Canny Edge Detection

STONE

# STEP 1

**利用高斯滤波器完成src图像的去噪。**

使用opencv中的GaussianBlur函数完成。

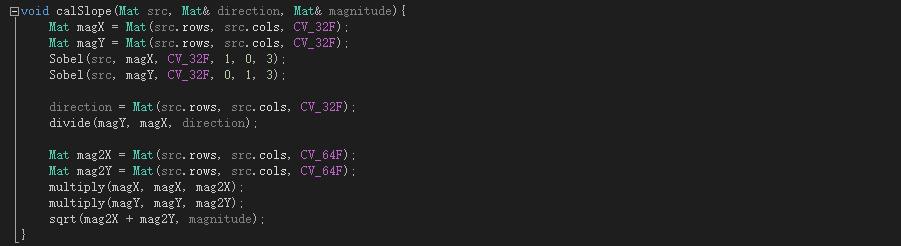
# STEP 2

**利用Sobel算子计算梯度的幅值(magnitude)和方向(direction)。**

使用Opencv中的Sobel函数完成两个方向梯度图像生成(magx, magy)，之后根据公式

|G| = |Gx| + |Gy|计算magnitude，根据公式d = Gy / Gx计算direction。

这部分我将所有操作封装在calSlope函数中，获得magnitude与direction，代码如下：



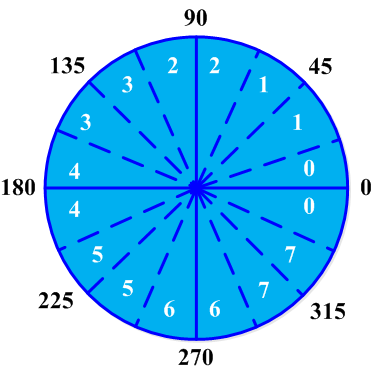
# STEP 3

**使用非极大值抑制算法（NMS）将局部梯度非最大的值去除来确定边缘。**

此处算法未使用Opencv中的函数。

首先从对从上一步获取的direction进行分析，获取其角度值。

参考下图进行说明：



对获取的角度值根据算法分成0&4, 1&5,2&6,3&7四个区域，对应下示意图进行分析

$ & #

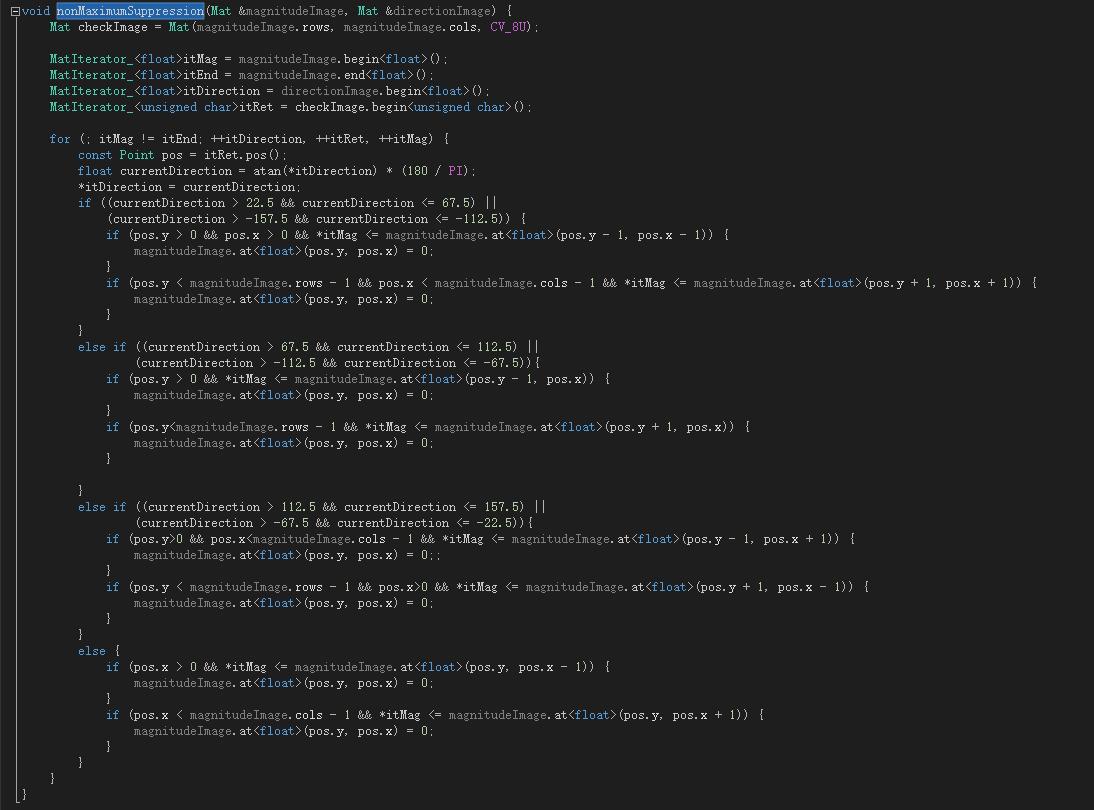
\* O \*

# & $

例如对于0&4这一个区域中的值，中心点O对应的梯度角方向直线经过的是\*的位置，那么O点像素与\*对应的两个位置点像素比较来获得结果，其他与之同理。

在区域的每一点上，将中心点像素与梯度角方向上的两个点的像素进行比较，如果比中心点像素比任意一个位置的像素小，那么说明该点非极大值，将其抑制（中心点像素设为0）。

我将这一部分的实现封装在nonMaximumSuppression函数中，获得的结果保存在magnitude中，代码如下：



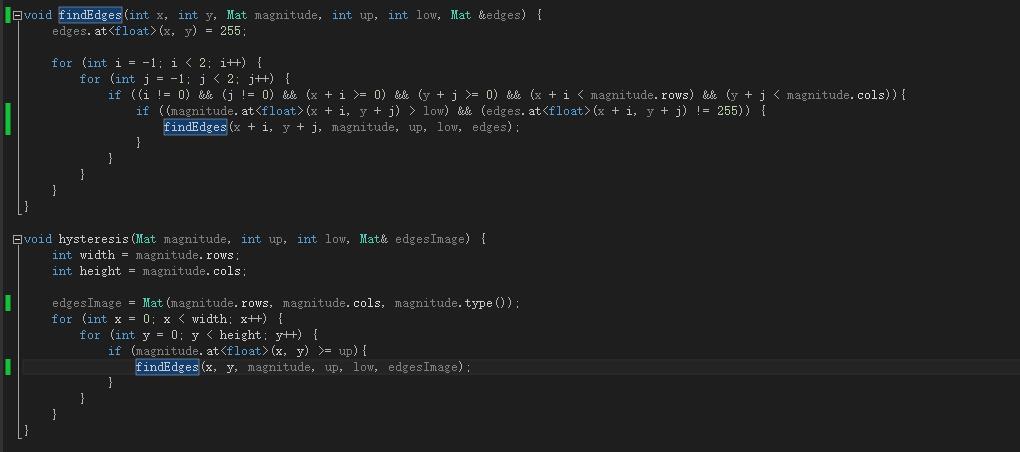
# STEP 4

**使用hysteresis双阈值算法获取边缘。**

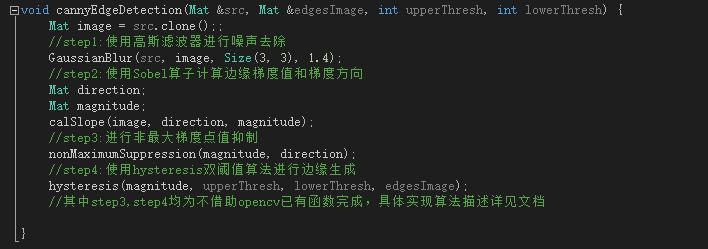
此处算法未使用Opencv中的函数。

先使用一个高阈值（UP）来进行检测，这样会获得间断的边缘，然后对于边缘点位置作为中心点进行八格临点位置中进行递归寻找，直到找到能连接闭合(某边缘点像素为255)且该中心点像素高于低阈值（LOW）【UP≈2\*LOW】时完成，这样对全图进行扫描完成边缘获取。

我将这一部分代码封装在hysteresis函数中，其中该函数会调用findEdge函数，该函数递归调用自身完成边缘连接，hysteresis函数将生成的边缘保存在magnitude中，代码如下：



# Final



使用本算法进行边缘获取，原图与在UP=90，LOW=45的情况下生成的边缘图片如下：

