

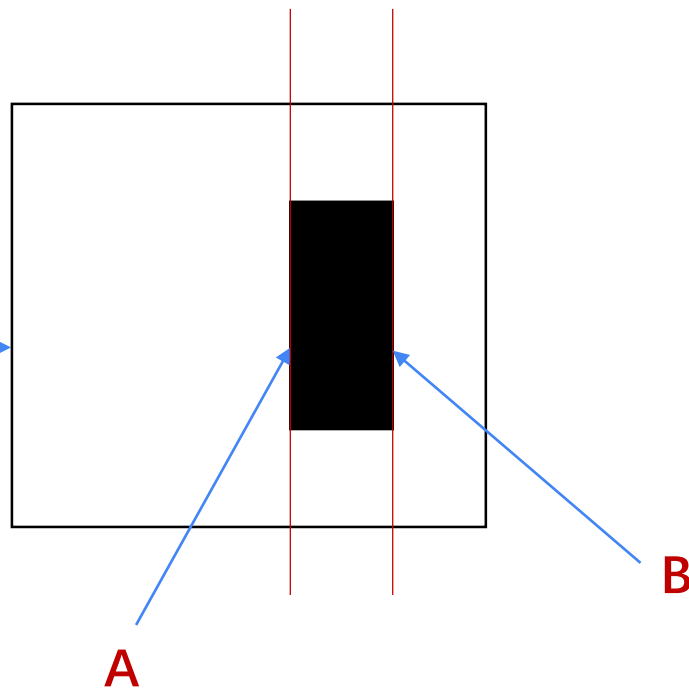
Python+OpenCV图像处理

图像梯度

sobel算子的理论基础

李大羊

示例图像

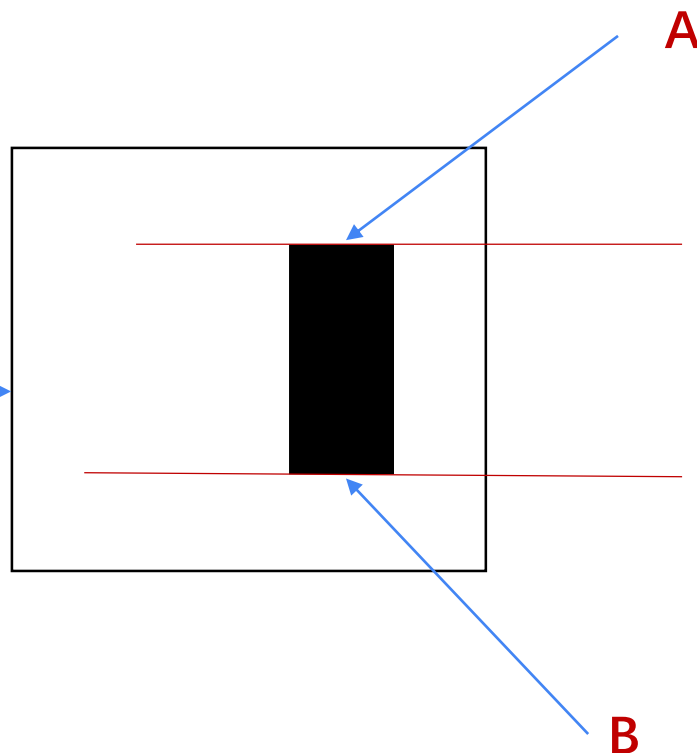


水平梯度、边界：

示例图像内，黑色块位置：

A列和B列，其右侧像素值与左侧像素值的差值不为零，是边界；
其余列，右侧像素值与左侧像素值的差值均为零，不是边界。

示例图像



垂直梯度、边界：

示例图像内，黑色块位置：

A行和B行，其下一行像素值与上一行侧像素值的差值不为零，是边界；
其余行，其下一行像素值与上一行像素值的差值均为零，不是边界。

$G_x =$

-1	0	1
-2	0	2
-1	0	1

\times

原始图像

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} P1 & P2 & P3 \\ P4 & P5 & P6 \\ P7 & P8 & P9 \end{bmatrix}$$

$$P5_x = (p3 - p1) + 2 * (p6 - p4) + (p9 - p7)$$

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} P1 & P2 & P3 \\ P4 & P5 & P6 \\ P7 & P8 & P9 \end{bmatrix}$$

$$P5_x = (p3 - p1) + 2 * (p6 - p4) + (p9 - p7)$$

右侧像素值减去左侧像素值，中间行参数稍大为2

$G_y =$

-1	-2	-1
0	0	0
1	2	1

\times 原始图像

$$G_y = \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \times \begin{array}{|c|c|c|} \hline P1 & P2 & P3 \\ \hline P4 & P5 & P6 \\ \hline P7 & P8 & P9 \\ \hline \end{array}$$

$$P5_y = (p7 - p1) + 2 * (p8 - p2) + (p9 - p3)$$

$$G_y = \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \times \begin{array}{|c|c|c|} \hline P1 & P2 & P3 \\ \hline P4 & P5 & P6 \\ \hline P7 & P8 & P9 \\ \hline \end{array}$$

$$P5_y = (p7 - p1) + 2 * (p8 - p2) + (p9 - p3)$$

下一行像素值减去上一行像素值，中间列参数稍大为2

计算近似梯度值：

$$G = \sqrt{G_x^2 + G_y^2}$$

$$G = \sqrt{G_x^2 + G_y^2} \quad \longrightarrow \quad G = |G_x| + |G_y|$$

简化版本

$$G = |G_x| + |G_y|$$

P1	P2	P3
P4	P5	P6
P7	P8	P9

$$P5_{\text{sobel}} = |p5_x| + |p5_y|$$

Sobel算子

P1	P2	P3
P4	P5	P6
P7	P8	P9

-1	0	1
-2	0	2
-1	0	1

$$P5_x = (p3 - p1) + 2 * (p6 - p4) + (p9 - p7)$$

-1	-2	-1
0	0	0
1	2	1

$$P5_y = (p7 - p1) + 2 * (p8 - p2) + (p9 - p3)$$

$$P5_{sobel} = |p5_x| + |p5_y| = |(p3 - p1) + 2 * (p6 - p4) + (p9 - p7)| + |(p7 - p1) + 2 * (p8 - p2) + (p9 - p3)|$$

Python+OpenCV图像处理

图像梯度

sobel算子的理论基础

李大羊

lilizong@gmail.com