin{bmatrix} I & t \end{bmatrix} \bar{x} \quad (2.14)$$

$I$  是  $(2 \times 2)$  单位矩阵或

$$\bar{x}' = \begin{bmatrix} I & t \\ 0^T & 1 \end{bmatrix} \bar{x} \quad (2.15)$$

其中  $0$  是零向量。使用  $2 \times 3$  矩阵可得到更紧凑的表示法, 而使用全秩  $3 \times 3$  矩阵 (通过附加  $\begin{bmatrix} 0^T & 1 \end{bmatrix}$  行可以从  $2 \times 3$  矩阵中获得) 使得使用矩阵乘法和计算逆变换成为可能。注意, 在任何两侧出现增强向量 (如  $\bar{x}$ ) 的方程中, 它始终可以用全齐次向量  $\tilde{x}$  替换。

**旋转 + 平移:** 这种变换也称为二维刚体运动或二维欧几里德变换 (因为欧几里德距离被保留)。它可以写成  $x' = Rx + t$  或

$$x' = \begin{bmatrix} R & t \end{bmatrix} \bar{x} \quad (2.16)$$

当

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (2.17)$$

是一个正交旋转矩阵，此时， $RR^T = I$ 、 $|R| = 1$ 。

**缩放旋转**: 也称为相似性变换，这种变换可以表示为  $x' = sRx + t$ ，其中  $s$  是任意比例因子。也可以写成

$$x' = \begin{bmatrix} sR & t \end{bmatrix} \bar{x} = \begin{bmatrix} a & -b & t_x \\ b & a & t_y \end{bmatrix} \bar{x}, \quad (2.18)$$

我们不再要求  $a^2 + b^2 = 1$ 。相似性变换保持直线之间的角度。

**仿射**: 仿射变换可以写做  $x' = A\bar{x}$ ， $A$  是任意的  $2 \times 3$  的矩阵，即：

$$x' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \bar{x}. \quad (2.19)$$

在仿射变换下，平行线保持平行。

**投影**: 这种转换，也称为透视变换或同伦变换，在齐次坐标上运算，

$$\tilde{x}' = \tilde{H}\tilde{x}, \quad (2.20)$$

式中  $\tilde{H}$ ，是任意  $3 \times 3$  矩阵。请注意， $\tilde{H}$  是齐次的，即它仅定义为一个标度，两个仅按标度不同的  $\tilde{H}$  矩阵是等价的。为了得到不均匀的结果  $x$ ，必须对得到的齐次坐标进行归一化，即：

$$x' = \frac{h_{00} + h_{01}y + h_{02}}{h_{20} + h_{21}y + h_{22}} \text{ 和 } y' = \frac{h_{10} + h_{11}y + h_{12}}{h_{20} + h_{21}y + h_{22}}. \quad (2.21)$$

透视变换保留直线（即，变换后保持直线）。

**二维变换的层次结构**: 前面的一组转换如图 2.4 所示，并在表 2.1 中进行了总结。最简单的方法是将它们看作一组（可能受限的） $3 \times 3$  矩阵，它们在二维齐次坐标向量上运行。《Hartley and Zisserman (2004)》包含二维平面变换层次结构的更详细描述。

上述转换形成一组嵌套的组，即它们在组合下闭合，并且具有同一组成员的逆。（在以后将这些转换应用于第 3.6 节中的图像时，这将非常重要。）每个（更简单）组是下面更复杂组的子集。在最近的一些机器人学教程中讨论了此类李群及其相关代数（原点处的切线空间）的数学，例如《Dellaert

表 2-1 2D 坐标变换的层次结构, 列出变换名称、矩阵形式、自由度、保留的几何属性以及助记符图标。每个变换还保留其下各行中列出的属性, 即相似性不仅保留角度, 还保留平行度和直线。 $2 \times 3$  矩阵用第三行扩展, 形成一个完整的  $3 \times 3$  矩阵, 用于齐次坐标变换。

转化	矩阵	自由度	保存	图标
1 翻译	$\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	2	方向	
刚性 (欧几里得)	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	3	长度	
相似点	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	4	角度	
姻亲	$[\mathbf{A}]_{2 \times 3}$	6	平行	
投影变换	$[\tilde{\mathbf{H}}]_{3 \times 3}$	8	直线	

and Kaess 2017; Blanco 2019; Solà, Deray, and Atchuthan 2019》, 其中二维旋转和刚性变换称为  $\text{SO}(2)$  和  $\text{SE}(2)$ , 表示特殊的正交群和特殊的欧氏群。<sup>①</sup>

**同向矢量:** 虽然上述变换可以用于变换二维平面中的点, 但它们是否也可以直接用于变换直线方程? 考虑齐次方程  $\tilde{\mathbf{x}} \cdot \tilde{\mathbf{l}} = 0$ 。如果我们变换  $\mathbf{x}' = \tilde{\mathbf{H}}\mathbf{x}$ , 我们得到

$$\tilde{\mathbf{l}} \cdot \tilde{\mathbf{x}}' = \tilde{\mathbf{l}}^T \tilde{\mathbf{H}} \tilde{\mathbf{x}} = \left( \tilde{\mathbf{H}}^T \tilde{\mathbf{l}} \right)^T \tilde{\mathbf{x}} = \tilde{\mathbf{l}} \cdot \tilde{\mathbf{x}} = 0, \quad (2.22)$$

也就是,  $\tilde{\mathbf{l}}' = \tilde{\mathbf{H}} -^T \tilde{\mathbf{l}}$ 。因此, 射影变换对二维直线或三维法线等共向量的作用可以用矩阵的转置逆来表示, 因为射影变换矩阵是齐次的, 因此矩阵的转置逆相当于  $\tilde{\mathbf{H}}$  的伴随。吉姆·布林 (1998) 描述了 (在第 9 章和第 10 章) 标注和操纵共向量的来龙去脉。

虽然上面的转换是我们使用最广泛的转换, 但有时会使用一些额外的转换。**挤压和拉伸:** 这种变换改变了图像的纵横比,

$$\begin{aligned} x' &= s_x x + t_x \\ y' &= s_y y + t_y, \end{aligned}$$

上式是仿射变换的一种限制形式。不幸的是, 它没有与表 2.1 中列出的组进行准确的嵌套。

<sup>①</sup> 术语“特殊”是指不反射的期望条件, 即  $\det[R] = 1$ 。

122 平面流: 这个八参数变换 (Horn 1986; Bergen, Anandan et al. 1992; Girod, Greiner, and Niemann 2000)

$$x' = a_0 + a_1x + a_2y + a_6x^2 + axy$$

$$y' = a_3 + a_4x + a_5y + ax^2 + a_6xy,$$

123 当平面经历小的三维运动时出现。因此, 它可以被认为是一个小的运动近似到全同态。它的主要吸  
124 引力在于运动参数  $a_k$  是线性的, 这通常是估计的量。

125 双线性插值: 八参数变换 ( Wolberg 1990 ),

$$x' = a_0 + a_1x + a_2y + a_6xy$$

$$y' = a_3 + a_4x + a_5y + axy,$$

126 可用于插值由于正方形的四个角点的运动而产生的变形。(事实上, 它可以插值任何四个非共线点  
127 的运动。)虽然变形在运动参数中是线性的, 但它通常不保留直线(仅保留平行于方轴的直线)。但  
128 是, 它通常非常有用, 例如, 在使用样条插值稀疏网格时(第 9.2.2 节)。

### 129 2.1.2 3D 转换

130 用于插值三维坐标变换的集合与二维变换的集合非常相似, 并在表 2.2 中总结。与 2D 一样, 这  
131 些转换形成一组嵌套的组。哈特利和齐瑟曼(2004 年, 第 2.4 节)对这一层次结构作了更详细的描  
132 述。

133 平移. 三维变换可以被写作  $X' = X + t$  或

$$\mathbf{x}' = [\mathbf{I}t]\bar{\mathbf{x}} \quad (2.23)$$

134 其中  $\mathbf{I}$  是 (3x3) 恒等矩阵。

135 旋转 + 平移. 也称为三维刚体运动或三维欧氏变换或 SE (3), 它可以写成  $X' = RX + t$  或

$$\mathbf{x}' = [\mathbf{R} \quad \mathbf{t}]\bar{\mathbf{x}} \quad (2.24)$$

136 其中  $\mathbf{R}$  是一个  $3 \times 3$  正交旋转矩阵,  $\mathbf{R}\mathbf{R}^T = \mathbf{I}$  且  $|\mathbf{R}| = 1$ 。请注意, 有时使用来描述刚性运动更  
137 方便。

$$x' = R(x - c) = Rx - Rc, \quad (2.25)$$

138 其中  $\mathbf{c}$  是旋转中心(通常是摄像机中心)。紧凑地参数化三维旋转是一项非常重要的任务, 我们将在  
139 下面详细描述。

140 缩放旋转. 三维相似变换可以表示为  $\mathbf{x}' = s\mathbf{R}\mathbf{x} + \mathbf{t}$ , 其中  $s$  是任意尺度因子。它也可以写成

$$\mathbf{x}' = [s\mathbf{R}'\mathbf{t}]\bar{\mathbf{x}}. \quad (2.26)$$

表 2-2 3D 坐标转换的层次结构。每个变换还保留其下各行中列出的属性，即相似性不仅保留角度，还保留平行度和直线。3×4 矩阵被扩展为第四行，以形成用于齐次坐标变换的完整的 4×4 矩阵。助记符图标是在 2D 绘制的，但意在暗示在一个完整的 3D 立方体中发生的转换。

转化	矩阵	自由度	保存	图标
翻译	$\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{3 \times 4}$	3	方向	
刚性（欧几里得）	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{3 \times 4}$	6	长度	
相似点	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{3 \times 4}$	7	角度	
姻亲	$[\mathbf{A}]_{3 \times 4}$	12	平行	
投影变换	$[\tilde{\mathbf{H}}]_{4 \times 4}$	15	直线	

141 此变换保留直线和平面之间的角度。

142 **仿射**. 仿射变换被写成  $\mathbf{x}' = \mathbf{A}\bar{\mathbf{x}}$ ,  $\mathbf{A}$  是任意的 3×4 矩阵, 即

$$\mathbf{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{bmatrix} \bar{\mathbf{x}}. \quad (2.27)$$

143 在仿射变换下，平行线和平面保持平行。

144 **投影**. 这种变换，也称为三维透视变换、同调或共线变换，是在齐次坐标上进行的，

$$\hat{\mathbf{x}}' = \tilde{\mathbf{H}}\hat{\mathbf{x}}, \quad (2.28)$$

145 其中  $\tilde{\mathbf{H}}$  是任意 4×4 齐次矩阵。在 2D 中，为了获得不均匀的结果，必须对得到的齐次坐标  $\mathbf{x}$  进行归  
146 一化。透视变换保留直线（即，变换后它们保持直线）。

### 147 2.1.3 3D 旋转

148 二维和三维坐标变换之间的最大区别在于，三维旋转矩阵  $\mathbf{R}$  的参数化不是那么简单，因为存在几种  
149 不同的可能性

150 **欧拉角**

旋转矩阵可以形成为围绕三个基本轴 (例如,  $x$ 、 $y$  和  $z$ , 或  $x$ 、 $y$  和  $x$ ) 的三次旋转的乘积。这通常是一个坏主意, 因为结果取决于应用变换的顺序。<sup>②</sup>更糟糕的是, 在参数空间中不总是可能平滑移动, 即, 有时一个或多个欧拉角响应于旋转的微小变化而急剧变化。<sup>③</sup>由于这些原因, 在这本书里, 我们甚至没有给出欧拉角的公式——感兴趣的读者可以查阅其他教科书或技术报告 (福杰拉斯 1993; Diebel 2006)。请注意, 在某些应用中, 如果已知旋转是一组单轴变换, 则总是可以使用一组明确的刚性变换来表示它们。

### 轴/角度 (指数扭曲)

旋转可以由旋转轴  $\hat{n}$  和角度  $\theta$  表示, 或者等效地由 3D 向量  $\omega = \theta\hat{n}$  表示。图 2.5 显示了我们如何计算等效旋转。首先, 我们将向量  $v$  投影到轴  $\hat{n}$  上, 以获得

$$v_{\parallel} = \hat{n}(\hat{n} \cdot v) = (\hat{n}\hat{n}^T) v. \quad (2.29)$$

不受旋转影响的  $v$  分量。接下来, 我们从  $\hat{n}$  开始计算  $v$  的垂直残差,

$$v_{\perp} = v - v_{\parallel} = (I - \hat{n}\hat{n}^T) v. \quad (2.30)$$

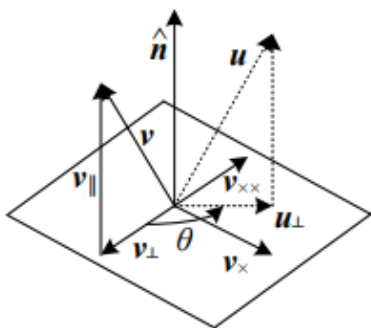


图 2-5 绕轴  $\hat{n}$  旋转角度  $\theta$

我们可以用叉积把这个向量旋转  $90^\circ$ ,

$$v_{\times} = \hat{n} \times v = [\hat{n}]_{\times} v. \quad (2.31)$$

<sup>②</sup>但是, 在特殊情况下, 例如描述云台的运动, 这些角度可能更直观。

<sup>③</sup>在机器人学中, 这有时被称为万向锁。

162 式中,  $[\hat{\mathbf{n}}]_{\times}$  是向量  $\hat{\mathbf{n}} = (\hat{n}_x, \hat{n}_y, \hat{n}_z)$  的叉积运算符的矩阵形式,

$$[\hat{\mathbf{n}}]_{\times} = \begin{bmatrix} 0 & -\hat{n}_z & \hat{n}_y \\ \hat{n}_z & 0 & -\hat{n}_x \\ -\hat{n}_y & \hat{n}_x & 0 \end{bmatrix}. \quad (2.32)$$

163 注意把这个向量再旋转  $90^\circ$  等于再取一次叉积,

$$\mathbf{v}_{\times \times} = \hat{\mathbf{n}} \times \mathbf{v}_{\times} = [\hat{\mathbf{n}}]_{\times}^2 \mathbf{v} = -\mathbf{v}_{\perp}$$

164 因此

$$\mathbf{v}_{\parallel} = \mathbf{v} - \mathbf{v}_{\perp} = \mathbf{v} + \mathbf{v}_{\times \times} = (\mathbf{I} + [\hat{\mathbf{n}}]_{\times}^2) \mathbf{v}$$

165 我们现在可以计算旋转向量  $\mathbf{u}$  的平面内分量

$$u_{\perp} = \cos \theta v_{\perp} + \sin \theta v_{\times} = \left( \sin \theta [\hat{\mathbf{n}}]_{\times} - \cos \theta [\hat{\mathbf{n}}]_{\times}^2 \right) v$$

166 将所有这些项放在一起, 我们得到最终的旋转矢量

$$\mathbf{u} = \mathbf{u}_{\perp} + \mathbf{v}_{\parallel} = (\mathbf{I} + \sin \theta [\hat{\mathbf{n}}]_{\times} + (1 - \cos \theta) [\hat{\mathbf{n}}]_{\times}^2) \mathbf{v} \quad (2.33)$$

167 因此, 我们可以将与围绕轴  $[\hat{\mathbf{n}}]_{\times}$  的  $\theta$  旋转相对应的旋转矩阵写成

$$\mathbf{R}(\hat{\mathbf{n}}, \theta) = \mathbf{I} + \sin \theta [\hat{\mathbf{n}}]_{\times} + (1 - \cos \theta) [\hat{\mathbf{n}}]_{\times}^2 \quad (2.34)$$

168 这就是罗德里格斯公式 (Ayache 1989)。轴  $[\hat{\mathbf{n}}]_{\times}$  和角度  $\theta$  的乘积,  $\boldsymbol{\omega} = \theta \hat{\mathbf{n}} = (\omega_x, \omega_y, \omega_z)$ , 是三维旋  
169 转的最小表示。如果  $\theta$  以度为单位存储, 则通过  $90^\circ$  倍数等公共角度的旋转可以精确表示 (并转换  
170 为精确矩阵)。不幸的是, 这种表示法不是唯一的, 因为我们总是可以将  $360^\circ$  的倍数 ( $2\pi$  弧度) 加  
171 到  $\theta$  上, 得到相同的旋转矩阵。同样,  $(n, \theta)$  和  $(-n, -\theta)$  表示相同的旋转。

172 但是, 对于较小的旋转 (例如, 旋转校正), 这是一个很好的选择。特别是, 对于小的 (无穷小  
173 的或瞬时的) 旋转和用弧度表示的  $\theta$ , 罗德里格斯公式简化为

$$\mathbf{R}(\boldsymbol{\omega}) \approx \mathbf{I} + \sin \theta [\hat{\mathbf{n}}]_{\times} \approx \mathbf{I} + [\theta \hat{\mathbf{n}}]_{\times} = \begin{bmatrix} 1 & -\omega_z & \omega_y \\ \omega_z & 1 & -\omega_x \\ -\omega_y & \omega_x & 1 \end{bmatrix} \quad (2.35)$$

174 它给出了旋转参数  $\boldsymbol{\omega}$  和  $\mathbf{R}$  之间很好的线性关系。我们也可以写出  $\mathbf{R}(\boldsymbol{\omega})\mathbf{v} \approx \mathbf{v} + \boldsymbol{\omega} \times \mathbf{v}$ , 当我们想计  
175 算  $\mathbf{R}\mathbf{v}$  相对于  $\boldsymbol{\omega}$  的导数时,

$$\frac{\partial \mathbf{R}\mathbf{v}}{\partial \boldsymbol{\omega}^T} = -[\mathbf{v}]_{\times} = \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix} \quad (2.36)$$



另一种通过有限角度导出旋转的方法称为指数扭曲 (Murray、Li 和 Sastry 1994)。一个角度  $\theta$  的旋转等价于通过  $\theta/k$  的  $k$  次旋转, 在  $k \rightarrow \infty$  的极限下, 我们得到

$$R(\hat{n}, \theta) = \lim_{k \rightarrow \infty} \left( I + \frac{1}{k} [\theta \hat{n}]_{\times} \right)^k = \exp[\omega]_{\times} \quad (2.37)$$

如果我们将矩阵指数展开为泰勒级数 (使用恒等式,  $[\hat{n}]_{\times}^{k+2} = -[\hat{n}]_{\times}^k$ , 并再次假设  $\theta$  以弧度表示),

$$\begin{aligned} \exp[\omega]_{\times} &= I + \theta [\hat{n}]_{\times} + \frac{\theta^2}{2} [\hat{n}]_{\times}^2 + \frac{\theta^2}{2} [\hat{n}]_{\times}^3 + \cdots \\ &= I + \left( \theta - \frac{\theta^2}{3!} + \cdots \right) [\hat{n}]_{\times} + \left( \frac{\theta^2}{2} - \frac{\theta^2}{4!} + \cdots \right) [\hat{n}]_{\times}^2 \\ &= I + \sin \theta [\hat{n}]_{\times} + (1 - \cos \theta) [\hat{n}]_{\times}^2, \end{aligned} \quad (2.38)$$

这就产生了熟悉的罗德里格斯公式。

在机器人学 (和群论), 旋转称为  $SO(3)$ , 即 3D 中的特殊正交群。增量旋转  $\omega$  并用相应的李代数  $\mathfrak{se}(3)$  识别, 并且是制定旋转导数和对旋转估计中的不确定性建模的优选方式 (Blanco 2019 Solà、Deray 和 Atchuthan(2019))。

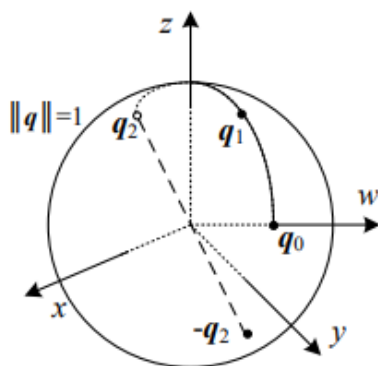


图 2-6 单位四元数活在单位球面  $\|\mathbf{q}\| = 1$  上。该图显示了三个四元数  $q_0, q_1$  和  $q_2$  的平滑轨迹。 $q_2$  的对跖点, 即  $-q_2$ , 代表与  $q_2$  相同的旋转。

**单位四元数** 单位四元数表示与角度/轴表示密切相关。单位四元数是单位长度的 4 矢量, 其分量可以写成  $\mathbf{q} = (q_x, q_y, q_z, q_w)$  或简称  $\mathbf{q} = (x, y, z, w)$ 。单位四元数位于单位球面  $\|\mathbf{q}\| = 1$  上, 对跖 (符号相反) 四元数  $\mathbf{q}$  和  $-\mathbf{q}$  代表相同的旋转 (图 2.6)。除了这种模糊性 (对偶覆盖), 旋转的单位四元数表示是唯一的。此外, 表示是连续的, 即当旋转矩阵连续变化时, 您可以找到连续的四元数表示, 尽管四元数球面上的路径在返回到“原点”  $\mathbf{q}_0 = (0, 0, 0, 1)$  之前可能会一直环绕。由于下面给出的这

189 些和其他原因, 四元数是计算机图形学中姿态和姿态插值的一种非常流行的表示方法。(Shoemake  
190 1985)。

191 四元数可以通过以下公式从轴/角度表示中导出

$$\mathbf{q} = (\mathbf{v}, \omega) = \left( \sin \frac{\theta}{2} \hat{\mathbf{n}}, \cos \frac{\theta}{2} \right) \quad (2.39)$$

192 其中  $\hat{\mathbf{n}}$  和  $\theta$  是旋转轴和角度。利用三角恒等式  $\sin \theta = 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2}$  and  $(1 - \cos \theta) = 2 \sin^2 \frac{\theta}{2}$  和  
193  $(1 + \cos \theta) = 2 \cos^2 \frac{\theta}{2}$ , 罗德里格斯公式可转换为

$$\begin{aligned} \mathbf{R}(\hat{\mathbf{n}}, \theta) &= \mathbf{I} + \sin \theta [\hat{\mathbf{n}}]_{\times} + (1 - \cos \theta) [\hat{\mathbf{n}}]_{\times}^2 \\ &= \mathbf{I} + 2 \sin \frac{\theta}{2} [\hat{\mathbf{n}}]_{\times} + 2 \cos^2 \frac{\theta}{2} [\hat{\mathbf{n}}]_{\times}^2 \end{aligned} \quad (2.40)$$

194 这建议了一个快速的方法, 通过一个四元数旋转一个向量  $\mathbf{v}$ , 使用一系列的叉积, 缩放和加法。为  
195 了得到  $\mathbf{R}(\mathbf{q})$  作为  $(x, y, z, w)$  的函数的公式, 回想一下

$$[\mathbf{v}]_{\times} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \text{ 和 } [\mathbf{v}]_{\times}^2 = \begin{bmatrix} -y^2 - z^2 & xy & xz \\ xy & -x^2 - z^2 & yz \\ xz & yz & -x^2 - y^2 \end{bmatrix}.$$

196 我们因此获得

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} 1 - 2(y^2 + z^2) & 2(xy - zw) & 2(xz + yw) \\ 2(xy + zw) & 1 - 2(x^2 + z^2) & 2(yz - xw) \\ 2(xz - yw) & 2(yz + xw) & 1 - 2(x^2 + y^2) \end{bmatrix}. \quad (2.41)$$

197 通过将  $1 - 2(y^2 + z^2)$  替换为  $(x^2 + w^2 - y^2 - z^2)$  等, 可以使对角线项更加对称。

198 单位四元数最好的方面是有一个简单的代数来组成表示为单位四元数的变量。给定两个四元数  
199  $\mathbf{q}_0 = (\mathbf{v}_0, w_0)$  和  $\mathbf{q}_1 = (\mathbf{v}_1, w_1)$ , 四元数乘法运算符定义为

$$\mathbf{q}_2 = \mathbf{q}_0 \mathbf{q}_1 = (\mathbf{v}_0 \times \mathbf{v}_1 + \omega_0 \mathbf{v}_1 + \omega_1 \mathbf{v}_0, \omega_0 \omega_1 - \mathbf{v}_0 \cdot \mathbf{v}_1), \quad (2.42)$$

200 性质为  $R(q_2) = R(q_0) R(q_1)$ 。请注意, 四元数乘法是不可交换的, 就像三维旋转和矩阵乘法是不  
201 可交换的一样。

202 求四元数的逆很容易: 只需翻转或的符号 (但不能两者都翻转!)。(您可以验证这是否具有在  
203 (2.41) 中转置矩阵的预期效果。)因此, 我们还可以将四元数除法定义为

$$\mathbf{q}_2 = \mathbf{q}_0 / \mathbf{q}_1 = \mathbf{q}_0 \mathbf{q}_1^{-1} = (\mathbf{v}_0 \times \mathbf{v}_1 + \omega_0 \mathbf{v}_1 - \omega_1 \mathbf{v}_0, -\omega_0 \omega_1 - \mathbf{v}_0 \cdot \mathbf{v}_1) \quad (2.43)$$

204 当需要两个旋转之间的增量旋转时, 这很有用。

特别是，如果我们想确定一个旋转是两个给定旋转之间的一部分，我们可以计算增量旋转，取一小部分角度，然后计算新的旋转。这个过程被称为球面线性插值或简称 *slerp* (Shoemake 1985)，在算法 2.1 中给出。请注意，Shoemake 提供了两个公式，而不是这里给出的公式。第一个将  $q$  乘以  $\alpha$ ，然后再乘以原始四元数，

$$q_2 = q_r^\alpha q_0 \quad (2.44)$$

而第二种方法将四元数视为球体上的 4 个向量并使用四元数，

$$q_2 = \frac{\sin(1-\alpha)\theta}{\sin\theta} q_0 + \frac{\sin\alpha\theta}{\sin\theta} q_1 \quad (2.45)$$

其中  $\theta = \cos^{-1}(q_0 \cdot q_1)$ ，点积直接位于四元数 4 矢量之间。所有这些公式都给出了可比较的结果，尽管当  $q_0$  和  $q_1$  接近时应该小心，这就是为什么我更喜欢使用反正切来确定旋转角度。

**procedure *slerp*( $q_0, q_1, \alpha$ ):**

1.  $q_r = q_1/q_0 = (\mathbf{v}_r, w_r)$

2. if  $w_r < 0$  then  $q_r \leftarrow -q_r$

3.  $\theta_r = 2 \tan^{-1}(\|\mathbf{v}_r\|/w_r)$

4.  $\hat{\mathbf{n}}_r = \mathcal{N}(\mathbf{v}_r) = \mathbf{v}_r/\|\mathbf{v}_r\|$

5.  $\theta_\alpha = \alpha \theta_r$

6.  $q_\alpha = (\sin \frac{\theta_\alpha}{2} \hat{\mathbf{n}}_r, \cos \frac{\theta_\alpha}{2})$

7. **return**  $q_2 = q_\alpha q_0$

**算法 2.1** 球面线性插值 (*slerp*)。首先根据四元数比率计算轴和总角度。(该计算可以在内部循环之外进行，内部循环为动画生成一组插值位置。) 然后计算增量四元数，并乘以起始旋转四元数。

哪种旋转表现法比较好？

3D 旋转的表示形式的选择部分取决于应用程序。

轴/角度表示最小，因此不需要对参数进行任何其他约束（每次更新后无需重新规范化）。如果角度以度表示，则更容易理解姿势（例如，绕  $x$  轴旋转  $90^\circ$ ），也更容易表达精确的旋转。当角度为弧度时，可以很容易地计算出  $R$  相对于  $\omega$  的导数 (2.36)。

另一方面，四元数可以更好地表示平滑移动的相机的轨迹，因为表示形式中没有间断。在旋转之间进行插值和链接刚性变换也更容易 (Murray, Li 和 Sastry 1994; Bregler 和 Malik 1998)。

我通常的偏好是使用四元数，但是使用增量旋转更新四元数，如第 11.2.2 节中所述。

## 2.1.4 3D 到 2D 投影

现在，我们知道了如何表示 2D 和 3D 几何图元以及如何在空间上进行变换，我们需要指定如何将 3D 图元投影到图像平面上。我们可以使用线性 3D 到 2D 投影矩阵来做到这一点。最简单的模型是拼字法，不需要进行划分即可获得最终（不均匀的）结果。更为常用的模型是透视图，因为它可以更准确地模拟真实相机的行为。

**正字法和准透视**正交投影只是将三维坐标的  $z$  分量放下以获得 2D 点  $x$ 。（在本节中，我们用  $p$  表示 3D 点，用  $x$  表示 2D 点。）这可以写成

$$x = [I_{2 \times 2} \mid 0] p \quad (2.46)$$

如果我们使用齐次（投影）坐标，则可以编写

$$\tilde{x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{p} \quad (2.47)$$

即，我们删除  $z$  分量，但保留  $w$  分量。正交法是长焦距（远摄）镜头和深度相对于距相机的距离较浅的物体的近似模型 (Sawhney 和 Hanson 1991)。它仅适用于远心镜头 (Baker and Nayar 1999, 2001)。

实际上，世界坐标（可能以米为单位测量尺寸）需要缩放以适合图像传感器（物理上以毫米为单位，但最终以像素为单位）。因此，比例正字法实际上更常用，

$$x = [sI_2 \times 2 \mid 0] p \quad (2.48)$$

该模型等效于首先将世界点投影到本地正面平行图像平面，然后使用常规透视图缩放该图像。对于场景的所有部分，缩放比例可以相同（图 2.7b），对于独立建模的对象，缩放比例可以不同（图 2.7c）。更重要的是，从运动估计结构时，缩放比例会因帧而异，从而可以更好地模拟物体接近摄像机时发生的缩放比例变化。

比例尺描记术是一种受欢迎的模型，用于重建远离相机的物体的 3D 形状，因为它极大地简化了某些计算。例如，可以使用最小二乘法（第 11.2.1 节）刺激姿势（相机方向）。如第 11.4.1 节 (Tomasi 和 Kanade 1992) 所讨论的，可以使用分解（奇异值分解）同时估计矫正术，结构和运动。

密切相关的投影模型是准透视的 (Aloimonos, 1990; Poelman 和 Kanade, 1997)。在该模型中，对象点再次被投影到平行于图像平面的局部参考上。但是，不如与其垂直于该平面投影，它们被投影到与物体中心视线平行的位置（图 2.7d）。随后是通常投影到最终像面上的投影，这同样具有可伸缩性。因此，这两个投影的组合是可行的，可以写成

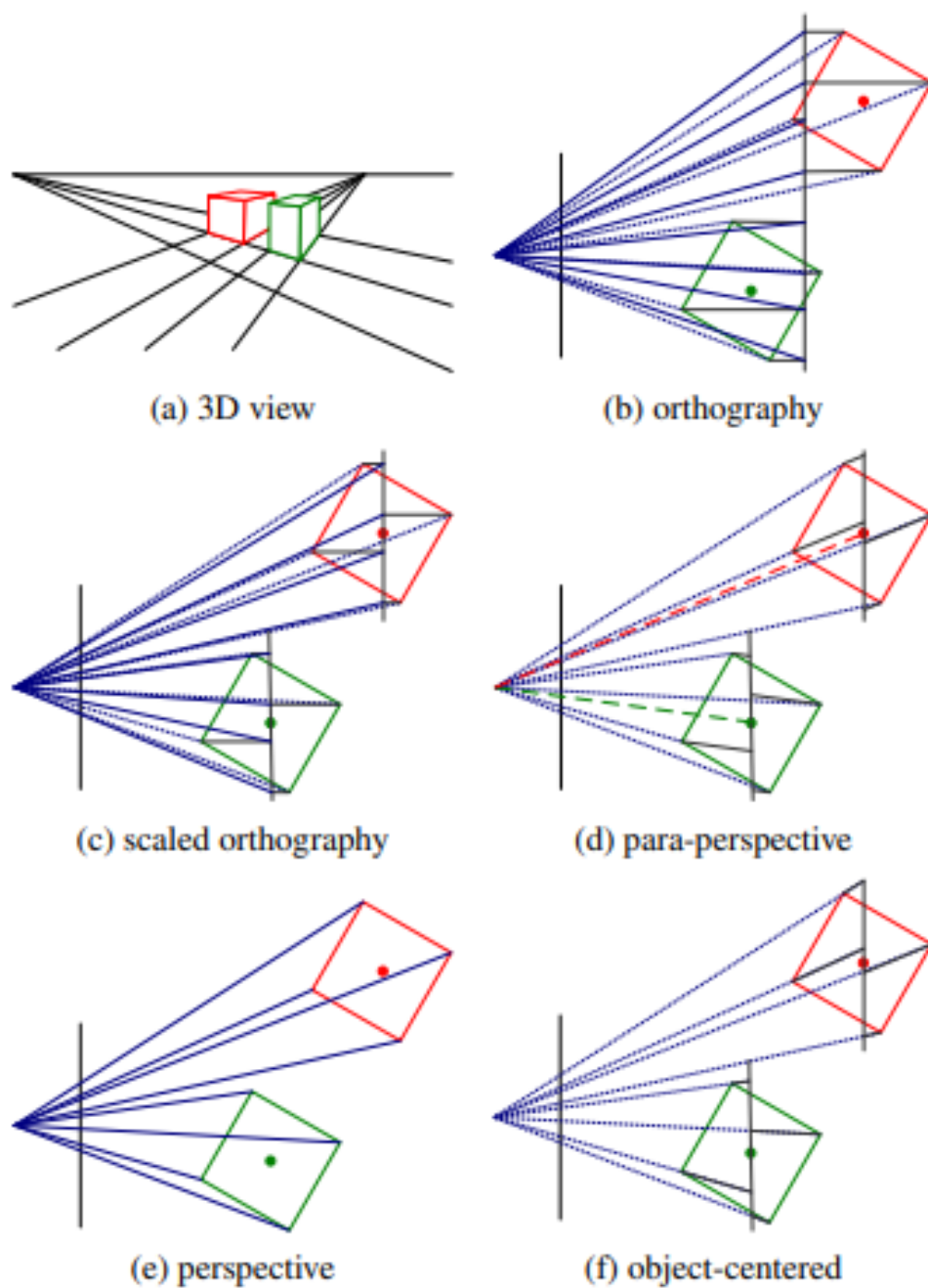


图 2-7 常用的投影模型:(a) 三维世界视图, (b) 正字法, (c) 比例正字法, (d) 准透视, (e) 透视, (f) 以物体为中心。每个图都显示了投影的俯视图。请注意, 在非透视投影中, 地平面和盒侧的平行线是如何保持平行的。

$$\tilde{x} = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{p} \quad (2.49)$$

247 请注意, 图 2.7b-d 中的 3D 平行线在投影后如何保持平行。平行透视提供了比比例正射投影法  
248 更准确的投影模型, 而不会增加逐像素透视划分的复杂性, 这使传统的因式分解方法无效 (Poelman  
249 和 Kanade 1997)。

## 250 看法

251 在计算机图形和计算机视觉中最常用的投影是真正的 3D 透视图 (图 2.7e)。在这里, 通过将点除以  
252 z 分量将它们投影到图像平面上。使用不均匀的坐标, 可以写成

$$\tilde{x} = P_Z(p) \begin{bmatrix} x/y \\ y/z \\ 1 \end{bmatrix} \tilde{p} \quad (2.50)$$

253 在齐次坐标中, 投影具有简单的线性形式,

$$\tilde{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{p} \quad (2.51)$$

254 即, 我们删除 p 的 w 分量。因此, 在投影之后, 不可能从图像恢复 3D 点的距离, 这对于 2D 成像传  
255 感器是有意义的。

256 计算机图形学系统的一种形式, 即两步投影, 该投影首先将 3D 坐标投影到范围  $(x, y, z) \in [-1, 1] \times$   
257  $[-1, 1] \times [0, 1]$  的标准化设备坐标中, 然后使用视口变换将这些坐标重新缩放为整数像素坐标 (Watt  
258 1995; OpenGL -ARB 1997)。然后, 使用  $4 \times 4$  矩阵表示 (初始) 透视投影

$$\tilde{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -z_{\text{far}}/z_{\text{range}} & z_{\text{near}} z_{\text{far}}/z_{\text{range}} \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{p} \quad (2.52)$$

259 其中 Znear 和 Zfar 是近 z 裁剪平面和远 z 裁剪平面,  $z_{\text{range}} = z_{\text{far}} - z_{\text{near}}$  请注意, 实际上前两行  
260 实际上是通过焦距和纵横比进行缩放的, 以便将可见光映射到  $(x, y, z) \in [-1, 1]^2$ 。保留第三行而不  
261 是删除第三行的原因是可见性操作 (例如 z 缓冲) 要求每个呈现的图形元素都具有一定的深度。

262 如果我们将  $z_{\text{range}}$  设置为 1,  $z_{\text{near}} = 1, z_{\text{far}} \rightarrow \infty$ , 并切换第三行的符号, 则归一化屏幕矢量的  
263 第三个元素将成为反深度, 即视差 (Okutomi 和 Kanade 1993)。因为在户外移动相机, 这在许多情况

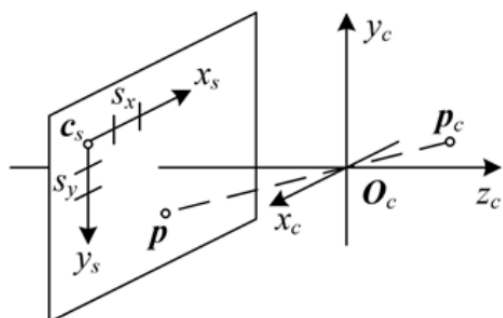


图 2-8 以 3D 相机为中心的点  $p_c$  在位置  $p$  上投影到传感器平面上的情况。 $O_c$  是光学中心（节点）， $C_s$  是传感器平面坐标系的 3D 原点，而  $s_x$  和  $s_y$  是像素间距。

下可能是方便的，所以相对于直接 3D 距离，相机的反深度通常是条件更好的参数化设置。尽管常规的 2D 图像传感器无法测量到表面点的距离，但是距离传感器（第 13.2 节）和立体匹配算法（第 12 章）可以计算出这些值。这样便可以使用  $4 \times 4$  矩阵的逆函数将基于传感器的深度或视差值  $d$  直接映射回 3D 位置（第 2.1.4 节）。如果我们使用（2.64）的  $4 \times 4$  满级矩阵表示透视投影，就可以做到这一点。

### 相机原理

一旦使用投影矩阵通过理想的针孔投影了 3D 点，我们仍然必须根据像素传感器间距和传感器平面相对于原点的相对位置来转换所得坐标。图 2.8 显示了所涉及的几何形状。在本节中，我们首先介绍使用传感器单应  $M_s$  从 2D 像素坐标到 3D 射线的映射，因为这在物理可测量的数量上比较容易解释。然后，我们将其与更常用的相机本征矩阵  $K$  相关联，该矩阵用于将 3D 相机中心点  $p_c$  映射到 2D 像素坐标  $\tilde{X}_s$ 。

图像传感器返回以整数像素坐标  $(x_s, y_s)$  索引的像素值，这些坐标通常从图像的左上角开始并向下和向右移动。（并非所有成像库都遵守该约定，而是直接调整其他坐标系。）将像素中心映射到 3D 坐标，首先按像素间距  $(s_x, s_y)$ （有时以微米表示，对于固态传感器）缩放  $(x_s, y_s)$  值，并相对于摄像头相对于相机的投影方向进行描述。中心  $O_c$ ，具有原点  $c_s$  和 3D 旋转  $R_s$ （图 2.8）。

然后将组合的 2D 到 3D 投影写为

$$p = [R_s c_s] \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = M_s \bar{x}_s \quad (2.53)$$

第三个  $3 \times 3$  矩阵的  $M$  列是沿着  $X_s$  和  $Y_s$  方向对应于图像像素阵列中单位步长的 3D 向量，而第三列是 3D 图像阵列原点  $C_s$ 。矩阵  $M_s$  由八个未知数参数化：三个参数描述旋转  $R_s$ ，三个参数描述平



移  $\mathbf{c}_s$ , 以及两个比例因子 ( $s_x, s_y$ )。请注意, 由于固态制造技术使这一点可以忽略不计, 因此在这里我们忽略了图像平面上两个轴之间偏斜的可能性。在实践中, 除非我们对传感器间距或传感器方向有准确的外部知识, 否则只有七个自由度, 因为不能仅基于外部图像测量就将传感器与原点的距离与传感器间距分开。

但是, 估计具有七个自由度的相机模型  $\mathbf{M}_s$  是不切实际的 (即, 经过适当的重新缩放后, 前两列是正交的), 因此大多数从业者都假定采用一般的  $3 \times 3$  齐次矩阵形式。

像素中心  $\mathbf{p}$  和 3D 相机中心点  $\mathbf{p}_c$  之间的关系由未知比例  $s$  给出,  $\mathbf{p} = s\mathbf{p}_c$ 。因此, 我们可以将  $\mathbf{p}_c$  和像素地址  $\tilde{x}_s$  的同类版本之间的完整投影写为

$$\tilde{x}_s = aM_s^{-1}p_c = Kp_c \quad (2.54)$$

$3 \times 3$  矩阵  $\mathbf{K}$  称为校准矩阵, 它描述了相机的本征 (与相机在空间中的方向相反, 称为外在性)。

从以上讨论中我们可以看出,  $\mathbf{K}$  在实践中具有七个自由度和八个自由度 ( $3 \times 3$  齐次矩阵的全维)。那么, 为什么大多数有关 3D 计算机视觉和多视图几何的教科书 (Faugeras 1993; Hartley and Zisserman 2004; Faugeras and Luong 2001) 将  $\mathbf{K}$  作为具有五个自由度的上三角矩阵进行处理? 尽管在这些

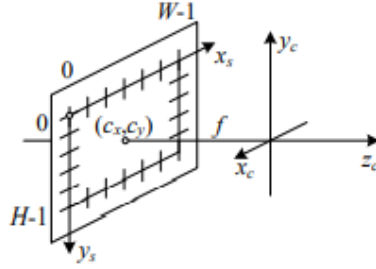


图 2-9 简化的相机内部特性显示了焦距  $f$  和图像中心  $(c_x, c_y)$ 。图像的宽度和高度分别为  $W$  和  $H$ 。

书中通常没有明确指出这一点, 但这是因为我们不能仅基于外部测量来恢复完整的  $\mathbf{K}$  矩阵。当基于外部 3D 点或其他测量值 (Tsai 1987) 校准摄像机 (第 11.1 节) 时, 我们最终同时使用一系列测量值来估计本征 ( $\mathbf{K}$ ) 和外在 ( $\mathbf{R}, \mathbf{t}$ ) 摄像机参数,

$$\tilde{x} = K[Rt]p_w = Pp_w \quad (2.55)$$

其中  $\mathbf{p}_w$  是已知的 3D 世界坐标, 并且

$$P = K[R | t] \quad (2.56)$$

被称为相机矩阵。查看此方程式, 我们看到可以将  $\mathbf{K}$  乘以  $\mathbf{R}_1$ , 然后将  $[\mathbf{R} | \mathbf{t}]$  乘以  $\mathbf{R}_1^T$ , 最后仍然得到有效的校准。因此, 仅凭图像测量就不可能知道传感器的真实方向和真实的相机内部特性。



K 的上三角形式的选择似乎是常规的。给定一个完整的  $3 \times 4$  相机矩阵  $P = K [R|t]$  我们可以使用 QR 分解计算上三角 K 矩阵 (Golub 和 Van Loan 1996)。(请注意术语的冲突：在矩阵代数教科书中, R 表示上三角 (对角线的右边) 矩阵; 在计算机视觉中, R 是正交旋转。)

有多种写 K 的上三角形式的方法。一种可能性是

$$K = \begin{bmatrix} f_x & s & c_s \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.57)$$

它使用独立的焦距  $f_x$  和  $f_y$  表示传感器 x 和 y 的尺寸。由于传感器未垂直于光轴安装, 因此入口 s 编码了传感器轴之间的任何可能的偏斜, 并且  $(c_x, c_y)$  表示以像素坐标表示的图像中心。图像中心在计算机视觉文献中也经常被称为主要点 (Hartley 和 Zisserman 2004), 尽管光学上, 主要点通常是镜头内部主平面与主轴线 (光学轴) 相交的 3D 点 (Hecht, 2015 年)。另一种可能性是

$$K = \begin{bmatrix} f & s & c_x \\ 0 & af & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.58)$$

其中已确定了宽高比 a, 并使用了公共焦距 f。实际上, 对于许多应用, 通过设置  $a = 1$  和  $s = 0$ , 可以获得甚至更简单的形式,

$$K = \begin{bmatrix} \hat{f} & 0 & c_x \\ 0 & \hat{f} & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.59)$$

通常, 将原点设置在图像的大致中心, 例如  $(c_x, c_y) = (W/2, H/2)$ , 其中 W 和 H 是图像的高度和宽度, 可以得到完美可用的具有一个未知数的相机模型, 即焦距 f。

图 2.9 显示了如何将这量可视化作为简化的成像模型的一部分。请注意, 现在我们将像平面放在节点 (透镜的投影中心) 的前面。y 轴的方向也已翻转, 以得到与大多数成像库处理垂直 (行) 坐标的方式兼容的坐标系。

### 关于焦距的一点注记

如何表达焦距的问题是一个在实现计算机视觉算法和讨论其结果时经常引起混乱的问题。这是因为焦距取决于用来测量像素的单位。

如果我们使用整数值对像素坐标进行编号, 例如  $[0, W) \times [0, H)$ , 则焦距 f 和公式 (2.59) 中的相机中心  $(c_x, c_y)$  可以表示为像素值。这些数量与摄影师使用的更熟悉的焦距有何关系?

包括图 2.10 说明了焦距 f 与传感器宽度 W 之间的关系, 并遵循公式

$$\tan \frac{\theta_H}{2} = \frac{W}{2f} \quad \text{或} \quad f = \frac{W}{2} \left[ \tan \frac{\theta_H}{2} \right]^{-1} \quad (2.60)$$

321 对于主动曝光区域  $24\text{mm} \times 36\text{mm}$  我们拥有传统 35mm 胶卷相机  $W = 36\text{mm}$ , 因此  $f$  也用毫米表示。  
 322 ④例如, “备用” 镜头单反相机通常配备 50mm 的长度, 而 85mm 是人像摄影的标准。由于我们处理  
 323 数字图像, 但是, 以像素表示  $W$  更方便, 因此可以使用焦距  $f$  直接在校准矩阵  $K$  中, 如 (2.59) 所  
 324 示。

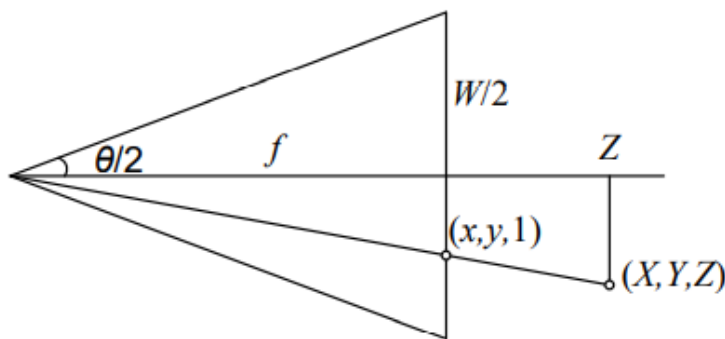


图 2-10 中心投影, 显示 3D 和 2D 坐标  $p$  和  $x$  之间的关系, 以及焦距  $f$ 、图像宽度  $W$  和水平视场  $\theta_H$  之间的关系

325 另一种可能性是缩放像素坐标, 以使它们从  $[-1, 1]$  开始较长的图像尺寸和  $[-a^{-1}, a^{-1}]$  沿  
 326 短轴, 其中  $a \geq 1$  图像宽高比 (与之前介绍的传感器单元宽高比相反)。这可以是使用修改过的归一  
 327 化设备坐标来完成,

$$x'_s = (2x_s - W) / S \text{ 和 } y'_s = (2y_s - H) / S, \quad \text{这里 } S = \max(W, H). \quad (2.61)$$

328 这样做的好处是, 焦距  $f$  和图像中心  $(c_x, c_y)$  变得独立图像分辨率, 这算法在使用多分辨率图像处  
 329 理时很有用, 例如图像金字塔 (第 3.5 节)。⑤使用  $S$  代替  $W$  也会使风景 (水平) 和人像 (垂直) 图  
 330 片的焦距相同, 35 毫米摄影机的保护套。(在某些计算机图形学教科书和系统中, 已归一化设备坐  
 331 标来自  $[-1, 1] \times [-1, 1]$ , 这需要使用两个不同的焦点长度来描述相机的内在特性 (Watt 1995)。在  
 332 (2.60) 中设置  $S = W = 2$ , 我们获得更简单 (无单位) 的关系

$$f^{-1} = \tan \frac{\theta_H}{2} \quad (2.62)$$

333 各种焦距表示之间的转换非常简单, 例如, 从无单位  $f$  到以像素表示的  $f$  乘以  $W / 2$ , 同时从  $f$   
 334 以像素表示, 等于等效的 35mm 焦距, 再乘以 18mm。

④ 435 毫米表示胶片的宽度, 其中 24 毫米用于曝光每一帧, 其余 11 毫米用于表演和帧编号

⑤ 为了在金字塔中的下采样步骤之后使转换真正准确, 必须保持  $W$  和  $H$  的浮点值, 因为如果它们在金字塔中的较大分辨率下是奇数, 它们可能变成非整数。

### 相机矩阵

现在我们已经展示了如何参数化校准矩阵  $\mathbf{K}$ ，我们可以将摄像机内在和外在共同获得单个  $3 \times 4$  相机矩阵

$$\mathbf{P} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \quad (2.63)$$

有时最好使用  $4 \times 4$  可逆矩阵，该矩阵可以通过删除  $\mathbf{P}$  矩阵的最后一行，

$$\tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} = \tilde{\mathbf{K}}\mathbf{E} \quad (2.64)$$

其中  $\mathbf{E}$  是 3D 刚体（欧几里得）变换，而  $\tilde{\mathbf{K}}$  是满级校准矩阵。 $4 \times 4$  相机矩阵  $\tilde{\mathbf{P}}$  可用于直接从 3D 世界坐标映射  $\tilde{\mathbf{P}}_w = (x_w, y_w, z_w, 1)$  到屏幕坐标（加上视差）， $x_s = (x_s, y_s, 1, d)$ ，

$$\mathbf{x}_s \sim \tilde{\mathbf{P}}\tilde{\mathbf{p}}_w \quad (2.65)$$

其中  $\sim$  表示按比例扩大的相等性。注意，在乘以  $\tilde{\mathbf{P}}$  后，向量为除以向量的第三个元素，即可获得归一化形式  $x_s = (x_s, y_s, 1, d)$ 。

### 平面加视差（投影深度）

通常，当使用  $4 \times 4$  矩阵  $\tilde{\mathbf{P}}$  时，我们可以自由地将最后一行重新映射到符合我们目的的一切（而不只是作为差异的“标准”解释，反深度）。让我们将  $\tilde{\mathbf{P}}$  的最后一行重写为  $p_3 = s_3 [\hat{n}_0 | c_0]$ ，其中  $\|\hat{n}_0\| = 1$ 。然后有等式

$$d = \frac{s_3}{z} (\hat{\mathbf{n}}_0 \cdot \mathbf{p}_w + c_0) \quad (2.66)$$

其中  $z = p_2 \cdot P_w = r_z \cdot (p_w - \mathbf{c})$  是  $p_w$  距相机中心  $\mathbf{C}$  的距离 (2.25) 沿光轴  $Z$ （图 2.11）。因此，我们可以将  $d$  解释为射影差异 3D 场景点  $p_w$  相对于参考平面的投影深度  $\hat{n}_0 \cdot p_w + c_0 = 0$ （Szeliski 和 Coughlan, 1997; Szeliski 和 Golland, 1999; Shade, Gortler 等, 1998; Baker, Szeliski 和 Anandan 1998）。（投影深度有时也称为视差使用术语“平面加视差”的重建算法（Kumar, Anandan 和 Hanna 1994 年; (Sawhney 1994)。）设置  $\hat{n}_0 = 0$  且  $c_0 = 1$ ，即将参考平面置于无穷大，会导致更标准的  $d = 1/z$  视差（Okutomi 和 Kanade 1993）。

图 2.11 规则视差（反深度）和投影深度（来自参考平面的视差）。另一种看待这种情况的方法是反转  $\tilde{\mathbf{P}}$  矩阵，以便我们可以映射像素和视差直接回到 3D 点，

$$\tilde{\mathbf{p}}_w = \tilde{\mathbf{P}}^{-1} \mathbf{x}_s \quad (2.67)$$

通常，我们可以选择  $\tilde{\mathbf{P}}$  来具有任何方便的形式，即使用任意投影来采样空间。设置多视图时，这特别方便立体重建算法，因为它允许我们扫描一系列平面（第 12.1.2 节）通过与变量（投影）采样最匹配的图像运动（Collins, 1996; Szeliski 和 Golland, 1999; Saito 和 Kanade, 1999）。

### 从一台相机映射到另一台

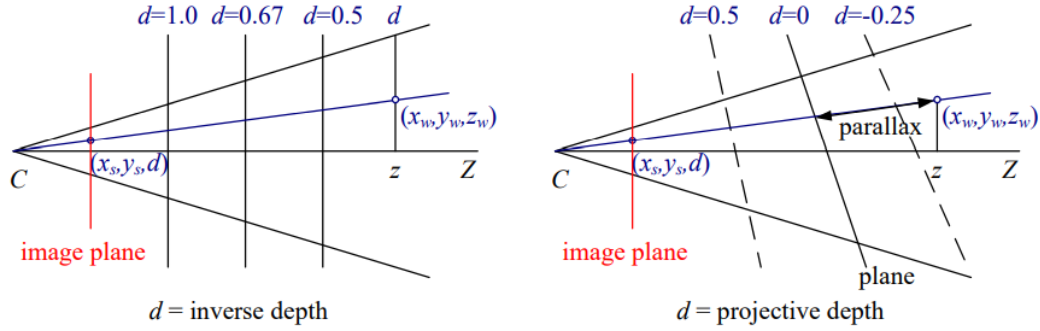


图 2-11 规则视差 (逆深度) 和投影深度 (相对于参考平面的视差)。

359 当从不同的相机位置拍摄 3D 场景的两个图像时会发生什么? 方向 (图 2.12a)? 使用 (2.64) 中  
360 的全等级  $4 \times 4$  摄像机矩阵  $\tilde{P} = \tilde{K}E$ , 我们可以将从世界到屏幕坐标的投影写为

$$\tilde{x}_0 \sim \tilde{K}_0 E_0 p = \tilde{P}_0 p \quad (2.68)$$

361 假设我们知道一个图像中像素的  $z$  缓冲区或视差值  $d_0$ , 我们可以使用以下方法计算 3D 点位置  
362  $p$

$$p \sim E_0^{-1} \tilde{K}_0^{-1} \tilde{x}_0 \quad (2.69)$$

363 然后将其投影到另一个图像中

$$\tilde{x}_1 \sim \tilde{K}_1 E_1 p = \tilde{K}_1 E_1 E_0^{-1} \tilde{K}_0^{-1} \tilde{x}_0 = \tilde{P}_1 \tilde{P}_0^{-1} \tilde{x}_0 = M_{10} \tilde{x}_0 \quad (2.70)$$

364 不幸的是, 我们通常无法正常访问像素的深度坐标摄影图像。但是, 对于平面场景, 如上文  
365 (2.66) 所述, 我们可以 2.1 几何图元和变换

366 图 2.12 将一个点投影到两个图像中: (a) 3D 点坐标  $(X, Y, Z, 1)$  和 2D 投影点  $(x, y, 1, d)$  之间的关系; (b) 诱发平面单应性通过点均位于公共平面上。

368 将 (2.64) 中  $P_0$  的最后一行替换为映射的一般平面方程  $\hat{n}_0 \cdot p + c_0$  点在平面上的  $d_0 = 0$  值 (图  
369 2.12b)。因此, 如果我们将  $d_0$  设为 0, 则可以忽略  $M_{10}$  的最后一列 (2.70) 以及它的最后一行, 因  
370 为我们不在乎最后一个  $Z$  缓冲区的深度。因此, 映射方程式 (2.70) 简化为

$$\tilde{x}_1 \sim \tilde{H}_{10} \tilde{x}_0 \quad (2.71)$$

371 其中  $\tilde{x}_1$  是一般的  $3 \times 3$  单应矩阵,  $\tilde{x}_1$  和  $\tilde{x}_0$  现在是 2D 齐次的坐标 (即 3 个向量) (Szeliski 1996), 这  
372 证明了使用 8 参数均值作为平面场景镶嵌的一般对齐模型 (Mann and Picard 1994; Szeliski (1996))。

373 我们不需要知道深度来执行摄像机间的另一种特殊情况映射是指摄像机正旋转时 (第 8.2.3 节),  
374 即  $t_0 = t_1$  时。在这种情况下, 我们可以写

$$\tilde{x}_1 \sim K_1 R_1 R_0^{-1} K_0^{-1} \tilde{x}_0 = K_1 R_{10} K_0^{-1} \tilde{x}_0 \quad (2.72)$$

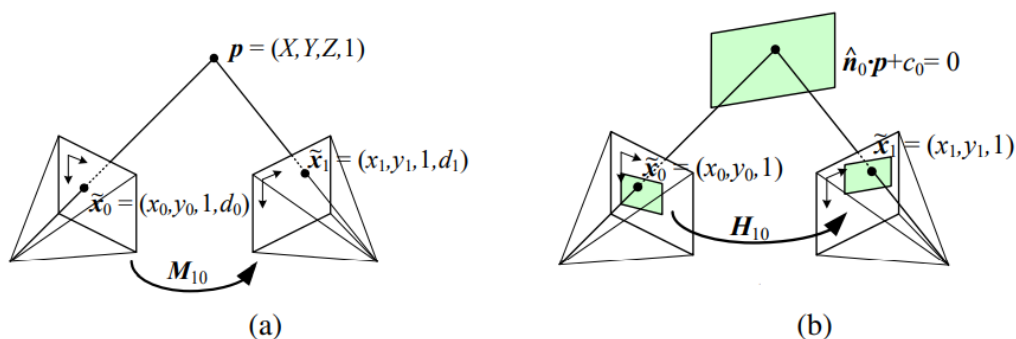


图 2-12 一个点被投影成两幅图像:(a) 三维点坐标  $(X, Y, Z, 1)$  和 2D 投影点  $(X, Y, 1, D)$  之间的关系; (b) 由全部位于公共平面  $\hat{n}_0 \cdot \mathbf{p} + c_0 = 0$  上的点诱导的平面单应性。

再次可以用  $3 \times 3$  单应性表示。如果我们假设校准矩阵具有已知的纵横比和投影中心 (2.59), 该单应性可以由旋转量和两个未知焦距参数化。这个特别的图像拼接应用中通常使用该配方 (第 8.2.3 节)。

### 以对象为中心的投影

使用长焦距镜头时, 通常难以可靠地进行估算仅从图像测量获得的焦距。这是因为焦距和到物体的距离高度相关, 因此很难将这两个效果分开。例如, 通过变焦镜头观看的物体的比例变化可能是由于缩放变化或朝向用户的运动。(这个效果被放到了阿尔弗雷德·希区柯克 (Alfred Hitchcock) 的电影《眩晕》(Vertigo) 中的戏剧性用法, 变焦和相机运动会产生令人不快的效果。)

如果我们写出对应于简单校准矩阵  $\mathbf{K}$  (2.59),

$$x_s = f \frac{\mathbf{r}_x \cdot \mathbf{p} + t_x}{\mathbf{r}_z \cdot \mathbf{p} + t_z} + c_x \quad (2.73)$$

$$y_s = f \frac{\mathbf{r}_y \cdot \mathbf{p} + t_y}{\mathbf{r}_z \cdot \mathbf{p} + t_z} + c_y \quad (2.74)$$

其中  $r_x$ ,  $r_y$  和  $r_z$  是  $\mathbf{R}$  的三行。如果到物体中心的距离 (物体的大小), 分母约为  $t_z$ , 而投影的总比例物体取决于  $f$  与  $t_z$  之比。因此, 很难区分这两个数量。

为了更清楚地看到这一点, 令  $\eta_z = t_z^{-1}$  且  $s = \eta_z f$ 。然后我们可以重写上面的内容方程为

$$x_s = s \frac{\mathbf{r}_x \cdot \mathbf{p} + t_x}{1 + \eta_z \mathbf{r}_z \cdot \mathbf{p}} + c_x \quad (2.75)$$

$$y_s = s \frac{\mathbf{r}_y \cdot \mathbf{p} + t_y}{1 + \eta_z \mathbf{r}_z \cdot \mathbf{p}} + c_y \quad (2.76)$$

(Szeliski 和 Kang, 1994; Pighin, Hecker 等, 1998)。投影  $s$  的比例可以如果我们正在看一个已知的物体 (即 3D 坐标  $\mathbf{p}$  是已知)。现在, 反距离  $\eta_z$  通常与  $s$  的估计值解耦, 并且可以根据对象旋转时的

391 缩短量估算。此外, 作为镜头变长, 即投影模型变为正交投影, 无需用正射影像模型代替透视成像  
392 模型, 因为相同的方程可以可以使用  $\eta_z \rightarrow 0$  (与  $f$  和  $t_z$  都变为无穷大相反)。这使我们能够形成正交  
393 重构技术 (例如因式分解) 与它们之间的自然联系投射/透视对应对象 (第 11.4.1 节)。

### 394 2.1.5 镜头畸变

395 以上成像模型均假设照相机遵循线性投影模型, 其中世界中的直线导致图像中的直线。(自然而然  
396 地不幸的是, 许多广角镜都有明显的径向畸变, 表现为直线投影中的可见曲率。(有关透镜光学器  
397 件 (包括色差) 的更详细讨论, 请参见第 2.2.3 节。)除非将这种畸变考虑在内因此, 不可能创建高  
398 度精确的真实感重建。为了例如, 通常不会考虑径向变形而构造的图像镶嵌由于像素融合之前对应  
399 特征的重合失调而显示模糊 (第 8.2 节)。

400 幸运的是, 在实践中补偿径向变形并不是那么困难。对于大多数镜头, 简单的四次畸变模型可  
401 以产生良好的效果。令  $(x_c, y_c)$  为在透视分割之后但在按焦距  $f$  和  $f$  缩放之前获得的像素坐标按图  
402 像中心  $(c_x, c_y)$  移动, 即

$$\begin{aligned} x_c &= \frac{\mathbf{r}_x \cdot \mathbf{p} + t_x}{\mathbf{r}_z \cdot \mathbf{p} + t_z} \\ y_c &= \frac{\mathbf{r}_y \cdot \mathbf{p} + t_y}{\mathbf{r}_z \cdot \mathbf{p} + t_z} \end{aligned} \quad (2.77)$$

403 径向畸变模型表示, 观察到的图像中的坐标向图像中心的方向偏移 (桶形失真) 或远离 (枕形失真),  
404 偏移量与其径向距离成正比 (图 2.13a-b)。<sup>⑥</sup>最简单径向变形模型使用低阶多项式, 例如

$$\begin{aligned} \hat{x}_c &= x_c (1 + \kappa_1 r_c^2 + \kappa_2 r_c^4) \\ \hat{y}_c &= y_c (1 + \kappa_1 r_c^2 + \kappa_2 r_c^4) \end{aligned} \quad (2.78)$$

405 这里的  $r_c^2 = x_c^2 + y_c^2$  和  $\kappa_1$  和  $\kappa_2$  称为径向变形参数。<sup>⑦</sup> 它也包括一个切向分量以解决镜片偏心的问题,  
406 最早是由 Brown (1966) 在摄影测量学文献中提出的, 因此有时被称为 Brown 或 Brown-Conrady 模  
407 型。但是, 失真的切向分量是通常会被忽略, 因为它们会导致估算值不稳定 (Zhang 2000)。

408 在径向变形步骤之后, 可以使用以下公式计算最终像素坐标

$$\begin{aligned} x_s &= f x'_c + c_x \\ y_s &= f y'_c + c_y. \end{aligned} \quad (2.79)$$

409 可以使用多种技术来估计给定的径向变形参数镜头, 如第 11.1.4 节所述。

<sup>⑥</sup> 变形镜头广泛用于正片制作, 不遵循这种径向失真模型。相反, 在第一近似下, 它们可以被认为诱导不同的垂直和水平缩放, 即非正方形像素。

<sup>⑦</sup> 有时  $x_c$  和  $x_{cis}$  的关系反过来表达, 即  $x_c = \hat{x}_c (1 + \kappa_1 \hat{r}_c^2 + \kappa_2 \hat{r}_c^4)$ 。如果我们通过除以  $f$  将图像像素映射成 (扭曲的) 光线, 这是很方便的。然后我们可以不分解光线, 在空间中获得真正的 3D 光线。





图 2-13 径向透镜畸变:(a) 桶形, (b) 枕形, (c) 鱼眼。鱼眼图像从一边到另一边跨越了几乎 180 度。

有时, 上述简化模型不能足够准确地模拟由复杂透镜产生的真实变形 (特别是在非常宽的角度下)。更完整的 anamorphic 模型还包括切向扭曲和偏离扭曲 (Slama 1980), 但这些扭曲在本书中没有涉及。

鱼眼透镜 (图 2.13c) 需要一个不同于传统的径向畸变多项式模型。鱼眼透镜近似地表现为距离光轴角度的等距投影器 (Xiong and Turkowski 1997),

$$r = f\theta, \quad (2.80)$$

这与方程 (8.55, 8.57) 所描述的极坐标投影相同。由于到中心的距离 (像素) 和视角之间大多是线性映射, 这种镜头有时被称为 f-theta 镜头, 这可能就是广受欢迎的理光 THETA 360 相机的名字由来。熊和 Turkowski(1997) 描述该模型可以扩展添加一个额外的二次修正的  $\phi$  和未知参数 (中心投影, 比例因子 s, 等等) 可以从一组重叠的鱼眼图像估计。使用直接 (灰度) 非线性最小化算法。

对于更大、更少规则的变形, 可能需要使用样条的参数变形模型 (Goshtasby 1989)。如果镜头没有单一的投影中心, 那么就有必要分别为每个像素对应的 3D 线 (而不是方向) 建模 (Gremban, Thorpe 和 Kanade 1988; Champleboux, Lavallée 等人, 1992a; Grossberg 和 Nayar 2001 年; Sturm 和 Ramalingam 2004; Tardif, Sturm

有一个微妙的问题与简单的径向失真模型有关, 这个问题经常被掩盖。我们引入了透视投影和最终传感器阵列投影步骤之间的非线性。因此, 一般来说, 我们不能将任意的  $3 \times 3$  矩阵  $K$  与一个旋转相乘, 使其成为上三角形式, 并将其吸收到全局旋转中。但是, 这种情况并没有一开始可能出现的那么糟糕。对于许多应用, 保持 (2.59) 的简化对角形式仍然是一个合适的模型。此外, 如果我们将径向和其他失真校正到保留直线的精度, 我们实际上已经将传感器转换回线性成像仪, 并且之前的分解仍然适用。

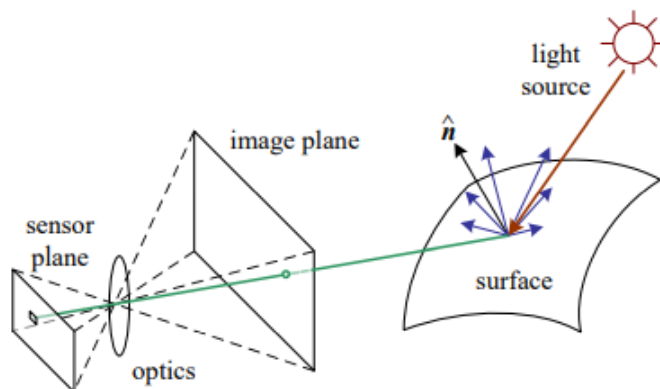


图 2-14 光度成像的简化模型。光由一个或多个光源发出，然后从物体表面反射。一部分光线射向摄像机。这个简化的模型忽略了现实场景中经常出现的多次反射。

## 2.2 光度成像

429

在为图像形成过程建模时,我们已经描述了世界上的三维几何特征如何投射到图像中的二维特征。然而,图像不是由二维特征组成的。相反,它们是由离散的颜色或强度值组成的。这些价值从何而来?它们是如何与环境中的灯光、表面属性和几何形状、相机光学和传感器属性联系在一起的(图 2.14)?在本节中,我们将开发一套模型来描述这些交互作用,并制定图像形成的生成过程。关于这些主题的更详细的论述可以在计算机图形和图像合成的教科书中找到(Cohen 和 Wallace 1993;Sillion 和 Puech 1994 年;瓦特 1995;格拉斯 1995;Weyrich, Lawrence 等人,2009;Hughes, van Dam 等人,2013;Marschner and Shirley 2015)。

### 2.2.1 光源

437

没有光,图像就不能存在。为了产生一幅图像,场景必须用一个或多个光源照亮。(某些模式,如荧光显微镜和 x 射线断层摄影术不适合这个模型,但我们在本书中不处理它们。)光源一般可分为点光源和面光源。

点光源起源于空间中的一个单一位置(如一个小灯泡),也可能是无限远(如太阳)。(注意,对于一些应用程序,如建模软阴影(半影),太阳可能必须作为一个区域光源处理。)点光源除了其位置外,还具有强度和颜色光谱,即波长  $L(\lambda)$  的分布。光源的强度随着光源和被光源之间距离的平方而衰减,因为同样的光在一个更大的(球面)区域上传播。光源也可能有方向衰减(依赖),但我们在简化模型中忽略了这一点。

区域光源更为复杂。简单的面积光源,如具有扩散器的荧光灯天花板灯具,可以建模为在所有

446



方向上相等的有限矩形区域发射光 (Cohen 和 Wallace 1993; Sillion 和 Puech 1994 年; 格拉斯 1995)。当分布具有很强的方向性时, 可以使用四维光场来代替 (Ashdown 1993)。

更复杂的光线分布, 例如, 接近于坐在室外庭院中的物体上的入射光照, 通常可以使用环境地图来表示 (Greene 1986)(最初称为反射地图 (Blinn and Newell 1976))。这个表示将入射光的方向  $\hat{v}$  映射到颜色值 (或波长  $\lambda$ ),

$$L(\hat{v}; \lambda), \quad (2.81)$$

等价于假设所有的光源都在无穷远。环境地图可以表示为立方体的集合 (Greene 1986), 可以表示为单一的经度纬度地图 (Blinn 和 Newell 1976), 也可以表示为反射球的图像 (Watt 1995)。获得真实世界环境地图的粗略模型的一种方便方法是, 拍摄一个反射镜面球体的图像 (有时会配以一个较暗的球体来捕捉高光), 并将该图像展开到所需的环境地图上 (Debevec 1998)。Watt(1995) 对环境映射进行了很好的讨论, 包括将三种最常用的表示形式的方向映射到像素所需的公式。

## 2.2.2 反射和阴影

当光线照射到物体表面时, 它被散射和反射 (图 2.15a)。许多不同的已经开发了一些模型来描述这种交互。在本节中, 我们首先描述最一般的形式, 双向反射分布函数, 然后看一些更专业的模型, 包括漫反射, 高光, 和 Phong 阴影模型。我们也讨论如何使用这些模型来计算与  $\mathbf{a}$  对应的全局照明现场。

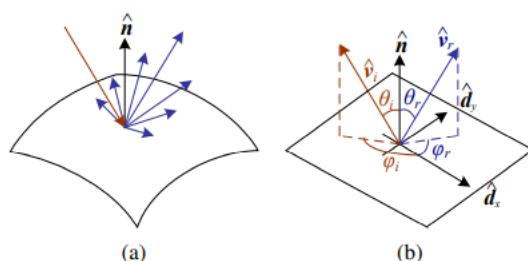


图 2-15 (a) 光在撞击表面时发生散射。(b) 双向反射率分布函数 (BRDF)  $f(\theta_i, \psi_i, \theta_r, \psi_r)$  由切角凹痕,  $\hat{v}_i$  和反射  $\hat{v}_r$ , 光线方向与局部表面坐标框架  $(\hat{\mathbf{d}}_x, \hat{\mathbf{d}}_y, \hat{\mathbf{n}})$ 。

## 双向反射分布函数 (BRDF)

光散射最一般的模型是双向反射分布函数 (BRDF)。<sup>⑧</sup> 相对于表面上的局部坐标系, BRDF 是一个四维函数, 它描述了在入射方向  $\hat{v}_i$ , 在反射方向  $\hat{v}_r$  发射的每个波长的多少 (图 2.15b)。函数可以写成相对于表面框架的入射角度和反射方向

$$f(\theta_i, \psi_i, \theta_r, \psi_r; \lambda) \quad (2.82)$$

BRDF 是倒数的, 也就是说, 由于光传输的物理特性, 你可以交换  $\hat{v}_i$  和  $\hat{v}_r$  的作用仍然得到相同的答案 (这有时被称为亥姆霍兹互惠)。

大多数表面都是各向同性的, 也就是说表面有首选方向就轻装运输而言。例外是各向异性的表面, 如刷过的表面 (划痕) 铝, 其反射率取决于相对于划痕方向的光方向。) 对于各向同性材料, 我们可以将 BRDF 简化

$$f_r(\theta_i, \theta_r, |\psi_r - \psi_i|; \lambda) \text{ (or) } f_r(\hat{v}_i, \hat{v}_r, \hat{n}; \lambda), \quad (2.83)$$

由于  $\theta_i, \theta_r$  和  $\phi_r - \phi_i$  可从  $\hat{v}_i, \hat{v}_r, \hat{n}$  方向计算。

在给定的条件下, 计算表面点  $p$  在  $\hat{v}_r$  方向上射出的光量在照明条件下, 我们将产品的入射光  $L_i(\hat{v}_i; \lambda)$  与 BRDF (一些作者把这个步骤称为卷积)。考虑到缩短因素  $\cos^+ \theta_i$ , 我们获得

$$L_r(\hat{v}_r; \lambda) = \int L_i(\hat{v}_i; \lambda) f_r(\hat{v}_i, \hat{v}_r, \hat{n}; \lambda) \cos^+ \theta_i d\hat{v}_i, \quad (2.84)$$

这里

$$\cos^+ \theta_i = \max(0, \cos \theta_i). \quad (2.85)$$

如果光源是离散的 (有限数目的点光源), 我们可以替换用求和积分,

$$L_r(\hat{v}_r; \lambda) = \sum_i L_i(\lambda) f_r(\hat{v}_i, \hat{v}_r, \hat{n}; \lambda) \cos^+ \theta_i. \quad (2.86)$$

给定表面的 BRDFs 可以通过物理建模 (Torrance 和麻雀 1967; 库克和托伦斯 1982 年; Glassner 1995), 启发式建模 (Phong 1975; Lafortune, Foo 等人, 1997 年), 或通过经验观 (Ward 1992; 威斯汀, 下午, 托兰斯 1992; Dana, van Ginneken 等, 1999; Marschner, Westin 等, 2000; Matusik, Pfister 等, 2003; Dorsey, Rushmeier 和 Sillion 2007; Weyrich, Lawrence 等人, 2009; Shi, Mo 等, 2019)。<sup>⑨</sup> 典型的 BRDFs 通常可以分为漫反射型和镜面反射型组件, 如下所述。

## 漫反射

<sup>⑧</sup> 实际上, 甚至存在更一般的光传输模型, 包括一些模拟沿表面的空间变化、次表面散射和大气效应的模型——见第 13.7.1 节 (多尔西、拉什梅尔和西利昂, 2007; Weyrich, Lawrence 等人, 2009 年)。

<sup>⑨</sup> 参见 <http://www1.cs.columbia.edu/CAVE/software/curet/> 了解一些经验抽样 BRDFs 的数据库。

漫射成分 (也被称为兰伯特或哑光反射) 将光从各个方向散射, 这是我们通常与阴影联系在一起的现象, 例如, 当照射一尊雕像时, 可以看到表面法线的平滑 (无光泽) 强度化 (图 2.16)。漫反射也常赋予强烈的体色因为它是由物体内部选择性吸收和再发射的光引起的材料 (沙佛 1985; 格拉斯 1995)。

当光在各个方向上均匀散射时, 即 BRDF 为常数,

$$f_d(\hat{v}_i, \hat{v}_r, \hat{n}; \lambda) = f_d(\lambda), \quad (2.87)$$



图 2-16 这张雕像的特写显示了漫反射 (平滑阴影) 和镜面反射 (亮色高光) 反射, 以及暗化的沟槽和折痕由于减少光的可见性和相互反射。(图片由加州理工学院视觉实验室提供, 网址: <http://www.vision.caltech.edu/archive.html>)。

光的量取决于入射光的方向和表面之间的角度  $\theta_i$ 。这是因为暴露在一定量光下的表面积变大了在倾斜的角度, 成为完全自阴影的出表面法线点远离灯光 (图 2.17a)。(想想你是如何让自己朝向太阳或者壁炉以获得最大的温暖, 手电筒斜斜地投射在墙上不如一个人直接指着它亮。) 漫反射的阴影方程可以因此可以写成

$$L_d(\hat{v}_r; \lambda) = \sum_i L_i(\lambda) f_d(\lambda) \cos^+ \theta_i = \sum_i L_i(\lambda) f_d(\lambda) [\hat{v}_i \cdot \hat{n}]^+, \quad (2.88)$$

其中

$$[\hat{v}_i \cdot \hat{n}]^+ = \max(0, \hat{v}_i \cdot \hat{n}) \quad (2.89)$$

### 镜面反射

典型 BRDF 的第二个主要成分是镜面反射 (光泽或高光), 这很大程度上取决于出射光的方向。考虑光反射 a 镜面 (图 2.17b)。入射光线反射的方向是绕表面法线  $\hat{n}$  旋转  $180^\circ$ 。使用与 (2.29-2.30) 式相同的符号, 我们可以计算镜面反射的方向  $\hat{s}_i$

$$\hat{s}_i = \mathbf{v}_{\parallel} - \mathbf{v}_{\perp} = (2\hat{n}\hat{n}^T - \mathbf{I}) \mathbf{v}_i. \quad (2.90)$$

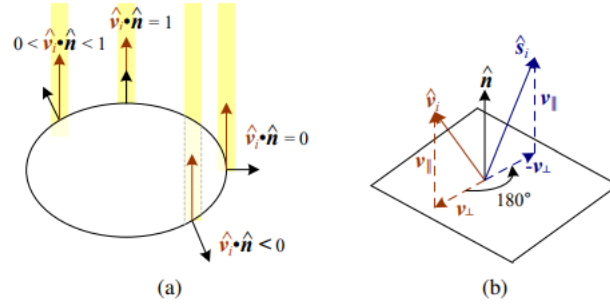


图 2-17 (a) 由于透视引起的反射光的减小取决于  $\hat{v}_i \cdot \hat{n}$ , 入射光方向  $\hat{v}_i$  与表面法线之间的余弦  $\hat{n}$ 。(b) 镜面 (镜面) 反射: 入射光线方向被反射到绕表面法线  $\hat{n}$  镜面方向  $\hat{s}_i$ 。

因此, 在给定方向  $\hat{v}_r$  反射的光的数量取决于角  $\theta_s$  因为  $-(v_r \cdot s_i)$  在视点方向  $v_r$  和镜面方向  $s_i$  之间。例如, Phong(1975) 模型使用角度余弦的幂, 因此, 在给定方向  $\hat{v}_r$  上反射的光的数量取决于视点方向  $\hat{v}_r$  和镜面方向  $\hat{s}_i$  之间的角度  $\theta_s = \cos^{-1}(\hat{v}_r \cdot \hat{s}_i)$ 。例如, Phong(1975) 模型使用角度余弦的幂,

$$f_s(\theta_s; \lambda) = k_s(\lambda) \cos^{k_e} \theta_s, \quad (2.91)$$

其中 Torrance and Sparrow(1967) 的微面模型使用高斯模型,

$$f_s(\theta_s; \lambda) = k_s(\lambda) \exp(-c_s^2 \theta_s^2). \quad (2.92)$$

大指数  $k_e$  (或反高斯宽度  $c_s$ ) 对应更多的镜面具有明显的高光, 而较小的指数更好的模型材料与更柔和的光泽。

#### 冯氏阴影

Phong(1975) 将反射的漫反射和镜面反射结合到另一个术语, 他称之为环境照明。这个术语说明了物体是通常不仅由点光源照明, 也由一般的漫射照明对应于相互反射 (如房间墙壁) 或远处的源, 如蓝色的天空。在 Phong 模型中, 环境项不依赖于表面取向, 但是取决于环境光照  $L_a(\lambda)$  和物体  $k_a(\lambda)$  的颜色,

$$f_a(\lambda) = k_a(\lambda) L_a(\lambda). \quad (2.93)$$

把所有这些内容放在一起, 我们得到补色渲染模型,

$$L_r(\hat{v}_r; \lambda) = k_a(\lambda) L_a(\lambda) + k_d(\lambda) \sum_i L_i(\lambda) [\hat{v}_i \cdot \hat{n}]^+ + k_s(\lambda) \sum_i L_i(\lambda) (\hat{v}_r \cdot \hat{s}_i)^{k_e}. \quad (2.94)$$

图 2.18 显示了一组典型的补色渲染模型组件, 它们是远离表面法线的角度的函数 (在包含照明方向和观察者的平面中)。

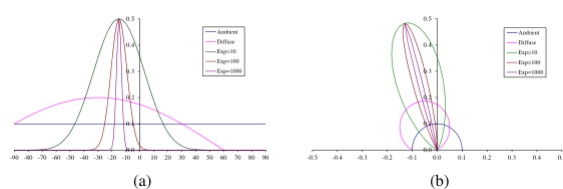


图 2-18 通过 Phong 阴影模型 BRDF 对固定入射光照方向的横截面:(a) 组件值作为远离表面法线角度的函数;(b) 极坐标图。丰指数  $k_e$  的值由“Exp”标签和光源是在一个角度  $30^\circ$  远离正常。

通常，环境和漫反射颜色分布  $k_a(\lambda)$  和  $k_d(\lambda)$  是相同的，因为它们都是由于表面材料内部的亚表面散射 (体反射 1985)。镜面反射分布  $k_s(\lambda)$  通常是均匀的白色，因为它是由不改变光颜色的界面反射引起的。(例外情况是金属材料，如铜，而不是更常见的绝缘材料，如塑料)。

环境照明  $L_a(\lambda)$  通常具有与直接光源  $L_i(\lambda)$  不同的色偏，例如，对于阳光充足的室外场景，它可以是蓝色，或者对于用蜡烛或白炽灯照明的室内，它可以是黄色。(阴影区域中的环境天空照明通常会导致阴影比场景中相应的照亮部分显得更蓝)。还要注意补色渲染模型 (或任何阴影模型) 的漫射分量取决于入射光源  $\mathbf{v}_i$  的角度，而镜面分量取决于观察者  $\mathbf{v}_r$  和镜面反射方向  $\mathbf{s}_i$  之间的相对角度 (镜面反射方向  $\mathbf{s}_i$  本身取决于入射光方向  $\mathbf{v}_i$  和表面法线  $\mathbf{n}$ )。

补色渲染模型在物理精度方面已经被计算机图形学中较新的模型所取代，包括库克和托兰斯 (1982) 基于托兰斯和斯派洛 (1967) 的原始微面模型开发的模型。虽然最初，计算机图形硬件实现了补色渲染模型，可编程像素着色器的出现使得使用更复杂的模型变得可行。

### 双色反射模型

托兰斯和斯派洛 (1967) 的反射模型也构成了谢弗 (1985) 双色反射模型的基础，该模型指出从单一光源照射的均匀材料的表观颜色取决于两项之和，

$$L_r(\mathbf{v}_r; \lambda) = L_i(\mathbf{v}_r, \mathbf{v}_i, \mathbf{n}; \lambda) + L_b(\mathbf{v}_r, \mathbf{v}_i, \mathbf{n}; \lambda) \quad (2.95)$$

$$= c_i(\lambda)m_i(\mathbf{v}_r, \mathbf{v}_i, \mathbf{n}) + c_b(\lambda)m_b(\mathbf{v}_r, \mathbf{v}_i, \mathbf{n}), \quad (2.96)$$

即在界面  $L_i$  反射的光的辐射和在表面体  $L_b$  反射的辐射。每一个都是一个简单的乘积，它是一个相对功率谱  $c(\lambda)$  和一个量级  $m(\mathbf{v}_r, \mathbf{v}_i, \mathbf{n})$  的乘积，前者只取决于波长，后者只取决于几何形状。(这个模型可以很容易地从补色渲染模型的一般化版本中推导出来，通过假设单个光源和没有环境照明，并重新排列术语。) 双色模型已经成功地用于计算机视觉中，以分割阴影变化较大的镜面彩色物体 (Klinker 1993)，并启发了局部双色模型的应用，如拜耳模式镶嵌 (Bennett, Uyttendaele 等人, 2006)。

### 全局照明 (光线跟踪和光能传递)

到目前为止，简单的阴影模型假设光线离开光源，从摄像机可见的表面反射，从而改变强度或颜色，并到达摄像机。事实上，光源可以被遮光器遮挡，光线可以在从光源到相机的过程中围绕场景多次反射。

传统上有两种方法用来模拟这种效应。如果场景主要是镜面的 (典型的例子是由玻璃物体和镜面或高度抛光的球构成的场景), 首选的方法是光线跟踪或路径跟踪 (Glassner 1995, Akenin-Müller 和 Haines 2002, Marschner 和 Shirley 2015), 它跟踪来自相机的单个光线, 穿过多次向光源的反弹 (反之亦然)。如果场景主要由均匀反照率简单几何照明器和表面组成, 光能传递 (全局照明) 技术是优选的 (科恩和华莱士 1993; Sillion and Puech 1994, Glassner 1995)。两种技术的结合也被开发出来 (华莱士, 科恩和格林伯格, 1987), 以及更一般的光传输技术, 用于模拟效果, 如由水波产生的焦散。

基本光线跟踪算法将光线与相机图像中的每个像素相关联, 并找到它与最近表面的交点。主要贡献可以使用前面介绍的简单阴影方程计算 (例如, 等式 (2.94)) 对于该表面元素可见的所有光源。(计算哪些表面被光源照亮的另一种技术是计算阴影图或阴影缓冲区, 即从光源的角度渲染场景, 然后将渲染的像素深度与该图进行比较 (威廉姆斯 1983; Akenin-Müller 和 Haines 2002)。然后, 可以沿着镜面反射方向向场景中的其他对象投射额外的次级光线, 跟踪镜面反射引起的任何衰减或颜色变化。

光能传递的工作方式是将照明值与矩形表面区域 (包括区域光源) 相关联。场景中任意两个 (相互可见的) 区域之间交换的光量可以被捕获为形状因子, 这取决于它们的相对方向和表面反射特性, 以及当光分布在离它越远的更大的有效球面上时的  $1/2$  衰减 (科恩和华莱士 1993; Sillion and Puech 1994, Glassner 1995)。然后可以建立一个大的线性系统来求解每个区域面片的最终亮度, 使用光源作为强制函数 (右手侧)。一旦系统被解决, 场景可以从任何期望的角度进行渲染。在某些情况下, 使用计算机视觉技术从照片中恢复场景中的全局照明是可能的 (于, Debevec 等人, 1999)。

基本的光能传递算法不考虑某些近场效应, 例如内角和划痕变暗, 或者由其他表面的部分阴影导致的有限环境照明。这种效应已经在许多计算机视觉算法中得到利用 (Nayar, Ikeuchi, 和 Kanade 1991, 兰格和扎克 1994)。

虽然所有这些全局照明效果都会对场景的外观产生强烈的影响, 从而对其进行 3D 解释, 但本书并没有详细介绍。(请参见第 13.7.1 节, 了解如何从真实场景和对象中恢复快速傅立叶变换。)

### 2.2.3 光学

场景中的光线一旦到达相机, 在到达传感器 (模拟胶片或数字硅) 之前, 仍必须通过镜头。对于许多应用来说, 只要把透镜当作一个理想的针孔就够了, 这个针孔只是把所有的光线都投射到一个共同的投射中心 (图 2.8 和 2.9)。然而, 如果我们想处理聚焦、曝光、渐晕和像差等问题, 我们需要开发一个更复杂的模型, 这就是光学研究的来源 (穆勒 1988; Ray 2002, Hecht 2015)。

图 2.19 显示了最基本的透镜模型, 即由单片玻璃组成的薄透镜, 其两侧曲率非常低且相等。根据透镜定律 (可以使用光线折射的简单几何参数推导出该定律), 到物体的距离  $z_o$  和透镜后面形成聚焦图像的距离  $z_i$  之间的关系可以表示为

$$\frac{1}{z_o} + \frac{1}{z_i} = \frac{1}{f} \quad (2.97)$$



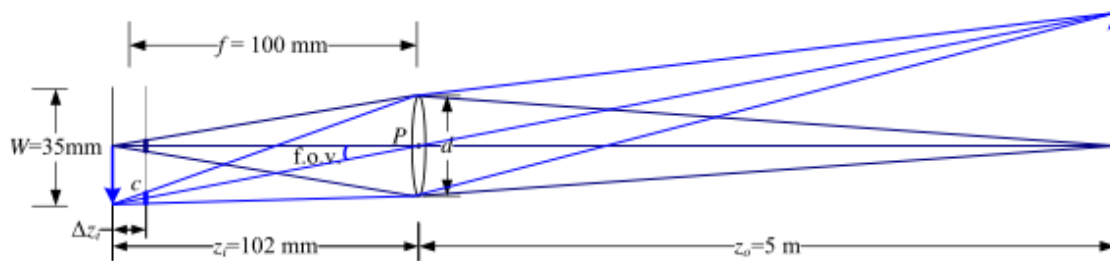


图 2-19 一个焦距为  $f$  的薄透镜将来自透镜前距离  $z_o$  的平面的光聚焦到透镜后距离  $z_i$  的平面上，在那里  $\frac{1}{z_o} + \frac{1}{z_i} = \frac{1}{f}$ 。如果焦平面 ( $c$  旁边的垂直灰线) 向前移动，不再关注图像和模糊圈  $c$  (小厚线段) 取决于图像平面运动的距离  $\Delta z_i$  相对于镜头光圈直径  $d$ 。视野 (f.o.v) 取决于传感器之间的比例宽度  $W$  和焦距  $f$  (或者更精确地说, 聚焦点距离  $z_i$ , 这通常非常接近  $f$ )。

其中  $f$  被称为透镜的焦距。如果我们让  $z_o \rightarrow \infty$ ，即我们调整透镜 (移动像面) 使无穷远处的物体聚焦，我们得到  $z_i = f$ ，这就是为什么我们可以把焦距为  $f$  的透镜想象成 (相当于一次近似) 距焦面距离为  $f$  的针孔 (图 2.10)，它的试图由 (2.60) 给出。

如果焦平面移出其适当的焦点内设置  $z_i$  (例如。通过扭曲镜头上的聚焦环)，在  $z_o$  的物体不再聚焦，如图 2.19 的灰色平面所示。模糊度是通过模糊圈测量的 (灰色平面上显示的是短而粗的蓝色线段)。⑩用相似三角形可导出混淆圆方程；它取决于相对于原始焦点距离  $z_i$  的焦平面运动的距离  $z_i$  和孔径的直径  $d$  (见练习 2.4)。

场景中允许的深度变化将混淆圈限制在一个可接受的数值，通常称为景深，是焦距和光圈的函数，如许多镜头标记所示 (图 2.20)。因为这个景深取决于光圈直径  $d$ ，我们还必须知道这是如何随通常显示的  $f$  数变化的，它通常被表示为  $f/\#$  或  $N$ ，并被定义为

$$f/\# = N = \frac{f}{d} \quad (2.98)$$

其中焦距  $f$  和孔径  $d$  以相同的单位 (例如，毫米) 测量。

通常写  $f$  数的方法是使用实际数代替式子中的井号，即  $f/1.4$ ,  $f/2$ ,  $f/2$ ,  $f/8$ , ...  $f/22$ 。(或者，我们可以说  $N = 1.4$ , 等) 解释这些数字的一个简单方法是注意焦距除以  $f$  数得到直径  $d$ ，所以这些只是孔径的公式<sup>11</sup>。

请注意， $f$  数的通常级数是逗号，是根号  $\sqrt{2}$  的倍数，因为这相当于每次选择一个较小的  $f$  数时，入瞳面积增加一倍。(这个翻倍也叫改变曝光一个曝光值或者 EV。它对到达传感器的光量的影响与曝光时间加倍相同，例如从  $1/125$  到  $1/250$ ，参见练习 2.5。)

⑩如果光圈不是完全圆形的，例如，如果它是由六边形光圈引起的，有时可能在实际模糊函数中看到这种效果 (Levin, Fergus 等, 2007; Joshi, Szeliski, 和 Kriegman 2008)，或者在射向太阳时看到的“闪光”。

<sup>11</sup>这也解释了为什么变焦镜头的  $f$  值会随着当前的变焦 (焦距) 设置而变化。

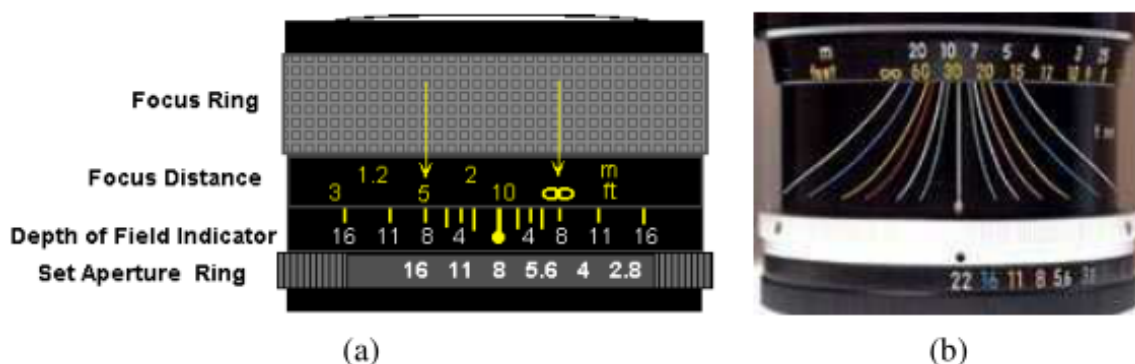


图 2-20 规则 and 变焦镜头的景深指示。

既然你知道了如何在  $f$  数和光圈直径之间转换，你可以根据焦距  $f$ 、混淆圆  $c$  和焦距  $z_0$  来构建你自己的景深图，如练习 2.4 中所解释的，看看这些图与你在实际镜头上观察到的有多匹配，如图 2.20 所示。

当然，真正的透镜不是无限薄的，因此会受到几何像差的影响，除非使用复合元素来校正它们。经典的五种赛德尔像差在使用三阶光学时出现，包括球面像差、彗差、散光、场曲率和畸变 (穆勒 1988; Ray 2002; Hecht 2015)。

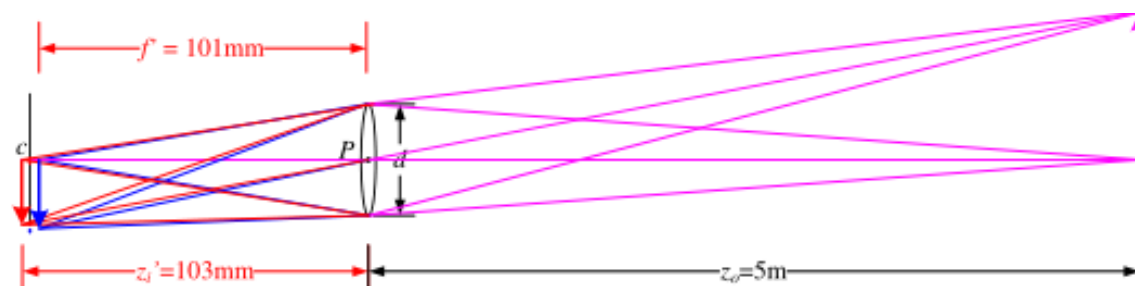


图 2-21 在一个受色差影响的镜头中，不同波长的光 (例如，红色和模糊的箭头) 以不同的焦距  $f'$  聚焦，因此有不同的深度  $z_i'$ ，导致几何 (平面内) 位移和失焦。

587

## 588 色差

如图 2.21 所示，由于玻璃的折射率随着波长的变化而略有变化，简单的透镜会出现色差，即不同颜色的光聚焦在稍微不同的距离上 (因此放大系数也略有不同)。与波长相关的放大系数，即横向色差，可以模拟为每种颜色的径向失真 (第 2.1.5 节)，因此，可以使用第 11.1.4 节中描述的技术进行校准。由纵向色差引起的波长相关模糊可以使用第 10.1.4 节中描述的技术进行校准。不幸的是，由纵向像差引起的模糊可能更难消除，因为较高的频率可能被强烈衰减，因此难以恢复。

593



为了减少色差和其他种类的像差，今天的大多数摄影镜头是由不同的玻璃元件 (具有不同的涂层) 制成的复合镜头。这种透镜不能再被建模为具有所有光线都必须穿过的单个节点 P (当用针孔模型近似透镜时)。相反，这些透镜既有一个前节点，光线通过它进入透镜，又有一个后节点，光线通过它离开到达传感器。在实践中，当执行仔细的摄像机校准时，例如，当确定围绕哪个点旋转以捕捉无视差全景图时，只有前节点的位置是感兴趣的 (参见第 8.2.3 节和 (利特尔菲尔德 2006; Houghton 2013))。

然而，不是所有的透镜都可以被建模为具有单个节点。特别是，非常广角的透镜，如鱼镜头（第 2.1.5 节）和某些由透镜和曲面镜组成的反折射成像系统（贝克和纳亚尔 1999 年）没有一个点让所有获得的光线通过。在这种情况下，最好明确构建像素坐标和 3D 之间的映射函数（查找表）。太空

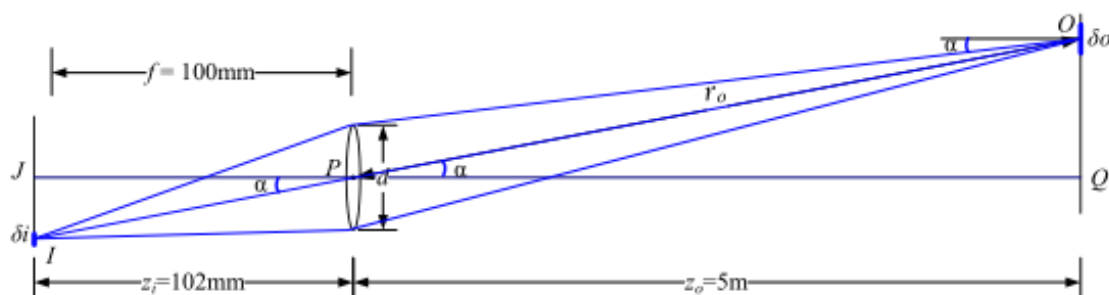


图 2-22 照射到表面积, 像素上的光量取决于孔径直径  $d$  与焦距  $f$  之比的平方, 以及离轴角  $\alpha$  余弦的四次方, 即  $\cos^4 \alpha$

中的光线 (格雷班、索普和卡纳德 1988; Champleboux, Lavalée et al. 1992 a; Grossberg 和 Nayar 2001 Sturm 和 Ramalingam 2004 Tardif, Sturm 等人, 2009), 如第 2.1.5 节所述

## 光晕效果

现实世界镜头的另一个属性是渐晕，即图像亮度向图像边缘下降的趋势。两种现象通常有助于这种效果(雷 2002)。第一种称为自然渐晕，是由于物体表面、投影像素和镜头光圈的缩短，如图 2.22 所示。考虑以离轴角  $\alpha$  离开尺寸为  $\delta_o$  的物体表面的光。因为这个补片相对于相机镜头是缩短的，所以到达镜头的光量减少了一个因子  $\cos\alpha$ 。到达透镜的光量也受到通常的  $1/r^2$  衰减的影响；在这种情况下，距离  $r_o = z_o/\cos\alpha$ 。光通过的孔径的实际面积被一个附加因子  $\cos\alpha$  缩短，即从点 O 看到的孔径是一个尺寸为  $d \times d \cos\alpha$  的椭圆。把所有这些因素放在一起，我们看到离开 O 并通过光圈到达位于 I 的图像像素的光量成正比于

$$\frac{\delta o \cos \alpha}{r_a^2} \pi \left( \frac{d}{2} \right)^2 \cos \alpha = \delta o \frac{\pi d^2}{4 z_a^2} \cos^4 \alpha \quad (2.99)$$

由于三角形  $\delta OPQ$  和  $\delta IPJ$  是相似的, 物体表面的投影面积  $\delta_o$  和图像像素  $\delta_i$  与  $z_o: z_i$  具有相同的

614 (平方) 比率,

$$\frac{\delta o}{\delta i} = \frac{z_o^2}{z_i^2} \quad (2.100)$$

615 把 (2.99) 和 (2.100) 合并, 我们得到到达像素  $i$  的光量与孔径  $d$ 、聚焦距离  $z_i = f$  和离轴角 之间的  
616 最终公式,

$$\delta_o \frac{\pi}{4} \frac{d^2}{z_o^2} \cos^4 \alpha = \delta_i \frac{\pi}{4} \frac{d^2}{z_i^2} \cos^4 \alpha \approx \delta_i \frac{\pi}{4} \left(\frac{d}{f}\right)^2 \cos^4 \alpha, \quad (2.101)$$

617 其被称为场景辐射  $L$  和到达像素传感器的光 (辐照度)  $E$  之间的基本辐射关系,

$$E = L \frac{\pi}{4} \left(\frac{d}{f}\right)^2 \cos^4 \alpha, \quad (2.102)$$

618 (Horn 1986; Nalwa 1993; Ray 2002; Hecht 2015)。注意, 在这个等式中, 光量是如何依赖于像素表  
619 面积的 (这就是为什么在单镜头反射 (SLR) 数码相机中, 较小的传感器比单镜头反射 (SLR) 数  
620 码相机噪音大得多), f-stop 的平方反比  $N = f/d$  (2.98), 以及  $\cos^4$  离轴衰减的四次方, 这是自然  
621 的渐晕术语。

622 另一种主要的渐晕, 称为机械渐晕, 是由复合透镜中透镜元件外围附近光线的内部遮挡引起的,  
623 如果不对实际透镜设计进行完整的光线跟踪, 则很难用数学方法进行描述。<sup>12</sup> 然而, 与自然渐晕不  
624 同, 通过减小相机孔径 (增加 f 数值), 可以减少机械渐晕。如第 10.1.3 节所述, 还可以使用积分球、  
625 均匀照明目标或摄像机旋转等特殊装置对其进行校准 (以及自然校准)。

## 626 2.3 数码相机

627 从一个或多个光源开始, 反射出世界上的一个或多个表面, 并通过相机的光学元件 (透镜), 光最终  
628 到达成像传感器。到达这个传感器的光子是如何转换成我们观察数字图像时观察到的数字 (R, G,  
629 B) 值的? 在本节中, 我们将开发一个简单的模型, 用于解释最重要的影响, 如曝光 (增益和快门速  
630 度)、非线性映射、采样和混叠以及噪声。图 2.23 基于 Healey 和 Kondepudy (1994 年)、Tsin、Ramesh  
631 和 Kanade (2001 年)、Liu、Szeliski 等人 (2008 年) 开发的相机模型, 显示了现代数码相机处理阶  
632 段的简单版本。Chakrabarti、Scharstein 和 Zickler (2009) 开发了一个复杂的 24 参数模型, 该模型  
633 与数码相机中执行的处理更加匹配, 而 Kim、Lin 等人 (2012)、Hasinoff、Sharlet 等人 (2016) 以及  
634 Karaimer 和 Brown (2016) 提供了现代相机内处理管道的最新模型。最近, Brooks, Mildenhall 等人  
635 (2019) 开发了相机内图像处理管道的详细模型, 以便将 (未处理的) 噪声 JPEG 图像转化为原始图  
636 像, 以便更好地去噪, 而 Tseng, Yu 等人 (2019 年) 开发了可用于图像质量优化的可调相机处理管  
637 道模型。

638 落在成像传感器上的光通常由一个主动感测区域拾取, 在曝光持续时间内进行积分 (通常表示  
639 为几秒钟内的快门速度, 例如 1/125, 1/60, 1/30), 然后传递到一组感测放大器。目前用于数码相机  
640 和摄像机的传感器主要有电荷耦合器件 (CCD) 和互补金属氧化物半导体 (CMOS) 两种。

<sup>12</sup>There are some empirical models that work well in practice (Kang and Weiss 2000; Zheng, Lin, and Kang 2006).

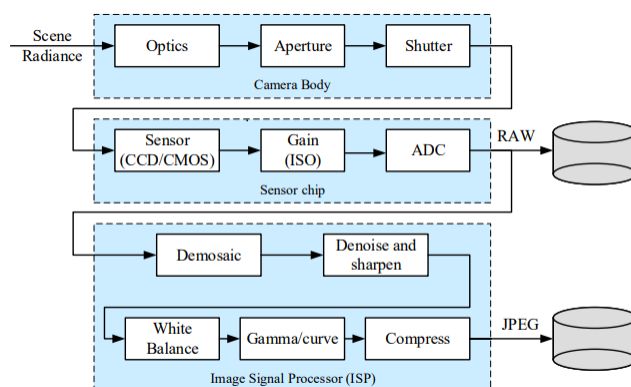


图 2-23 图像传感管道，显示各种噪声源以及典型的数字后处理步骤。

在 CCD 中，光子在曝光时间内聚集在每个活跃的阱中。然后，在传输阶段，电荷以一种“桶旅”的形式从一口井传输到另一口井，直到它们沉积在感测放大器上，感测放大器将信号放大并传递给模数转换器（ADC）13。当一个过度曝光的像素的电荷溢出到相邻的像素时，较旧的 CCD 传感器容易出现晕染，但是大多数较新的 CCD 都有防晕技术（“槽”），多余的电荷会溢出。

在 CMOS 中，光子击中传感器直接影响光电探测器的导电性（或增益），光电探测器可以有选择地选通以控制曝光时间，并在使用多路复用方案读出之前进行局部放大。传统上，CCD 传感器在数字单反相机等对质量敏感的应用中优于 CMOS，而 CMOS 更适合于低功耗应用，但现在 CMOS 被用于大多数数码相机。在 CCD 中，光子在曝光时间内聚集在每个活跃的阱中。然后，在传输阶

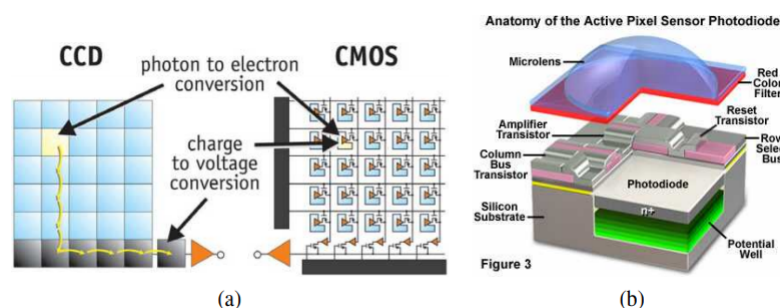


图 2-24 数字成像传感器:(a) ccd 将光电产生的电荷从一个像素移动到另一个像素，并在输出节点将其转换为电压;CMOS 成像器将电荷转换为像素内的电压 (Litwiller2005)©PhotonicsSpectra2005;(b) 横切图的 mos 像素传感器，从 <http://olympus.magnet.fsu.edu/primer/digitalimaging/cmosimagesensors>。超文本标记语言

段, 电荷以一种“桶旅”的形式从一口井传输到另一口井, 直到它们沉积在感测放大器上, 感测放大器将信号放大并传递给模数转换器 (ADC)<sup>13</sup>。当一个过度曝光的像素的电荷溢出到相邻的像素时, 较旧的 CCD 传感器容易出现晕染, 但是大多数较新的 CCD 都有防晕技术 (“槽”), 多余的电荷会溢出。

在 CMOS 中, 光子击中传感器直接影响光电探测器的导电性 (或增益), 光电探测器可以有选择地选通以控制曝光时间, 并在使用多路复用方案读出之前进行局部放大。传统上, CCD 传感器在数字单反相机等对质量敏感的应用中优于 CMOS, 而 CMOS 更适合于低功耗应用, 但现在 CMOS 被用于大多数数码相机。

影响数字图像传感器性能的主要因素是快门速度、采样间距、填充因子、芯片尺寸、模拟增益、传感器噪声以及模数转换器的分辨率 (和质量)。这些参数的许多实际值可以从嵌入数字图像的 EXIF 标记中读取。其他可从相机制造商的规格表或相机审查或校准网站获取<sup>14</sup>。

**快门速度.** 快门速度 (曝光时间) 直接控制到达传感器的光量, 从而确定图像是曝光不足还是曝光过度。(对于明亮的场景, 需要大光圈或慢快门速度来获得浅景深或运动模糊, 摄影师有时会使用中性感密度滤光片。) 对于动态场景, 快门速度还决定结果图片中的运动模糊量。通常, 较高的快门速度 (较少的运动模糊) 会使后续分析更容易 (有关消除这种模糊的技术, 请参见第 10.3 节)。然而, 当捕获视频以供显示时, 可能需要一些运动模糊来避免频闪效应。

**采样间距.** 采样间距是成像芯片上相邻传感器单元之间的物理间距 (图 2.24)。具有较小采样间距的传感器具有较高的采样密度, 因此为给定的有源芯片区域提供较高的分辨率 (以像素为单位)。然而, 一个较小的间距也意味着每个传感器的面积较小, 不能积累那么多光子; 这使得它对光不敏感, 更容易产生噪声。

**填充因子.** 填充系数是有效感应区域大小, 作为理论可用感应区域的一部分 (水平和垂直取样间距的乘积)。更高的填充因子通常更可取, 因为它们会导致更多的光捕获和更少的混叠 (见第 2.3.1 节)。然而, 这必须与在活动感测区域之间放置额外电子元件的需要相平衡。摄像机的填充系数可通过光度摄像机校准过程进行经验测定 (见第 10.1.4 节)。

**芯片尺寸.** 传统上, 视频和点播相机使用的芯片面积较小 (1/4 1/2 英寸的传感器<sup>15</sup>), 而数码单反相机则试图接近传统的 35mm 胶片框尺寸<sup>16</sup>。当整体设备尺寸不重要时, 最好使用更大的芯片尺寸, 因为每个传感器单元可以更具感光性。(对于小型相机来说, 更小的芯片意味着所有的光学元

---

<sup>13</sup>In digital still cameras, a complete frame is captured and then read out sequentially at once. However, if video is being captured, a rolling shutter, which exposes and transfers each line separately, is often used. In older video cameras, the even fields (lines) were scanned first, followed by the odd fields, in a process that is called interlacing.

<sup>14</sup><http://www.clarkvision.com/imagedetail/digital.sensor.performance.summary/>.

<sup>15</sup>These numbers refer to the “tube diameter” of the old vidicon tubes used in video cameras. The 1/2.5” sensor on the Canon SD800 camera actually measures 5.76mm × 4.29mm, i.e., a sixth of the size (on side) of a 35mm full-frame (36mm × 24mm) DSLR sensor.

<sup>16</sup>When a DSLR chip does not fill the 35mm full frame, it results in a multiplier effect on the lens focal length. For example, a chip that is only 0.6 the dimension of a 35mm frame will make a 50mm lens image the same angular extent as a 50/0.6 = 50 × 1.6 = 80mm lens, as demonstrated in (2.60).



件都可以按比例缩小。)然而,更大的芯片生产成本更高,这不仅是因为每个晶圆中可以装入的芯片更少,而且还因为芯片缺陷的概率随芯片面积线性增加。

**模拟增益.**在模数转换之前,感测信号通常由感测放大器放大。在摄像机中,这些放大器的增益传统上是由自动增益控制(AGC)逻辑控制的,它可以调整这些值以获得良好的整体曝光。在较新的数码相机中,用户现在通过ISO设置对该增益有一些额外的控制,ISO设置通常以ISO标准单位表示,例如100、200或400。由于大多数相机中的自动曝光控制也会调整光圈和快门速度,因此手动设置ISO会从相机控制中移除一个自由度,就像手动指定光圈和快门速度一样。理论上,更高的增益可以让相机在低光照条件下表现更好(当光圈已经达到最大值时,由于长时间曝光,运动模糊更少)。但实际上,较高的ISO设置通常会放大传感器噪声。

**传感器噪声.**在整个感测过程中,来自各种来源的噪声被添加,其中可能包括固定模式噪声、暗电流噪声、散粒噪声、放大器噪声和量化噪声(Healey和Kondepudy 1994; Tsin、Ramesh和Kanade 2001)。采样图像中存在的最终噪声量取决于所有这些量,以及入射光(由场景亮度和光圈控制)、曝光时间和传感器增益。此外,对于噪声是由于低光子计数引起的弱光条件,噪声的泊松模型可能比高斯模型更合适(Alter、Matsushita和Tang 2006; Matsushita和Lin 2007a; Wilburn、Xu和Matsushita 2008; Takamatsu、Matsushita和Ikeuchi 2008)。

如第10.1.1节所述,Liu,Szeliski et al.(2008)使用该模型以及Grossberg和Nayar(2004)获得的相机响应函数(CRF)的经验数据库来估计给定图像的噪声级函数(NLF),它预测给定像素处的总体噪声方差作为其亮度的函数(为每个颜色通道估计单独的NLF)。另一种方法是,当您在拍照前可以使用相机时,通过重复拍摄包含各种颜色和亮度的场景来预校准NLF,如图10.3b所示的麦克白颜色图(McCamy、Marcus和Davidson 1976)。(在估计方差时,一定要丢弃或降低梯度较大的像素,因为曝光之间的小偏移会影响这些像素处的感测值。)不幸的是,由于传感系统内发生复杂的相互作用,可能必须对不同的曝光时间和增益设置重复预校准过程。

在实际应用中,大多数计算机视觉算法,如图像去噪、边缘检测和立体匹配,都至少得益于对噪声水平的初步估计。除非能够预先校准相机或重复拍摄同一场景,最简单的方法是寻找接近常量值的区域,并估计这些区域中的噪声方差(Liu,Szeliski et al.2008)。

**ADC分辨率.**成像传感器内发生的模拟处理链的最后一步是模数转换(ADC)。虽然可以使用多种技术来实现该过程,但两个重要的量是该过程的分辨率(产生多少位)和噪声级(这些位中有多少位在实践中有用)。对于大多数相机来说,引用的比特数(8比特用于压缩的JPEG图像,16比特用于一些DSLR提供的原始格式)超过了实际可用比特数。最好的辨别方法是简单地校准给定传感器的噪声,例如,通过重复拍摄同一场景并绘制估计的噪声作为亮度的函数(练习2.6)。

**数字后处理.**一旦到达传感器的辐照度值被转换为数字位,大多数相机在压缩和存储像素值之前会执行各种数字信号处理(DSP)操作来增强图像。这些包括滤色器阵列(CFA)去马赛克化、白点设置和通过伽玛函数映射亮度值,以增加信号的感知动态范围。我们将在第2.3.2节中介绍这些主题,但在此之前,我们将返回到与传感器阵列填充因子相关的别名主题。

**新颖的成像传感器.**成像传感器和深度传感器等相关技术的能力继续快速发展。跟踪这些发展的会

711 议包括由成像科学和技术协会主办的电子成像科学和技术研讨会和世界图像传感器组织。

### 712 2.3.1 采样和混叠

713 当入射到图像传感器上的光场落在成像芯片中的活动感测区域上时会发生什么情况? 到达每个活动  
714 单元的光子被整合, 然后数字化, 如图 2.24 所示。然而, 如果芯片上的填充因子很小, 并且信号没  
715 有其他频带限制, 则会发生视觉上不平滑的混叠。

716 为了探索混叠现象, 让我们先看看一维信号 (图 2.25), 其中有两个正弦波, 一个频率为  $f = 3/4$ ,  
717 另一个频率为  $f = 5/4$ 。如果我们以  $f = 2$  的频率对这两个信号进行采样, 我们会看到它们产生相  
718 同的样本 (以黑色显示), 所以我们说它们是混叠的<sup>①</sup>。为什么这是一个坏的影响? 本质上, 我们不  
719 能再重建原始信号, 因为我们不知道这两个原始频率中的哪一个存在。

720 事实上, Shannon 的采样定理表明, 从瞬时样本重建信号所需的最小采样率 (Oppenheim 和 Schafer  
721 1996; Oppenheim、Schafer 和 Buck 1999) 必须至少是最高频率的两倍<sup>18</sup>,

$$f_s \geq 2f_{max} \quad (2.103)$$

722 信号中的最大频率称为奈奎斯特频率, 最小采样频率  $r_s = 1/f_s$  称为奈奎斯特速率。

723 然而, 你可能会问, 由于成像芯片实际上是对有限区域内的光场进行平均, 点采样的结果是否仍  
724 然适用? 传感器区域的平均值确实会衰减一些较高的频率。然而, 即使填充因子为 100%, 如图 2.25  
725 的右图所示, 高于奈奎斯特极限的频率 (采样频率的一半) 仍会产生混叠信号, 尽管其幅度小于相  
应的带限信号。一个更令人信服的论据, 为什么混叠是糟糕的, 可以通过使用低质量滤波器 (如方

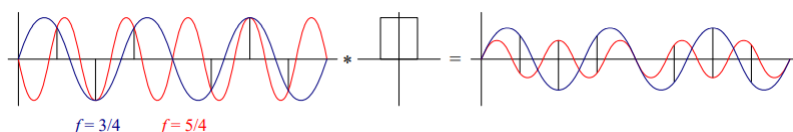


图 2-25 一维信号的混叠: 在  $f = 2$  处采样时,  $f = 3/4$  处的蓝色正弦波和  $f = 5/4$  处的红色正弦波具有相同的数字采样。即使经过 100% 填充因子盒滤波器的卷积, 两个信号, 虽然不再是相同的幅度, 仍然是混叠的意义上, 采样的红色信号看起来像一个反向的低幅度版本的蓝色信号。(右边的图像被放大了, 以便更好地显示。实际的正弦值分别是其原始值的 30% 和 -18%。)

726

727 块 (正方形) 滤波器) 对信号进行降采样来看出。图 2.26 显示了高频啁啾图像 (之所以称为啁啾图  
728 像是因为频率随时间增加), 以及使用 25% 填充因子面积传感器、100% 填充因子传感器和高质量 9

<sup>①</sup> [17] An alias is an alternate name for someone, so the sampled signal corresponds to two different aliases.

<sup>18</sup> The actual theorem states that  $f_s$  must be at least twice the signal bandwidth but, since we are not dealing with modulated signals such as radio waves during image capture, the maximum frequency suffices.

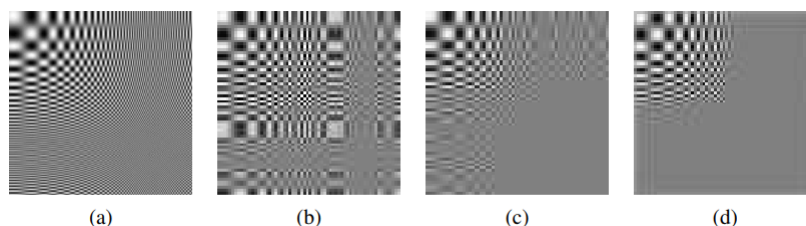


图 2-26 二维信号的混叠:(a) 原始全分辨率图像;(b) 下采样 4×, 采用 25% 填充因子盒滤波器;(c) 下采样 4×, 采用 100% 填充因子盒滤波器;(d) 下采样 4×, 采用高质量的 9 抽头滤波器。请注意, 更高的频率是如何与较低质量的滤波器混叠成可见的频率, 而 9 抽头滤波器完全删除这些更高的频率。

抽头滤波器对其进行采样的结果。下采样（抽取）滤波器的其他示例见第 3.5.2 节和图 3.29。

预测成像系统（甚至图像处理算法）将产生的混叠量的最佳方法是估计点扩散函数（PSF），它表示特定像素传感器对理想点光源的响应。PSF 是由光学系统（透镜）引起的模糊和芯片传感器的有限积分区域的组合（卷积）。<sup>19</sup>

如果我们知道镜头的模糊函数和成像芯片的填充因子（传感器区域形状和间距）（另外，可选地，抗混叠滤波器的响应），我们可以卷积这些（如第 3.2 节所述）以获得 PSF。图 2.27a 显示了透镜的 PSF 的一维横截面，其模糊函数假设为半径等于像素间距  $s$  的圆盘加上水平填充因子为 0.8 的传感芯片。取这个 PSF 的傅里叶变换（第 3.4 节），我们得到调制传递函数（MTF），从中我们可以估计出混叠量为  $f \leq f_s$  奈奎斯特频率之外的傅里叶幅度的面积<sup>20</sup>。如果我们对镜头进行散焦，使模糊函数的半径为  $2s$ （图 2.27c），我们会看到混叠量显著减少，但图像细节量也会减少（频率更接近  $f = f_s$ ）。

在实验室条件下，可以通过观察点光源（如从后面照亮的一张黑色纸板上的针孔）来估计 PSF（达到像素精度）。然而，该 PSF（针孔的实际图像）仅精确到像素分辨率，并且虽然它可以模拟更大的模糊（例如由散焦引起的模糊），但是它不能模拟 PSF 的子像素形状并预测混叠量。第 10.1.4 节中描述的另一种技术是查看校准模式（例如，由倾斜阶梯边缘组成的模式（Reichenbach, Park, and Narayanswamy 1991; Williams and Burns 2001; Joshi, Szelski, and Kriegman 2008）），其理想的外观可以重新合成到亚像素精度。

除了在图像采集期间发生之外，混叠还会引入各种图像处理操作中，例如重采样、上采样和下采样。第 3.4 节和第 3.5.2 节讨论了这些问题，并展示了如何仔细选择过滤器可以减少操作注入的混叠量。

<sup>19</sup>成像芯片通常在成像芯片前插入一个光学抗混叠滤波器，以减少或控制混叠的数量。

<sup>20</sup>光子晶体光纤的复傅里叶变换实际上被称为光学传递函数（OTF）（Williams 1999）。其幅度被称为调制传递函数（MTF），其相位被称为相位传递函数（PTF）。

### 2.3.2 颜色

在 2.2 节中, 我们看到照明和表面反射是波长的函数。当入射光撞击成像传感器时, 来自光谱不同部分的光以某种方式集成到我们在数字图像中看到的离散的红、绿、蓝 (RGB) 颜色值中。这个过程是如何工作的, 我们如何分析和操作颜色值?

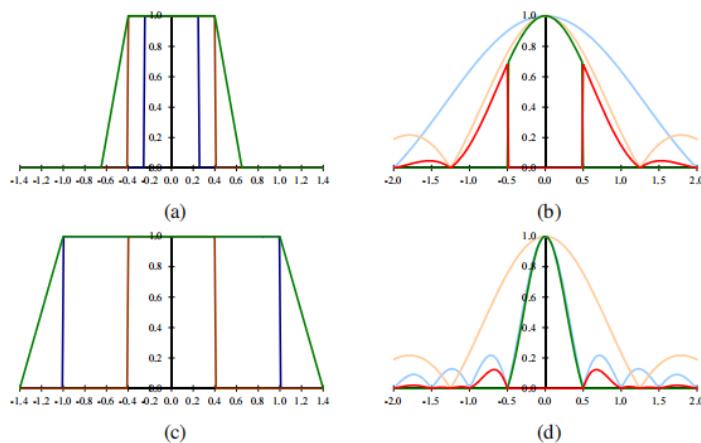


图 2-27 采样点扩散函数 (PSF): (a) 中模糊盘 (蓝色) 的直径等于像素间距的一半, 而 (c) 中的直径是像素间距的两倍。传感芯片的水平填充系数为 0.8, 以棕色显示。这两个核的卷积给出了点扩散函数, 用绿色表示。功率谱的傅里叶响应在 (b) 和 (d) 中绘出。发生混叠的奈奎斯特频率以上区域以红色显示。

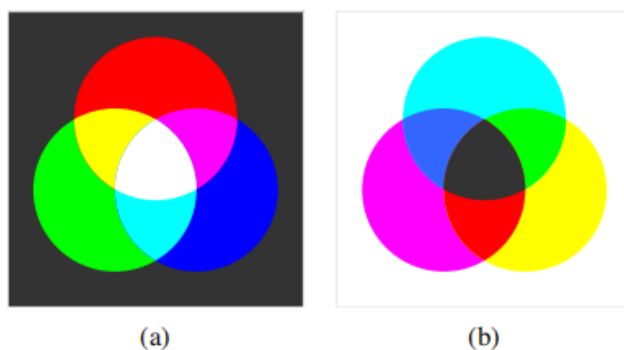


图 2-28 原色和二次色: (a) 红色、绿色和蓝色的附加色可以混合产生青色、品红色、黄色和白色; (b) 减色青色、品红色和黄色可以混合产生红色、绿色、蓝色和黑色。

你可能还记得童年时混合颜料以获得新颜料的神奇过程。你可能还记得, 蓝色 + 黄色代表绿色,



红色 + 蓝色代表紫色, 红色 + 绿色代表棕色。如果你再早些时候再讨论这个话题, 你可能会发现合适的减色法原色实际上是青色 (一种浅蓝色绿色)、品红色 (粉色) 和黄色 (图 2.28b), 尽管黑色也经常用于四色印刷 (CMYK)<sup>21</sup>。如果你后来参加了任何绘画课程, 你会发现颜色可以有更奇特的名字, 如浅红色、蓝色和红色。减色剂被称为减色剂, 因为涂料中的颜料吸收光谱中的某些波长。

稍后, 您可能已经了解了附加原色 (红色、绿色和蓝色) 以及如何添加它们 (用幻灯机或在计算机显示器上) 来产生青色、品红色、黄色、白色以及我们在电视机和显示器上通常看到的所有其他颜色 (图 2.28a)。

通过什么过程, 红、绿两种不同的颜色有可能相互作用产生第三种颜色, 如黄色? 波长是否以某种方式混合在一起产生了新的波长?

你可能知道正确答案与物理混合波长无关。相反, 三原色的存在是人类视觉系统的三刺激 (或三色) 性质的结果, 因为我们有三种不同的视锥细胞, 每种视锥细胞选择性地对光谱的不同部分做出反应 (Glassner 1995; Wandell 1995; Wyszecki and Stiles 2000; Livingstone 2008; Frisby and Stone 2010; Reinhard, Heidrich et al. 2010; Fairchild 2013)<sup>22</sup>。请注意, 对于机器视觉应用, 如遥感和地形分类, 最好使用更多的波长。同样, 监视应用通常可以受益于近红外范围的传感。

### CIE RGB 和 XYZ

为了测试和量化感知的三色理论, 我们可以尝试将所有单色 (单波长) 颜色复制为三种适当选择的原色的混合。(使用棱镜或专门制造的滤色器可以获得纯波长的光。) 在 20 世纪 30 年代, 国际照明委员会 (CIE) 通过使用红色 (700.0 纳米波长)、绿色 (546.1 纳米波长) 和蓝色 (435.8 纳米波长) 的原色进行这种颜色匹配实验, 对 RGB 表示进行了标准化。

图 2.29 显示了用标准观察者进行这些实验的结果, 即对大量受试者的感知结果进行平均<sup>23</sup>。您会注意到, 对于蓝-绿范围内的某些纯光谱, 必须添加负数量的红光, 也就是说, 必须向要匹配的颜色添加一定数量的红光, 才能获得颜色匹配。这些结果也为变色体的存在提供了一个简单的解释, 变色体是具有不同光谱的颜色, 在感知上是无法区分的。请注意, 在一种光线下是混合色的两种织物或油漆颜色在不同的光线下可能不再是混合色。

由于与混合负光相关的问题, CIE 还开发了一种新的颜色空间, 称为 XYZ, 它包含了其正八分之一范围内的所有纯光谱颜色。(它还将 Y 轴映射到亮度, 即感知的相对亮度, 并将纯白映射到对角 (等值) 向量。) 从 RGB 到 XYZ 的转换由下式给出

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.104)$$

<sup>21</sup>可以使用橙色、绿色和紫色等附加油墨来进一步扩展色域。

<sup>22</sup>另请参见马克·费尔柴尔德的网页, <http://markfairchild.org/WhyIsColor/books·links.html>。

<sup>23</sup>正如迈克·布朗在他关于颜色的教程 (布朗 2019) 中指出的那样, 标准观察者实际上是一个在 20 世纪 20 年代只接受了 17 个英国科目的普通人。

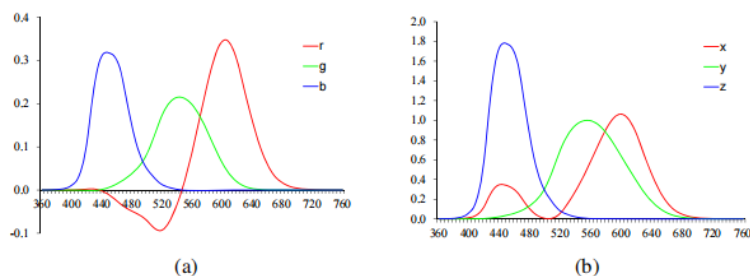


图 2-29 标准 CIE 颜色匹配函数:(a) 通过将纯色匹配到  $R = 700.0$  纳米、 $G = 546.1$  纳米和  $B = 435.8$  纳米原色而获得的  $\bar{r}(\lambda)$ 、 $\bar{g}(\lambda)$ 、 $\bar{b}(\lambda)$  颜色光谱; (b) 图是  $\bar{x}(\lambda)$ 、 $\bar{y}(\lambda)$ 、 $\bar{z}(\lambda)$  颜色匹配函数, 它们是  $(\bar{r}(\lambda)$ 、 $\bar{g}(\lambda)$ 、 $\bar{b}(\lambda))$  光谱的线性组合。

虽然 CIE XYZ 标准的官方定义对矩阵进行了归一化, 使得对应于纯红色的 Y 值为 1, 但更常用的形式是省略前导分数, 使得第二行加起来为 1, 即 RGB 三元组 (1, 1, 1) 映射为 Y 值 1。根据 (2.104), 线性混合图 2.29a 中的  $(\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda))$  曲线, 我们获得了图 2.29b 中所示的结果  $(\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda))$  曲线。请注意, 现在所有三个光谱 (颜色匹配函数) 都只有正值, 并且  $\bar{y}(\lambda)$  曲线如何与人类感知的亮度相匹配。

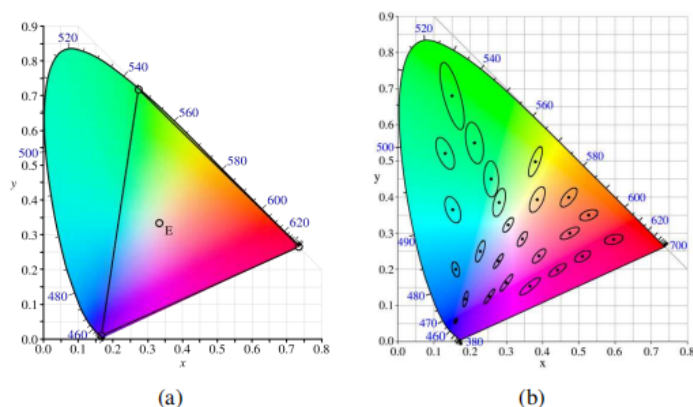


图 2-30 CIE 色度图, 显示沿周线的纯单波长光谱颜色和沿相应 (x, y) 值绘制的 E 白点。(a) 红色、绿色和蓝色原色不跨越整个色域, 因此需要添加负数量的红色来跨越蓝绿色范围; (2) 碎石椭圆显示了具有相同可辨别性的颜色区域, 并构成了实验室感知颜色空间的基础。

如果我们把 XYZ 值除以  $X+Y+Z$  的和, 就得到色度坐标

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z}, \quad (2.105)$$

总计为 1。色度坐标丢弃给定颜色样本的绝对强度，仅代表其纯色。如果我们将图 2.29b 中的单色  $\lambda$  参数从  $\lambda$  纳米扫描到  $\lambda = 800$  纳米，我们得到了图 2.30a 中所示的熟悉的色度图。该图显示了大多数可感知的每个颜色值的  $(x, y)$  值。(当然，本书中的 CMYK 复制过程实际上并没有跨越可感知颜色的整个色域。) 外曲线边缘表示所有纯单色颜色值在  $(x, y)$  空间中的映射位置，而连接两个端点的下直线被称为紫色线。插图三角形跨越了原始配色实验中使用的红色、绿色和模糊单波长原色，而 E 表示白点。

因此，当我们想要区分亮度和色度时，一个方便的颜色值表示是  $Y_{xy}$ (亮度加上两个最明显的色度分量)。

### $L^*a^*b^*$ 颜色空间

虽然 XYZ 颜色空间有许多方便的属性，包括区分亮度和色度的能力，但它实际上并不能预测人类对颜色或亮度差异的感知程度。

因为人类视觉系统的响应大致是对数的(我们可以感觉到大约 0.01 的相对亮度差异)，所以 CIE 定义了一个称为  $L^*a^*b^*$ (有时也称为 CIELAB) 的 XYZ 空间的非线性重映射，其中亮度或色度的差异在感觉上更加均匀，如图 2.30b 所示<sup>24</sup>。

亮度的  $L^*$  分量定义为

$$L^* = 116f\left(\frac{Y}{Y_n}\right) \quad (2.106)$$

其中， $Y_n$  是标称白色的亮度值(飞兆半导体 2013 年)并且

$$f(t) = \begin{cases} t^{1/3} & t > \delta^3 \\ t/(3\delta^2) + 2\delta/3 & \text{else} \end{cases} \quad (2.107)$$

是立方根的有限斜率近似， $\delta = 6/29$ 。生成的 0...100 标度大致测量等量的亮度感知。

以类似的方式， $a^*$  和  $b^*$  组件被定义为

$$a^* = 500 \left[ f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \quad \text{and} \quad b^* = 200 \left[ f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right] \quad (2.108)$$

其中， $(X_n, Y_n, Z_n)$  是测得的白点。图 2.33 I-k 显示了样本彩色图像的  $L^*a^*b^*$  表示。

### 彩色照相机

虽然前面的讨论告诉我们如何唯一地描述任何颜色的感知三刺激描述(光谱分布)，但它没有告诉我们 RGB 静态和视频相机实际上是如何工作的。他们只是测量红色(700.0 纳米)、绿色(546.1 纳米)和蓝色(435.8 纳米)标称波长的光量吗？彩色显示器仅仅发射这些波长吗？如果是这样，它们如何发射负红光来再现青色范围内的颜色？

事实上，RGB 摄像机的设计在历史上一直是基于电视机中彩色荧光粉的可用性。当标准清晰度彩色电视(NTSC)被发明时，驱动阴极射线管(CRT)中三个彩色电子枪的 RGB 值和明确定义感知颜

<sup>24</sup>另一个被称为  $L^*a^*b^*$  的感知驱动的颜色空间被同时开发和标准化(飞兆半导体 2013)。

色的 XYZ 值之间的映射被定义 (该标准被称为 ITU-R BT.601)。随着高清晰度电视和较新的显示器的出现, 一种称为 ITU-R BT.709 的新标准产生了, 它规定了每种原色的 XYZ 值,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix} \quad (2.109)$$

实际上, 每个彩色摄像机根据其红、绿、蓝传感器的光谱响应函数来整合光,

$$\begin{aligned} R &= \int L(\lambda) S_R(\lambda) d\lambda \\ G &= \int L(\lambda) S_G(\lambda) d\lambda \\ B &= \int L(\lambda) S_B(\lambda) d\lambda \end{aligned} \quad (2.110)$$

其中,  $L(\lambda)$  是给定像素处的入射光谱, 而  $\{S_R(\lambda), S_G(\lambda), S_B(\lambda)\}$  是相应传感器的红、绿、蓝光谱灵敏度。

我们能说出相机实际上有什么光谱灵敏度吗? 除非相机制造商向我们提供这些数据, 或者我们观察相机对整个单色光光谱的响应, 否则这些灵敏度不是由标准 (如 BT.709) 规定的。相反, 重要的是给定颜色的三刺激值产生指定的 RGB 值。制造商可以自由使用灵敏度与标准 XYZ 定义不匹配的传感器, 只要它们以后可以转换 (通过线性变换) 为标准颜色。

类似地, 虽然电视和计算机显示器应该产生由等式 (2.109) 指定的 RGB 值, 但是没有理由它们不能使用数字逻辑来将输入的 RGB 值转换成不同的信号来驱动每个颜色通道<sup>①</sup>。正确校准的显示器使这些信息可用于执行色彩管理的软件应用程序, 从而使现实生活中的颜色、屏幕上的颜色和打印机上的颜色尽可能匹配。

### 彩色滤光片阵列

虽然早期的彩色电视摄像机使用三个摄像机 (管) 来执行传感, 后来的摄像机使用三个独立的 RGB 传感芯片, 但今天的大多数数字静态和视频摄像机使用滤色器阵列 (CFA), 其中交替的传感器被不同的滤色器覆盖 (图 2.24)。<sup>26</sup>

今天彩色相机中最常用的模式是拜耳模式 (拜耳 1976), 它在一半的传感器上放置绿色滤光片 (棋盘模式), 在其余的传感器上放置红色和蓝色滤光片 (图 2.31)。绿色滤光片的数量是红色和蓝色滤光片的两倍, 这是因为亮度信号主要由绿色值决定, 视觉系统对亮度中的高频细节比对色度中的高频细节更敏感 (这一事实在彩色图像压缩中得到利用——见第 2.3.3 节)。对缺失的颜色值进行插值以使所有像素都具有有效的 RGB 值的过程称为去马赛克, 详见第 10.3.1 节。

<sup>①</sup> 最新的有机发光二极管电视显示器现在引入了更高的动态范围 (HDR) 和更宽的色域 (WCG), <https://www.cnet.com/how-to/what-is-wide-color-gamut-wcg/>.

<sup>26</sup> Foveon 的芯片设计将红色、绿色和蓝色传感器堆叠在一起, 但从未得到广泛采用。在 [https://en.wikipedia.org/wiki/Color 滤色器阵列](https://en.wikipedia.org/wiki/Color滤色器阵列) 中可以找到多年来提出的替代滤色器阵列的描述。

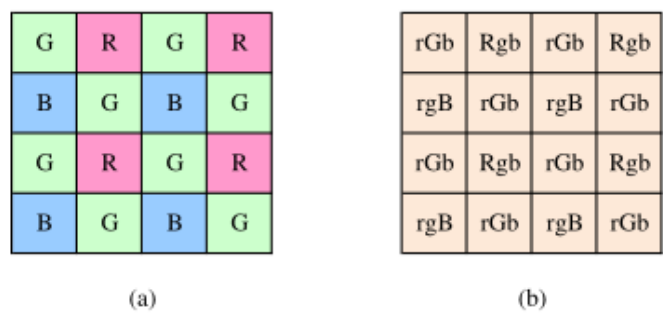


图 2-31 拜耳 RGB 模式:(一) 彩色滤光片阵列布局; (b) 插值像素值, 未知 (猜测) 值显示为小写。

同样, 彩色液晶显示器通常使用红色、绿色和蓝色交替条纹的滤光器来模拟全色显示的体验。和以前一样, 因为视觉系统在亮度上比色度具有更高的分辨率 (锐度), 所以可以对 RGB (和单色) 图像进行数字预滤波, 以增强对清晰度的感知 (Betrissey, Blinn 等人, 2000; 普拉特 2000b)。

色彩平衡

在对感测到的 RGB 值进行编码之前, 大多数相机会执行某种颜色平衡操作, 试图将给定图像的白点移动到更接近纯白 (相等的 RGB 值) 的位置。如果颜色系统和照明相同 (BT.709 系统使用日光照明 D65 作为其参考白色), 变化可能很小。然而, 如果光源是强烈着色的, 例如室内白炽灯 (通常会产生黄色或橙色), 补偿可能非常显著。

执行颜色校正的简单方法是将每个 RGB 值乘以不同的因子 (即, 对 RGB 颜色空间应用对角矩阵变换)。更复杂的变换, 有时是映射到 XYZ 空间的结果, 实际上是执行颜色扭曲, 即它们使用一般的  $3 \times 3$  颜色变换矩阵。<sup>27</sup>练习 2.9 让您探索其中一些问题。

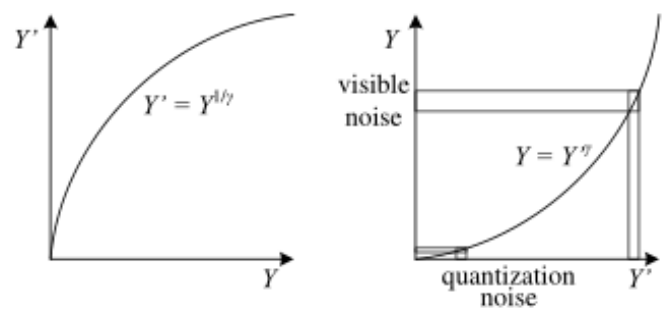


图 2-32 拜耳 RGB 模式:(一) 彩色滤光片阵列布局; (b) 插值像素值, 未知 (猜测) 值显示为小写。

<sup>27</sup>你们中那些年龄足够大, 还记得早期彩色电视的人, 自然会想到电视机上的色调调节旋钮, 这可能会产生真正奇怪的结果。



## Gamma

在黑白电视的早期,用于显示电视信号的阴极射线管中的磷光体对其输入电压做出非线性响应。电压和产生的亮度之间的关系用一个叫做  $\gamma$  的数字来描述,因为公式是粗略的

$$B = V^\gamma \quad (2.111)$$

$\gamma$  约为 2.2。为了补偿这种影响,电视摄像机中的电子设备将通过逆伽马预先映射感测到的亮度,典型值为  $\frac{1}{\gamma} = 0.45$ 。

$$Y' = Y^{\frac{1}{\gamma}} \quad (2.112)$$

信号在传输前通过这种非线性的映射有一个有益的副作用: 在传输过程中增加了噪声 (记住,这是模拟时代!) 会在信号较暗的区域 (在接收器处应用伽马射线后) 降低,在该区域信号更容易看到 (图 2.32)。<sup>28</sup>(请记住,我们的视觉系统对亮度的相对差异大致敏感。)

当彩色电视发明时,人们决定在将红、绿、蓝信号组合起来进行编码之前,将它们分别通过相同的伽马非线性。今天,即使我们的传输系统中不再有模拟噪声,信号在压缩过程中仍然是量化的 (参见第 2.3.3 节),因此对检测值应用逆伽马仍然是有用的。

不幸的是,对于计算机视觉和计算机图形学来说,图像中伽马的存在往往是有问题的。例如,辐射现象 (如阴影) 的正确模拟 (见第 2.2 节和等式 (2.88)) 发生在线性辐射空间中。完成所有计算后,应在显示前应用适当的伽马。不幸的是,许多计算机图形系统 (如阴影模型) 直接对 RGB 值进行操作,并直接显示这些值。(幸运的是,较新的彩色成像标准,如 16 位 sRGB,使用线性空间,这使得这不是一个问题 (Glassner 1995)。)

在计算机视觉中,情况可能更令人生畏。使用光度立体 (第 13.1.1 节) 等技术或更简单的操作 (如精确的图像去模糊) 来精确确定表面法线,要求测量在强度的线性空间中进行。因此,当执行诸如这些的详细定量计算时,必须首先撤销感测颜色值中的伽马和每图像颜色重新平衡。Chakrabarti、Scharstein 和 Zickler (2009) 开发了一个复杂的 24 参数模型,与当今数码相机执行的处理非常匹配;他们还提供了一个彩色图像数据库,您可以在自己的测试中使用。

然而,对于其他视觉应用,例如特征检测或立体和运动估计中的信号匹配,这种线性化步骤通常是不必要的。事实上,确定是否有必要撤销伽马射线需要仔细考虑,例如,在补偿图像拼接中的曝光变化的情况下 (参见练习 2.7)。

果所有这些处理步骤听起来都让模型感到困惑,那么它们就是。练习 2.10 让您尝试使用经验调查来梳理其中的一些现象,即拍摄彩色图表并比较原始和 JPEG 压缩的颜色值。

## 其他颜色空间

虽然 RGB 和 XYZ 是用于描述颜色信号的光谱内容 (以及因此的三刺激响应) 的原色空间,但是在视频和静止图像编码以及计算机图形学中已经开发了各种其他表示。

为视频传输开发的最早的彩色表示是为北美的 NTSC 视频开发的 YQ 标准和为欧洲的 PAL 开

<sup>28</sup>一种叫做压扩的相关技术是用于录音带的杜比降噪系统的基础。

发的密切相关的 YUV 标准。在这两种情况下,都希望有一个与常规黑白电视信号相当的亮度通道 Y(之所以如此称呼,是因为它只是粗略地模拟了真实亮度),以及两个较低频率的色度通道。

在这两种系统中, Y 信号 (或者更恰当地说,  $Y'$  亮度信号, 因为它是伽马压缩的) 是从以下获得的

$$Y'_{601} = 0.299R' + 0.587G' + 0.114B' \quad (2.113)$$

其中  $R'G'B'$  是伽玛压缩颜色分量的三元组。当在 BT.709 中使用较新的高清电视颜色定义时, 公式为

$$Y'_{709} = 0.2125R' + 0.7154G' + 0.0721B' \quad (2.114)$$

紫外线分量来自  $(B' - Y')$  和  $(R' - Y')$  的比例版本, 即:

$$U = 0.492111(B' - Y') \text{ and } V = 0.877283(R' - Y') \quad (2.115)$$

而智商分量是旋转 33 度的紫外线分量。在复合 (NTSC 和 PAL) 视频中, 色度信号在被调制和叠加到  $Y'$  亮度信号上之前, 先被水平低通滤波。向后兼容是通过让旧的黑白电视机有效地忽略高频色度信号 (因为电子速度慢) 或者在最坏的情况下, 将其作为高频模式叠加在主信号上来实现的。

虽然这些转换在计算机视觉的早期很重要, 当时帧抓取器将直接数字化复合电视信号, 但今天所有的数字视频和静止图像压缩标准都基于更新的 YCbCr 转换。YCbCr 与 YUV 密切相关 ( $C_B$  和  $C_R$  信号携带蓝色和红色色差信号, 比 UV 具有更有用的助记符), 但使用不同的比例因子来适应数字信号的 8 位范围。

对于视频,  $Y'$  信号被重新缩放以适合 [16...235] 值的范围, 而 Cb 和 Cr 信号被缩放以适合 [16...240] (Gomes 和 Velho 1997 Fairchild 2013)。对于静止图像, JPEG 标准使用没有保留值的全部八位范围,

$$\begin{bmatrix} Y' \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \quad (2.116)$$

其中, “绿蓝”值是 8 位伽马压缩的颜色分量 (即, 当我们打开或显示一幅 JPEG 图像时, 我们获得的实际 RGB 值)。对于大多数应用来说, 这个公式并不那么重要, 因为你的图像读取软件会直接为你提供八位伽马压缩的  $R'G'B'$  值。但是, 如果您试图仔细进行图像去块 (练习 4.1), 这些信息可能会很有用。

您可能会遇到的另一个颜色空间是色调、饱和度、值 (HSV), 它是 RGB 颜色立方体在非线性的色度角度、径向饱和度百分比和亮度启发值上的投影。更详细地说, 值被定义为平均或最大颜色值, 饱和度被定义为与对角线的比例距离, 色调被定义为色轮周围的方向 (霍尔 (1989) 描述了确切的公式); 休斯、范·达姆等人 (2013 年); Brown (2019))。这种分解在图形应用中是非常自然的, 例如颜色选择 (它近似于用于颜色描述的孟塞尔图)。图 2.33 l-n 显示了一个样本彩色图像的 HSV 表示, 其中饱和度使用灰度进行编码 (饱和 = 较暗), 色调被描述为一种颜色。



如果您希望您的计算机视觉算法只影响图像的值(亮度), 而不影响其饱和度或色调, 一个更简单的解决方案是使用(2.105)中定义的  $Y_x y$ (亮度 + 色度) 坐标或更简单的颜色比率,

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B} \quad (2.117)$$

(图 2.33 e-h)。在操纵亮度(2.113)之后, 例如, 通过直方图均衡化过程(第 3.1.4 节), 您可以将新亮度与旧亮度的比率乘以每个颜色比率, 以获得调整后的 RGB 三重图。

虽然所有这些颜色系统听起来可能令人困惑, 但最终, 你使用哪一个可能并不重要。波因顿在他的色彩常见问题解答中, <http://www.poynton.com/ColorFAQ.html> 指出, 感知驱动的  $L^*a^*b^*$  系统在质量上类似于我们主要处理的伽马压缩的  $R'G'B'$  系统, 因为两者在实际强度值和被操纵的数字之间都有分数幂标度(近似于对数响应)。和所有情况一样, 在决定使用哪种技术之前, 仔细考虑你正在努力实现的目标。

### 2.3.3 压缩

相机处理管道的最后一个阶段通常是某种形式的图像压缩(除非你使用无损压缩方案, 如相机 RAW 或 PNG)。

所有彩色视频和图像压缩算法首先将信号转换为 YCbCr(或一些密切相关的变体), 这样它们可以以比色度信号更高的保真度压缩亮度信号。(回想一下, 人类视觉系统对颜色的频率响应要比对亮度变化的频率响应差。)在视频中, 通常对 Cb 和 Cr 进行水平分二的次采样; 对于静止图像(JPEG), 子采样(平均)同时在水平和垂直方向进行。

一旦对亮度和色度图像进行适当的下采样, 并将分割成单独的图像, 它们将被传递到块变换阶段。这里最常用的技术是离散余弦变换(DCT), 它是离散傅里叶变换(DFT)的实值变体(见 3.4.1 节)。DCT 是一个合理的 approximation to the Karhunen-Loève or eigenvalue decomposition of natural image patches, 即, 同时将最大能量打包到第一个系数并对角化像素之间的联合协方差矩阵(使变换系数在统计上独立)的分解。MPEG 和 JPEG 都使用  $8 \times 8$  DCT 变换(Wallace 1991; Le Gall 1991), 尽管更新的变种, 包括新的 AV1 开放标准<sup>29</sup>, 使用更小的  $4 \times 4$  甚至  $2 \times 2$  块。可选的转换, 如小波(Taubman 和 Marcellin 2002)和叠接变换(Malvar 1990, 1998, 2000)被用于压缩标准, 如 JPEG 2000 和 JPEG XR。

经过变换编码后, 系数值被量化成一组小整数, 可以使用可变位长方案编码, 如 Huffman 码或算术码(Wallace 1991; Marpe, Schwarz 和 Wiegand 2003)。(直流(最低频率)系数也从前一个区块的直流值自适应预测。术语“DC”来源于“直流电”, 即信号的非正弦或非交流部分。量化中的步长是由 JPEG 文件的质量设置控制的主要变量(图 2.34)。

一个压缩算法的质量通常是通过其峰值信噪比(PSNR)来报告的, 这是由平均均方误差,

$$MSE = \frac{1}{n} \sum_{\mathbf{x}} [I(\mathbf{x}) - \hat{I}(\mathbf{x})]^2 \quad (2.118)$$

<sup>29</sup><https://aomedia.org/>

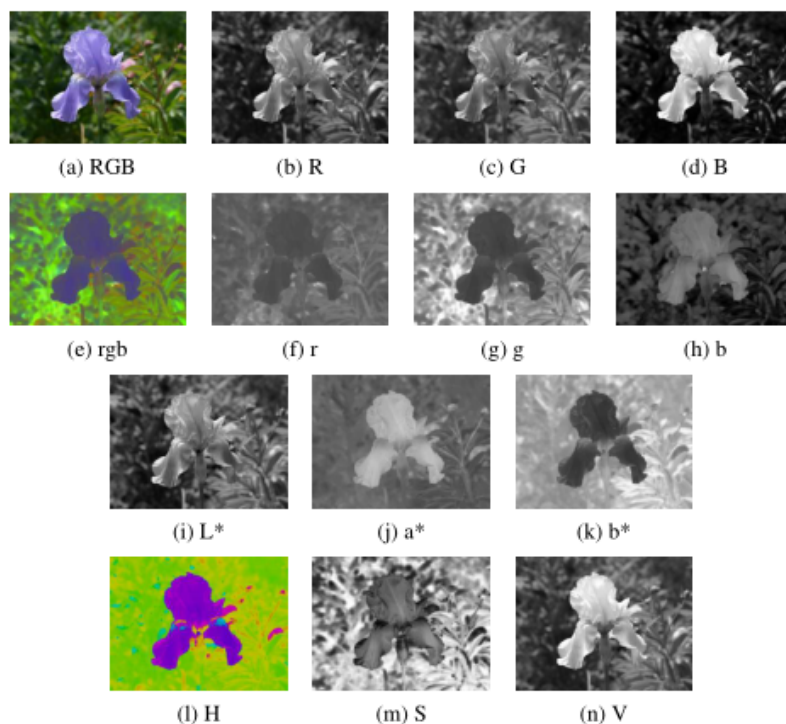


图 2-33 颜色空间转换:(a-b)RGB;(e-h)rgb。(i-k) $L^*a^*b^*$ ;(l-n)HSV。请注意，rgb, $L^*a^*b^*$ 和 HSV 值都被重新缩放以适应打印页面的动态范围。

其中  $I(x)$  是原始的未压缩图像， $\hat{I}(x)$  是其压缩后的对应图像，或等价地，均方根误差 (RMS 误差)，其定义为

$$RMS = \sqrt{MSE} \quad (2.119)$$

PSNR 定义为

$$PSNR = 10 \log_{10} \frac{I_{\max}^2}{MSE} = 20 \log_{10} \frac{I_{\max}}{RMS} \quad (2.120)$$

其中， $I_{\max}$  最大的信号范围，例如，255 为 8 位图像。虽然这只是图像压缩工作原理的一个高级草图，但是理解这些技术引入的伪影是很有用的，以便在各种计算机视觉应用程序中得到补偿。还需要注意的是，研究人员目前正在开发基于深度神经网络的新型图像和视频压缩算法，例如 (Rippel 和 Bourdev 2017; Mentzer, Agustsson 等人 2019; Rippel, Nair 等人 2019) 和 <http://www.compression.cc/>。看看这些技术产生了什么样的不同工件是很有趣的。

## 2.4 补充阅读

正如我们在本章开头提到的, 这本书提供了一个非常丰富和深刻的主题集的简短总结, 传统上涵盖了许多独立的领域。

更多关于点、线、面和投影的几何介绍可以在多视图几何的教科书中找到 (Faugeras 和 Luong 2001; Hartley 和 Zisserman 2004) 和计算机图形学 (Watt 1995; OpenGL-ARB 1997; Hughes, van Dam 等人, 2013; Marschner 和 Shirley 2015)。更深入的主题包括高阶原语, 如二次曲线、二次曲线和三次曲线, 以及三视图和多视图几何。

图像形成(合成)过程传统上作为计算机图形课程的一部分来教授 (Glassner 1995; 瓦特 1995; Hughes, van Dam 等人, 2013; Marschner 和 Shirley 2015), 但它也在基于物理的计算机视觉中进行了研究 (Wolff, Shafer, 和 Healey, 1992a)。对相机镜头系统的行为进行了光学研究 (Möller 1988; 雷 2002; 赫克特 2015 年)。

Healey 和 Shafer(1992) 写了一些关于颜色理论的好书; 万德尔 (1995); Wyszecki 和 Stiles (2000); 仙童 (2013), 与利文斯通 (2008) 提供了一个更有趣和非正式的介绍色彩感知的主题。Mark Fairchild 的彩色书籍页面和链接<sup>30</sup>列出了许多其他资源。

有关采样和混叠的主题在信号和图像处理的教科书中有涉及 (Crane 1997; Jahne 1997; Oppenheim 和 Schafer 1996; Oppenheim, Schafer 和 Buck 1999; 普惠 2007; 俄国人 2007; Burger and Burge 2008; 冈萨雷斯和伍兹 2017)。

包括上述许多主题(图像形成, 镜头, 颜色和采样理论)的精彩细节的两个课程是 Marc Levoy 的数字摄影课程在斯坦福大学 (Levoy 2010) 和 Michael Brown 的教程在 ICCV 2019 (Brown 2019) 的图像处理管道。

## 2.5 练习题

**给学生的提示:** 这一章是相对轻的练习, 因为它包含了大部分背景材料和没有那么多有用的技术。如果你真的想了解多视角几何的通道, 我鼓励你阅读和做 Hartley 和 Zisserman(2004) 提供的练习。同样, 如果你想要一些与图像形成过程相关的练习, Glassner(1995) 的书中充满了具有挑战性的问题。

**例 2.1: 最小二乘交点和直线拟合超前。**式 (2.4) 给出了假设两条二维直线表示为齐次坐标, 两条直线的交点如何表示为它们的叉积。

1. 如果给你多于两行, 并想找到一个点  $\tilde{x}$  来最小化到每一行的平方距离之和,

$$D = \sum_i (\tilde{\mathbf{x}} \cdot \tilde{\mathbf{l}}_i)^2, \quad (2.121)$$

你怎么计算这个量?(提示: 属于  $\tilde{x}^T \tilde{I}_i$ , 并将数量的平方转化为二次型, 属于  $\tilde{x}^T A \tilde{x}$ 。)

<sup>30</sup><http://markfairchild.org/WhyIsColor/books/links.html>.

2. 为了将一条直线拟合到一堆点上, 你可以计算点的质心 (平均值) 以及这个平均值周围点的协方差矩阵。表明沿协方差椭圆 (最大特征向量) 的长轴通过质心的线使到这些点的平方距离的和最小化。

3. 这两种方法本质上是不同的, 尽管射影对偶性告诉我们点和线是可以互换的。为什么这两种算法如此明显地不同? 他们真的在减少不同的目标吗?

**例 2.2: 2D 变换编辑器。**编写一个程序, 让您可以交互式地创建一组矩形, 然后修改它们的“姿态”(2D 变换)。你应执行以下步骤:

1. 打开一个空窗口 (“画布”)。2. Shift 拖动 (橡皮筋) 创建一个新的矩形。3. 选择变形模式 (运动模型): 平移、刚性、相似、仿射或透视。4. 拖动轮廓线的任意一角以更改其转换。这个练习应该建立在一组像素坐标和转换类上, 可以自己实现, 也可以从软件库实现。也应该支持创建的表示的持久性 (保存和加载)(对于每个矩形, 保存其转换)。

**例 2.3: 3D 查看器。**编写一个简单的 3D 点、线和多边形查看器。导入一组点和线命令 (原语) 以及一个查看转换。交互式地修改对象或摄像机变换。这个查看器可以是您在 (练习 2.2) 中创建的查看器的扩展。简单地用它们的 3D 等价替换视图转换。

(可选) 添加一个 z-buffer 来移除多边形的隐藏表面。

(可选) 使用 3D 绘图包并编。

**例 2.4: 焦距和景深。**例 2.4: 焦距和景深。计算出镜头上的焦距和景深指标是如何确定的。

1. 计算并绘制焦距  $z_o$  作为焦距  $f$  (例如 100mm) 从焦距  $\Delta z_i = f - z_i$  移动的距离的函数。这是否解释了你在一个典型的透镜上看到的焦距的双曲线变化 (图 2.20)?

2. 计算一个给定的焦点设置  $z_{oas}$  的视场深度 (最小和最大焦距), 混淆直径  $c$  (使它成为传感器宽度的一部分) 的函数, 焦距  $f$ , 以及  $f$  光圈编号  $N$  (与光圈直径  $d$  有关)。这是否解释了像图 2.20a 所示的支架镜框上通常的景深标记?

3. 现在考虑一个焦距不同的变焦镜头  $f$ 。假设当你变焦时, 镜头保持焦距, 也就是说, 从后节点到传感器平面的距离  $z_i$  自动调整为固定的焦距  $z_o$ 。景深指标如何随着焦距的变化而变化? 你能复制一个二维图来模拟图 2.20b 中镜头上的视力线的弯曲深度吗?

**例 2.5:  $f$  值和快门速度。**列出常见的  $f$  值和你的相机提供的快门速度。在老款单反相机上, 可以在镜头和快门速度表盘上看到它们。在新的相机上, 你必须看电子取景器 (或 LCD 屏幕/指示灯), 当你手动调整曝光。

1. 这些是几何级数吗? 如果有, 比率是多少? 这些与暴露值 (EVs) 有什么关系?

2. 如果你的相机有 1/60 和 1/125 的快门速度, 你认为这两个速度是两倍还是  $125/60 = 2.083$  倍?

3. 你认为这些数字有多准确? 你能设计出一种方法来精确测量光圈是如何影响到达传感器的光量的以及准确的曝光时间是多少吗?

**例 2.6: 噪音声级校正。**用安装在三脚架上的相机重复拍摄一个场景, 估计你的相机中的噪声量。(如果你有单反相机, 购买远程快门是一项不错的投资。) 或者, 取一个具有恒定颜色区域的场景 (如颜色检查表), 通过拟合每个颜色区域的平滑函数来估计方差, 然后从预测函数中取差。

- 1000 1. 为每个颜色通道分别绘制估计的方差作为水平的函数。
- 1001 2. 改变相机的 ISO 设置; 如果你不能做到这一点, 减少你的场景中的整体灯光 (关灯, 拉上窗
- 1002 帘, 等到黄昏)。噪声量是否随着 ISO/增益变化很大?
- 1003 3. 将你的相机与其他不同价位或生产年份的相机进行比较。是否有证据表明“一分钱一分货”?
- 1004 数码相机质量是否会随着时间的推移而改善?
- 1005 **例 2.7: 图像拼接中的伽马校正。**这是一个相对简单的谜题。假设您得到了两个图像, 它们是您想要
- 1006 缝合的全景图的一部分 (见章节 8.2)。这两张图片是用不同的曝光拍摄的, 所以你需要调整 RGB 值,
- 1007 以便它们在接缝线上匹配。是否有必要撤销颜色值中的伽玛值以实现这一点?
- 1008 **例 2.8: 肤色检测。**设计一个简单的肤色探测器 (Forsyth 和 Fleck 1999; Jones and Rehg 2001; Vezhnevets,
- 1009 Sazonov 和 Andreeva 2003; Kakumanu, Makrogiannis 和 Bourbakis 2007) 基于色度或其他颜色属。
- 1010 1. 取各种人的照片, 并计算每个像素的 xy 色度值。
- 1011 2. 裁剪照片或用绘画工具显示哪些像素可能是皮肤 (例如脸和手臂)。
- 1012 3. 计算这些像素的颜色 (色度) 分布。你可以使用像均值和协方差测量这样简单的方法, 也可以
- 1013 使用像均值漂移分割算法这样复杂的方法 (见第 7.5.2 节)。您可以选择使用非皮肤像素来建模背景
- 1014 分布。
- 1015 4. 使用计算出的分布来找到图像中的皮肤区域。一种简单的可视化方法是将所有非皮肤像素绘
- 1016 制成给定的颜色, 比如白色或黑色。
- 1017 5. 你的算法对色彩平衡 (场景照明) 有多敏感?
- 1018 6. 一个更简单的色度测量方法, 比如颜色比 (2.117), 是否也同样有效?
- 1019 **例 2.9: 白点平衡——棘手。**进行白点调整的一种常用 (相机内或后期处理) 技术是拍一张白纸, 然后
- 1020 调整图像的 RGB 值, 使其成为中性颜色。
- 1021 1. 描述一下你将如何调整一张图片的 RGB 值 ( $R_w, G_w, B_w$ ), 以使这个颜色为中性 (不需要改
- 1022 变太多曝光)。
- 1023 2. 你的转换是否包含一个简单的 RGB 值 (每通道) 缩放, 或者你是否需要一个完整的  $3 \times 3$  颜色
- 1024 扭曲矩阵 (或其他东西)?
- 1025 3. 将 RGB 值转换为 XYZ。适当的修正现在只依赖于 XY (或 XY) 值吗? 如果是这样, 当你转换
- 1026 回 RGB 空间时, 你是否需要一个完整的  $3 \times 3$  颜色扭曲矩阵来达到同样的效果?
- 1027 4. 如果你在直接 RGB 模式中使用纯对角线缩放, 但在 XYZ 空间中以扭曲告终, 你如何解释这
- 1028 种明显的二分法? 哪种方法是正确的? (或者有可能这两种方法实际上都不正确?)
- 1029 如果你想知道你的相机到底是做什么的, 继续下一个练习。
- 1030 **例 2.10: 相机内彩色处理——具有挑战性。**如果您的相机支持 RAW 像素模式, 取一对 RAW 和 JPEG
- 1031 图像, 看看您是否可以推断出相机在将 RAW 像素值转换为最终的颜色校正和伽玛压缩的 8 位 JPEG
- 1032 像素值时正在做什么。
- 1033 1. 从并列的 RAW 和颜色映射像素值之间的对应关系推导出你的颜色过滤器数组中的模式。在



这个阶段使用颜色检查表，如果它使你的生活更容易。您可能会发现将 RAW 图像分割成四个单独的图像 (对偶数和奇数列和行进行子采样)，并将每个新图像作为一个“虚拟”传感器来处理是很有帮助的。

2. 通过拍摄包含强烈颜色边缘的具有挑战性的场景 (如第 10.3.1 节所示) 的照片来评估拼接算法的质量。

3. 如果你可以在改变你的相机的颜色平衡值后拍摄相同的照片，比较这些设置如何影响这个处理。

4. 将你的结果与在 (Chakrabarti, Scharstein, 和 Zickler 2009; Kim, Lin 等人, 2012; Hasinoff, Sharlet 等人 2016; Karaimer and Brown 2016; Brooks, Mildenhall 等人 2019 年) 或使用他们的彩色图像数据库中可用的数据。