Sobel,

Sobel算子是像素图像边缘检测中最重要的算子之一.在技术上，它是一个离散的一阶[差分](https://baike.baidu.com/item/%E5%B7%AE%E5%88%86" \t "https://baike.baidu.com/item/Sobel%E7%AE%97%E5%AD%90/_blank)算子，用来计算图像亮度函数的一阶梯度之近似值。在图像的任何一点使用此算子，将会产生该点对应的梯度矢量或是其法矢量。

该算子包含两组3x3的[矩阵](https://baike.baidu.com/item/%E7%9F%A9%E9%98%B5" \t "https://baike.baidu.com/item/Sobel%E7%AE%97%E5%AD%90/_blank)，分别为横向及纵向，将之与图像作平面[卷积](https://baike.baidu.com/item/%E5%8D%B7%E7%A7%AF" \t "https://baike.baidu.com/item/Sobel%E7%AE%97%E5%AD%90/_blank)，即可分别得出横向及纵向的亮度[差分](https://baike.baidu.com/item/%E5%B7%AE%E5%88%86" \t "https://baike.baidu.com/item/Sobel%E7%AE%97%E5%AD%90/_blank)近似值。如果以A代表原始图像，Gx及Gy分别代表经横向及纵向[边缘检测](https://baike.baidu.com/item/%E8%BE%B9%E7%BC%98%E6%A3%80%E6%B5%8B" \t "https://baike.baidu.com/item/Sobel%E7%AE%97%E5%AD%90/_blank)的图像，其公式如下:

[IMG_256](https://baike.baidu.com/pic/Sobelç®å­/11000092/0/11385343fbf2b21119bc68a2c88065380dd78e48?fr=lemma%26ct=single)

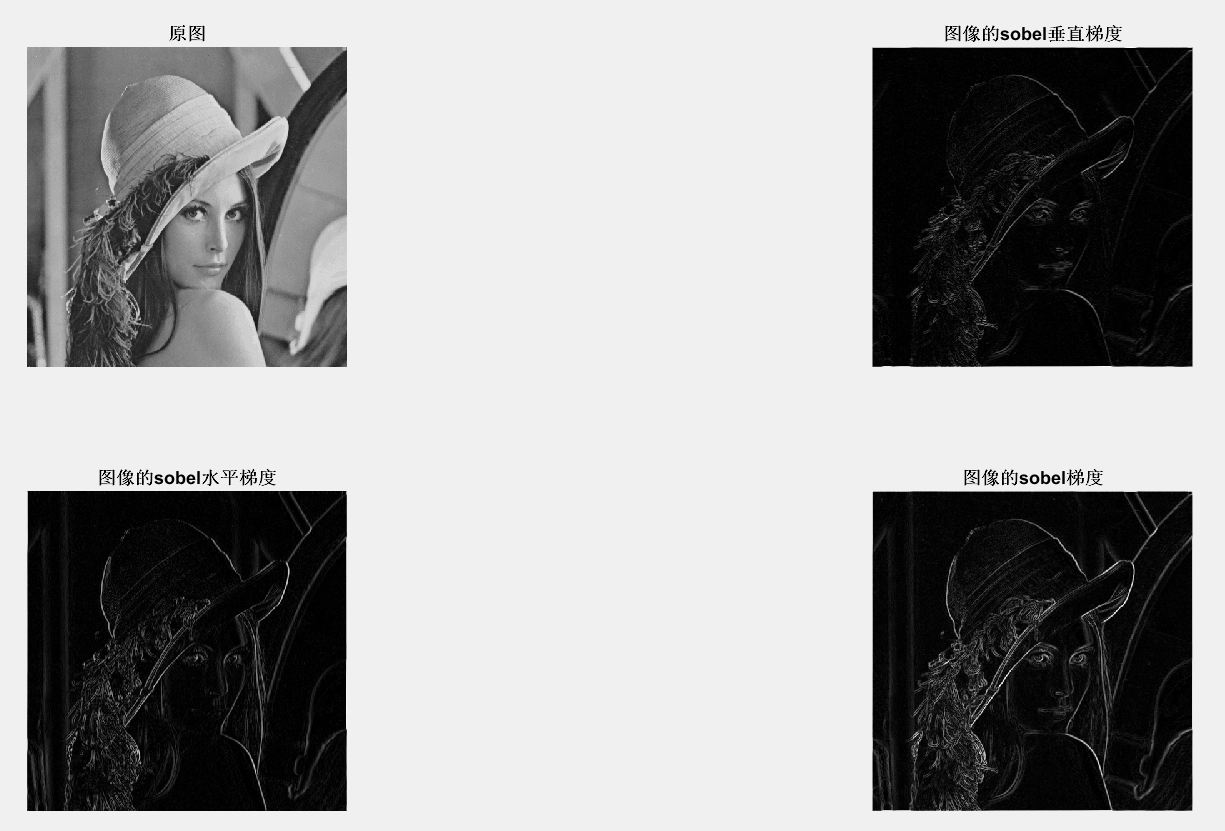
图像的每一个像素的横向及纵向梯度近似值可用以下的公式结合，来计算梯度的大小。

[IMG_257](https://baike.baidu.com/pic/Sobelç®å­/11000092/0/718e25c7978d5cfed0006077?fr=lemma%26ct=single)

可用以下公式计算梯度方向。

[IMG_258](https://baike.baidu.com/pic/Sobelç®å­/11000092/0/969cbf441985fda0b3b7dc71?fr=lemma%26ct=single)

在以上例子中，如果以上的角度Θ等于零，即代表图像该处拥有纵向边缘，左方较右方暗。



Robert,

Roberts边缘检测算子是一种利用局部差分算子寻找边缘的算子,Robert算子图像处理后结果边缘不是很平滑。经分析，由于Robert算子通常会在图像边缘附近的区域内 产生较宽的响应，故采用上述算子检测的边缘图像常需做细化处理，边缘定位的精度不是很高

任意一对互相垂直方向上的差分可以看成求梯度的近似方法，罗伯特算子是利用这种原理，采用的对角方向相邻两像素值之差代替该梯度值，  
它在实际应用中可用如下公式表示:

g(x,y)={[f(x,y)-f(x+1,y+1)]2+[f(x+1,y)-f(x,y+1)]2}12

其中f(x,y)是输入图像,g(x;y)是输出图像。

再选取适当的门限阈值TH，如果g（x，y）里的某个像素点 大于此门限阈值TH，则将对应得像素点认为是阶跃边缘点。

这样就得到了边缘轮廓。



Prewitt,

Prewitt算子是一种一阶[微分算子](https://baike.baidu.com/item/%E5%BE%AE%E5%88%86%E7%AE%97%E5%AD%90" \t "https://baike.baidu.com/item/Prewitt%E7%AE%97%E5%AD%90/_blank)的[边缘检测](https://baike.baidu.com/item/%E8%BE%B9%E7%BC%98%E6%A3%80%E6%B5%8B" \t "https://baike.baidu.com/item/Prewitt%E7%AE%97%E5%AD%90/_blank)，利用像素点上下、左右邻点的[灰度](https://baike.baidu.com/item/%E7%81%B0%E5%BA%A6/4615393" \t "https://baike.baidu.com/item/Prewitt%E7%AE%97%E5%AD%90/_blank)差，在边缘处达到极值检测边缘，去掉部分伪边缘，对噪声具有平滑作用 。其原理是在图像空间利用两个方向模板与图像进行[邻域](https://baike.baidu.com/item/%E9%82%BB%E5%9F%9F" \t "https://baike.baidu.com/item/Prewitt%E7%AE%97%E5%AD%90/_blank)[卷积](https://baike.baidu.com/item/%E5%8D%B7%E7%A7%AF" \t "https://baike.baidu.com/item/Prewitt%E7%AE%97%E5%AD%90/_blank)来完成的，这两个方向模板一个检测水平边缘，一个检测垂直边缘。



对数字图像f(x，y)，Prewitt算子的定义如下：

G(i)=|[f(i-1,j-1)+f(i-1,j)+f(i-1，j+1)]-[f(i+1,j-1)+f(i+1，j)+f(i+1，j+1)]|

G(j)=|[f(i-1,j+1)+f(i,j+1)+f(i+1，j+1)]-[f(i-1,j-1)+f(i,j-1)+f(i+1，j-1)]|

则 P(i,j)=max[G(i),G(j)]或 P(i,j)=G(i)+G(j)

经典Prewitt算子认为：凡[灰度](https://baike.baidu.com/item/%E7%81%B0%E5%BA%A6/4615393" \t "https://baike.baidu.com/item/Prewitt%E7%AE%97%E5%AD%90/_blank)新值大于或等于[阈值](https://baike.baidu.com/item/%E9%98%88%E5%80%BC" \t "https://baike.baidu.com/item/Prewitt%E7%AE%97%E5%AD%90/_blank)的像素点都是边缘点。即选择适当的阈值T，若P(i,j)≥T，则(i,j)为边缘点，P(i,j)为边缘图像。这种判定是欠合理的，会造成边缘点的误判，因为许多噪声点的[灰度值](https://baike.baidu.com/item/%E7%81%B0%E5%BA%A6%E5%80%BC/10259111" \t "https://baike.baidu.com/item/Prewitt%E7%AE%97%E5%AD%90/_blank)也很大，而且对于幅值较小的边缘点，其边缘反而丢失了。



Laplacian,

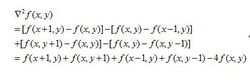
Laplacian 算子是n维欧几里德空间中的一个二阶微分算子，定义为梯度grad的散度div。可使用运算模板来运算这定理定律。

如果f是二阶[可微](https://baike.baidu.com/item/%E5%8F%AF%E5%BE%AE" \t "https://baike.baidu.com/item/Laplacian%E7%AE%97%E5%AD%90/_blank)的实函数，则f的拉普拉斯算子定义为：

(1) f的拉普拉斯算子也是笛卡儿坐标系中的所有非混合二阶[偏导数](https://baike.baidu.com/item/%E5%81%8F%E5%AF%BC%E6%95%B0" \t "https://baike.baidu.com/item/Laplacian%E7%AE%97%E5%AD%90/_blank)求和：

(2) 作为一个二阶[微分算子](https://baike.baidu.com/item/%E5%BE%AE%E5%88%86%E7%AE%97%E5%AD%90" \t "https://baike.baidu.com/item/Laplacian%E7%AE%97%E5%AD%90/_blank)，拉普拉斯算子把C函数映射到C函数，对于k ≥ 2。表达式(1)（或(2)）定义了一个算子Δ : C(R) → C(R)，或更一般地，定义了一个算子Δ : C(Ω) → C(Ω)，对于任何[开集](https://baike.baidu.com/item/%E5%BC%80%E9%9B%86" \t "https://baike.baidu.com/item/Laplacian%E7%AE%97%E5%AD%90/_blank)Ω。

对于阶跃状边缘，[导数](https://baike.baidu.com/item/%E5%AF%BC%E6%95%B0" \t "https://baike.baidu.com/item/Laplacian%E7%AE%97%E5%AD%90/_blank)在边缘点出现零交叉，即边缘点两旁[二阶导数](https://baike.baidu.com/item/%E4%BA%8C%E9%98%B6%E5%AF%BC%E6%95%B0" \t "https://baike.baidu.com/item/Laplacian%E7%AE%97%E5%AD%90/_blank)取异号。据此，对[数字图像](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E5%9B%BE%E5%83%8F" \t "https://baike.baidu.com/item/Laplacian%E7%AE%97%E5%AD%90/_blank){f（i，j）}的每个像素，取它关于x轴方向和y轴方向的[二阶差分](https://baike.baidu.com/item/%E4%BA%8C%E9%98%B6%E5%B7%AE%E5%88%86" \t "https://baike.baidu.com/item/Laplacian%E7%AE%97%E5%AD%90/_blank)之和，表示为

[](https://baike.baidu.com/pic/Laplacianç®å­/3228202/0/728da9773912b31ba6b10b618618367adbb4e19d?fr=lemma%26ct=single)

函数的拉普拉斯算子也是该函数的黑塞矩阵的迹,可以证明，它具有各向同性，即与坐标轴方向无关，坐标轴旋转后梯度结果不变。如果邻域系统是4 邻域，Laplacian 算子的模板为：

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

如果邻域系统是8 邻域，Laplacian 算子的模板为：

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 1 |
| 1 | -8 | 1 |
| 1 | 1 | 1 |

前面提过，Laplacian 算子对噪声比较敏感，所以图像一般先经过平滑处理，因为平滑处理也是用模板进行的，所以，通常的分割算法都是把Laplacian 算子和平滑算子结合起来生成一个新的模板。



Canny

**Canny边缘检测算子**是John F. Canny于1986年开发出来的一个多级边缘检测算法。Canny的目标是找到一个最优的边缘检测算法，最优边缘检测的含义是：

好的检测- 算法能够尽可能多地标识出图像中的实际边缘。

好的定位- 标识出的边缘要与实际图像中的实际边缘尽可能接近。

最小响应- 图像中的边缘只能标识一次，并且可能存在的图像噪声不应标识为边缘。

为了满足这些要求Canny使用了[变分法](https://baike.baidu.com/item/%E5%8F%98%E5%88%86%E6%B3%95/83603" \t "https://baike.baidu.com/item/Canny%E7%AE%97%E5%AD%90/_blank)，这是一种寻找满足特定[功能](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD/10346898" \t "https://baike.baidu.com/item/Canny%E7%AE%97%E5%AD%90/_blank)的[函数](https://baike.baidu.com/item/%E5%87%BD%E6%95%B0/301912" \t "https://baike.baidu.com/item/Canny%E7%AE%97%E5%AD%90/_blank)的方法。最优检测使用四个[指数](https://baike.baidu.com/item/%E6%8C%87%E6%95%B0/3519666" \t "https://baike.baidu.com/item/Canny%E7%AE%97%E5%AD%90/_blank)函数项的和表示，但是它非常近似于[高斯函数](https://baike.baidu.com/item/%E9%AB%98%E6%96%AF%E5%87%BD%E6%95%B0/2670951" \t "https://baike.baidu.com/item/Canny%E7%AE%97%E5%AD%90/_blank)的一阶[导数](https://baike.baidu.com/item/%E5%AF%BC%E6%95%B0/579188" \t "https://baike.baidu.com/item/Canny%E7%AE%97%E5%AD%90/_blank)。

### 寻找梯度

图像中的边缘可能会指向不同的方向，所以Canny算法使用4个mask检测水平、垂直以及对角线方向的边缘。原始图像与每个mask所作的卷积都存储起来。对于每个点我们都标识在这个点上的最大值以及生成的边缘的方向。这样我们就从原始图像生成了图像中每个点亮度梯度图以及亮度梯度的方向。

### 跟踪边缘

较高的亮度梯度比较有可能是边缘，但是没有一个确切的值来限定多大的亮度梯度是边缘多大又不是，所以Canny使用了[滞后](https://baike.baidu.com/item/%E6%BB%9E%E5%90%8E/1219119" \t "https://baike.baidu.com/item/Canny%E7%AE%97%E5%AD%90/_blank)阈值。

IMG_256

它在梯度方向的三阶方向导数满足符号条件

IMG_257

其中

IMG_258

表示用高斯核平滑原始图像得到的[尺度空间](https://baike.baidu.com/item/%E5%B0%BA%E5%BA%A6%E7%A9%BA%E9%97%B4" \t "https://baike.baidu.com/item/Canny%E7%AE%97%E5%AD%90/_blank)表示{\displaystyle L}计算得到的偏导数。用这种方法得到的边缘片断是连续曲线，这样就不需要另外的边缘跟踪改进。滞后阈值也可以用于亚像素边缘检测。

## 参数

[编辑](https://baike.baidu.com/item/Canny%E7%AE%97%E5%AD%90/javascript:;)

Canny算法包含许多可以调整的参数，它们将影响到算法的计算的时间与实效。

[高斯滤波器](https://baike.baidu.com/item/%E9%AB%98%E6%96%AF%E6%BB%A4%E6%B3%A2%E5%99%A8" \t "https://baike.baidu.com/item/Canny%E7%AE%97%E5%AD%90/_blank)的大小：第一步所用的平滑滤波器将会直接影响Canny算法的结果。较小的滤波器产生的模糊效果也较少，这样就可以检测较小、变化明显的细线。较大的滤波器产生的模糊效果也较多，将较大的一块图像区域涂成一个特定点的颜色值。这样带来的结果就是对于检测较大、平滑的边缘更加有用，例如彩虹的边缘。

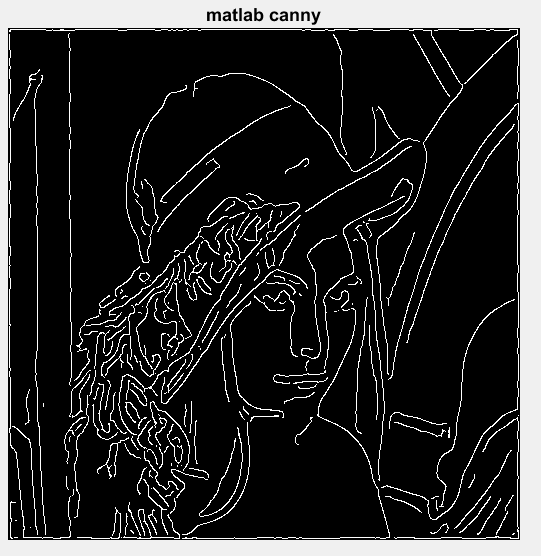
[阈值](https://baike.baidu.com/item/%E9%98%88%E5%80%BC/7442398" \t "https://baike.baidu.com/item/Canny%E7%AE%97%E5%AD%90/_blank)：使用两个阈值比使用一个阈值更加灵活，但是它还是有阈值存在的共性问题。设置的阈值过高，可能会漏掉重要信息；阈值过低，将会把枝节信息看得很重要。很难给出一个适用于所有图像的通用阈值。目前还没有一个经过验证的实现方法。

Canny算法的目标是找到一个最优的边缘检测算法，最优边缘检测的含义是：   
(1)最优检测：算法能够尽可能多地标识出图像中的实际边缘，漏检真实边缘的概率和误检非边缘的概率都尽可能小；   
(2)最优定位准则：检测到的边缘点的位置距离实际边缘点的位置最近，或者是由于噪声影响引起检测出的边缘偏离物体的真实边缘的程度最小；   
(3)检测点与边缘点一一对应：算子检测的边缘点与实际边缘点应该是一一对应。

为了满足这些要求 Canny 使用了变分法（calculus of variations），这是一种寻找优化特定功能的函数的方法。最优检测使用四个指数函数项表示，但是它非常近似于高斯函数的一阶导数。

2.算法的实现步骤：   
Canny边缘检测算法可以分为以下5个步骤：   
(1)应用高斯滤波来平滑图像，目的是去除噪声   
(2)找寻图像的强度梯度（intensity gradients）   
(3)应用非最大抑制（non-maximum suppression）技术来消除边误检（本来不是但检测出来是）   
(4)应用双阈值的方法来决定可能的（潜在的）边界   
(5)利用滞后技术来跟踪边界



经过均值滤波处理后的canny

Sobel



0.1,0.05,0.01

Prewitt

0.1,0.05,0.01

canny普遍比前俩数值大，0.5就有图片显示，但是那两个是黑色的

越小检测出的边缘越丰富

边缘检测是[图像处理](https://baike.baidu.com/item/%E5%9B%BE%E5%83%8F%E5%A4%84%E7%90%86/294902" \t "https://baike.baidu.com/item/%E8%BE%B9%E7%BC%98%E6%A3%80%E6%B5%8B/_blank)和[计算机视觉](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E8%A7%86%E8%A7%89/2803351" \t "https://baike.baidu.com/item/%E8%BE%B9%E7%BC%98%E6%A3%80%E6%B5%8B/_blank)中的基本问题，边缘检测的目的是标识数字图像中亮度变化明显的点。图像属性中的显著变化通常反映了属性的重要事件和变化。 这些包括（i）深度上的不连续、（ii）表面方向不连续、（iii）物质属性变化和（iv）场景照明变化。 边缘检测是图像处理和[计算机视觉](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E8%A7%86%E8%A7%89/2803351" \t "https://baike.baidu.com/item/%E8%BE%B9%E7%BC%98%E6%A3%80%E6%B5%8B/_blank)中，尤其是[特征提取](https://baike.baidu.com/item/%E7%89%B9%E5%BE%81%E6%8F%90%E5%8F%96/8827539" \t "https://baike.baidu.com/item/%E8%BE%B9%E7%BC%98%E6%A3%80%E6%B5%8B/_blank)中的一个研究领域。

空间域　英文： spatial domain。 释义： 又称图像空间(image space)。由图像像元组成的空间。在图像空间中以长度(距离)为自变量直接对像元值进行处理称为空间域处理。  
  
频率域。 英文： spatial frequency domain。 释义： 以频率(即波数)为自变量描述图像的特征,可以将一幅图像像元值在空间上的变化分解为具有不同振幅、空间频率和相位的简振函数的线性叠加,图像中各种频率成分的组成和分布称为空间频谱。这种对图像的频率特征进行分解、处理和分析称为频率域处理或波数域处理。  
  
二者关系：

  空间域与频率域可互相转换。在频率域中可以引用已经很成熟的频率域技术,处理的一般步骤为：①对图像施行二维离散傅立叶变换或小波变换,将图像由图像空间转换到频域空间。②在频率域中对图像的频谱作分析处理,以改变图像的频率特征。即设计不同的数字滤波器,对图像的频谱进行滤波。比如在空间图像里不好找出噪声的模式，如果变换到频率域，则比较好找出噪声的模式，并能更容易的处理。