

Canny 边缘检测算子的实现

Wang Ruchen

May 27, 2015

OpenCV 和 MATLAB 中 Canny 算子进行非极大值抑制和双阈值检测的实现方法是不同的。

1. 在 MATLAB 中

1.1 非极大值抑制

MATLAB 中 Canny 算子在实现非极大值抑制过程中，是比较该点和梯度方向上两点的梯度大小实现的。如图1，将 C 点的梯度值与梯度方向（蓝色线表示梯度方向）上两点 $dTmp1$ 和 $dTmp2$ 的梯度值进行比较，判断 C 点是否为极大值点。如果 C 点的梯度值比 $dTmp1$ 和 $dTmp2$ 都要大，则 C 点为极大值点，保留该点的梯度值；否则将该点的梯度值置 0。

然而，已知 C 点邻域内 8 个点的梯度值，而不知道 $dTmp1$ 和 $dTmp2$ 的梯度值。要得到 $dTmp1$ 和 $dTmp2$ 的梯度值就需要根据邻域内的已知梯度值进行线性插值，即根据图1中 $g1$ 和 $g2$ 的梯度值插值得到 $dTmp1$ 的值，根据 $g3$ 和 $g4$ 的梯度值插值 $dTmp2$ 插值。具体怎样求得 $dTmp1$ 和 $dTmp2$ 的值？已知 g_x, g_y 为 C 点在 x, y 方向的梯度， $weight = \frac{|g_x|}{|g_y|}$ ，由此可以得到（在下公式中 $dTmp1, dTmp2, g_1, g_2, g_3, g_4$ 分别代表该点的梯度值）：

$$weight = \frac{|g_x|}{|g_y|} = \frac{|dTmp1 - g_2|}{|g_1 - g_2|} = \frac{|dTmp1 - g_2|}{|C - g_2|} = ctan\theta \quad (1)$$

$$weight = \frac{|g_x|}{|g_y|} = \frac{|dTmp2 - g_4|}{|g_3 - g_4|} = \frac{|dTmp2 - g_4|}{|C - g_4|} = ctan\theta \quad (2)$$

根据公式 (1)(2) 可以推导出，计算 $dTmp1$ 和 $dTmp2$ 梯度值的公式如下：

$$dTmp1 = weight * g_1 + (1 - weight) * g_2 \quad (3)$$

$$dTmp2 = weight * g_3 + (1 - weight) * g_4 \quad (4)$$

$weight$ 为在计算 $dTmp1$ 和 $dTmp2$ 时 g_1 和 g_3 所占的权重， $(1-weight)$ 为 g_2 和 g_4 所占的权重。例

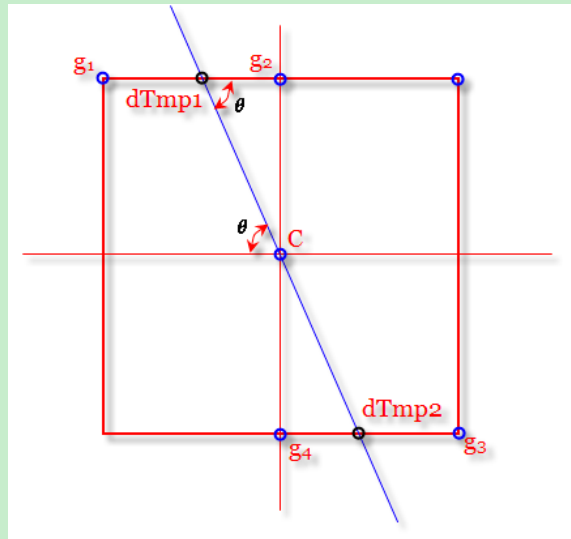


图 1: 非极大值抑制

如，当 $\text{weight}=1$ 时， $d\text{Tmp1} = g_1$ ， $d\text{Tmp2} = g_3$ ；当 $\text{weight}=0$ 时， $d\text{Tmp1} = g_2$ ， $d\text{Tmp2} = g_4$ 。

1.2 双阈值

MATLAB 中 Canny 算子是自动设置双阈值的。

通过统计得到边缘图的灰度直方图，然后计算得到累积直方图。用累积直方图 0.7 处的值作为高阈值。低阈值 = 高阈值 * 0.4。

2. 在 OpenCV 中

2.1 非极大值抑制

OpenCV 中 Canny 函数在实现非极大值抑制时，不进行插值，直接比较该点与其 8 邻域内梯度方向附近两点的梯度值大小。例如如图2，图中黑色线表示该点的梯度方向，梯度方向在 0 和 4 区域范围内，只需与 8 邻域内 a 点和 b 点进行比较。同理，如果在 1 和 5 区域内，与 8 邻域内 c 点和 d 点进行比较……

2.2 双阈值

OpenCV 中 Canny 函数的双阈值是人为设定的。

```
1 Canny(InputArray image, OutputArray edges, double threshold1, double threshold2)
```

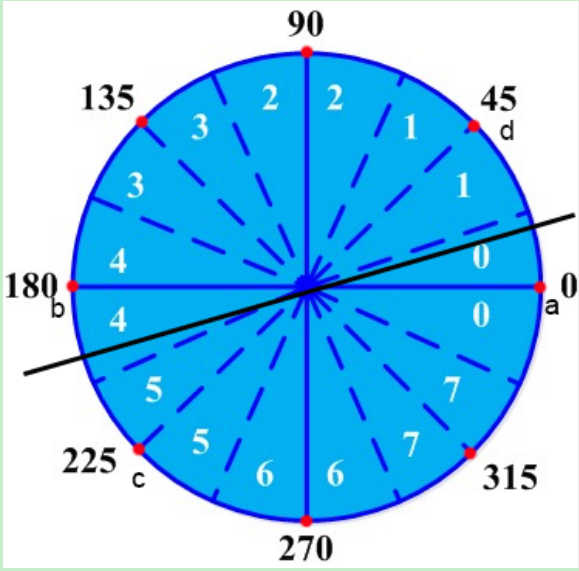


图 2: 非极大值抑制