第6章　机器人视觉

6.1　引言

迄今为止，所描述的S-R和状态机使用十分有限的传感器输入，它们只能提供网格世界中与其毗邻的单元的信息。实际上还存在许多其他表达 agent所处世界的重要信息的传感器输入——音响、温度和压力，等等。传感器可用于各种机器以使其能对所处环境作出响应。

动物用眼睛一瞥，其视觉感官就能提供大量有关其所处世界的信息。赋予机器人“看”的功能正是“机器人视觉 (*computer vision*)”这个学科所研究的问题之一。这一领域十分广阔，不仅包括通用技术，而且也包括为数众多的专用技术——如字符识别、相片解释、脸谱识别、指纹识别和机器人控制等等。

尽管对我们人类来说，“看”轻而易举，但这对机器来说却是一件非常困难的事。这其中的困难主要来源于难以控制的照明、影象和复杂而难以描述的物体，如那些室外场景中的物体、非刚性物体或啮合其他物体的物体。其中有些困难在人造环境中，如建筑物的室内景观，可得以减轻，而且在这种环境中研究计算机视觉往往更成功。这里限于篇幅，仅介绍机器人视觉的一些主要概念。

计算机视觉首先是在一组感光性原件上，如电视摄像机的光电管，生成一个场景的图象（对立体视觉需生成两个或两个以上的图象。本章的后面部分会介绍立体视觉）。这个图象是摄像机通过镜头对在视野中的场景进行一个透视投影，然后光电元件将其转换成一个二维的、随时间变化的亮度矩阵图象 $I(x,y,t)$，其中x和y为光电元件在数组中的位置，t为时间（对有色视觉，需形成三个这样的矩阵来分别代表三原色。但我们在这里只考虑单色的情况，同时排除了可变时间——即假设一个静态场景）。一个由视觉引导的响应 agent必须通过处理这个矩阵来产生这个场景的图标模型或者一组特征，从而使它能直接计算一个动作。

如图6-1所示，透视投影是多对一的变换。多个不同的场景可能生成相同的图象。更麻烦的是，图象易受到周围光线不足或其他因素的干扰，这样，我们就不能直接转换图象来重建场景。因此，agent通过运用可能处于有关场景中的物体的特定知识、有关场景中的各种表面的特性以及由这些表面反射回摄像机的周围照明度等一般知识来从图象中获取有用的信息。

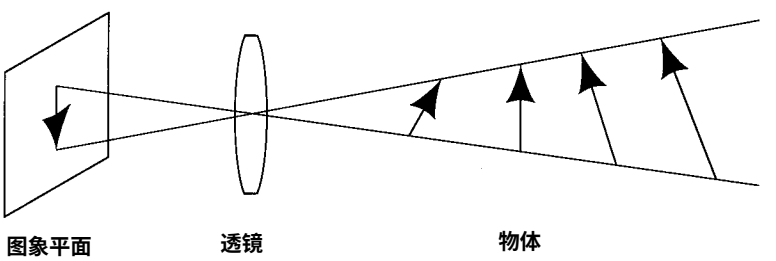


图6-1　图象生成过程的多对一特性

希望获取信息的种类取决于 agent的目的和任务。若要让一个 agent平安地通过一个混乱的

环境，这个agent必须了解其中物体的位置、边界、通路以及它所经路径表面的特性。 agent若想要操纵物体，就必须知道这些物体的位置、大小、形状、成分和构造等。对其他目的而言，agent也许应了解颜色并能识别它们的类别。 agent也许还应具备根据每隔一段时间所有以上信息的变化来预测将来可能的变化。从一个或多个图象中获取此类信息将极其困难，所以，如前所述，我只能给出这类技术的一个概况。

6.2 操纵一辆汽车

在S-R agent 的一些应用中，神经网络可用来把图象的亮度矩阵直接转换成动作。其中的一个突出的例子就是用来驾驶一辆汽车的ALVINN系统[⊖][Pomerleau 1991, Pomerleau 1993]。在介绍机器人视觉的一般过程之前，让我们先来看看这个系统。此系统的输入来自一个低解析度（30×32）的电视图象。一个电视摄像机被架在汽车上对准前面的道路，电视图象被采样并为神经网络产生一系列960维的输入向量。此网络如图6-2所示。

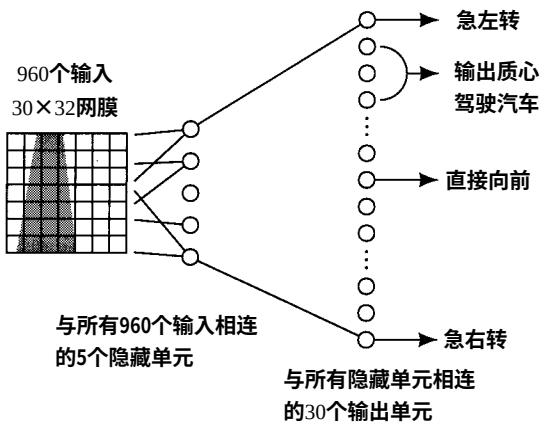


图6-2 ALVINN网络

网络的第一层有5个隐藏单元，第二层有30个输出单元，所有以上单元均为sigmoid单元。输出单元通过线性排列来控制汽车的角度。若此输出单元队列的顶端附近的一个输出单元的输出比其他大多数输出单元高，则车往左行驶；若在此队列的底端附近的一个单元的输出较高，则车往右行驶。计算出所有这些输出单元的响应的“质心”，并且把此车的驾驶角度设置为完全向左和完全向右之间相应的一个值。

此系统由一个改进过的“在空中（*on-the-fly*）”训练方式来训练：让一个真人驾驶员开车，实际的驾驶角度被作为相应输入的正确标志。网络以反向传播的方式递增地训练，从而使它能用驾驶员所指定的驾驶角度来响应实际驾驶车辆时出现的每一个视觉模式。整个训练大概要用五分钟的驾驶时间。

这个简单的训练过程经过改进后避免了两个潜在的问题。首先，因为驾驶员通常能很好地驾驶车辆，所以网络决不会经历任何偏离中心的车辆位置和不正确的车辆方位。其次，对长而笔直的路面，网络会在长时间的训练后，只知道产生笔直向前的驾驶角度；这种训练会使以前有关沿弯曲路面的训练无效。我们并不希望通过让驾驶员偶尔不稳定地驾驶来避免这些问题，因为系统将学会模拟这种不稳定的行为。

⊖ 基于神经网络的自治的地面车辆。

所以，我们在软件中平移和旋转每个初始图象，从而产生 14个补充图象，在这些图象中，车辆相对于路面的位置不同。再用一个模型来告知系统哪个图象应用哪个驾驶角度，并给出驾驶员为初始图象指定的驾驶角度，这样，系统将产生另外 14个有标志的训练向量，并把它们加到一般训练过程中所遇到的那些训练向量上去。

经过训练，ALVINN可驾驶各种“试验”汽车在没有标线的铺筑过的路面、吉普车道、有标线的城市街道和州际高速公路上行驶。 ALVINN曾在高速公路上以高于 100公里／小时的速度连续行驶了120公里。

6.3 机器人视觉的两个阶段

尽管ALVINN的表演给人的印象很深，但许多机器人的任务需要对更高分辨率的图象进行更为复杂的处理。因为许多任务要求机器人了解场景中的物体，下面将集中讨论有关找出物体的技术。首先，什么是一个物体？在人造环境中，如建筑物的室内景观中，门口、家具、其他agent、人、墙及地板等等都是物体。在外部自然环境中，动物、植物、人造结构、汽车及道路等等都是物体。人造环境对机器人视觉来说通常比较容易，因为其中的物体有比较规则的边缘和表面。

有两种计算机视觉技术对勾勒出与场景中的物体相关的各部分图象的轮廓十分有用。一种技术是在图象中寻找“边缘”。一个图象边缘是图象的一部分，图象亮度或其他图象的特性在此处陡然变化。另一种技术试图把图象分为几个区域，一个区域也是图象的一部分，图象亮度或其他图象的特性在此处缓慢变化。图象中的边缘和区域之间的边界，经常但不总是与场景中产生图象的那些重要的、与物体相关的不连续点相对应。图 6-3是一些不连续点的例子[⊖]。根据照明强度、表面特性和摄像机的角度，这些不连续点可由图象边缘和图象区域边界来表示。这样，获取这些图象特征成为机器人视觉要完成的一项重要的任务。

视觉处理过程可分成两个主要阶段，如图 6-4所示。图象处理阶段主要把原始图象转换成更适合于景物分段的图象。图象处理包括降低噪声、增强边缘和寻找图象区域等不同的滤波操作。景物分析主要试图从已处理的图象中产生一个对原始场景的图标描述或基于特征的描述，并提供agent所处场景中与特定任务有关的信息。把机器人视觉分为两个阶段只是对这个过程的简化。实际的机器人视觉涉及更多的阶段，而且这些阶段一般都相互影响。

以后会详细讨论这两个阶段。但现在，为了了解一下概况，先来讨论一下图 6-5所描绘的处

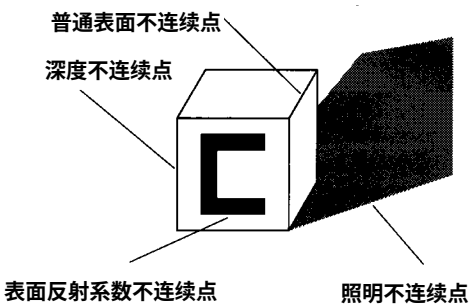


图6-3 场景的不连续点

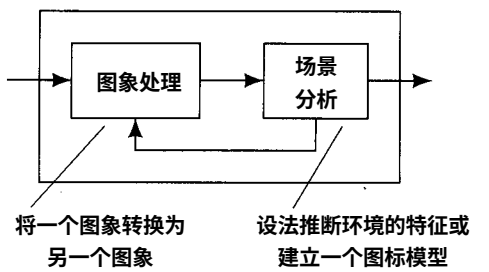


图6-4 机器人视觉的两个阶段

[⊖] 节选自[Nalwa 1993, p.77]

于网格空间的机器人。机器人的视野中有三个标有 A、B和C的玩具积木、一个门口和一个房间的一角。首先，图象处理排除伪造的噪声并增强物体的边缘以及其他不连续点。接着，已知世界中的物体的形状均由直线边界构成，景物分析会产生一个对此世界的图标表示——与用于计算机图形学中的模型相似。通常，这个图标模型用来更新存储在内存中的更全面的环境模型，然后计算出适合于这个假设环境状态的动作。

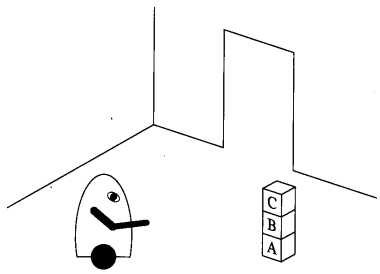


图6-5 机器人和元具积木都在一个房间里

根据其任务，图标模型并不一定要象计算机图形学模型那样描绘所有细节。若当前的任务仅与玩具积木有关，那么房间角落和门的位置就无关紧要了。我们不妨再假设只有积木的布局比较重要，那么，图标模型应为一个表结构((*C B A FLOOR*))，它表示C在B上，B在A上,而A在地板上。若C被移到地板上，那么图标模型应为((*C FLOOR*)(*B A FLOOR*))（也可以是((*B A FLOOR*)(*C FLOOR*)），但这里我们假设积木的相对水平位置无关紧要，这样，表结构的第一级元素的顺序就无表达意义)。因为每一个元件的最后一个元素均为*FLOOR*，所以我们可以去掉这一项来缩短表结构。

对于根本不用图标模型的机器人来说，景物分析会用另一种方法把处理过的图象直接转换成适合于机器人任务的特征。如，若机器人必须判定积木 C上是否有其他积木，那么，一个对环境的描述应包括一个特征值，如*CLEAR_C*，积木C上无其他物体时这个特征值为1，否则为0(这里，为了使这些特征便于记忆，没有使用通常所用的x。必须记住，这些名字仅仅能帮助我们记忆，却无法帮助机器人记忆)。本例中，景物分析仅从已处理过的图象中计算特征的值。从以上例子中我们可以看出，景物分析完全根据所设计的机器人和它要完成的任务而定。

6.4 图象处理

6.4.1 平均法

假设初始图象可表达为一个 $m\times n$数组$I(x,y)$，我们称之为“图象亮度数组(*image intensity array*)”。它把图象平面分成许多被称为“象素(pixel)”的单元。这些数字表示这幅图象中某点的光亮度。图象中一些不规则之处可通过求平均数的方法得以平滑。这个平滑操作就是把一个求平均数的窗口在整个数组中滑动。这一求平均数的窗口对准每个象素的中心，并计算出在求平均数窗口内的数字的加权总和，然后把此象素的初始值替换为这个加权总和。这种滑动并求和的操作称为“卷积(*convolution*)”。若我们希望所得的数组是二进制数字（1或0)，那么就必须把这些加权总和与一个阈值比较。平均法不仅将压缩孤立的噪音点，而且将减小图象的卷曲度（*crispness*)，并放弃那些微不足道的图象元素。

卷积是从信号处理中得来的操作。它通常被解释成对波形（沿时间轴滑动）的一维的操作。若我们沿一个信号$s(t)$滑动或卷积一个函数$w(t)$后，将得到平均信号$s^{\star}(t)$：

$$s^{\star}(t)=\int s(u)w(u-t)du=s(t)\star w(t)$$

用★来表示卷积。

图象处理中的二维离散式卷积如下：

$$I^{\star}(x,y)=I(x,y)\star W(x,y)=\sum_{u=-\infty}^{\infty}\sum_{v=-\infty}^{\infty}I(u,v)W(u-x,v-y)$$

这里，$I(x,y)$是初始图象的数组，$W(u,v)$是卷积加权函数。假设$I(x,y)=0$当且仅当$x<0$或$x\geq n$，且$y<0$或$y\geq m$（这样，这个卷积操作会在图象的边界附近产生一些“边缘效应”）。

有时，我们把加权函数$W(x,y)$的值在x和y构成的长方形内看作1，长方形之外看作0。长方形的大小决定平滑度，长方形越大平滑度越高。图6-6展示了一个求平均数操作是如何对一个二进制图象先用一个长方形平滑函数平滑，然后将其与阈值比较来进行操作的（设图中的黑色象素的亮度值高而白色象素的亮度值低或为0。这个假设看起来好象是反的，但这样可以使图解简单些）。我们发现，这个平滑操作加粗了宽线，去除了窄线和微小细节。

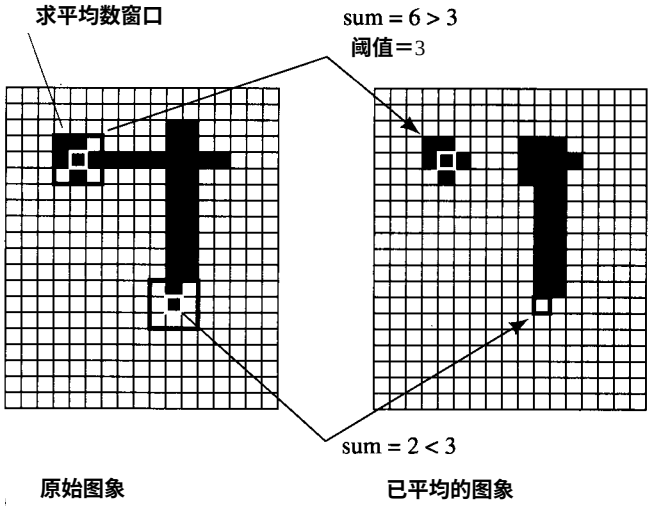


图6-6 求平均数操作的元素

用于平滑的常用函数是一个二维高斯函数（Gaussian）

$$W(x,y)=G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

此函数描述的表面是铃铛形的，如图6-7（图中，移动了x和y轴的位置以便清楚地展示高斯表面）。高斯函数的标准差σ决定了表面的“宽度”，从而也就决定了平滑度。$G(x,y)$有一项是求x和y的积分。图6-8展示了采用不同的平滑因素对由积木和机器人组成的网格空间场景进行高斯平滑后的三个图象以及它们的初始图象[⊖]（通常，图象平滑和过滤的离散操作对离散值进行插值以增强表现效果）。

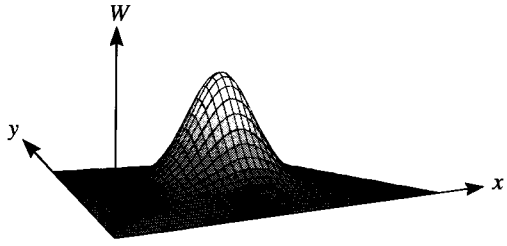
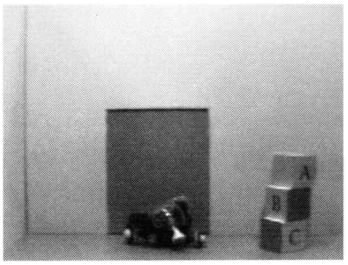


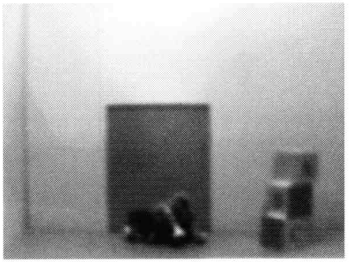
图6-7 高斯平滑函数

可以发现图6-8中的图象一个比一个模糊。我们可以这样来解释这种现象：假想图象亮度函数$I(x,y)$表示一个长方形导热板的初始温度场。随着时间的流逝，导热板各向同性地散热导致高温与低温混在一起。依照这种观点，图6-8中排列的图象按时间先后依次表现了当时的温度场。Koenderink[Koenderink 1984]指出，用标准差σ的高斯函数来卷积一个图象等同于在已知图象亮度域情况下求一个时间与σ成比例的扩散方程的解。

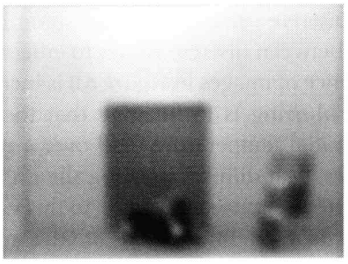
⊖ 在此，我要感谢Charles Richad，他创造了本章所用的图象并为这些图象的处理操作进行了编程。



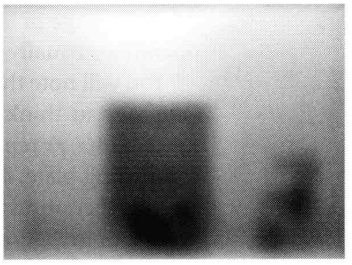
原始图象



高斯宽度为两个象素



高斯宽度为4 个像素



高斯宽度为8 个象素

图6-8 用高斯函数过滤的图象平滑过程

6.4.2 边缘增强

如前所述，计算机视觉常常涉及图象边缘的提取，然后用这些边缘来把图象转换成某种线条图形。转换后的图象中的轮廓可与场景所包含的各类物体的原型（即模型）的轮廓特征相比较。获取轮廓的方法之一是先增强图象中的边界和边缘，边缘可以是图象各部分之间的任意边界，这些边缘的特性，如亮度，彼此之间明显不同。这样的边缘常常与图 6-3所示的重要物体特性有关。

首先讨论一维图象，这意味着 $I(x,y)$只随x的变化而变化，并不随 y变化。然后介绍一些二维图象。我们可以通过在一维图象上卷积一个位于垂直线上的、一半为负一半为正的窗口（如图 6-9）来增强这些图象的边缘强度。在图象的平均部分此窗口的总和为 0。

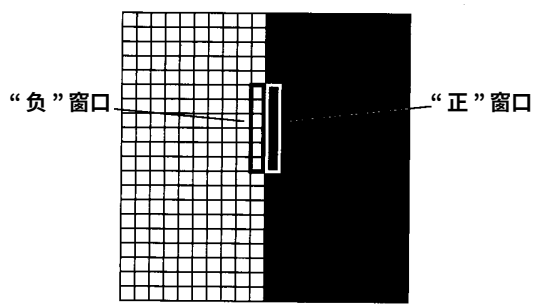


图6-9 边缘增强

若在图象上沿x方向卷积图 6-9所示的窗口，那么，在与y方向连成一条线的边缘处会形成一个尖峰。这个操作与对图象亮度函数相对于 x第一次求导（即dI/dx）极其相似。若我们对它们进行第二次求导，效果会更明显。由此，我们将得到一个经过 0点的波形，边缘的一边为正波形而另一边为负波形。以上操作在一维空间的效果图如图 6-10所示。这样，亮度的变化比较平滑，不象图 6-9中的变化那么剧烈。当然，图象亮度的变化越陡， dI/dx的顶峰就会越窄。图象边缘出现在$d^2I/dx^2 = 0$处，即图象的二次导数的 0点。

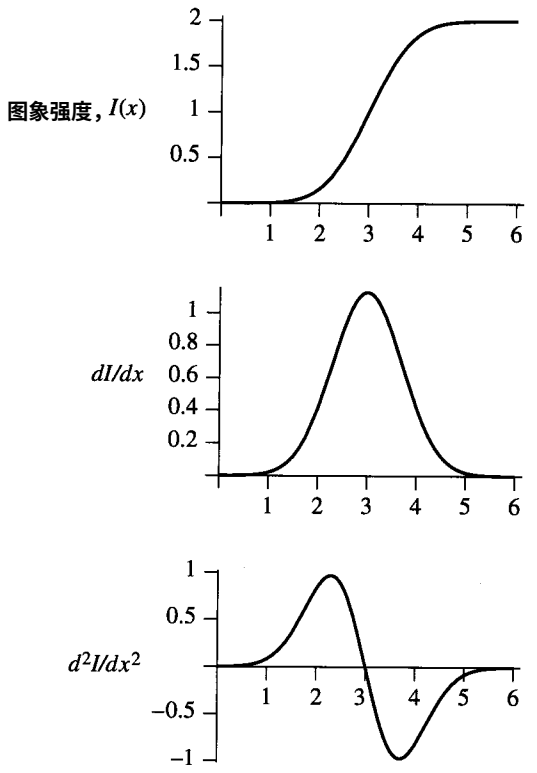


图6-10 对图象强度求导

6.4.3 边缘增强与平均法的结合

边缘增强本身将在增强边缘的同时突出图象中的假噪声元素。为了减小对噪声的敏感度，可以先用平均法再用边缘增强来把两种操作结合起来。于是，我们首先用一维高斯函数对连续的一维图象进行平滑处理。这一高斯函数为：

$$G(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}$$

其中，σ为标准差，它是此平滑函数的宽度的一个量度。用高斯函数平滑后得到过滤后的图象：

$$I^*(x)=I(x)\star G(x)=\int I(u)G(u-x)\,du$$

随后，通过边缘增强得出：

$$d^2[I^*(x)]/dx^2=d^2[I(x)\star G(x)]/dx^2=d^2\left[\int I(u)G(u-x)\,du\right]/dx^2$$

因为求导和求积分的顺序可互换，上式等同于 $I(x)\bigstar d^2G(x)/dx^2$。这样把平滑过程与边缘增强结合后，我们可以用一个高斯曲线的二次导数来卷积一维图象，而不必对图象二次求导。

现在，来看看二维图象。我们需要一个二次求导型操作来增强任一方位的边缘强度。拉普拉斯变换就是这样的操作。$I(x,y)$的拉普拉斯变换的定义如下：

$$\nabla^2 I(x,y) = \partial^2 I(x,y)/\partial x^2 + \partial^2 I(x,y)/\partial y^2$$

如果要在二维空间中把边缘增强和高斯平滑结合起来，我们必须改变求导和卷积的顺序（与在一维空间所做的一样），因此得到

$$I(x,y) \star [\partial^2 G(x,y)/\partial x^2 + \partial^2 G(x,y)/\partial y^2]$$

二维高斯函数的拉普拉斯变换有点像一顶倒置的帽子，如图6-11所示（这里，再次移动了坐标空间）。它又被称为“*sombrero*（宽边帽）函数”，帽宽决定了平滑度。

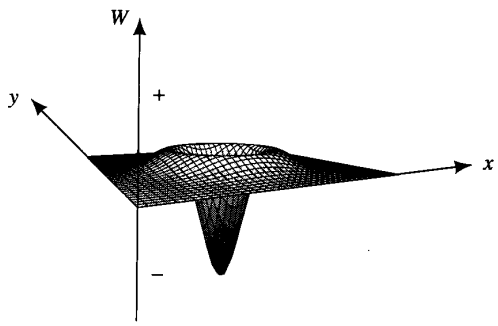


图6-11 用于拉普拉斯过滤的Sombrero函数

然后，通过用这个帽函数来卷积图象，就可以完成整个求平均和边缘寻找的操作。这个操作又被称为“拉普拉斯过滤（*laplacian filtering*）”，它产生的图象叫做“拉普拉斯过滤图象”。根据脊椎动物视网膜上的视觉处理已被发现与拉普拉斯过滤有些相似。而且，拉普拉斯过滤图象的过0点可用来产生大体轮廓图。拉普拉斯过滤和标出过0点这两个过程构成了所谓的“Marr-Hildreth”算子[Marr & Hildreth 1980]（Marr-Hidreth算子的输出就是Marr所称的“原始轮廓”的一部分）。图6-12是以上操作对网格空间场景的图象所产生的效果⊖。我们注意到，Marr-Hilldreth算子为画出由简单边界组成的场景的各部分的轮廓素描提供了合理的基础，然而它却未能很好地画出图中位于门口的那个稍微复杂的机器人的轮廓。

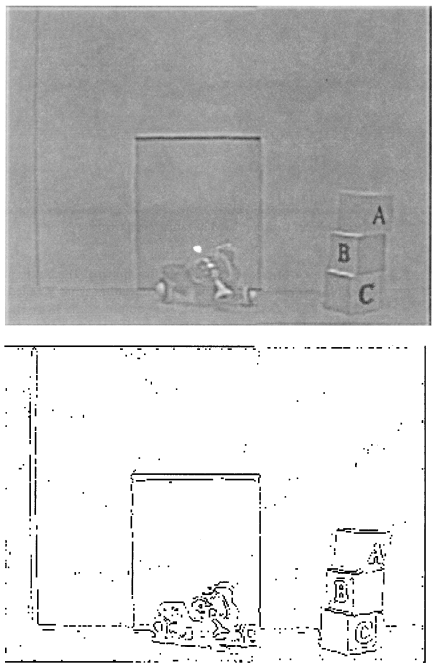


图6-12 拉普拉斯过滤和Marr-Hildreth算子（拉普拉斯变换的宽为1象素）

实际还存在许多其他边缘增强和线条寻找的操作，有些产生的效果比这种易于解释且十分

⊖ 在计算图6-12的图象时，用段交叉点来代替过0点。图象亮度必须与0周围的波段相交才能被显现出来。

流行的拉普拉斯变换更好。其中突出的有：Canny变换（[Canny 1986]）、Sobel变换（归功于Irwin Sobel的[Pingle 1969]）、Hueckel变换（[Hueckel 1973]）和Nalwa-Binford变换（[Nalwa & Binford 1986]）。值得注意的是，Marr-Hildreth和其他边缘增强变换开始均把象素标号为可能处于图象边缘或线条的候选者，然后再把这些候选者连接起来形成轮廓或者其他简单的曲线。

6.4.4 区域查找

另一种处理图象的方法试图在图象中查找亮度或其他特性，如纹理等变化不突然的“区域”。从某种意义上来讲，查找区域是查找轮廓的对等物（*dual*）；这两种技术均把图象分割成我们所希望的与场景相关的若干部分，但由于二者均对噪声比较敏感，因此这两种技术通常用来互补。

首先，我们必须定义什么是图象的一个区域。一个区域就是一组满足以下特性的相互连接的象素：

1) 一个区域由类似的成分组成。常用的同质特性（*homogeneity property*）如下：

(a) 在这个区域中，象素的亮度值之间的差别不超过某个 ε。

(b) k次多项式（k的值比较低且事先指定）的表面可与此区域内象素的亮度值以小于 ε的最大误差（即表面与区域亮度值之间的误差）拟合。

2) 任意两个毗邻的区域内的所有象素的组合不满足同质特性。

通常，把一个图象分割成区域的方式不止一种，但每个区域总是与世界中的一个物体或其有意义的一部分相对应。

与边缘增强和线条查找一样，用来把图象分割成区域的技术有很多，下面介绍其中一种叫做“分割—合并（*split-and-merge*）”的方法[Horowitz & Pavlidis 1976]。若用一种简单的方法来描述，可首先把整个图象作为一个候选区域。为了便于图解，我们设此图象为一个由$2^l \times 2^l$数组组成的正方形。显然，这个候选区域并不满足“区域”这一定义，因为图象中所有的象素集不满足同质特性（具有相同亮度的图象除外）。于是我们把每个不满足同质特性的候选区域分割成另外四个等大的候选区域，并一直这样分割下去直至无需再进行分割。图6-13是一个人工的8×8图象的分割过程，其中运用了亮度的差别不超过1个单元这个同质特性。当无须再进行分割时，可以合并那些满足此同质特性的彼此毗邻的候选区域。可用不同的顺序进行兼并——所得的最终区域也会不同。实际上，在分割完成之前也可以进行合并。为了简化图解，图6-13中的所有合并在最后一步进行。

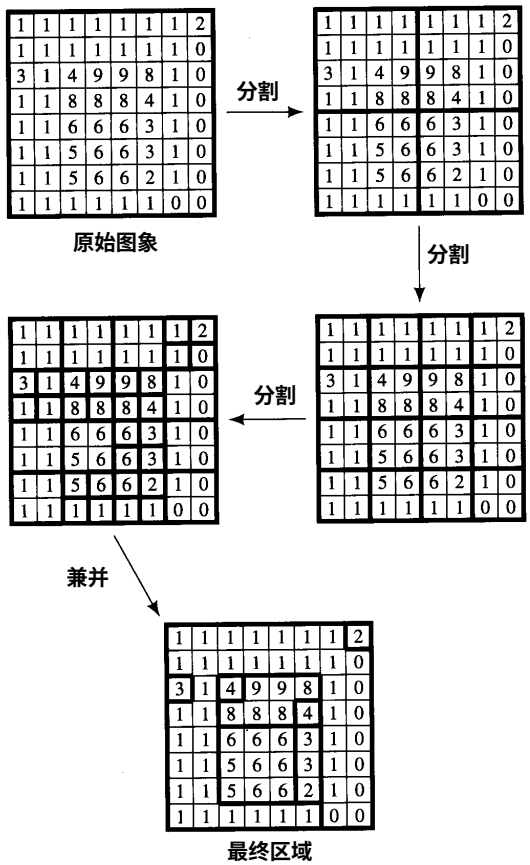


图6-13 分割和兼并候选区域

图6-13中，用的是低分辨率的图象来说

明区域查找过程。图6-14展示了一个高分辨率的图象经过此过程后的结果。与图6-13一样，我们发现了一些微小区域和许多不规则的区域边界。通常，可以通过去掉微小区域（其中有些是较大区域的过渡地带）、规整边界轮廓以及考虑可能处于场景中的物体的已知形状等等来“净化”用分割—合并算法找到的区域。图6-14中，网格世界里沿墙的亮度梯度产生了许多区域。

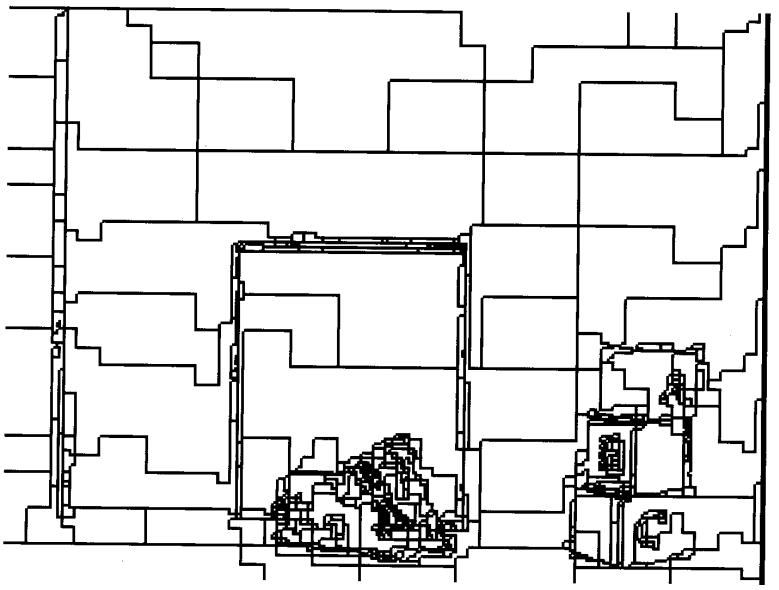


图6-14 在一个网格场景中用分割—合并算法查找而来的区域

回想一下，在讨论用高斯函数平滑图象时，提到了各向同性热扩散过程。Perona和Malik[Perona & Malik 1990]曾提出一个可用于生成区域的各向异性扩散过程。这个过程鼓励亮度变化小的平滑而拒绝亮度变化大的平滑。如[Nalwa 1993, p. 96]所述，这一过程结果将形成亮度相同的不同区域，而区域之间的边界的亮度梯度很高。

6.4.5 运用亮度以外的其他图象的属性

边缘增强和区域查找还可以基于除图象亮度的同质特性以外的其他图象属性。世界上众多物体的表面反光度有细微的差别，我们称之为视觉纹理。如一片草地、一块地毯、一簇树叶、动物的皮毛等等，它们的表面反光度均彼此不同。而这些物体反光度的强异会在图象高度上产生类似细微差别。

计算机视觉研究者已经能识别多种视觉纹理，并且开发了各种纹理分析工具，其中有结构化方法和统计方法。这些方法或者用来把某种具体纹理的图象的各个部分进行归类，或者用来把图象分割成各自拥有特殊纹理的不同区域。结构化方法力图用由原始“texels”（即是由黑白部分组成的微小形状）构成的棋盘形布置来表示图象区域（[Ballard & Brown 1982, 第6章]有完整的讨论）。

统计方法基于以下观点：图象区域的亮度值的概率分布能很好地描述图象的纹理。举一个简略的例子：有这样一张草地的图象，图中的草叶垂直生长。这个图象将有这样一个概率分布：被低亮度区域分割开的、窄且垂直分布的高亮度区域有峰值。最近，由Zhu及其同事所著的[Zhu, Wu, & Mumford 1998]介绍了估算各种视觉纹理的概率分布的方法。一旦我们知道这些分

布，就可以用它们来对纹理归类，并在此基础上分割图象。

除了纹理，我们可能还会用到其他图象属性。若用一个直接的方法来测量场景中的物体到摄像机的距离（如用一个激光距离探测器），则可以产生一个“距离图象”（图象中每个象素值表达场景中某点到摄像机的距离），并找出陡然的距离变化。至于运动和颜色等可感知或计算的其他特性，可以通过图象处理操作而得到。

6.5 场景分析

在用以上所讨论的技术对图象进行处理后，我们力图从中获取所需的有关场景的信息。计算机视觉的这个阶段被称为“场景分析（*scene analysis*）”。由于场景—图象的转换是多对一的，场景分析需要其他补充图象或有关将遇到的场景种类的大体信息。我将在后面描述立体视觉时讨论补充图象的运用；这里，将介绍用场景知识获取场景信息的不同方法。

这些所需的场景知识可能十分概括（如物体的反光度），也可能十分具体（如场景可能包括门旁的一些叠放在一起的箱子）。它可能清楚，也可能模糊。如，在线条查找算法的操作中就可能建有“什么构成了一条线”这个模糊知识。处于这两种极端之间的其他场景信息包括摄像机的位置、照明光源的位置和场景位于办公楼内还是室外等等。下面所讨论的场景知识均属于以上范围。有关这方面更详细的论述，请参阅计算机视觉的教材。

表面反光度特性和图象亮度的明暗常用来给出场景中光滑物体形状的信息。而图象明暗尤其能帮助我们计算物体的表面法线。Horn及其同事已经发明了从图象明暗中推出形状的方法（请参阅[Horn 1986]中的描述）。图象中所表达的纹理元素是场景中的元素的透视投影，这使从纹理中获取形状和质深（*qualitative-depth*）更加容易。

如前所述，我们有时需要一个场景的图标模型，但有时仅需场景的某些特征。图标景物分析通常力图建立一个场景或部分场景的模型。基于特征的景物分析仅获取当前任务所需的场景的特征。一种有代表性的基于特征的景物分析被称作“面向任务的（*task-oriented*）”或“意图（*purposive*）”视觉（可参见[Ballard 1991,Aloimonos 1993]）。

6.5.1 解释图象中的线条和曲线

对已知包含直线物体的场景（如建筑物内部场景和网格世界的场景）进行分析时，其中关键的一步就是图象中线条的假定（这些线条后来将与场景的关键元素相关）。可以通过采用把直线段与边缘或区域的边界拟合的技术来生成直线。对于包含曲线物体的场景，我们可以把圆锥截面（如椭圆、抛物线和双曲线）与原始轮廓或区域的边界拟合（请参阅[Nalwa & Pauchon 1987]）来生成曲线。在经过去除短线、在端点处连接直线和曲线这些技术操作后，把图象转化成一个线条画（*line drawing*），这幅线条画可用于进一步解释。

有很多把场景特性与线条画的元素相结合的策略。这样的结合称为“解释（*interpreting*）”线条画。这里，将介绍一种解释线条画的策略。在这种策略中，已知场景仅包含平面，从而使相交于一点的平面不超过三个（这种平面组合体称为“三面体顶点多面体（*trihedral vertex polyhedra*）”）。图6-15是此种场景的一个典型例子，它是一个由边界墙、地板、天花板和一地板上的正方体组成的室内场景。在这样的场景中，由两个相交平面组成的场景的边缘只有三种。一种边缘的两个相交平面的其中一个遮住了另一个（即在场景中只能看见其中的一个平面），这种边缘称为“*occlude*”。图6-15中用箭头（→）做标记的就是occlude，箭头沿边缘的指向使

得遮住另一个平面的平面位于箭头的右边。另两种边缘的两个相交平面在场景中均可见。其中形成的凸边称为“刀刃（*blade*）”，图中的标记为加号（+）；形成的凹边称为“折痕（*fold*）”，图中的标记为减号（−）。

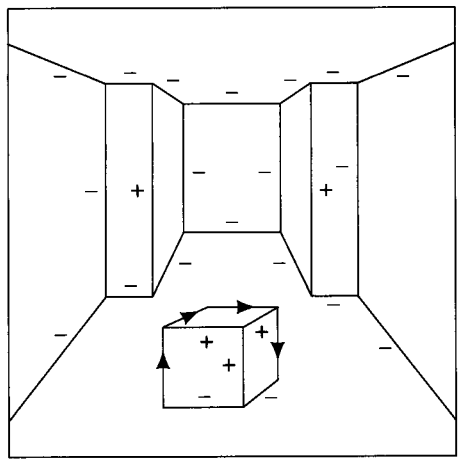


图6-15 一个房间的场景

假设能从这样的场景中得到一个可转化成线条画的图象。如图6-15所示。我们能通过这种给场景中的线条作标记的方法来精确地描述场景中的边缘的种类吗？在一定条件下，我们能。首先，从一个“一般的视点”来得到一个场景的图象——即这种视点不会使场景中的任两条边缘在图象中连成一条线。然后再用基于以下事实的方法来给图象中的线条作标记——（在我们假设的三面体顶点多面体中）只能存在有限种给图象中线条的连接点作标记的方法。图6-16便是这些标记方法。尽管还存在其他许多种给图象中线条的连接点作标记的方法，但在多面体的场景中，图6-16所示的就是所有可能存在的标记种类。

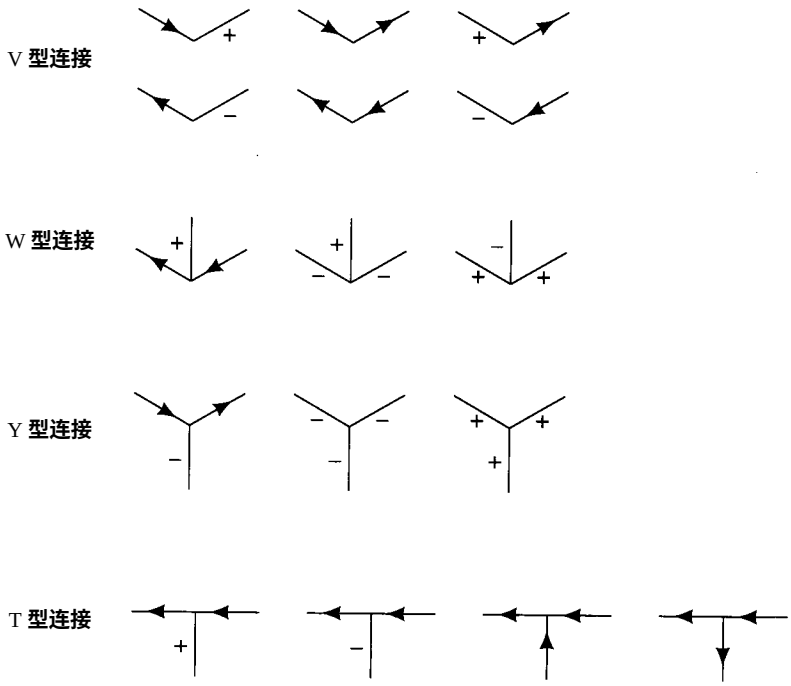


图6-16 线条的连接点的标记（根据[Huffman 1971]）

标记线条的景物分析过程如下：首先，根据线条连接的形状，给图象中所有的连接点分别标上V、W、Y或T。在图6-17的房间场景的图象中，已经按以上方法给连接点作好了标记。然后再给图象中的线条分别标上+、−或→，但必须遵循图6-16中的规则。而且，连接两个连接点的图象线条的标记必须前后一致。这些约束条件通常（但不是总是）导致只能有一种标记方法。若这些标记前后不一致，那么，在把图象转化成线条画时就会出错，或者这时所用的场景

不是三面体多面体。在给图象线条作标记时，由这些约束条件产生的问题在人工智能中称为“约束满足问题”。在后面会讨论解决这类普遍问题的方法，但你现在不妨试着给图 6-17作出一种前后一致的标记（当然，如果图象的标记与图6-15中场景的标记一一相应，那么这就是一种前后一致的标记方法。然而，这个场景是被规定有那些标记的。你能想出一种自动地为图象线条作前后一致标记的方法吗？）。

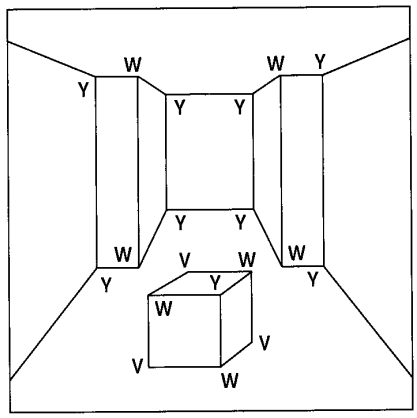


图6-17 按类型给图像连接点作标记

[Guzman 1986, Huffman 1971, Clowes 1977]首先开始研究给三面体顶点多面体图象中的线条作标记的景物分析技术，[Waltz 1975]以及其他书籍对此作了延伸。对包含非平面表面的场景的类似分析也获得了一定的成功。有关线条画解释的更详尽的解说（附引用），请参阅[Nalwa 1993, 第4章]。

大量有关场景的有用信息能从对线条画中线条和曲线的解释中得到。如机器人可预测到，向一个垂直的fold(场景中的一个凹形边缘)移动最终会使其到达一个角落。绕过多面体障碍可以通过绕过垂直的blade（凸形边缘）来实现。只要有足够的有关场景的一般知识，那么所需的特征或图标模型就可以直接从已解释过的线条画中得到。

6.5.2 基于模型的视觉

计算机视觉技术运用越来越多的场景知识，现在来看看运用可能出现在场景中的物体模型的技术。例如，若已知场景由机器人组装所用的工业部件和零件所组成，那么，这些部件和零件的形状模型可用来帮助解释图象。下面概括性地介绍一些用于基于模型的视觉方法。若想作进一步的了解，请参阅[Binford 1982, Grimson 1990, Shirai 1987]。

正如线条和曲线可与图象中的区域的边界部分拟合，模型的透视投影也可与模型的各部分进行拟合。例如，如果已知一个场景包括一个平行六面体（如图6-15中的场景），我们可以把此平行六面体的一个投影与此场景图象的各个元素相拟合。此平行六面体有其给定大小、位置和方位的参数。调节这些参数，找到一组参数能使此平行六面体与图象中的一组线条很好地拟合。

研究者们还运用通用圆筒（*generalized cylinder*）[Binford 1987]作为建构模型的积木。此通用圆筒如图6-18所示。每个圆筒有九个参数。图 6-19是用此种通用圆筒对一个人体的粗略的场景重建。这种方法也适用于分层表示，因为每个圆筒可与一组更小的、能更精确地表示图形的圆筒相连接。正如Nalwa所述，用分层的通用圆筒来表达场景中的景物“说起来容易，做起来难”[Nalwa 1993, p. 293]，但这种方法已经运用于一些物体识别的应用中[Brooks 1981]。有关三维结构模型表示的运用，请参阅[Ballard & Brown 1982, 第9章]。

我们可用不同的模型元素和模型拟合来生成一个整个场景的图标模型，或得到足够的有关场景的信息来获取当前任务所需的特征。通过把实际图象与用场景分析得来的图标模型构建的模拟图象进行比较，基于模型的方法能测试这些模拟图象的准确度。这些模拟图象必须由运用参数的模型来绘制，而这些参数与图象处理过程所用的参数（如摄像机角度等）相似。这样，就需要照明、表面反光特征以及计算机图形学的绘图过程的其他各方面的所有合适的模型。

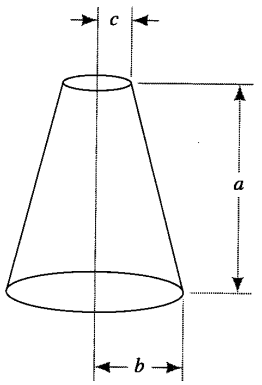


图6-18 一个通用圆筒

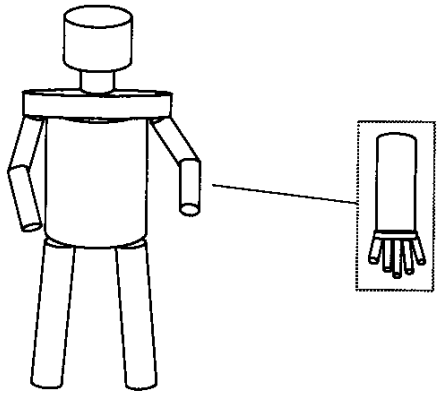


图6-19 运用通用圆筒的一个场景模型

6.6 立体视觉和深度信息

透视投影会使一个大而远的物体与一个与其相似的小而近的物体所产生的图象相同。这样，从单个图象估量物体的距离就十分困难了。但我们可运用立体视觉 (*stereo vision*)来得到深度信息，这种方法是基于两个（或两个以上）图象的三角测量计算的运用。

然而，在讨论立体视觉之前，需要指出，在某些情况下如具备适当预备知识时，我们可以从单一图象中获取深度信息。例如，对图象的纹理分析（考虑场景纹理的透视变形这一因素）表明场景中有些元素比另一些更近。在一定的情况下，我们还能得到更精确的深度信息。例如，在一个办公室场景中，若已知所观察的物体在地板上和摄像机离此地板的高度，那么，运用从摄像机镜头的中心到图象上某一恰当的点的角度，可算出与物体的距离。图 6-20图解了这种计算（角度α可用摄像机的焦距和图象的维数来计算）。可用相似的方法来计算与门口的距离以及物体的大小等等。

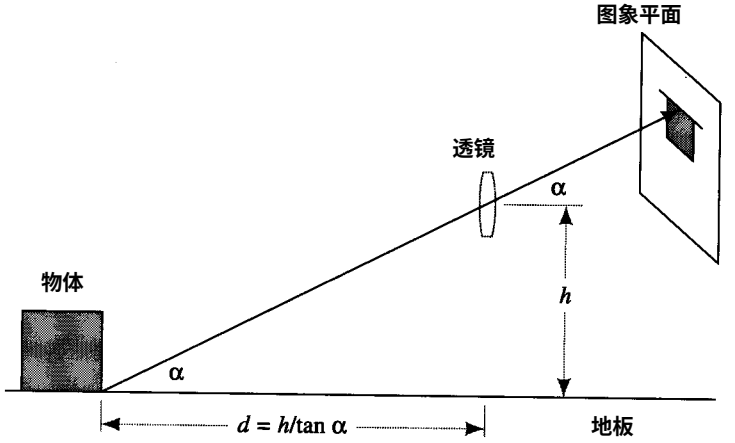


图6-20 从单个图象中计算深度

立体视觉也运用三角测量，其基本观点十分简单。我们来讨论一下图 6-21所示的二维设置。这里有两个透镜，它们的中心被基线 (*baseline*)*b*分开。与透镜距离为*d*的一个场景点通过这两个

透镜所产生的图象点如图所示。透镜到这些图象点的角度可用来计算 d。若测量角度和基线的精确度不变，那么基线越长，物距越短，可得到的准确度越高。图 6-21简化了实际情况，它假设光轴相互平行，两个图象平面位于同一平面，且场景点与两个平行光轴也位于同一平面。当把这些条件一般化时，会出现更复杂的几何运算，但三角测量这一大体观念不变（动物和一些机器人可把光轴旋转到场景中一个感兴趣的物体的一点上）。

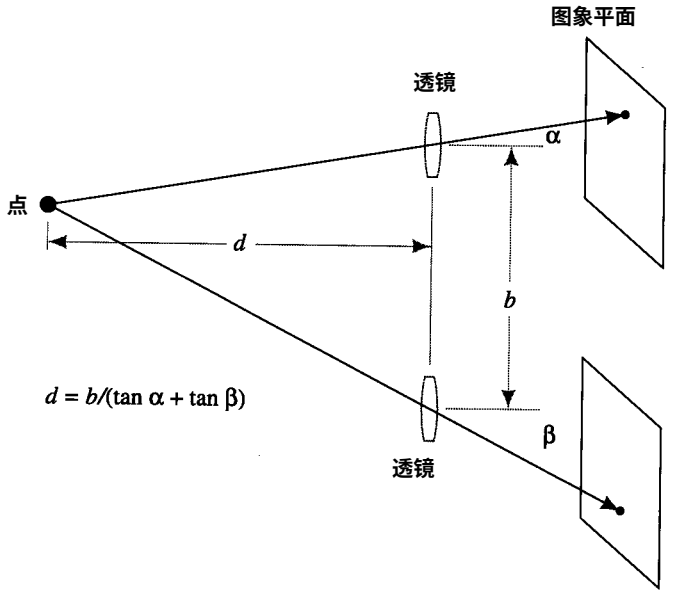


图6-21 立体视觉的三角计算

在立体视觉中，三角计算并不是最复杂的问题。在包含不止一个点的场景中（通常都是这样！），（为了计算出与此场景点的距离）必须判断两个图象中哪两个点与此场景点相对应。换句话说，若与一个场景点相对应的图象点落在一个图象所给定的象素中，我们就必须在另一个图象中找出与其相应的象素。这一过程被称为“对应问题（*correspondence problem*）”。本书篇幅有限，无法细致地讨论有关对应问题的技术，但将作些评价。

首先，几何分析表明，我们只需沿一维空间（而不是二维）搜寻一个图象中给定的一个象素在另一个图象中相应的象素。该一维空间称为“核线（*epipolar line*）”。一维搜寻可通过对两个图象中沿相应核线的亮度轮廓相互结合完成。其次，在众多应用中，我们无须找出图象点个体对之间的关联，而只需找出图象中更大的元素对（如线条[⊖]）之间的关联，而对每个图象的场景分析可提供相互关联的线条对的线索。有关立体视觉的精彩评论，请参阅[Nalwa 1993, 第7章]。

6.7 补充读物和讨论

开发ALVINN的小组又继续开发了其他的自动驾驶系统，有的技术与本章所讨论的内容有关（[Thorpe, et al. 1992]）。[Herbert, et al. 1997]也描述了一个在空旷地形中驾驶且配有立体和红外线传感器的自动交通工具的机动软件。

尽管许多应用要求从计算机视觉系统中获取整个场景的三维模型，但机器人通常只需要足

⊖ 或者是线条相交的图象点。

以引导其行动的信息。与力图计算完整场景模型的早期视觉研究不同，有些研究者却集中力量研究他们所认为的“意图视觉”的更相关的任务。[Horswill 1993]举出了一个简单的面向任务的机器人视觉系统的例子。这些面向任务的视觉系统足以满足汽车机器人的驾驶需要。请参阅[Churchland, Ramachandran, & Sejnowski 1994]。

通过对人类和动物视觉的心理学和神经生理学的研究，我们已经学到了有关视觉感知过程的很多知识。[Gibson 1950, Gibson 1979]还研究了其他一些现象，如变化中的视野是如何给出一个移动物体的环境信息的。[Julesz 1971]发现人类可利用对任一点的统计资料中的不连续点来感知深度。[Marr & Poggio 1997] 创建了一个较为合理的立体视觉神经模型。

对青蛙的实验表明[Letvinn, et al. 1959]，青蛙的视觉系统仅注意整体照明的变化（如一个渐渐逼近的庞大的动物的影子所引起的变化）和小而黑的物体（如苍蝇）的迅速运动。对猴子的实验表明[Hubel & Wiesel 1968]，它们视野中短的定向线段组才能刺激它们视觉皮层中的神经。对马蹄形蟹的实验表明，它们视觉系统中相毗邻的神经元彼此抑制 (*lateral inhibition*)，从而使产生的效果类似后来我们所知的拉普拉斯过滤[Reichardt 1965]。有关生物学视觉的其他资料请参阅[Marr 1982, Hubel 1988]。

[Bhanu & Lee 1994]采用遗传技术来分割图象。

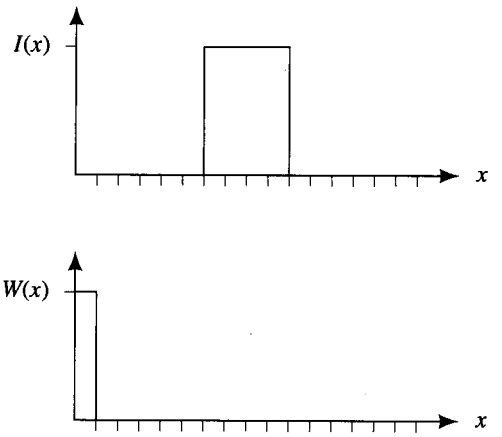
有关计算机视觉的主要会议有国际计算机视觉会议（ ICCV）、欧洲计算机视觉会议（ECCV）以及计算机视觉与模式识别（ CVPR）。权威的期刊是《I nternational Journal of Computer Vision》。

计算机视觉的教材包括[Nalwa 1993, Horn 1986, Ballard & Brown 1982, Jain, Kasturi, & Schunck 1995, Faugeras 1993]。[Fischler & Firschein 1987]为重要的论文集。[Gregory 1996]通俗地说明了人类如何“看”。

习题

6.1 设计一个视觉系统来探测明亮背景中的黑色小物体。设这些物体的图象均为边长为个象素的正方形。你的系统用来产生一个布尔特征，当单个这样的图象出现在一个100×100的图象数组的任一处时，其值为1；而出现在其他图象中时，其值为0。先用一个神经网络（有一个隐藏层面），再用一个特殊设计的加权函数进行卷积。描述一下你如何设计此系统。

6.2 一个一维图象函数$I(x)$，一个一维加权函数$W(x)$如下图所示。请画出$I(x)$ ★$W(x)$的图形。



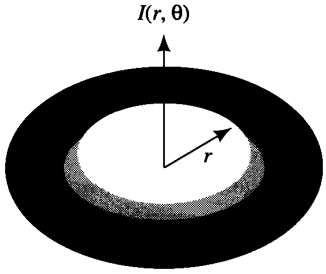
6.3 一个简单的边缘查找（称为“ Roberts cross ”）用以下定义从图象$I(x，y)$中计算出另一个图象$I^{\star}(x, y)$：

$$I^{*}(x,\gamma)=\sqrt{(I(x,\gamma)-I(x+\Delta,\gamma+\Delta))^{2}+(I(x,\gamma+\Delta)-I(x+\Delta,\gamma))^{2}}$$

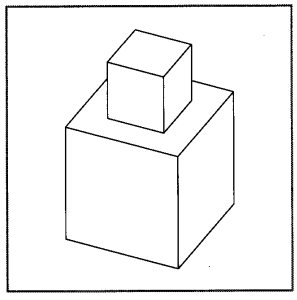
其中，Δ为一个单象素偏移量，并且采用正根。对小的象素，用微积分近似求出这个定义，再用此近似值算出$I^{\star}(x,y)$，其中$I(x,y)$按极坐标来给定：

$$I(r,\theta)=1 \text{ for } r<9$$
$$I(r,\theta)=0 \text{ for } r>10$$
$$I(r,\theta)=10-r \text{ for } 9<r<10$$

假设此圆形图象的半径为20（注意：用极坐标表示此图象有助于你得出最终解！）。图象亮度的草图如下：



6.4 复制下图并用6.5.1节中所讨论的顶点类别和线条标记给图中的线条作标记。若存在不止一种前后一致的标记方法，请列出你所想象的每一种，并描绘每一种的物理解释。



6.5 你能为下图中的著名的复杂图象（被称为“ Penrose三角 ”）中的线条作出一种前后一致的标记吗？假设此图形被悬挂于空荡的空间。讨论一下“ 局部一致 ”与“ 全局一致 ”的关系。