**1、SIFT综述**

尺度不变特征转换(Scale-invariant feature transform或SIFT)是一种计算机视觉的算法，用来检测与描述影像中的局部性特征，它在空间尺度中寻找极值点，并提取出其位置、尺度、旋转不变量，此算法由 David Lowe在1999年所发表，2004年完善总结。其应用范围包含目标、机器人地图感知与导航、影像拼接、3D模型建立、手势识别、影像追踪和动作比对。此算法有其专利，专利拥有者为英属哥伦比亚大学。

局部影像特征的描述与侦测可以帮助辨识物体，SIFT 特征是基于物体上的一些局部外观的兴趣点而与影像的大小和旋转无关。对于光线、噪声、些微视角改变的容忍度也相当高。使用 SIFT特征描述对于部分物体遮蔽的侦测率也相当高，甚至只需要3个以上的SIFT物体特征就足以计算出位置与方位。在现今的电脑硬件速度下和小型的特征数据库条件下，辨识速度可接近即时运算。SIFT特征的信息量大，适合在海量数据库中快速准确匹配。

SIFT算法的特点有：

1. SIFT特征是图像的局部特征，其对旋转、尺度缩放、亮度变化保持不变性，对视角变化、仿射变换、噪声也保持一定程度的稳定性；

2. 独特性（Distinctiveness）好，信息量丰富，适用于在海量特征数据库中进行快速、准确的匹配；

3. 多量性，即使少数的几个物体也可以产生大量的SIFT特征向量；

4. 高速性，经优化的SIFT匹配算法甚至可以达到实时的要求；

5. 可扩展性，可以很方便的与其他形式的特征向量进行联合。

SIFT算法可以解决的问题：

目标的自身状态、场景所处的环境和成像器材的成像特性等因素影响图像配准/目标识别跟踪的性能。而SIFT算法在一定程度上可解决：

1. 目标的旋转、缩放、平移（RST）

2. 图像仿射/投影变换（视点viewpoint）

3. 光照影响（illumination）

4. 目标遮挡（occlusion）

5. 杂物场景（clutter）

6. 噪声

SIFT算法的实质是在不同的尺度空间上查找关键点(特征点)，并计算出关键点的方向。SIFT所查找到的关键点是一些十分突出，不会因光照，仿射变换和噪音等因素而变化的点，如角点、边缘点、暗区的亮点及亮区的暗点等。

**2主要步骤**

**1)、尺度空间的生成；**

**2)、检测尺度空间极值点；**

**3)、精确定位极值点；**

**4)、为每个关键点指定方向参数；**

**5)、关键点描述子的生成。**

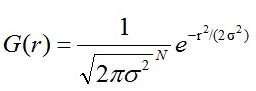
2.1高斯模糊

SIFT算法是在不同的尺度空间上查找关键点，而尺度空间的获取需要使用高斯模糊来实现，Lindeberg等人已证明高斯卷积核是实现尺度变换的唯一变换核，并且是唯一的线性核。本节先介绍高斯模糊算法。

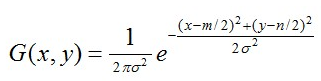
2.1二维高斯函数

高斯模糊是一种图像滤波器，它使用正态分布(高斯函数)计算模糊模板，并使用该模板与原图像做卷积运算，达到模糊图像的目的。

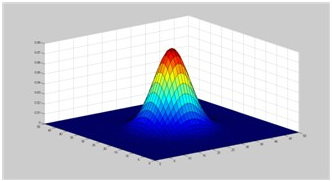
N维空间正态分布方程为：

 （1-1）

其中，是正态分布的标准差，值越大，图像越模糊(平滑)。r为模糊半径，模糊半径是指模板元素到模板中心的距离。如二维模板大小为m\*n，则模板上的元素(x,y)对应的高斯计算公式为：

 （1-2）

  在二维空间中，这个公式生成的曲面的等高线是从中心开始呈正态分布的同心圆，如图2.1所示。分布不为零的像素组成的卷积矩阵与原始图像做变换。每个像素的值都是周围相邻像素值的加权平均。原始像素的值有最大的高斯分布值，所以有最大的权重，相邻像素随着距离原始像素越来越远，其权重也越来越小。这样进行模糊处理比其它的均衡模糊滤波器更高地保留了边缘效果。



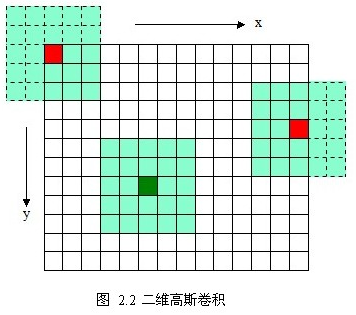
理论上来讲，图像中每点的分布都不为零，这也就是说每个像素的计算都需要包含整幅图像。在实际应用中，在计算高斯函数的离散近似时，在大概3σ距离之外的像素都可以看作不起作用，这些像素的计算也就可以忽略。通常，图像处理程序只需要计算的矩阵就可以保证相关像素影响。

2.2 图像的二维高斯模糊

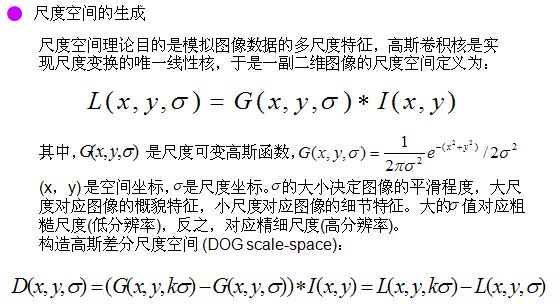
根据σ的值，计算出高斯模板矩阵的大小()，使用公式(1-2)计算高斯模板矩阵的值，与原图像做卷积，即可获得原图像的平滑(高斯模糊)图像。为了确保模板矩阵中的元素在[0,1]之间，需将模板矩阵归一化。5\*5的高斯模板如表2.1所示。

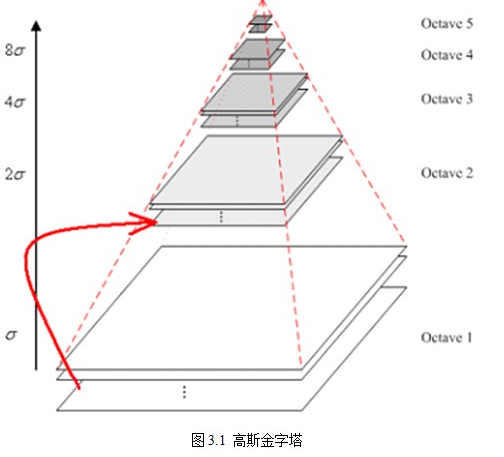


下图是5\*5的高斯模板卷积计算示意图。高斯模板是中心对称的。





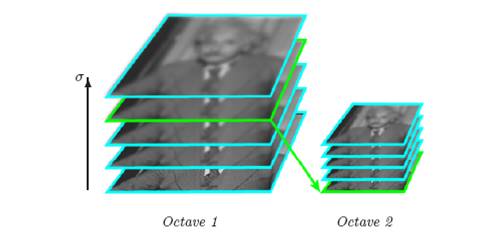


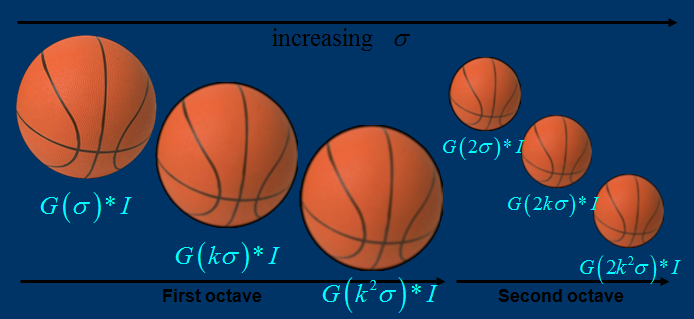


　　　　　　　　　　　　　 L(x,y,σ), σ= 1.6 a good tradeoff



　　　　　　　　　　　　   D(x,y,σ), σ= 1.6 a good tradeoff



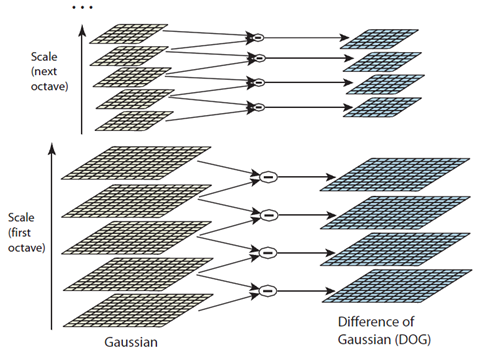


**关于尺度空间的理解说明：图中的2是必须的，尺度空间是连续的。在  Lowe的论文中 ，**

**将第0层的初始尺度定为1.6，图片的初始尺度定为0.5. 在检测极值点前对原始图像的高斯**

**平滑以致图像丢失高频信息，所以 Lowe 建议在建立尺度空间前首先对原始图像长宽扩展**

**一倍，以保留原始图像信息，增加特征点数量。尺度越大图像越模糊。**



**next octave 是由 first octave 降采样得到（如2）**

IMG_262，IMG_263

尺度空间的所有取值，s为每组层数，一般为3~5

对于给定的彩色图像，转化为灰度图像，用不同大小的σ做高斯平滑（按照 3σ 准则，高斯核矩阵的大小设为 (6σ+1)⋅(6σ+1) ，并保证行和列为奇数），再此基础上将图像降采样得到不同大小的组(octave)，每组若干图像(interval)。详细描述如下：

为了得到更多的特征点，将图像扩大为原来的两倍。假设原图像已有 σ=0.5 的高斯平滑，而我们需要第一个octave的第一张图像的 σ=1.6 ，按照定理1，我们要对扩大两倍的图像做一次高斯平滑，σ=[1.62−(0.5×2)2]1/2 。

上一个octave的图像的长度和宽度分别是下一个octave的图像的两倍。因此图像组数(octaves)可由图像大小决定，将其设为 log2(min(height,width))− 2 ，这样将使顶层octave图像的长度和宽度最小值在8像素左右。

设第m个octave的第n张图像相对于原始图像的参数σ为 sigma(m,n)则sigma(1,1)=σ0=1.6每个octave有s+1张图像（即intervals），这样得到的高斯差分金字塔(DoG)每个octave将有s张图像，我们设s为3。为了满足在不同octave间尺度的连续性，并使

sigma(m,n)= 2⋅sigma(m−1,n)，按照定理1，则：

sigma(1,n)=σ0⋅kn−1,其中k=21/s

sigma(m,n)=σ0⋅2m−1⋅kn−1

如上图所示，在第一个octave中尺度为k3⋅σ0的“最后”一张图像进行下采样得到第二个octave的第一张图像，尺度仍为k3⋅σ0=2⋅σ0。

但实际上我们需要做出更多不同尺度的高斯平滑图像，这是因为在后续高斯差分金字塔的极值检测中，需要前后两级尺度都存在图像。如图中红框所示，高斯差分金字塔中每个octave有s幅图像，则需要高斯金字塔中每个octave包含s+3幅图像。其中第s+1幅图像用作下一个octave第一幅图像的降采样。

具体实现中并未对单幅图像多次进行高斯平滑，而是由上一幅图像进行高斯平滑得到下一幅图像并迭代之，按照定理1计算σ。

---------------------

（高斯差分金字塔)

2002年Mikolajczyk在详细的实验比较中发现尺度归一化的高斯拉普拉斯函数的极大值和极小值同其它的特征提取函数，例如：梯度，Hessian或Harris角特征比较，能够产生最稳定的图像特征。

而Lindeberg早在1994年就发现高斯差分函数（Difference of Gaussian ，简称DOG算子）与尺度归一化的高斯拉普拉斯函数非常近似。其中和的关系可以从如下公式推导得到：



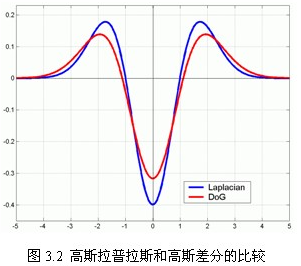
利用差分近似代替微分，则有：



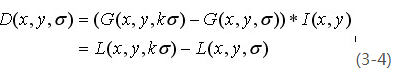
  因此有



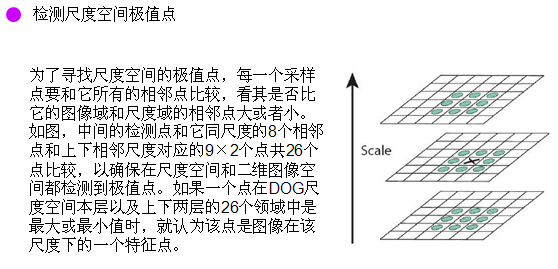
其中k-1是个常数，并不影响极值点位置的求取。



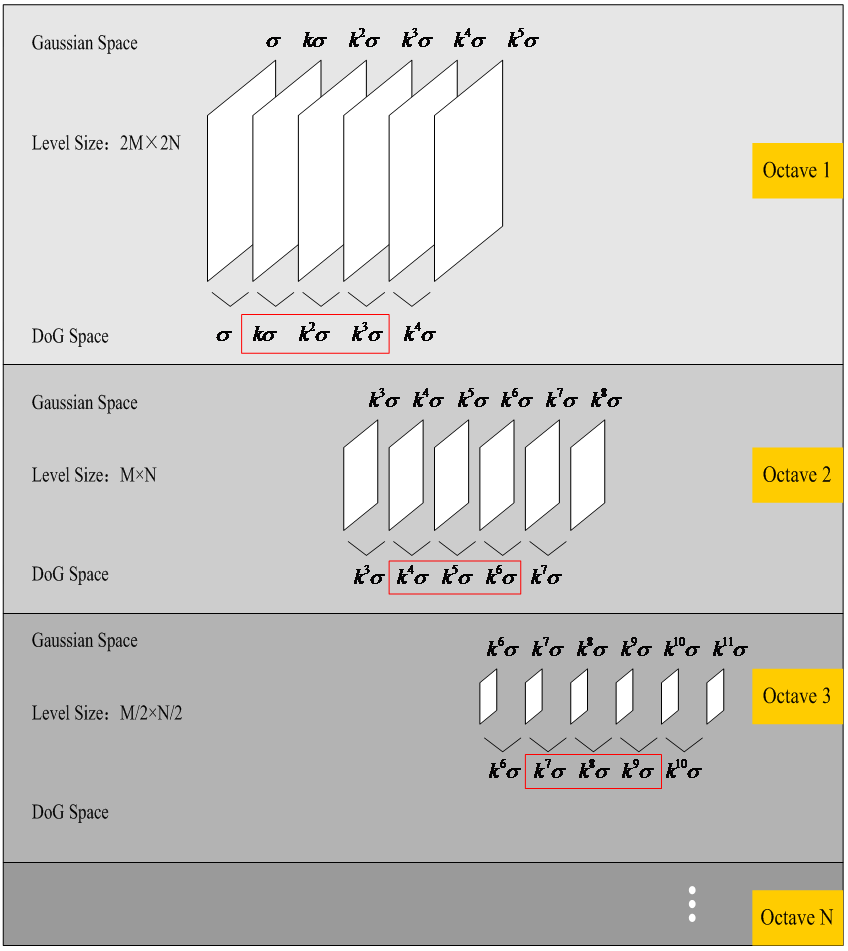
如图3.2所示，红色曲线表示的是高斯差分算子，而蓝色曲线表示的是高斯拉普拉斯算子。Lowe使用更高效的高斯差分算子代替拉普拉斯算子进行极值检测，如下：



在实际计算时，使用高斯金字塔每组中相邻上下两层图像相减，得到高斯差分图像，如图3.3所示，进行极值检测。）



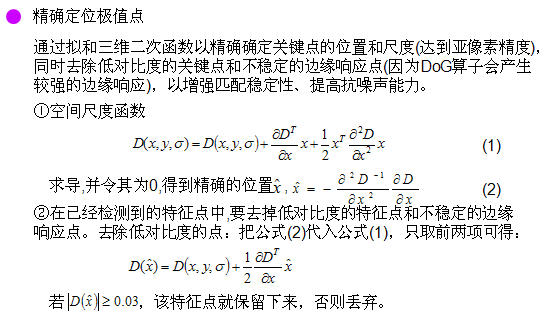
　     同一组中的**相邻尺度（由于k的取值关系，肯定是上下层）**之间进行寻找



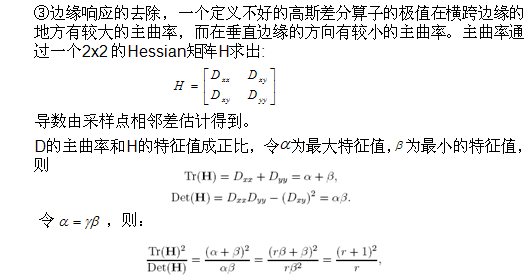
     在极值比较的过程中，每一组图像的首末两层是无法进行极值比较的，**为了满足尺度**

**变化的连续性，**我们在每一组图像的顶层继续用高斯模糊生成了 3 幅图像，高斯金字

     塔有每组S+3层图像。DOG金字塔每组有S+2层图像.



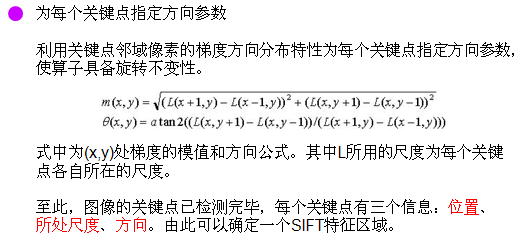
IMG_267



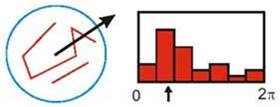
　　 　　  If ratio > (r+1)2/(r), throw it out (SIFT uses r=10)

表示DOG金字塔中某一尺度的图像x方向求导两次

**通过拟和三维二次函数以精确确定关键点的位置和尺度（达到亚像素精度）？**



**直方图中的峰值就是主方向，其他的达到最大值80%的方向可作为辅助方向**



**Identify peak and assign orientation and sum of magnitude to key point**

**The user may choose a threshold to exclude key points based on their**

**assigned sum of magnitudes.**

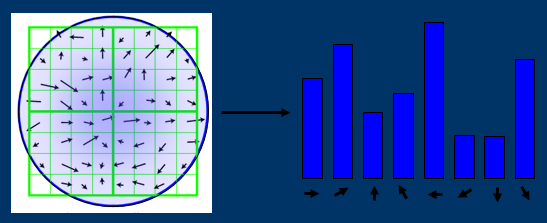
      利用关键点邻域像素的梯度方向分布特性为每个关键点指定方向参数，使算子具备

  旋转不变性。以关键点为中心的邻域窗口内采样，并用直方图统计邻域像素的梯度

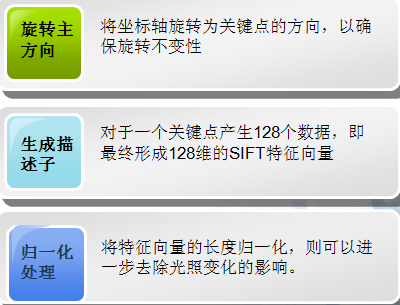
  方向。梯度直方图的范围是0～360度，其中每10度一个柱，总共36个柱。随着距

  中心点越远的领域其对直方图的贡献也响应减小.Lowe论文中还提到要使用高斯函

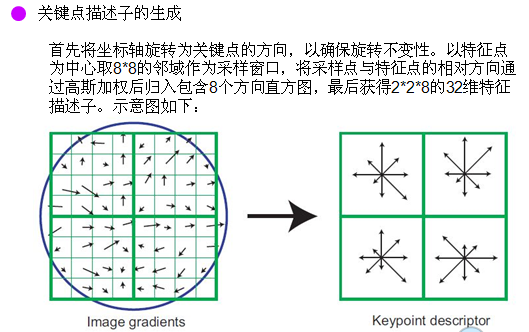
  数对直方图进行平滑，减少突变的影响。



**关键点描述子的生成步骤**

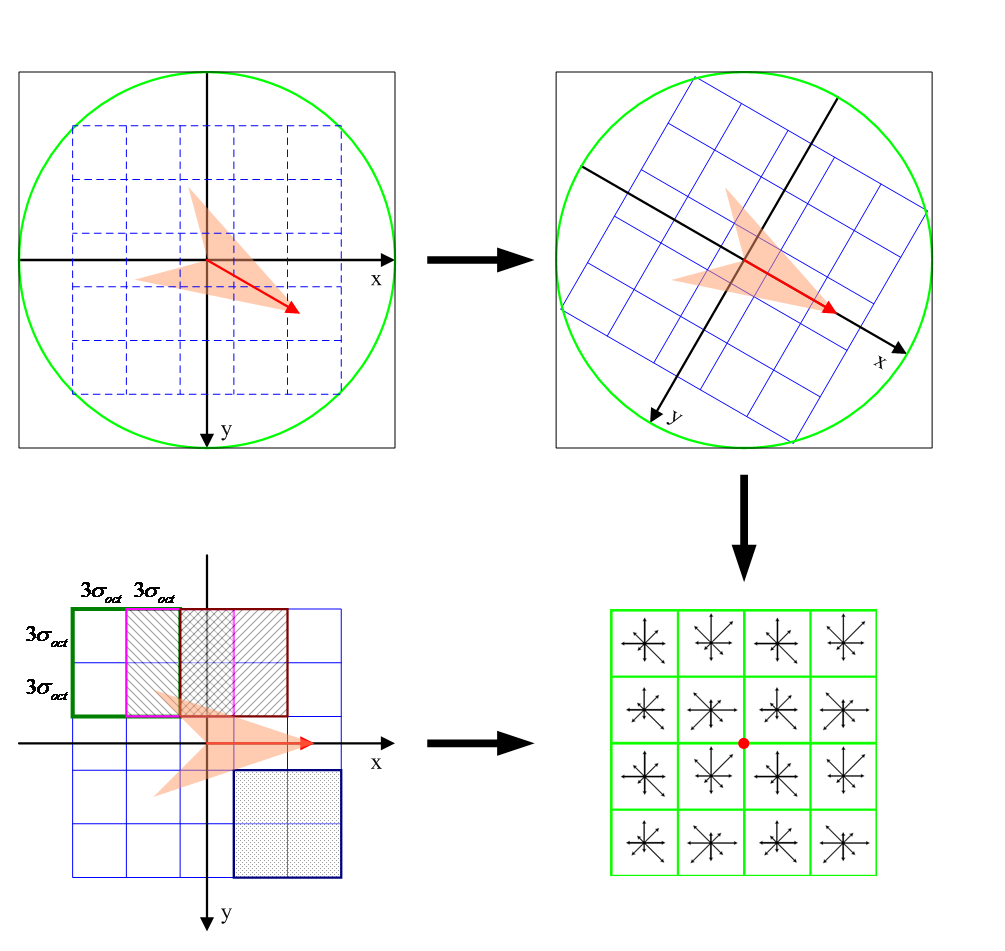


**通过对关键点周围图像区域分块，计算块内梯度直方图，生成具有独特性的向量，这个向量是该区域图像信息的一种抽象，具有唯一性。**



  每一个小格都代表了特征点邻域所在的尺度空间的一个像素 ，箭头方向代表了像素梯

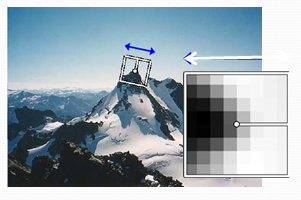
  度方向，箭头长度代表该像素的幅值。然后在4×4的窗口内计算8个方向的梯度方向直方图。**绘制每个梯度方向的累加可形成一个种子点。**

****

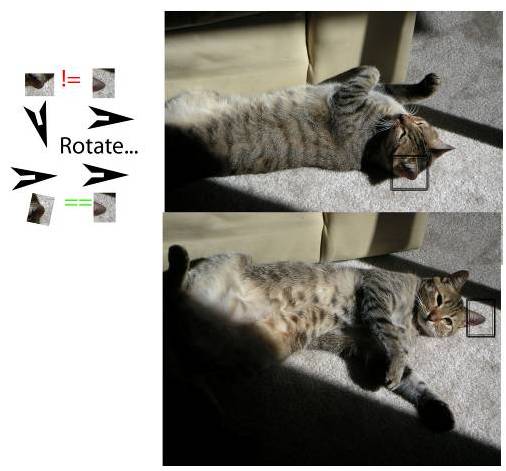
   每个直方图有8方向的梯度方向，每一个描述符包含一个位于关键点附近的四个直方图

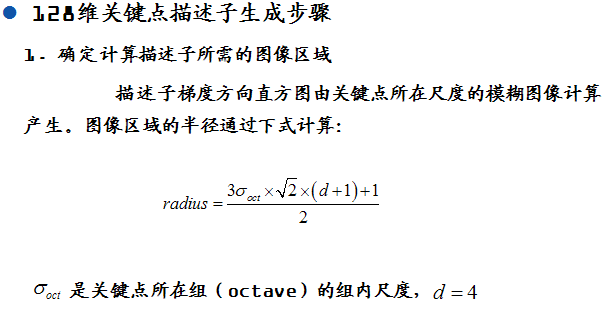
  数组.**这就导致了SIFT的特征向量有128维.**（先是一个4×4的来计算出一个直方图，

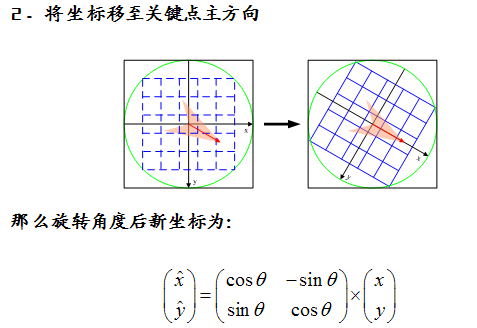
  每个直方图有8个方向。所以是4×4×8=128维）将这个**向量归一化之后，就进一步去除了光照的影响。**

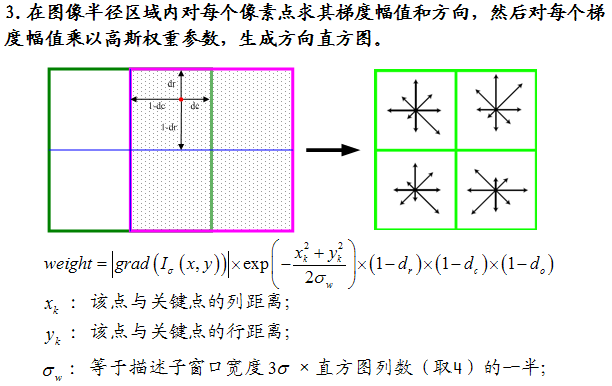


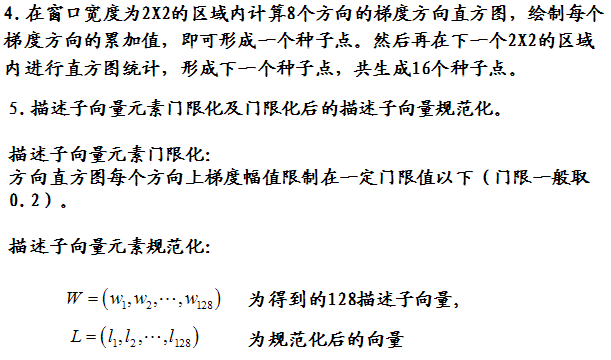
**旋转为主方向**

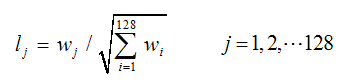












**基本概念及一些补充**

**什么是局部特征？**

　　•局部特征从总体上说是图像或在视觉领域中一些有别于其周围的地方

　　•局部特征通常是描述一块区域，使其能具有高可区分度

　　•局部特征的好坏直接会决定着后面分类、识别是否会得到一个好的结果

**局部特征需具备的特性**

　•重复性

　　•可区分性

　　•准确性

　　•数量以及效率

•不变性

**局部特征提取算法-sift**

　　•SIFT算法由D.G.Lowe 1999年提出，2004年完善总结。后来Y.Ke将其描述子部分用PCA代替直方图的方式，对其进行改进。

 　　•SIFT算法是一种提取局部特征的算法，在尺度空间寻找极值点，提取位置，尺度，旋转不变量

　　•SIFT特征是图像的局部特征，其对旋转、尺度缩放、亮度变化保持不变性，对视角变化、仿射变换、噪声也保持一定程度的稳定性。

　　•独特性好，信息量丰富，适用于在海量特征数据库中进行快速、准确的匹配。

　　•多量性，即使少数的几个物体也可以产生大量SIFT特征向量。

　　•可扩展性，可以很方便的与其他形式的特征向量进行联合。

**尺度空间理论**

　　•尺度空间理论目的是模拟图像数据的多尺度特征

　　•其基本思想是在视觉信息图像信息处理模型中引入一个被视为尺度的参数, 通过连续变化尺度参数获得不同尺度下的视觉处理信息, 然后综合这些信息以深入地挖掘图像的本质特征。

**描述子生成的细节**

　　•以极值点为中心点，并且以此点所处于的高斯尺度sigma值作为半径因子。对于远离中心点的梯度值降低对其所处区域的直方图的贡献，防止一些突变的影响。

　　•每个极值点对其进行三线性插值，这样可以把此极值点的贡献均衡的分到直方图中相邻的柱子上

**归一化处理**

　　•在求出4\*4\*8的128维特征向量后，此时SIFT特征向量已经去除了尺度变化、旋转等几何变形因素的影响。而图像的对比度变化相当于每个像素点乘上一个因子，光照变化是每个像素点加上一个值，但这些对图像归一化的梯度没有影响。因此将特征向量的长度归一化，则可以进一步去除光照变化的影响。

　　•对于一些非线性的光照变化，SIFT并不具备不变性，但由于这类变化影响的主要是梯度的幅值变化，对梯度的方向影响较小，因此作者通过限制梯度幅值的值来减少这类变化造成的影响。

**PCA-SIFT算法**

　　•PCA-SIFT与标准SIFT有相同的亚像素位置，尺度和主方向。但在第4步计算描述子的设计，采用的主成分分析的技术。

　　•下面介绍一下其特征描述子计算的部分：

　　　　•用特征点周围的41×41的像斑计算它的主元，并用PCA-SIFT将原来的2×39×39维的向量降成20维，以达到更精确的表示方式。

　　　　•它的主要步骤为，对每一个关键点：在关键点周围提取一个41×41的像斑于给定的尺度，旋转到它的主方向 ；计算39×39水平和垂直的梯度，形成一个大小为3042的矢量；用预先计算好的投影矩阵n×3042与此矢量相乘；这样生成一个大小为n的PCA-SIFT描述子。