# 第 5 章

CHAPTER 5

# 特征点的检测与匹配

基于标志物实现空间注册的关键在于获得标志物上的 3D 点到图像上相应点的对应关系。这种对应关系可以通过人工平面标志获得(第 3 章),或者通过跟踪器得到(第 4 章),但以上方法对标志物的类型和适用范围有较大的限制。一个自然的问题是能否使用更为广泛的一类物体作为增强现实的标志物? 例如,能否允许平面标志上打印的是任意图案,甚至是用任意非平面的 3D 物体作为标志物? 本章将讨论用自然物体作为标志物的方法。如图 5.1 所示是以一本书的封面作为标志物的增强现实效果。与图 3.1 所示的系统相比,其主要区别在于标志物可以是具有任意图案的平面物体。目前市面上流行的基于增强现实的儿童绘本辅助阅读技术,即是基于同样的原理。当用摄像头扫描书页时,可以基于图像识别定位书页,然后在书页上叠加预先制作好的增强现实动画。

视频演示

(a) 输入图像　(b) 虚实融合结果

(c) 虚拟物体随视角变化　(d) 虚拟物体随标志物运动

图 5.1　基于自然标志物的增强现实(见彩插)

与第 3 章所述的平面标志方法相比,使用自然图像作为标志物的最大困难在于无法通过简单的图像处理精确定位标志物的边界和角点。对于这种情况而言,现有技术主要基于物体表面的纹理信息来识别物体并计算其 3D 位姿,这在很大程度上依赖图像的局部特征

方法。图像的局部特征是在计算机视觉领域需深入研究的问题，在许多视觉系统中已有较为成熟的应用。现有的视觉 3D 重建技术都依赖于足够且稳定的特征点对应，第 6 章也会有所涉及。此外，局部特征方法也是诸如图像拼接、大位移立体匹配等关键技术的基础。在深度学习出现之前，包括图像识别、检测与检索等语义相关的视觉任务，也在很大程度上依赖于图像的局部特征。

## 5.1　特征检测基本原理

特征检测的目标是从给定的输入图像中提取适合用于匹配的点的集合，即特征点。衡量特征点好坏的标准主要有两个。一是可区分性，即特征点所在位置的图像内容要与其他特征点有足够的区分度，以尽量减少错误匹配。例如，在各方向上都没有明显变化的平坦区域是不适合作为特征点的。二是可重复性，即场景中的同一点，在不同观测条件下得到的图像都能被特征检测算法稳定地检测到。此外，由于在实际应用中经常要求系统能够在移动设备等低计算能力的设备上实时运行，因此计算效率也是需要考虑的重要因素。

一般说来，图像中适合作为特征点的像素位置主要包含两类，一类是角点，另一类是斑点。二者虽然在概念上对应不同的图像结构，但实际计算过程存在一定的关联性，以下将分别进行介绍。

### 5.1.1　角点检测

角点原理来源于人对角点的感性判断。如图 5.2 所示，根据局部窗口内所含图像块内容随局部窗口位置移动而变化的特点，可以把图像区域分为以下几类。

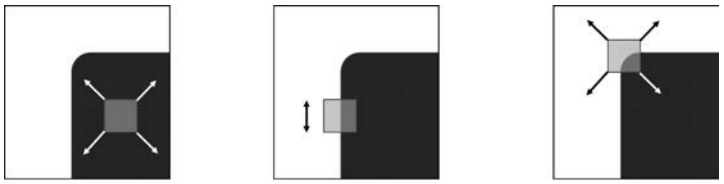



图 5.2　角点检测对图像区域的分类

(1) 平坦区域：窗口沿各个方向移动时所对应的图像块变化不明显。

(2) 边缘：窗口沿垂直于边缘的方向移动时图像块变化明显，而沿边缘方向移动时图像块无明显变化。

(3) 角点：窗口沿各方向移动时图像块变化都很明显。

显然，平坦区域和边缘上的点都不适合作为特征点。平坦区域的相互像素之间没有可区分性，而边缘点与相同边缘上的其他点没有可区分性。

为了描述局部图像块内容随窗口移动时的变化，可以假设窗口 $W$ 发生位置偏移$(u,v)$，比较偏移前后窗口中每一个像素点的灰度变化值，并使用灰度差的平方和来度量图像块内容的变化：

$$E(u,v)=\sum_{(x,y)\in W} w(x,y)(I(x+u,y+v)-I(x,y))^2 \tag{5.1}$$

其中，$w(x,y)$为窗口函数，用于表示窗口中像素$(x,y)$的权重。$I(x+u,y+v)$为平移后的图像灰度；$I(x,y)$为平移前的图像灰度。对于较小的偏移量$(u,v)$来说，可以利用泰勒公式对$I(x+u,y+v)$进行近似：

$$\begin{aligned} I(x+u,y+v) &= I(x,y)+I_x u+I_y v+O(u^2,v^2) \\ &\approx I(x,y)+(I_x\ I_y)\begin{pmatrix}u\\v\end{pmatrix} \end{aligned} \tag{5.2}$$

由此可得：

$$E(u,v)=\sum_{(x,y)\in W} w(x,y)\left((I_x\quad I_y)\begin{pmatrix}u\\v\end{pmatrix}\right)^2 \tag{5.3}$$

其中

$$\left((I_x\quad I_y)\begin{pmatrix}u\\v\end{pmatrix}\right)^2=(u\quad v)\begin{pmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{pmatrix}\begin{pmatrix}u\\v\end{pmatrix} \tag{5.4}$$

由此，误差函数$E(u,v)$可以表示成如下形式：

$$E(u,y)=(u\quad v)\left(\sum_{(x,y)\in w} w(x,y)\begin{pmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{pmatrix}\right)\begin{pmatrix}u\\v\end{pmatrix} \tag{5.5}$$

令

$$\boldsymbol{H}=\sum_{(x,y)\in w} w(x,y)\begin{pmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{pmatrix} \tag{5.6}$$

注意$\boldsymbol{H}$是一个2×2矩阵，只与图像内容和窗口函数相关，而与偏移向量$(u,v)$无关。根据$E(u,v)$的定义，$\boldsymbol{H}$实际上包含了图像块沿各个方向的运动信息。

假设$\lambda_1$和$\lambda_2$是矩阵$\boldsymbol{H}$的两个特征值，相应特征向量分别为$\boldsymbol{v}_1$、$\boldsymbol{v}_2$。不妨假设$\lambda_1\geqslant\lambda_2$，则$\boldsymbol{v}_1$实际上是$E(u,v)$变化最快的方向，而$\boldsymbol{v}_2$则是$E(u,v)$变化最慢的方向。因此，给定$\lambda_1$和$\lambda_2$，就可以按以下规则对图像区域进行分类。

(1) 如果$\lambda_1$和$\lambda_2$都很小，说明图像在该像素处沿各方向变化都不明显，因此相应像素应属于平坦区域。

(2) 如果$\lambda_1$和$\lambda_2$中一个较大，另一个较小，且二者差异较大，则说明图像在该像素处只在一个方向变化明显，因此相应像素应在图像边缘附近。

(3) 如果$\lambda_1$和$\lambda_2$的值都比较大，且二者数值相当，则说明图像在该像素处沿各方向变化都比较明显，因此相应像素是图像的角点。

在实际计算过程中，为了减小计算量，可以不显式计算特征值$\lambda_1$和$\lambda_2$。注意到矩阵$\boldsymbol{H}$的行列式$\det(\boldsymbol{H})=\lambda_1\lambda_2$，而迹$\operatorname{trace}(\boldsymbol{H})=\lambda_1+\lambda_2$，因此可以定义以下角点响应指标：

$$R=\det(\boldsymbol{H})-\alpha\operatorname{trace}(\boldsymbol{H})^2=\lambda_1\lambda_2-\alpha(\lambda_1+\lambda_2)^2 \tag{5.7}$$

其中$\alpha$是一个常数，通常取值为0.04～0.06。显然，响应值$R$只在$\lambda_1$和$\lambda_2$都较大时才较大，因此可以根据$R$的大小鉴别角点。上述角点检测方法即著名的哈里斯(Harris)角点检测。

### 5.1.2 斑点检测

斑点是图像中易于检测和定位的另一类结构。不同于角点，斑点通常为图像中的一个

区域，面积较小且与区域外有明显差异。理想的斑点是一个 2D 高斯函数，如图 5.3(a)所示。图 5.3(b)显示了一幅向日葵的图像，其中每朵向日葵都可以看作是图像中的一个斑点。

(a) 高斯函数形成的斑点　　(b) 向日葵形成的斑点区域

图 5.3　斑点图像

基于斑点所在区域的内外图像差异明显的特点，斑点检测的基本思想是计算以某一点为中心的内外区域差异，并以此作为斑点响应函数。很明显，图像的拉普拉斯(Laplacian)算子即是这样的操作。在数学上，2D 函数 $f(x,y)$的拉普拉斯算子定义为

$$\nabla^2 f=\frac{\partial^2 f}{\partial x^2}+\frac{\partial^2 f}{\partial y^2} \tag{5.8}$$

图像的拉普拉斯是对上式的离散化，基于图像求导法则很容易得到：

$$\nabla^2 f=4f(x,y)-(f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)) \tag{5.9}$$

可见标准拉普拉斯算子计算的是中心像素 $f(x,y)$与其四邻域像素值之差，可以反映像素与其邻域像素的差异大小。

最早的斑点检测器是 Beaudet 等人提出的 Hessian 检测器。将图像看作定义在 2D 空间域上的函数，在每个像素处可以计算其 Hessian 矩阵：

$$\boldsymbol{H}=\begin{pmatrix}\dfrac{\partial^2 I}{\partial x^2} & \dfrac{\partial^2 I}{\partial x\partial y}\\ \dfrac{\partial^2 I}{\partial x\partial y} & \dfrac{\partial^2 I}{\partial y^2}\end{pmatrix} \tag{5.10}$$

可见，Hessian 矩阵主要包含对图像的二阶导，且拉普拉斯算子即 Hessian 矩阵的迹。实际上，如果将像素值看作是每个像素处的高度值，则图像等同于一曲面函数。Hessian 矩阵所刻画的是曲面上每点沿各方向的曲率变化情况，其最大和最小特征值对应的特征向量分别对应于曲面上曲率变化最快和最慢的方向。这一点非常类似于角点检测器中自相关矩阵的性质，因此也可以通过分析 Hessian 矩阵的特征值来检测斑点。在 Hessian 检测器中采用矩阵 $\boldsymbol{H}$ 的行列式作为特征的响应值，这是因为 $\det(\boldsymbol{H})=\lambda_1\lambda_2$ 只有在特征值 $\lambda_1$ 和 $\lambda_2$ 都比较大时才能取得较大值，而计算行列式比计算特征值的计算量要小得多。

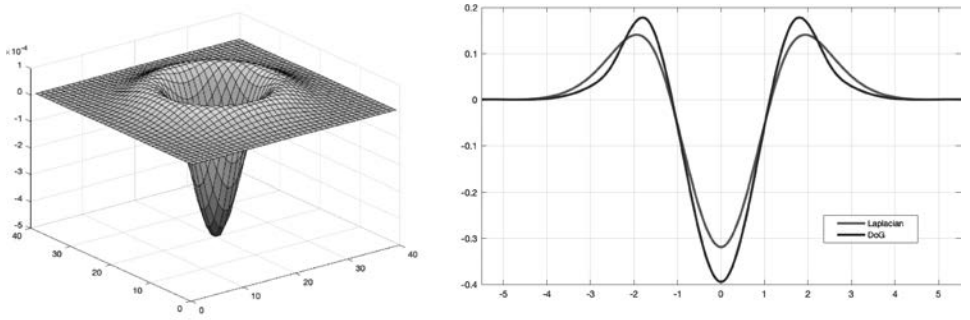
与 Hessian 检测器相比，高斯-拉普拉斯(Laplacian-of-Gaussian，LoG)算子是一种更直观且更适合多尺度扩展的斑点检测器。对于 2D 的高斯函数来说

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}\mathrm{e}^{-(x^2+y^2)/2\sigma^2} \tag{5.11}$$

其拉普拉斯变换为

$$\mathrm{LoG}(x,y)=\frac{\partial^2 G}{\partial x^2}+\frac{\partial^2 G}{\partial y^2}=\frac{1}{\pi\sigma^4}\left(\frac{x^2+y^2}{2\sigma^2}-1\right)\mathrm{e}^{-(x^2+y^2)/2\sigma^2} \tag{5.12}$$

LoG 算子在 2D 图像上显示为一个圆对称函数，如图 5.4 所示。




(a) LoG算子　　(b) LoG算子在$y$=0处的截面

图 5.4　图像的 LoG 算子

LoG 算子是以 LoG 函数为核函数的空间域滤波器。如图 5.4 所示，LoG 函数中心的权重为负值，而外侧周围的权重为正值，因此当其与图像卷积时，所得结果是内外区域的差异值。以 LoG 算子滤波的结果为斑点检测的响应值，再经过非极大值抑制和阈值过滤，便可以得到特征点集合。注意通过改变 $\sigma$ 的值可以调整 LoG 函数的形状，$\sigma$ 越大，LoG 函数中的内部区域半径就越大，相应地可以检测出半径（尺度）更大的斑点。

为了更高的计算效率，也可以采用 DoG（Difference of Gaussian）算子近似计算 LoG 函数。DoG 表示两个相近尺度高斯函数的差，可以理解为以不同 $\sigma$ 对图像进行高斯滤波的结果的差值。要理解 DoG 与 LoG 的关系，首先应验证式（5.11）所示的 2D 高斯函数，有以下等式成立：

$$\frac{\partial G}{\partial\sigma}=\sigma\,\nabla^2 G \tag{5.13}$$

其中$\nabla^2 G$ 即 LoG 算子。利用差分近似微分可以得到：

$$\sigma\,\nabla^2 G=\frac{\partial G}{\partial\sigma}\approx\frac{G(x,y,k\sigma)-G(x,y,\sigma)}{k\sigma-\sigma} \tag{5.14}$$

因此

$$G(x,y,k\sigma)-G(x,y,\sigma)\approx(k-1)\sigma^2\,\nabla^2 G \tag{5.15}$$

其中 $k\sigma$ 表示 $\sigma$ 的一个相邻尺度（如高斯金字塔中的一个相邻层次）。式（5.15）左边即高斯的差分（DoG），因此 DoG 和 LoG 之间只差一个常量的缩放。在 5.3.1 节中将看到，采用 DoG 算子可以显著降低 LoG 金字塔的计算开销。

### 5.1.3　尺度不变性

物体在图像中的尺度对应于其在图像中所占区域的大小。图像分辨率、图像拍摄时物体离相机的距离、相机参数等都可能引起物体的尺度变化。因此，当同样的物体在不同图像中出现时，其尺度可能出现较大的差异，如图 5.5 所示。如何实现不同尺度的区域之间的稳

定匹配，是特征检测和匹配方法需要解决的首要问题。

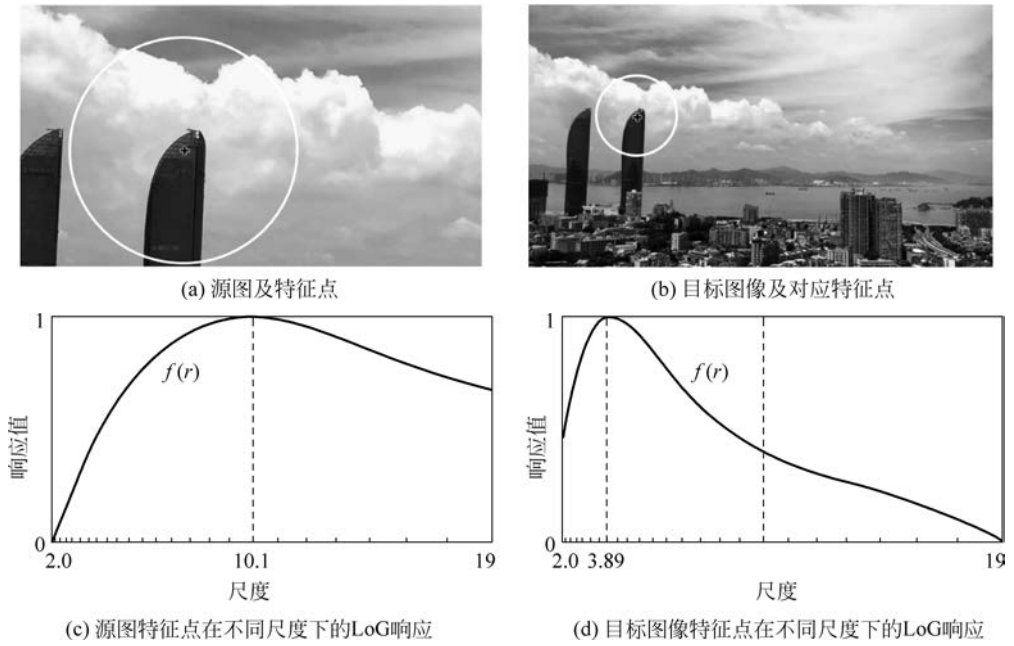



图 5.5　图像的尺度与尺度选择理论示意图

实现尺度不变的难点在于输入图像中物体的尺度是未知的。因此，为了应对特征点的尺度变化，一种方法是对源图像中的每个特征点，搜索其在目标图像中所有可能的尺度变化。这要求计算目标图像中每个特征点在所有可能尺度上的特征描述，并与源图像中的特征点进行匹配。这一方面会导致显著的冗余计算，另一方面会因为引入大量不稳定特征而导致更多的错误匹配。

是否存在一个方法，能够为每个特征点赋予一个尺度半径 $r$，并且当物体尺度改变时，$r$ 能够相应地进行调整？如图 5.5 所示，虽然物体尺度相差较大，但尺度变化前后圆环内部的图像内容基本是一致的。

尺度选择理论可用于解决上述问题。给定图像上一个特征点 $x$，可以定义一个能够反映以 $x$ 为中心，半径为 $r$ 的区域特征的特征函数 $f(r)$。特征函数的定义可以是多样化的，如用于斑点检测的 LoG 算子，即是较为理想的特征函数。LoG 算子实际上反映了区域内外的像素值差异。因此，考虑图像上一圆形斑点，LoG 算子将在其权值正负的边界与圆形斑点边界重合时达到最大响应值。基于这个观察，尺度选择理论的基本思想在于通过改变特征函数 $f$ 的尺度半径参数 $r$，可以得到 $f$ 在 $x$ 点处关于 $r$ 的响应曲线。选择响应曲线中最大值对应的尺度半径，作为特征点 $x$ 的关联尺度，并称其为 $x$ 的特征尺度。

由于采用上述方法计算的特征尺度是跟图像内容相关的，因此当图像缩放时，特征尺度会随图像内容的变化而改变。例如，当图像被缩放到原始大小的 1/2 时，特征函数的响应曲线会被横向压缩，其最大值点对应的尺度也将变为原来的 1/2，由此可以实现尺度不变性。注意，由于实际情况一般不会如此理想，因此，响应曲线中会出现多个局部极大值点。这种情况下，一般会把所有局部极大值点的位置都选择为特征尺度。这也是为什么常见的尺度

不变特征检测方法，如尺度不变特征变换（scale invariant feature transform，SIFT）等，会在同一位置检测到多个特征点的原因。多个特征点虽然空间位置相同，但具有不同的特征尺度。

## 5.2 特征匹配基本原理

特征检测可以对每一幅输入图像检测到一系列的特征点。如果待匹配的图像之间有明显的尺度变化，则可以进一步基于尺度选择原理为每个特征点决定一个尺度半径，并在进行特征匹配之前将特征点周围的相应图像区域缩放到同一尺度。因此，在特征匹配阶段，可以不考虑图像的尺度变化，这样，特征匹配就等同于比较两个相同大小图像块的相似性问题。假设 $\varphi(P,Q)$ 是计算两个图像块 $P$、$Q$ 之间差异的函数，且 $P$、$Q$ 具有相同的大小。特征匹配的关键在于定义适当的 $\varphi(\cdot)$，以使得匹配特征点对之间具有较小的差异，而不匹配的特征点对之间具有较大的差异。

注意，即使图像块 $P$、$Q$ 来自一对匹配的特征点，其对应的像素值也可能有较大的差异。这主要是由于两方面因素的影响：一是图像拍摄环境的光照变化，或者相机曝光度的变化等，这会导致图像块之间不同的亮度、颜色等；二是图像块内部的几何形变，这主要由物体形变或观测视角的变化引起，将导致对应像素之间没有严格对齐。上述两方面原因可能导致匹配的特征点对之间计算得到的相似性较低，从而无法获得正确的匹配，这是特征匹配需要解决的主要问题。

### 5.2.1 处理像素值变化

首先假设待匹配图像块之间没有几何形变。注意这种情况一般并不成立，但是如果待匹配图像差异较小，如对视频中的连续两帧图像来说，可以认为局部图像块 $P$、$Q$ 之间的几何形变是可以忽略的。在这种情况下，一个最简单且常用的图像块匹配度量是对应像素值差的平方和（sum of square difference，SSD）：

$$\varphi^{\mathrm{ssd}}(P,Q)=\sum_{x\in\Omega}\|P(\boldsymbol{x})-Q(\boldsymbol{x})\|^2 \tag{5.16}$$

其中 $\Omega$ 表示图像块所在区域，$\boldsymbol{x}$ 是图像块内的一个像素位置。SSD 完整地保留了图像块之间的颜色差异，在图像之间不存在明显像素值变化的时候具有较好的表现。不过，由于其平方项对像素值的较大差异增长很快，因此对像素值变化和几何形变都很敏感。为此，可以将平方项改为绝对值，即采用像素值差的绝对值和（sum of absolute difference，SAD）：

$$\varphi^{\mathrm{sad}}(P,Q)=\sum_{x\in\Omega}|P(\boldsymbol{x})-Q(\boldsymbol{x})| \tag{5.17}$$

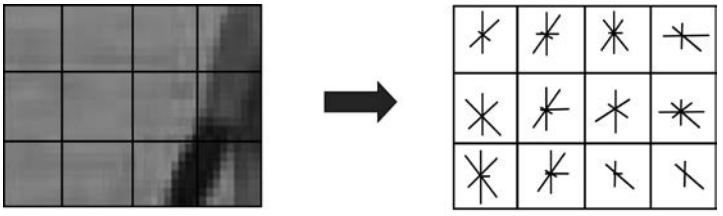
不过，SAD 对于像素值变化的影响仍然具有线性响应。为获得更好的稳定性，更常采用的度量函数是归一化互相关性（normalized cross-correlation，NCC）：

$$\varphi^{\mathrm{ncc}}(P,Q)=\left\langle\frac{P-\bar{P}}{\|P-\bar{P}\|},\frac{Q-\bar{Q}}{\|Q-\bar{Q}\|}\right\rangle \tag{5.18}$$

这里把 $P$、$Q$ 看作是由像素值依次排列组成的向量，$\langle,\rangle$ 表示向量的内积，$\bar{P}$、$\bar{Q}$ 表示图像块内所有像素值的均值。注意 $\varphi^{\mathrm{ncc}}$ 是一个相似性函数而不是差异函数，其值的范围为 $[-1,1]$，值越小表示差异越大；如果 $P$、$Q$ 相等，则 $\varphi^{\mathrm{ncc}}(P,Q)=1$。为了理解 NCC 对像素值变化的

稳定性，可以假设 $P$、$Q$ 之间的像素值变化可以用一个线性函数来拟合，即存在 $a$、$b$，使得 $Q(\boldsymbol{x})=aP(\boldsymbol{x})+b$。容易证明，NCC 对线性的像素值变化是不变的，即如果 $P$、$Q$ 之间只有线性变化，则 $\varphi^{\text{ncc}}(P,Q)=1$。

另一种对像素值变化稳定的方法是图像的方向梯度直方图（histograms of oriented gradients，HOG）。由于 HOG 方法基于图像的梯度计算图像块的相似度，因此对像素值变化也具有很好的稳定性。图 5.6 显示了 HOG 的基本原理。对给定的图像块来说，首先将图像块均匀分为相同大小的网格，然后对每个网格里的小图像块，统计像素的梯度方向直方图。注意，每个像素 $\boldsymbol{x}$ 的梯度是一个 2 维向量 $(\theta_x,\omega_x)$，其中 $\theta_x$ 表示梯度方向，$\omega_x$ 表示梯度强度。每个网格里的梯度直方图一般把 0 到 $2\pi$ 的方向分为 8 份，然后计算落在每个角度范围内的梯度强度并以此之和作为直方图的值。将每个小网格的直方图相连，便得到了整个图像块的 HOG 特征。如果图像块被分为 $m$ 个小网格，则整个图像块的特征是一个长度为 $8m$ 的向量，再对该向量进行归一化就得到了最终的 HOG 特征。获得 HOG 特征之后，两个图像块的差异可取为相应 HOG 特征的欧氏距离。



(a) 输入图像块及其划分　　(b) 梯度方向直方图

图 5.6　计算梯度方向直方图

注意图像梯度实际上是邻近像素的差分，所以很容易验证 HOG 对线性的光照变化也是不变的。此外，由于 HOG 是对一定区域内像素的统计特征，因此相对于 SSD、NCC 等方法，HOG 对像素位置的变化也更不敏感，可以在一定程度上处理图像块的几何形变和对齐误差。

NCC 和 HOG 虽然都可以较好地处理线性的像素值变化，但计算开销都比较大。对计算性能要求较高的情况，通常都采用二值特征来计算图像块之间的差异。如图 5.7 所示为一种典型的二值特征编码方法。对一个 $3\times3$ 的图像块来说，将图像块 $P$ 内非中心像素 $\boldsymbol{x}$ 与中心像素 $\boldsymbol{c}$ 进行比较，并按以下规则进行编码：

$$\hat{P}(\boldsymbol{x})=\begin{cases}0 & P(\boldsymbol{x})\leqslant P(\boldsymbol{c})\\ 1 & P(\boldsymbol{x})>P(\boldsymbol{c})\end{cases} \tag{5.19}$$

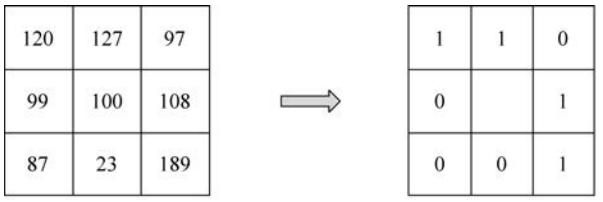
经过上述变换之后，$\hat{P}$ 可以被看作是一个二进制串。对待比较的图像块 $P$、$Q$，可以通过 $\hat{P}$、$\hat{Q}$ 中不同的二进制位的个数来计算其差异：

$$\varphi^{\text{binary}}(P,Q)=\sum_{\boldsymbol{x}\in\Omega}\hat{P}(\boldsymbol{x})\,\text{xor}\,\hat{Q}(\boldsymbol{x}) \tag{5.20}$$

其中，xor 表示异或操作。

二值特征不仅计算高效，而且对图像的噪声和非线性的像素值变化也有较好的稳定性，因此在图像匹配中得到了广泛关注。针对局部特征描述，一种代表性的方法是位特征描述子（binary robust independent elementary features，BRIEF）。与图 5.7 所示的二值编码方

法相比，BRIEF 的区别在于其像素值比较不再固定与中心像素进行，而是可能发生于任意两个像素之间。根据一定的规则，采样一定数量的像素对，并比较像素对的相对亮度关系，生成二值编码。像素对可以随机生成，也可以按一定规则生成（如在靠近中心的区域采样更多等）。



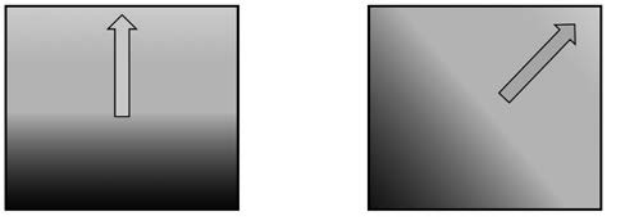

(a) 图像块的灰度值　　(b) 图像块的二值特征

图 5.7　图像块的二值特征编码方法

## 5.2.2　旋转不变性

考虑图像块 $P$、$Q$ 可能包含几何变换和变形，其中最重要的也是最基本的两种变换即缩放和旋转。尽管整幅源图像和目标图像之间的几何形变可能很复杂，但对于其中包含的一个局部图像块，其几何形变一般可以较好地用一个 2D 相似变换来近似表示。如 5.1.3 节所述，图像块的缩放可以通过尺度不变特征检测来处理，在特征匹配阶段主要考虑图像块的旋转。

现有特征匹配方法处理旋转不变性的思路大致相同。如图 5.8 所示，假设图像块 $Q$ 是 $P$ 经过旋转变换之后的结果，则 $Q$ 的像素可以经过旋转一定角度与 $P$ 中像素对齐。因此，如果能为每个图像块根据其像素值的空间变化规律计算一个方向向量 $\boldsymbol{\nu}$，则 $\boldsymbol{\nu}$ 也会随着图像块的旋转一起旋转。假设已经计算得到图像块 $P$、$Q$ 对应的方向向量 $\boldsymbol{\nu}(P)$、$\boldsymbol{\nu}(Q)$，则可以很容易地通过对 $P$ 或 $Q$ 旋转一定角度来消除它们之间的旋转差异。



(a) 原图像及其主方向　　(b) 旋转后的图像及其主方向

图 5.8　图像块的主方向与旋转不变性

上述过程中的方向向量 $\boldsymbol{\nu}$ 称为图像块的主方向。可见，实现旋转不变性的关键在于正确估计图像块的主方向。一个估计主方向的最简单的方法是计算图像块中心的像素梯度。不过，由于单个像素的梯度方向很不稳定，因此一般不采用这种方法。在 5.3 节中，将结合具体的特征匹配技术，介绍几种代表性的主方向估计方法。

# 5.3　特征检测与匹配的代表性方法

在 5.1 和 5.2 节中，介绍了特征检测与匹配的一些基本原理。本节将结合几种代表性的局部特征方法（SIFT、SURF 和 ORB），介绍更多与实现相关的细节。

### 5.3.1　SIFT 特征

尺度不变特征变换(scale invariant feature transform,SIFT)是最早被广泛采用的局部特征方法。直到现在,如果不考虑计算性能,大部分情况下仍会采用 SIFT 方法。SIFT 特征检测主要基于 5.1 节介绍的斑点检测原理,基于图像金字塔实现了多尺度(尺度不变)的特征检测与匹配。

如图 5.9(a)所示是 SIFT 中图像金字塔的构造过程。对输入图像而言,首先构造如图 5.9(a)所示的高斯金字塔。注意该金字塔与普通高斯金字塔有很大不同。在普通的高斯金字塔中,相邻层的降采样率一般为 2,其构造过程是先用选定的高斯核函数 $G(x,y,\sigma)$ 对第 $l$ 层进行高斯滤波,再将滤波后的图像降采样为第 $l$ 层分辨率的一半,所得结果即为第 $l+1$ 层。但是,对于图像匹配来说,由于物体尺度一般是连续变化的,因此普通高斯金字塔在尺度空间的采样显然过于稀疏,这会导致位于中间尺度的特征点无法正确匹配。为此,需要提高对尺度空间的采样率。SIFT 中的高斯金字塔构造方法正是出于提高尺度空间采样率的目的而设计的。注意,其中有连续的 $k$ 层图像大小相同,如在图 5.9(a)中 $k=5$,这样连续的 $k$ 层被称为一个八度组(Octave,原意是音乐中的八度音阶)。降采样只在两个相邻的八度组之间进行(将图像缩小一半),而在同一个八度组内,每次仅等量地增大高斯滤波的尺度参数 $\sigma$。所以,一个八度组即相当于普通高斯金字塔中的一层,$k$ 值越大,则在尺度空间的采样率越高。

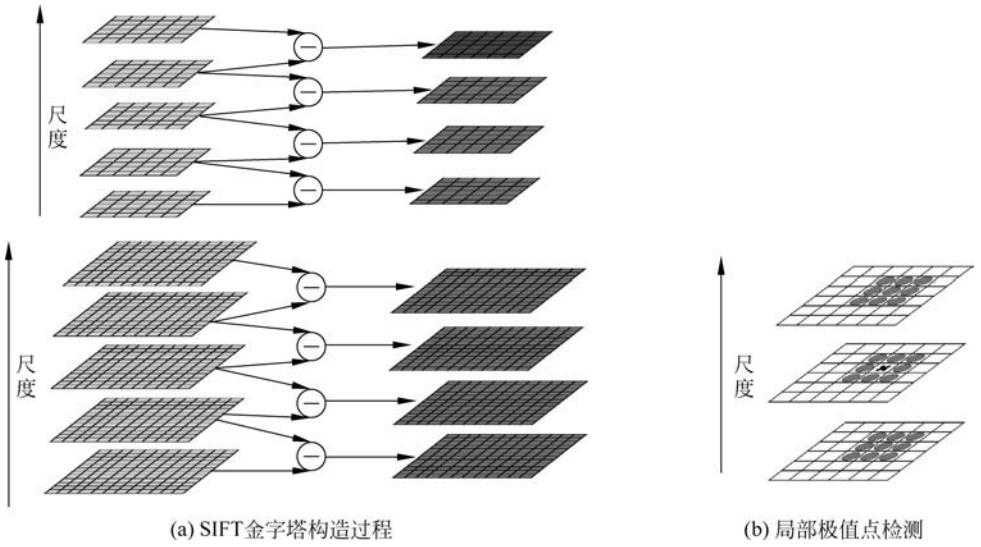



图 5.9　SIFT 特征提取算法

高斯金字塔构造完成后,再基于 5.1.2 节介绍的 DoG 算子计算拉普拉斯(Laplacian)金字塔。采用 DoG 算子计算拉普拉斯金字塔可以显著减少计算量。图像金字塔构造完成之后,可以在金字塔中检测尺度空间的局部极值点,并以此作为特征点的候选。如图 5.9(b)所示,检测局部极值点需要将每个样本点跟它的 8 个最近邻点及其上、下相邻尺度空间中各 9 个最近邻点(总共 26 个最近邻点)进行比较,如果某个样本点比所有 26 个近邻点都大或都小,则该样本点被认为是一个局部极值点。注意,这样得到的每个极值点都有一个 3D 坐

标$(x,y,\sigma)$，其中$(x,y)$是特征点的空间坐标，$\sigma$则是特征点的尺度，相当于尺度选择理论中得到的特征尺度，可用于决定计算特征描述符的尺度半径。

通过极值点检测得到的只是特征点的候选，要获得最终的特征点，还需要经过两个步骤：一是优化特征点坐标$(x,y,\sigma)$，获得亚像素的坐标值，提高特征点定位的精度；二是移除低响应值的点和边缘点。

SIFT的特征描述主要基于5.2节介绍的HOG描述子。具体地，对特征点$(x,y,\sigma)$来说，首先根据$\sigma$确定的采样间隔，以$(x,y)$为中心采集一个16×16网格像素的梯度。再将16×16的网格均匀分为4×4的子网格，并在每个子网格内计算一个长度为8的梯度方向直方图。因此，每个特征点将获得一个长度为4×4×8=128的特征描述子。

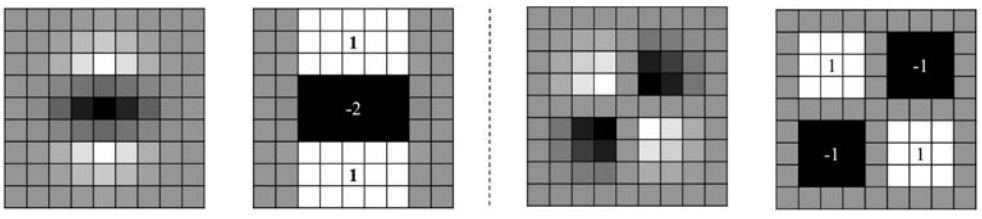
如5.2节所述，获得旋转不变性的关键在于估计特征点的主方向。SIFT的主方向估计也是基于梯度直方图。对特征点关联区域内像素的梯度方向分布进行统计，取强度最大的方向作为主方向。此外，如果某个方向的强度大于最大强度的80%，则该方向也被认为是一个主方向，这时需要新增加一个特征点来表示该方向的特征描述子。这也是为什么在SIFT特征检测的结果中，往往有多个特征点具有完全相同的空间坐标。这样做的主要目的是增强主方向估计的稳定性。根据SIFT原论文的描述，只有不到15%的特征点会被赋予多个主方向，但是这样做可以显著改善特征匹配的效果。

### 5.3.2 SURF特征

加速稳健特征(speeded up robust features，SURF)的提出主要是为了对SIFT特征的计算进行加速。SURF特征检测基于5.1.2节介绍的Hessian斑点检测原理。为了进行多尺度检测，首先定义尺度空间的Hessian矩阵为

$$\boldsymbol{H}(\boldsymbol{x},\sigma)=\begin{pmatrix}L_{xx}(\boldsymbol{x},\sigma) & L_{xy}(\boldsymbol{x},\sigma)\\ L_{xy}(\boldsymbol{x},\sigma) & L_{yy}(\boldsymbol{x},\sigma)\end{pmatrix} \tag{5.21}$$

其中$\boldsymbol{x}$是像素的空间坐标，$\sigma$是尺度参数，$L_{xx}$、$L_{xy}$、$L_{yy}$是2D高斯函数$G(x,y,\sigma)$的二阶偏导$G_{xx}$、$G_{xy}$、$G_{yy}$分别与图像卷积后的结果。为了加速Hessian矩阵的计算，在SURF中采用均值滤波来近似计算$L_{xx}$、$L_{xy}$、$L_{yy}$。如图5.10所示，与图像的拉普拉斯算子一样，卷积核$G_{xx}$、$G_{xy}$、$G_{yy}$实际上也是对不同区域内像素的差分，因此可以用均值滤波来近似。采用均值滤波的优点在于其计算可以通过积分图进行加速。对一幅输入图像而言，只需要构造一次积分图，之后任意窗口大小(任意尺度)的均值滤波都可以在常数时间复杂度内完成。SURF特征的响应值即$\boldsymbol{H}$矩阵的行列式，与SIFT特征一样，SURF也是将尺度空间的3D极值点作为特征点的候选。




(a) $L_{yy}$　(b) $L_{yy}$近似　(c) $L_{xy}$　(d) $L_{xy}$近似

图5.10　采用均值滤波近似计算Hessian矩阵

SURF 特征描述子的计算过程如图 5.11 所示。对每个特征点$(x,y,\sigma)$,以$(x,y)$为中心,大小为 $20s\times20s$ 的子窗口内像素进行计算,其中 $s$ 由 $\sigma$ 决定,对应于特征的尺度大小。每个 $s\times s$ 大小的图像块称为一个子区域,$20\times20$ 的子区域被进一步均匀划分为 $4\times4$ 的网格,每个子网格包含 $5\times5$ 的子区域。在每个子网格内,首先在每个子区域位置上,利用图 5.11 所示的小波核计算 $x$ 和 $y$ 方向的图像梯度 $\mathrm{d}x$ 和 $\mathrm{d}y$,这实际上是在每个 $2s\times2s$ 的图像块内进行水平或垂直方向上的差分,这样计算得到的图像梯度称为小波梯度。然后对每个子网格包含的 $5\times5$ 个子区域,计算一个描述子:

$$v=\left(\sum \mathrm{d}x,\sum \mathrm{d}y,\sum|\mathrm{d}x|,\sum|\mathrm{d}y|\right) \tag{5.22}$$

最后将所有 $4\times4$ 个子网格的描述子连接,即得到 SURF 的特征点描述子。因此,每个特征点的描述子大小为 $4\times4\times4=64$。

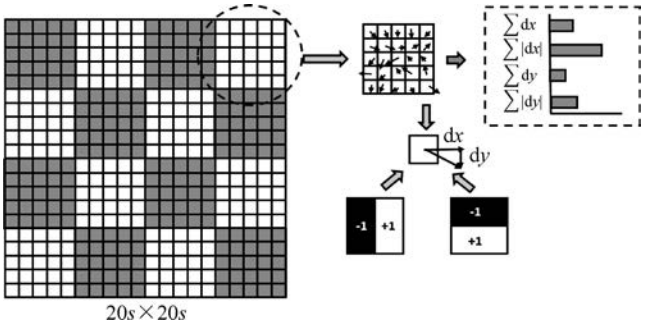



图 5.11　SURF 特征描述子的计算过程

SURF 主方向与 SIFT 主方向的计算具有相同的原理,都是选梯度分布最强的方向作为主方向,但是实现方式不同。首先,在 SURF 中,梯度计算采用的是小波梯度。其次,为了计算最强的梯度方向,SURF 中采用 $\pi/3$ 大小的角度扫描窗进行搜索,这与 SIFT 中采用梯度直方图进行搜索在很大程度上是等价的。

### 5.3.3　ORB 特征

方向快速特征及位特征描述子(oriented-fast and rotated brief,ORB)是比 SIFT 和 SURF 更为高效的特征,往往被用于实时性要求比较高的应用中(如 SLAM)。从其命名可以看出,它实际上结合了 FAST 角点检测和 BRIEF 特征描述子。虽然快速特征(features from accelerated segment test,FAST)是一种非常快速的角点检测方法,但是没有为角点估计主方向,也没有进行多尺度的检测。BRIEF 在 5.2.1 节中已有介绍,是一种高效的二值特征描述子,但没有考虑尺度和旋转不变性。ORB 将二者进行了结合,解决了尺度不变和旋转不变的问题。

在特征检测阶段,ORB 首先构造图像金字塔,并在金字塔的每一层上进行 FAST 角点检测,以得到特征点的候选。对候选特征点而言,计算 Harris 角点响应,并基于角点响应值消除弱特征点和边缘点,剩余的特征点即为 ORB 特征点,相应的金字塔的层数用于决定计算特征描述子的尺度半径,并基于 BRIEF 方法计算特征描述,最终生成长度为 256 的二进制串描述。

在 ORB 中,主方向的估计采用了一种被称为“亮度质心”的原理。对于以一个特征点

为中心的某个区域 $\Omega$ 来说，首先定义

$$m_{pq}=\sum_{(x,y)\in\Omega}x^p y^q I(x,y) \tag{5.23}$$

亮度质心于是可以计算为

$$C=\left(\frac{m_{10}}{m_{00}},\frac{m_{01}}{m_{00}}\right) \tag{5.24}$$

上式中 $m_{00}$ 实际上是区域内像素亮度的均值，$m_{10}$ 和 $m_{01}$ 分别是以像素亮度为权重对 $x$ 和 $y$ 坐标的加权平均。所以，亮度质心会往亮度较大的方向偏移。获得亮度质心后，主方向即特征点到亮度质心连线的方向，可以计算为

$$\theta=\arctan2(m_{01},m_{10}) \tag{5.25}$$

注意，如果区域内像素的亮度都比较低，则亮度质心的计算是不稳定的。不过，由于特征点所在位置一般是像素亮度变化较大的位置，因此这种情况一般很少出现。

## 5.4 基于特征点对应的位姿估计

在第 3 章和第 4 章中，介绍了基于点对应估计标志物 3D 位姿，进而实现空间注册的基本方法。对自然标志物而言，可首先利用本章所述特征匹配方法获得输入图像与标志物模板之间的特征点对应，再基于点对应估计标志物的当前位姿。然而，由于特征匹配获得的特征点对应都包含一定的误匹配，且实际情况下误匹配的比率往往较高（超过 50%），因此直接基于这些点对应估计位姿参数将难以获得准确的结果。本节将主要讨论在包含误匹配的情况下进行位姿估计的方法。

### 5.4.1 鲁棒最小二乘

常规的最小二乘方法在噪声符合正态（高斯）分布时是一个合适的选择。但是，在实际情况中，对应点中往往存在着外点。所谓外点，即误差较大的对应点，也可以被称为离群点。相应地，误差很小的对应点称为内点。由于外点在最小二乘法中计算出的残差可能会非常大，因此，对计算结果可能会有破坏性的影响，如果考虑外点，则一般不能采用普通的最小二乘进行参数估计。图 5.12 显示了在直线拟合问题中，外点对拟合结果的影响。可见即使只有一个外点，也可能对结果造成极大的影响。

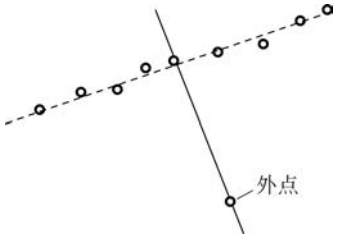



图 5.12 外点对最小二乘结果会有破坏性的影响

为了改善最小二乘法在有外点情况下的稳定性，可以采用鲁棒的最小二乘法：

$$E_{\mathrm{RLS}}(\Delta\boldsymbol{\Theta})=\sum_i \|r_i\|^q \tag{5.26}$$

在最小二乘中，由于 $q=2$，因此惩罚能量将随着残差的增大而快速增长。为了减小外点对结果的影响，需要取 $q<2$，$q$ 越小，外点对结果的影响也越小。为了求解，可以将式(5.26)写为

$$E_{\mathrm{RLS}}(\Delta\boldsymbol{p})=\sum_i \|r_i\|^q=\sum_i \|r_i\|^{q-2}\|r_i\|^2=\sum_i \omega(r_i)\|r_i\|^2 \tag{5.27}$$

其中 $\omega(r_i)=\|r_i\|^{q-2}$ 可以被看作是一个权重函数，并基于当前的残差值进行计算。对给定的权重而言，式(5.27)变成了一个加权最小二乘问题，可以采用线性方法进行求解。初始时，可以设置 $\omega(r_i)=1$，即求解一个普通的最小二乘问题作为初始解，随后更新残差和权重函数，如此迭代直至收敛。上述方法被称为迭代重加权最小二乘(iteratively reweighted least squares，IRLS)。

采用鲁棒最小二乘可以在一定程度上减少外点的影响，但是如果外点数量较多或误差很大，基于普通最小二乘估计的初始解也会与真值有较大的偏差，这会导致 IRLS 和类似梯度下降的算法不能收敛至最优解。因此，鲁棒最小二乘只适用于外点影响不太大的情况。

## 5.4.2　随机采样一致性算法

为了更稳定地处理外点，可以采用随机采样一致性(random sample consensus，RANSAC)算法。RANSAC 是一种随机算法，可以处理较高比例和很大误差的外点。

RANSAC 算法的思想非常简单，在包含外点的数据集中，采用不断迭代的方式，去寻找最优的参数模型。对于每次迭代而言，都执行以下过程。

(1) 随机选取一小组点，然后用这组点拟合一个参数化模型 $\theta_i$。选取点的个数遵循“最小采样”原理，即估计一个参数模型所需的最小点数。例如，2D 仿射变换一共有 6 个自由度，则至少需要 3 个 2D 点对；透视变换一共有 8 个自由度，则至少需要 4 个点对。

(2) 用当前模型 $\theta_i$ 去测试其他所有点，如果某个点与模型的偏差小于一定阈值 $\varepsilon$，则认为该点与当前模型一致。找出所有与 $\theta_i$ 一致的点的集合，并记一致点的数量为 $m_i$。

重复执行上述过程，直至找到一个足够好的模型($m_i$ 足够大)，或者达到最大迭代次数，并选 $m_i$ 最大的一次作为结果。为了改善结果的精度，一般在迭代结束后，还需要再用所有估计出来的内点，用最小二乘法重新拟合模型。上述过程中的第 1 步称为“采样”，第 2 步称为“验证”。采样的目的是生成假设模型，而验证则是评估模型的正确性。因此，RANSAC 算法即是通过不断地“假设-验证”选择最优模型的过程。

需要注意，RANSAC 算法并不能保证收敛到全局最优解。理论上，只有至少有 1 次采样得到的点集不包含外点时，RANSAC 算法才有可能得到正确解。不妨设 $k$ 为算法的迭代次数，$n$ 为每次采样的数据点的个数，$w$ 为数据集中内点的比率。则 $n$ 个点均为内点的概率为 $w^n$，$1-w^n$ 为 $n$ 个点中至少有一个外点的概率，$(1-w^n)^k$ 为算法在 $k$ 次迭代中每次都采样到了外点的概率。假设 $p$ 为算法在 $k$ 次迭代中至少有一次采样到的点均为内点，则有

$$1-p=(1-w^n)^k \tag{5.28}$$

对此式两边取对数有

$$k=\frac{\log(1-p)}{\log(1-w^n)} \tag{5.29}$$

表 5.1 为达到 99%的成功概率($p>0.99$)所需要的试验迭代次数 $k$。其中横向参数为外点在数据中所占比例 $1-w$，纵向参数为抽取数据个数 $n$。可见，随着采样点数和外点所占比例的增加，所需的迭代次数会迅速增大。这也是在采样过程中需要遵循最小采样原则的原因。在外点比例和迭代次数一定的情况下，采用较小的采样点数有助于提高成功的概率。

表 5.1 RANSAC 方法所需的迭代次数与采样点数和外点比率的关系

| 外点所占比例/% | 迭代次数 | | | | | | |
|---|---|---|---|---|---|---|---|
| | 采样点数=2 | 采样点数=3 | 采样点数=4 | 采样点数=5 | 采样点数=6 | 采样点数=7 | 采样点数=8 |
| 5 | 2 | 3 | 3 | 4 | 4 | 4 | 17 |
| 10 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 20 | 5 | 7 | 9 | 12 | 16 | 20 | 26 |
| 25 | 6 | 9 | 13 | 17 | 24 | 33 | 44 |
| 30 | 7 | 11 | 17 | 26 | 37 | 54 | 78 |
| 40 | 11 | 19 | 34 | 57 | 97 | 163 | 272 |
| 50 | 17 | 35 | 72 | 146 | 293 | 588 | 1177 |

## 5.5 连续特征跟踪

回忆本章所介绍的局部特征方法，由于其匹配过程是基于特征描述子在整个图像范围内进行的最近邻搜索，因此可以处理特征点的大幅度运动。不过，在视频中，由于相邻两帧差异不大，对应特征点的位置偏移一般都较小，因此进行全局搜索一方面计算效率较低，另一方面也会导致更多的误匹配。对相邻视频帧间的特征匹配，一般采用连续特征跟踪的方法进行处理。给定上一帧的一组特征点，特征跟踪通过在每个特征点的某个局部邻域内进行搜索，来获得特征点在当前帧中的最优匹配。

连续特征跟踪通常采用托马斯-卢卡斯-卡纳德(Tomasi-Lucas-Kanade，KLT)方法。KLT 实际上是 Tomasi 等人提出的特征检测方法和 Lucas、Kanade 所提出的特征跟踪方法的结合。特征检测用于在初始化时(第 1 帧)获取特征点，并在跟踪过程中对跟踪丢失和新出现的特征点进行补充。KLT 的特征检测方法与 Harris 角点检测类似，接下来主要介绍其特征跟踪的基本原理。

假设 $T$ 和 $I$ 分别是视频中的第 $t$ 帧和第 $t+1$ 帧。考虑像素点 $T(x,y)$，算法的目标是要计算其在 $I$ 中的对应像素 $I(x',y')$。记 $x'=x+u$，$y'=y+v$，则寻找对应像素 $(x',y')$ 可以描述为以下优化问题：

$$(u,v)=\underset{u,v}{\operatorname{argmin}} \mid I(x+u,y+v)-T(x,y) \mid^2 \tag{5.30}$$

这里假设 $T$、$I$ 都是灰度图像，所以 $T(x,y)$、$I(x',y')$ 都是标量值。注意式(5.30)利用了图像的**灰度不变假设**，即认为场景中同一点，在图像中对应像素的颜色值不随时间变化，因此 $T(x,y)=I(x',y')$。

由于单个像素点的亮度在其附近一般存在较多相似值，因此只用单个像素点求解公式(5.30)将会非常不稳定。为此，进一步假设邻近的像素具有相似的运动(相邻视频帧间差异较小)，则可以认为在以 $(x,y)$ 为中心的某个局部图像块 $\Omega$ 内，像素的偏移值都是相同的。这样可以利用 $\Omega$ 内所有像素来估计特征点对应像素的偏移值 $(u,v)$：

$$(u,v)=\underset{u,v}{\operatorname{argmin}} \sum_{(x,y)\in\Omega} \mid I(x+u,y+v)-T(x,y) \mid^2 \tag{5.31}$$

由于 $I$ 关于 $(u,v)$ 的变化一般是非线性的，因此式(5.31)是一个非线性优化问题。不过，对于较小的 $(u,v)$ 来说，可以假设像素值 $I(x+u,y+v)$ 关于 $(u,v)$ 的变化是近似线性的，因

此可以用泰勒公式进行近似：

$$I(x+\Delta u, y+\Delta v) \approx I(x,y) + \frac{\partial I}{\partial x}\Delta u + \frac{\partial I}{\partial y}\Delta v \tag{5.32}$$

其中$\frac{\partial I}{\partial x}$、$\frac{\partial I}{\partial y}$分别是 $I$ 在 $x$ 和 $y$ 方向的梯度，可以用图像梯度算子进行计算。将式(5.32)代入式(5.31)中，可以得到关于未知参数 $\Delta\boldsymbol{\Theta} = (\Delta u, \Delta v)^{\mathrm{T}}$ 的线性形式

$$\Delta\boldsymbol{\Theta} = \underset{\Delta \boldsymbol{p}}{\operatorname{argmin}} \sum_{(x,y)\in\Omega} | I(x,y) - T(x,y) + \nabla I \Delta\boldsymbol{\Theta} |^2 \tag{5.33}$$

其中$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$是图像的梯度向量。式(5.33)可以采用最小二乘法来进行求解。为了提高计算效率，可以采用以下解析形式直接计算

$$\Delta\boldsymbol{\Theta} = \left(\sum_{(x,y)\in\Omega} \nabla I^{\mathrm{T}} \nabla I\right)^{-1} \sum_{(x,y)\in\Omega} \nabla I^{\mathrm{T}} (T(x,y) - I(x,y)) \tag{5.34}$$

得到 $\Delta\boldsymbol{\Theta}$ 以后，可更新对应点的位置，并重新计算$\nabla I$，如此迭代直至 $\Delta\boldsymbol{\Theta}$ 趋近于 0。

在上述方法中，假设像素灰度值在空间和时间上都是连续变化的。这个假设在实际情况中往往不成立，尤其在时间维度上，如果图像 $T$、$I$ 之间偏移较大，则上述方法将无法收敛到正确的对应点。为了应对这种情况，可以采用由粗到精(coarse-to-fine)的方法，先在低分辨率图像上获得对应点的初值，再逐步在更高分辨率的图像上进行求精。

## 扩展阅读

对标志物进行稳定地跟踪和空间注册是增强现实系统工作的基础。在现实场景中，基于局部特征的方法需要能够处理标志物的各种运动和几何变化。本章主要讨论了特征匹配的尺度不变性和旋转不变性，针对更一般的情况，可以进一步采用仿射不变的特征检测方法。由于仿射不变的特征检测方法为每个特征点决定了一个椭圆形的区域，可以表示关联区域在不同视角下的透视变换，因此可以更好地处理透视差异较大的情况。近年来，基于学习的方法也被用于局部特征检测及特征描述子计算，并在多个数据集上取得了比传统方法更好的效果。感兴趣的读者可以参考 Y. H. Jin 等人在 2021 年发表的论文“Image Matching Across Wide Baselines: From Paper to Practice”。

在实际的增强现实应用中，标志物可能有很多，因此在进行图像匹配之前，需要首先对当前帧中出现的标志物进行识别。现有的增强现实系统通常都采用基于图像检索的方式进行标志物识别。为此，需要先将标志物模型库中的每个模板图像经过预处理，然后再表示为一个描述向量。输入图像也经过同样的处理过程后被转换成一个描述向量，之后再根据向量搜索进行匹配识别。这其中需要解决的一个关键问题是输入图像中的标志物和模型库中的模板图像在空间上是没有配准的，甚至可能有非常大的差异，并且还存在背景干扰。为解决该问题，最为经典的方法是采用 S. Zisserman 在 2003 年提出的视觉词袋(bag of visual words，BoVW)模型。视觉词袋模型来源于自然语言处理领域对文本的表示方法，主要基于一段文本中单词出现的频率(直方图)来描述文本。对于图像而言，一个单词就相当于通过局部特征方法(如 SIFT 等)计算得到的图像中一个特征点的描述向量，一幅图像包含的所有特征的描述向量就相当于一段文本。为了生成所有单词的集合，可以对模型库中所有模

板图像包含的局部特征描述进行聚类，并以每个类别的中心作为一个候选单词。视觉词袋模型对一幅图像的表示就是图像包含的所有特征点的描述向量在单词集合上的统计直方图。

## 习题

1. Harris 角点检测将图像区域分为哪几类？分别怎么判定？

2. 直观地解释：为什么 LoG 算子可以用于检测斑点？

3. 简述尺度选择理论的基本原理。

4. 常见特征检测和匹配方法实现旋转不变性的基本原理是什么？

5. SIFT 特征检测的结果，往往会有多个特征点的像素坐标完全相同，请问可能是由哪些原因造成的？

6. 什么情况需要用 RANSAC 方法进行参数估计？

7. 在实际情况下，由于外点比率未知，因此 RANSAC 所需的最大迭代次数是很难根据式(5.29)进行预估的。有什么方法可以用来减少 RANSAC 的迭代次数，从而降低计算量？

8. 改进在第 3 章实现的简易增强现实系统，使之可以用任意图案作为平面跟踪标志，实现如图 5.1 所示的效果。