

Harris 角点的原理与实验

July 7, 2014

1 问题

Harris 角点是什么？如何寻找图像中的角点？本文基于《Learning OpenCV》第十章“Corner Finding”一节及 OpenCV 源代码写就。

2 解答

2.1 角点是什么？

直观上，就是图像中的“角”，如正方形的四角。相对方形内部的任意点来说，在下一帧中更容易确定其位置。

2.2 Harris 角点的定义？

Harris 角点是角点用数学定义描述：在两个方向的导数都是极大值。仍然用正方形色块来描述：

- 内部点：在 X 方向导数为 0；Y 方向导数为 0；
- 边缘点：
 - 水平边缘：X 方向导数为 0；Y 方向导数为极大值
 - 垂直边缘：X 方向导数为极大值；Y 方向导数为 0
- 角点：X 方向导数为极大值；Y 方向导数为极大值。

引入 Hessian Matrix:

$$H(p) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \quad (1)$$

这个矩阵由 p 点处的二阶导数、偏导组成。

Harris 角点的定义是，如果一点处的 Hessian Matrix 满足以下条件：

$$Det(H) - k * Trace(H) * Trace(H) < \epsilon \quad (2)$$

就认为是角点。

其中：

- $Det(H)$ 是对矩阵 H 求行列式的值
- $Trace(H)$ 是矩阵 H 的“迹”
 - k 是根据经验指定的一个系数。OpenCV 中选的是 0.04
 - ϵ 是根据经验指定的一个阈值。OpenCV 做法是：用所有差值中的最大值，乘上一个 quality level。

2.3 原理到代码的对应

这个是做图像处理比较挠头的一部分，明明原理懂了，到写代码却一筹莫展。因此这里对此多写一些。

2.3.1 导数

以 X 方向的导数为例，设 $I(x, y)$ 为点 (x, y) 处的灰度值，那么该点处的导数就是：

$$\frac{\partial I}{\partial x} = I(x + 1, y) - I(x - 1, y) \quad (3)$$

即右侧点减去左侧点的灰度值。

对应的，Y 方向的导数为：

$$\frac{\partial I}{\partial x} = I(x, y + 1) - I(x, y - 1) \quad (4)$$

即下方点减去上方点的灰度值。

对啦，就是用 Sobel 算子对图像做卷积。最简单的，就是：

$$[-1, 0, 1] \quad (5)$$

$$[-1, 0, 1]^T \quad (6)$$

得到两个方向的 Sobel 图， Dx, Dy 。

2.3.2 二阶导数与偏导

参考 OpenCV 的代码，二阶导数由一阶导数相乘而来。非常简单。（却是不太明白其中的数学原理。）上面求出了 X 方向的一阶导数，记为 dx ；Y 方向的一阶导数记为 dy ，那么：

$$\frac{\partial^2 I}{\partial x^2} = dx * dx \quad (7)$$

$$\frac{\partial^2 I}{\partial x \partial y} = \frac{\partial^2 I}{\partial y \partial x} = dx * dy \quad (8)$$

$$\frac{\partial^2 I}{\partial y^2} = dy * dy \quad (9)$$

非常简单，是不是？

2.3.3 行列式与矩阵的迹

以矩阵

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix} \quad (10)$$

为例：

- $Det(H) = a * c - b * b$
- $Trace(H) = (a + c)$

2.3.4 实践中的优化

理论上，理论与实践是严格一致的；实践上，实践与理论总是有差距。上面的分析是以一点处的导数组成 Hessian Matrix；实际上，对任一点，会计算其 8 邻域所有点的导数和，然后再生成 Hessian Matrix 来计算角点。

在代码里，即对 Dx, Dy 分别用算子：

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (11)$$

求卷积。

2.3.5 代码与实验

源代码在[这里](#)。

可以与 OpenCV 的实现达到同样的效果。

3 结论

- 矩阵、导数的理论很重要。也挺有意思。
- 从理论到代码的实现好像关心的人很少。经常找到的资料略过不提。还是我太菜了？