

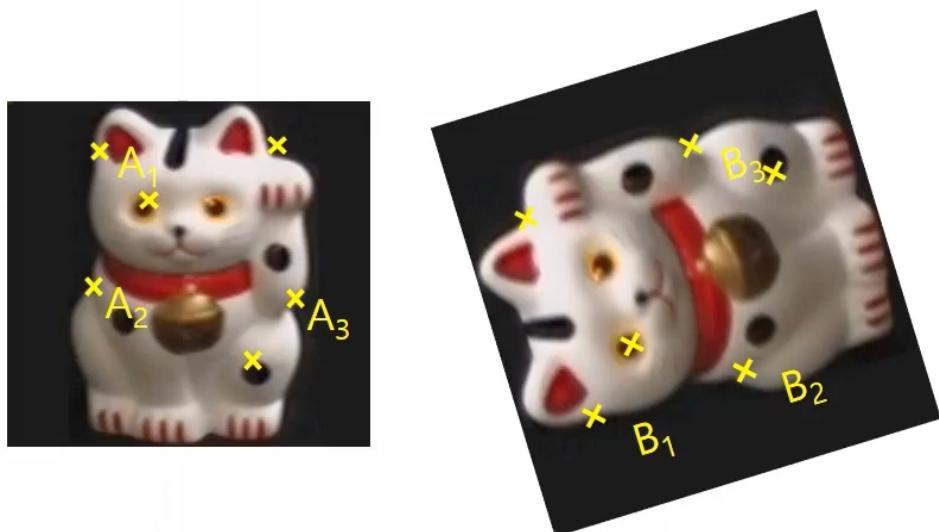


Feature descriptor

학습목표

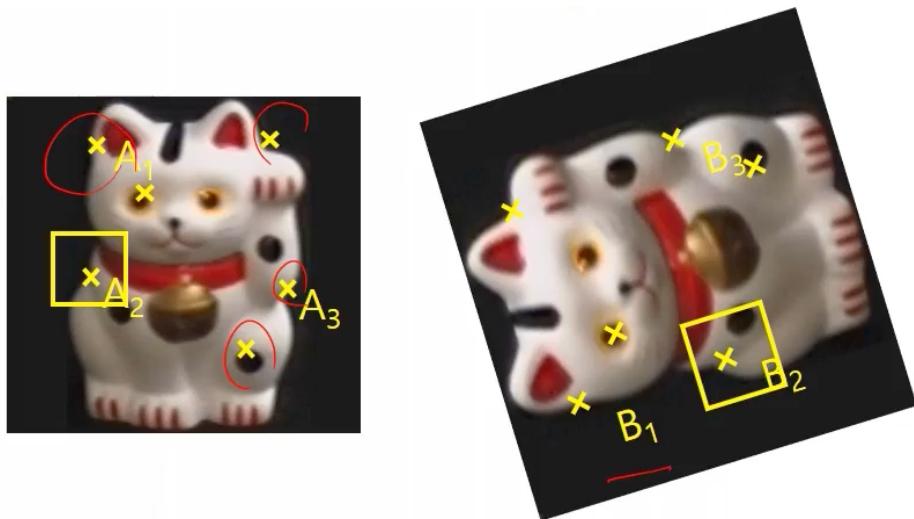
Feature descriptor에 대해 학습한다

두 이미지의 key point를 노란색 엑스로 표시했다고 치면

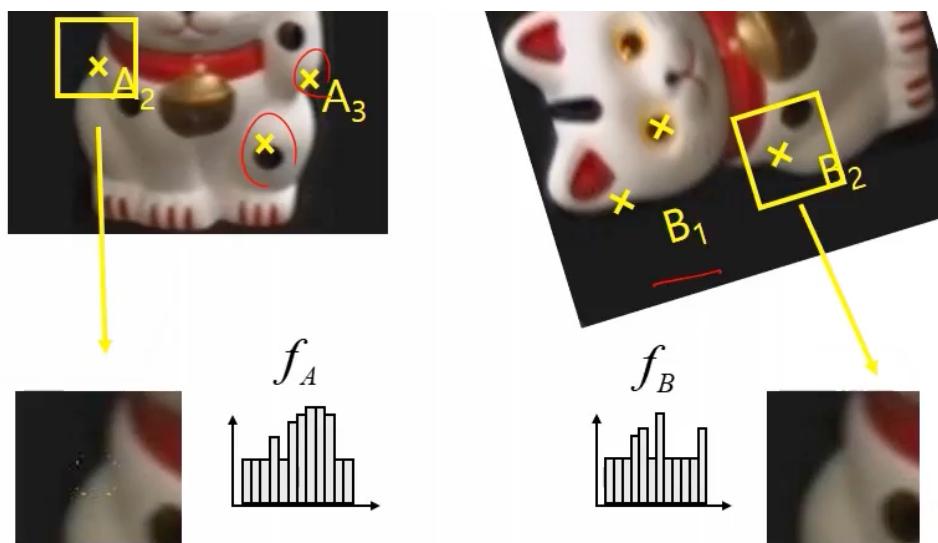


A_1 과 B_1 은 같은 위치일텐데 이 위치를 표현하기 위해 백터를 사용함

근데 딱 해당하는 픽셀만 기술하는게 아니라



이런식으로 local한 범위를 잡고



분포로 나타낼수 있을 것임 그리고 y축의 값들(특징)을 벡터형태로 나타낼수 있고 이것을

$$d(f_A, f_B) < T$$

로 거리를 재는데 Threshold보다 낮다고 하면 같다라고 표현 할 수 있을것임

이런 거리 방법을 표현하는 것은 **L1, L2, similarity** 등 많게 표현 할 수 있는데 여기서는 다
루지 않을 것임

Image patch



이것도 특징 표현의 하나의 방법으로 여길 수 있지만

What are the problems?

How can you be less sensitive to absolute intensity values?

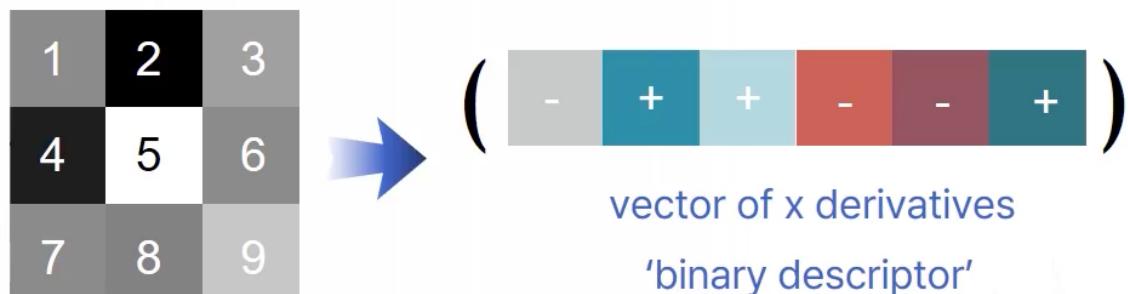


같은 패치가 밝기가 다르면? 거리를 잰다면 높게 나올 것임 → 다른 이미지???

뭔가 이상함

그렇다면 gradients를 이용하여 변화량을 기록해볼까?

Image gradients



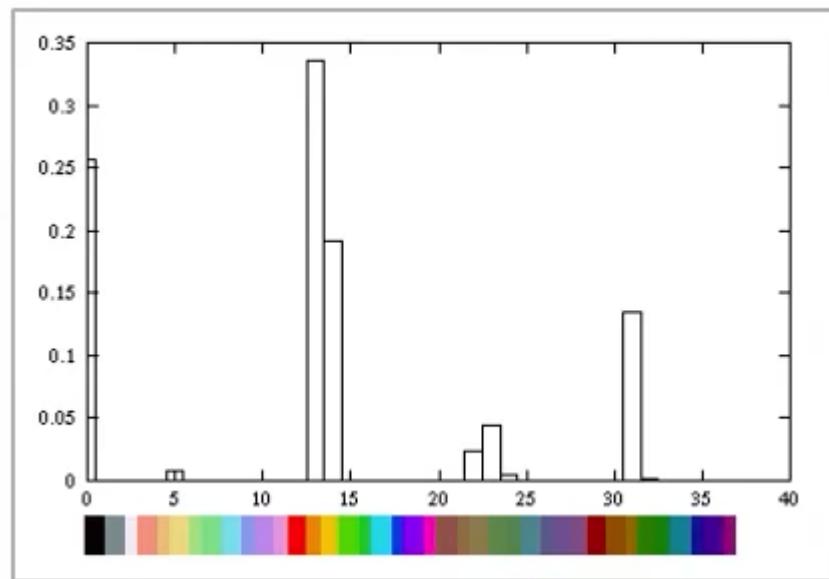
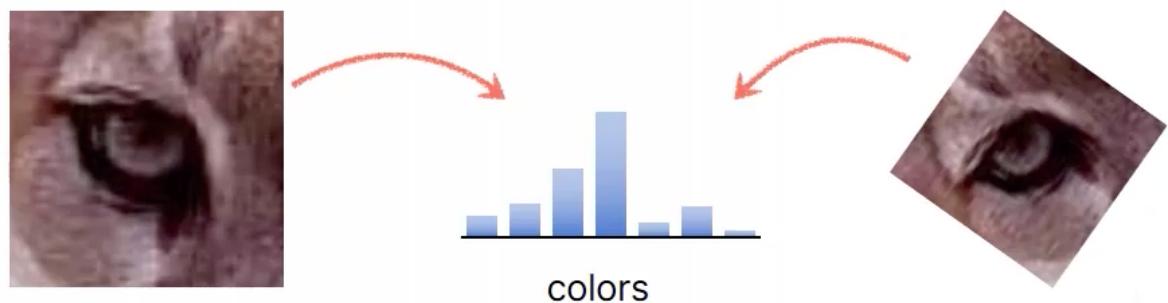
상대성을 이용하여 밝기에 관해서는 Robust하게 표현될 수 있겠지만 한계가 있음



이렇게 Rotation이 된 패치라면?? 이렇게 기하학적인 변환에 약함(동일하게 가로로 scanning 하니깐)

이런 geometric conversion에 Robust하게 하려면 histogram을 이용해보자

Color histogram

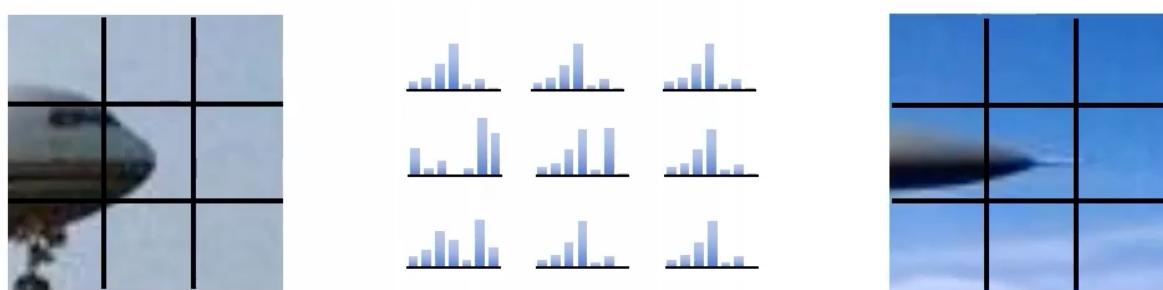


빈도수로 판단하기 때문에 geometric conversion에는 좀더 로버스트해짐
하지만



비슷한 색상 패턴의 패치는 서로 다른 patch인데 같은 patch라고 인식해 버릴 수가 있음

그래서



하나의 패치를 여러개의 셀로 분할해서 표현한다면(좀더 deep하게 본다면) 히스토그램의 문제를 조금 해결할 수 있을 것임

❓ Then, How can we be completely invariant to rotation?



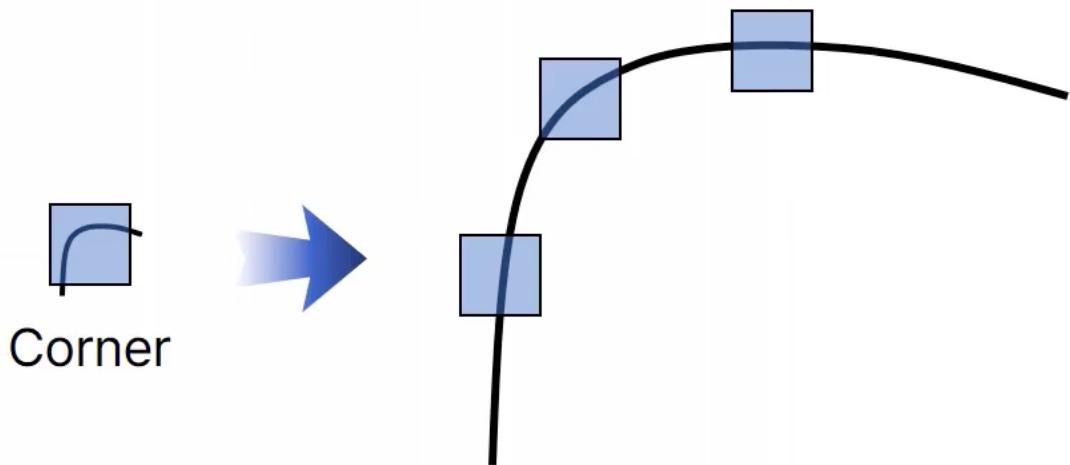
save the orientation angle θ along with (x, y, s)

패치를 기록할 때 어떠한 방식이든 각도를 예측해서 0° 로 맞춰놓고 기록하자

어떻게 할까?

SIFT(Scale Invariant Feature Transform)

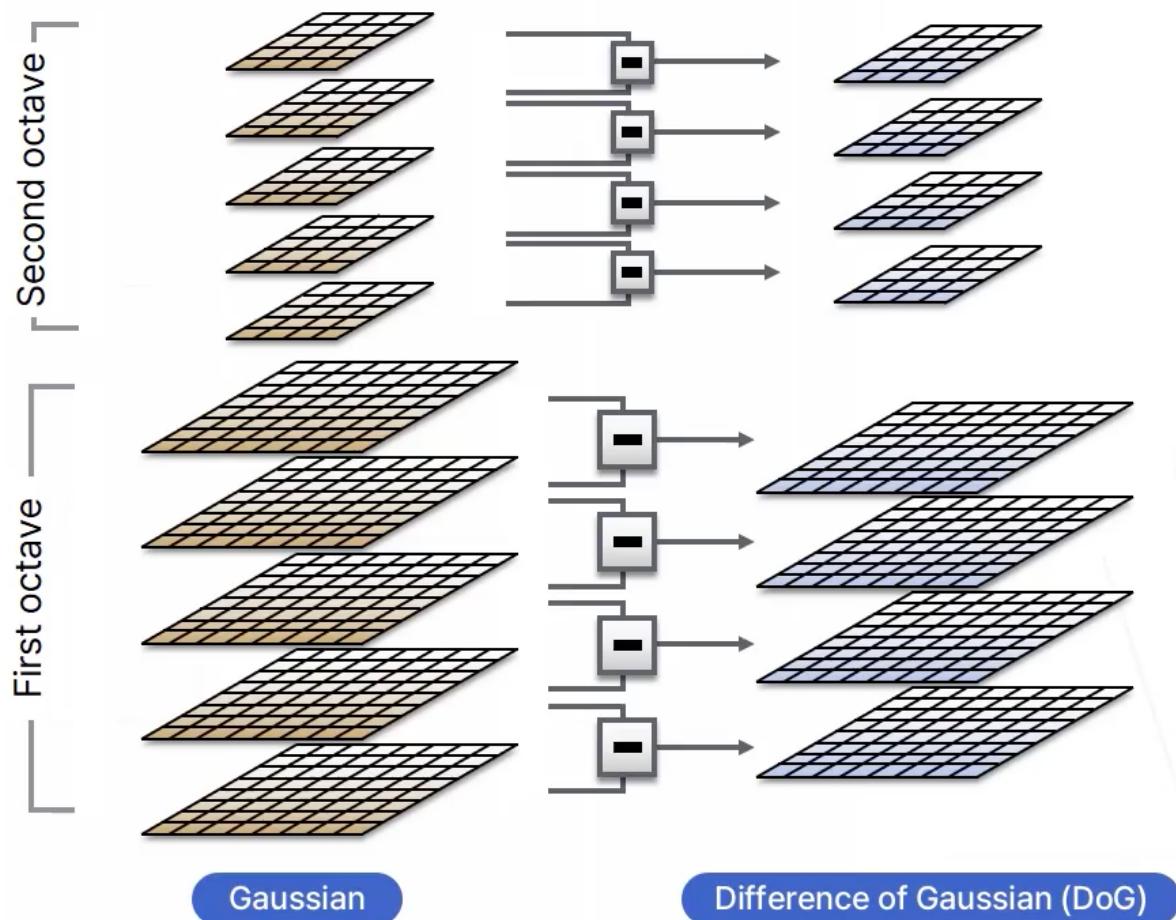
1. Multi-scale extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor



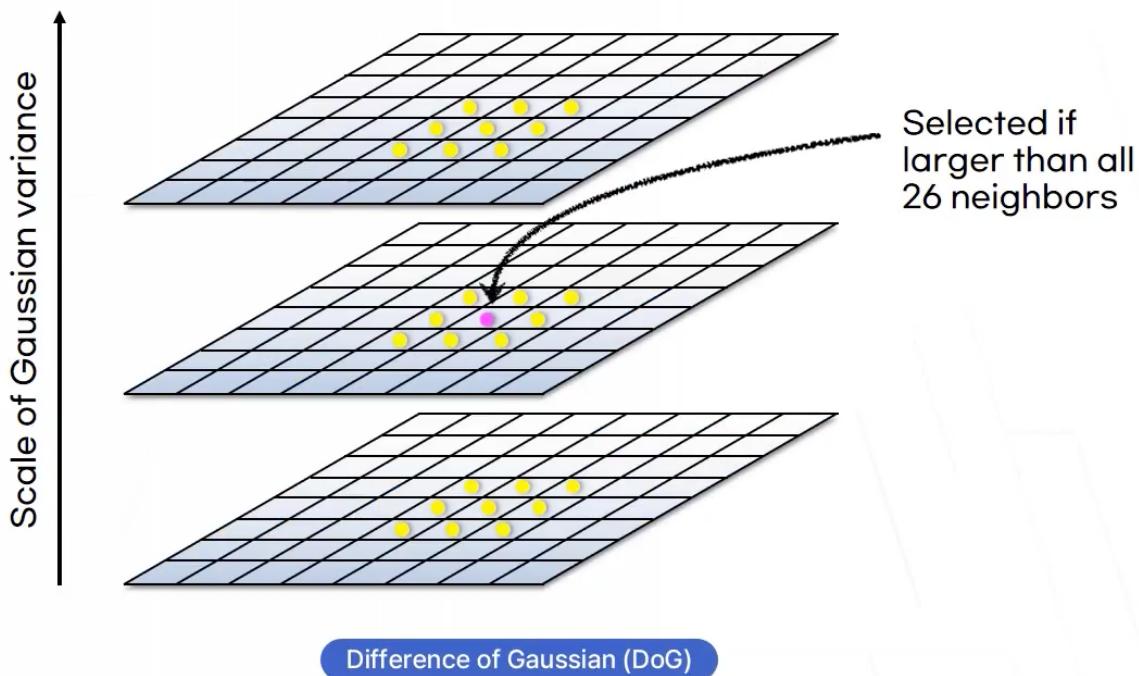
기존의 Harris corner detector에서는 패치의 크기 때문에 edge를 Corner로 판단해버리는 등 문제가 있었음. 패치크기를 키울수도 없고(피사체가 큰지 작은지 모르기 때문에)..

이러한 문제를 해결하고자 SIFT에서는 Multi scale 방식을 차용하였음

1. Multi-scale extrema detection



원래의 사이즈에서도 key point를 뽑고 resize한 이미지에서도(작은 scale)에서도 key point를 뽑음.



맨 밑에는 약한 σ (patch의 필터에서)를 주고 올라갈수록 강한 σ 를 주어 blur를 강하게 주는데 이것이 의미하는 이유는 σ 에 따라 주변값들의 영향을 조절해줌 → 윗 레이어 아래 레이어를 뺀다면 다른 가우시안끼리 얼마나 다른지 추출할수 있으니 Difference of Gaussian을 얻을 수 있음

2. Keypoint localization

Discard low-contrast keypoints

$$D(\mathbf{x}) + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x} < \text{threshold}$$

주변간의 차이를 계산해서 Contrast가 높은것만 남겨주는 거 하나 있고 edge를 제거하고자 Interest points에서 했던 covariance matrix를 계산을해서 eigenvalues를 계산해주는 것

Eliminate edge response

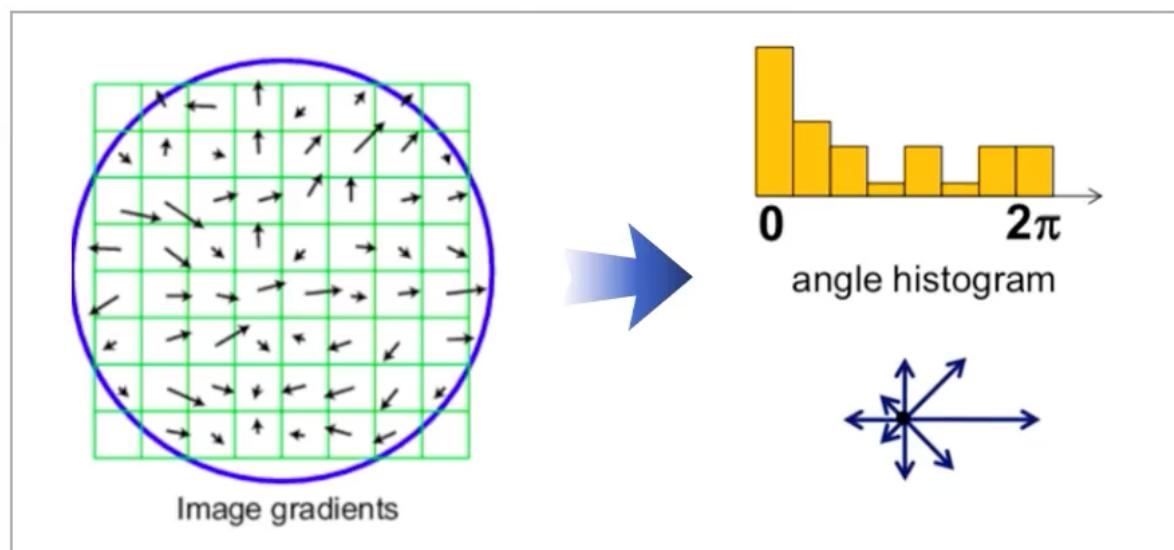
$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \text{ or } \begin{bmatrix} D_{xx} & D_{xy} & D_{xs} \\ D_{yx} & D_{yy} & D_{ys} \\ D_{sx} & D_{sy} & D_{ss} \end{bmatrix} \rightarrow \text{Compute eigenvalue}$$

The ratio of the eigenvalues λ_1 and $\lambda_2 <$ threshold r

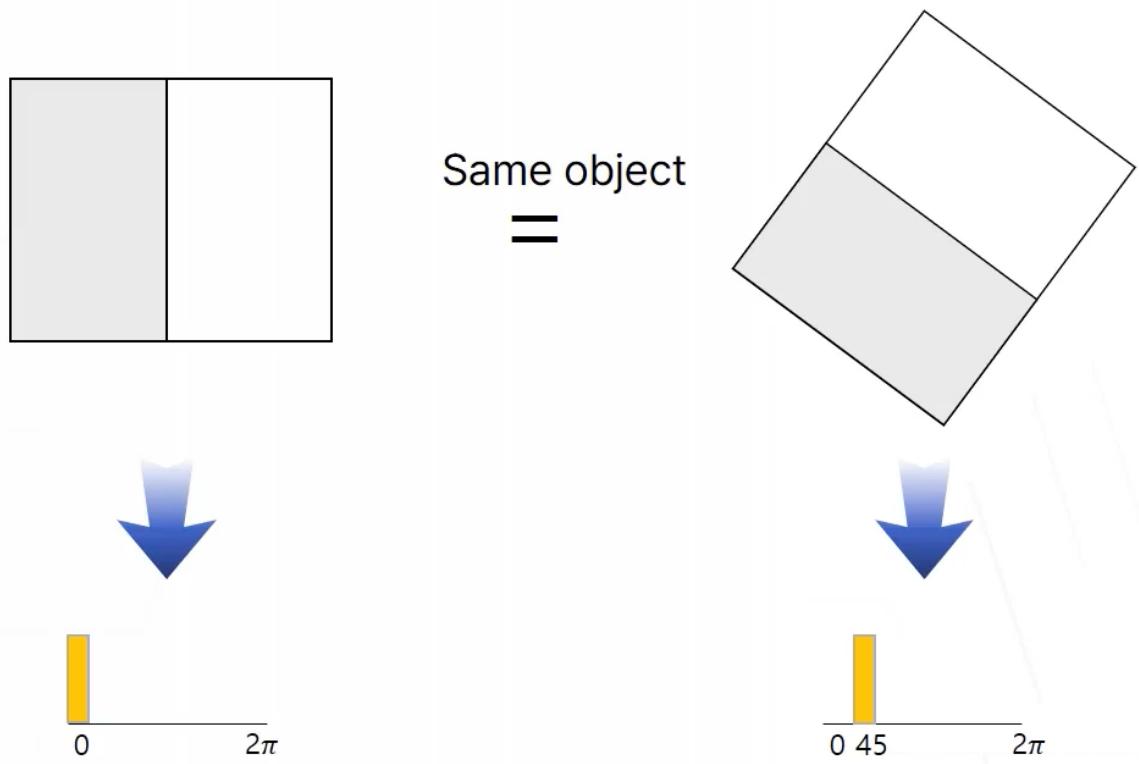
$$\frac{\text{Trace}(\mathbf{H})}{\text{Det}(\mathbf{H})} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} < \frac{(r+1)^2}{r}$$

3. Orientation Assignment

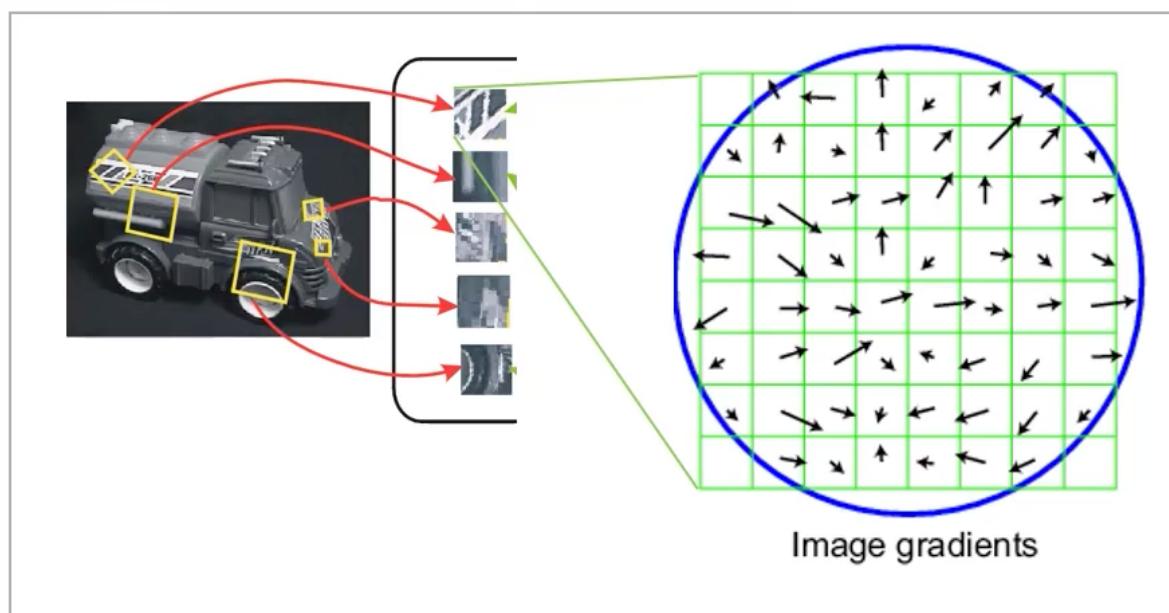
1. Take 16×16 square window around detected feature from blurred image associated with the keypoint's scale
2. Compute image gradients(magnitude and angle)
3. Throw out weak edges
4. Create histogram



오른쪽 그림의 가운데 선부분의 주각도가 45도이기 때문에 -45도 만큼 회전 시켜주면 됨



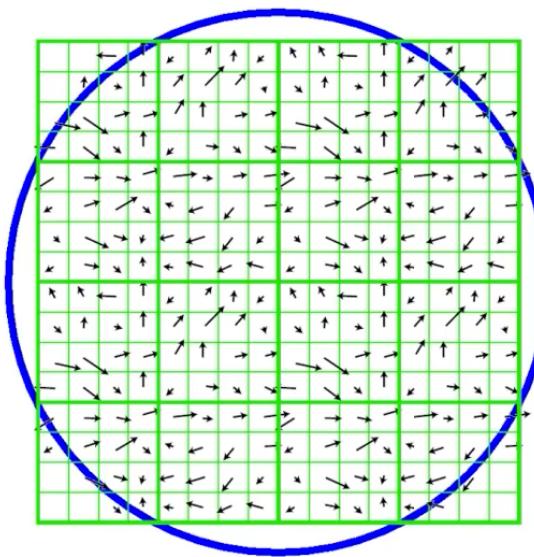
왼쪽 그림에서 key point가 5개 뽑혔으면



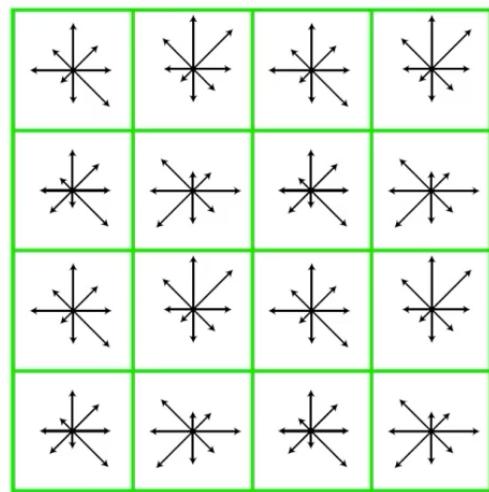
gradient를 계산을해서 각도만큼 돌려줘서 계산

4. Keypoint descriptor

Image Gradients
(4 x 4 pixel per cell, 4 x 4 cells)

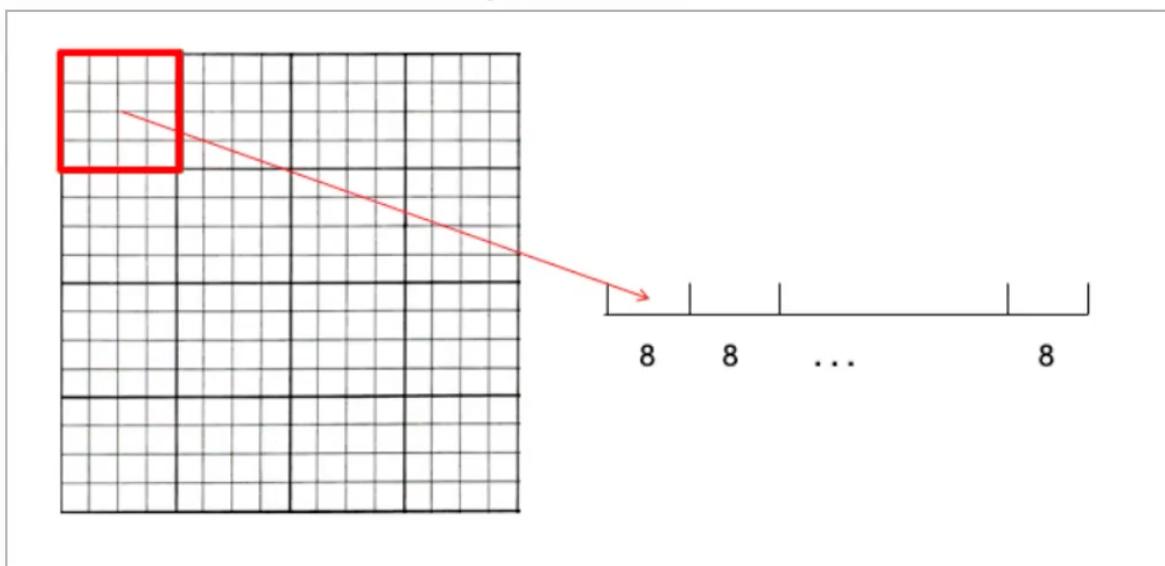


SIFT descriptor
(16 cells x 8 directions = 128 dims)

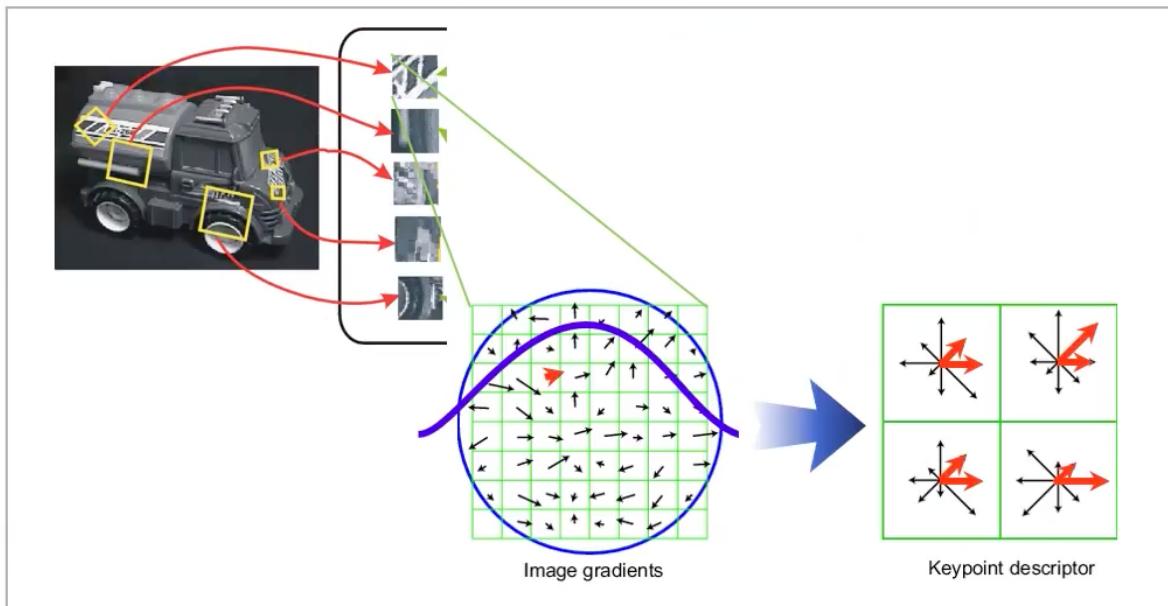


4×4 array of gradient orientation histogram weighted by gradient magnitude and Gaussian - weighted circular window

8 orientations $\times 4 \times 4$ array = 128 dimensions



Trilinear interpolation



A given gradient contributes to 8 bins: 4 in space times 2 in orientation

인접한 셀에도 영향을 줄수 있게 설계하는 방법도 있음

128-dim vector normalized to 1

Threshold gradient magnitudes to avoid influence of higt gradients

SURF('Speeded' Up Robust Features)

Compute Haar wavelet response at each pixel in patch

center of detected feature

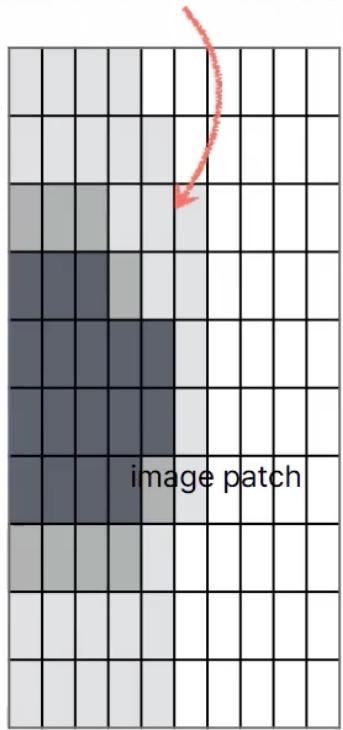
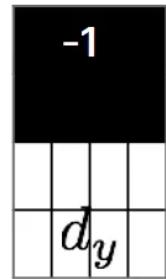
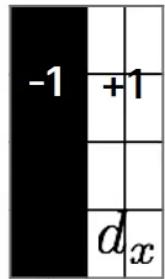


image patch

Haar wavelets filters



(Gaussian weighted from center)

*How would do you compute
the filter response?*

▼ Integral Image를 먼저 보면..

1.

31	2	4	33	5	36
12	26	9	10	29	25
13	17	21	22	20	18
24	23	15	16	14	19
30	8	28	27	11	7
1	35	34	3	32	6

2.

31	33	37	70	75	111
43	71	84	127	161	222
56	101	135	200	254	333
80	148	197	278	346	444
110	186	263	371	450	555
111	222	333	444	555	666

$$15 + 16 + 14 + 28 + 27 + 11 =$$

$$101 + 450 - 254 - 186 = 111$$

적분처럼 쌓아올린 이미지를 만들겠다 라고 생각하면 됨

그림을 보면 해당하는 위치까지 계속 더해줌 ex) 2. $37 = 31 + 2 + 4$

장점이 뭐냐? 연산을 할때 계산이 빨라짐

2.

31	33	37	70	75	111
43	71	84	127	161	222
56	101	135	200	254	333
80	148	197	278	346	444
110	186	263	371	450	555
111	222	333	444	555	666

$15 + 16 + 14 + 28 + 27 + 11 =$
 $101 + 450 - 254 - 186 = 111$

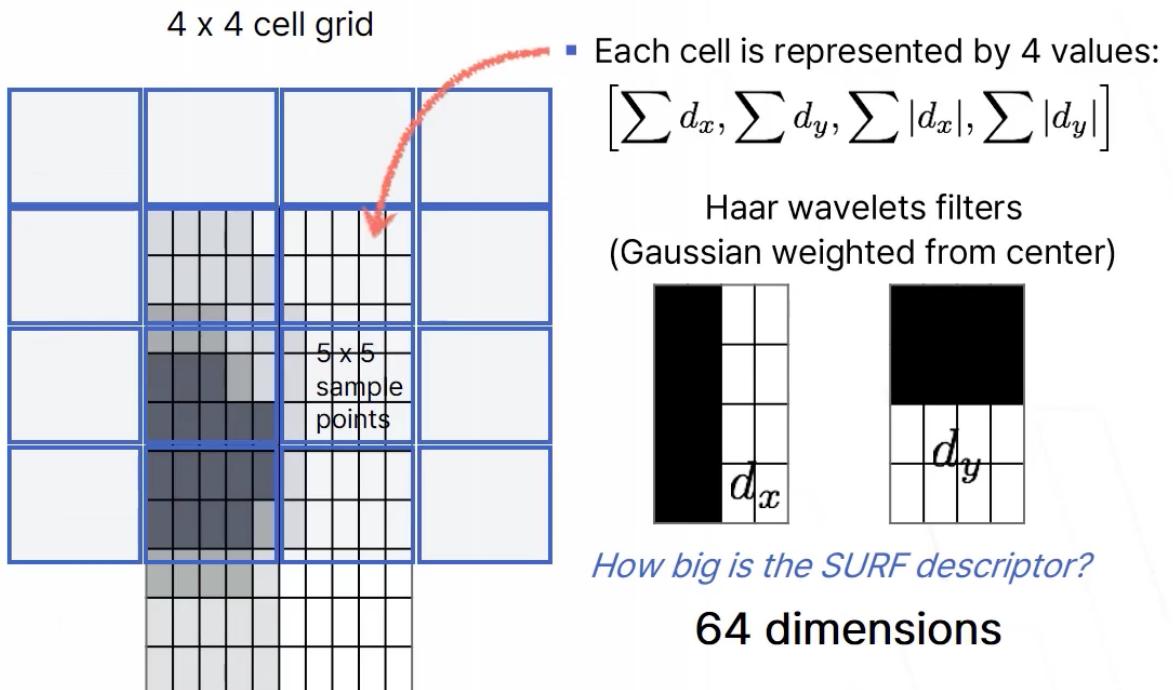
네모난 영역을 더하고 싶을때? 위와 같이 간단하게 계산할 수 있음.

covariance matrix을 구했을때 \sum 으로 다 더해줘야 하잖슴!

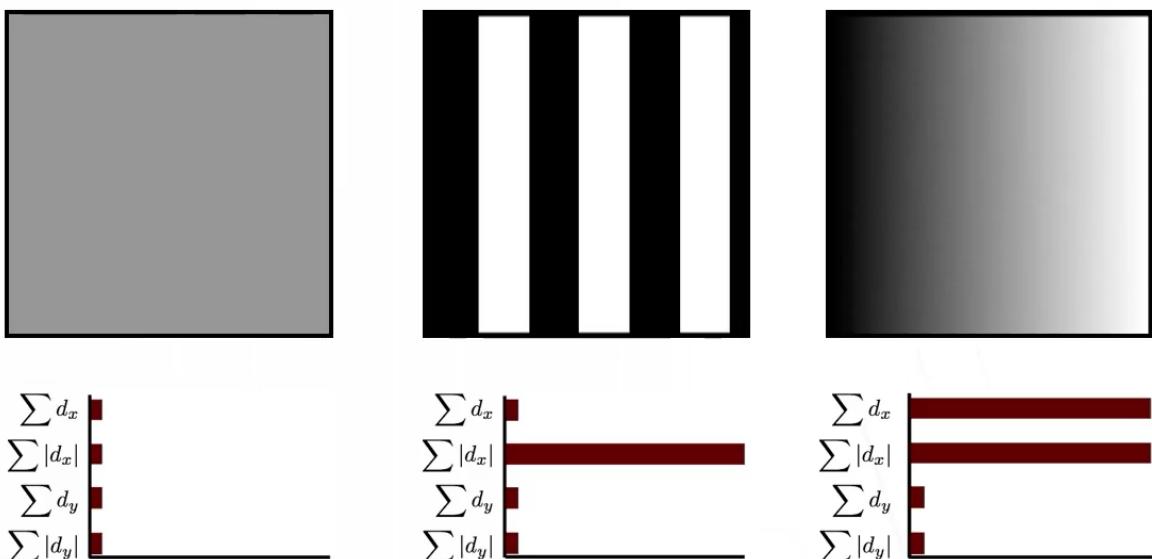


How would you compute the filter response?

- Filtering using a sliding window can be slow
- Haar wavelets are just sums over blocks
- Use integral images for efficiency(6 operations)



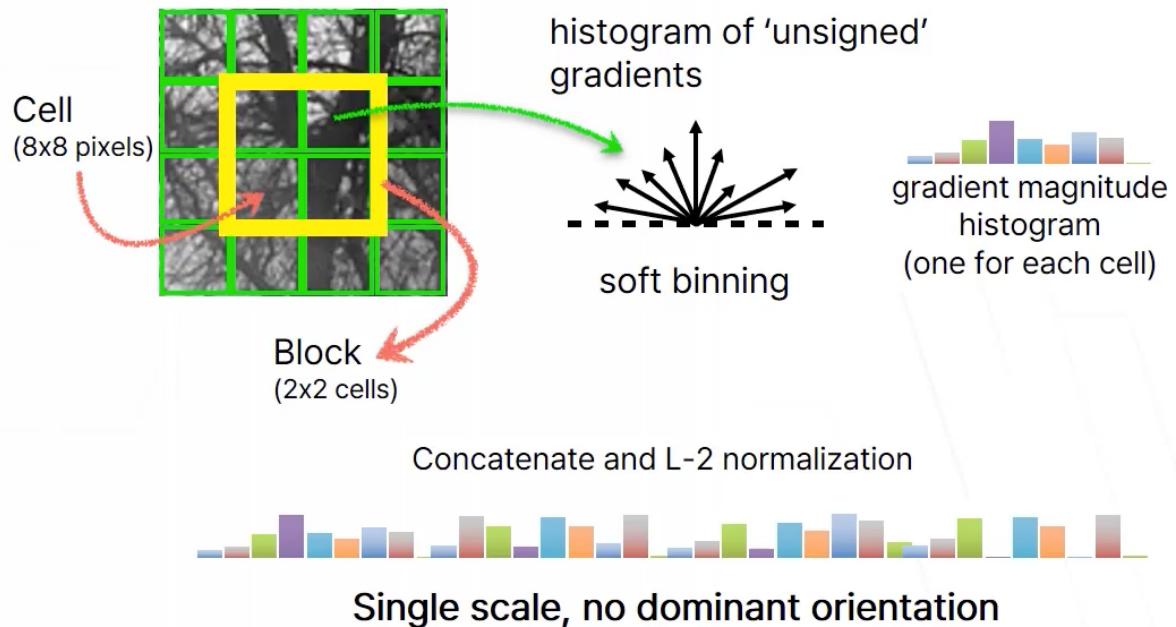
하나의 픽셀을 중심으로 16개의 Cell로 나눠서 haar wavelets filter를 거친 벡터들을 1렬로 쭉 나열하면 $1 \times 4 \times 16 = 64$ dimensions가 될것임



그래서 4개의 식이 필요한 것임

HOG(Histograms of Oriented Gradients, CVPR, 2005)

사람을 “타겟”(중점)으로 둘서 만든 기술임. 보행자 한정



patch가 있다고 했을 때, Cell로 쪼갬, 하나의 Block을 추출하고 이 블럭에 대해 gradients histogram을 구함(부호를 생각하지 않고, 0~180)

왜 부호를 무시할 수 있을까?

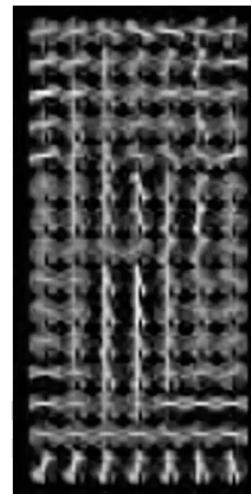
사람의 얼굴로 생각하면(원) 방향이나 방향을 동일하게 쳐도 될것임

1 cell step size

128 pixels
16 cells
15 blocks



visualization



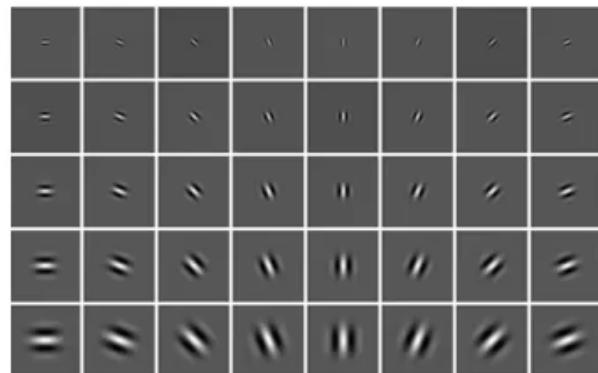
$$15 \times 7 \times 36 = 3780$$

64 pixels
8 cells
7 blocks

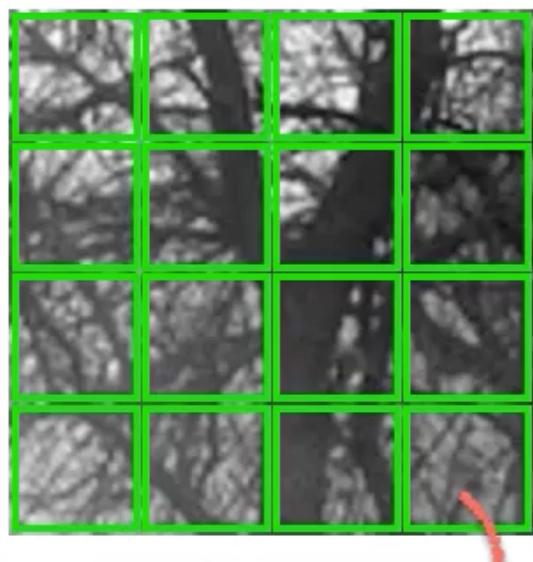


GIST

Filter bank

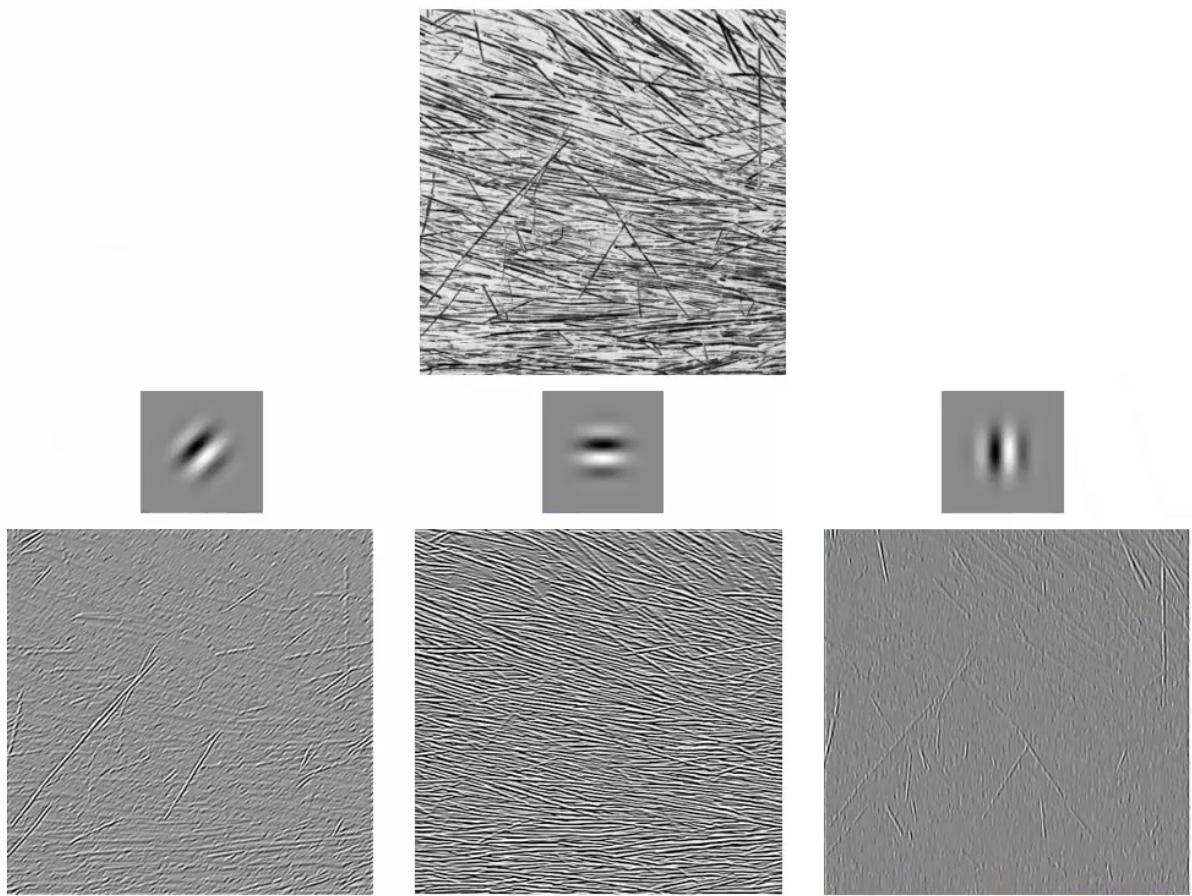


4 x 4 cell



averaged filter responses

1. Compute filter responses(filter bank of Gabor filters)
2. Divide image patch into 4×4 cells
3. Compute filter response averages for each cell
4. Size of descriptor is $4 \times 4 \times N$, where N is the size of the filter bank



필터를 미리 만들어서 필터링을 통해 구조적인 특징을 추출하는 것임