[SIFT论文中文版](http://www.cnblogs.com/cuteshongshong/archive/2012/05/25/2506374.html)：

SIFT的主要步骤：

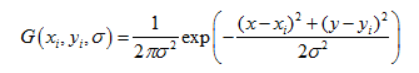
1. 尺度空间极值检测

就是利用高斯模糊获取尺度空间

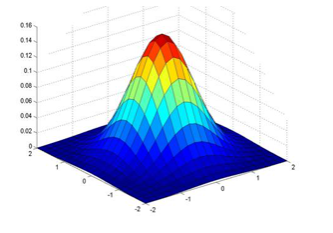
1. 关键点定位
2. 方向确定
3. 关键点描述

**尺度空间极值检测**

二维空间高斯函数：



图像：

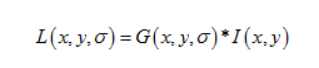


卷积过程中，以每个像素点为中心都会进行一次加权平均，从而造成‘模糊’

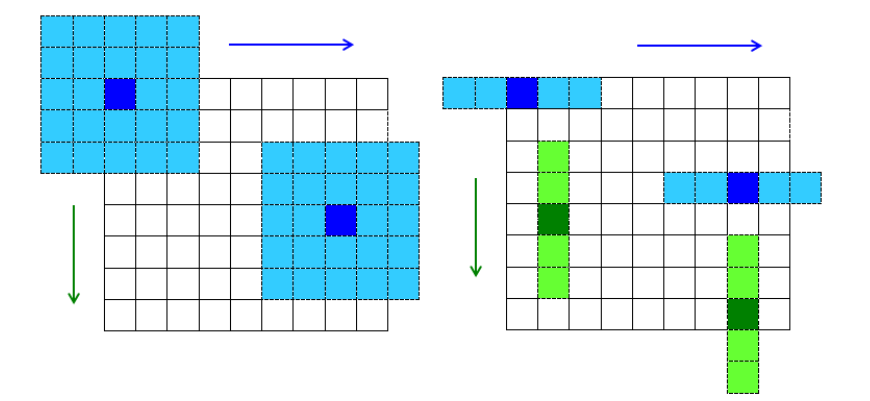
[‘高斯卷积核是实现尺度变换的唯一变换核，并且是唯一的线性核’](http://www.tandfonline.com/doi/abs/10.1080/757582976)

关于高斯变换的尺度空间：

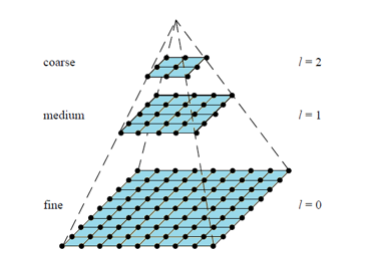
1. 尺度空间中各尺度图像的模糊程度逐渐变大，能够模拟人在距离目标由近到远时目标在视网膜上的形成过程。尺度越大图像越模糊。
2. 一个图像的尺度空间L(x,y,σ) ,定义为原始图像I(x,y)与一个可变尺度的2维高斯函数G(x,y,σ)卷积运算。



1. 通常计算会使用3σ原则，只计算**(6σ+1)\*(6σ+1)**就可以保证相关像素影响。
2. **线性可分：**使用二维矩阵变换的高斯模糊可以通过在水平和竖直方向各进行一维高斯矩阵变换相加得到。



金字塔表达：



* **尺度空间表达是由不同高斯核平滑卷积得到，在所有尺度上有相同的分辨率；**
* **而金字塔多分辨率表达每层分辨率减少固定比率。**

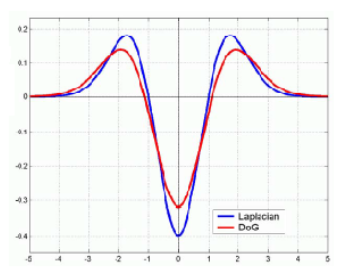
DoG（Difference of Gaussian）：理解成函数近似（像数值计算中的逼近）

1. LoG:结合尺度空间表达和金字塔多分辨率表达，就是在使用尺度空间时使用金字塔表示，拉普拉斯金字塔（[《The Laplacian pyramid as a compact image code》](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1095851&tag=1" \t "_blank)）。

LoG算子：

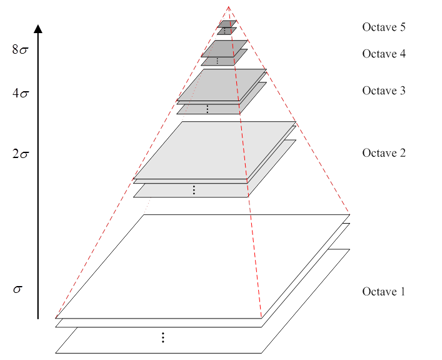


1. DoG：高斯差分金字塔，算是对LoG的近似（在SIFT中：某一尺度上的特征检测可以通过对两个相邻高斯尺度空间的图像相减）



SIFT中，**通过构建高斯金字塔进行DoG计算：**

1. 金字塔每层多张图像合称为一组（Octave），每组有多张（也叫层Interval）图像。另外，降采样时，金字塔上边一组图像的第一张图像（最底层的一张）是由前一组（金字塔下面一组）图像的倒数第三张隔点采样得到。



原图像为金子塔的第一层，每次降采样所得到的新图像为金字塔的一层(每层一张图像)，每个金字塔共n层。金字塔的层数根据图像的原始大小和塔顶图像的大小共同决定，其计算公式如下：（其中M，N为原图像的大小,t为塔顶图像的最小维数的对数值）

* 1. 三个参数：

IMG_256—尺度空间坐标

     O—组(octave)数

    S— 组内层数

* 1. 具体计算：



组数：



高斯模糊系数（组内）：



既：



相邻组的尺度关系：

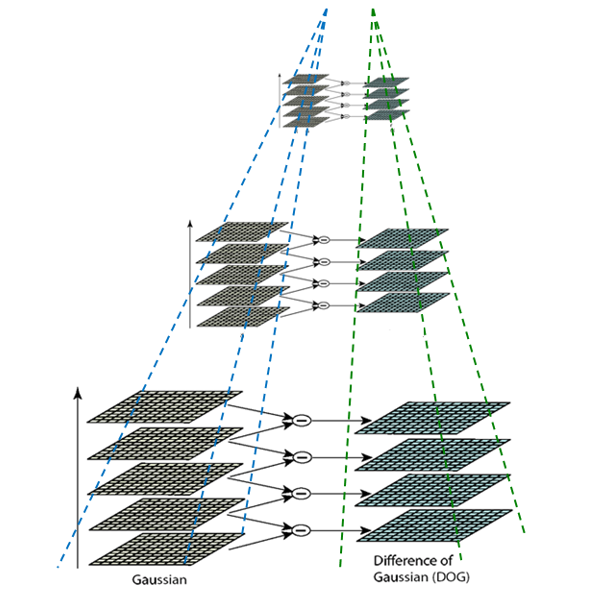
其中小s为组数

所以，**相邻两组的同一层尺度为2倍的关系**。

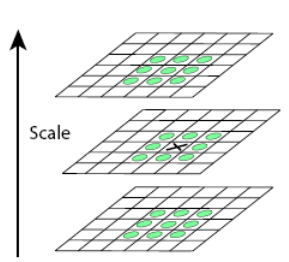
最终尺度序列总结为：

http://img.my.csdn.net/uploads/201210/14/1350191002_1514.png

1. 构建：



1. 至此，可以利用得到的DoG进行极值点的大致定位（注意图像的尺度变换等以及图像本身的离散特性使得极值点的定位只是大致定位）
2. 每一个像素点都要和邻近的所有点比较：



即和它同尺度的8个相邻点和上下相邻尺度对应的9×2个点共26个点比较

1. [具体算法](https://blog.csdn.net/jinshengtao/article/details/50119817)：

(1)遍历不同分辨率和尺度高斯差分金字塔，获取特定分辨率和尺度下的高斯差分图像

(2)同组同层比较，对Io,s内的每个像素Pcenter，首先比较Pcenter与阈值PRE\_COLOR\_THRES，若不满足条件，则返回false，即该点不是极值点，转到(4)；否则，将Pcenter与其周围8邻域像素点比较，只要该邻域内存在相对于Pcenter的最大和最小值像素点，则认为Pcenter是该层的极值点，返回true，转到(3)

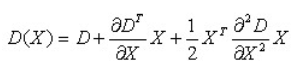
(3)同组不同层比较，在Io,s上下两层高斯差分图像Io,s-1和Io,s+1中，分别将Pcenter同其对应的上下层邻域进行比较，如果在上下两层邻域内均存在相对于Pcenter的最大最小值，则认为Pcenter是相邻层的极值点，返回true，否则就不是极值点，返回false，转到(4)

(4)对于返回值为true的，将Pcenter加入极值点坐标集合ret中

1. 说明：
   * 1. 为了在每组中检测S个尺度的极值点，则DOG金字塔每组需S+2层图像，而DOG金字塔由高斯金字塔相邻两层相减得到，则高斯金字塔每组需S+3层图像，实际计算时S在3到5之间。推荐取值正常是3
     2. 这样产生的极值点并不全都是稳定的特征点，因为某些极值点响应较弱，而且DOG算子会产生较强的边缘响应

**关键点检测定位与筛选（注意之前已获取大致点位置）**

1. 方式是直接利用函数拟合得到更为精确的极值点
2. 去除低对比度的点：
3. 不友好的二维泰勒公式-->直接看结论



其中：



求泰勒展开的极值点：

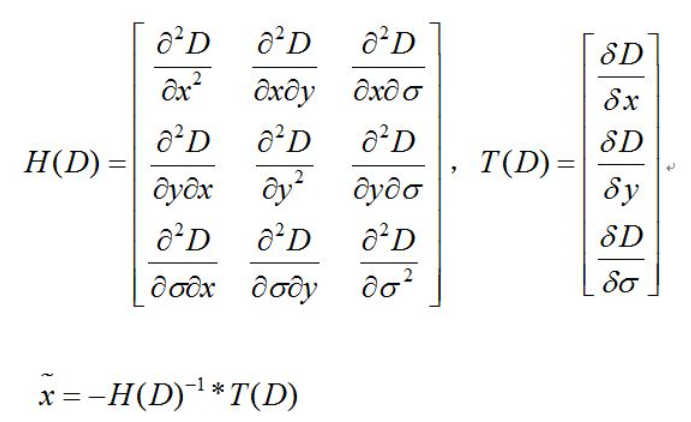


SIFT作者认为如果偏移量中任何一个分量大于0.5，则这个极值点和另一个采样点（图像中的另一个像素）离得更近，应当寻找邻近点，原文设置的寻找深度为5。

代入原式，化简，有：



以上的计算，二阶导数用Hessian矩阵，一阶导数用梯度，它们均通过相邻像素的差值计算。这里的Hessian矩阵是由三元函数的二阶偏导数构成的方阵，描述了DOG函数的局部曲率，引入T(D) (过程???)：



1. 算法流程：

遍历每个极值点，计算新的插值坐标偏移量iter\_offset。若不满足要求，则寻找（寻找方式）近邻点，重新比较，若满足OFFSET\_THRES阈值要求，则更新极值坐标，并将新坐标代入D(x\_hat)，求得的结果dx\_hat与阈值CONTRAST\_THRES比较，若满足要求，则更新极值点的真实坐标sp->real\_coor

1. 若：

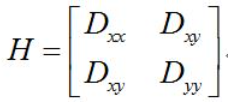


该特征点就保留下来，否则丢弃。(Lowe论文中使用0.03，Rob Hess等人实现时使用0.04/S），另外还有一个规则：‘当它在任一维度上的偏移量大于0.5时（即x或y或IMG_256），意味着插值中心已经偏移到它的邻近点上，所以必须改变当前关键点的位置’，系前面的off\_set

1. 去除边缘响应点
   1. DoG本身在图像边缘有较强响应：

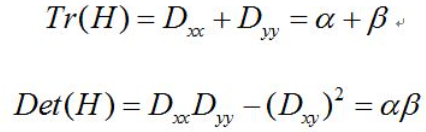
曲面上每个点（非平点）都有两个主方向，并且沿这两个主方向的法曲率（即两个主曲率）分别是曲面在该点法曲率的最大值和最小值。对于边缘上的点，沿垂直于边缘的方向上，法曲率最大，而沿边缘的方向上，法曲率最小。因此对于分布在边缘上附近的极值点，它们的法曲率最大值和最小值之比（即两个主曲率之比），一般情况下要比非边缘点的比值大。根据这种思想，我们可以设一个比值的阈值，当比值大于这个阈值就认为极值点在边缘上。

利用Hessian矩阵：

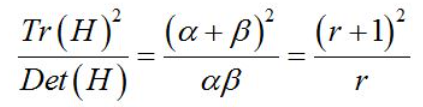


这里的微分同样利用差分近似计算

又因为DOG的主曲率和H的特征值成正比，我们只需要计算 H 的较大特征值与较小特征值的比例即可。设α 是较大的特征值， β 是较小的特征值，则：



通常这里的行列式不会是负值，如果出现负值的情况，即两个主曲率不同号，我们将丢弃这个点，不将其视为极值点。设 r=α/ β ，我们可得：



这里当 r≥1，(r+1)2/r是r的单调递增函数，因此要计算主曲率的比值（即 r）是否在某阈值之下，为了避免求H的特征值，只需要判断上式左边的项是否在阈值之下即可，通常r取10。

* 1. 具体算法：

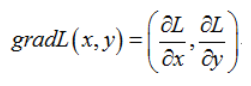
对于每个候选特征点，计算其Hessian矩阵，然后计算该矩阵的迹和行列式的比值，如果该比值大于阈值则保留该候选关键点，否则剔除。

**方向确定**

1. 原理：
2. 旋转不变性:

实现旋转不变的基本思想是采用“相对”的概念。利用关键点邻域像素的梯度方向分布特性，我们可以为每个关键点指定方向参数方向。而后面定义的关键点描述特征符是相对于这个主方向的，因而可以实现匹配时图像的旋转无关性。我们通过梯度直方图统计法来确定关键点的方向，即统计以关键点为原点，利用所有在此区域内的像素点的梯度形成一个方向直方图。

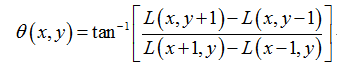
具体说来，对于每个特征点P(x,y,δ,σ)所处高斯图像金字塔的δ组σ层图像，其像素点L(x,y)的梯度：(这里的图像是高斯模糊后的图像，非高斯差分图像)



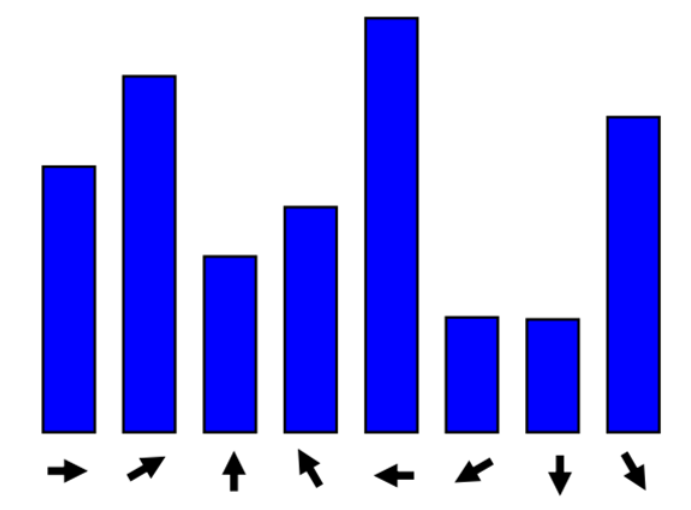
梯度的幅值：

IMG_256

梯度的方向：



  直方图的横坐标是梯度方向，共 36 项，每项代表了 10 度的范围；纵坐标是梯度大小，对于归到横坐标上任一项内所有的点，将其梯度大小相加，其和作为纵坐标。下图应该有36项，为了示意只画了8项。(根据Lowe的建议，直方图统计半径采用3\*1.5\*σ)



从直方图中选出纵坐标值最大的一项所在方向作为该关键点的主方向。如果另一个相当于主峰值 80%能量的峰值时，也将其作为该关键点的方向，称为辅方向。特征点有多个方向的情况下，实际上是在此位置上有多个关键点，他们的方向不同。这些方向可以增强匹配的鲁棒性，Lowe的论文指出大概有15%关键点具有多方向，但这些点对匹配的稳定性至为关键。

另外，为了获得更好的稳定性，可以对关键点邻域的梯度大小进行高斯加权。每相邻三个bin采用高斯加权，根据Lowe的建议，模板采用[0.25,0.5,0.25]，并连续加权两次。

1. 具体算法：
2. 遍历特征点集合points，搜索每个特征点的邻域，半径为rad，生成含有36柱的方向直方图，梯度直方图范围0~360度，其中每10度一个柱。
3. 利用高斯加权对方向直方图进行两次平滑，增加稳定性
4. 通过峰值比较，求取关键点方向（可能是多个方向）；
5. 通过Taylor展开式对上述峰值进行二次曲线拟合，计算关键点精确方向，即重新计算峰值所在bin的值；
6. 根据bin的值还原角度，作为特征点的方向。

每个关键点有三个信息：位置、尺度、方向

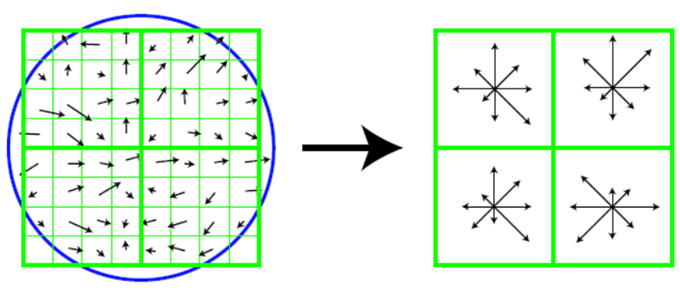
**特征描述符计算**

1. 描述符构成：

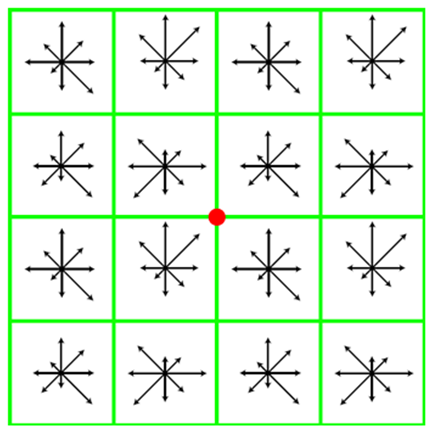
我们需要一组向量将这个关键点表达出来，并且这组向量不单包括关键点，还应包括关键点周围对其有贡献的像素点。我们还期望这组向量对仿射变换、光照变换等具有一定的鲁棒性，这些不变特性将会作为目标匹配的依据。

描述子的基本思路：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。

如下图所示，左侧图片为在关键点周围取一个8\*8邻域，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。



右侧图片为四分之一个邻域，由4个大小2\*2的像素区域组成，每个子区域生成一个八方向的梯度直方图，绘制每个梯度方向的累加可形成一个种子点。这样一个特征点由4\*4个种子点的信息所组成，如下图所示：



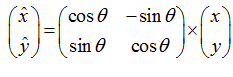
即每个子区域生成一个描述子，一个描述子中涉及 8 个方向。所以每个关键点有 4x4x8 = 128 维。

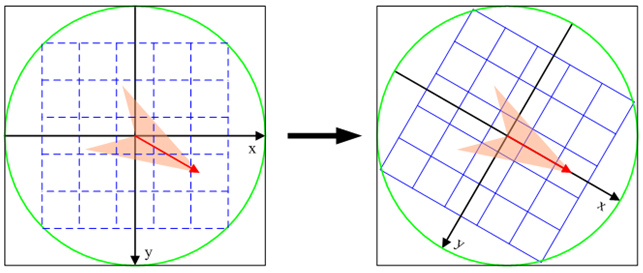
1. 计算过程：
   * 1. 描述符计算：[Link](https://blog.csdn.net/maweifei/article/details/53932782)（方式不一样）
2. 确定计算描述子所需的图像区域，图像区域的半径通过下式计算

IMG_256

其中，σoct是关键点所在组(octave)的组内尺度，d=4。

1. 将坐标移至关键点主方向，旋转角度后新坐标：





1. 在图像半径区域内对每个像素点求其梯度幅值和方向，后对每个梯度幅值乘以高斯权重参数，生成方向直方图。描述子梯度方向直方图由关键点所在尺度的模糊图像计算产生。方向直方图的计算如下：

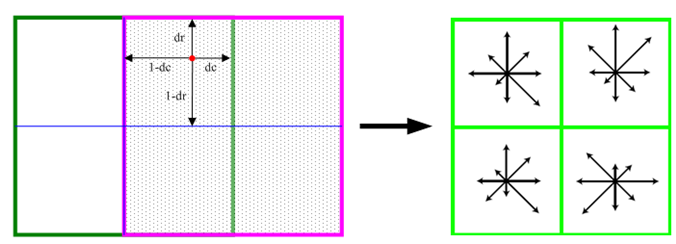
IMG_259

xk：该点与关键点的列距离

yk：该点与关键点的行距离

σw：等于描述子窗口宽度3σ\*直方图列数（取4）的一半

weight：表示方向直方图某个bin的数值

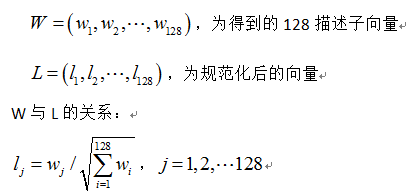


1. 在窗口宽度为2X2的区域内计算8个方向的梯度方向直方图，绘制每个梯度方向的累加值，即可形成一个种子点。然后再在下一个2X2的区域内进行直方图统计，形成下一个种子点，共生成16个种子点。
2. 描述子向量元素门限化及门限化后的描述子向量规范化。

描述子向量元素门限化是指方向直方图每个方向上梯度幅值限制在一定门限值以下（门限一般取0.2）。

为了对光线变化更具鲁棒性， 描述子都被归一化到单位长度。如果图像的对比度发生变化，每个像素值都会乘上一个数值，归一化后，对比度的影响被消除了。对于图像亮度的变化，每个像素值都会加上一个数值，然而这对计算的梯度是没有影响的。因此，该描述子对亮度的仿射变换是鲁棒的。对于非线性的光线变化，梯度大小会受影响，但是梯度的方向不会有大的变化，我们可以根据前面描述的生物学原理解决此问题。在 128 维的单位向量中，滤除梯度大小大于0.2的梯度值，然后重新归一化。也就是说，梯度大小的作用被虚弱了，而方向信息的作用被强化了。0.2是实验得出的经验值。

描述子向量元素规范化，即归一化。我们设：



至此，我们就得到了sift特征的128维向量…