# 图像检索

## 图像搜索的一般步骤

对于这种图像搜索的算法，一般是三个步骤：

1. **将目标图片进行特征提取**，描述图像的算法很多，用的比较多的是：**SIFT描述子**，**指纹算法函数**，**bundling features算法**，**hash function（散列函数）**等。也可以根据不同的图像，设计不同的算法，比如**图像局部N阶矩**的方法提取图像特征。

2. **将图像特征信息进行编码**，并**将海量图像编码做查找表**。对于目标图像，可以对分辨率较大的图像进行降采样，减少运算量后在进行图像特征提取和编码处理。

3. **相似度匹配运算**：利用目标图像的编码值，在图像搜索引擎中的图像数据库进行全局或局部的**相似度计算**；根据所需要的鲁棒性，设定阈值，然后将相似度高的图片预保留下来；最后应该还有**进一步筛选最佳匹配图片**，这个应该还是用到**特征检测算法**。

https://www.zhihu.com/question/38857926

在前dl时代，大家都是手工设计feature，比如object retrieval就是sift特征，sketch就是轮廓特征，person就是color，face就是LBP，clothing可能就是纹理特征，诸如此类。**每种任务都会有特定的feature**，不存在说太多的通用检索情况。到了dl时代，大家发现cnn特征对于任何图片都是有效的，这才开始了所谓的一个cnn搞定所有通用检索的情况。

但是到了cnn时代，大家更关注的自然就是各个领域真正的领域信息了，开始考虑工业可用性了。比如person reid，或者face recognition，最关注的就是如何减少光照和姿态的影响，包括特征点寻找和局部对齐。object retrieval则是如何找到更好的patch，clothing retrieval则开始关注的cross domain问题。

如果说前dl时代，大家关注如何提高性能，那么dl时代，大家关注的就是如何真正的做到工业可用性，这是完全不一样的思路。而且像图文互搜这种前dl时代很难解决的问题，现在也开始有了比较好的性能提升了，这都是有迹可循的。

总的来说，cbir正在兴起，里面还有很多机制可以探讨，就说softmax这种简单的分类手段来训练并抽取特征，用相同的网络，不同的人甚至能有20个点以上的性能差距，这都是值得大家去好好探讨原理的。有多东西，不是说靠trick就行的，要想整个领域真正发展，我们更加需要去探讨其中的机理，而非当做黑盒抽取特征。

## 基于内容的图片检索CBIR

传统的图像检索过程，先**通过人工对图像进行文字标注**，再**利用关键字来检索图像**，这种依据图像描述的字符匹配程度提供检索结果的方法，简称“以字找图”，既耗时又主观多义。

基于内容的图像检索克服“以字找图”方式的不足，**直接从待查找的图像视觉特征出发**，在图像库（查找范围）中找出与之相似的图像，这种**依据视觉相似程度给出图像检索结果**的方式，简称“以图找图”。

### 三个层次

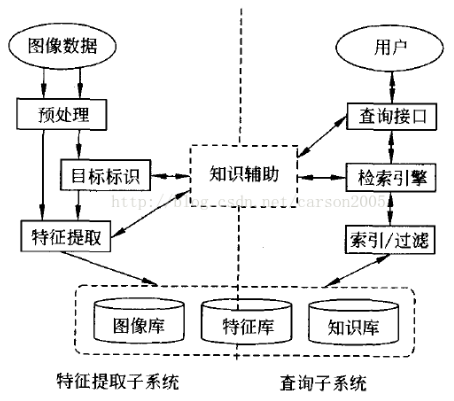
基于内容的图像检索分为三个层次：（1）依据提取图像本身的**颜色、形状、纹理等低层特征**进行检索；（2）基于图像的低层特征，通过识别图像中的对象类别以及对象之间的**空间拓扑关系**进行检索；（3）基于**图像抽象属性**（场景语义、行为语义、情感语义等）的**推理学习**进行检索；

### 四个热点

基于内容的图像检索技术研究的热点可以分为4个方面：最初的图像检索研究主要集中在如何**选择合适的全局特征去描述图像内容**和**采用什么样的相似性度量方法进行图像匹配**。第二个研究热点是**基于区域的图像检索方法**，其主要思想是用图像分割技术提取出图像中的物体，然后**对每个区域使用局部特征来描述**，**综合每个区域特征可以得到图像的特征描述**。这两个研究方向都是以图像为中心，对于需求缺乏分析。第三个研究热点就是针对这一问题而展开的，借助相关的反馈的思想，**根据用户的需求及时调整系统检索时用的特征和相似性度量方法**，从而缩小低层特征和高层语义之间的差距。第四个研究热点是研究如何**从多种渠道获取图像语义信息**，如何**将图像底层特征与图像关键词结合进行图像自动标注**以提高检索准确率等。

### 体系结构

基于内容图像检索的体系结构划分为**特征提取**和**查询**两个子系统。



**预处理**：包括图像格式的转换、规格化、图像的增强和去噪等。

**目标标识**：标识出图像中**用户感兴趣的区域或对象**，以便**针对目标进行特征提取并查询**。

**特征提取**：包括提取图像颜色、纹理、形状、空间位置关系等特征。

**数据库**：由**图像库**、**特征库**和**知识库**组成。图像库为数字化的图像信息；特征库包含**自动提取的内容特征**；知识库包含专门和通用知识，**有利于查询优化和快速匹配**，知识库中的知识表达可以更换以**适用于各种不同的应用领域**。

**检索引擎**：包括一个有效可靠的**相似性测度函数集**，通过索引、过滤模块达到快速检索的目的。

### 关键点

基于特征的图像检索有三个关键：

1）恰当的**选取图像特征**；

2）采取有效的**特征提取方法**；

3）准确的**特征匹配算法**；

利用各种特征对图像检索已经取得了相当的发展，大量的检索实验可以表明，**综合特征检索**要比单一特征检索更符合人类的视觉感受要求，检索效果会更好，但如何去找到**合适的权值将多个特征组合起来**是非常困难的。

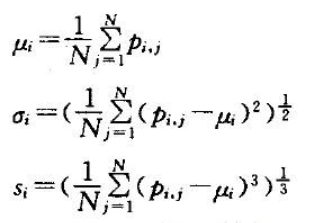
### 常用特征

目前在CBIR中最常用的特征一般有：**颜色特征**、**形状特征**和**纹理特征**。

#### 颜色特征

**颜色**是图像检索中最先被采用的特征，主要方法有：1）颜色直方图、2）颜色一致性矢量（color coherence vectors）、3）颜色相关图、4）颜色矩。

**颜色矩**是一种简单而有有效的颜色表示，主要采用图像中各颜色的均值和方差作比较，处理简单，可以用它**作为图像检索的初检**，为下一步的细检缩小搜索范围。它的数学基础是：**任何图像的颜色分布都可以通过其各阶矩来表示**。然而，颜色分布的大部分信息都集中在它的低阶矩上，所以可以用颜色的**一阶矩（均值）、二阶矩（方差）和三阶矩（偏度）**来**近似估计图像的总体颜色分布**。图像的颜色矩一共只需要9个分量（3个颜色分量，每个分量上3个低阶矩），**与其他的颜色特征相比是非常简洁的**。在实际应用中为避免低次矩较弱的分辨能力，**颜色矩常和其它特征结合使用**，而且一般在使用其它特征前起到**过滤缩小范围**（narrow down）的作用。



**颜色一致性矢量**：当图像中颜色相似的像素所占据的连续区域的面积大于一定的阈值时，该区域中的像素为**聚合像素**，否则为**非聚合像素**。

**颜色相关图**：强调同一颜色在图像中的空间距离相关性，但计算量比较大。

**颜色直方图**：直方图的横轴表示亮度，从左到右亮度依次提升；纵轴表示像素数量，从下到上像素依次增多。若一直方图的分布左偏，说明该照片亮度整体偏低。

#### 形状特征

形状是物体的一个重要特征，但由于物体形状的自动获取很困难，基于形状的检索一般仅限于非常容易识别的物体。形状可以用面积、离心率、圆形度、形状度、曲率、分形维等全局和局部特征来表示。

#### 纹理特征

纹理是图像中一种重要而又难以描述的特征，航空、遥感图片、织物图案、复杂的自然风景以及动物植物等都含有纹理。通常来讲，把图像中局部不规则，而宏观有规律的特性称之为纹理。以纹理特性为主导的图像称之为**纹理图像**，以纹理特征为主导的区域称之为**纹理区域**。纹理是图像的一个重要特征，一般认为纹理就是纹理元素有规律的排列组合，而把具有重复性、形状简单、强度一致的区域看做纹理元素。视觉纹理特征主要有：粗糙度、对比度、方向度、线象度、规整度、粗略度等。

# SIFT特征

https://www.cnblogs.com/wangguchangqing/p/4853263.html

https://blog.csdn.net/lhanchao/article/details/52345845

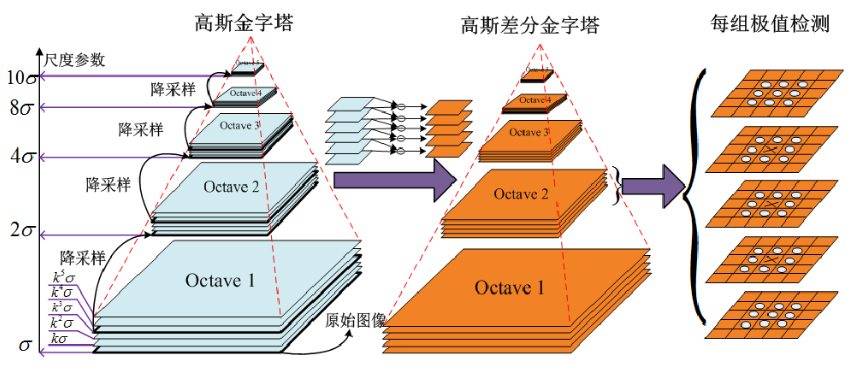
https://blog.csdn.net/sulanqing/article/details/16370879

https://www.shiyanlou.com/courses/1011

Scale Invariant Feature Transform，尺度不变特征变换。SIFT特征对旋转、尺度缩放、亮度变化等保持不变性，是一种稳定的局部特征。

SIFT算子是**把图像中检测到的特征点用一个128维的特征向量进行描述**，因此一幅图像经过SIFT算法后表示为一个**128维的特征向量集**，该特征向量集具有对图像缩放，平移，旋转不变的特征，对于光照、仿射和投影变换也有一定的不变性，是一种非常优秀的局部特征描述算法。

## SIFT特征检测的步骤



SIFT特征不只具有尺度不变性，即使改变旋转角度，图像亮度或拍摄视角，仍然能够得到好的检测效果。

1）**构建尺度空间**：这是一个初始化操作，通过**生成尺度空间**来创建原始图像的多层表示以保证尺度不变性。

2）**LoG近似**：使用Laplacian of Gaussian能够很好地找到找到图像中的兴趣点，但是需要大量的计算量，所以使用创建更容易的表示来近似它。

3）**找到特征点**：利用很快的近似，我们可以找到特征点，它们是Difference of Gaussian图像的极大极小值。

4）**除去不好的特征点**：边界和低亮度区域是不好的特征点，除去它们以使得算法有效和鲁棒，在这里使用近似Harris Corner检测器。

5）**给特征点赋值一个方向**：为每个特征点计算一个方向，依照这个方向做进一步的计算，这个操作有效地取消了方向的影响，使得算法具有旋转不变性。

6）**生成SIFT特征**：利用位置上的尺度和旋转不变性，能够生成一个表示，它能帮助唯一地识别特征，通过这个表示，可以很容易识别寻找的特征。

### 构建尺度空间

https://blog.csdn.net/ro9er/article/details/7404425

https://blog.csdn.net/xiaowei\_cqu/article/details/8067881

https://blog.csdn.net/sulanqing/article/details/16370837

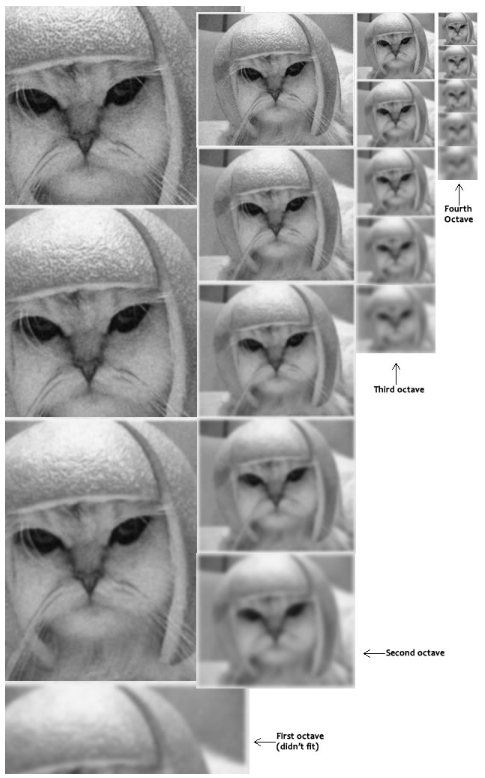
现实世界的对象只会在某些尺度上有意义。比如面前一棵树，如果在大的尺度上，树是有意义的，从另一个角度来说，树叶的信息被抛弃了。尺度空间就是从数字图像的角度模拟这些概念。如果要去除一些详细的信息，你必须确保你没有引入新的错误的细节，一个较好的做法是**运用高斯核来进行模糊处理**：

其中，是尺度参数，可以看作是模糊处理规模的大小，值越大，模糊的越多。

为了创建一个尺度空间，我们可以利用原始的图像来逐渐生成模糊后的图像，下图显示了图像是如何摆脱细节的，如猫的胡须。

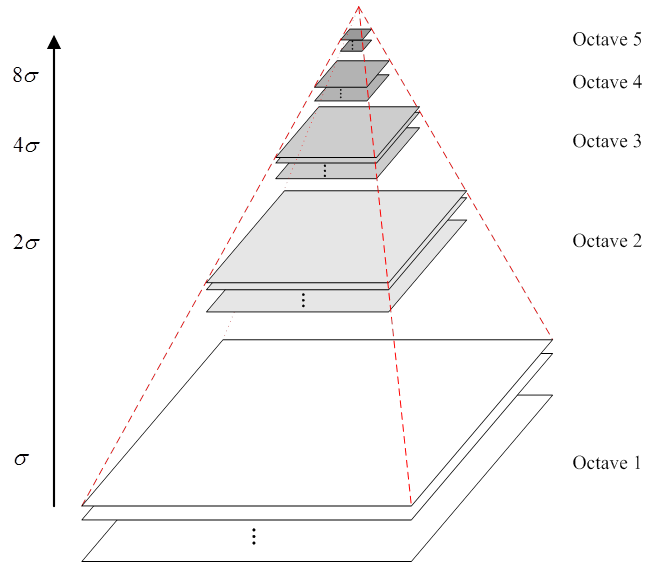


SIFT将尺度空间带入下一个的等级，你拿到原始图像，持续产生模糊图像，然后你将原始图像尺度缩小一半，再次产生模糊图像，持续如此。统一大小的图像（纵向）属于一个octave，下图有4个octave，每一个octave有5幅图像。每幅图都是在增长的尺度（模糊的规模）上建立的。



每个图像的模糊量是很重要的，假设一幅图像的模糊量是σ，那么下一幅图像的模糊量是k\*σ，这里的k是一个常量。在同一组中，下一幅图像比前一幅图像在σ上相差sqrt(2)，即 。

构造高斯金字塔：



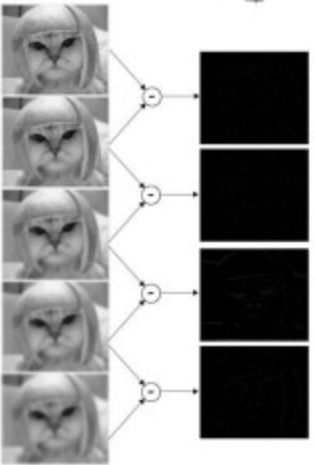
### LoG近似DoG

在上一步，我们通过逐渐模糊图像，创建了图像的尺度空间。

现在使用这些模糊的图像来生成另一类图像组Differece of Gaussain(DoG)，这些DoG图像很适合在图像中找感兴趣的特征点。

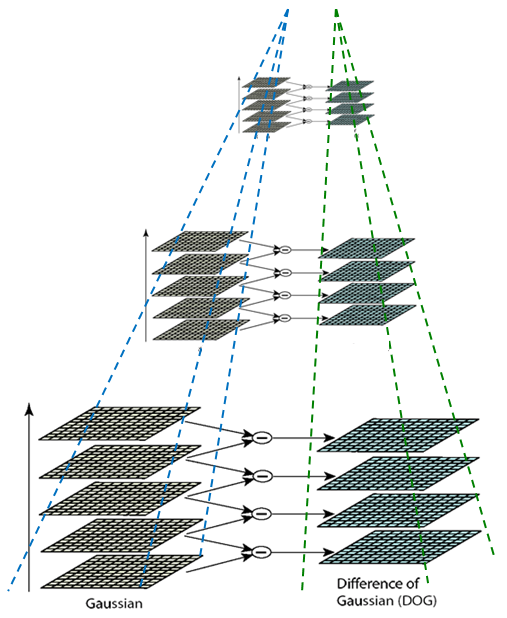
Laplacian of Gaussian操作按下面进行：取一张图像将其模糊一点，并且计算它的二阶导数，这样可以定位图像中的边界和直角。这些边界和直角对于找到特征点是很有帮助的。但是二阶导数对噪音是非常敏感的，通过使用模糊操作平滑噪音，可以稳定二阶导数。这个问题计算所有二阶导数需要大量的计算，所以使用近似操作。

Difference of Gaussian图像近似等于Laplacian of Gaussian，并且我们用一个简单的**减法**来替换计算量大的Laplacian操作，这些DoG图像是尺度不变的：



在这一组图像中，对每两幅尺度相邻的图像做减法操作，就可以生成相应的DoG图像。每组图像中两个连续图像被选择，并用一幅图像减去另一幅图像，然后下一对图像也被选择，这个过程重复进行，并用于每一组图像。由于尺度数设为5，**每组图像可以得到4幅DoG图像**。这个DoG图像是尺度不变的Laplacian of Gaussian，它对于检测特征点是很有用的。

构造DoG（差分高斯）金字塔：



#### 为什么使用差分金字塔

1）差分高斯图像可以**直接由高斯图像相减获得**，简单高效

2）差分高斯函数是**尺度规范化的高斯拉普拉斯函数的近似**，而高斯拉普拉斯函数的极大值和极小值点是一种**非常稳定的特征点**（与梯度特征、Hessian特征和Harris角点相比）

### 找到特征点

https://blog.csdn.net/sulanqing/article/details/16370879

到现在，我们已经生成了一个尺度空间，并且使用尺度空间来计算Difference of Gaussian，它们被用于计算尺度不变性的Laplacian of Gaussian的近似。

通过得到的DoG图像可以找到好的特征点（构建高斯金字塔找到极值点），分为三个子步骤：

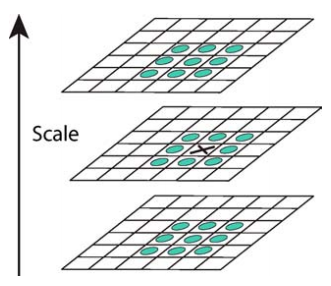
1）在DoG图像中**找到极大或极小像素点**（确定潜在特征点）；

2）找到子像素级的**极大极小值点**（精确确定特征点的位置）；

3）去除不稳定的特征点（消除边界和低对比度的特征）。

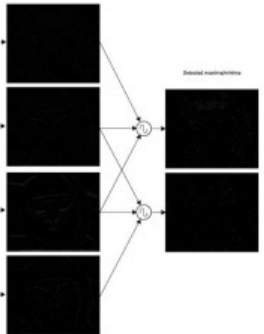
#### 极大或极小像素点（确定潜在特征点）

第一步是粗糙地找到极大极小值像素点，这很简单，我们可以**扫描每个像素并且检测所有的邻接像素点**，邻接像素点不仅包括当前图像中的邻接像素，而且包括上一层和下一层图像中的邻接像素。



在DoG图像中找极大极小像素点，X标记当前像素点，绿色的圈标记邻接像素点，用这个方式，最多检测26个像素点。如果它是所有邻接像素点的最大值或最小值点，X被标记为特征点。

通常对于非极大或极小值点不需要遍历所有26个邻接像素点，少数的几个检测就足够抛弃它。注意最高层和最底层的尺度是不需要检测的，经过上一个操作，我们从每组图像中得到4幅DoG图像，只需要对中间的两幅DoG图像进行极大极小值像素点进行检测：



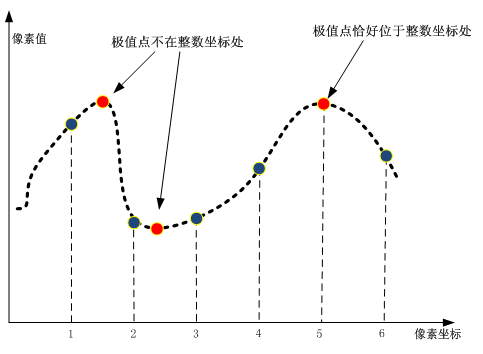
尺度空间中的极值点已经确定出来了，下面有两个问题需要解决：

1）这些点是最终我们确定的SIFT特征点集的**超集**，该超集里包含许多不稳定的特征点，因此必须去掉这些不稳定的特征点。这些不稳定的特征点主要包含两类：低对比度的点（对噪声敏感）和边缘点。

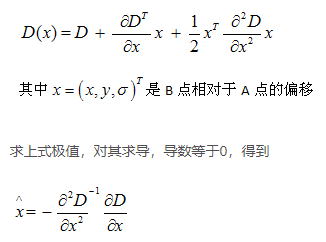
2）这一步骤中极值点的坐标还是离散的整数值，如何精确确定特征点的位置。

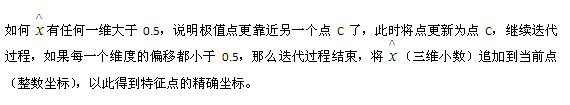
#### 子像素的极大或极小像素点（精确确定特征点的位置）

由于图像是一个离散的空间，特征点的位置的坐标都是整数，但是极值点的坐标并不一定就是整数。



如何从离散空间中估计出极值点的精确位置是重要的。为了精确确定极值点坐标，Brown和Lowe使用了三元二次函数，通过迭代确定极值点的位置，具有良好的效果。主要是根据**泰勒公式**，泰勒公式作用：**用值已知的点A估计点A附近的某点B的值**：





#### 去除不稳定的特征点

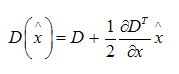
从上一步能产生许多特征点，其中的一些特征点位于**边界**，或者**亮度上没有足够的对比度**。在这些情况下，它们都不是有用的特征，所以我们要抛弃它们。

使用类似于Harris Corner检测中所使用的方法来**去除位于边界上的特征**，通过**检测亮度来去除低对比度的特征**。

##### 去除对比度低的点

如果从DoG图像中检测到的特征点的亮度值少于某个值，那么该特征点就被拒绝。因为我们已经得到子像素的特征点（极大或极小像素点），我们可以得到子像素的亮度，如果亮度少于某个值，那么就拒绝它。

在求出了极值点的精确的位置后，将求出的 x 带入原式，得：



利用这个函数去除对比度低的点，lowe文中，当D(x)<=0.03时，去除这个特征点。

##### 去除边缘点

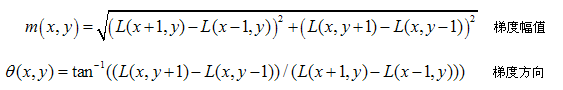
要想移除边界特征，我们要计算特征点附近高斯模糊图像的梯度。

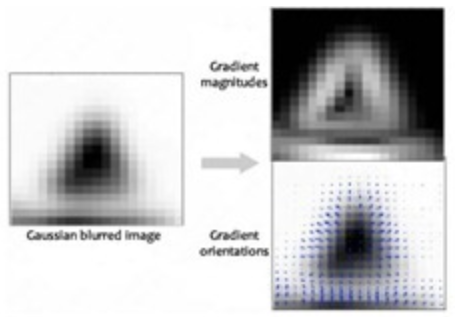
### 确定特征点方向

在上个步骤，我们得到了合理的特征点，并且这些特征点被测试是在位置上稳定的，容易识别匹配的。由于我们已经知道特征点是在哪个尺度被检测到的，所以这些特征点也是**尺度不变**的。接下来要对每个特征点赋一个方向，以使这些特征点具有**旋转不变性**。特征点所拥有的不变性越多越好。

在特征点的确定过程中，特征点的坐标以及尺度被确定下来（坐标很重要，尺度更重要，到后来，你会发现尺度这个参量在整个描述子生成过程中越来越重要，因为正是运用尺度参量实现的Scale-Invariant，尺度不变！），**根据特征点的尺度选择高斯金字塔中的图像**，然后**在这幅图像上确定该特征点的方向**。这里体现的正是尺度不变的思想。

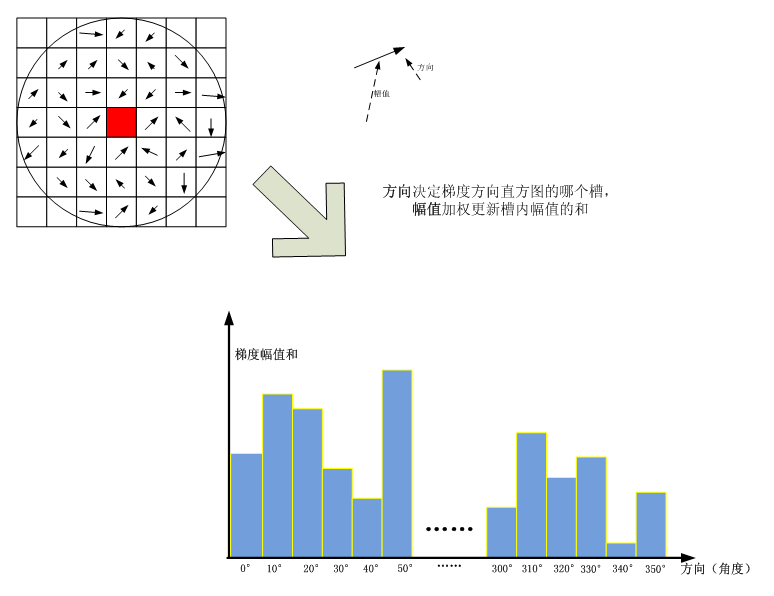
在**每个特征点**周围**计算图像的梯度方向和大小**，我们可以**得到最显著的方向**，并且将该方向赋给该特征点。后面的操作都相对该方向进行计算，确保了旋转不变性。





将高斯模糊图像分解为梯度大小图和方向图

构造该点邻域梯度方向直方图，将一圆周360°划分成36个槽，从0°开始每槽递增10°。根据邻域点的方向、梯度的幅值以及距离特征点的远近构建上述梯度方向直方图，如下图所示：



根据梯度直方图，**直方图峰值所对应的的角度就是当前特征点A的方向**，同时如果某角度的梯度幅值和>=峰值的80%，那么就产生一个新的特征点B，这个点的坐标、尺度等参数同A，但是角度不同。

至此，特征点的方向确定完毕。

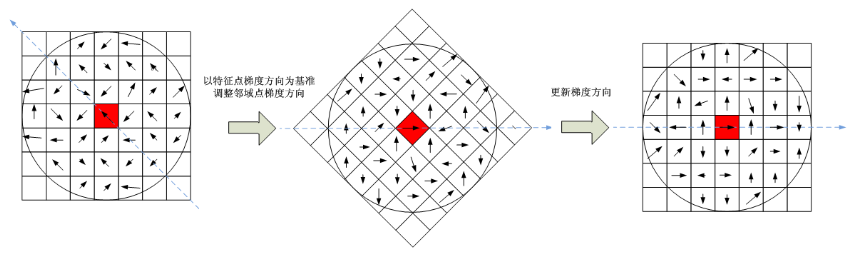
### 生成图像局部描述符

https://blog.csdn.net/sulanqing/article/details/16371345

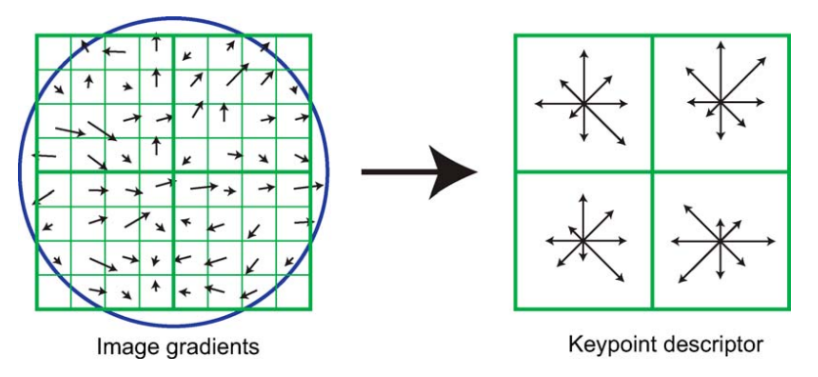
http://www.cvvision.cn/8523.html

到了本步骤，图像中**每个特征点的坐标、尺度以及方向都确定了**，下面开始根据这些信息获得描述子。

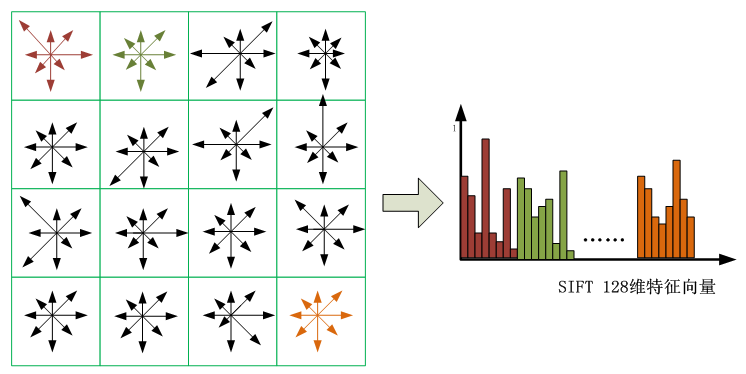
与上文中提到的确定特征点方向类似，生成描述子也是根据**以特征点为中心的图像局部信息**：首先**根据特征点的尺度选择高斯金字塔中的图像**，然后**计算特征点邻域范围内各点的梯度方向和梯度的幅值**，并**根据上文得到的特征点梯度方向更新这些梯度的方向**，以此达到描述子的方向不变性。



方向不变性完成后，开始**计算特征描述符**了。描述符计算过程同样**基于梯度方向直方图**，只是这次直方图的槽是以45°划分的（因此每个直方图只有8个槽），而不是10°。具体过程如下图所示：



统计每个4×4块的方向梯度直方图，为了去除光照对描述子的影响，对梯度直方图进行**归一化处理**。然后**将每个直方图槽数据串联即可得到SIFT描述子**，lowe提出当梯度方向直方图是4×4维的时候，SIFT描述子具有最好的区分度，因此这就诞生了老生长谈的128维SIFT特征向量，如下图所示：



如上图所示，将特征点周围**16\*16**的窗口分解为**16个4\*4**的子窗口，在每个4\*4的子窗口中，计算出梯度的大小和方向，并用一个8个bin的直方图来统计子窗口的平均方向。

每个特征点创建一个唯一标识它的“指纹”，SIFT算法作者将它称为SIFT描述子(descriptor)。所生成的SIFT描述子**既要能让相同场景中图像的特征点能够正确匹配**，而且还**要让不同场景中图像的特征点能够正确区分**。