시험: 10/20 화, 수업시간,
손으로 써서, 카메라(휴대폰노트북, 웹캠).
수업: 필요한경우, 금요일 1학시반 보강.

# Lecture 05: Feature Detection and Matching II

[AIX7021] Computer Vision

---

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

SIFT, { feature detection, descriptor, matching }

# SIFT: overview

- **Scale Invariant Feature Transform** (SIFT) is an image descriptor for image-based matching and recognition developed by David Lowe (1999, 2004). This descriptor as well as related image descriptors are used for a large number of purposes in computer vision, e.g., point matching between different views of a 3-D scene and view-based recognition. The SIFT descriptor is invariant to

  - translations
  - rotations
  - scaling transformations

  in the image domain and robust to moderate perspective transformations and illumination variations. It has been proven to be very useful in practice for image matching and object recognition under real-world conditions.

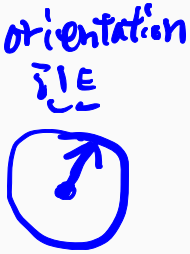
# SIFT: overview


Publication

- D. G. Lowe, “Object recognition from local scale-invariant features,” in Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV), pp. 1150–1157, 1999. [doi]
- D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” International Journal of Computer Vision, no. 60, vol. 2, pp. 91–110, 2004.

# SIFT: overview

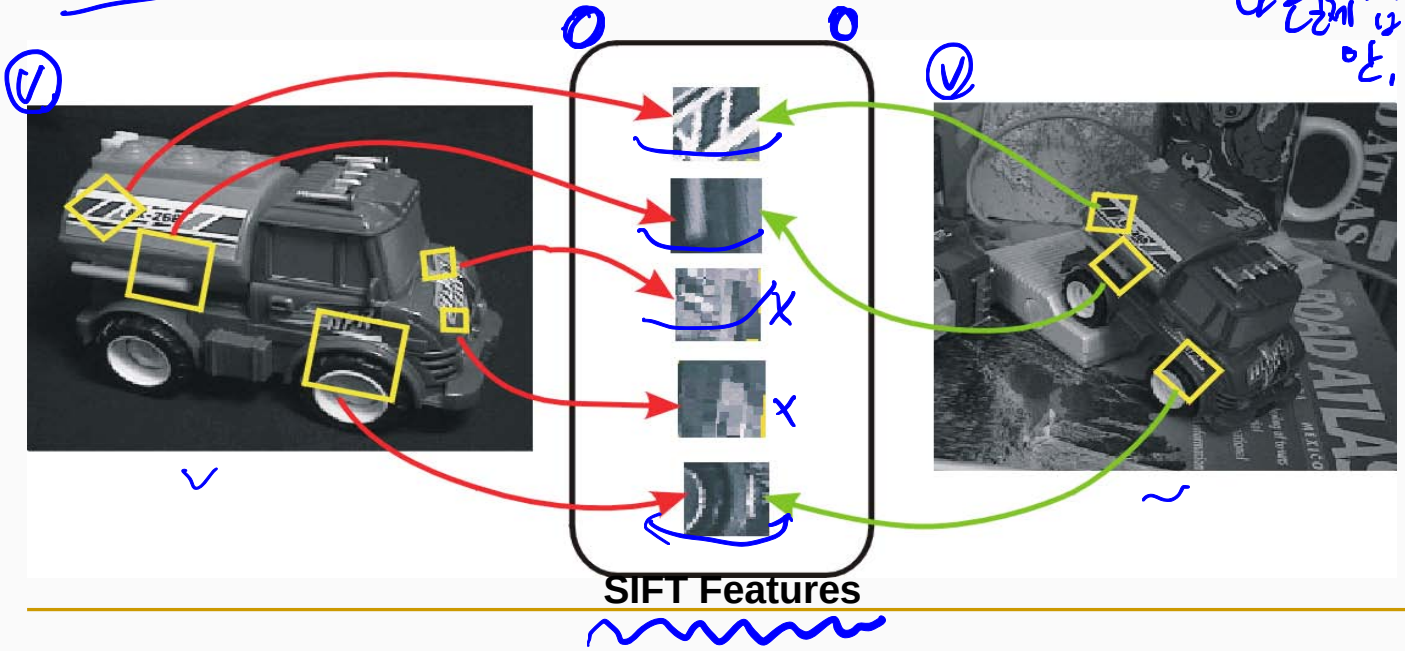
The SIFT descriptor comprised a method for detecting interesting points from a grey-level image at which statistics of local gradient directions of image intensities were accumulated to give a summarizing description of the local image structures in a local neighbourhood around each interest point, with the intention that this descriptor should be used for matching corresponding interest points between different images.



Later, the SIFT descriptor has also been applied at dense grid which have ben shown to lead to better performance for tasks such as object categorization, texture classification, image alignment, and biometrics.

## Idea of SIFT

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



## Claimed Advantages of SIFT

- **Locality:** features are local, so robust to occlusion and clutter (no prior segmentation)
- **Distinctiveness:** individual features can be matched to a large database of objects
- **Quantity:** many features can be generated for even small objects
- **Efficiency:** close to real-time performance
- **Extensibility:** can easily be extended to wide range of differing feature types, with each adding robustness

# SIFT: overview

Procedure.


여러단계변화 ↕ pyramid를 도입하여 각각 scale space에서 수행.

1. **Scale-space extrema detection**: search over multiple scales and image locations
2. **Keypoint localization**: fit a model to determine location and scale

   suppression, Hessian, Gradient norm. // scale.
3. **Orientation assignment**: compute best orientation(s) for each keypoint region

   one or more
4. **Keypoint description**: use local image gradients at selected scale and rotation to describe each keypoint region

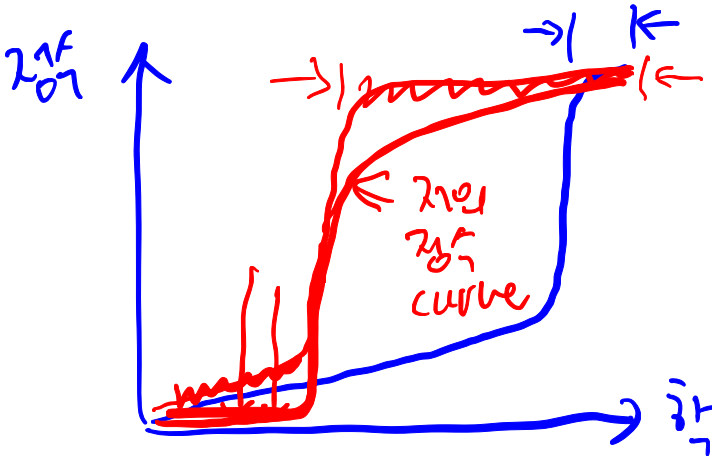
↓

matching

# SIFT: overview

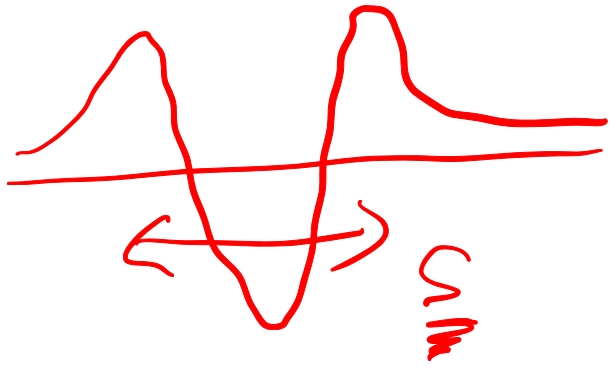
1. Interest point detection
   - 1.1 Scale-invariant interest points from scale-space extrema
   - 1.2 Interpolation
   - 1.3 Suppression of interest point responses along edges
2. Image descriptor
   - 2.1 Scale and orientation normalization
   - 2.2 Weighted position-dependent histogram of local gradient directions
   - 2.3 ...

# Interest point detection




$G(x,y;s)$

$\nabla^2 G$



# Scale-invariant interest points from scale-space extrema

The original SIFT descriptor was computed from the image intensities around interesting locations in the image domain which can be referred to as interest points, alternatively key points. These interest points are obtained from scale-space extrema differences-of-Gaussians (DoG) within a difference-of-Gaussian pyramid.

≃ Laplacian of Gaussian

A Gaussian pyramid is constructed from the input image by repeated smoothing and subsampling, and a difference-of-Gaussians pyramid is computed from the differences between the adjacent levels in the Gaussian pyramid. Then, interest points are obtained from the points at which the difference-of-Gaussians values assume extrema with respect to both the spatial coordinates in the image domain and the scale level in the pyramid.

# Scale-invariant interest points from scale-space extrema

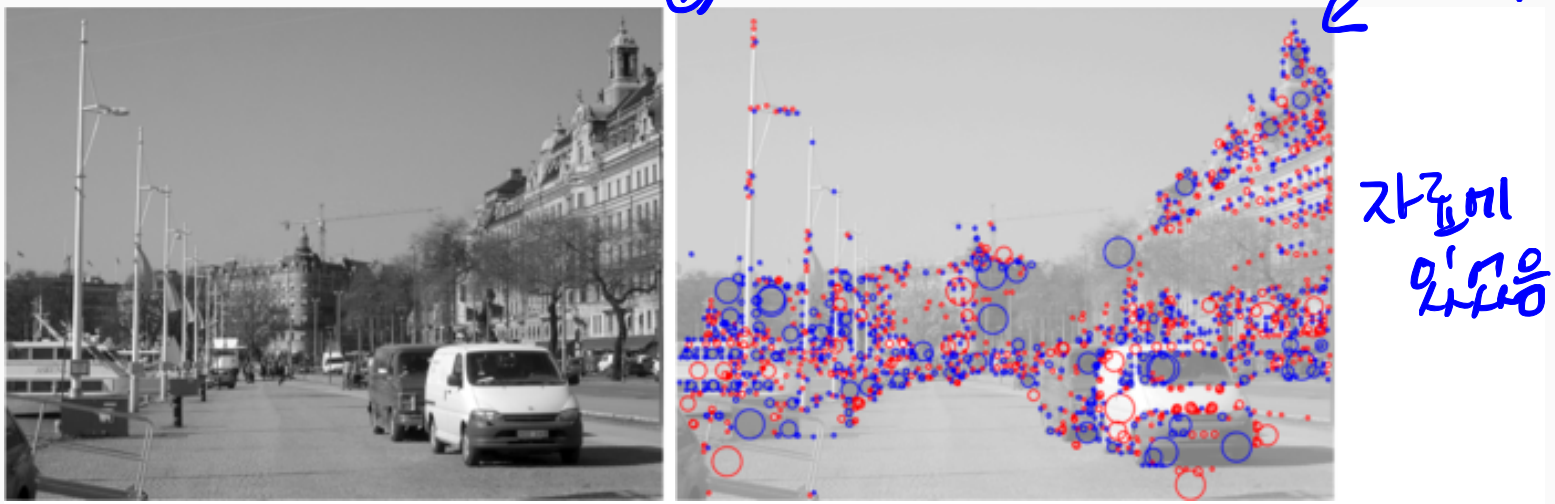
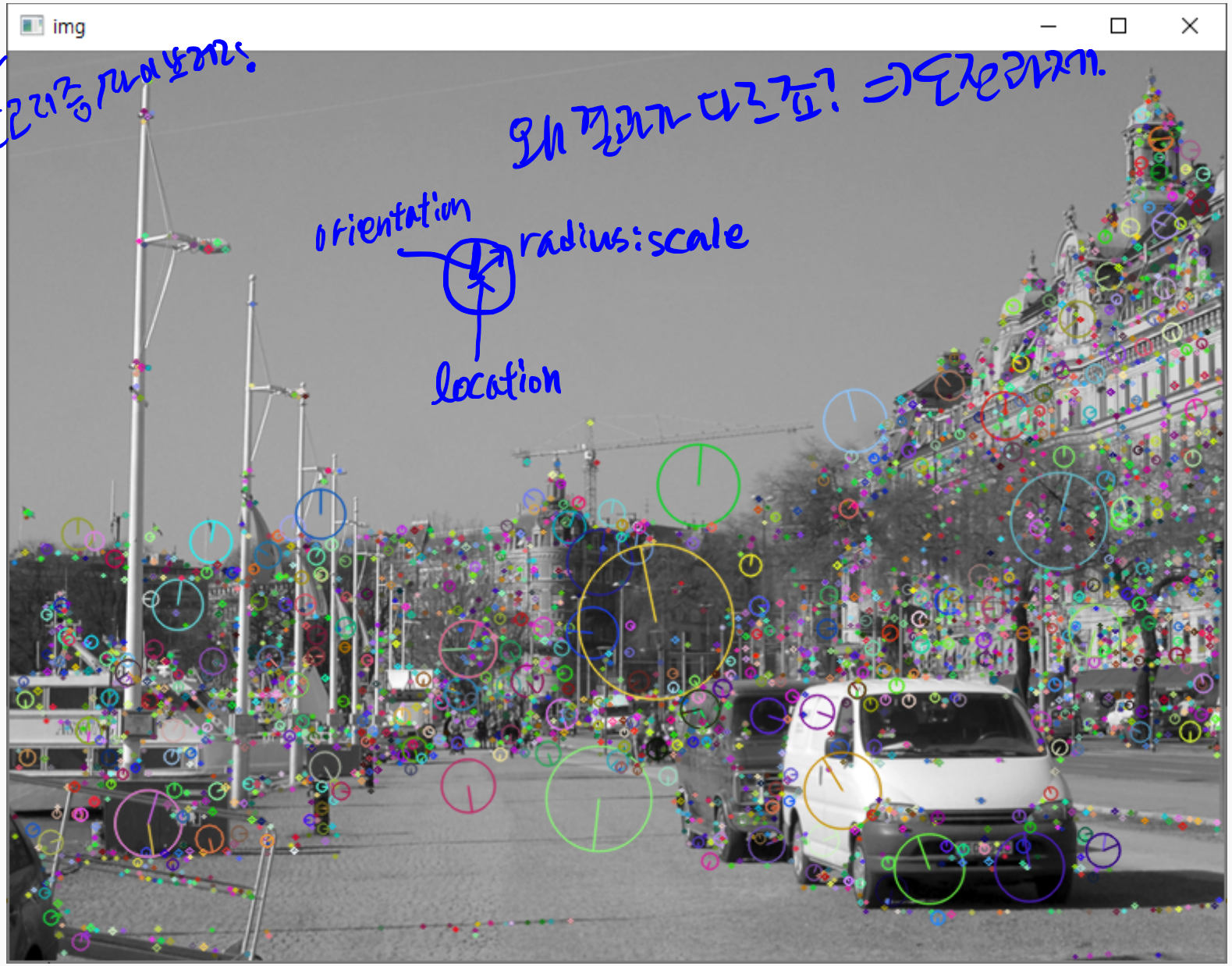


**Figure 1:** Scale-invariant interest points detected from a grey-level image using scale-space extrema of the Laplacian. The radii of the circles illustrate the selected detection scales of the interest points. Red circles indicate bright image features with $\nabla^2 L < 0$, whereas blue circles indicate dark image features with $\nabla^2 L > 0$.

# Scale-invariant interest points from scale-space extrema



# Scale-invariant interest points from scale-space extrema

This method for detecting interest points in the SIFT operator can be seen as a variation of a scale-adaptive blob detection method, where blobs with associated scale levels are detected from *scale-space extrema* of the scale-normalized Laplacian. The scale-normalized Laplacian is normalized with respect to the scale level in scale-space and is defined as

LoG filter.

$$\nabla^2_{norm} L(x, y; s) = s(L_{xx} + L_{yy}) = s\left(\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}\right) = s\nabla^2(G(x, y; s) * f(x, y)) \tag{1}$$

$f(x,y) \rightarrow \nabla^2_{norm} L(x,y;s) = s\nabla^2(G * f) = s(\nabla^2 G) * f$

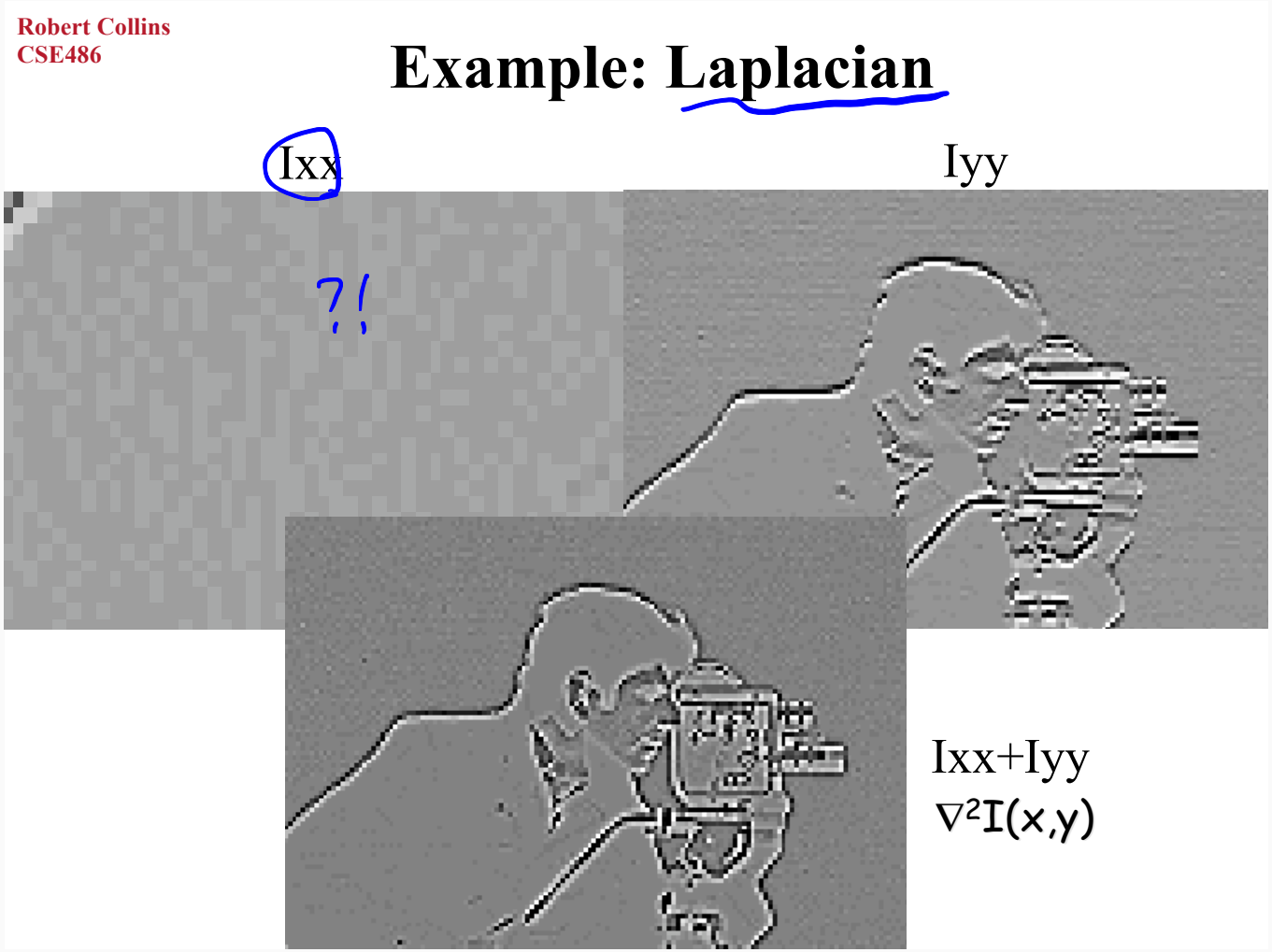
norm → scale normalization

from smoothed image values $L(x, y; s)$ compute from the input image $f(x, y)$ by convolution with Gaussian kernels

$$G(x, y; s) = \frac{1}{2\pi s} \exp\left(-\frac{x^2 + y^2}{2s}\right) \tag{2}$$

of different width $s = \sigma^2$.

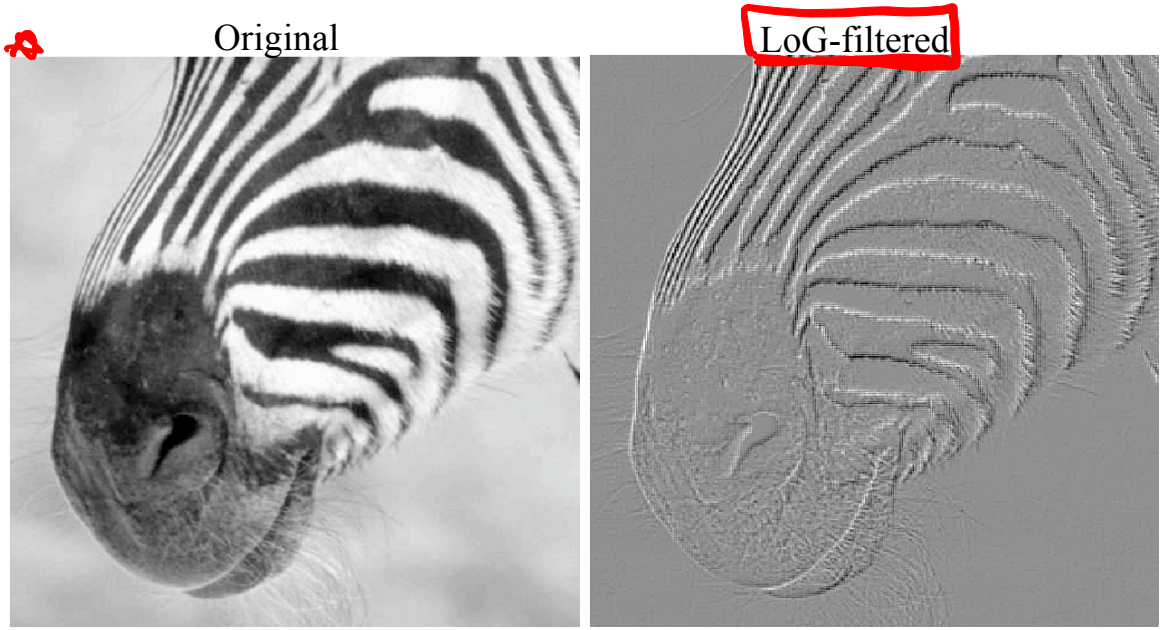
# Scale-invariant interest points from scale-space extrema



# Scale-invariant interest points from scale-space extrema

Robert Collins
CSE486

## Effect of LoG Operator




Band-Pass Filter (suppresses both high and low frequencies)

Why? Easier to explain in a moment.

Robert Collins
CSE486

# Zero-Crossings as an Edge Detector

Raw zero-crossings (no contrast thresholding)

LoG sigma = 2, zero-crossing

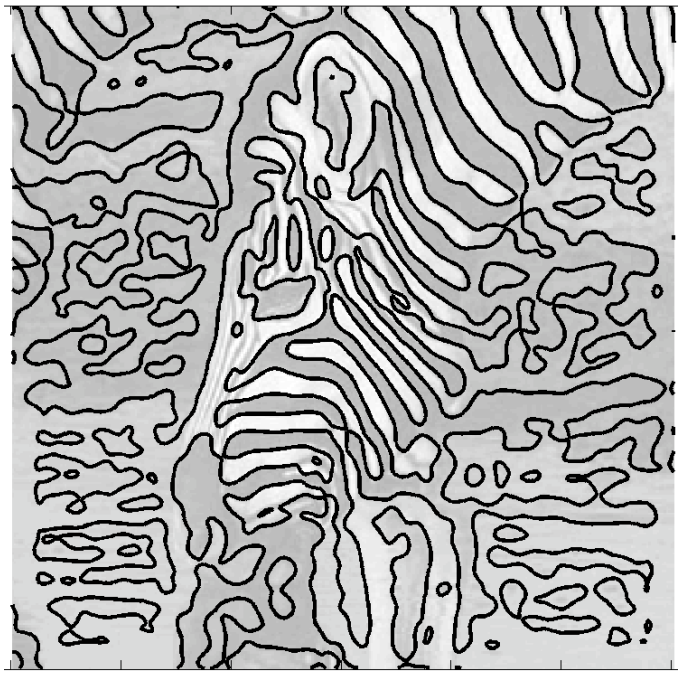
LoG ~ edge corner blob.

$\sigma^2 = s$

Robert Collins
CSE486

## Zero-Crossings as an Edge Detector

Raw zero-crossings (no contrast thresholding)



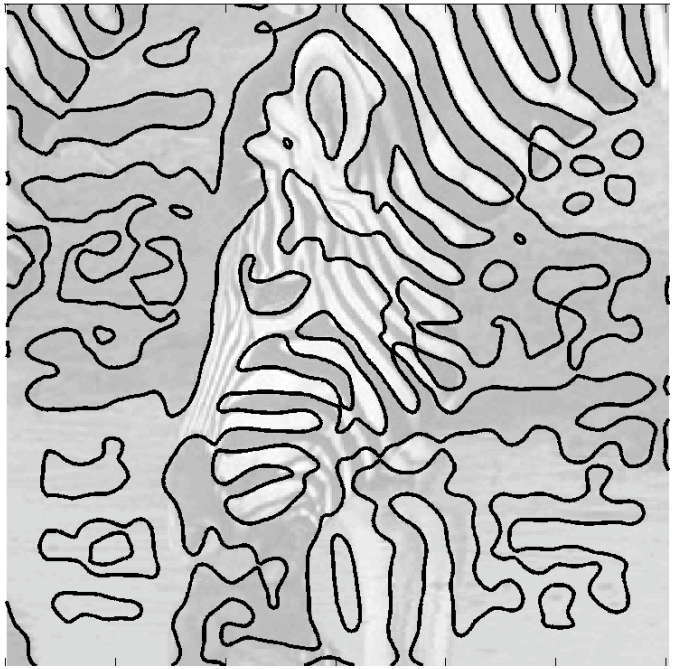
LoG sigma = 4, zero-crossing

5 x 4

Robert Collins
CSE486

## Zero-Crossings as an Edge Detector

Raw zero-crossings (no contrast thresholding)

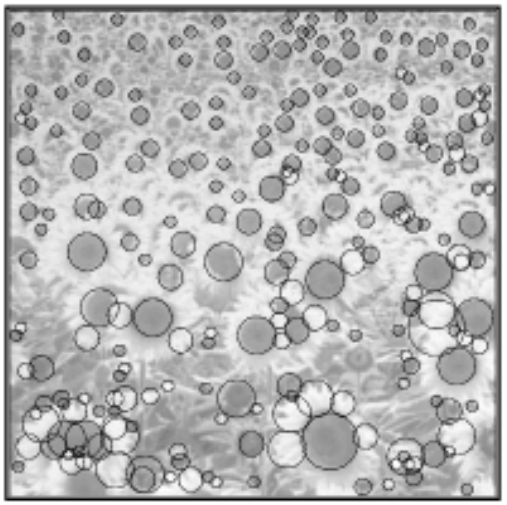


LoG sigma = 8, zero-crossing

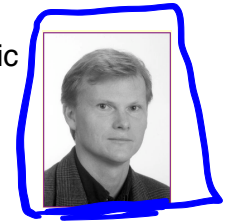
5 x 16

Robert Collins
CSE486

# Other uses of LoG: Blob Detection



Lindeberg: ``Feature detection with automatic scale selection''. International Journal of Computer Vision, vol 30, number 2, pp. 77--116, 1998.



< D. Lowe SIFT 이것이 blob을 한사람.

# Scale-invariant interest points from scale-space extrema

## Different of Gaussians on image pyramid

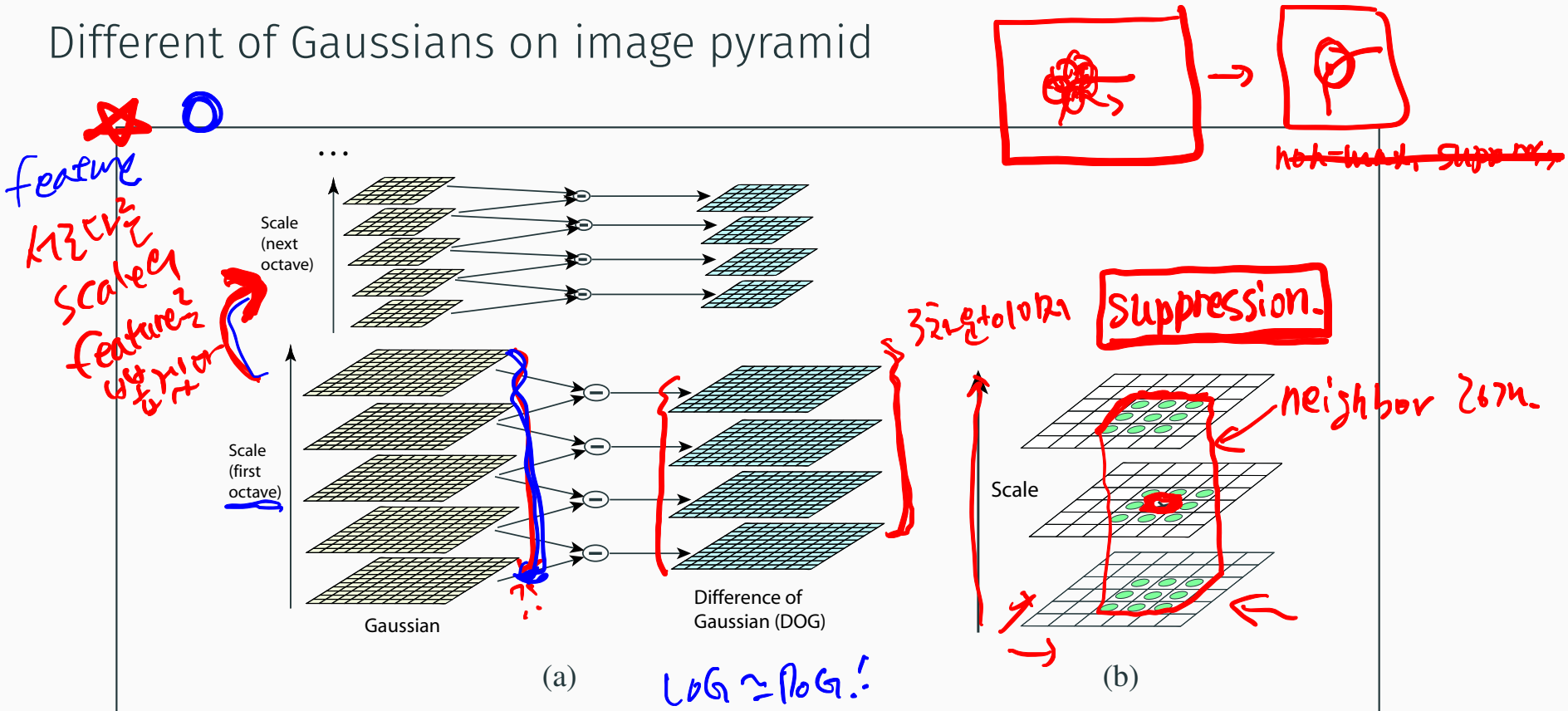
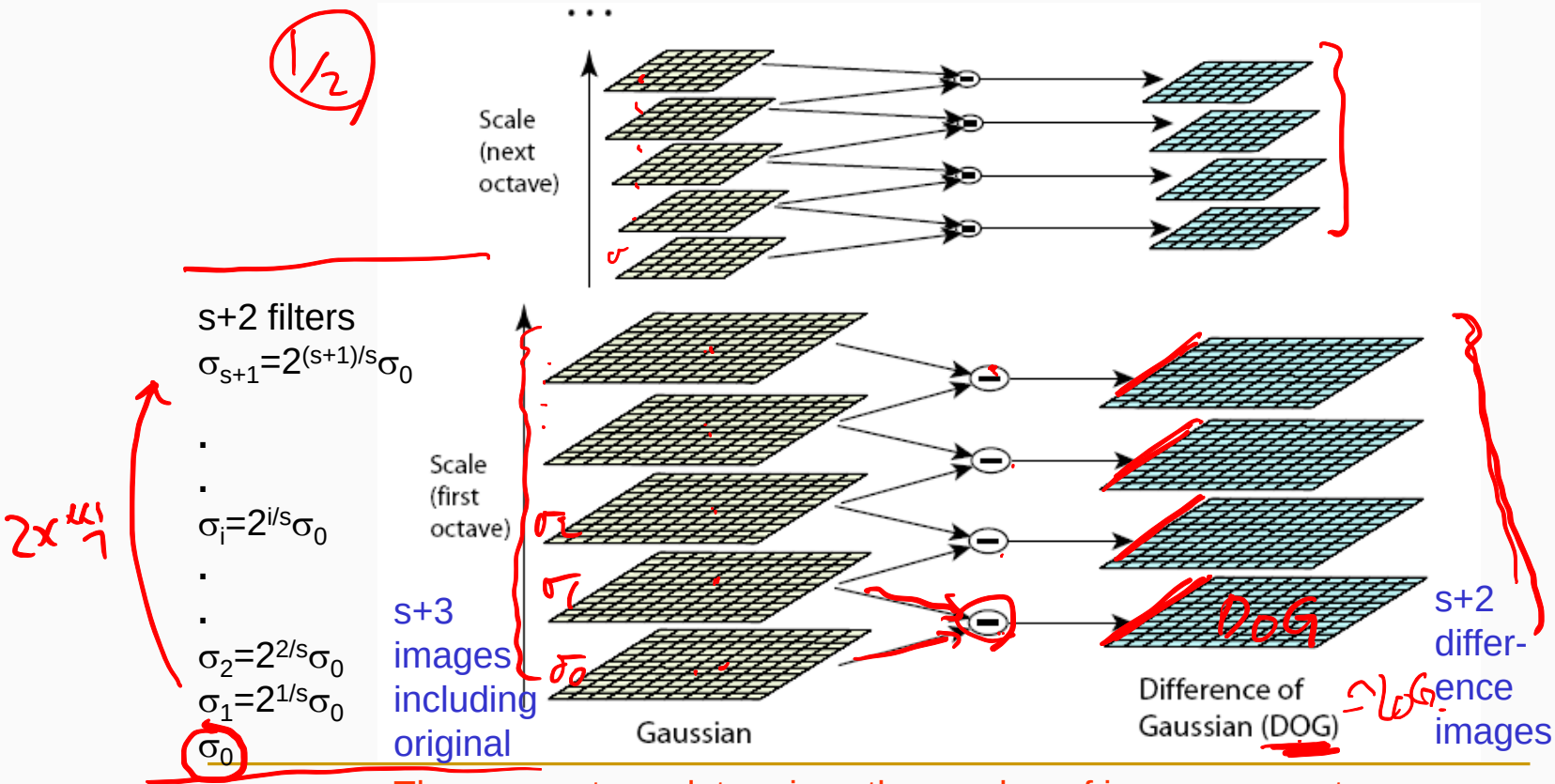


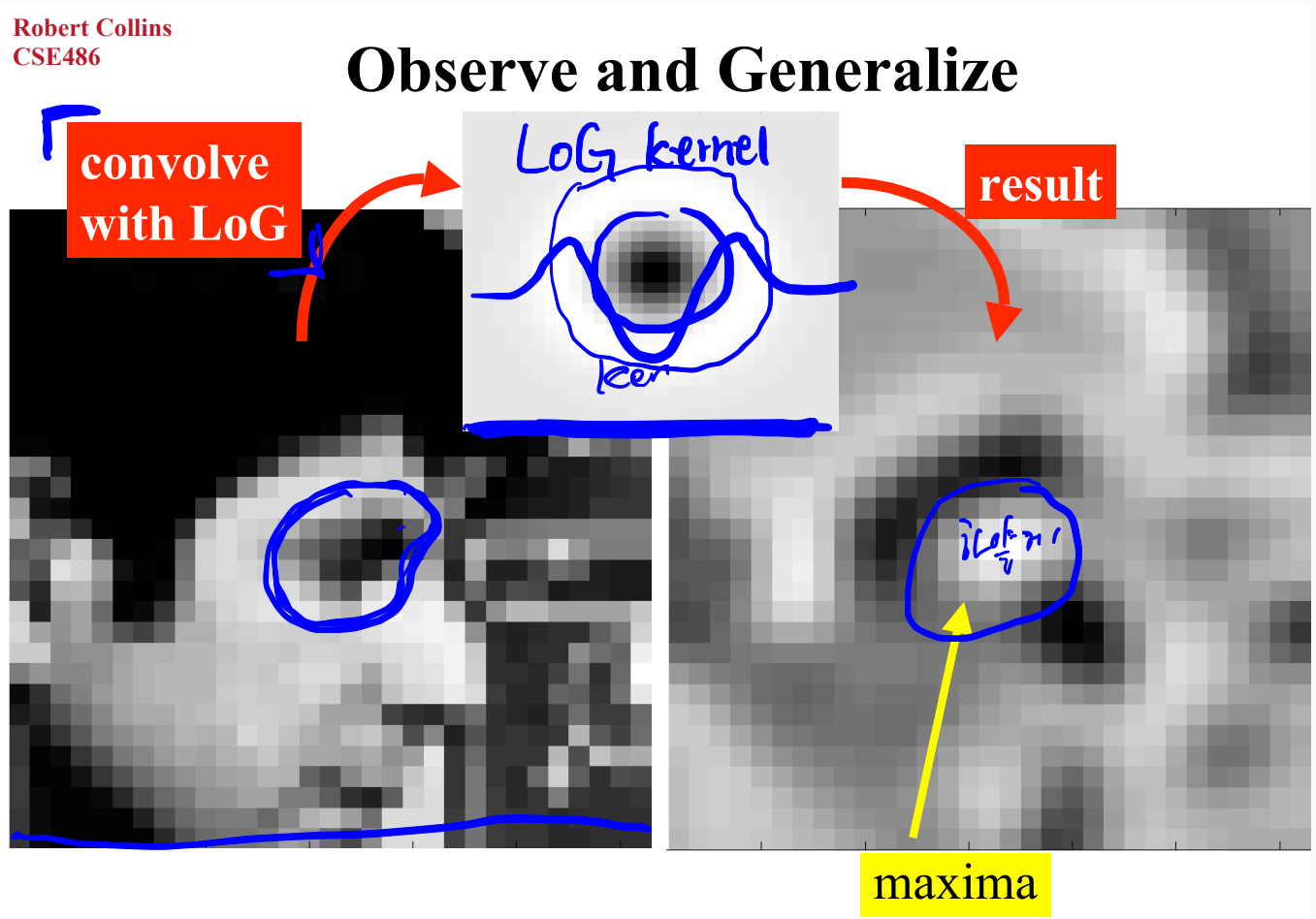

**Figure 4.11** Scale-space feature detection using a sub-octave Difference of Gaussian pyramid (Lowe 2004) © 2004 Springer: (a) Adjacent levels of a sub-octave Gaussian pyramid are subtracted to produce Difference of Gaussian images; (b) extrema (maxima and minima) in the resulting 3D volume are detected by comparing a pixel to its 26 neighbors.

# Lowe's Pyramid Scheme




The parameter **s** determines the number of images per octave.

# Scale-invariant interest points from scale-space extrema

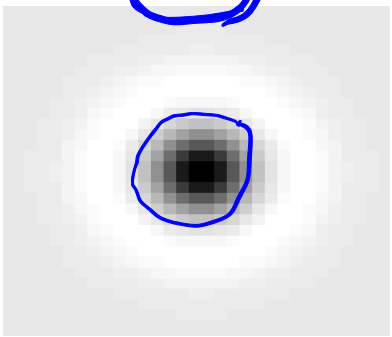



LoG 모양 ~ blob 방법.

Robert Collins
CSE486

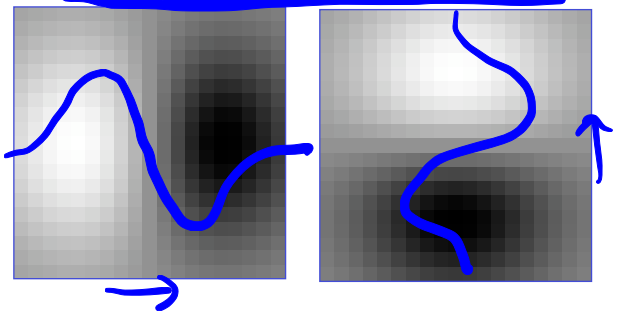
## Observe and Generalize

LoG



Looks like dark blob on light background

Derivative of Gaussian



Looks like vertical and horizontal step edges

Recall: Convolution (and cross correlation) with a filter can be viewed as comparing a little "picture" of what you want to find against all local regions in the mage.

# Scale-invariant interest points from scale-space extrema

Then, the scale-space extrema are detected from the points $(x, y; s)$ in scale-space at which the scale-normalized Laplacian assumes local extrema with respect to space and scale. In a discrete setting, such comparisons are usually made in relation to all neighbours of a point in a $3 \times 3 \times 3$ neighbourhood over space and scale. The difference-of-Gaussian operator constitutes an approximation of the Laplacian operator

$$DOG(x, y; s) = L(x, y; s + \Delta s) - L(x, y; s) \approx \frac{\Delta s}{2} \nabla^2 L(x, y; s) \qquad (3)$$

which by the implicit normalization of the differences-of-Gaussian responses, as obtained by a self-similar distribution of scale level $\sigma_{i+1} = k\sigma_i$ used by Lowe, also constitutes an approximation of the scale-normalized Laplacian with
$\Delta s \nabla^2 L = (k^2 - 1) t \nabla_L^2 = (k^2 - 1) \nabla_{norm}^2 L$, thus implying

$$DOG(x, y; s) \approx \frac{(k^2 - 1)}{2} \nabla_{norm}^2 L(x, y; s) \qquad (4)$$

# Interpolation

Both the difference-of-Gaussians approach by Lowe and the Laplacian approach by Lindeberg and Bretzner involve the fitting of a quadratic polynomial to the magnitude values around each scale-space extremum to localize the scale-space extremum with a resolution higher than the sampling density over space and scale. This post-processing stage is in particular important to increase the accuracy of the scale estimates for the purpose of scale normalization.

Robert Collins
CSE486

## Efficient Implementation
## Approximating LoG with DoG

$\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)$

LoG can be approximate by a Difference of two Gaussians (DoG) at different scales

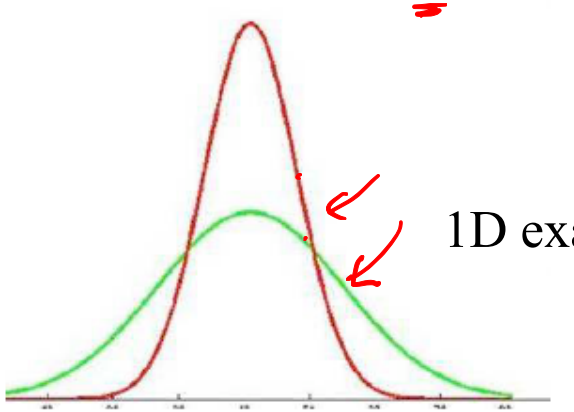
$\nabla^2 G_\sigma \approx G_{\sigma_1} - G_{\sigma_2}$

DoG

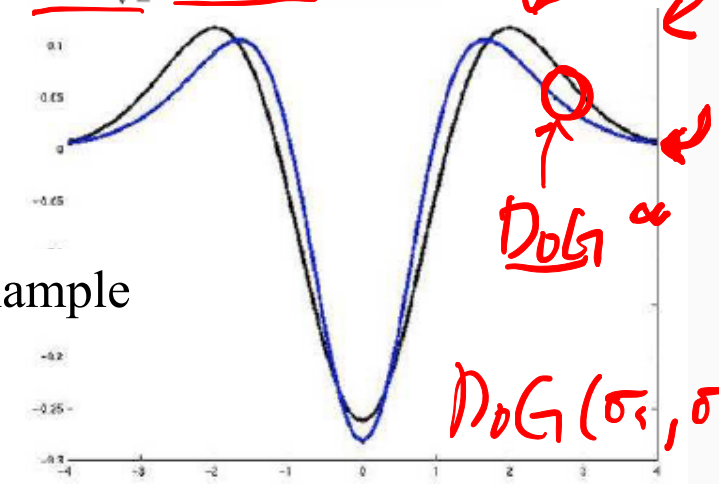
$\sigma \rightarrow 2\sigma$

LoG = $\nabla^2 G$

Best approximation when:
$\sigma_1 = \frac{\sigma}{\sqrt{2}}$, $\sigma_2 = \sqrt{2}\sigma$



1D example



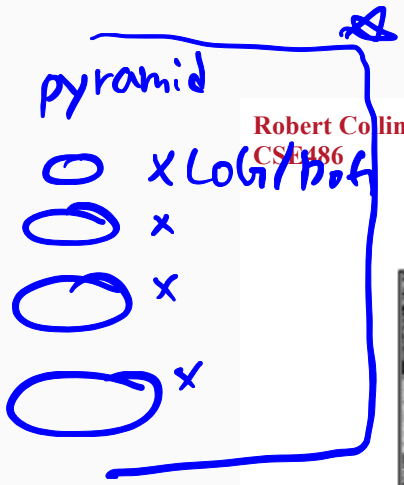
LoG(σ)

DoG ∝

DoG($\sigma_1, \sigma_2$)

LoG $\nabla^2$ 를 구하려고 하는데, 근사적인 ≃ scale 들을 가지고 DoG 를 구하려고 함.
$\sigma$, $2\sigma$, $4\sigma$, 등...

# Scale-invariant interest points from scale-space extrema

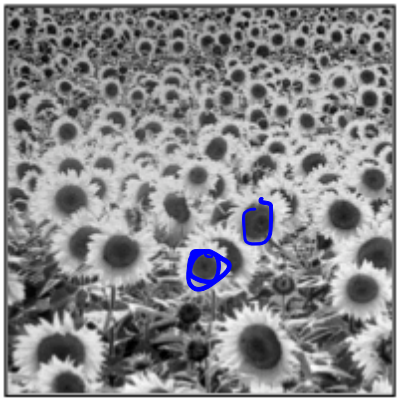


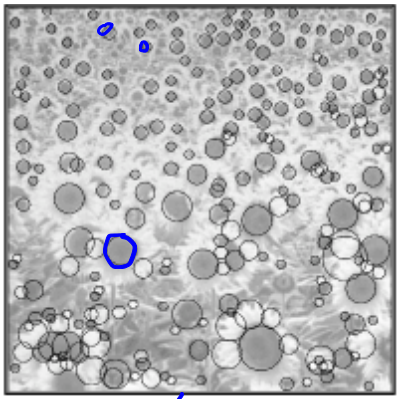

Robert Collins
CSE486

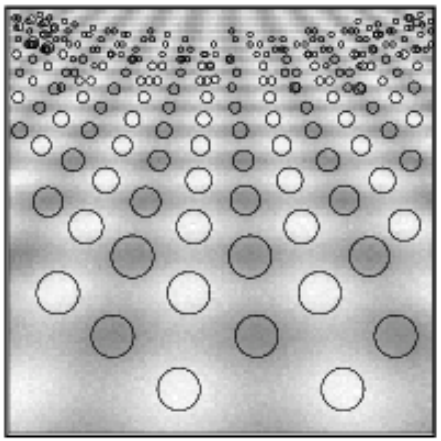
scale invariant.

## Back to Blob Detection

~ edge corner = 다른 방향의 SIFT 특징점 제안.







서로 다른 scale에 blob을 추출함.

← 인식기.

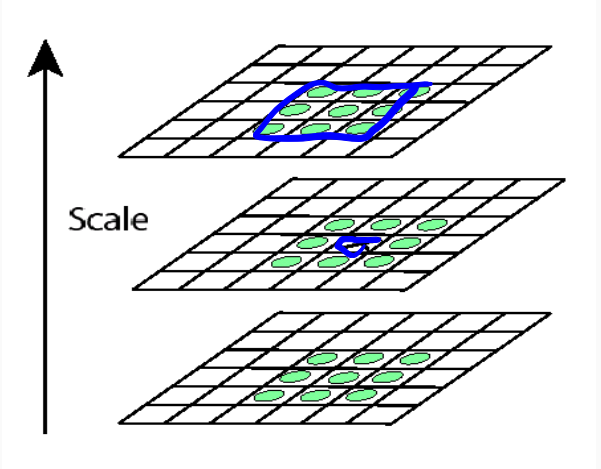
**Lindeberg: blobs are detected as local extrema in space and scale, within the LoG (or DoG) scale-space volume.**

# Key point localization

Suppression

s+2 difference images.
top and bottom ignored.
s planes searched.

- Detect maxima and minima of difference-of-Gaussian in scale space
- Each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below

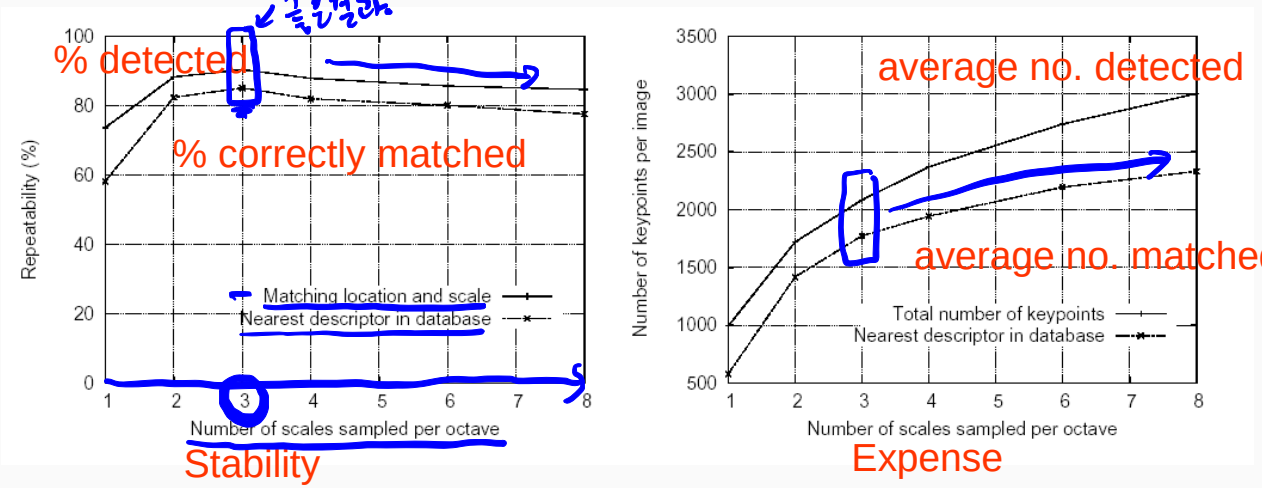



For each max or min found, output is the **location** and the **scale**.

3x3x3 ? en 3D(UF?

# Suppression of interest point responses along edges

Scale-space extrema detection: experimental results over 32 images that were synthetically transformed and noise added.




- Sampling in scale for efficiency
  - How many scales should be used per octave? S=?
    - More scales evaluated, more keypoints found
    - S < 3, stable keypoints increased too
    - S > 3, stable keypoints decreased
    - S = 3, maximum stable keypoints found

# Eliminating the Edge Response

- Reject flats:
  - $|D(\hat{\mathbf{x}})| < 0.03$
- Reject edges:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

Let $\alpha$ be the eigenvalue with larger magnitude and $\beta$ the smaller.

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

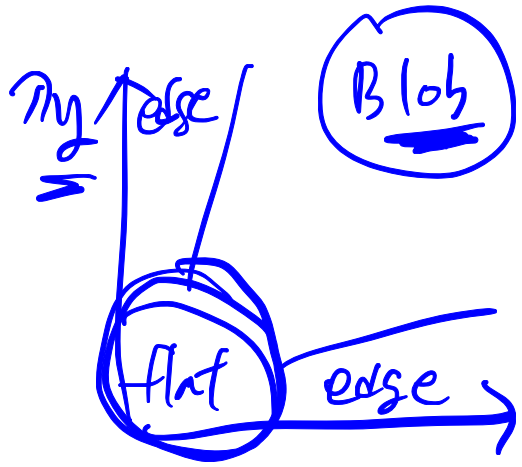
$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

Let $r = \alpha/\beta$. So $\alpha = r\beta$

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} \begin{matrix} \leq \\ > \end{matrix} \frac{(r+1)^2}{r},$$

$(r+1)^2/r$ is at a min when the 2 eigenvalues are equal.

- $r < 10$




$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

$\lambda_x \leftarrow eig(H)$

$$R = \det(H) - \alpha\, \text{trace}^2(H) \gtrless \text{threshold}$$

$$\det = \prod \lambda_i, \quad \text{trace} = \sum \lambda_i$$

# Suppression of interest point responses along edges

In addition to responding to blob-like and corner-like image structures, the Laplacian operator may also lead to strong responses along edges. To suppress such points, which will be less useful for matching, Lowe (1999, 2004) formulated a criterion in terms of the ratio between the eigenvalues of the Hessian matrix

$$H = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix} \tag{5}$$

computed at the position and the scale of the interest point, which can be reformulated in terms of the trace and the determinant of the Hessian matrix to allow for more efficient computations

$$\frac{\det(HL)}{\mathrm{trace}^2(HL)} = \frac{L_{xx}L_{yy} - L_{xy}^2}{(L_{xx} + L_{yy})^2} \geq \frac{r}{(r+1)^2} \tag{6}$$

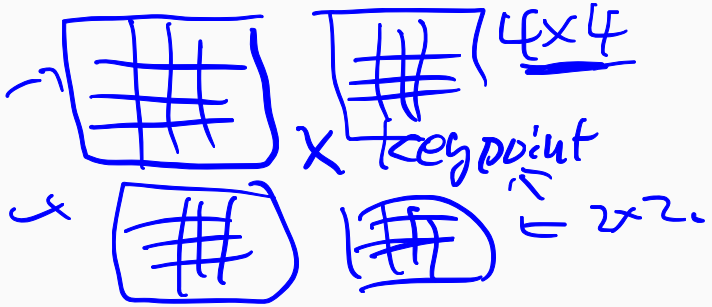
where $r \geq 1$ denotes an upper limit on the permitted ratio between the larger and the smaller eigenvalues. To suppress image features with low contrast, the interest points are usually also thresholded on the magnitude of the response.

# Image descriptor

# Image descriptor

feature / keypoint / image descriptor → matching

At each interest point as obtained above, an image descriptor is computed. The SIFT descriptor proposed by Lowe (1999, 2004) can be seen as a position-dependent histogram (statistic) of local gradient directions (angle orientation) around the interest point. To obtain scale invariance of the descriptor, the size of this local neighbourhood needs to be normalized in a scale-invariant manner. To obtain rotational invariance of the descriptor, a dominant orientation in this neighbourhood is determined from the orientations of the gradient vectors in this neighbourhood and is used for orienting the grid over which the position-dependent histogram is computed with respect to this dominant orientation to achieve rotational invariance.



# Scale and orientation normalization

In the SIFT descriptor, the size estimate of an area around the interest point is determined as a constant times the detection scale $s$ of the interest point.

To determine a preferred orientation estimate for the interest point, a local histogram of gradient directions is accumulated over a neighborhood around the interest point with (i) the gradient directions computed from gradient vector $\nabla L(x, y; s)$ at the detection scale $s$ of the interest point and (ii) the area of the accumulation window proportional to the detection scale $s$. To find the dominant orientation, peaks are detected in this orientation histogram. To handle situations where there may be more than one dominant orientation around the interest point, multiple peaks are accepted if the height of secondary peaks is above 80% of the height of the highest peak. In the case of multiple peaks, each peak is used for computing a new image descriptor for the corresponding orientation estimate.

# Weighted position-dependent histogram of local gradient directions

Given these scale and orientation estimate for an interest point, a rectangular grid is laid out in the image domain, centered at the interest point, with its orientation determined by the main peak(s) in the histogram and with the spacing proportional to the detection scale of the interest point. From experiments, Lowe (1999, 2004) found that a $4 \times 4$ grid is often a good choice. For each point on this grid, a local histogram of local gradient directions at the scale of the interest point

$$\angle \nabla L = \operatorname{atan2}(L_y, L_x) \tag{7}$$

$$|\nabla L| = \sqrt{L_x^2 + L_y^2} \tag{8}$$

# Weighted position-dependent histogram of local gradient directions

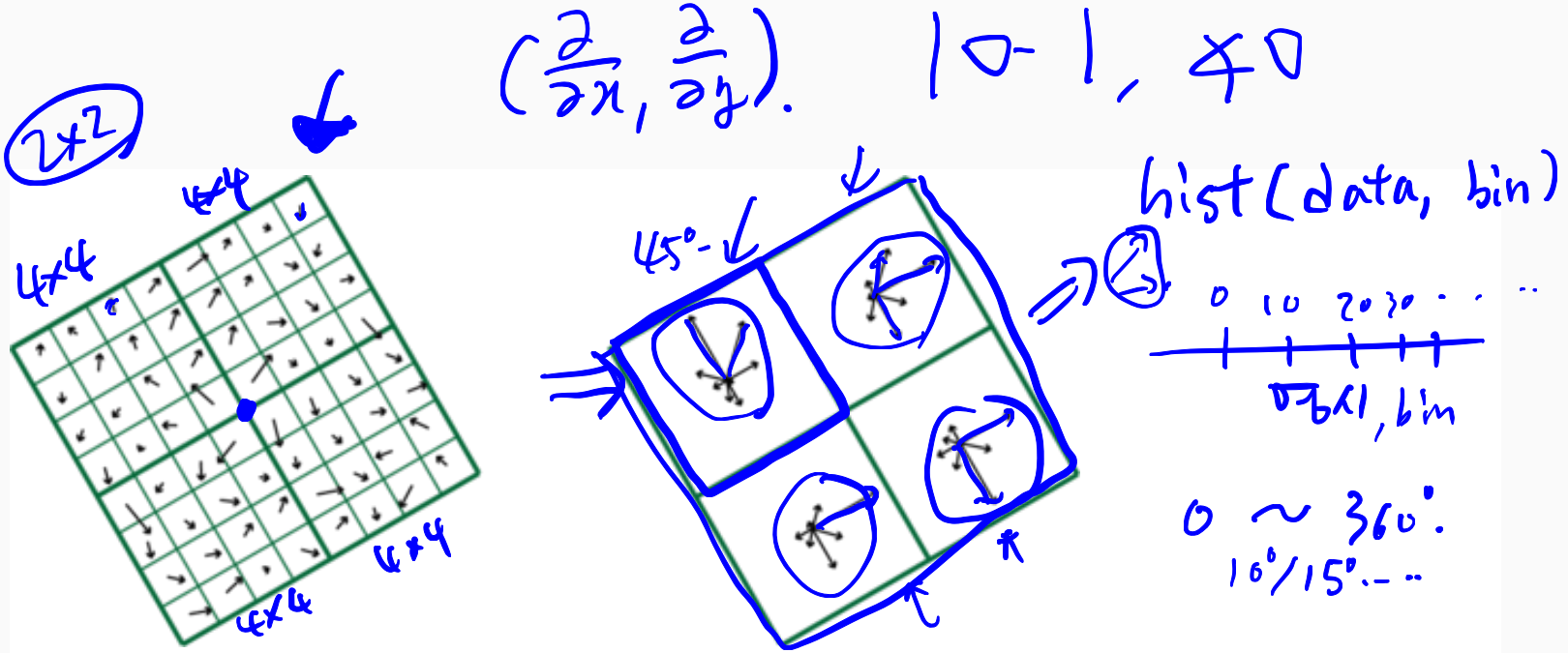
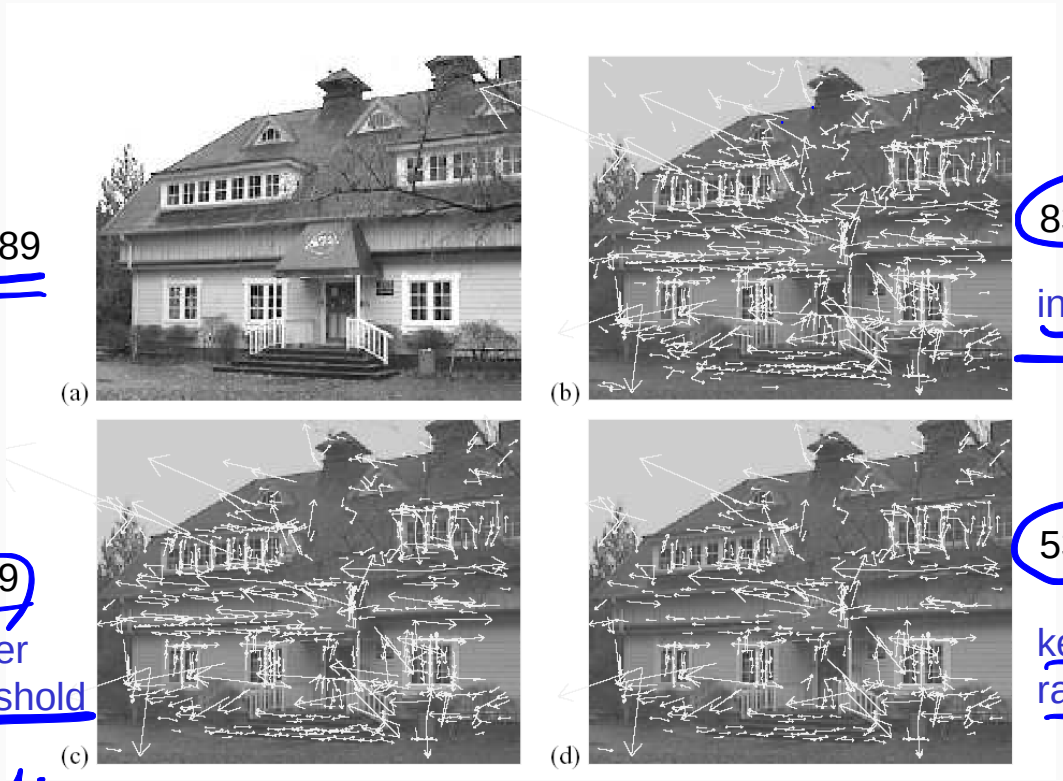


**Figure 2:** Illustration of how the SIFT descriptor is computed from sampled values of the gradient orientation and the gradient magnitude over a locally adapted grid around each interest point, with the scale factor determined from the detection scales of the interest point and the orientation determined from the dominant peak in a gradient orientation histogram around the interest point. This figure shows an image descriptor computed over a $2 \times 2$ whereas the SIFT descriptor is usually computed over a $4 \times 4$ grid.

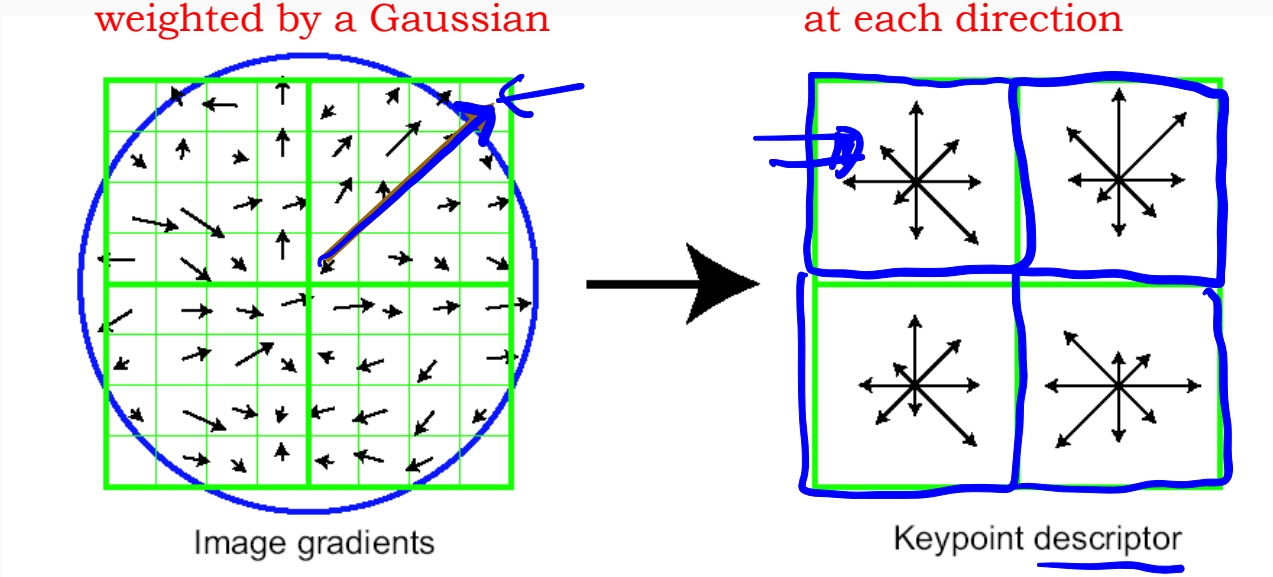
# Keypoint localization with orientation



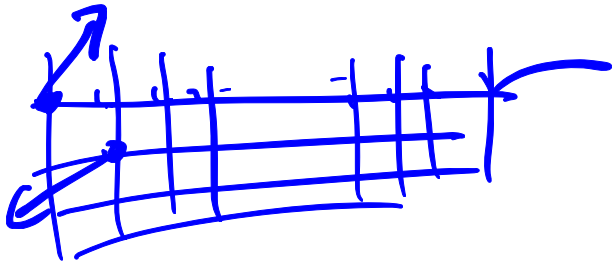
# Lowe's Keypoint Descriptor (shown with 2 X 2 descriptors over 8 X 8)

gradient magnitude and orientation at each point weighted by a Gaussian

orientation histograms: sum of gradient magnitude at each direction

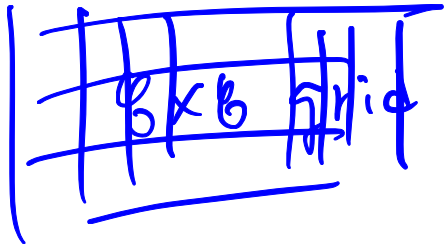



In experiments, 4x4 arrays of 8 bin histogram is used, a total of 128 features for one keypoint



$\left(\frac{\partial}{\partial x}G, \frac{\partial}{\partial y}G\right)$ conv. , $[-1\ 0\ 1]$ conv. kernel $\begin{bmatrix}-1\\0\\1\end{bmatrix}$ gradient.

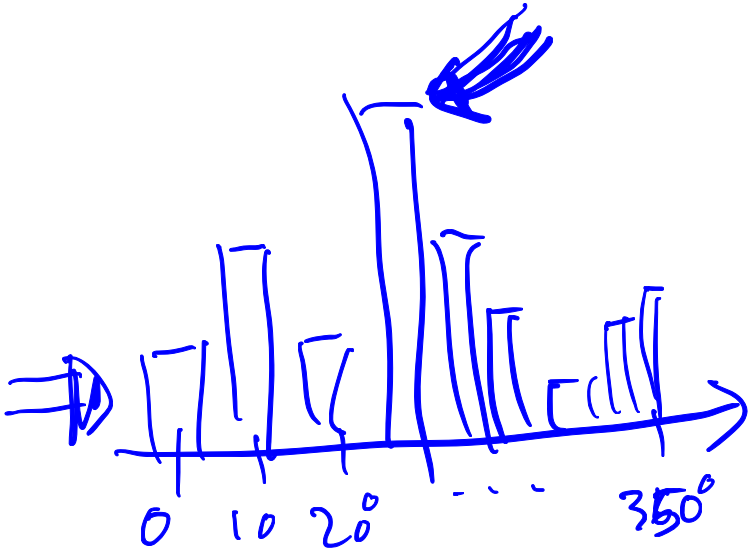
● key point 를 계산한다.




8×8 grid = 64개의 location.

⇒ 각각의 위치에서 gradient vector를 구함.

$(L_x, L_y)$ 라고하면




⇒ $\theta = \text{atan2}(L_y, L_x)$.

해당 위치에서 각각의 각도가 나옴.

⇒ histogram 을 그려서

⇒ 가장 많이 나타난 각도를 선정!

# Appendix

# Reference and further reading

Textbook

- "Chap 4: Feature detection and matching" of R. Szeliski, Computer Vision: Algorithms and Applications
- "Chap 5" of Forsyth and Ponce, Computer Vision: A Modern Approach

Publication


- D. G. Lowe, "Object recognition from local scale-invariant features," in Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV), pp. 1150–1157, 1999. [doi]
- D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, no. 60, vol. 2, pp. 91–110, 2004.
- T. Lindeberg, "Scale invariant feature transform," 2012. [doi]

# Reference and further reading

Link 강추, youtube

- M. Shah, UCF CRCV Video Lectures | Lecture 05 - Scale-invariant Feature Transform (SIFT) (youtube)
- CSE576 Course Slide, University of Washington (2011)
- R. Collins, CSE486 | Lecture 11: LoG and DoG Filters, Penn State University

Laplacian, difference of Gaussian.

- https://www.vlfeat.org/overview/sift.html

opensource code, SIFT 👍?!

# HW: 도전과제

저한테메일→1차적 판단→수강생 공유→다음 수업시간 설명
→검증,의견제기→통과하면 이득.

Due: 10월 9일(금요일) 오후 1시 30분까지

- 숙제는 필수가 아니며, 제출은 이메일 s.park@dgu.edu로 합니다.
- 제출한 코드는 모든 수강생에게 공유되며, 보강시간에 별도로 본인이 직접 구현한 원리를 설명합니다. (수업)
- 제대로 구현이 되어 있고 교수와 수강생이 이를 모두 납득하는 경우, 묻지도 따지지도 않고 모든 숙제 만점 드립니다.

- python으로 SIFT를 직접 구현하여 cv2 library의 sift.detect와 동일한 key point를 추출하는 것을 보여주세요.

안내면 불이익X, 냈는데 틀려도 →이익X, 프로그램한 경험이 얻는 것이 많음.