Canny Edge Detector

Edge Detection

What is edge? (discontinuity) sharp changes of intensity

How many types of edges?
① surface normal
② depth
③ surface color
④ illumination

How to charaterize edges? large derivative

$$f'(x) = \frac{f(x+1) - f(x-1)}{2}$$

1
2
1

Huming

* (1D derivative)

1	0	-1

=

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

Horizontal Sobel filter S_x

$$\frac{\partial f}{\partial x} = S_x \otimes f, \quad \frac{\partial f}{\partial y} = S_y \otimes f$$

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right], \quad \theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right), \quad \|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

S_y

DoG filter : non-zero yield a thick edge
LoG filter : zero-crossing yield a thin edge

$$f''(x) = \frac{f(x+1) - 2f(x) + f(x-1)}{4}$$

1	-2	1

laplace filter

Canny Edge Detector

1. filter image with x, y derivatives of Gaussian
2. ① gradient magnitude = $\sqrt{(X' \text{Dof} G)^2 + (Y' \text{Dof} G)^2}$

   ② gradient orientation = theta = atan2(yDeriv, xDeriv)
3. Non-maximum suppression for each orientation { map orientation or interpolation
4. "Hysteresis" Thresholding (double thresholding) { > high threshold ⟹ strong edge; < low threshold ⟹ noise; in between ⟹ weak edge

link edge from strong edge to weak edge, only if weak edge near strong edge.

Corner Detection

interest point (keypoint) ⟹ find corner (gradients in two different orientation)

why not lines? (aperture problem)

method: shift window in any direction should give a large change of intensity.

$$E_{WSSD}(u,v) = [u,v]\, M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \sum_x \sum_y w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

(original $E_{WSSD}(u,v) = \sum_x \sum_y w(x,y)[I(x+u,y+v) - I(x,y)]^2$

(weighted summed square difference)

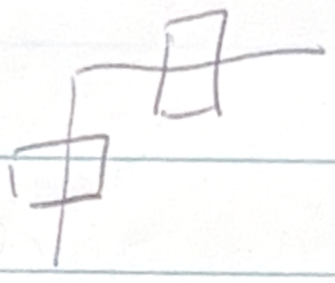
eigen-decomposition ⟹ diagonalize to $M = V \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} V^T$

a special case of SVD

corresponding to λ_1, λ_2 (eigenvalue), the amount of intensity change

orthogonal 正交的

column of V: direction of fastest and slowest changes,

example: edge :  $\lambda_1 \gg \lambda_2$ or $\lambda_2 \gg \lambda_1$

corner :  λ_1 and λ_2 are large $\lambda_1 \sim \lambda_2$

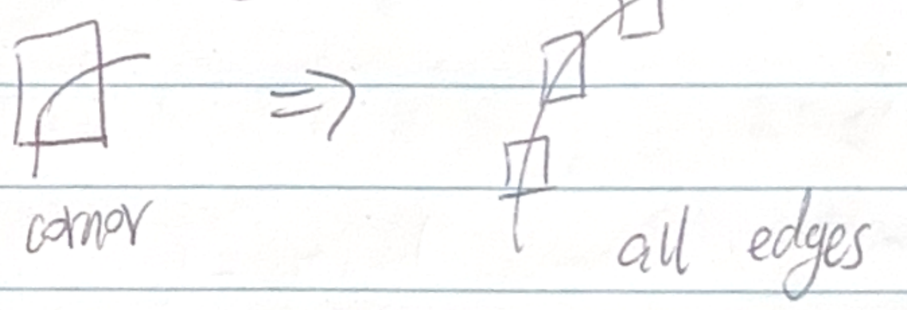
flat :  λ_1 and λ_2 are small

Harris & Stephens (1998): "Rotationally Invariant"

$$R = \lambda_1\lambda_2 - \alpha(\lambda_1+\lambda_2)^2 = \det(M) - \alpha\,\mathrm{trace}(M)^2$$

$R<0 \Rightarrow$ edge
$R>0 \Rightarrow$ corner
$|R|$ small $\Rightarrow$ flat region

" but not Scale Invariant "



Scale Invariant Keypoint Detection & Feature Descriptor

Goal = Matching objects / Images

Detection = ① find interest point of each image (independently, scale invariant).

Signature Function : ex. Blob Detection - LoG

Characteristic Scale is the scale that produce peak (max or min) of the Laplacian response.

$$\nabla^2 g(x,y,\sigma) = \frac{\partial^2 g(x,y,\sigma)}{\partial x^2} + \frac{\partial^2 g(x,y,\sigma)}{\partial y^2}$$

approximation = Difference of Gaussian

↳ Lowe's DoG (why? seperable convolution?)

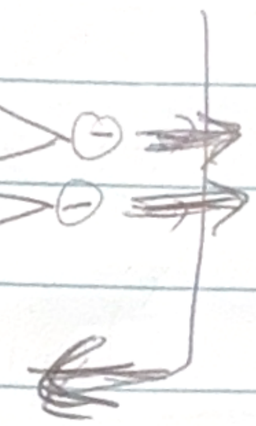
Lowe's DoG:

① compute a Gaussian image pyramid (many octaves)

ex: 2st octave ↑ down sample
1st octave

② compute <u>difference of Gaussians</u>

ex: $I_2 = I * G_{k^2\sigma}$
$I_1 = I * G_{k\sigma}$
$I_0 = I * G_\sigma$



③ at every scale
④ find local maxima in scale
⑤ a bit of pruning of bad maxima

$$D(x,y,\rho) = I(x,y) * (G(x,y,k\rho) - G(x,y,\rho))$$

for $\rho = \{\sigma, k\sigma, k^2\sigma, \dots, k^{s-1}\sigma\}$ $k = 2^{1/s}$

s = ~~n~~ numbers of layers in one octave.

Description ⑥ Extract a <u>feature descriptor</u> around each interest point.
(repeatable, distinctive, compact, efficient)
invariant not too high dimension

SIFT (scale invariant feature transform)

1. just take Gaussian-blurred image, and compute gradient magnitude and orientation.

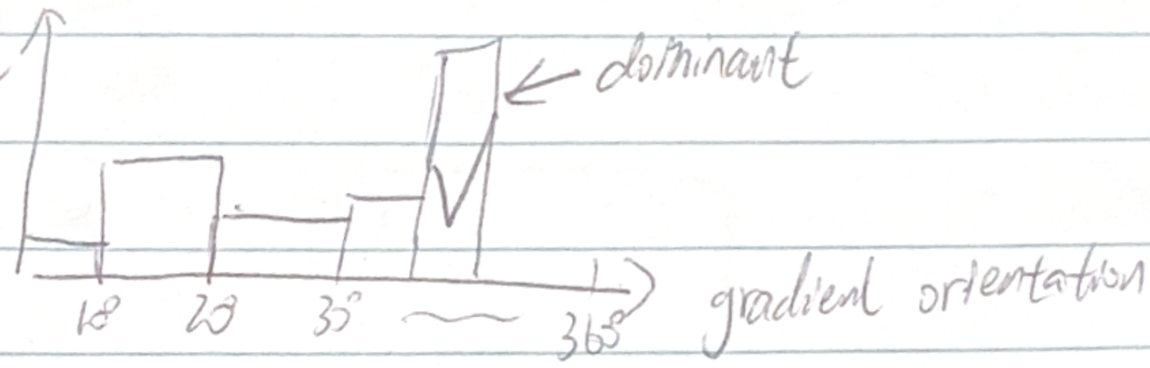
ex:
$I_2 = I * G_{k^2\sigma}$
$I_1 = I * G_{k\sigma}$
$I_0 = I * G_\sigma$

↙ just take this and compute ① gradient magnitude ② gradient orientation

2. compute Dominant Orientation

histogram of gradient orientation, increment of 10°

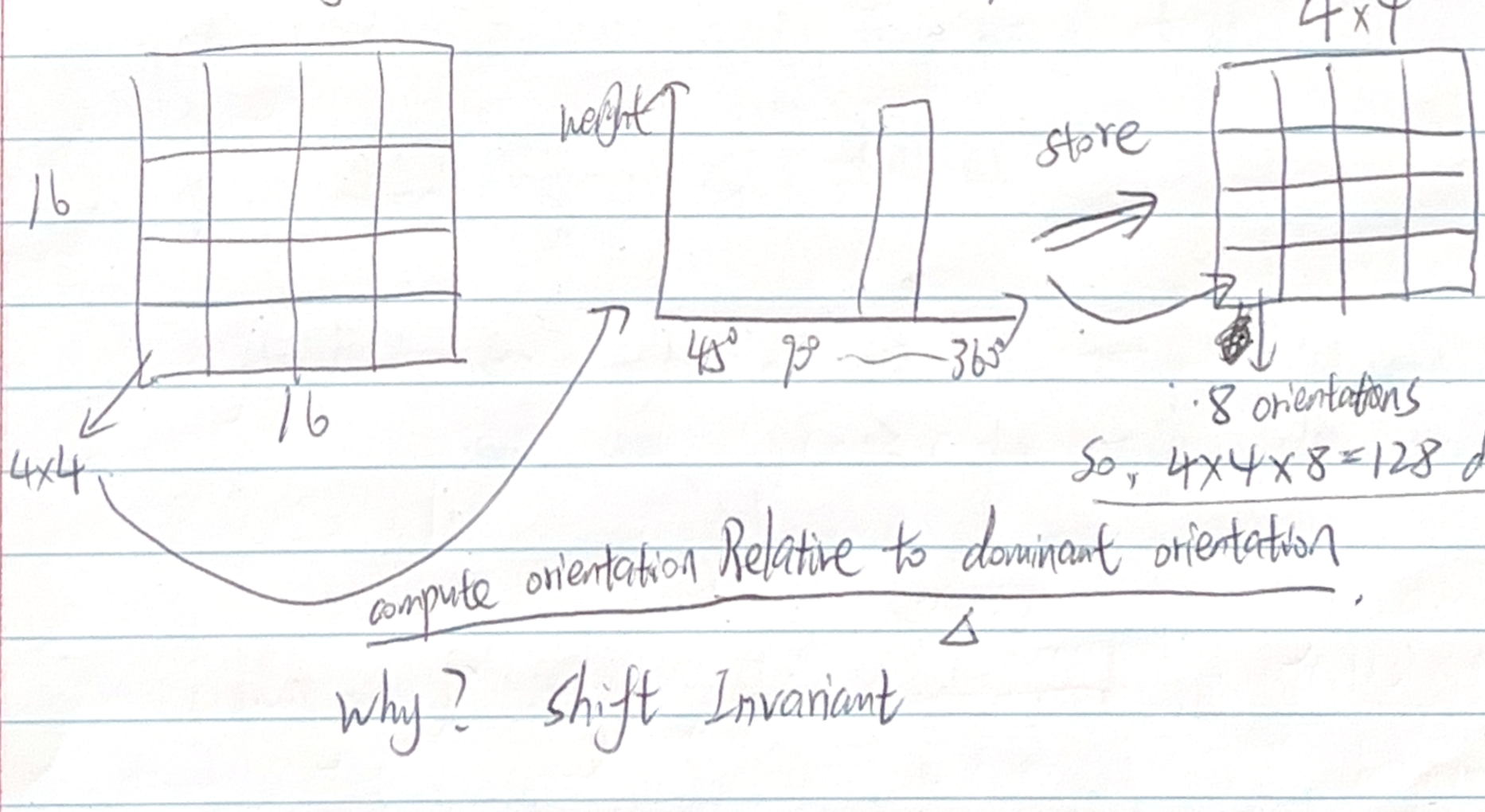
ex. 16x16



actually orientation closer to keypoint center contributes more, because we used Gaussian filter to blur.

3. compute the Feature Vector

ex: divide 16x16 to many 4x4 grid, in which compute histogram of orientation for 8 orientation bins spaced apart by 45°, so we can form a 4x4 SIFT descriptor.



So, 4x4x8=128 dimensional vector

compute orientation Relative to dominant orientation.

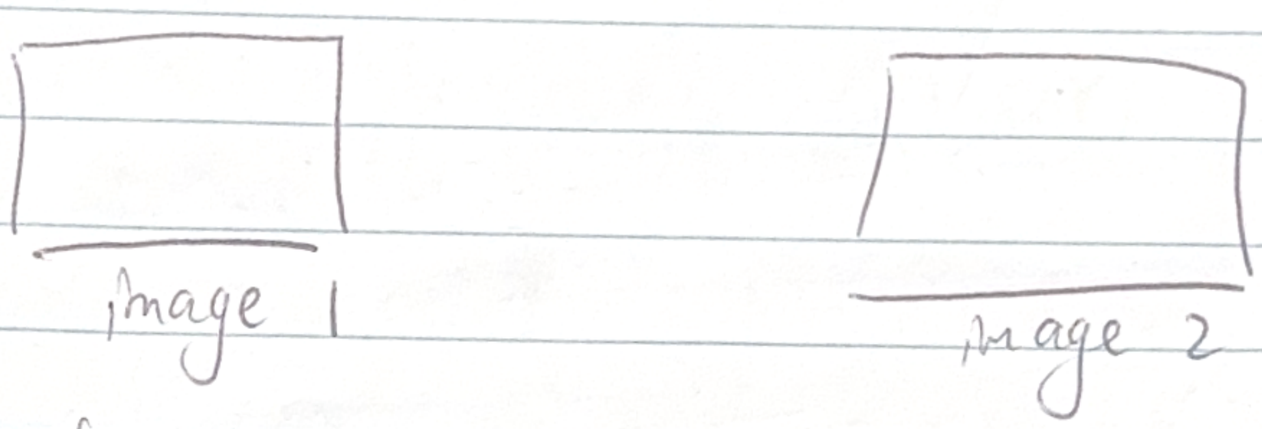
△

Why? Shift Invariant

4. Post-processing

1. normalize the 128-d vector to unit length
2. value clip to 0.2, the renormalization again.
(3.) use PCA (principal component analysis) to reduce dimensions to 10-d

Match ③ compute Euclidean distance for every SIFT descriptor of the first image to every ~~~ of the second image.



find the ~~closest~~ closest and the second closest.

$$\phi = \frac{\text{closest}}{\text{second closest}} < \text{threshold}$$

threshold too high ⟹ too many false positive
too low ⟹ too many false negative

problem : false positive, which need to be solved ~~when~~ in homography transformation.

(more accuracy, set a very small threshold for error)