Lect 7 Corner & Blob detector

Good feature should be

1. Repeatable.

   Invariant w.r.t. Scaling. rotation

2. Saliency

   Each feature is distinct and can be uniquely found

3. Compactness and efficiency

   Not too many

4. Locality

   Can be found even in small region and resist to attack like blurring.

Corner Detector

We will classify region to ① flat ② edge ③ corner.

1. flat: if we move ↑↓←→, no big change

2 edge: no change among one direction. but large change among other

3 corner: once move, it will change

We inspect: $$E(u,v) = \sum_{(x,y)\in W} [I(x+u, y+v) - I(x,y)]^2$$

W: window

We use first-order taylor expension.

$$I(x+u, y+v) \doteq I(x,y) + u I_x + v I_y$$

$$I_x = \frac{\partial}{\partial x} I(x,y)$$

$$I_y = \frac{\partial}{\partial y} I(x,y)$$

$$\therefore E(u,v) \doteq \sum_{(x,y)\in W} \left(I_x^2 u^2 + I_y^2 v^2 + 2 I_x I_y uv \right)$$

$$= [u, v] \underbrace{\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}}_{\text{Second Moment Matrix } M} \begin{bmatrix} u \\ v \end{bmatrix}$$

Obviously, $Tr(M) \geq 0$, $Det(M) \geq 0$, $Rank(M) \leq 2$. so M is PSD.

$M = Q \Lambda Q^T$ $\quad\quad$ $a+b^2 \geq 2ab$

Now, inspect simplest case.

When $M = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$, $[u,v] M \begin{bmatrix} u \\ v \end{bmatrix} \simeq au^2 + bv^2$

if $a \simeq 0$, b is big $\Rightarrow$ big change over v } edge

if $b \simeq 0$ a is big $\Rightarrow$ big change over u }

if a, b both big $\Rightarrow$ big chang in all directions. $\rightarrow$ corner

if a, b both small $\Rightarrow$ no change $\rightarrow$ flat

More general case.

$M = Q \Lambda Q^T$. then $[u,v] Q \Lambda Q^T \begin{bmatrix} u \\ v \end{bmatrix} = C$ is a ellipse formula.

We set up a formula to measure the difference of a, b.

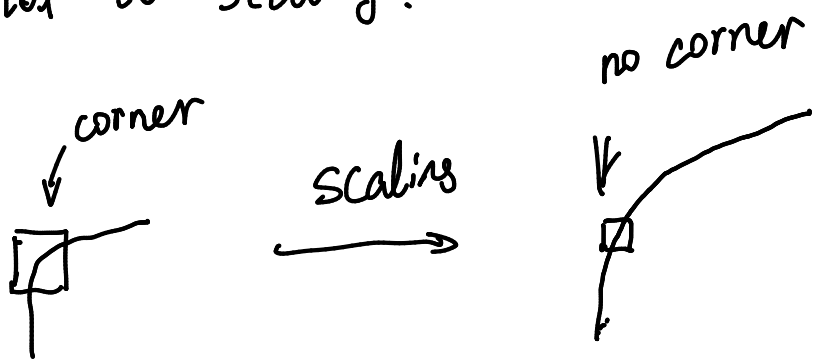
define. $R = \det(M) - \alpha Tr(M)^2 = ab - \alpha(a+b)^2$ $\quad \alpha \in [0.04, 0.06]$

- $R > 0 \rightarrow$ corner
- $R < 0 \rightarrow$ edge
- $R \simeq 0 \rightarrow$ flat

We hope corners to be invariant to photometric transformation, (say brighten)
covariant to geometric transformation
(say. shift, rotate)

But this naive method could only resist to constant addition.
not to scaling.

eg.



So, corner detector of this kind is.

- shift covariant
- rotation covariant

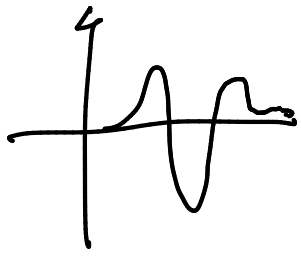
not to scaling

Blob Detection

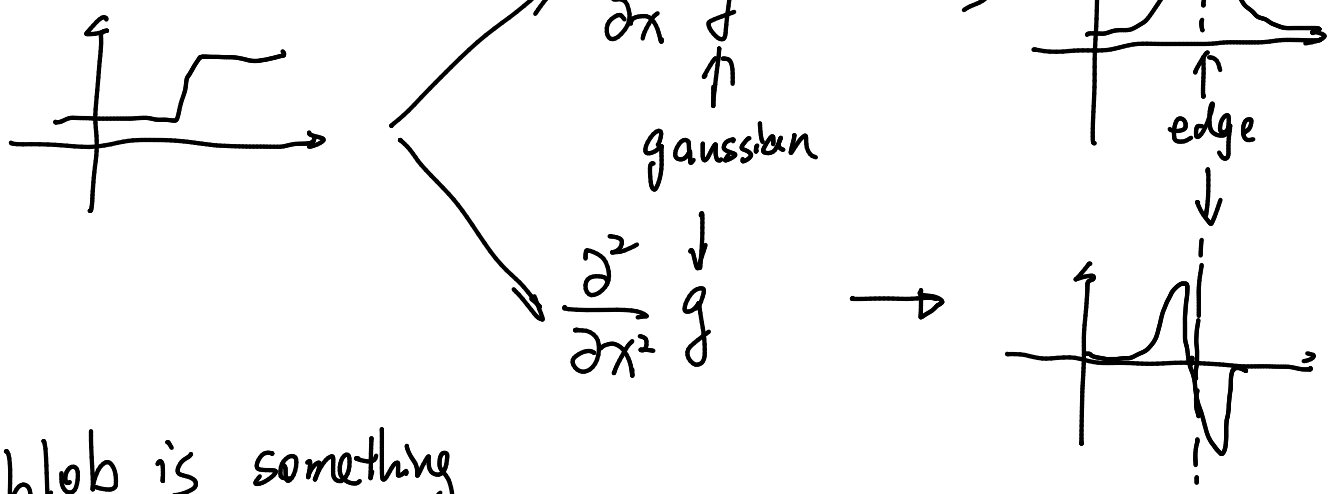
We want feature to be covariant w.r.t transformation.
↳ Blob detector.

Blob filter is Laplacian of Gaussian.

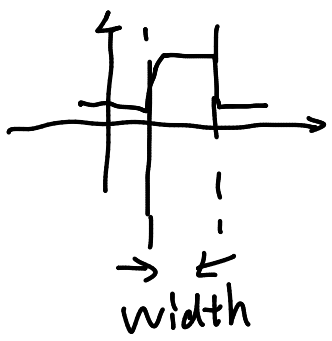
$$\nabla^2 g = \frac{\partial^2}{\partial x^2} g + \frac{\partial^2}{\partial y^2} g \quad \text{(unnormalized)}$$



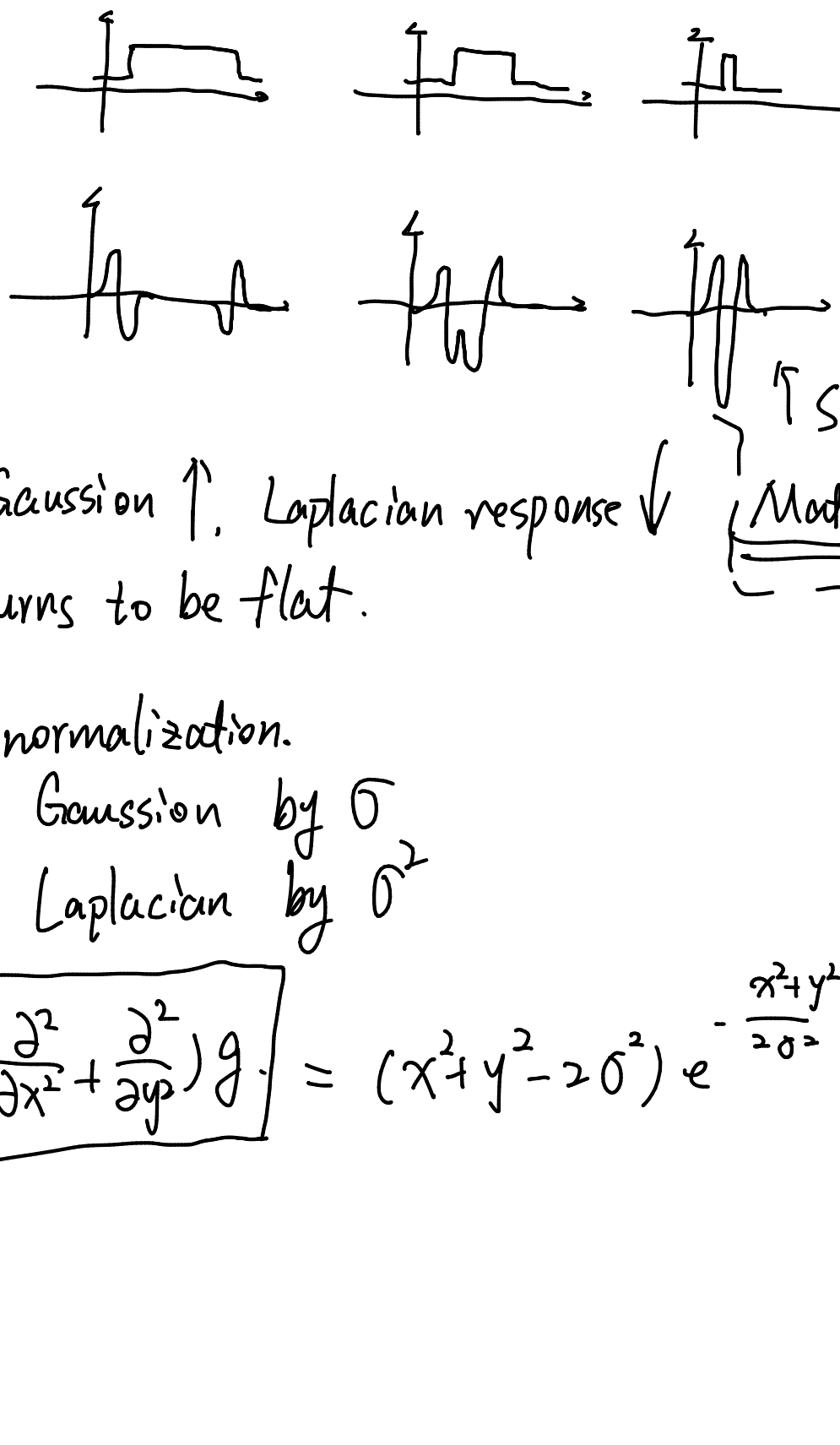
for edge



blob is something



blob of different size



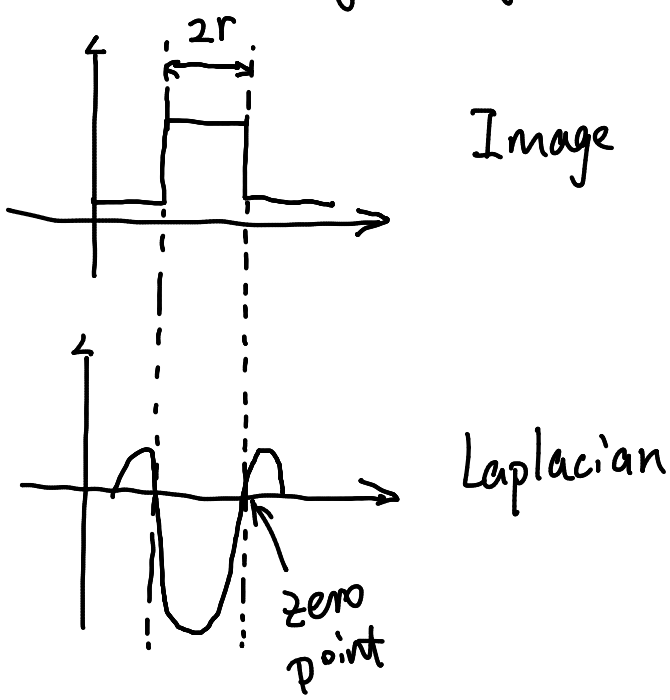
However. as σ^2 of Gaussion ↑. Laplacian response ↓ because distribution turns to be flat.

↳ We need scale normalization.

by [multiple Gaussion by σ
multiple Laplacian by σ^2

Now $\boxed{\nabla^2 g = \sigma^2 \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) g} = (x^2 + y^2 - 2\sigma^2) e^{-\frac{x^2+y^2}{2\sigma^2}}$

The case we get largest Laplacian response.



When blob is of radius r,
we get maximum response when

$\sigma = \sqrt{2}\, r$

Algorithm (Version 1)

1. Convolve image with blob filter of different σ)
2. find maximum point over different σ around its neighbour.

Algorithm (Version 2)

1. Resize image to different size. convolve with same filter
2. find maximum point as previous alg.

Simplification.

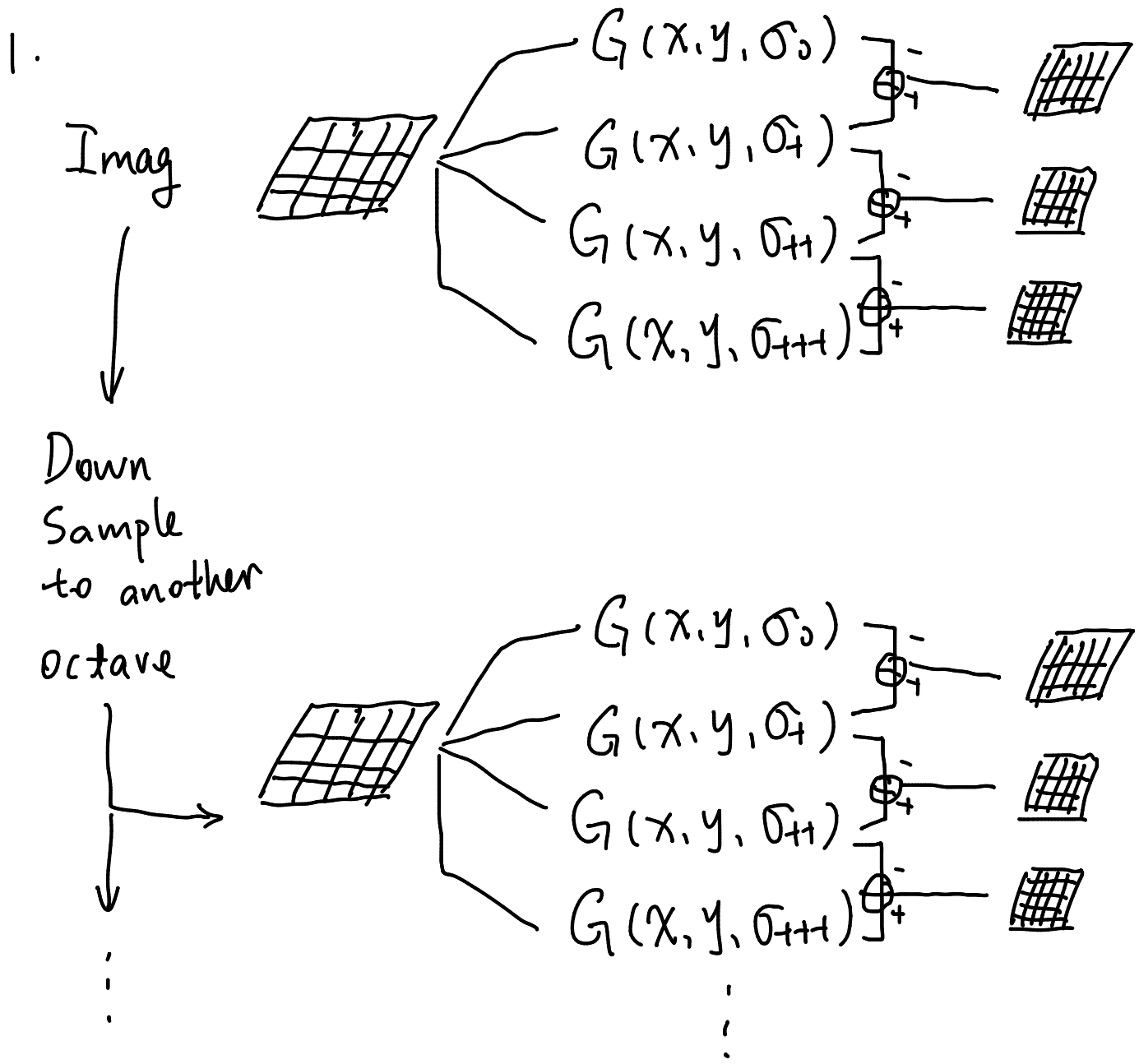
Laplacian of Gaussian $\approx$ Difference of Gaussian

$L = \sigma^2 \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) G(x, y, \sigma)$

$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$
when k is not very big.

Algorithm (Version 3)

1.



We can simply use $k\sigma_0 = \sigma_1$, $k\sigma_1 = \sigma_{1+1}$ --

2. Scale the result of each octave back and compare.

Remember.

If we change σ^2, according to 3-σ rule, we have to change filter size. So, instead, down-sample or resize image is better w.r.t computation. But, the cost is resize's precision.

From Detection → Feature Description.

We not only want "location" of each feature, but also more "info" to describe even identify these features.

Naive method

pure pixel value

Problem: Rotation ambiguity

Better solution

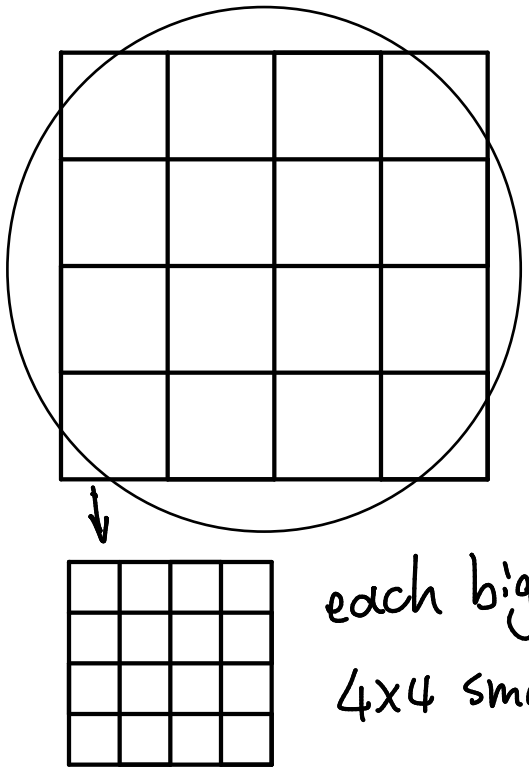
1. For pixels in a circle/patch, compute the gradient's direction.
2. Define the "orientation" of this patch to be the most shown gradient direction by histograph.

↳ SIFT algorithm.

Detection is covariant, because blob is covariant

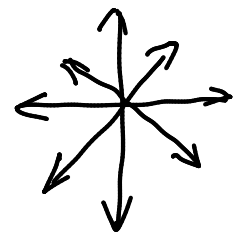
Description is invariant, because gradient is rotational invariant

For SIFT, each blob is not described by 1 direction, but a $8 \times 16 = 128$ direction vector.



4x4 big grid

each big grid turns to a 8-D vector with histograph of



8-directions.

each big grid has 4x4 small grid

So, 4x4=16 big grid =>

4x4x8=128 directions

Affine adaptation.

If a image is squeezed and rotated, we can detect the blob, but pixels may be quite different. We can find the second moment $M = R^{-1}\begin{bmatrix}\lambda_1 & \\ & \lambda_2\end{bmatrix}R$ and find out the transformation.