Lecture note 4 : Feature extraction

프로젝트 기반 딥러닝 이미지처리
한국인공지능아카데미 x Hub Academy

강사 : 김형욱 (hyounguk1112@gmail.com)

Agenda

1. Review : Image feature
2. Edge and corner detection
3. SIFT algorithm
4. Contour detection

Review : image feature

Image feature의 정의와 주요 개념

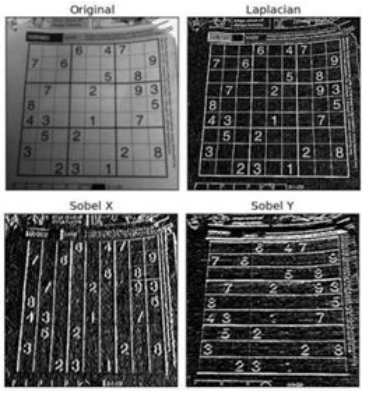
영상 특징(Image features) : 영상들간의 유사성을 계산하기 위한 영상의 패턴을 의미

키포인트(Keypoints) : 영상 특징이 위치한 영상 내 좌표 또는 스케일 정보를 의미

영상표현자(Descriptor) : 영상 특징에 대한 정량적 표현. 주로 값(스칼라), 벡터, 행렬, 히스토그램 등으로 표현됨.

Primary features

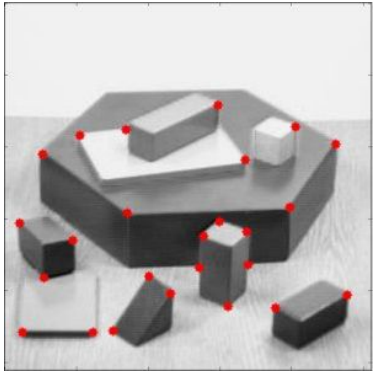
- Edges
- Corners
- Blobs



Edge

How about averaging values of pixels?

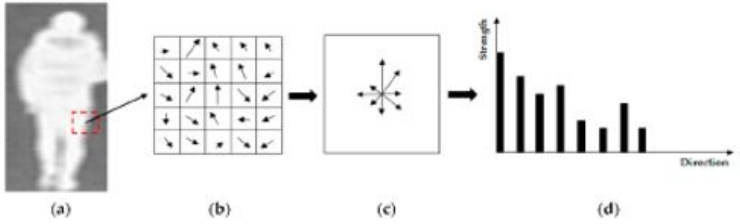
It’s ok! But better ones could exist for most cases


Corner

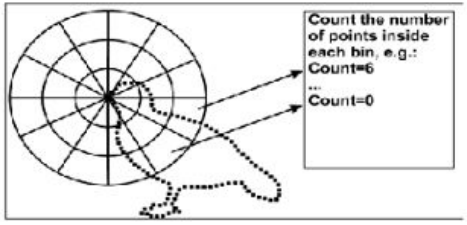
What is good feature?

- Well-designed features

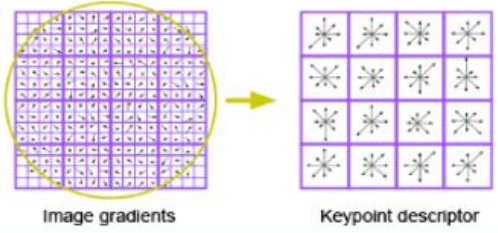
HoG (Histogram of Gradients)



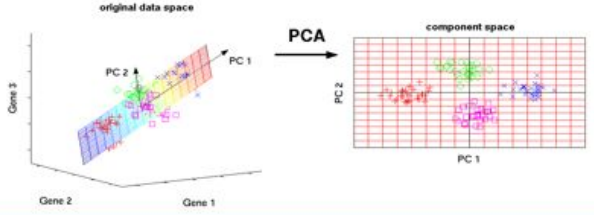
Shape Context



SIFT (Scale-Invariant Feature Transform)



PCA (Principal Components Analysis)



Edge and corner detection

Image gradient : Sobel filter

Sobel filter는 영상의 수직/수평 방향 edge를 검출하기 위한 필터
컨볼루션 연산이 수행되는 지역(Region)의 좌우측 또는 상하의 **픽셀값 변화**를 계산함.

Original image

-1	0	1
-2	0	2
-1	0	1

G_x

1	2	1
0	0	0
-1	-2	-1

G_y

Result of Sobel operator

Image gradient

In practice, it is common to use:

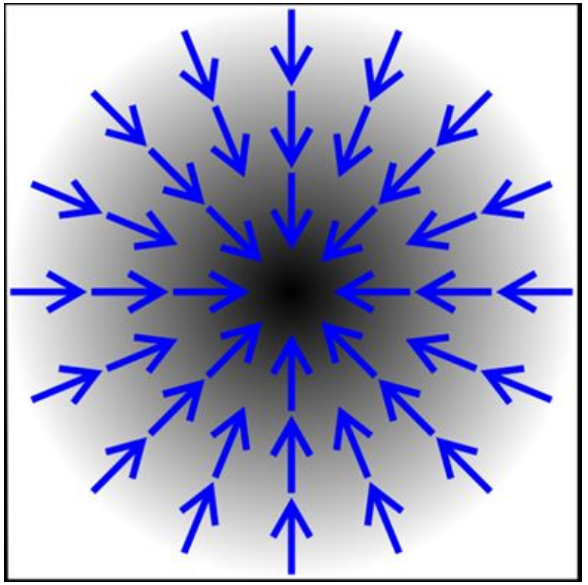
$$g_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad g_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Magnitude:

$$g = \sqrt{g_x^2 + g_y^2}$$

Orientation:

$$\Theta = \tan^{-1}\left(\frac{g_y}{g_x}\right)$$



두 종류의 Sobel 필터는 영상에서 각각 수직/수평 방향의 픽셀값 변화(미분값)를 계산할 수 있고, 이를 통해 Gradient field를 계산할 수 있음.

Image gradient

In practice, it is common to use:

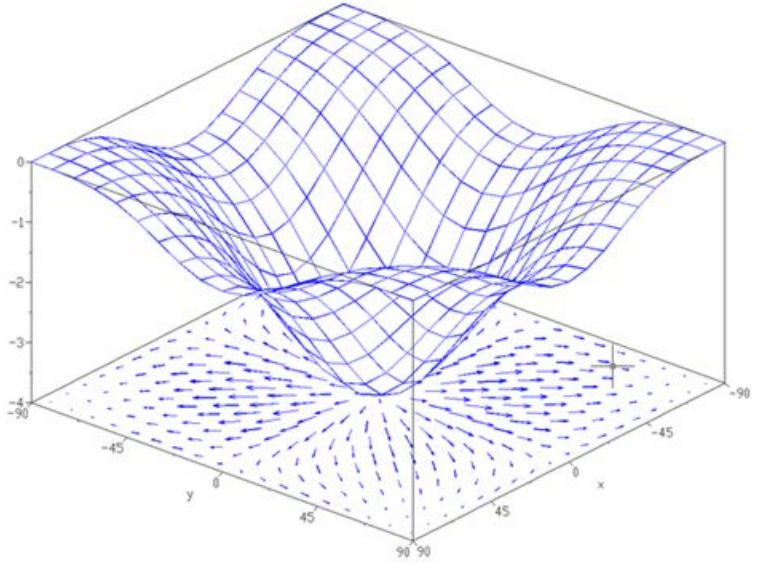
$$g_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \qquad g_y = \begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Magnitude:

$$g = \sqrt{g_x^2 + g_y^2}$$

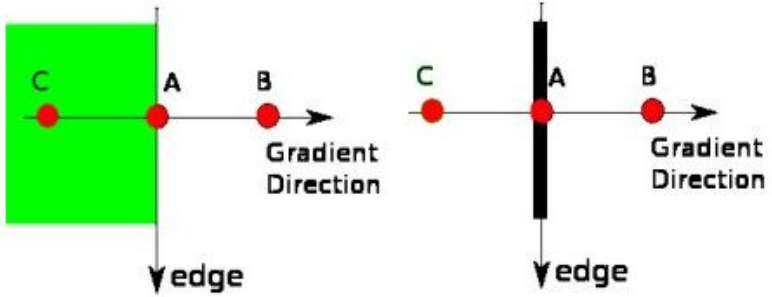
Orientation:

$$\Theta = \tan^{-1}\left(\frac{g_y}{g_x}\right)$$

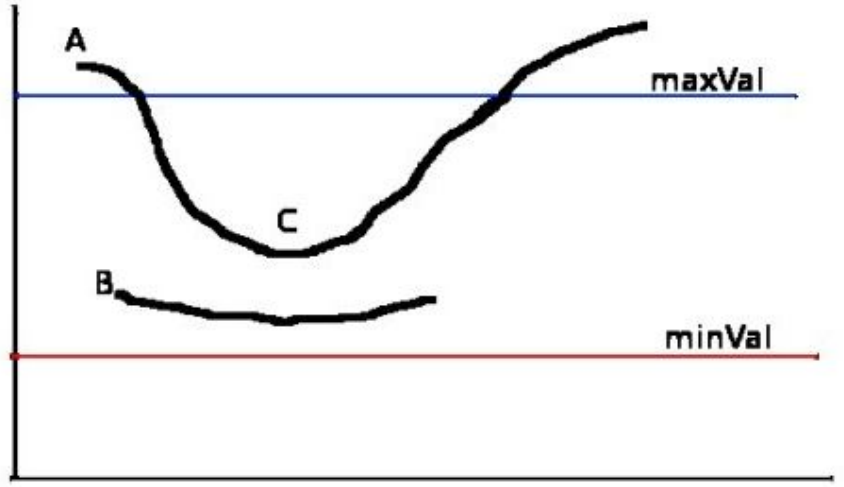


두 종류의 Sobel 필터는 영상에서 각각 수직/수평 방향의 픽셀값 변화(미분값)를 계산할 수 있고, 이를 통해 Gradient field를 계산할 수 있음.

Canny edge detection

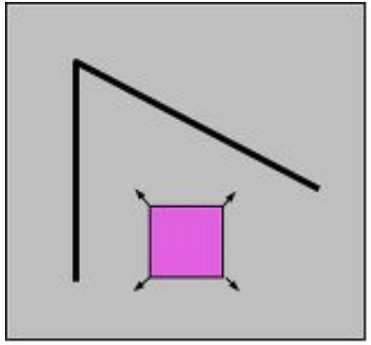


Gradient의 크기가 주변과 비교하여 최대값이 아닌 Pixel은 Gradient를 0로 변경하기

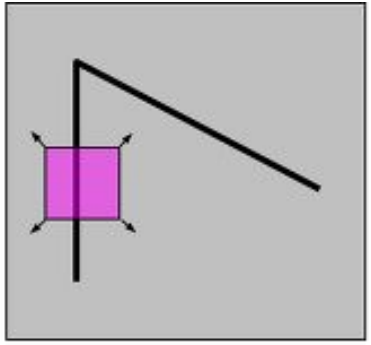


Hysteresis thresholding :
maxVal와 minVal을 통한 edge 판별

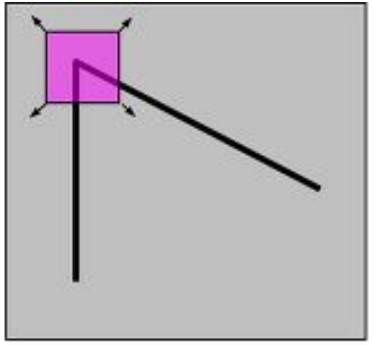
Harris corner detection



"flat" region:
no change in all directions

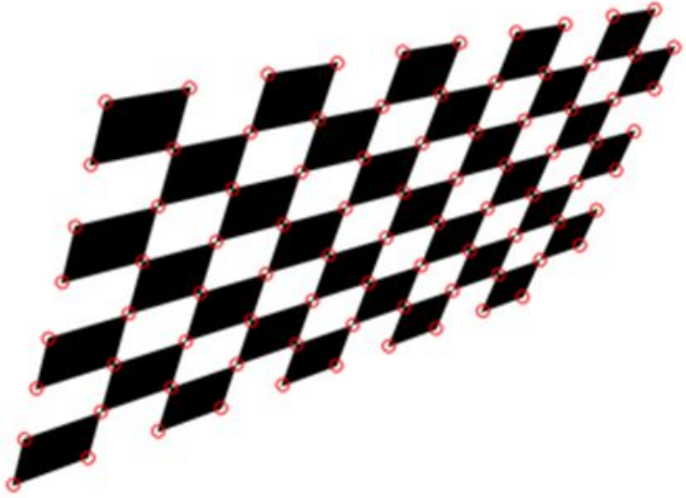


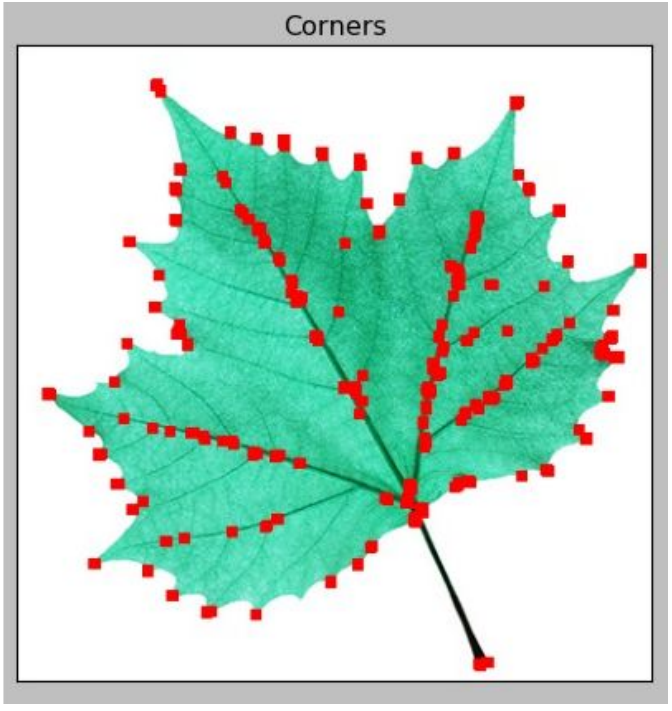
"edge":
no change along the edge direction



"corner":
significant change in all directions

Harris corner detection

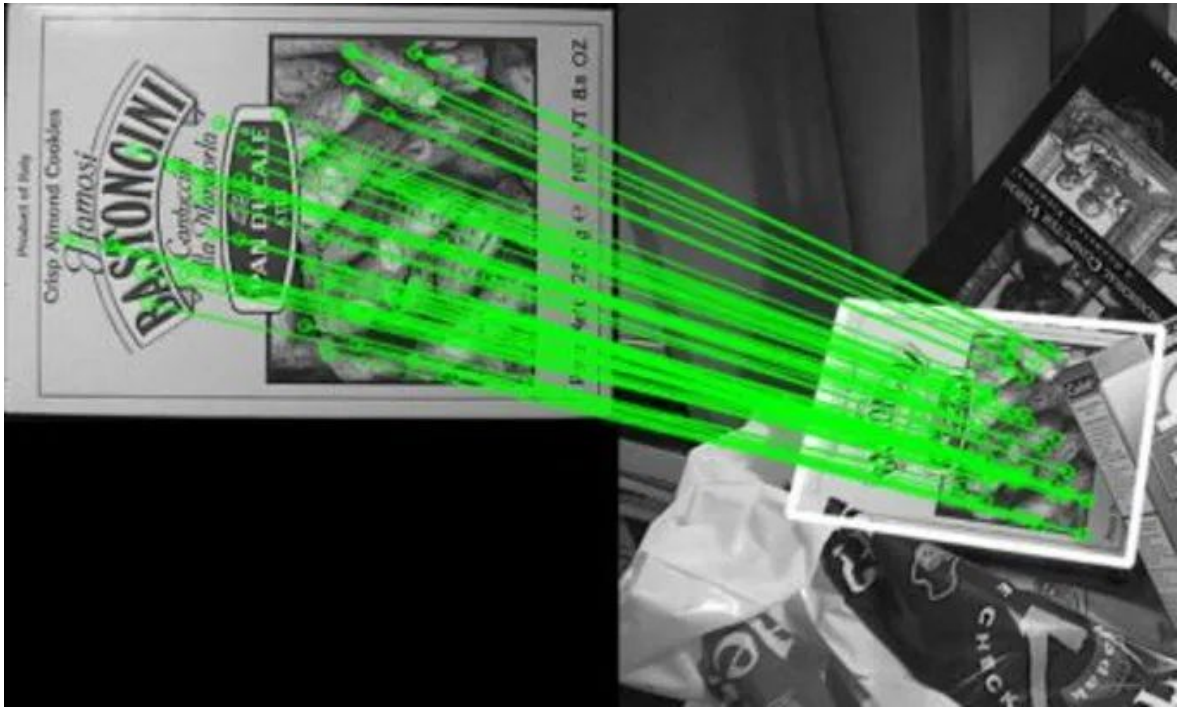




SIFT algorithm

Feature matching

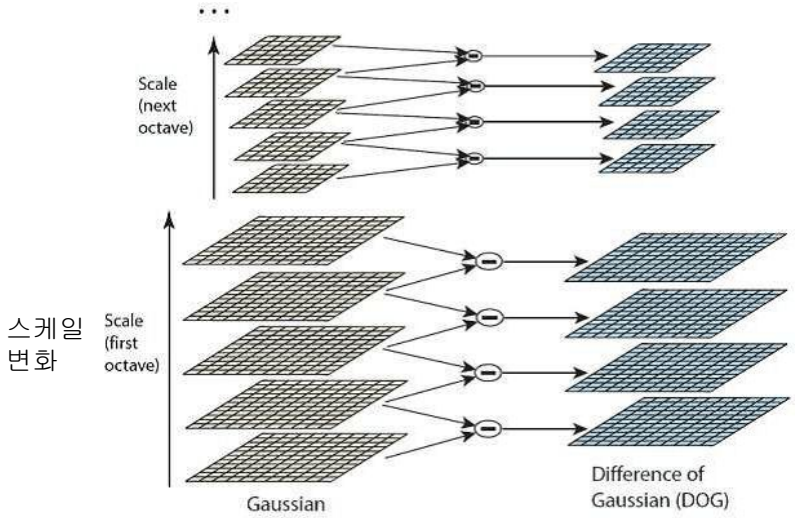
두 영상의 작은 지역들의 특징을 표현하는 표현자(descriptor)를 생성하여 서로 유사도가 높은 표현자들 간의 정합(Matching)을 수행하는 것.

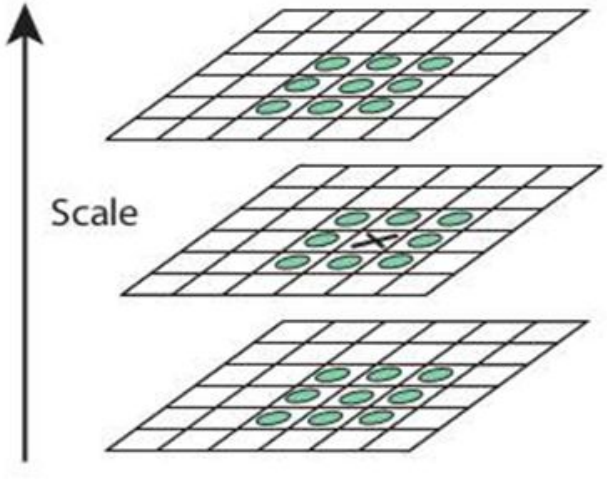


SIFT algorithm의 아이디어

나무에서 붉은 박스 내부를 바라볼 때,
안경을 쓴 사람과 벗은 사람은 각각 어떤 것을 보게 될까?

Scale 변화에 따른 픽셀값 변화





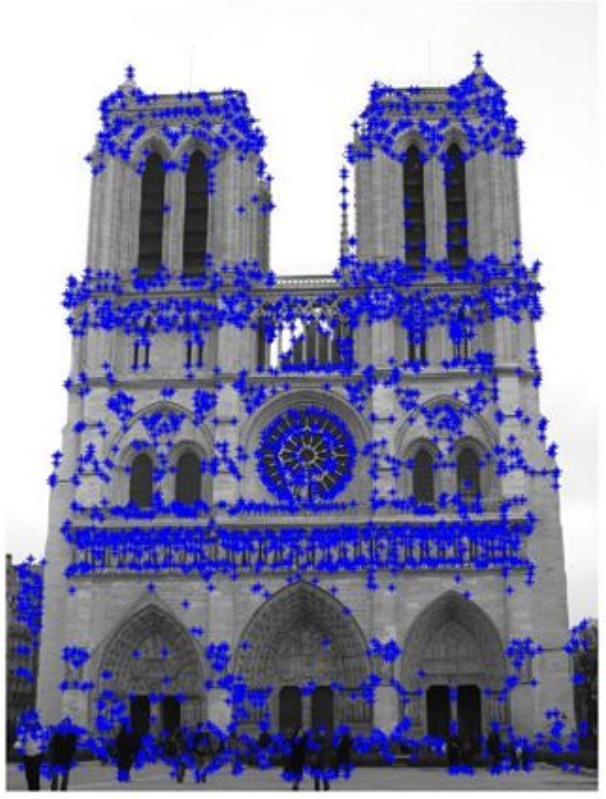
Scale 변화에 따른 픽셀값 변화값 계산

DOG 이미지들에서 주변 26개 점에 대해 minima와 maxima인 픽셀들을 찾아낸다. Scale 변화와 관계 없이 Pixel값 변화가 큰 point를 찾아내는 것을 의미(Scale robustness)

SIFT : keypoint detection

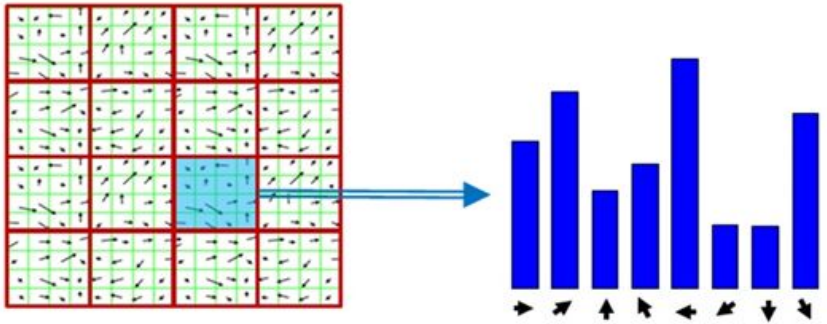
- Eliminates any low-contrast keypoints (A extrema is less than a threshold value, it is rejected)
- Eliminates edge keypoints

So what remains is strong interest points

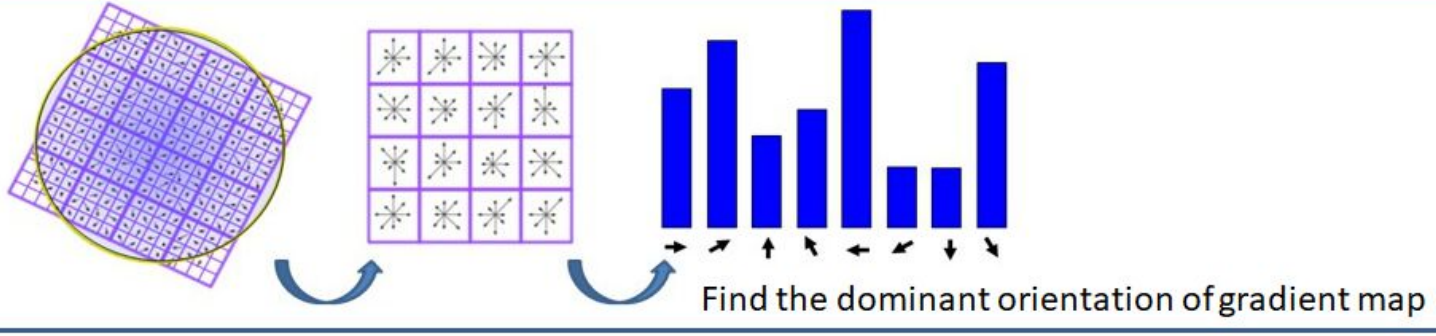


SIFT : Feature extraction

- Orientation Assignment & Keypoint descriptor

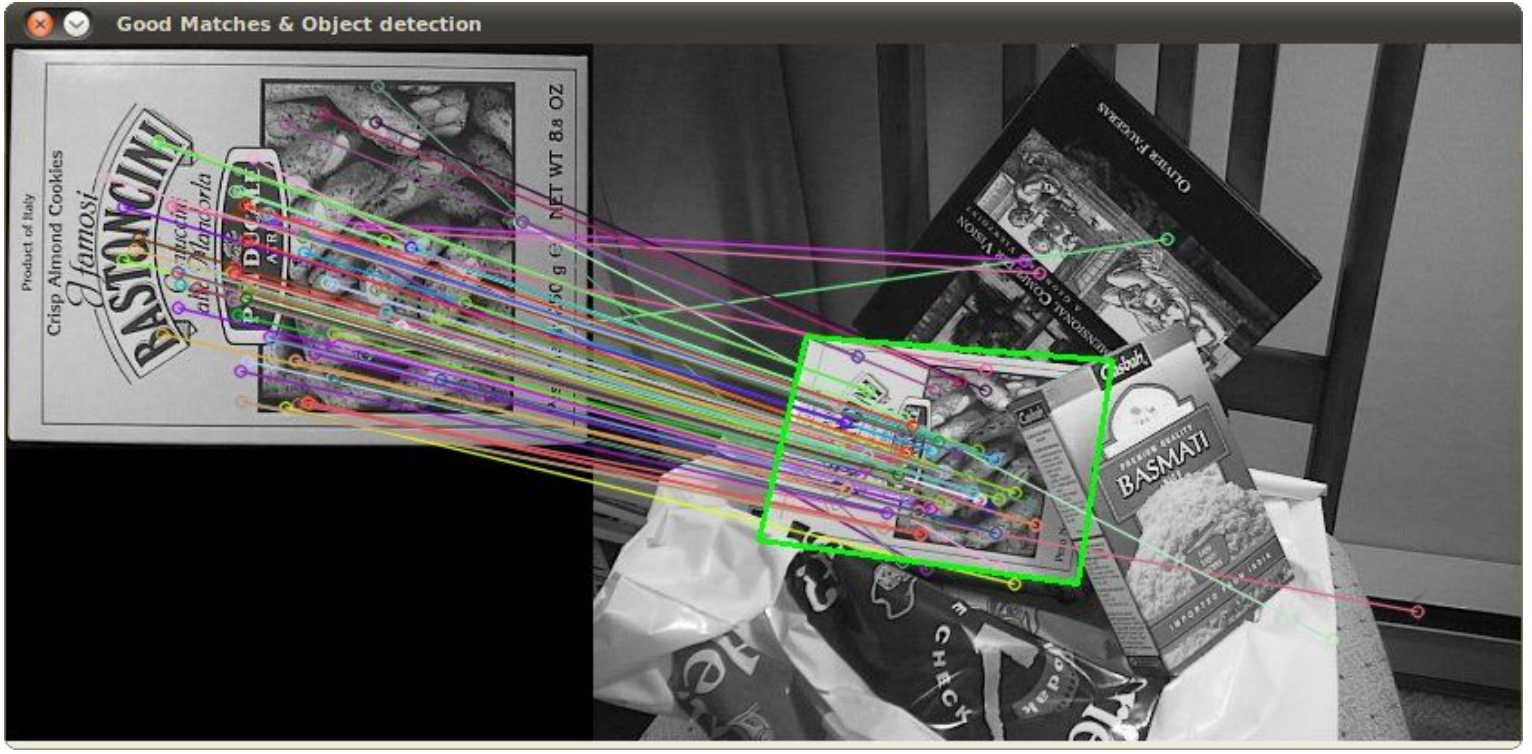


1. 각각의 4x4 patch에 대해 8-bin의 histogram을 만든다.
2. Dominant orientation (keypoint 주변 gradient의 주방향)을 계산한다.
3. 16개의 8-bin histogram의 값들을 Dominant orientation 각도만큼 shift시켜 준다



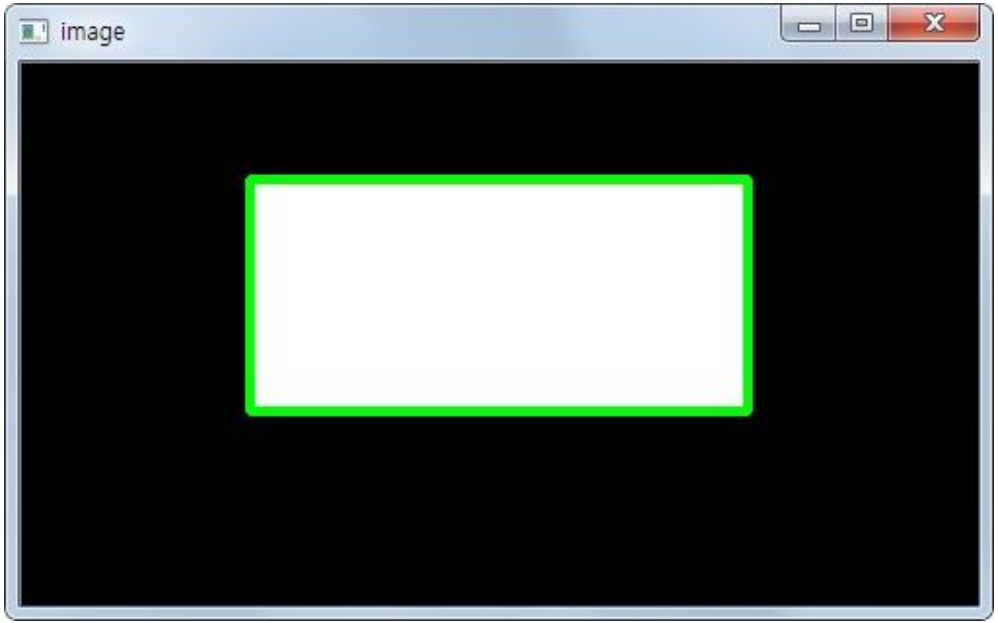
SIFT : Feature matching

SIFT : Feature matching

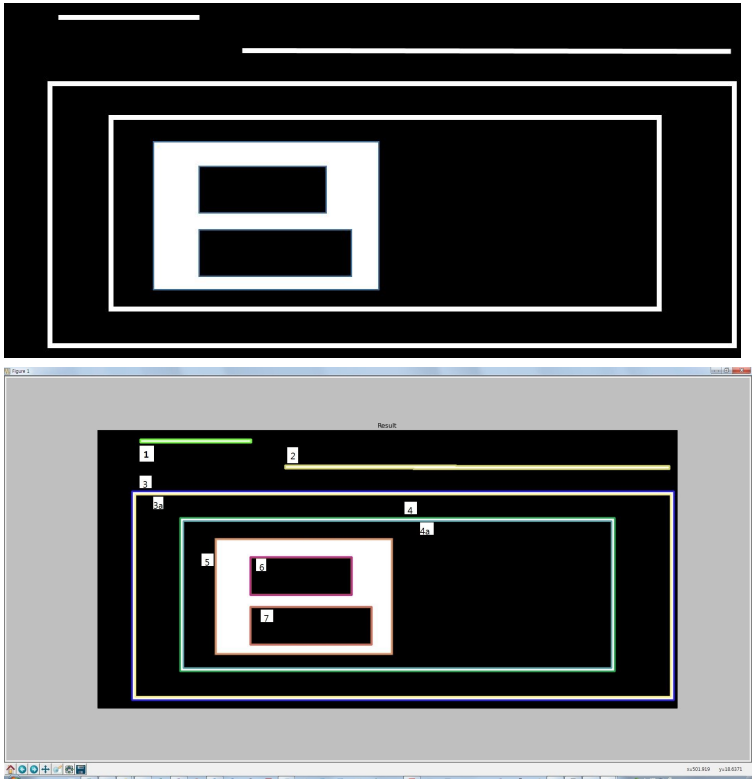


Contour detection

Image contour



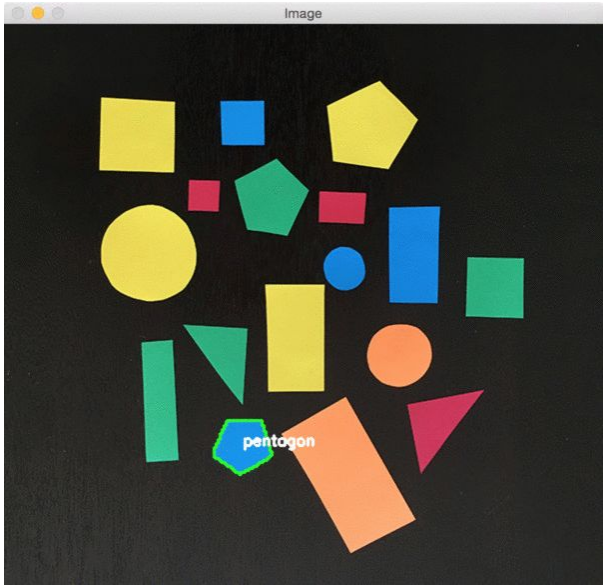
간단한 Shape에서의 Contour 검출



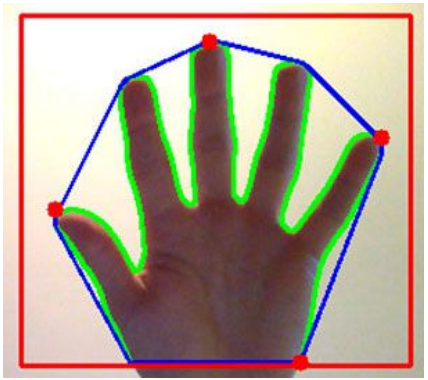
Contour들간의 계층 구조

Image ref : https://opencv-python.readthedocs.io/en/latest/doc/15.imageContours/imageContours.html#goal

Shape detection and Convex hull



Shape detection



Convex hull

Image ref : https://www.pyimagesearch.com/2016/02/08/opencv-shape-detection/
https://opencv-python.readthedocs.io/en/latest/doc/15.imageContours/imageContours.html#goal

Free Google Slides Templates