南京農業大學

计算机视觉与图形图像处理 HW2

指导老师： 黄君贤

学年： 2024-2025

学期： 第一学期

姓名： 杨义琦

学号： 11522216

院系： 人工智能学院

完成日期： 2024 年 12 月 20 日

In class, we discussed how, given a windowing function $w(s,t)$, we can use the following *covariance* metric:

$$E_w(u,v;x,y) = \sum_{s,t} w(s,t)[I(x-s+u, y-t+v) - I(x-s, y-t)]^2, \tag{1}$$

in order to identify whether an image patch centered at (x,y) looks like a corner. In particular, large values of E_w for all possible displacements (u,v) of the window indicate that the patch is a corner.

题目 1

题目描述：

1. Assuming that the displacements u and v are small, show that the metric of Equation (1) can be approximated as:

$$E_w(u,v;x,y) \approx [u,v] \cdot \mathcal{M}_w(x,y) \cdot [u,v]^T, \tag{2}$$

where $\mathcal{M}_w(x,y)$ is the *covariance* metrix:

$$\mathcal{M}_w(x,y) = \begin{bmatrix} \sum_{s,t} w(s,t) I_x(x-s,y-t) I_x(x-s,y-t) & \sum_{s,t} w(s,t) I_x(x-s,y-t) I_y(x-s,y-t) \\ \sum_{s,t} w(s,t) I_y(x-s,y-t) I_x(x-s,y-t) & \sum_{s,t} w(s,t) I_y(x-s,y-t) I_y(x-s,y-t) \end{bmatrix}. \tag{3}$$

题解：

我们从原始的协方差度量 $E_w(u,v;x,y)$ 开始，并对其进行泰勒展开。首先，注意到 $I(x-s+u, y-t+v)$ 是 $I(x-s, y-t)$ 的一个小位移版本。对其进行一阶泰勒展开：

$$I(x-s+u, y-t+v) = I(x-s, y-t) + \nabla I(x-s, y-t) \cdot [u,v]^T + O(u^2, v^2)$$

其中 $\nabla I(x-s, y-t) = [I_x(x-s, y-t), I_y(x-s, y-t)]$ 是图像梯度。代入原始表达式：

$$\begin{aligned} E_w(u,v;x,y) &= \sum_{s,t} w(s,t) \left[I(x-s, y-t) + \nabla I(x-s, y-t) \cdot [u,v]^T + O(u^2, v^2) - I(x-s, y-t) \right]^2 \\ &= \sum_{s,t} w(s,t) \left[\nabla I(x-s, y-t) \cdot [u,v]^T + O(u^2, v^2) \right]^2 \end{aligned}$$

展开平方项：

$$E_w(u,v;x,y) = \sum_{s,t} w(s,t) \left([\nabla I(x-s, y-t)]^T [u,v]^T \right)^2 + O(u^3, v^3)$$

注意到：

$O(u^3, v^3)$ 是高阶小量，在小位移情况下可以忽略 $[\nabla I(x-s, y-t)]^T = [I_x(x-s, y-t), I_y(x-s, y-t)]$

因此：

$$E_w(u,v;x,y) \approx \sum_{s,t} w(s,t) \begin{bmatrix} I_x(x-s,y-t) & I_y(x-s,y-t) \end{bmatrix} [u,v]^T [u,v] \begin{bmatrix} I_x(x-s,y-t) & I_y(x-s,y-t) \end{bmatrix}$$

$$= [u,v] \cdot \left(\sum_{s,t} w(s,t) \begin{bmatrix} I_x(x-s,y-t)I_x(x-s,y-t) & I_x(x-s,y-t)I_y(x-s,y-t) \\ I_y(x-s,y-t)I_x(x-s,y-t) & I_y(x-s,y-t)I_y(x-s,y-t) \end{bmatrix} \right) \cdot [u,v]^T$$

这正是题目中要求证明的形式：

$$E_w(u,v;x,y) \approx [u,v] \cdot \mathcal{M}_w(x,y) \cdot [u,v]^T$$

其中 $\mathcal{M}_w$ 就是给定的协方差矩阵。证毕。

题目 2

题目描述：

2. Show that the covariance matrix can be writen equivalently as:

$$\mathcal{M}_w(x,y) = w(x,y) * \begin{bmatrix} I_x(x,y)I_x(x,y) & I_x(x,y)I_y(x,y) \\ I_y(x,y)I_x(x,y) & I_y(x,y)I_y(x,y) \end{bmatrix} \tag{4}$$

where * indicates convolution of the windowing functio $w(x,y)$ with each element of the matrix.

题解：

已知协方差的定义：

$$\mathcal{M}_w(x,y) = \begin{bmatrix} \sum_{s,t} w(s,t)I_x(x-s,y-t)I_x(x-s,y-t) & \sum_{s,t} w(s,t)I_x(x-s,y-t)I_y(x-s,y-t) \\ \sum_{s,t} w(s,t)I_y(x-s,y-t)I_x(x-s,y-t) & \sum_{s,t} w(s,t)I_y(x-s,y-t)I_y(x-s,y-t) \end{bmatrix}.$$

对于矩阵的第一个元素（左上角），展开其定义：

$$\sum_{s,t} w(s,t)I_x(x-s,y-t)I_x(x-s,y-t) = \sum_{s,t} w(s,t) \cdot I_x(x-s,y-t)I_x(x-s,y-t) \quad = w * (I_x I_x)$$

这里使用了卷积的定义：

$$w * f = \sum_{s,t} w(s,t) f(x-s,y-t)$$

同理，对于矩阵的其他元素：

第一行第二列：$\sum_{s,t} w(s,t)I_x(x-s,y-t)I_y(x-s,y-t) = w * (I_x I_y)$

第二行第一列：$\sum_{s,t} w(s,t)I_y(x-s,y-t)I_x(x-s,y-t) = w * (I_y I_x)$

第二行第二列：$\sum_{s,t} w(s,t)I_y(x-s,y-t)I_y(x-s,y-t) = w * (I_y I_y)$

因此，整个协方差矩阵可以写为：

$$\mathcal{M}_w(x,y) = w(x,y) * \begin{bmatrix} I_x(x,y)I_x(x,y) & I_x(x,y)I_y(x,y) \\ I_y(x,y)I_x(x,y) & I_y(x,y)I_y(x,y) \end{bmatrix}$$

证毕。

题目 3

题目描述：

3. As we discussed a class, we can derive various "cornerness" metrics that take the form of functionals of only the product and sum of the eigenvalues of the covariance matrix. Pick your favorite one (or propose your own), and explain how you would compute this metric effciently for the entire image, using only convolutions and element-wise operations between images. You can explain this either verbally, or using pseudocode.

题解：

经典的角点检测指标之一是 **Harris 角点响应函数**，其定义如下：

$$R = \det(\mathcal{M}) - k \cdot (\text{trace}(\mathcal{M}))^2,$$

其中：

- $\mathcal{M}$ 是协方差矩阵；
- $\det(\mathcal{M}) = \lambda_1\lambda_2$，为矩阵的行列式，对应特征值的乘积；
- $\text{trace}(\mathcal{M}) = \lambda_1 + \lambda_2$，为矩阵的迹，对应特征值的和；
- k 是经验参数，通常取 $k \in [0.04, 0.06]$。

我们选择该指标，因为它同时利用了特征值的乘积和平方和，可以有效检测角点。

为了高效地计算整个图像的角点性响应 R，我们可以利用卷积和逐元素操作，具体步骤如下：

1. **计算图像梯度：**

   使用 Sobel 算子或者类似的滤波器计算图像梯度：

   $$I_x = \frac{\partial I}{\partial x}, \quad I_y = \frac{\partial I}{\partial y}.$$

2. **计算二次梯度项：**

   计算以下三个图像，通过逐元素平方和乘积得到：

   $$I_{xx} = I_x^2, \quad I_{yy} = I_y^2, \quad I_{xy} = I_x I_y.$$

3. **卷积平滑：**

   使用窗口函数 $w(x, y)$ 对上述图像进行卷积平滑（例如，高斯核），得到局部加权和：

   $$S_{xx} = w * I_{xx}, \quad S_{yy} = w * I_{yy}, \quad S_{xy} = w * I_{xy}.$$

   这里 $*$ 表示卷积操作。

4. **计算 $\det(\mathcal{M})$ 和 $\text{trace}(\mathcal{M})$：**

   对每个像素位置，计算：

   $$\det(\mathcal{M}) = S_{xx}S_{yy} - S_{xy}^2,$$

   $$\text{trace}(\mathcal{M}) = S_{xx} + S_{yy}.$$

5. **计算响应 R：**

对每个像素位置，计算：

$$R = \det(\mathcal{M}) - k \cdot (\text{trace}(\mathcal{M}))^2.$$

伪代码实现如下：

```
输入：图像 I，窗口函数 w，参数 k
输出：角点响应 R

1. 计算梯度：
I_x = sobel_filter_x(I)
I_y = sobel_filter_y(I)

2. 计算二次梯度项：
I_xx = I_x ** 2
I_yy = I_y ** 2
I_xy = I_x * I_y

3. 卷积平滑：
S_xx = convolve(I_xx, w)
S_yy = convolve(I_yy, w)
S_xy = convolve(I_xy, w)

4. 计算行列式和迹：
det_M = S_xx * S_yy - S_xy ** 2
trace_M = S_xx + S_yy

5. 计算角点性响应：
R = det_M - k * (trace_M ** 2)
```