IIAI30013

Computer Vision

HW2: Harris Corner Detection

Instructor: YuanFu Yang

yfyangd@nycu.edu.tw

國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

HW2: Harris Corner Detection

- Homework due: 4/9 23:59
- Late submissions will incur a penalty of one point for each day overdue.
- The assignment allows a maximum extension of 3 days (it will not be accepted if submitted later than 3 days).
- Submit files: code and report (4 questions), and submit them in both **.zip** and **PDF** file formats respectively.
- This assignment can be carried out using Colab or completed on your PC.

HW2: Harris Corner Detection

- With the Harris corner detector described in slides (IIAI30013_03_03.pdf), mark the detected corners on the image.

HW2: Harris Corner Detection

- **Algorithm**

1) Filter the image with a Gaussian.
2) Estimate intensity gradient in two perpendicular directions for each pixel.
3) Compute *M* matrix for each image window to get their *cornerness* scores.
4) Find points whose surrounding window gave large **corner response** (f > threshold)
5) Take the points of local maxima, i.e., perform non-maximum suppression

HW2: Harris Corner Detection

- **Gaussian Smooth**

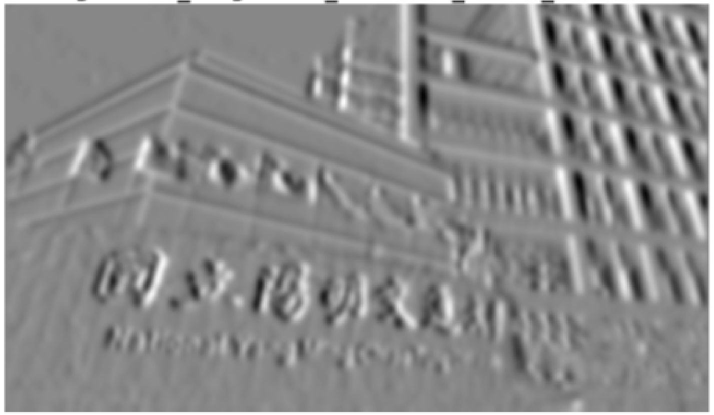
gaussian blur of kernel size = 5

gaussian blur of kernel size = 10

HW2: Harris Corner Detection

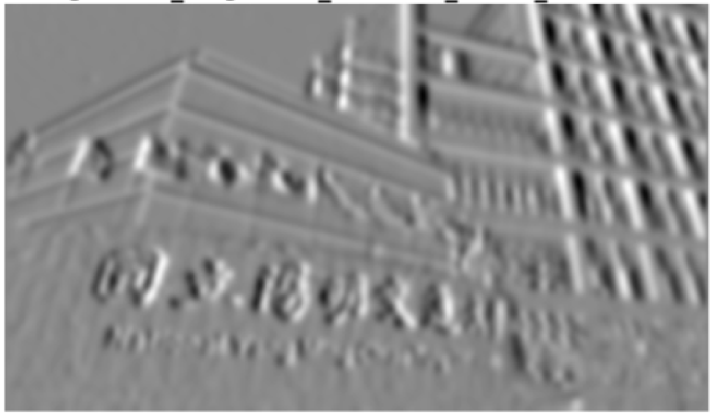
- **Compute Gradients**

gradient_magnitude_Gaussian_kernel_size=5



gradient_magnitude_direction_kernel_size=5

gradient_magnitude_Gaussian_kernel_size=10



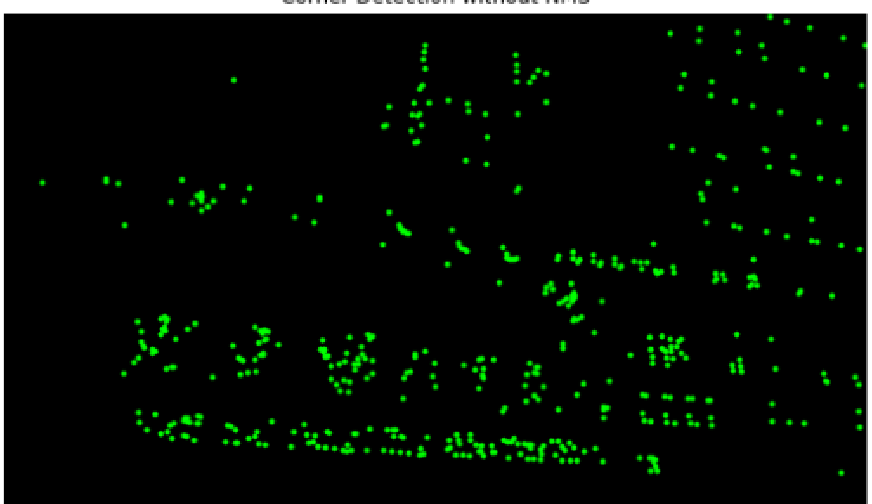
gradient_magnitude_direction_kernel_size=10

HW2: Harris Corner Detection

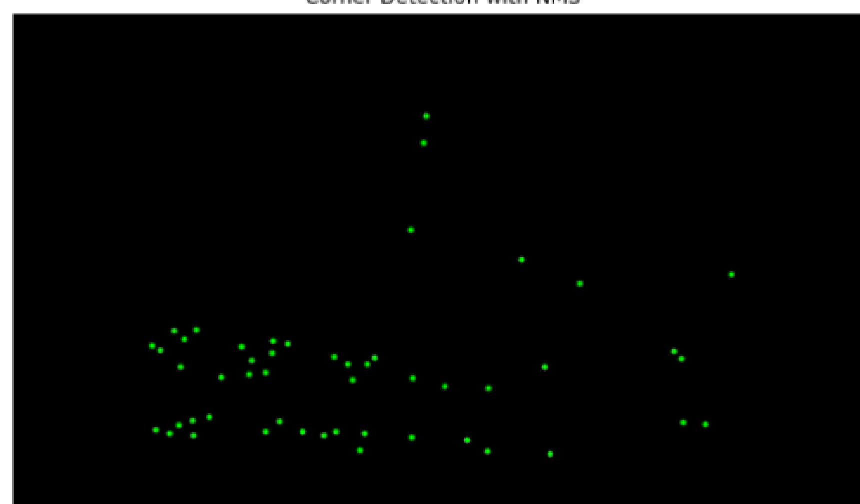
- **Non-Maximum Suppression**

Original Image

Corner Detection without NMS

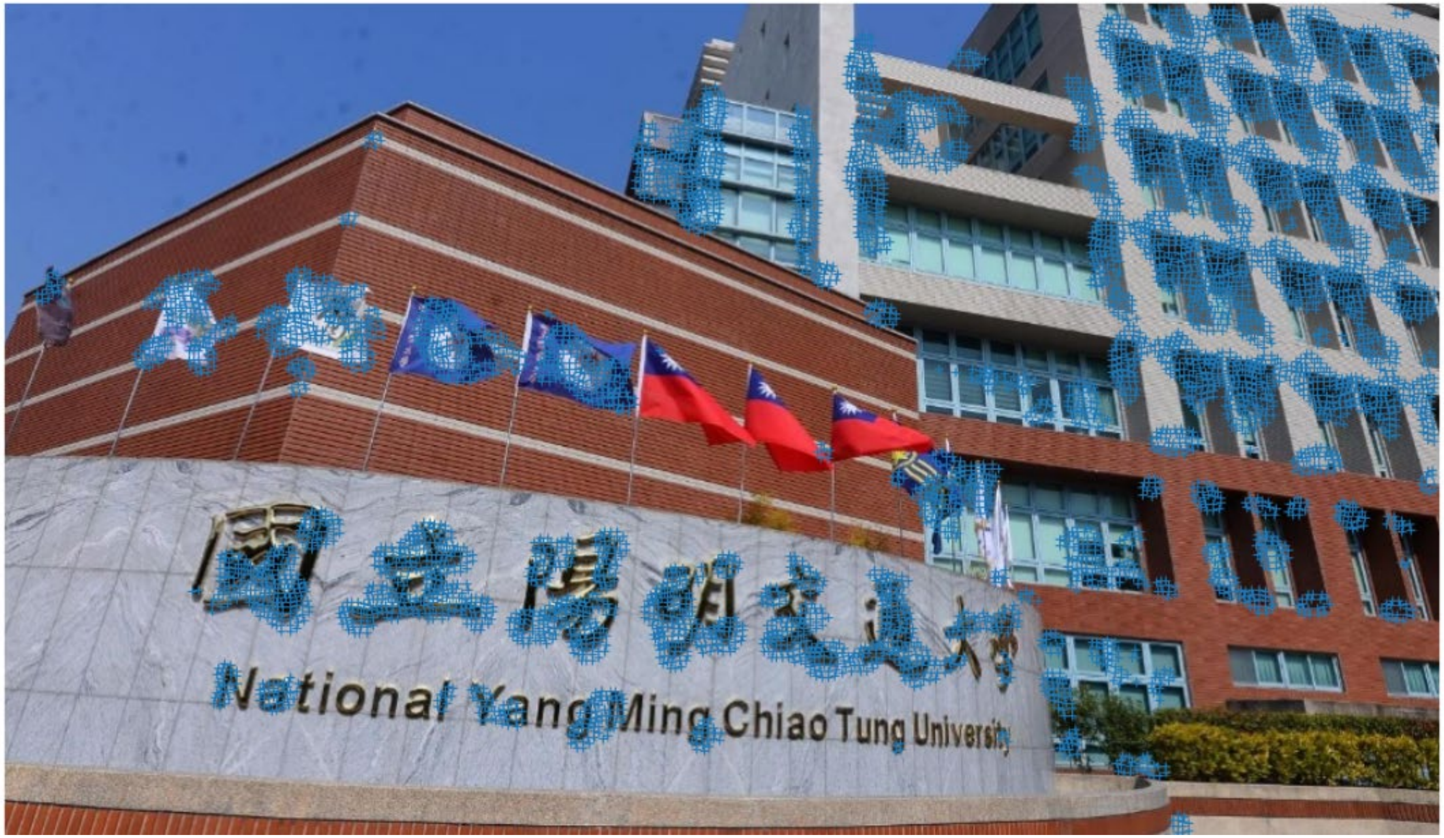


Corner Detection with NMS



HW2: Harris Corner Detection

- **Results**



HW2: Harris Corner Detection

A. Functions:

- **gaussian_smooth()**: filter images with Gaussian blur.
- **sobel_edge_detection()**: apply the Sobel filters to the blurred images and compute the magnitude and direction of gradient. (You should eliminate weak gradients by proper threshold.)
- **structure_tensor()**: use the gradient magnitude above to compute the structure tensor (second-moment matrix).
- **nms()**: perform non-maximal suppression on the results above along with appropriate threshold for corner detection.

HW2: Harris Corner Detection

B. Results:

- a. Original image
  - i. Gaussian smooth results: σ=5 and kernel size=5 and 10 (**2 images**)
  - ii. Sobel edge detection results
    - (1) magnitude of gradient (Gaussian kernel size=5 and 10) (**2 images**)
    - (2) direction of gradient (Gaussian kernel size=5 and 10) (**2 images**) (You can choose arbitrary color map to display)
  - iii. Structure tensor + NMS results (Gaussian kernel size=10)
    - (1) window size = 3x3 (**1 image**)
    - (2) window size = 30x30 (**1 image**)
- b. Final results of rotating (by 30°) original images (**1 image**)
- c. Final results of scaling (to 0.5x) original images (**1 image**)

HW2: Harris Corner Detection

C. Report:

- a. Discuss the results of blurred images and detected edges between different kernel sizes of Gaussian filter.
- b. Discuss the difference between 3x3 and 30x30 window sizes of structure tensor.
- c. Discuss the effect of non-maximal suppression.
- d. Discuss the results of rotated and scaled image. Is Harris detector rotation-invariant or scale-invariant? Explain the reason.

HW2: Harris Corner Detection

D. Notice:

- a. You should **NOT** use any functions which can get the result directly in each steps.
(cv2.Sobel, cv2.Laplacian, cv2.cornerHarris, skimg.feature.local_binary_pattern, etc.)
- b. Your code should display and output image results mentioned above.
- c. You should provide a **README** file about your execution instructions.

HW2: Harris Corner Detection

A. Functions:

gaussian_smooth(): filter images with Gaussian blur.

```
from scipy import ndimage

def gaussian_smooth(size, sigma=1):
    ########################################################################
    # TODO:                                                                #
    #   Perform the Gaussian Smoothing                                     #
    #   Input: window size, sigma                                          #
    #   Output: smoothing image                                            #
    ########################################################################

    ########################################################################
    #                                                                      #
    #                           End of your code                           #
    #                                                                      #
    ########################################################################
    return img

from scipy.ndimage.filters import convolve
img_filtered_K5 = convolve(img_Gray, gaussian_smooth(size=5,sigma=5))
img_filtered_K10 = convolve(img_Gray, gaussian_smooth(size=10,sigma=5))
```

HW2: Harris Corner Detection

A. Functions:

sobel_edge_detection(): apply the Sobel filters to the blurred images and compute the magnitude and direction of gradient.

```
def sobel_edge_detection(im):
    #########################################################################
    # TODO:                                                                 #
    #   Perform the sobel edge detection                                    #
    #   Input: image after smoothing                                        #
    #   Output: the magnitude and direction of gradient                     #
    #########################################################################

    #########################################################################
    #                                                                       #
    #                           End of your code                            #
    #                                                                       #
    #########################################################################
    return  (gradient_magnitude, gradient_direction)
```

HW2: Harris Corner Detection

A. Functions:

structure_tensor(): use the gradient magnitude above to compute the structure tensor (second-moment matrix).

```
def structure_tensor(gradient_magnitude, gradient_direction, k):
    ###########################################################################
    # TODO:                                                                   #
    #   Perform the cornermess response                                       #
    #   Input: gradient_magnitude, gradient_direction                         #
    #   Output: second-moment matrix of Structure Tensor                      #
    ###########################################################################

    ###########################################################################
    #                                                                         #
    #                          End of your code                               #
    #                                                                         #
    ###########################################################################
    return  StructureTensor
```

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

$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

HW2: Harris Corner Detection

A. Functions:

nms(): perform non-maximal suppression on the results above along with appropriate threshold for corner detection.

```
def NMS(harrisim,window_size=30,threshold=0.1):
    ########################################################################
    # TODO:                                                                #
    #   Perform the Non-Maximum Suppression                                #
    #   Input: Structure Tensor, window size, threshold                    #
    #   Output: filtered coordinators                                      #
    ########################################################################

    ########################################################################
    #                                                                      #
    #                           End of your code                           #
    #                                                                      #
    ########################################################################
    return filtered_coords
```

HW2: Harris Corner Detection

- **ReadMe Example:**

Homework 2 說明

1. 本次作業用到的Library: `os`, `cv2`, `numpy`, `matplotlib`, `scipy`

2. py 檔: `hw2.py` (執行檔), `Harris_Corner_Detection.py` (Function)

3. 若要執行程式, 產生圖片, 請輸入: `python hw2.py`

4. Function 包含: `gaussian_smooth`, `sobel_edge_detection`, `structure_tensor`, `NMS`, `rotate`

5. results 內容包含5個子Folder, 分別為:

> (1) `Gaussian smooth results`: 2張圖片, 分別是 Gaussian smooth results: σ=5 and kernel size=5 與 Gaussian smooth results: σ=5 and kernel size=10 images)
>
> (2) `Sobel edge detection results`: 4張圖片, 分別是 magnitude of gradient (Gaussian kernel size=5 and 10) (2 images) 與 direction of gradient (Gaussian kernel size=5 and 10) (2 images)
>
> (3) `Structure tensor + NMS results`: 2張圖片, 分別是 window size = 3x3 與 window size = 30x30
>
> (4) `Final results of rotating`: 1張圖片, 內容為 Final results of rotating (by 30°) original images
>
> (5) `Final results of scaling`: 1張圖片, 內容為 Final results of scaling (to 0.5x) original images

HW2: Harris Corner Detection

- **Submit File Format:**

results
Harris_Corner_Detection.py
hw2.py
original.jpg
README.md

Final results of rotating
Final results of scaling
Gaussian smooth results
Sobel edge detection results
Structure tensor + NMS results

3. 若要執行程式，產生圖片，請輸入：`python hw2.py`

A. Functions:

- **gaussian_smooth()**: filter images with Gaussian blur.
- **sobel_edge_detection()**: apply the Sobel filters to the blurred images and compute the magnitude and direction of gradient. (You should eliminate weak gradients by proper threshold.)
- **structure_tensor()**: use the gradient magnitude above to compute the structure tensor (second-moment matrix).
- **nms()**: perform non-maximal suppression on the results above along with appropriate threshold for corner detection.

IIAI30013

Computer Vision

HW2: Harris Corner Detection

Q & A

Instructor: YuanFu Yang

yfyangd@nycu.edu.tw