**SIFT算法**

## 1、算法简介

Sift（Scale-invariant feature transform）尺度不变特征转换，CV界中赫赫有名的算法，由David Lowe提出。

Sift算法可以将一幅图像映射（变换）为一个局部特征向量集；特征向量具有平移、缩放、旋转不变性，同时对光照变化、仿射及投影变换也有一定的不变性。

SIFT算法的特点有：（只是理论，所以看看就好）

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特征向量；

4.高速性，经优化的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。而算法在一定程度上可解决：

**1.目标的旋转、缩放、平移**

**2.投影变换**

**3.光照影响**

**4.目标遮挡**

**5.杂物场景**

**SIFT算法的实质是在不同的尺度空间上查找关键点，并计算出关键点的方向。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。**

**SIFT算法可以分解为如下四步：**

**1.高斯差分（DoG）滤波：搜索所有尺度上的图像位置。通过高斯微分函数来识别潜在的对于尺度和旋转不变的兴趣点。**

**2.尺度空间的极值检测和关键点位置确定：对DoG金字塔中的每一层，进行尺度空间的极值检测（极大值和极小值），把每一个极值点作为候选点，在每个候选的位置上，通过一个拟合精细的模型来确定位置和尺度。关键点的选择依据于它们的稳定程度。**

**3.关键点方向确定：基于图像局部的梯度方向，分配给每个关键点位置一个或多个方向。所有后面的对图像数据的操作都相对于关键点的方向、尺度和位置进行变换，从而提供对于这些变换的不变性。**

**4.构建关键点特征描述符：在每个关键点周围的内，在选定的尺度上测量图像局部的梯度。这些梯度被变换成一种表示，这种表示允许比较大的局部形状的变形和光照变化。**

## 2.高斯差分（DoG）滤波：

### 2.1图像尺度空间

在图像信息处理中引入一个名为尺度的参数，通过对图像进行一些变换，获得在多个尺度空间下的图像空间序列，对这些序列可以进行一些特征的提取等操作，可以实现边缘，角点检测和不同分辨率上的特征提取。

对这个尺度空间的理解：它可以模拟人在距离目标由近到远的过程中，目标在视网膜当中形成的图像的过程。尺度越大图像越模糊，相当于我们观察远处物体，这时候关注该物体的轮廓。如下图，我们看到远方只有两个人的外形，并不能看到衣服上的花纹。



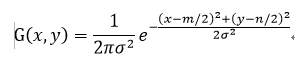
尺度空间满足视觉不变性。

* 满足灰度不变性和对比度不变性：当我们用眼睛观察物体时，当物体所处背景的光照条件变化时，视网膜感知图像的亮度水平和对比度是不同的，因此要求尺度空间算子对图像的分析不受图像的灰度水平和对比度变化的影响。
* 满足平移不变性、尺度不变性、欧几里德不变性以及仿射不变性：相对于某一固定坐标系，当观察者和物体之间的相对位置变化时，视网膜所感知的图像的位置、大小、角度和形状是不同的，因此要求尺度空间算子对图像的分析和图像的位置、大小、角度以及仿射变换无关。

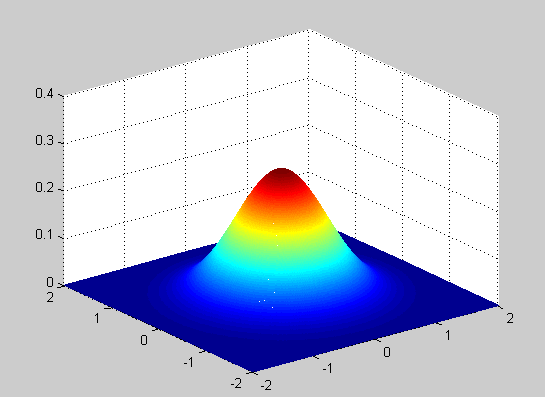
一个图像的尺度空间表示在该尺度下，该坐标处的值。

### 2.2二维高斯函数和高斯模糊

利用高斯滤波器和图像进行卷积运算，可对图像进行模糊处理。公式如下（这是一个二维的高斯滤波器）：



二维高斯曲面如下：



其中σ为正态分布的标准差，在高斯模糊中，它越大，图像越模糊。

这里要定义一个模糊半径γ, 表示模板元素到模板中心的距离。

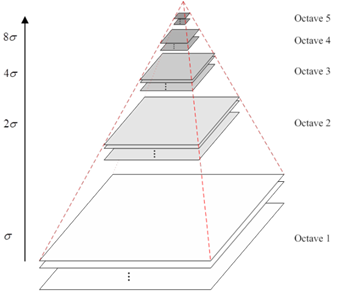
当然上面只是一个连续的曲面，在对图像进行高斯模糊的过程中需要的是高斯模板，这个模板和图像卷积便可得到高斯模糊图像。正态分布中，在大于3\*σ的范围之外存在的概率占仅0.3%，所以在3\*σ的范围之外，那些像素所起作用基本可以忽略不计了，所以高斯模板只需要计算的大小即可。根据的值计算可以计算出高斯模板。

最后提几条高斯模糊的特性（后面的理解中会被用到）

* 高斯模糊具有圆对称性，模板是中心对称的
* 对同一张图片进行连续多次高斯模糊与只用一次大的高斯模糊，可以达到一样的效果。需要满足的是方和根的关系。如两次的模糊值分别为3和4，达到的效果可以只用5就可以。

### 2.3高斯金字塔

高斯金字塔主要是为了得到不同尺度的图片。高斯金字塔是一个原始图像，产生几组（octave）每一组中又包含着几层（interval）。



当然在构建高斯金字塔之前还需要确定的是我们需要构建该金字塔的阶数（o）和每一组的层数（s）。

高斯金字塔的构建主要就是分成两步走

对图像进行不同尺度的高斯模糊。

对高斯金字塔进行降采样

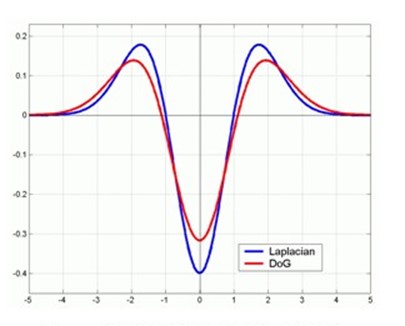
在高斯金字塔的构建中，图像每一组的大小与相应阶数的对应关系为：（原始图像以512\*512为例）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 阶数 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 大小 | 512 | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 |

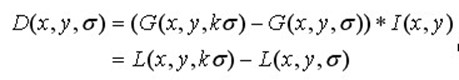
### 2.4高斯差分金字塔（这一部分是结论性的知识）

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（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 算子）与尺度归一化的高斯拉普拉斯函数非常近似。

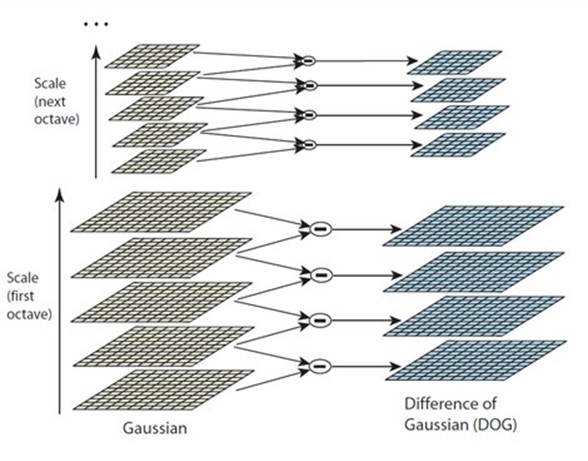
高斯拉普拉斯和高斯差分的比较如图：



如图所示，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。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：



学习前面那么多的知识，只是为了对**sift特征点**的出场做铺垫。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7所示，进行极值检测。这样就能得到sift特征点的候选人，对，只是候选。



### 2.5高斯金字塔生成的细节

上述所有的知识已经把**sift关键点候选人的选举**办法说清楚了，现在该讲一讲选举过程中所应该注意的4个问题。

这两个问题归根到底还是高斯金字塔构建的过程中的4个问题。

* 金字塔的阶数（o）的确定
* 每一组层数（s）的确定
* 每一层的尺度（σ）的确定
* 下一组的图片降采样母本的确定

**下面一一解答：**

1、金字塔的阶数（O）一般为4，也可以根据图像大小来选择，但是要满足下列关系：

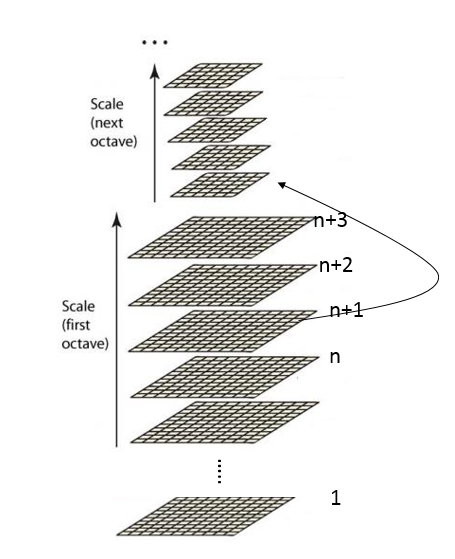
。 M，N分别表示图像的行数和列数。

2、每一阶的层数（S）一般选择5或者6，一般选择6的时候效果最好。在这边就要根据前面的说明，特征点的选举是要在相邻的两层差分金字塔上面进行检测的，所以要得到n个尺度的特征点，就要在层的差分金字塔上检测，然而要产生n+2层的差分金字塔，就要有n+3层的高斯金字塔，这样相邻的相减，才能产生n+2层差分。注意：**这里的检测都是同阶里面不同层的操作**。所以S = n + 3.

3、尺度因子的选择。先要理解几个概念：

* 高斯模糊尺度：这个尺度是我们产生模板的尺度
* 摄像头模糊的尺度：这个尺度是图像被相机镜头模糊后的尺度，一般为固定值，这里定义0.5。
* 图像的尺度：这个尺度是摄像头模糊尺度和高斯金字塔尺度的合作用，满足方和根的关系。Lowe定义图片的尺度为1.6。

这里插入解答第4个问题，即下一阶的第一层图像是根据上一阶的倒数第三层图像进行降采样得到的。



好了回来继续解答第三个问题，

在一阶的图像内，每一层之间的高斯模糊的尺度因子的比值为k,

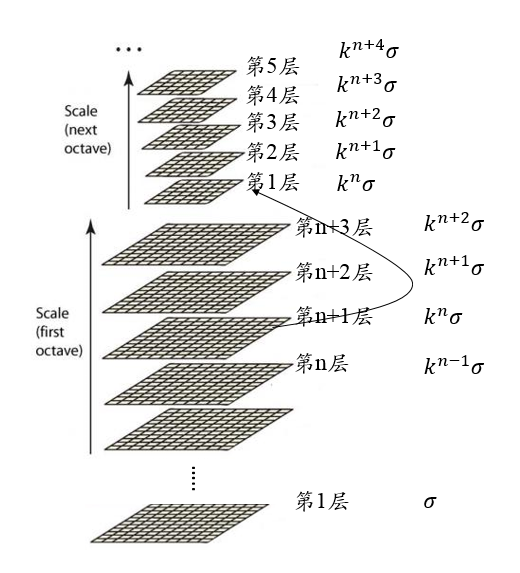


于是同一阶的第s层高斯模糊尺度就变成了

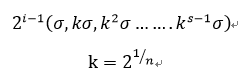


这里是这一阶的第一层图像的高斯模糊尺度

第n+1阶的图像，它的高斯模糊尺度是，这个图像的高斯模糊带到了下一阶的第一层图像中去，于是不同阶相同层的高斯模糊尺度是2倍关系。如下图所示



综上可以概括出阶内及阶之间的尺度关系了。



i：表示第i阶的图像

s：表示阶内第s层的图像

在理清楚图像互相之间的关系之后，我们需要的是第一阶，第一层的高斯模糊尺度，这样就能根据上述关系，搞到所有的高斯模糊尺度了。还记得两个数吗？0.5和1.6，这时候就派上用场了。在lowe的论文中，他定义图片的尺度是1.6，被摄像头模糊的尺度是0.5，于是可以算得高斯模糊是

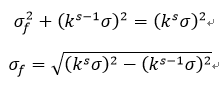


但是这个1.52对我们来说并没有啥意思，lowe为了获得更多的特征点（姑且这么解释吧）他先对原始图像进行扩大一倍。（有各种插值的方法，这里用双线性插值）然后呢再对它进行高斯平滑作为第一阶第一层，这时候它的高斯模糊尺度为



于是得到最开始的高斯模糊尺度。这下可以得到所有的高斯模糊尺度了。

到此为止，我们讲完所有理论的工作了，然而**在实际实现中，同一阶第s+1层的图片是在第s层图片的基础上进行高斯模糊得到的。故，第s+1层图片由第s层图片用一个尺度为σf产生的模板进行高斯卷积得到。**

****

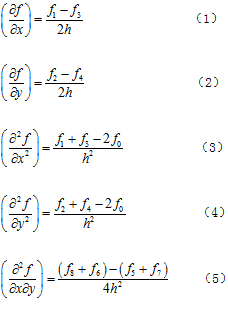
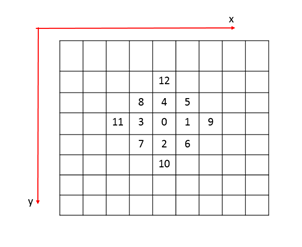
**最后产生高斯金字塔，相邻层相减，便得到高斯差分金字塔。**

## 3. 极值检测和定位

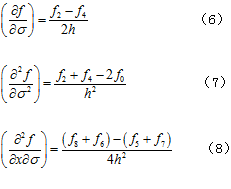
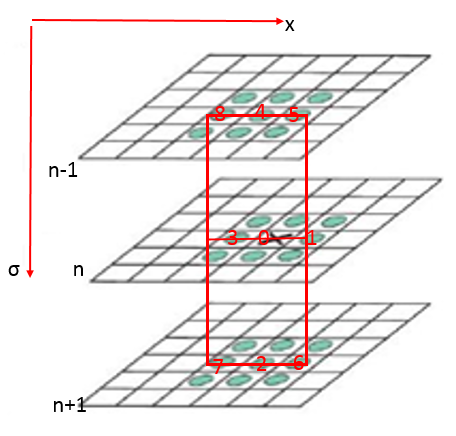
Sift的第二个步骤主要是在图像中**找到那些尺度不变性的点，这里叫做特征点**。这些特征点是由DOG金字塔中的图像通过一系列的筛选规则得到的，所以这一步骤就是为了介绍这些筛选规则，以及用这些筛选规则怎么选出代表。

### 3.1.有限差分法求导

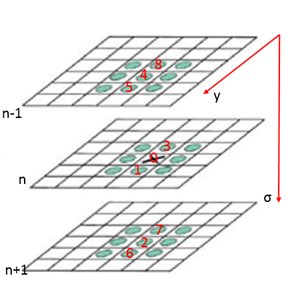
这个知识主要是介绍对于**图像中函数的求导法则**，介绍的是二元函数的求导。当然**在尺度空间中，像素值f是坐标（x，y）和尺度σ的三元函数**，它的求导法则可以类比。这里的求导其实可以理解成，每一个像素值对哪一个方向求导，则与其他方向上的无关，就是把其他方向看成一个常数来理解。在同一层的图像中，与尺度无关，所以把尺度看成一个常数。



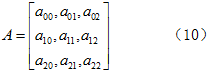
推广到三元函数，就是增加σ轴向的求导，这里把y方向上的看成无关，来求导尺度轴和x方向的关系。



以此类推y和σ方向上的关系：



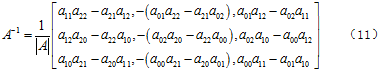
### 3.2.三阶矩阵求逆的公式



该矩阵存在逆矩阵，则它的行列式不等0，即：



所以



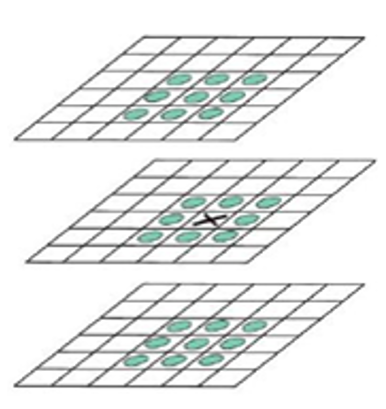
### 3.3 选举条例

1.第0轮推举（阈值检测）

由于我们不能保证图片就是完全没有噪声的，所以在推选之前我们要排除一些对比度比较低的点，这些点很不稳定，容易受到干扰，本着宁缺毋滥的原则，先把不稳定因子排除。这个阈值一般是自己设定的，在opencv中设定它为：0.5\*T/S,这里的T，lowe 定义它为0.04，s是该点所在的层。拿到这个入场券的同学，进入下一轮选举

2.第一轮选举（极值检测）

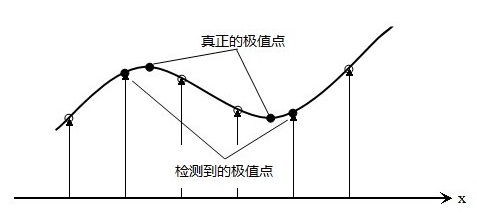
在上一篇博客里讲到了，某某某证明了，高斯差分金字塔里的**极值点比一般的（角点，Hessian，梯度函数）这些方法得到的点性质稳定**。所以很自然的就要用到极值点来做候选人。在sift算法里面极值点的检测方法是：



对图像中的每一个点进行遍历，判断每一个点是否是极值，判断的标准是，把这个点与相邻的层，上下层9+9=18个点和同一层中8个点，共18+8=26个点进行比较，看这个点是否是最大值，还是最小值，若是，则保留下来，当做候选人。

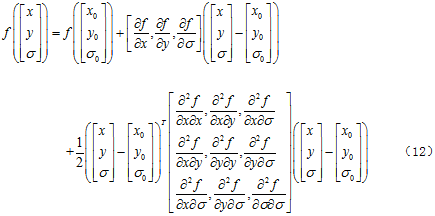
3.第二轮选举（极值点精确定位）

在sift算法中，把尺度看成连续的，而在前面的极值检测中，只是把它当成离散点来计算，于是极值点的检测就会出现下图的情况，这怎么能符合Lowe同学严谨的风格呢？



所以在这一轮的选举中，我们的目标就是找到精确的极值点，其实这里的精确点的位置，是一个亚像素级别的概念，比如一个数的个位十位百位都确定了，要更加精细的去找这个数的小数点后的那个数据，就用**插值**来计算。

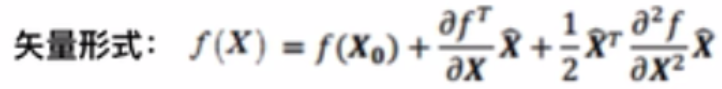
这里我们需要高数和线性代数的一些知识。高数中提过，函数f（x）在x0处可以进行泰勒展开。于是我们也在这个检测到的极值点处，对它进行泰勒级数展开（这里只展开到二阶项，高次项砍掉），以此来拟合这个三维二次函数。于是得到



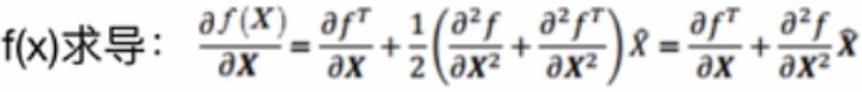
把上面的式子写成矢量的形式：



好了我们拟合出来了该点附近的函数了，X表示拟合之后连续空间下面的插值点坐标，设则表示：相对于插值中心的偏移量，这下公式（13）可以用偏移量表示：

 （14）

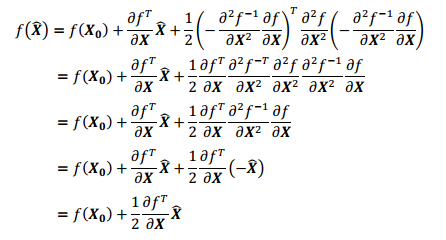
该咋求极值呢？当然是求导啦！导数为0的那个点，于是得到下面的求导式子：

 (15)

让导数等于0，就可以得到极值点下的相对于插值中心的偏移量：

（16）

把式(16)代入式（14）可以得到该极值点下的极值为：

(17)

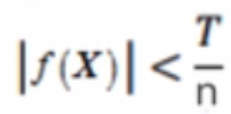
Ok，其实到这里，已经把精确的极值点找到了，但是真的是这样的吗？答案当然不是。上述的知识只是告诉我们找到偏移量的一个大的方向，真正的细节并没有说明白！

这里该注意几个问题：

* 当求出来的偏移量很大的时候，这时就表明精确的极值点已经完全偏离了离散比较得到的极值点，这时候就得删除它
* 当求出来的偏移量大于0.5，（只要x，y，和σ任意一个量大于0.5）就表示这个插值点偏离了插值中心，这时候就应该改变插值中心，继续使用上述的泰勒展开进行拟合，一直到插值偏移量小于0.5,（指的是三个量的偏移都小于0.5）
* 当然在第二个问题中，它的解决办法是一个迭代的过程，当然它也有自己的迭代停止条件，但是如果要迭代很多次才能找到那个精确点，那说明这个点本身就有自己不稳定的成分，（与问题1类似）所以就要设置一个迭代次数的限制阈值，超过这个阈值，迭代停止！这个点也得出局！

4.第三轮选举（低对比度筛选）

在上一步的检测中我们得到了精确点的位置了，也得到精确点的值了，这时候又有一个问题来了，精确点的值如果很小，那很大程度上是不稳定的点，于是很遗憾，这些精确点应该出局！所以这一轮的标准是：

 (18)

论文中T=0.04，n表示处于该组的第几层。

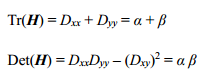
5.第四轮筛选（消除边缘效应）

在DOG函数中，它存在比较强的边缘效应，而当特征点在边缘的时候，这些点就会很不稳定，（1是因为这些点比较难定位，他们具有定位的歧义性。2是因为这些点容易受到噪声的干扰而变得不稳定）所以呢，我们应该把这些披着羊皮的狼（边缘效应很强的点）找出来。这里的方法和Harris角点检测算法的原理相似，通过Hessian矩阵得到：

(19)

其中分别表示对DOG图像的像素在x轴和y轴方向上的二阶导数和二阶混合偏导数。由于该像素点的主曲率和H的特征值成正比，该矩阵的特征值代表着x轴和y轴方向上的梯度。但是为了避免求解矩阵的特征值我们只要知道这两个值的比例就可以得到该点的主曲率。令α为最大的特征值，β为最小的特征值。

引入两个量 

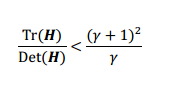
(20)

这里我们先删除掉那些行列式为负数的点，即，因为如果像素的曲率有不同的符号，则该点肯定不会是特征点。

接着设并且 ，其中，于是得到：

(21)

上面式子的结果只和两个特征值的比例有关，只有在两个特征值相等的时候，式(21)才最小，随着的增大，该式越大，说明两个特征值的比值就越大，即在某一方向上的梯度值就越大，而另一方向上的梯度越小，这便是边缘所符合的情况。所以为了剔除这些边缘点，我们让γ值小于一定的阈值，因此为了检测主曲率是否在某个阈值γ之下，只需检测

（22）

对于不满足式22的点，删除！当然这里需要给出Lowe所建议的值，γ为10.

# 4. 方向角度确定

方向梯度直方图的描述：<https://blog.csdn.net/passball/article/details/82254256>

第四部分的具体介绍：<https://blog.csdn.net/zddblog/article/details/7521424>

<https://www.cnblogs.com/fcfc940503/p/11492540.html>

<https://www.cnblogs.com/Alliswell-WP/p/SIFT.html>