

Computer Graphics

실습 4.

2020. 10. 07

박 화 종

aqkrghkwhd@naver.com

실습 소개

- 과목 홈페이지

- 충남대학교 사이버 캠퍼스 (<http://e-learn.cnu.ac.kr>)

- TA 연락처

- 박화종
- 공대 5호관 506호 컴퓨터비전 연구실
- aqkrghkwhd@naver.com

- 실습 튜터

- 최수민(00반)
- eocjstnals12@naver.com
- 신준호(01반)
- wnsgh578@naver.com

목 차

- 2주차 과제 리뷰
- 3주차 과제 추가 설명
- 실습
 - Harris Corner Detection
 - Dilation
- 과제
 - Harris Corner Detection

2주차 과제 리뷰

- 노이즈가 있는 이미지의 노이즈 제거 및 시간 측정



original



average filter 사용



Gaussian filter 사용

2주차 과제 리뷰

```
def my_filtering(src, filter):
    (h, w) = src.shape
    (f_h, f_w) = filter.shape
    pad_img = my_padding(src, filter)
    dst = np.zeros((h, w))

    #####
    # TODO #
    # filtering 구현 #
    # 4중 for문을 이용해 구현할것! #
    #####

    for row in range(h):
        for col in range(w):
            sum = 0
            for m_row in range(f_h):
                for m_col in range(f_w):
                    sum += pad_img[row + m_row, col+m_col] * filter[m_row, m_col]
            dst[row, col] = sum

    return dst
```

2주차 과제 리뷰

```
def my_average_filter(src, fshape, verbose=False):
    (h, w) = src.shape
    if verbose:
        print('average filtering')

    #####
    # TODO #
    # average filter 생성 #
    #####

    filter = np.ones(fshape)
    filter = filter / (fshape[0] * fshape[1])

    if verbose:
        print('<average filter> - shape:', fshape)
        print(filter)

    dst = my_filtering(src, filter)
    return dst
```

2주차 과제 리뷰

```
def my_get_Gaussian_filter(fshape, sigma=1):
    (f_h, f_w) = fshape

    #####
    # TODO #
    # gaussian filter 생성 #
    #####

    y, x = np.mgrid[-(f_h // 2):(f_h // 2) + 1, -(f_w // 2):(f_w // 2) + 1]
    #2차 gaussian mask 생성
    filter_gaus = 1 / (2 * np.pi * sigma**2) * np.exp(-((x**2 + y**2)/(2 * sigma**2)))
    #mask의 총 합 = 1
    filter_gaus /= np.sum(filter_gaus)
    return filter_gaus
```

3주차 과제 추가 설명

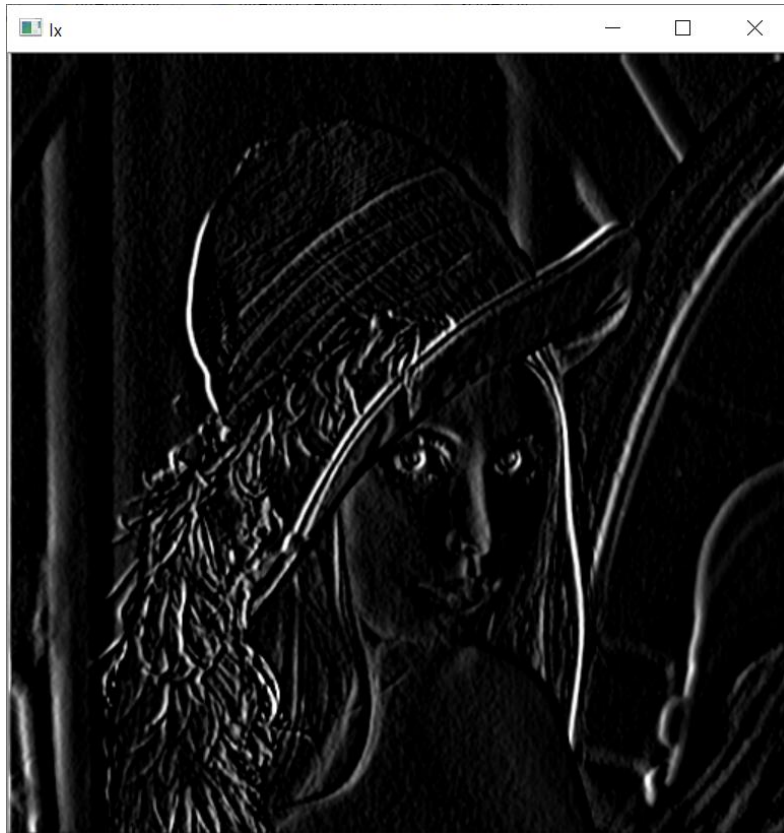
- 27page의 이미지가 sobel + Gaussian 을 사용한 결과
- 38page는 제가 설명을 잘못했습니다.. 죄송합니다.
 - 38page는 그냥 double thresholding 을 진행하면 before에서 after로 이런식으로 바뀐다고 설명을 했어야 했는데 잘못 설명했습니다. 그냥 double thresholding 참고용 페이지입니다. (없는 페이지다 라고 생각하시면 됩니다.)
- 참고 : 저는 주로 과제 설명 맨 첫 페이지에 어떻게 결과가 나와야 하는지를 보여주고 과제 설명을 시작합니다.

3주차 과제 추가 설명

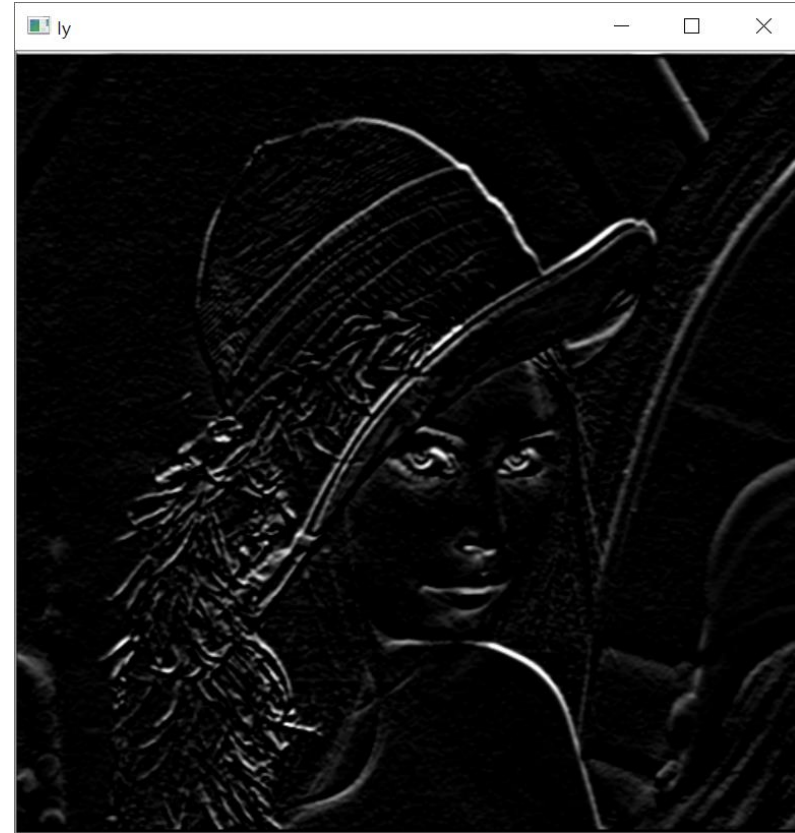
- hyperparameter setting
 - Sobel filter size = 3x3
 - Gaussian filter size = 3x3
 - Gaussian filter sigma = 1
 - Hightreshold value = 46 : 이건 cv2.threshold함수 사용시 자동으로 설정됨

3주차 과제 추가 설명

- Canny Edge Detection



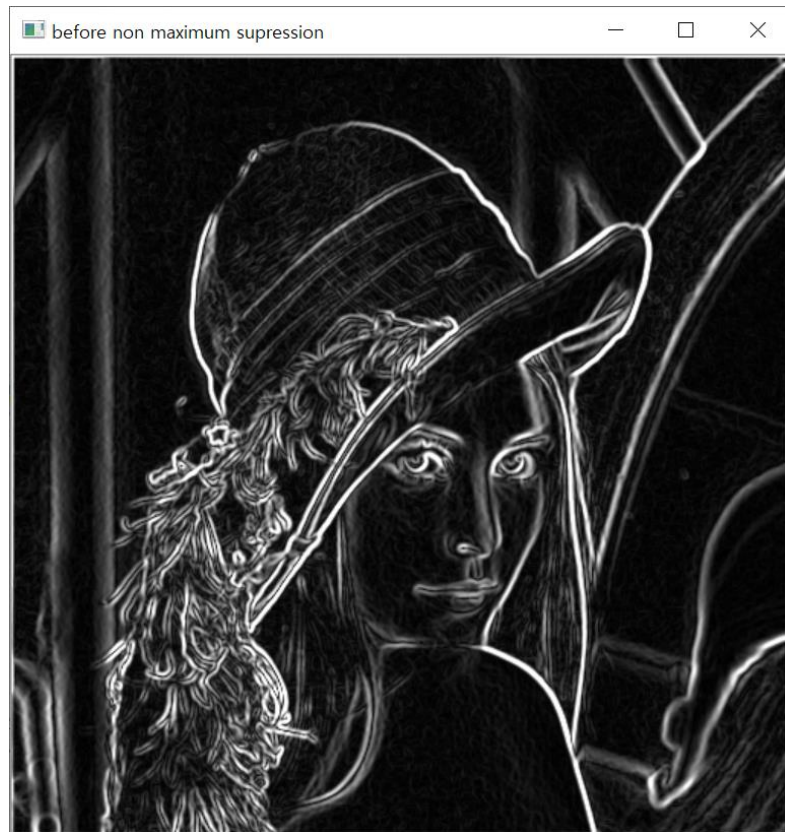
lx



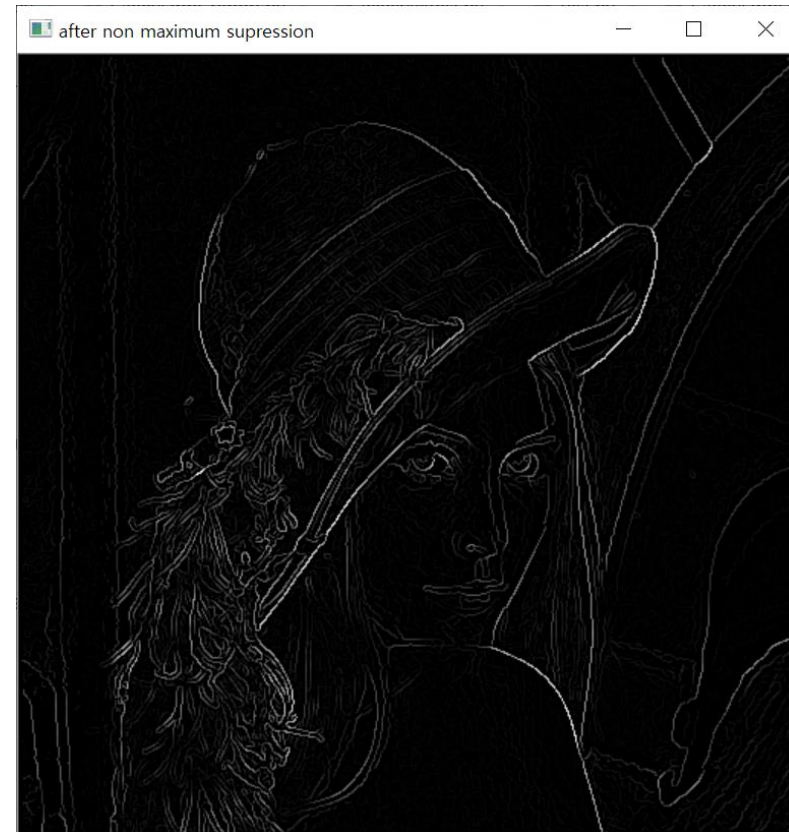
ly

3주차 과제 추가 설명

- Canny Edge Detection



before non-maximum



after non-maximum

3주차 과제 추가 설명

- Canny Edge Detection

실제 크기 & 최종 결과

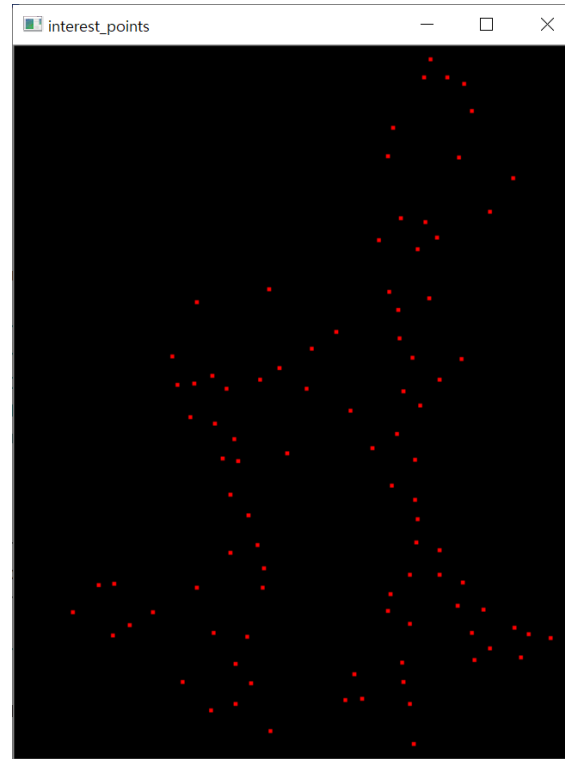


실습

- Harris Corner Detection



original



interest points



harris corner

실습

Harris Detector [Harris88]

67/108

- Harris Corner Detection

Covariance matrix

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

$$\det M = \lambda_1 \lambda_2$$

$$\text{trace } M = \lambda_1 + \lambda_2$$

1. Image derivatives
(optionally, blur first)



2. Square of derivatives



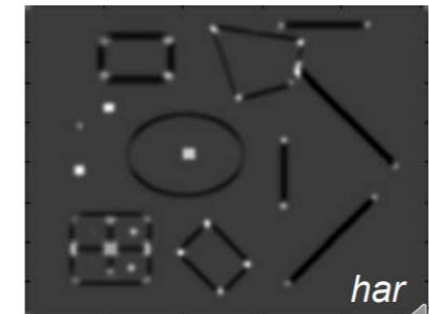
3. Gaussian filter $g(\sigma_l)$



4. Cornerness function – both eigenvalues are strong

$$\begin{aligned} har &= \det[\mu(\sigma_I, \sigma_D)] - \alpha [\text{trace}(\mu(\sigma_I, \sigma_D))]^2 = \\ &= g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha [g(I_x^2) + g(I_y^2)]^2 \end{aligned}$$

5. Non-maxima suppression



출처 : 교수님 이론 pdf 67page

실습

• Harris Corner Detection

- Sobel filter 적용 – I_x 와 I_y 구하기
- $(I_x)^2$ $(I_y)^2$ $I_x \cdot I_y$ 구하기
- $(I_x)^2$ $(I_y)^2$ $I_x \cdot I_y$ 에 Gaussian filter 적용하기
- M 완성
- $\det M$, $\text{trace } M$ 구하기
- 특정 부분이 얼마나 코너인지 계산
- 근처에 코너가 많은 경우 코너 수 줄이기

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

M을 구하기 위해선 I_x 와 I_y 가 필요

$$\det M = \lambda_1 \lambda_2$$

$$\text{trace } M = \lambda_1 + \lambda_2$$

실습

- Harris Corner Detection

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

```
"""
```

Harris Corner Detector in OpenCV

OpenCV has the function `cv.cornerHarris()` for this purpose. Its arguments are:

`img` - Input image. It should be grayscale and float32 type.

`blockSize` - It is the size of neighbourhood considered for corner detection

ksize - Aperture parameter of the Sobel derivative used.

`k` - Harris detector free parameter in the equation.

```
"""
```


실습

• Harris Corner Detection

```
def main():
    src = cv2.imread('../image/zebra.png')
    gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY).astype(np.float32)

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

    cv2.imshow('original', src)
    har = ((dst - np.min(dst)) / np.max(dst - np.min(dst)) * 255 + 0.5).astype(np.uint8)
    cv2.imshow('har', har)

    dst[dst < 0.01 * dst.max()] = 0 #thresholding

    interest_points = np.zeros((dst.shape[0], dst.shape[1], 3))
    interest_points[dst != 0] = [0, 0, 255]
    cv2.imshow('interest_points', interest_points)

    src[dst != 0] = [0, 0, 255]
    cv2.imshow('harris', src)

    cv2.waitKey()
    cv2.destroyAllWindows()

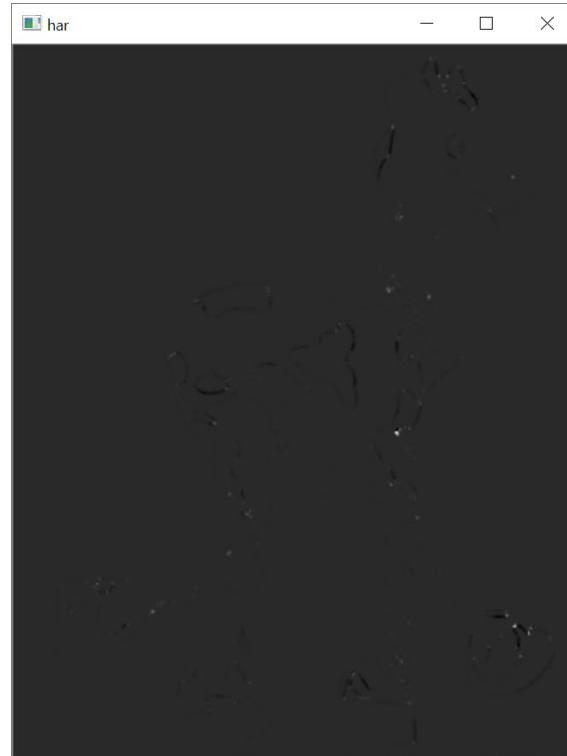
if __name__ == '__main__':
    main()
```

실습

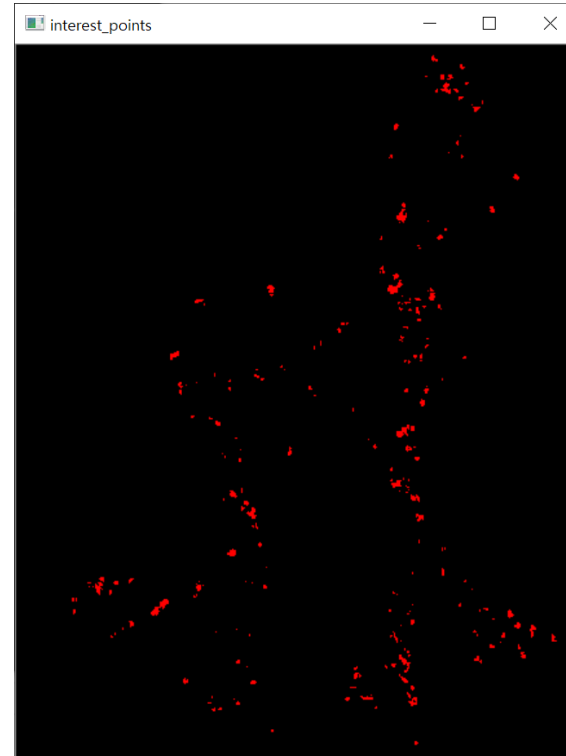
- Harris Corner Detection



original



cv2.cornerHarris
0 ~ 255



interest points



Harris Corner Detection

실습

• Harris Corner Detection

```
def main():  
    src = cv2.imread('../image/zebra.png')  
    gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY).astype(np.float32)  
  
    dst = cv2.cornerHarris(gray, 3, 3, 0.04)  
  
    cv2.imshow('original', src)  
    har = ((dst - np.min(dst)) / np.max(dst - np.min(dst)) * 255 + 0.5).astype(np.uint8)  
    cv2.imshow('har', har)  
  
    dst[dst < 0.01 * dst.max()] = 0 #thresholding  
    dst = find_local_maxima(dst, 21)  
  
    interest_points = np.zeros((dst.shape[0], dst.shape[1], 3))  
    interest_points[dst != 0] = [0, 0, 255]  
    cv2.imshow('interest_points', interest_points)  
  
    src[dst != 0] = [0, 0, 255]  
    cv2.imshow('harris', src)  
  
    cv2.waitKey()  
    cv2.destroyAllWindows()  
  
if __name__ == '__main__':  
    main()
```

실습

- Harris Corner Detection

```
def find_local_maxima(src, ksize):  
    (h, w) = src.shape  
    pad_img = np.zeros((h + ksize, w + ksize))  
    pad_img[ksize // 2:h + ksize // 2, ksize // 2:w + ksize // 2] = src  
    dst = np.zeros((h, w))  
  
    for row in range(h):  
        for col in range(w):  
            max_val = np.max(pad_img[row: row + ksize, col:col + ksize])  
            if max_val == 0:  
                continue  
            if src[row, col] == max_val:  
                dst[row, col] = src[row, col]  
  
    return dst
```

실습

- Harris Corner Detection



original



cv2.cornerHarris
0 ~ 255



interest points



Harris Corner Detection

실습

• Harris Corner Detection

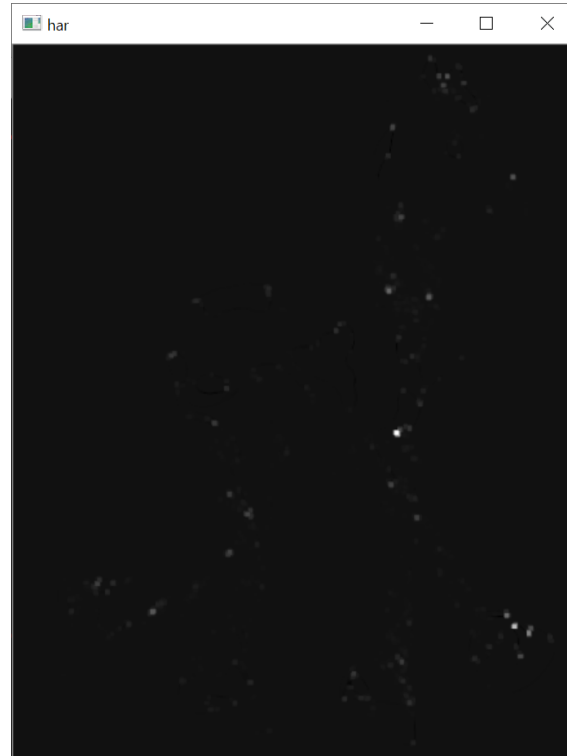
```
def main():  
    src = cv2.imread('../image/zebra.png')  
    gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY).astype(np.float32)  
  
    dst = cv2.cornerHarris(gray, 3, 3, 0.04)  
  
    dst = cv2.dilate(dst, None)  
  
    cv2.imshow('original', src)  
    har = ((dst - np.min(dst)) / np.max(dst - np.min(dst)) * 255 + 0.5).astype(np.uint8)  
    cv2.imshow('har', har)  
  
    dst[dst < 0.01 * dst.max()] = 0 #thresholding  
    dst = find_local_maxima(dst, 21)  
  
    interest_points = np.zeros((dst.shape[0], dst.shape[1], 3))  
    interest_points[dst != 0] = [0, 0, 255]  
    cv2.imshow('interest_points', interest_points)  
  
    src[dst != 0] = [0, 0, 255]  
    cv2.imshow('harris', src)  
  
    cv2.waitKey()  
    cv2.destroyAllWindows()
```

실습

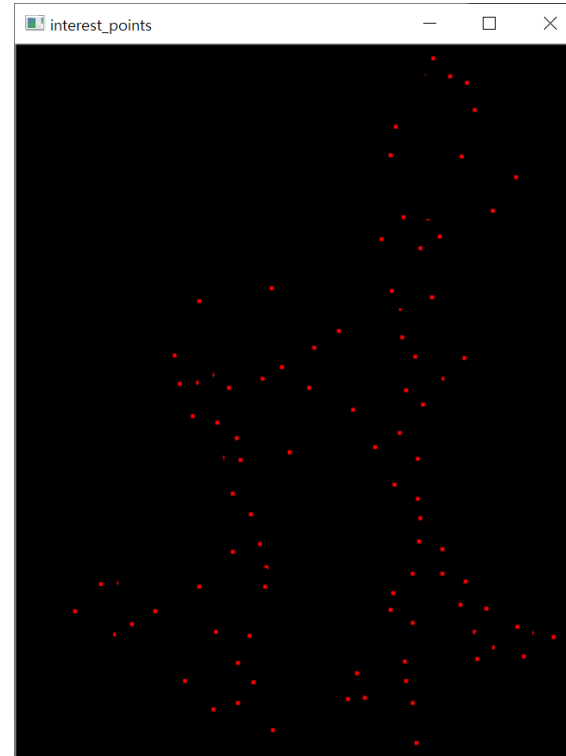
- Harris Corner Detection



original



cv2.cornerHarris
0 ~ 255



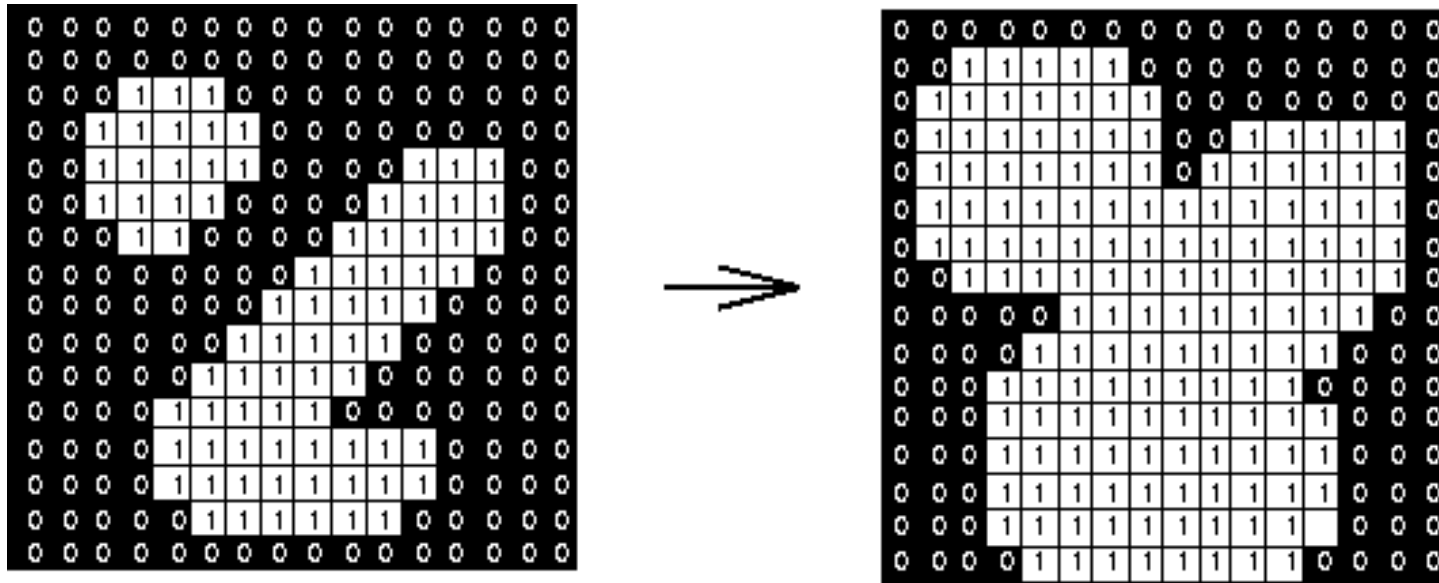
interest points



Harris Corner Detection

실습

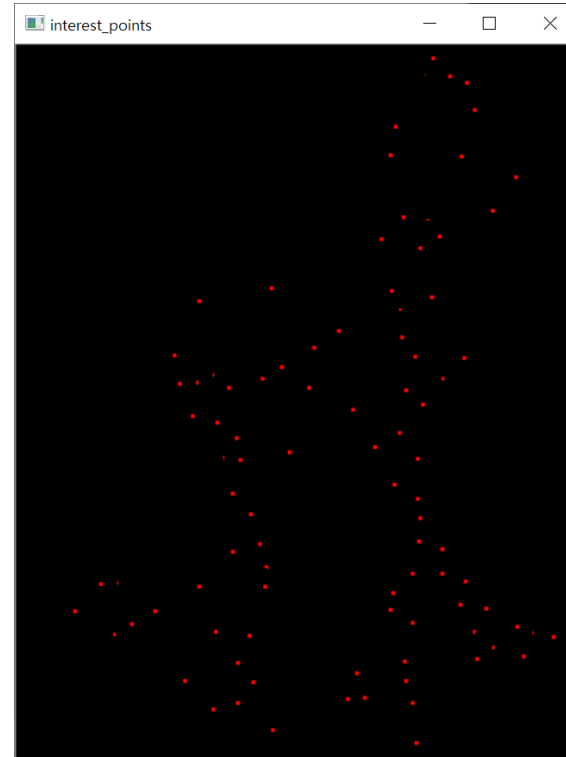
- **cv2.dilate(src, kernel)**
 - 이미지를 팽창시킨다.
 - Kernel 범위 내의 최댓값으로 현재 픽셀 값을 대체



3x3 kernel을 사용한 dilate 수행 예시

실습

- **cv2.dilate(src, kernel)**
 - 이미지를 팽창시킨다.
 - Kernel 범위 내의 최댓값으로 현재 픽셀 값을 대체

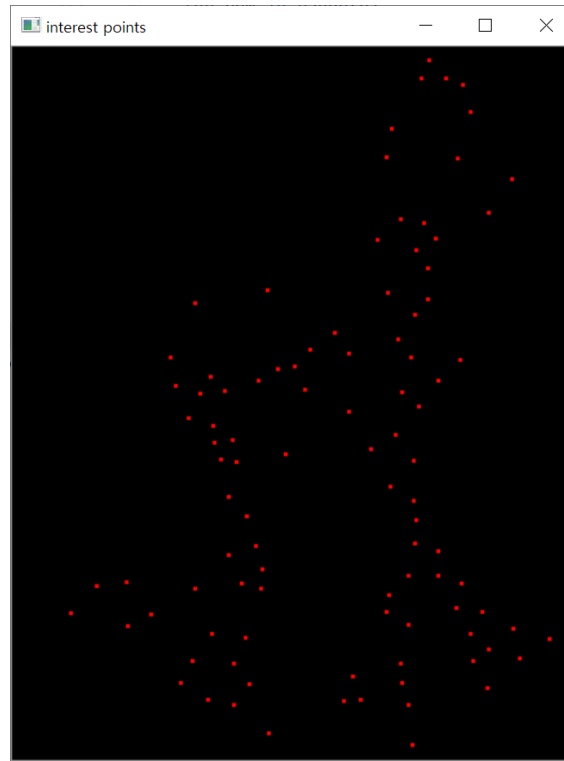


과제

- Harris Corner Detection 구현하기



original



my interest points



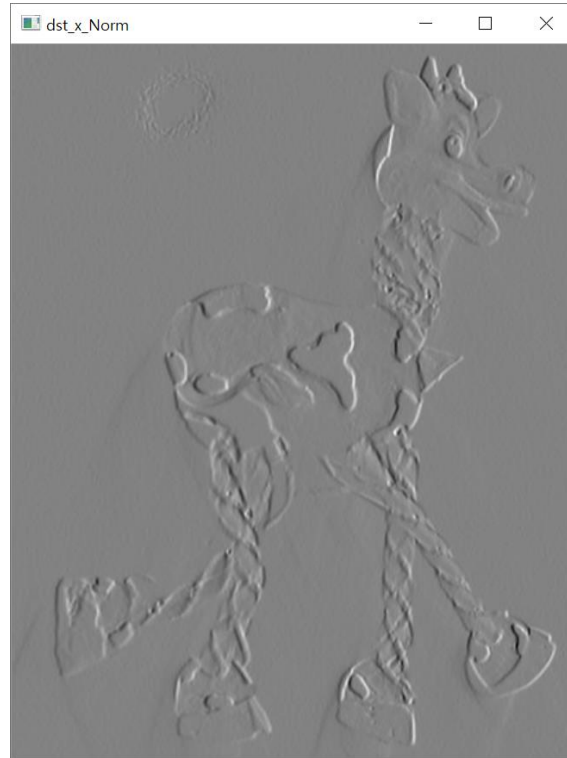
my harris corner detection

과제

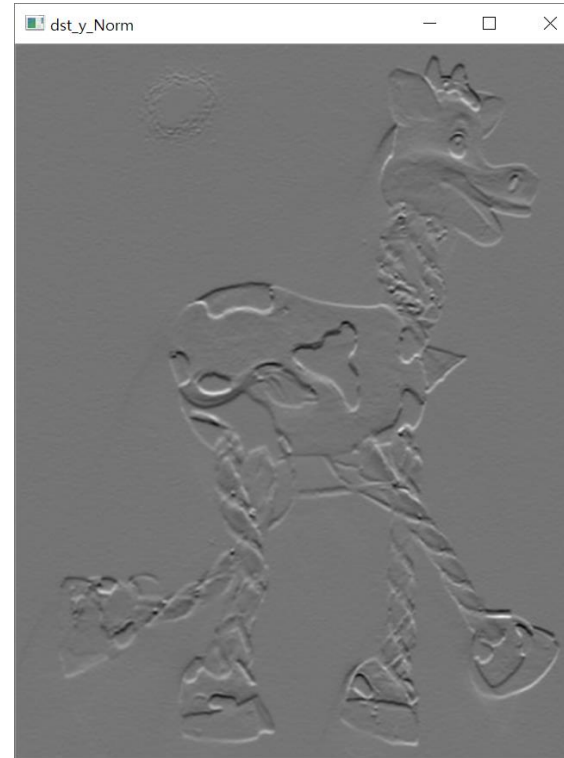
• Harris Corner Detection 구현하기



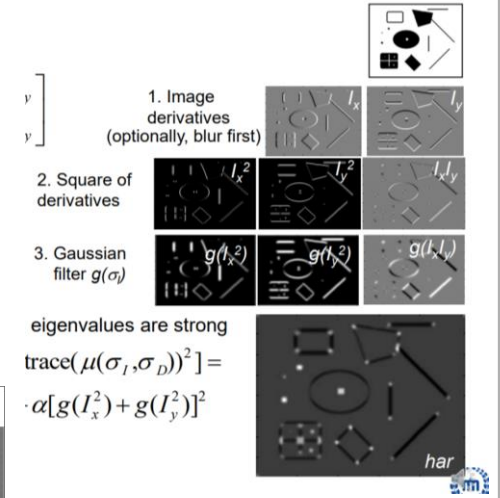
original



lx



ly



과제

• Harris Corner Detection 구현하기



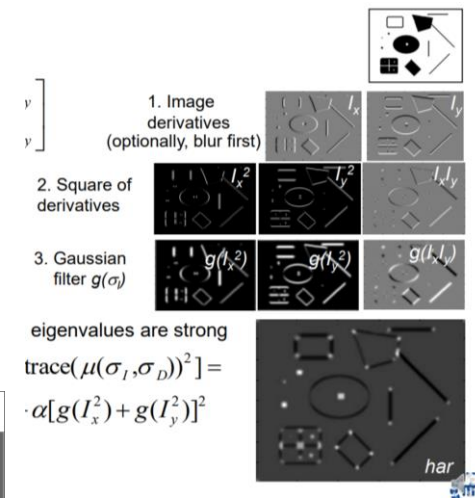
IxIx



IyIy



IxIy



과제

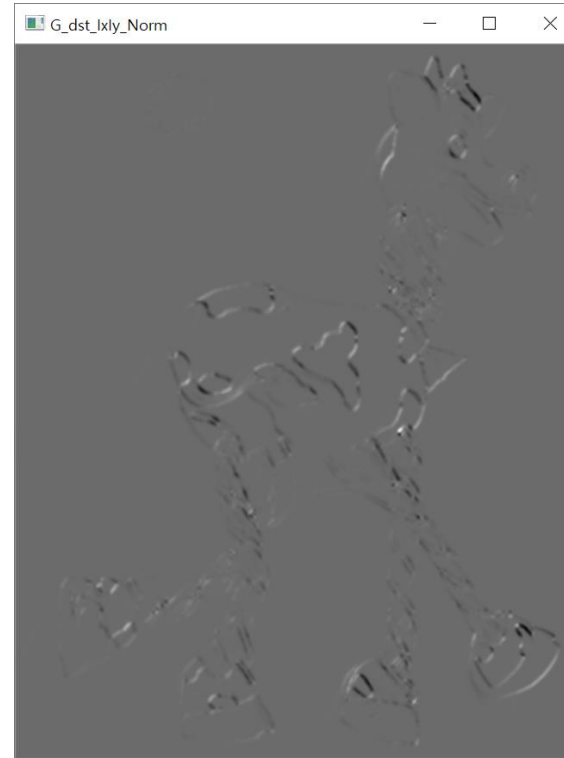
• Harris Corner Detection 구현하기



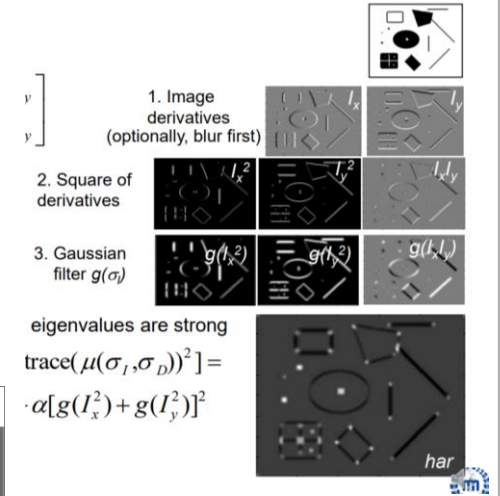
G_lxIx



G_lyIy



G_lxly



과제

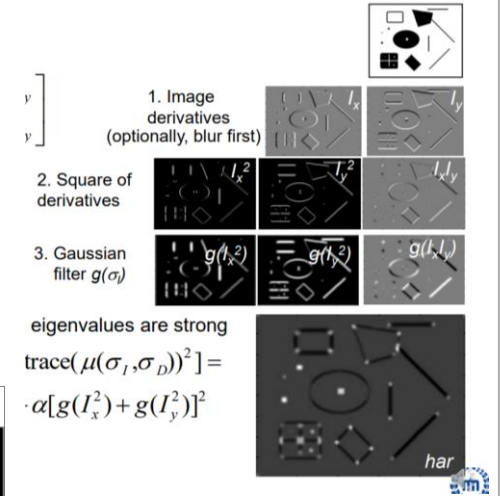
• Harris Corner Detection 구현하기



har before threshold

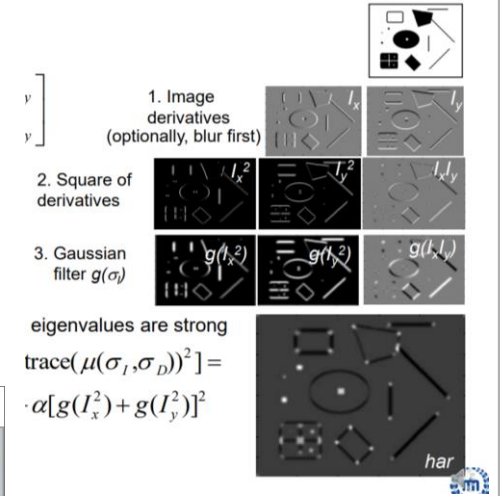


har after threshold

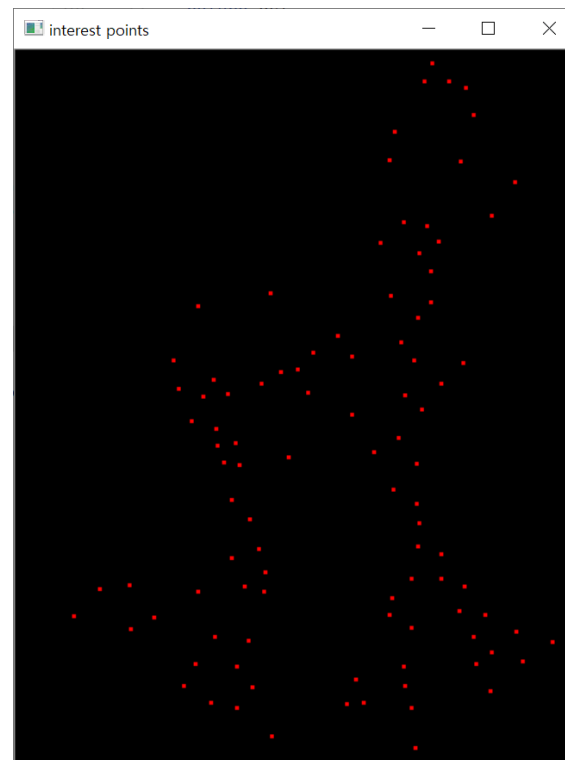


과제

• Harris Corner Detection 구현하기



original



my interest points



my harris corner detection

과제 - 구현

- 제공된 코드가 제대로 동작하도록 구현하기

- calc_derivatives 함수
- HarrisDetector 함수

- 사용한 hyperparameter

- sobel filter size = 3x3
- Gaussian filter size = 3x3
- Gaussian filter sigma = 1

- har = > 빨간색으로 표시
- 어떤 식을 사용해도 상관없음
- alpha = 0.04

- threshold rate = 0.01

- local maxima filter size = 21

$$\det M = \lambda_1 \lambda_2$$

$$\text{trace } M = \lambda_1 + \lambda_2$$

4. Cornerness function – both eigenvalues are strong

$$har = \det[\mu(\sigma_I, \sigma_D)] - \alpha [\text{trace}(\mu(\sigma_I, \sigma_D))]^2 =$$

$$g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha [g(I_x^2) + g(I_y^2)]^2$$



과제 - 보고서

- 보고서

- 내용:

- 이름, 학번, 학과
 - 구현 코드: 구현한 코드
 - 코드 설명: 구현한 코드에 대한 설명(설명은 1page를 넘기지 말 것, 1줄이어도 상관없음)
 - 이미지: 과제 첫 페이지(26page) 참고하여 이미지 3개 첨부
 - 느낀 점 : 결과를 보고 느낀 점, 혹은 과제를 하면서 어려웠던 점 등
 - 과제 난이도: 개인적으로 생각하는 난이도 (과제가 너무 쉬운 것 같다 등)

- .pdf 파일로 제출 (이 외의 파일 형식일 경우 감점)

- 파일 이름:

- [CG]20xxxxxxx_이름_n주차_과제.pdf

과제

- **제출 기한**

- 10월 20일 23시 59분까지 (최대 점수 10점)

- **추가 제출 기한**

- 10월 27일 23시 59분까지 (최대 점수 4점, 과제 총점 계산 후 -6점)
- 10월 28일 00시 00분 이후 (점수 0점)

- **채점**

- 구현을 못하거나(잘못 구현하거나) 보고서 내용이 빠진 경우 감점
- 아무것도 구현하지 못해도 과제 제출하면 기본점수 있음
- 다른 사람의 코드를 copy해서 제출시 보여준 사람, copy한 사람 둘 다 0점
- 내장함수 사용시 감점(내장함수를 사용해도 된다고 말 한 것 제외) – cv2.cornerHarris 사용 불가

- **제출 파일**

- 아래의 파일을 압축해서 [CG]20xxxxxxx_이름_n주차_과제.zip로 제출
 - .py 파일 전부
 - .pdf 보고서 파일

QnA