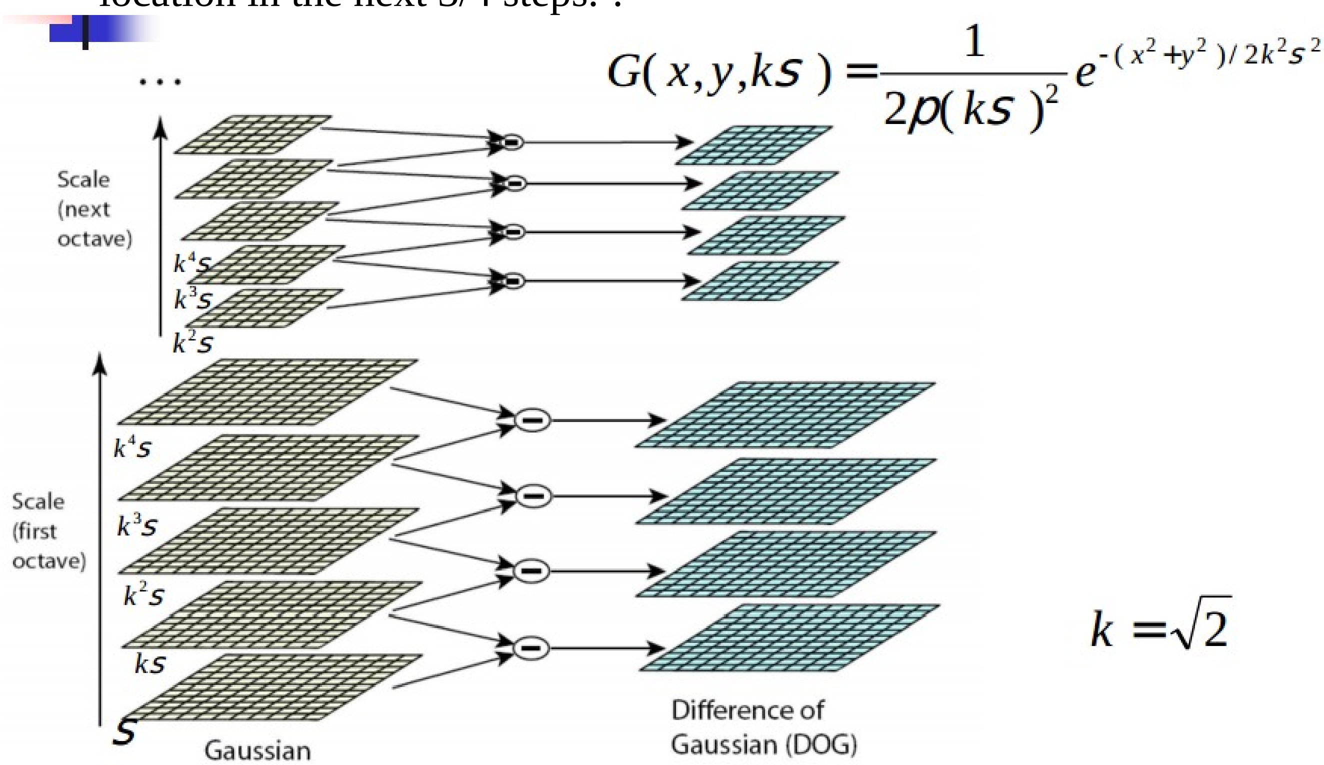
Scale Invariant Feature Transform

1. Scale space peak selection
- this step is used for finding potential location of features

- use different gaussians for capturing different scales (different scale space)
- subtract gaussians to get blob detector (laplacian of gaussian) because log can be approximated by dog and dog is computationally faster.
- find if pixel is maximum/minimum in 3x3x3 (8 pixels in same level, 9 above, 9 below) neighbourhood to select it as extrema point. (peak selection)

- use different octaves to reduce computation. For this, downscale the image, find extrema and upscale the location. But use the downscale location in the next 3/4 steps. ?



2. Key point localization
-previous step found potential points i.e. approximate location and could be features.

(a) Initial outlier rejection
- get proper location
- use taylor series expansion to find maximum of surface
- earlier step found approximate point in the neighbourhood (because of smoothing). This step finds the exact point in that neighbourhood.

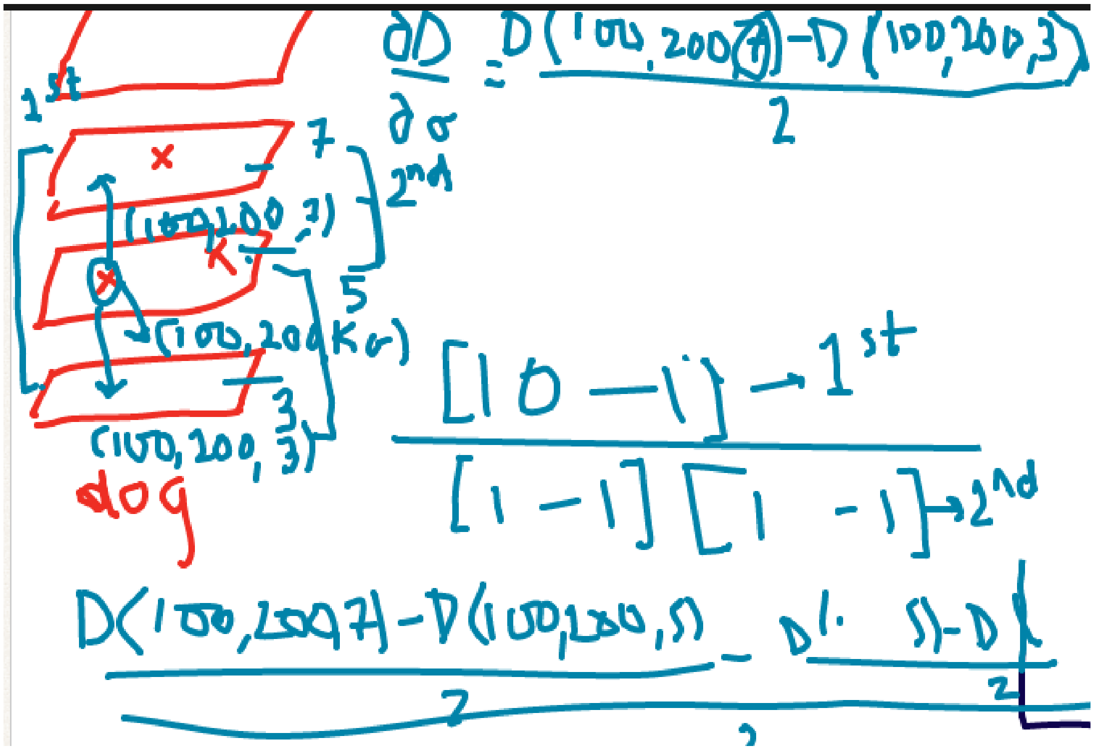
- for all the potential points (x,y,scale,octave), perform the following steps
- take the dog values
- calculate first derivative wrt x,y,sigma

$$X = (x, y, \sigma) \rightarrow Dog$$

$$\frac{\partial D}{\partial X} = \begin{bmatrix} \frac{\partial D}{\partial x} \\ \frac{\partial D}{\partial y} \\ \frac{\partial D}{\partial \sigma} \end{bmatrix} = \begin{bmatrix} \frac{D(x+1,y,\sigma)-D(x-1,y,\sigma)}{2} \\ \frac{D(x,y+1,\sigma)-D(x,y-1,\sigma)}{2} \\ \frac{D(x,y,\sigma+1)-D(x,y,\sigma-1)}{2} \end{bmatrix}$$

- calculate second derivative

$$H(x) = \begin{bmatrix} D_{xx} & D_{xy} & D_{x\sigma} \\ D_{yx} & D_{yy} & D_{y\sigma} \\ D_{\sigma x} & D_{\sigma y} & D_{\sigma\sigma} \end{bmatrix}$$

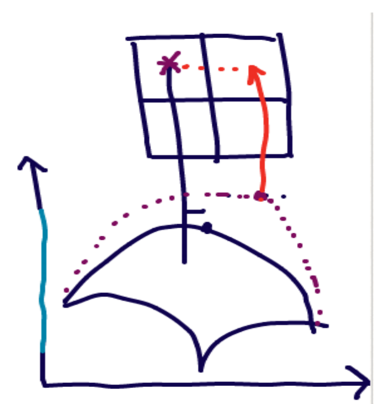


- find the proper offset in (x,y, sigma)(shift in location) of extreme. Add the offset if bigger than 0.5. This is the new location of that point.

$$\hat{x} = -\left(\frac{\partial^2 D}{\partial x^2}\right)^{-1}_{3\times 3}\left(\frac{\partial D}{\partial x}\right)_{3\times 1} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}_{3\times 1}$$

$$a > 0.5, \quad x = x + \text{ceil}(a)$$

*round instead of ceil



- substitute the offset in taylor series and threshold the value. Discard if less than 0.03

$$D(\hat{x}) = D(x, y, \sigma) + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x}$$

-this step rejects the points with less contrast.

(b) Further outlier rejection
-This step is rejecting edges.(in xy plane of dog plane – forget sigma)
-So use harris corner value.

- Form the hessian matrix for the dog surface at that point

H = [Dxx Dxy
Dxy Dyy]

- find eigenvalues lambda1, lambda2
- calculate corner value as
  r = lambda1/lambda2
- reject the points if r > 10

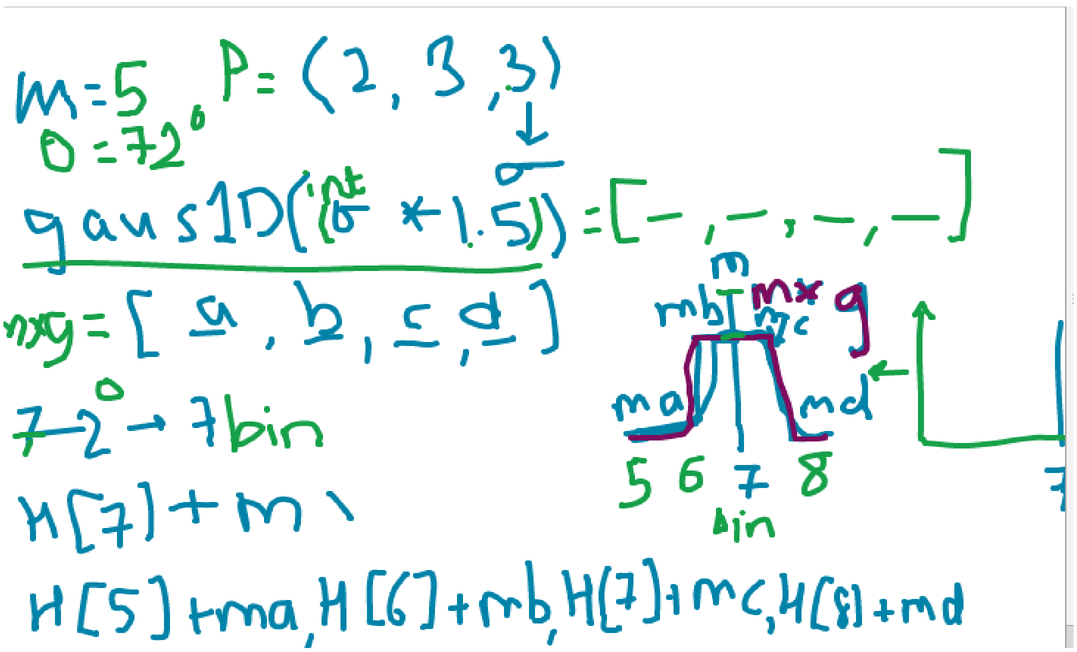
3. Orientation assignment
-find dominant orientations of the keypoint. This is for converting the feature into rotation invariant.

- find the magnitude and orientation for every point in 4x4 neighbourhood around that point for that scale.

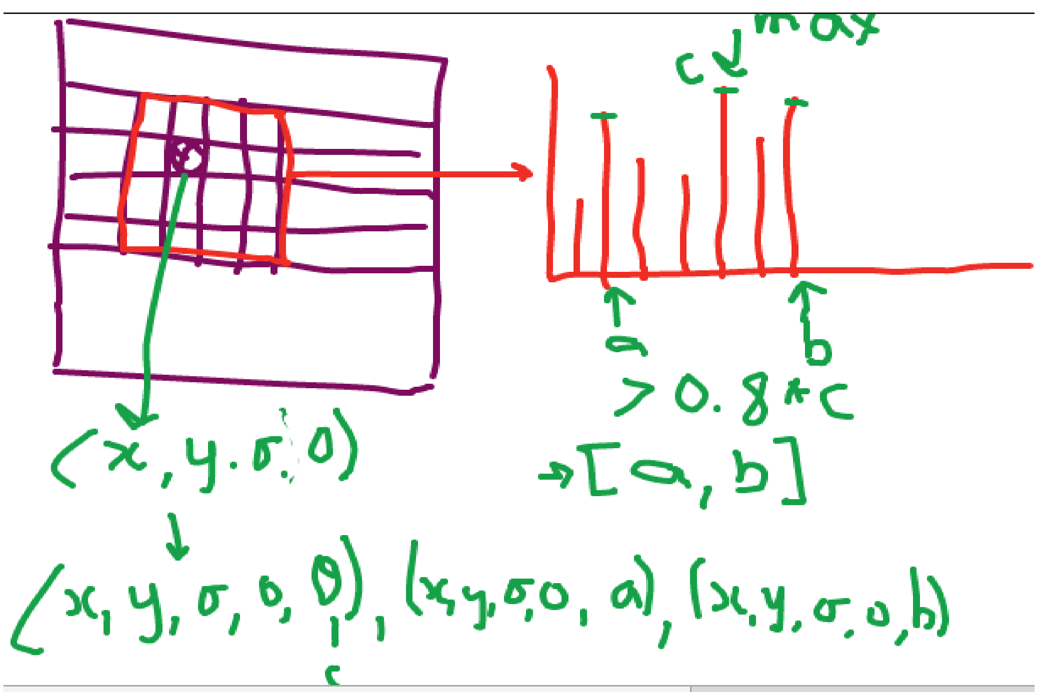
L = smoothed image at scale = gaussian (not dog)
dx = L(x+1,y) – L(x-1,y)   [-1 0 1]
dy = L(x,y+1) – L(x,y-1)   [-1 0 1]
magnitude(x,y) = sqrt(dx*dx + dy*dy)
orientation(x,y) = atan(dy/dx)

- create a weighted direction histogram.

36 bins (meaning each bin covers 10 degrees i.e. 0-10, 10-20, 20-30, etc.)
orientation' = orientation/10 -> assign to the bin
weight = magnitude and spatial gaussian filter with sigma = 1.5 * scale of
? point



- find the dominant directions in the histogram. Maximum = dominant direction. Any direction having > 0.8*domiant direction, consider them as potential points also.



4. Keypoint descriptor
-describing the key point as a high dimensional vector

- take 16x16 pixel window around the point
- compute magnitude and relative orientation (=orientation -dominant direction) for every pixel
- 16x16 divide into 4x4 regions. So, every block contains 4x4 pixels

4x4 pixels	4x4	4x4	4x4

- for every block, construct weighted histogram (8 bins, so 0-45,45-90, etc.,weight=magnitude and spatial gaussian)
- concatenate all the histograms. 8Bins x 16 blocks=> 128 vector

postprocessing
- normalize 128D vector to unit vector

[20,40,30,10] -> [20,40,30,10]/ 100
-> [0.2,0.4,0.3,0.1]
- for non-linear intensity transform, remove any value greater than 0.2 in the normalized vector and again renormalize ?

[0.2,0,0,0.1] -> [0.66,0,0,0.33]

5. keypoints matching
-match the keypoints against database or another image

feature1 -> 128D vector in image 1
feature2 -> 128D vector in image2

euclidean distance = sqrt((f1[1]-f2[1])^2 + (f1[2] -f2[2])^2 + ...)

image1 – featuresi1, featuresi2, features i3,...
image2 – featuresj1, featuresj2, features j3,...

- take featuresi1
- compute euclidean distance wrt featuresj1, featuresj2, featuresj3,...
- find feature with minimum euclidean distance --> matching feature1
- ratio = distance best match / distance 2nd best match
- consider the second best match (2nd minimum distance) if ratio <=0.8
- threshold distance, so both have to be less than threshold distance