

# Computer Vision HW1: Harris Corner Detector

2013011446 손윤하

## 1. Harris corner detector implementation

먼저 Harris corner detection을 수행하는 함수를 정의했다.

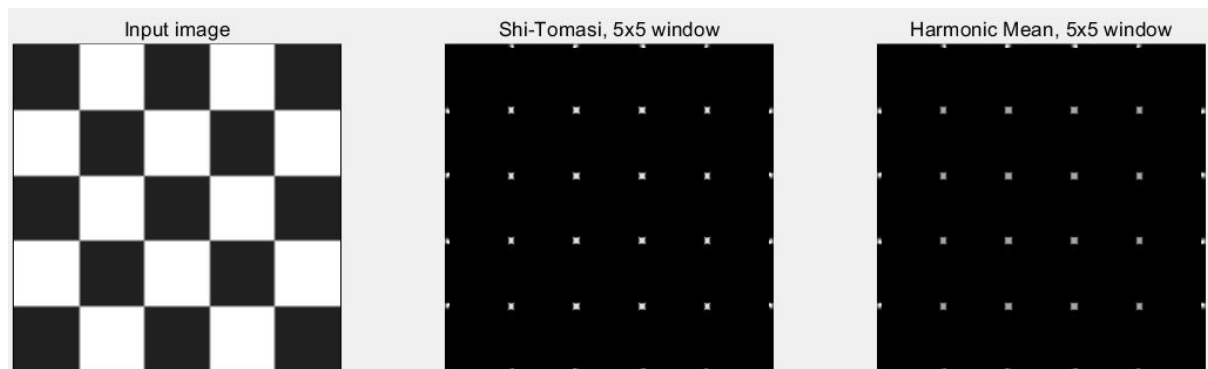
**Harris(image, window size, detector)**

- image: 디텍션을 적용할 이미지
- window size: 윈도우 크기 (3, 5, 9 등)
- detector: 'harmonic' 또는 'shi'
- detection 결과를 반환한다.

### Calculation of edge maps

$I_x$ 와  $I_y$ 를 계산하기 위해서(edge map)  $x$ ,  $y$  각 방향으로의 그래디언트를 적용시켰다. 각  $x$  또는  $y$  방향으로 ‘얼마나 변했는지’를 통해서 edge를 나타낼 수 있다. 결과는 예제에서 제시된  $[-1 \ 0 \ 1]$  필터와 거의 동일한데,  $[-1 \ 0 \ 1]$  자체가 그래디언트를 근사하기 위한 계수이기 때문이다(approximated by finite differences).

### 1.1 Checkerboard



### Handling boundary pixels

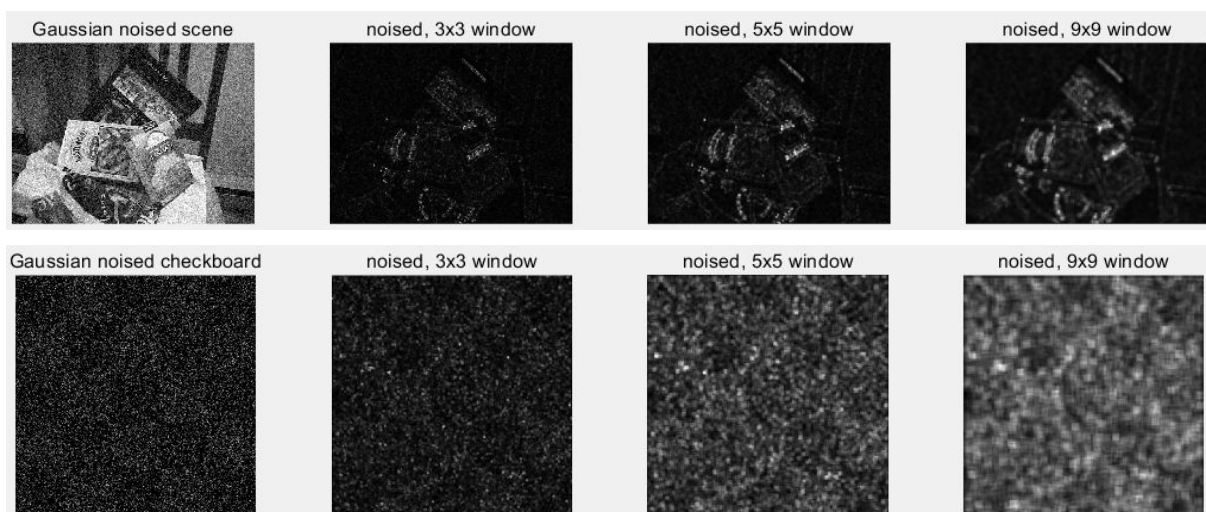
MATLAB의 `conv2`와 `imgradientxy` 함수에서 내부적으로 패딩 처리를 해주므로, 경계 픽셀에 대한 처리를 명시적으로 하지 않았다. (공식 문서 인용 - When applying the gradient operator at the boundaries of the image, values outside the bounds of the image are assumed to equal the nearest image border value.)

## 1.2 Test images



### Noised image

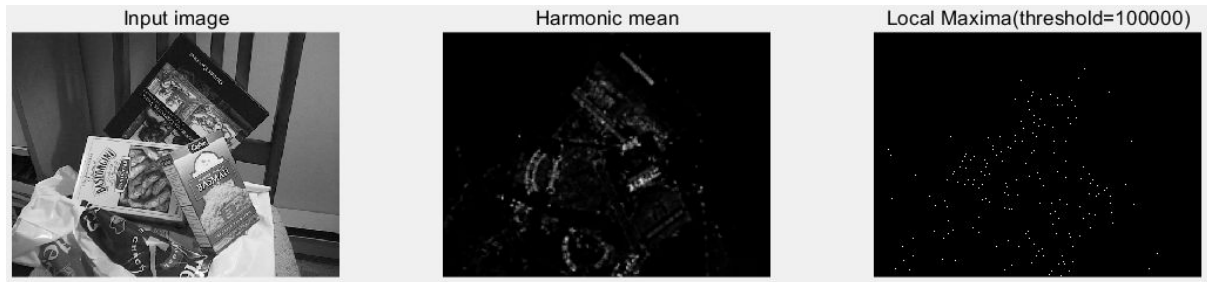
Matlab에서 제공하는 Gaussian 노이즈 함수(standard deviation = 3/255)를 적용했다. 노이즈 영상은 모서리 검출의 정확도가 크게 떨어졌다. 복잡한 영상에서는 변화가 눈에 잘 띄지 않지만, 보다 단순한 체커보드에서 노이즈의 효과가 확실히 드러난다.



### 1.3 Local maxima of harmonic mean values

가장 단순한 방법으로, 전체 이미지에 대해 5x5 윈도우를 움직이면서, 윈도우의 중심 픽셀이 그 윈도우 내에서 가장 큰 값일 때 local maxima라고 판단하도록 구현했다.

또 다른 방법으로, 그래디언트를 이용하여 어느 해당 픽셀이 변하는 각도를 계산하고( $\text{atan}(gy/gx)$  이용), 이를 양자화해서 이웃 8픽셀들중 두 픽셀을 선택, 크기를 비교하여 local maxima를 찾는 것도 가능할 것이다. (이웃 두 픽셀보다 클 때 극대)



(결과 이미지 리사이징으로 local maxima가 깨져 보임)

