FEATURE DETECTION

Harris

I. HARRIS

➢ BACKGROUND

When you are solving a puzzle and you look for the place of each piece, you are looking for specific pattern or specific feature to determine the exact location of this feature [1]. However, can you describe these features?

Let's explore the following example:

For A and B, you can hardly determine their exact location in the image as they are flat surface, for C and D, you know they are on the top of the building but still not the exact location. For E and F, you can for sure determine where they exist. This is a small example to understand the importance of corners as features.

➢ HARRIS ALGORITHM

1- Compute derivatives at each pixel

a. Calculate image derivative

b. Calculate square of derivative

2- Compute second moment matrix M in a Gaussian window around each pixel

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

- If either λ is close to 0, then this is **not** a corner, so look for locations where both are large.

1- Compute corner response function R and apply a threshold.

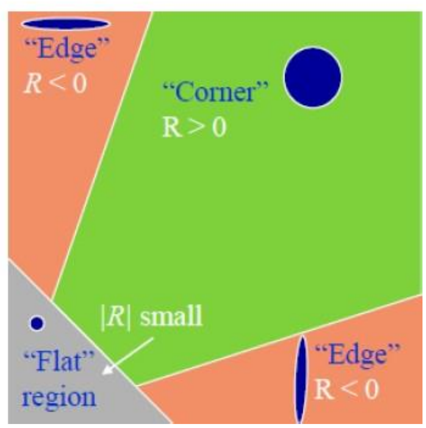
$$R = \det(M) - \alpha \, \mathrm{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha (\lambda_1 + \lambda_2)^2$$

α : constant (0.04 to 0.06)

$$det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$$

$$trace\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = a + d$$

- *R* is large for a corner
- *R* is negative with large magnitude for an edge
- |*R*| is small for a flat region



2- Find local maxima of response function (non-maximum suppression)

➢ HARRIS IN OPENCV

Use a grayscale image with the method cv2.cornerHarris to locate the corners using opencv. Usually, chessboard dominates as a basic example for corner detections.

Parameters:

- **src** – Input single-channel 8-bit or floating-point image.
- **blockSize** – Neighborhood size, calculates the covariation matrix of derivatives over the neighborhood where the derivatives are computed using the Sobel() operator.
- **ksize** – Aperture parameter for the Sobel() operator.
- **k** – Harris detector free parameter. See the formula below.
- **borderType** – Pixel extrapolation method.
- **Output** – Image to store the Harris detector responses. It has the type CV_32FC1 and the same size as src .

for each pixel (x,y) it calculates a (2x2) gradient covariance matrix $M^{(x,y)}$ over a (blocksize x blocksize) neighborhood. Then, it computes the following characteristic:

$$ds(x, y) = \det(M^{x,y}) - k \,.\, (trac(M^{x,y}))^2$$

```
filename = 'chessboard.jpg'

img = cv2.imread(filename)
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

gray = np.float32(gray)
```

```
dst = cv2.cornerHarris(gray,2,3,0.04)

#result is dilated for marking the corners, not important

dst = cv2.dilate(dst,None)

# Threshold for an optimal value, it may vary depending on the
image.

img[dst>0.01*dst.max()]=[0,0,255]
```

II. Assignment : Non-Maximum Suppression

Implement the missing codes in the attached script.