UNIVERSITÀ DEGLI STUDI DI PADOVA

The SIFT feature

Stefano Ghidoni

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

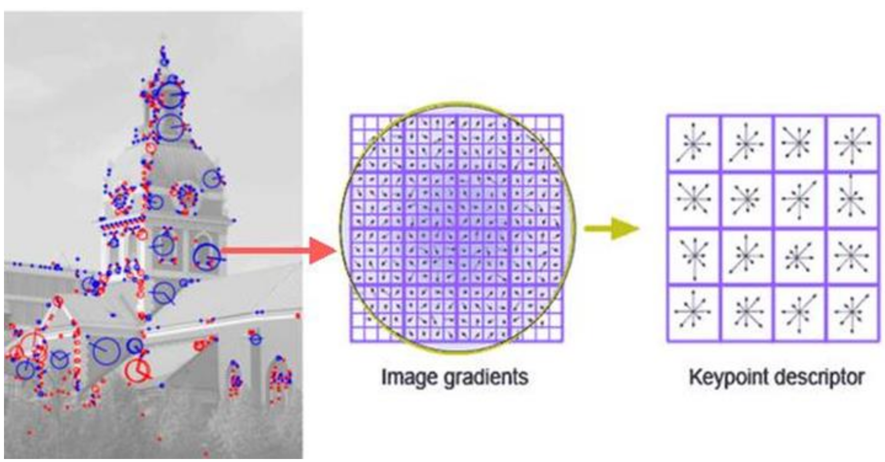
INTELLIGENT AUTONOMOUS SYSTEMS LAB

Agenda

- SIFT feature in detail
  - Keypoint detection
  - Descriptor calculation
- SIFT performance measurement

Scale Invariant Feature Transform

- Very reliable keypoint detector and descriptor
- Widely used



Although it's not always the case that a paper cited more contributes more to the field, a highly cited paper usually indicates that something interesting have been discovered. The following are the papers to my knowledge being cited the most in Computer Vision. (updated on 11/24/2013) If you want your "friend's" paper listed here, just comment below.

Cited by 21528 + 6830 (Object recognition from local scale-invariant features)

Distinctive image features from scale-invariant keypoints
DG Lowe - International journal of computer vision, 2004

Cited by 17671

A theory for multiresolution signal decomposition: The wavelet representation
SG Mallat – Pattern Analysis and Machine Intelligence, IEEE ..., 1989

Cited by 17611

A computational approach to edge detection
J Canny – Pattern Analysis and Machine Intelligence, IEEE ..., 1986

Cited by 15422

Snakes: Active contour models
M Kass, A Witkin, Demetri Terzopoulos - International journal of computer ..., 1988

Cited by 15188

Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images
Geman and Geman - Pattern Analysis and Machine ..., 1984

Cited by 11630+ 4138 (Face Recognition using Eigenfaces)

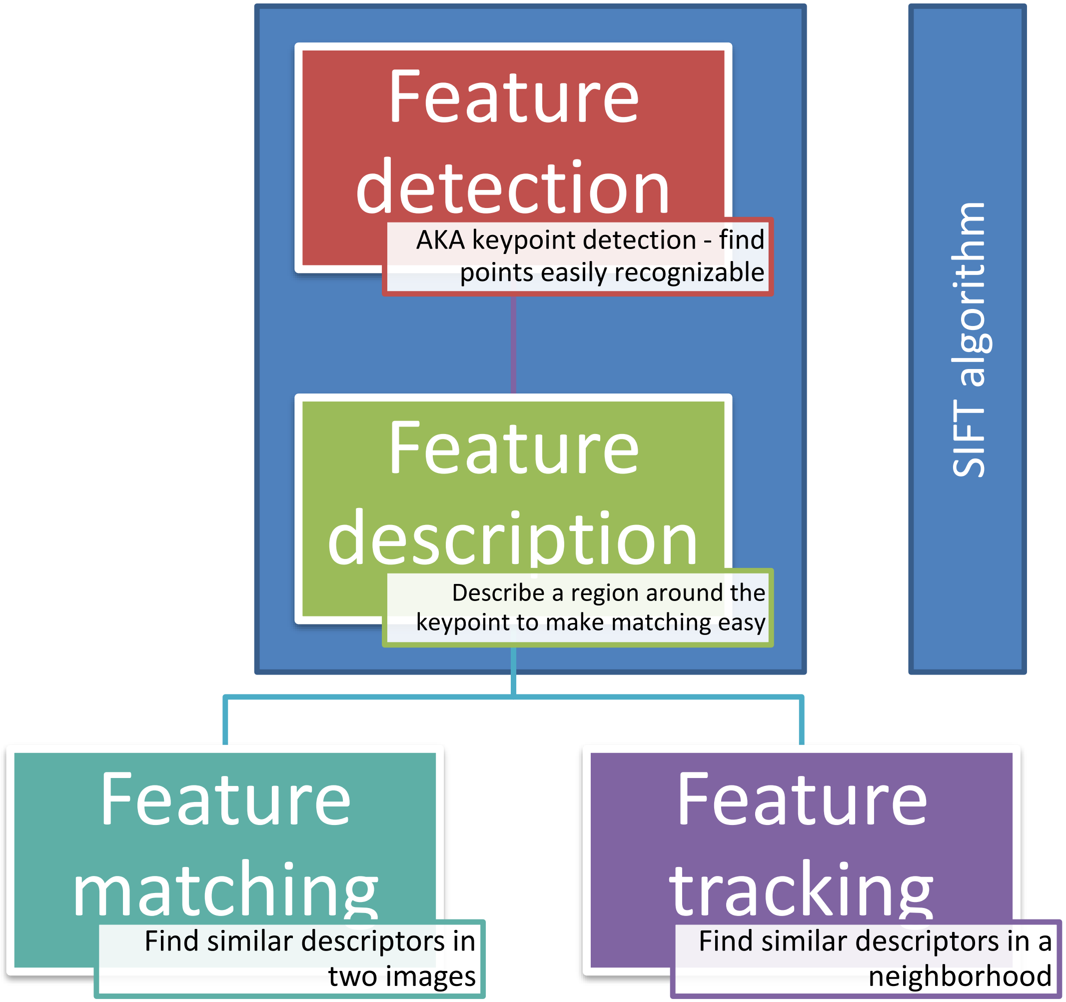
Eigenfaces for Recognition
Turk and Pentland, Journal of cognitive neuroscience Vol. 3, No. 1, Pages 71-86, 1991 (9358 citations)

Cited by 8788

Determining optical flow
B.K.P. Horn and B.G. Schunck, Artificial Intelligence, vol 17, pp 185-203, 1981

D. G. Lowe, "Distinctive image features from scale-invariant keypoints", International journal of computer vision, 60, 91-110 (2004)

Feature pipeline



Scale Invariant Feature Transform

- Why such a strong focus on scale?

Objects and scale

- Different scales
- What are the main differences?

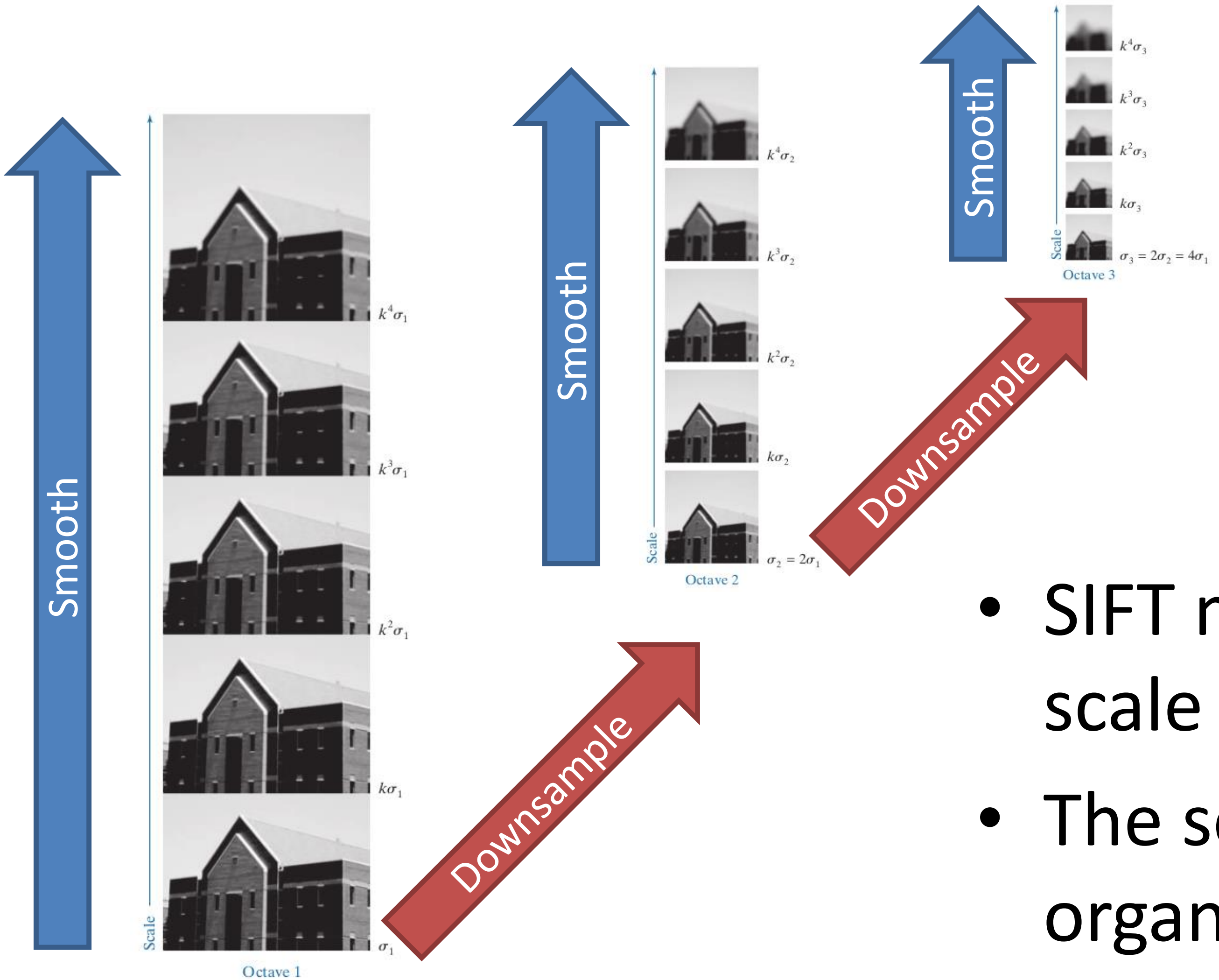
Strong points

- SIFT features:
  - Local – robust to occlusions
  - Distinctive – distinguish objects in large databases
  - Dense – many features can be found even on small objects
  - Efficiency – (rather) fast computation (what's the meaning of fast?)

SIFT algorithm

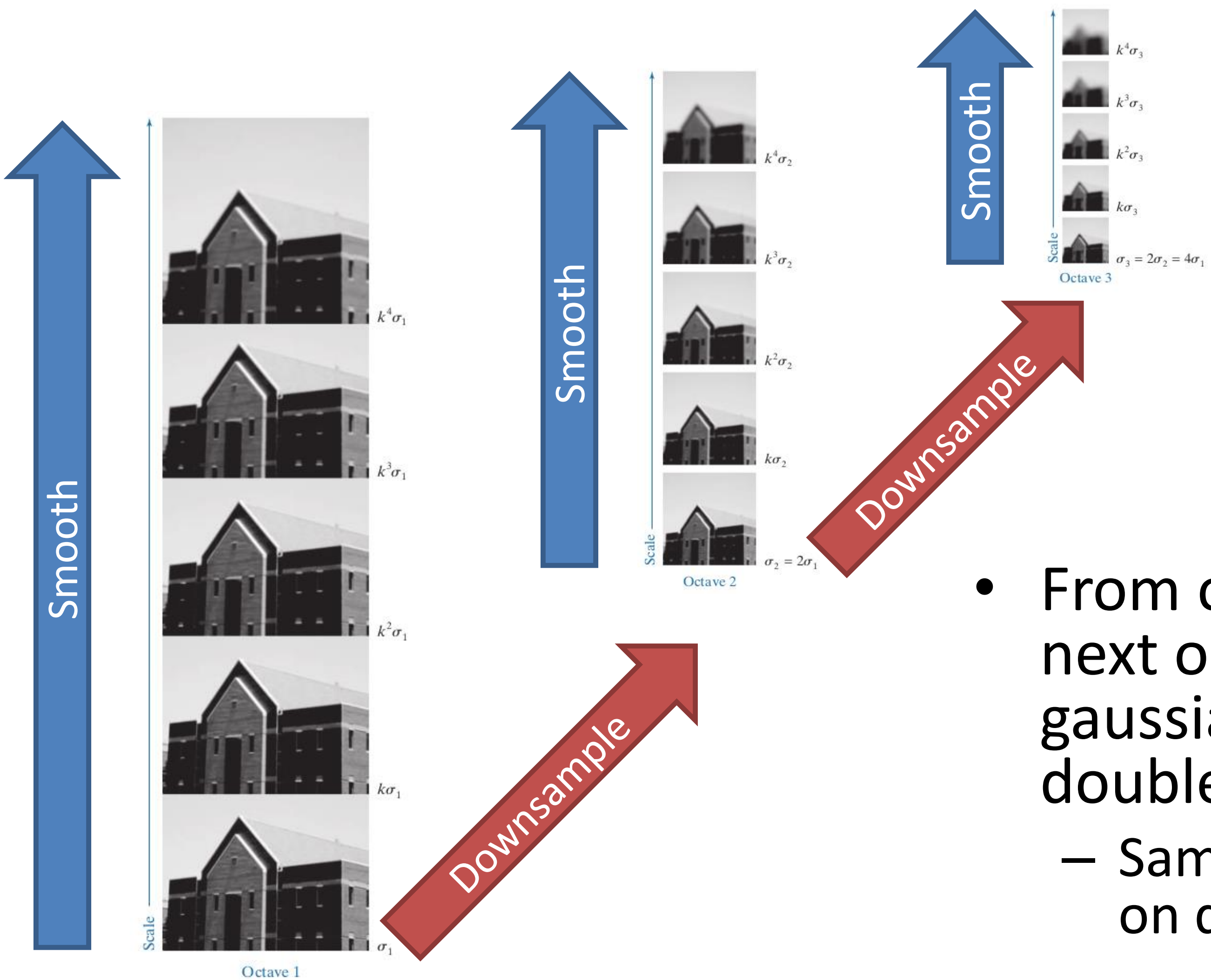
- Scale-space extrema detection
- Keypoint localization
- Orientation measurement
- Descriptor calculation

SIFT scale space



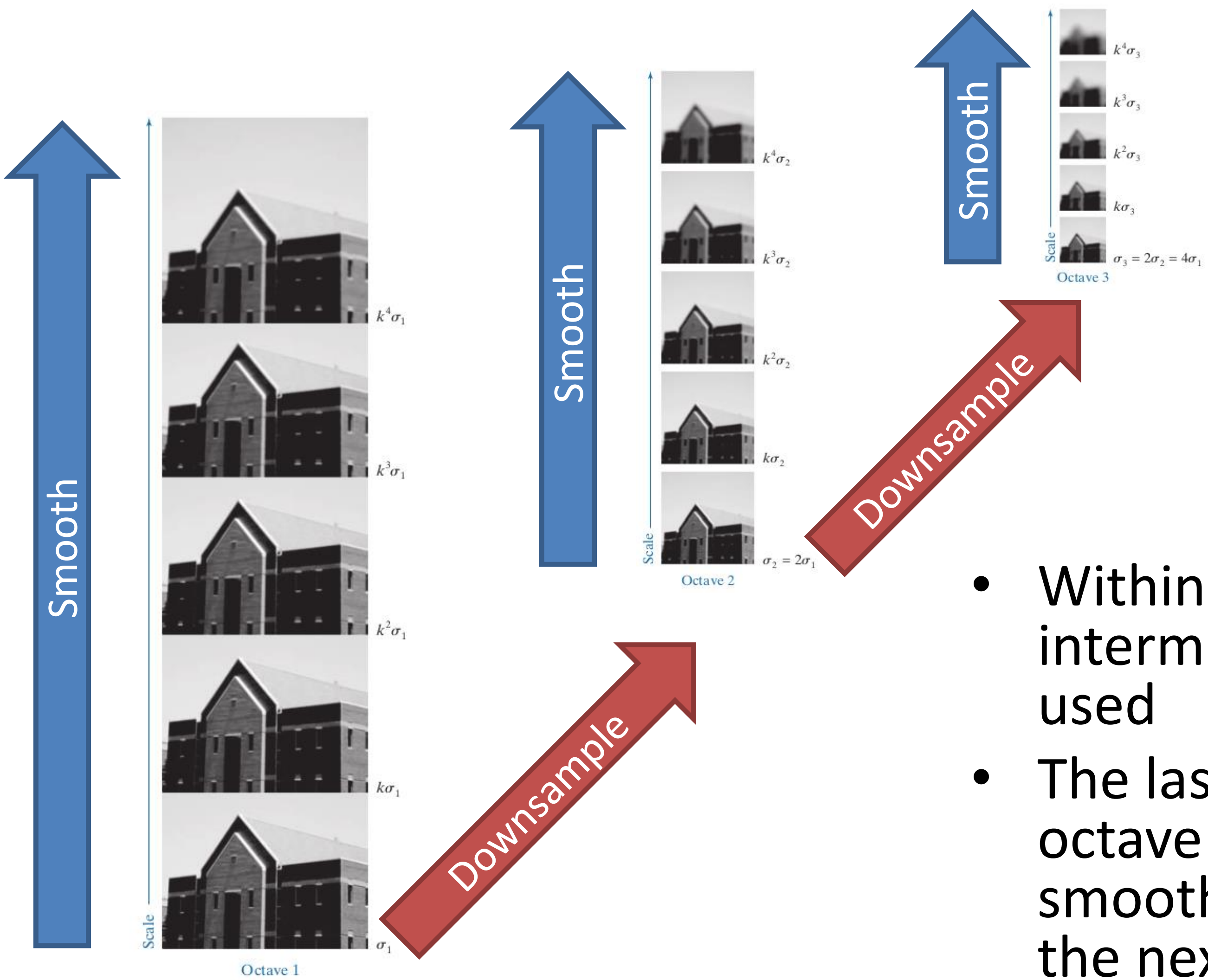
- SIFT makes use of a scale space
- The scale space is organized in octaves

SIFT scale space



- From one octave to the next one the σ of the gaussian smoothing is doubled
  - Same smoothing filter on downsampled image

SIFT scale space



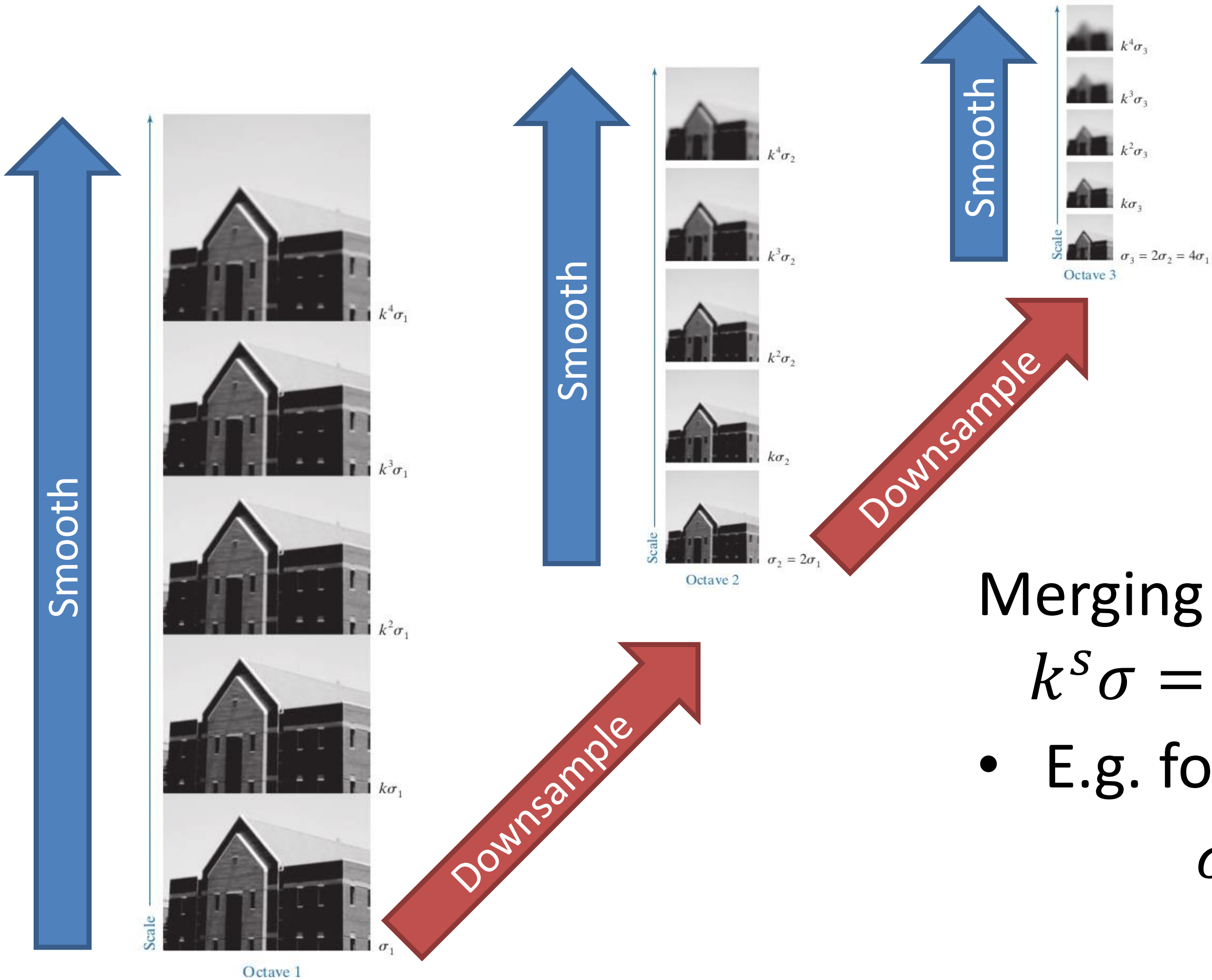
- Within one octave intermediate scales are used
- The last image of one octave has the same smoothing as the first in the next octave

SIFT scale space

Within one octave:

- s intervals
  - s+1 images
- Standard deviations
  - $\sigma, k\sigma, \mathrm{k}^2\sigma, \ldots, k^s\sigma$

SIFT scale space



Merging of two octaves:

$$k^s\sigma = 2\sigma \;\rightarrow k = 2^{1/s}$$

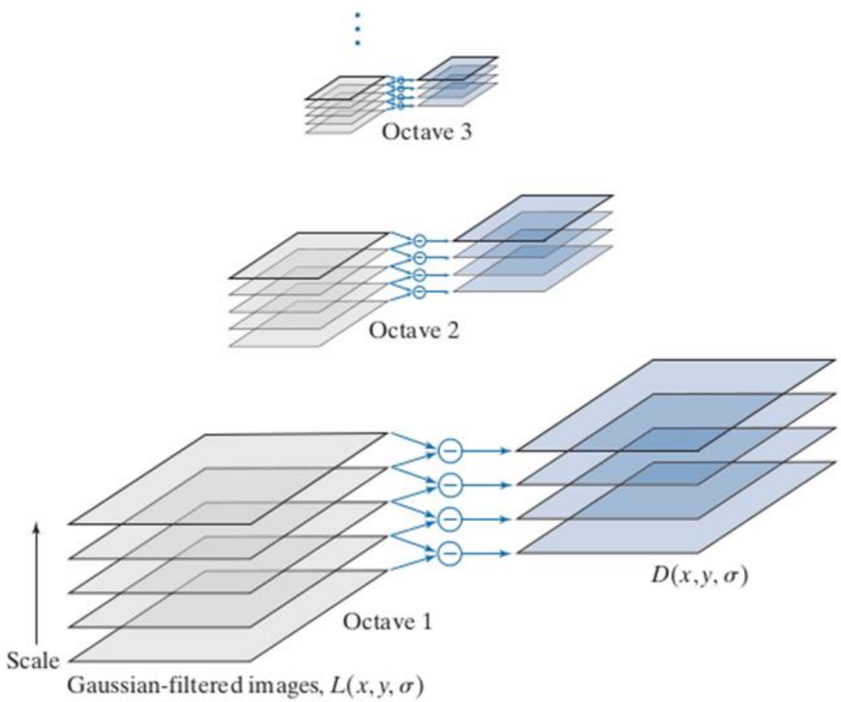
- E.g. for s=2 (3 images)

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

Layer subtraction

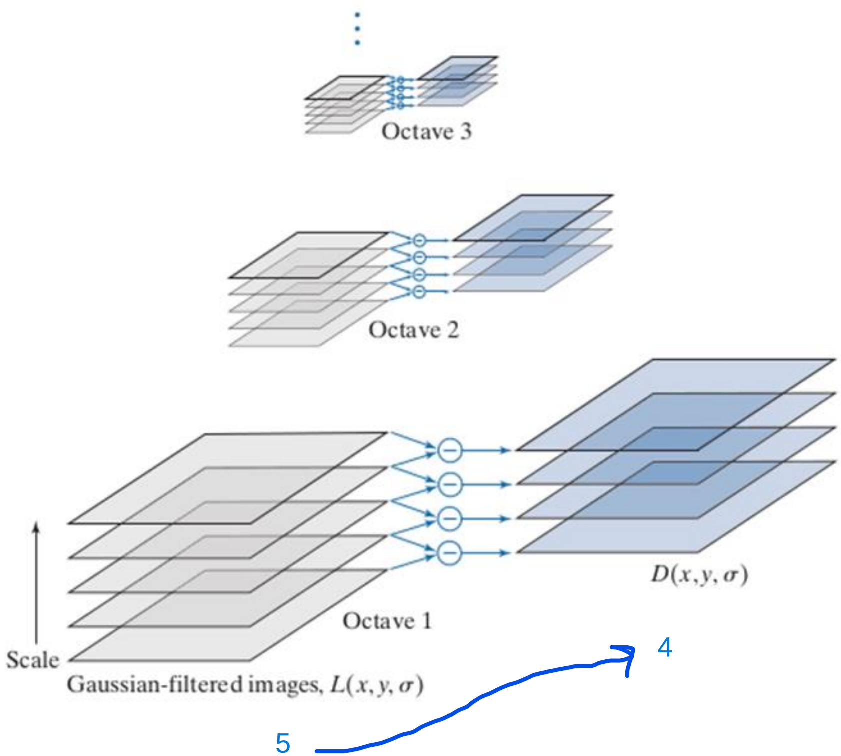
- Layers in the scale space are combined by subtracting consecutive layers

L – gaussian filtered image
D – difference of gaussian filtered images



Layer subtraction

- This requires two additional images in the scale space to process the first and last image

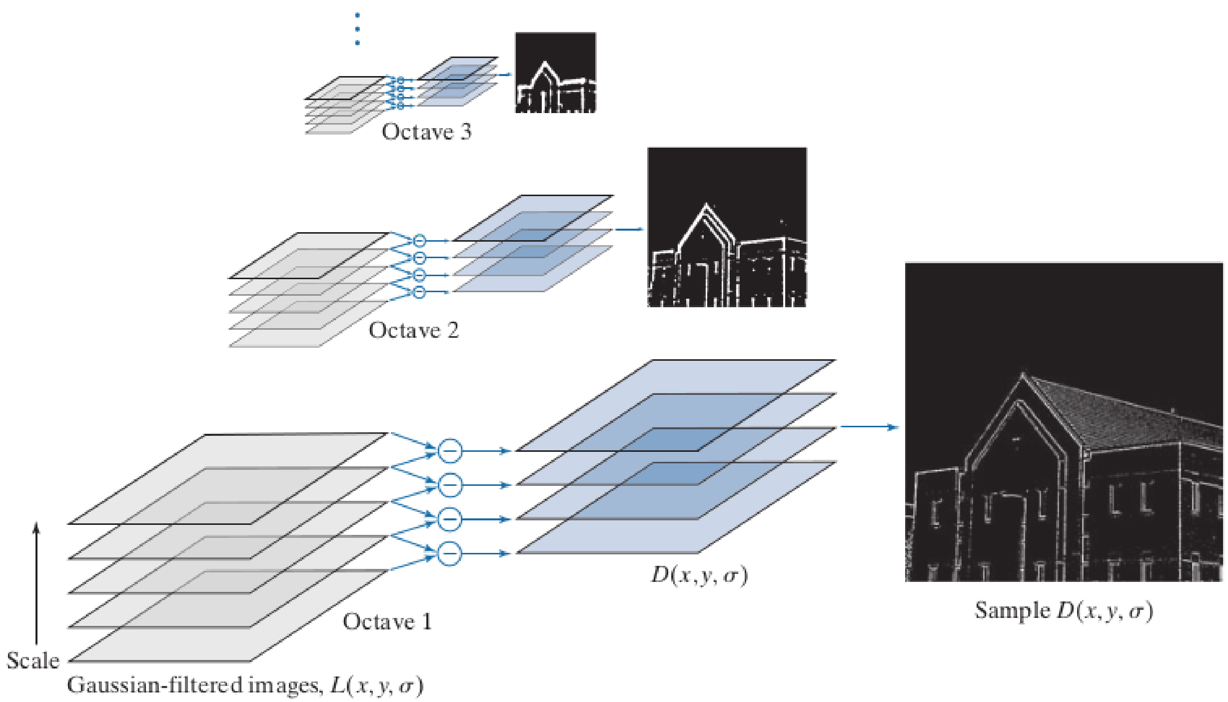


Octaves for $s = 2$

Octave	Scale 1	2	3	4	5
1	0.707	1.000	1.414	2.000	2.828
2	1.414	2.000	2.828	4.000	5.657
3	2.828	4.000	5.657	8.000	11.314

Layer subtraction

- Output images



DoG

- Subtracting consecutive layers means: evaluate the difference between two smoothed images
  - Same smoothing, different smoothing intensity
- Such filtering is called Difference of Gaussians (DoG) and is represented by the function

$$D(x, y, \sigma)$$

DoG

- Subtracting consecutive layers means: evaluate the difference between two smoothed images
  - Same smoothing, different smoothing intensity
- What is the meaning of this filter?
  - Let's go one step back: derivative filters

LoG

- Recall – we observed many times the pattern:
  - Smoothing
  - Derivative filter (edge detection filter)
- They can be combined into one filter
  - Derivative of the smoothing filter
- This is the concept exploited in the Laplacian of Gaussian (LoG) filter

LoG

- Laplacian of a Gaussian (LoG) is obtained using the gaussian filter:

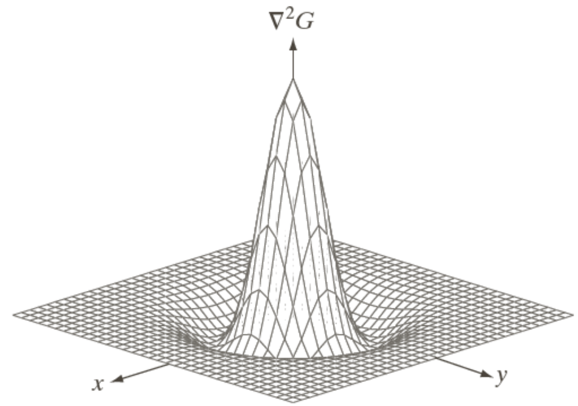
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

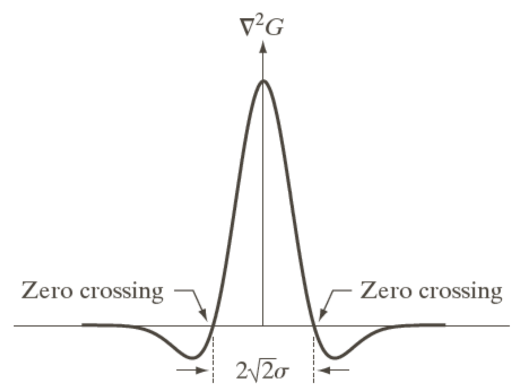
And considering its laplacian $\nabla^2 G(x, y)$ to filter the image:

- Filtering with LoG filter corresponds to:
  - Filter the input using $G(x, y)$
  - Compute the Laplacian of the resulting image

LoG – features

- Smoothing
- Noise removal at scales smaller than σ
- Zero-crossing
- Isotropic
- Typ filter size: $n \times n$ s.t.
  $n > 6\sigma$





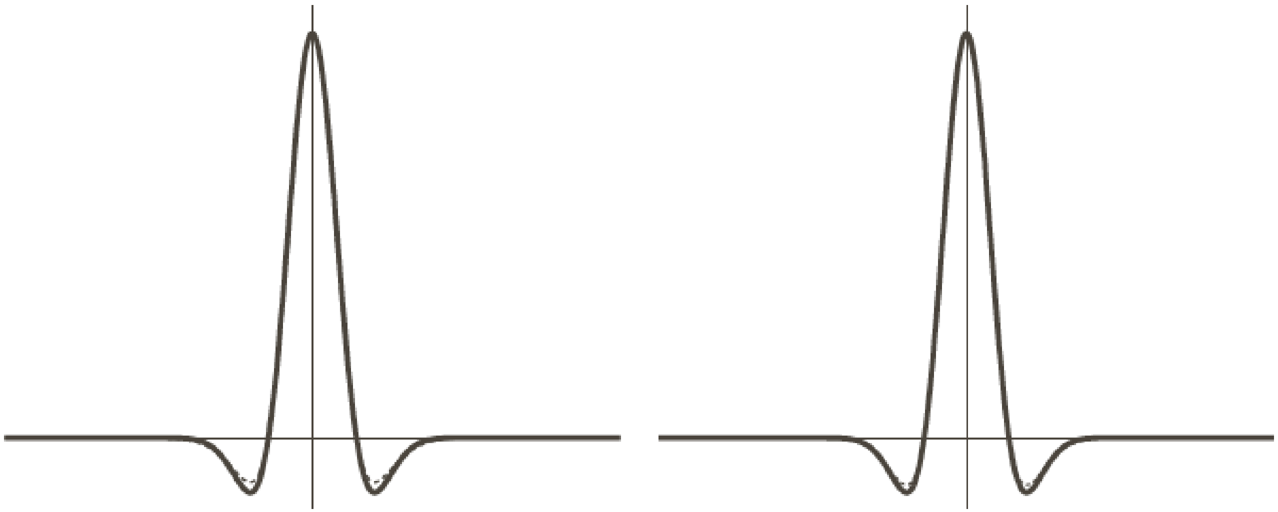
-LoG

0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

DoG vs LoG

- An approximation: DoG (Difference of Gaussians) – $\sigma_1 > \sigma_2$

$$D(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}$$

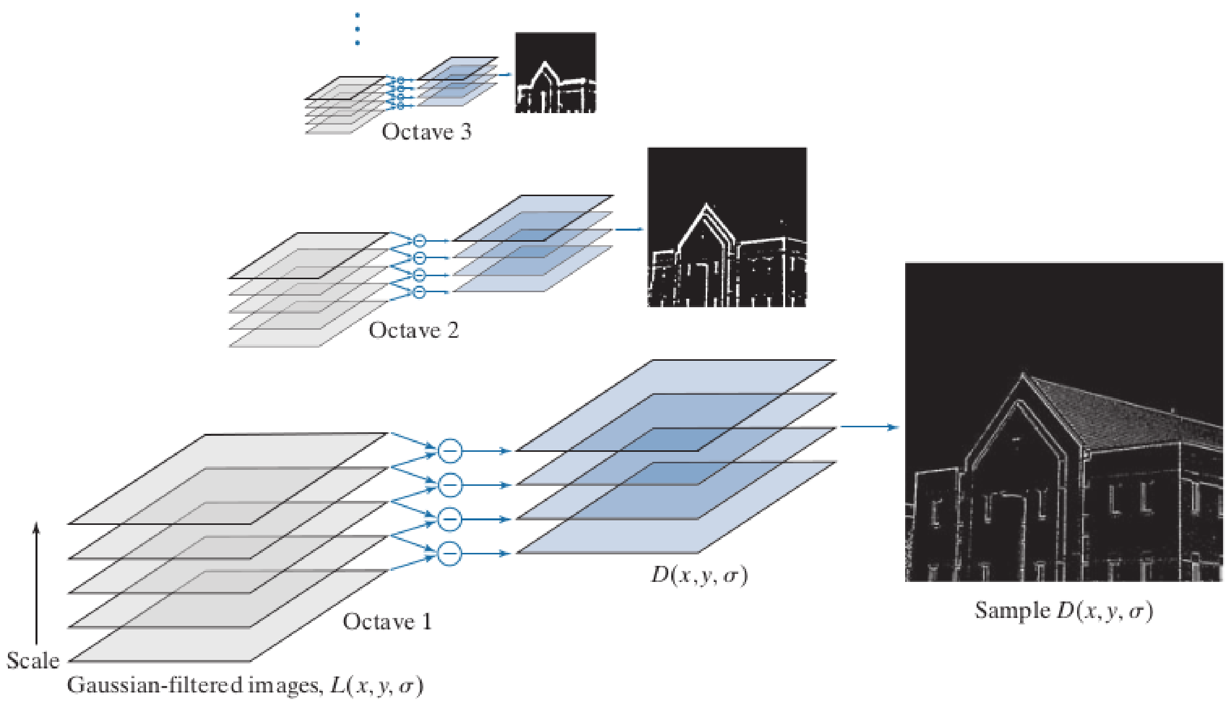


a b

FIGURE 10.23 (a) Negatives of the LoG (solid) and DoG (dotted) profiles using a standard deviation ratio of 1.75:1. (b) Profiles obtained using a ratio of 1.6:1.

SIFT DoG

- Back to SIFT: scale space + subtraction of consecutive layers is a DoG filter
  - Output images:



SIFT algorithm

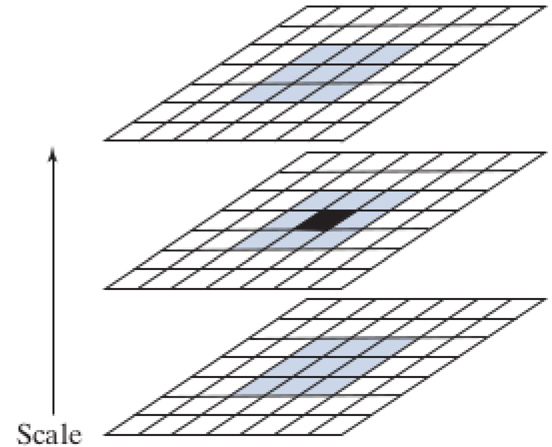
- Scale-space extrema detection
- Keypoint localization
- Orientation measurement
- Descriptor calculation

SIFT algorithm

- Scale-space extrema detection
- Keypoint localization
- Orientation measurement
- Descriptor calculation

Keypoint localization

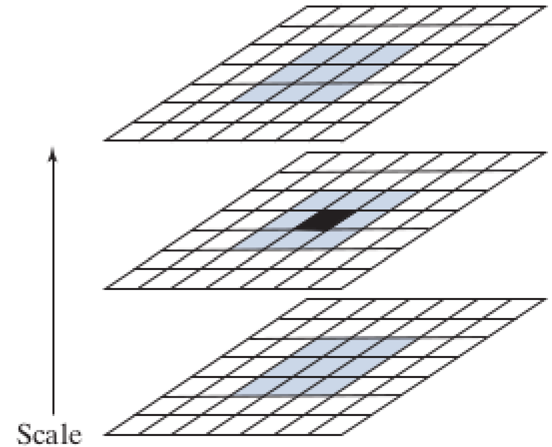
- Search for maxima and minima of the DoG
- Comparison with the 8-neighbors in the current, previous and next scale level
- This is done at **all scales**
  - This means: **scale independence**



Corresponding sections of three contiguous $D(x, y, \sigma)$ images

Keypoint localization

- Sub-pixel accuracy by means of interpolation
  - Taylor series expansion up to the quadratic term
- Interpolation along x, y, σ



Corresponding sections of three contiguous $D(x, y, \sigma)$ images

Hessian matrix

- Recall: SIFT is based on the laplacian – 2nd order derivative
- The Hessian matrix provides all the 2nd order derivatives

$$H = \begin{bmatrix} \partial^2 D/\partial x^2 & \partial^2 D/\partial x \partial y \\ \partial^2 D/\partial y \partial x & \partial^2 D/\partial y^2 \end{bmatrix} = \boxed{\begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}}$$

Similar to the auto-correlation matrix A

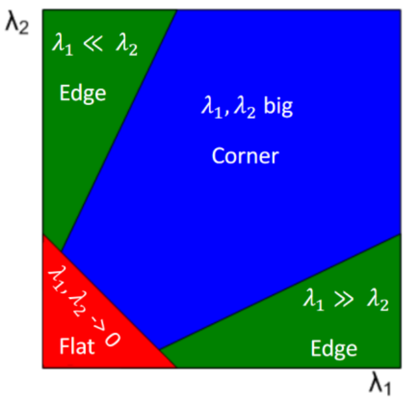
- Further processing is based on H

Keypoint refinement

- The Hessian matrix is used to:
  - Discard points with a low gradient

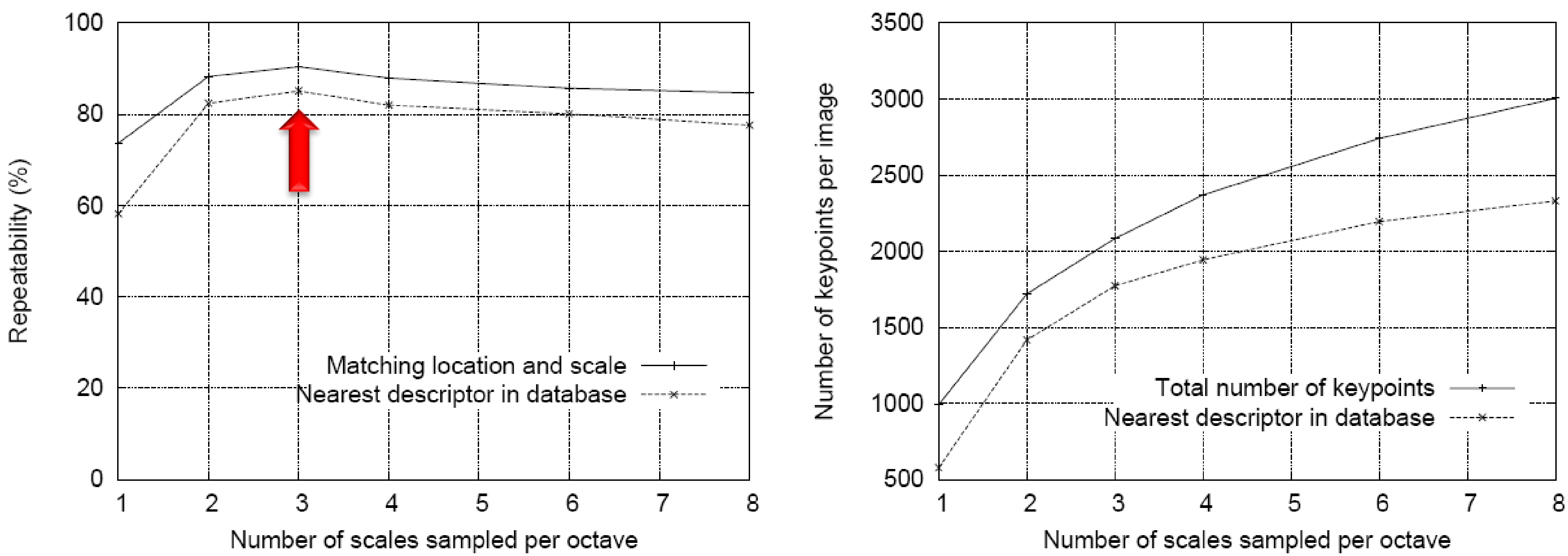
$$|D(\hat{x})| < 0.03$$

  - Discard edge points – require that both eigenvalues of H are large
    - Means: require that both curvatures are high
    - Similar to the discussion about Harris corners
    - Similar considerations about eigenvalues ratios



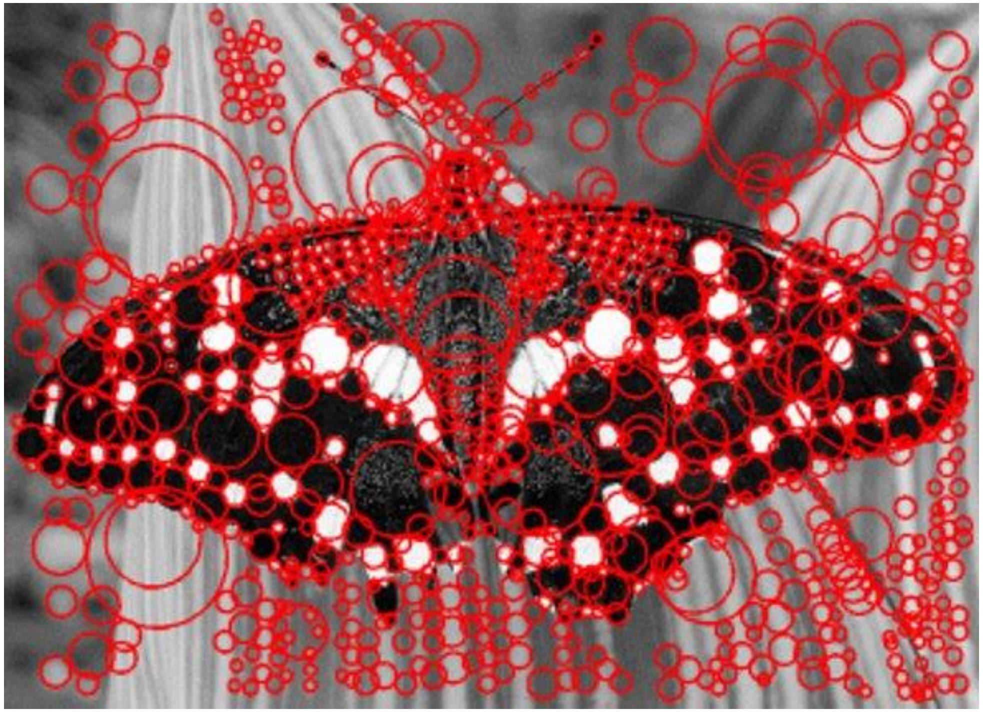
Scale sampling frequency

- More scale levels considered
  - More points
  - More unstable
  - Best value: s=3



Keypoint localization – example

- The output is a set of keypoints with associated scales



SIFT algorithm

- Scale-space extrema detection
- Keypoint localization
- Orientation measurement
- Descriptor calculation

SIFT algorithm

- Scale-space extrema detection
- Keypoint localization
- Orientation measurement
- Descriptor calculation

Keypoint orientation

- Each keypoint comes with its scale
- The gaussian smoothed image closest to that scale is selected (L)
  - This process is **independent from the scale**
- Compute gradient magnitude and orientation in the keypoint neighborhood using L

Keypoint orientation

- Gradient magnitude

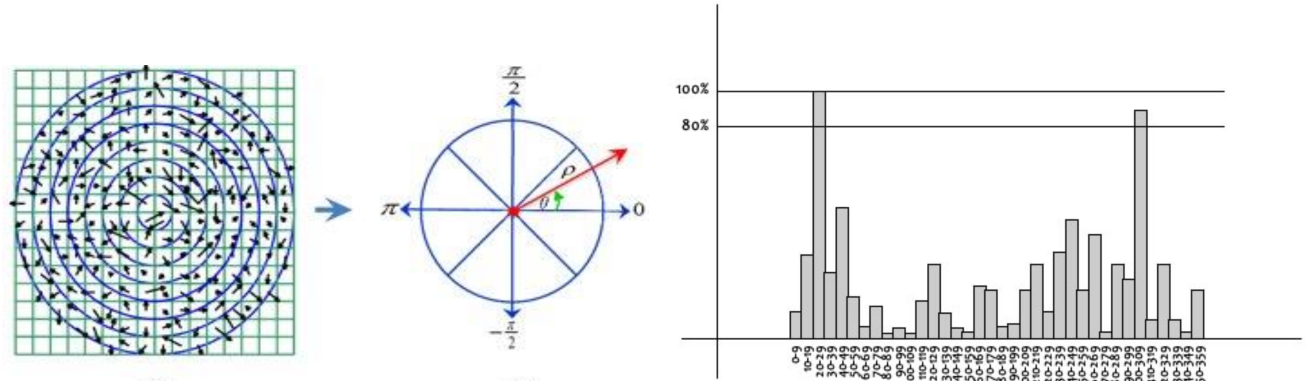
$$M(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2}$$

- Gradient orientation angle

$$\theta(x,y) = \tan^{-1}\left[\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right]$$

Keypoint orientation

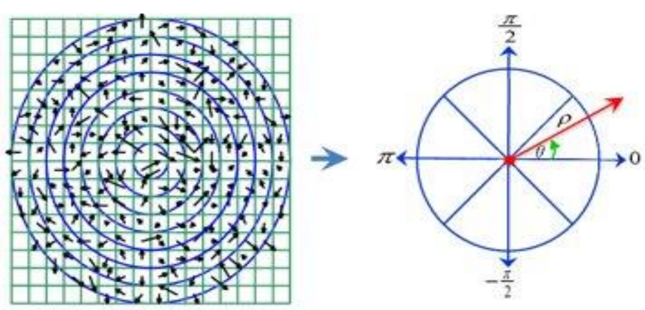
- To achieve **rotation invariance**, the dominant local direction shall be measured
- Build the histogram of gradient orientations
  - 36 bins of 10° each - region around the keypoint



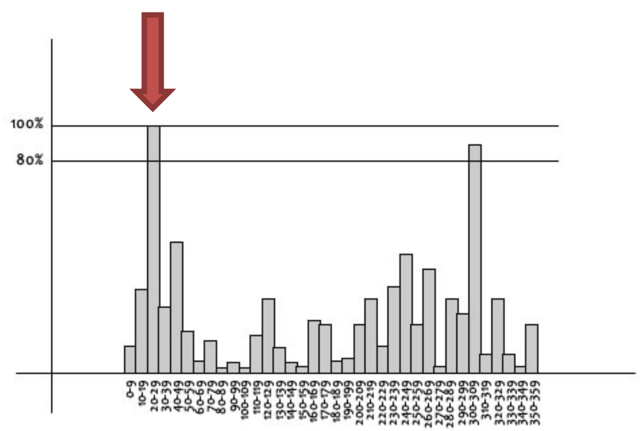
(a) (b)

Keypoint orientation

- To achieve **rotation invariance**, the dominant local direction shall be measured
- Build the histogram of gradient orientations
  - 36 bins of 10° each - region around the keypoint
- Look for the peak

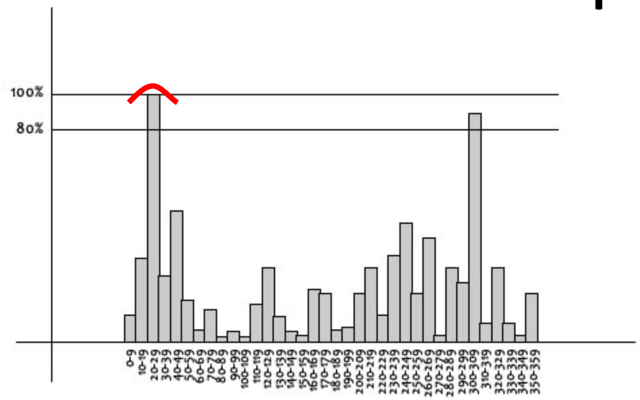


(a) (b)



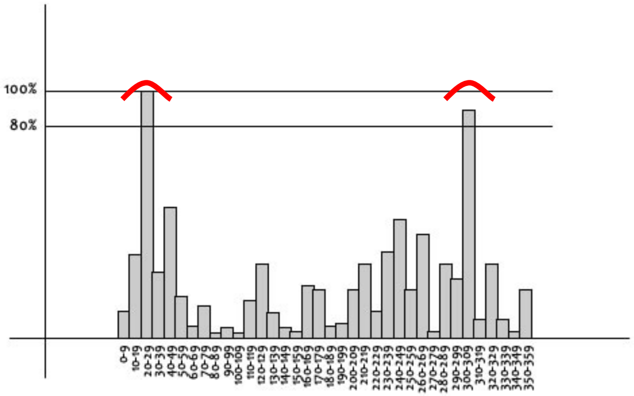
Keypoint orientation

- To achieve **rotation invariance**, the dominant local direction shall be measured
- Build the histogram of gradient orientations
  - 36 bins of 10° each - region around the keypoint
- Look for the peak
- Fit a parabola to the 3 bins close to the peak
  - Refine the peak location



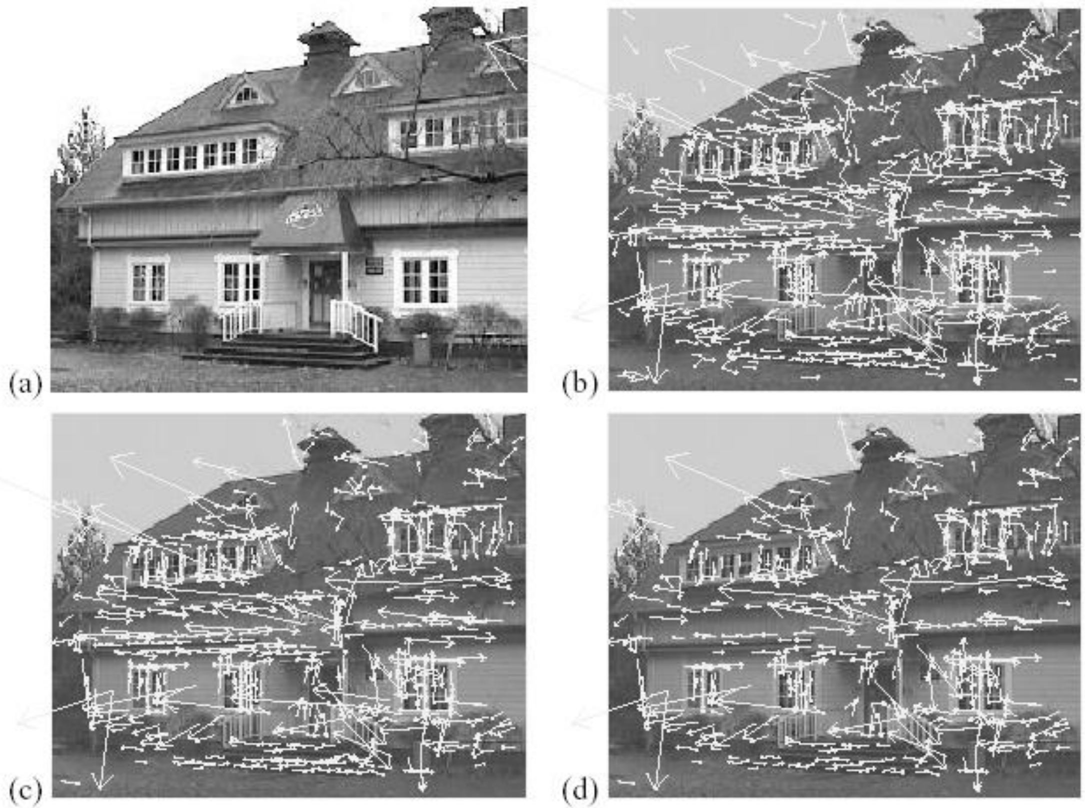
Keypoint orientation

- Secondary peaks are also considered
  - Peak value > 80% highest peak
- Secondary peaks duplicate the keypoint
  - A new keypoint is created, with the orientation set to the secondary peak



Keypoint localization – example

- Keypoint detection example
  - Arrows represent keypoint orientation



(a) 233x189 image
(b) 832 DOG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures

SIFT algorithm

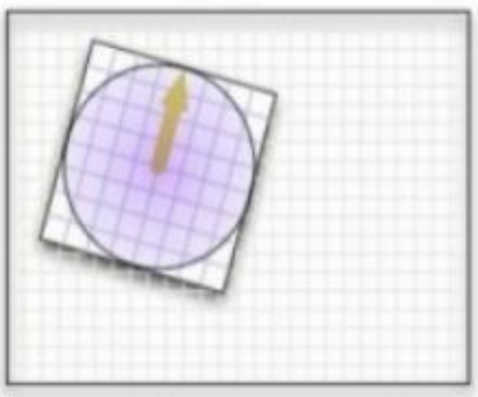
- Scale-space extrema detection
- Keypoint localization
- Orientation measurement
- Descriptor calculation

SIFT algorithm

- Scale-space extrema detection
- Keypoint localization
- Orientation measurement
- Descriptor calculation

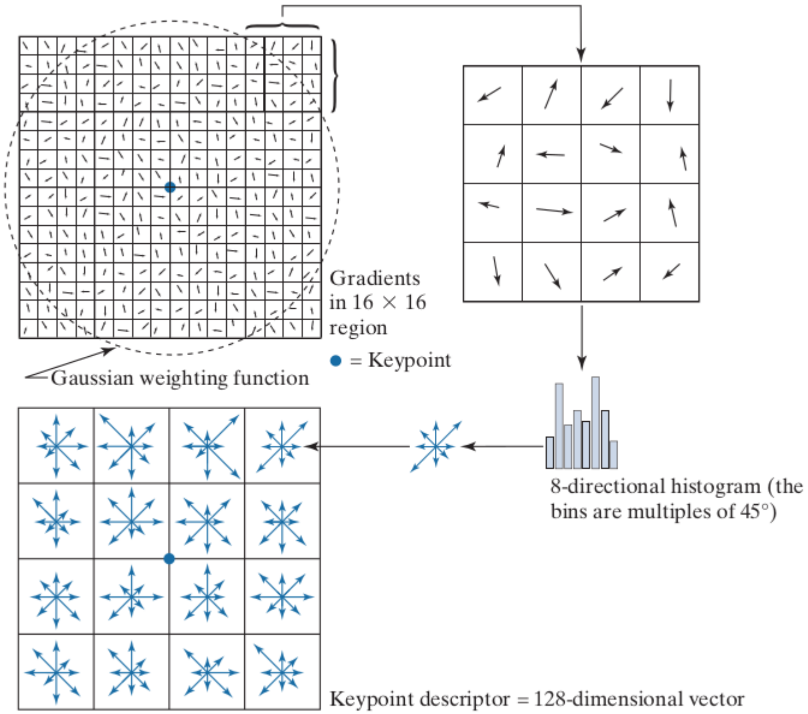
Descriptor

- SIFT evaluates a descriptor for each keypoint
  - In the image L corresponding to the keypoint scale
  - Considering coordinates that are rotated based on the measured keypoint orientation



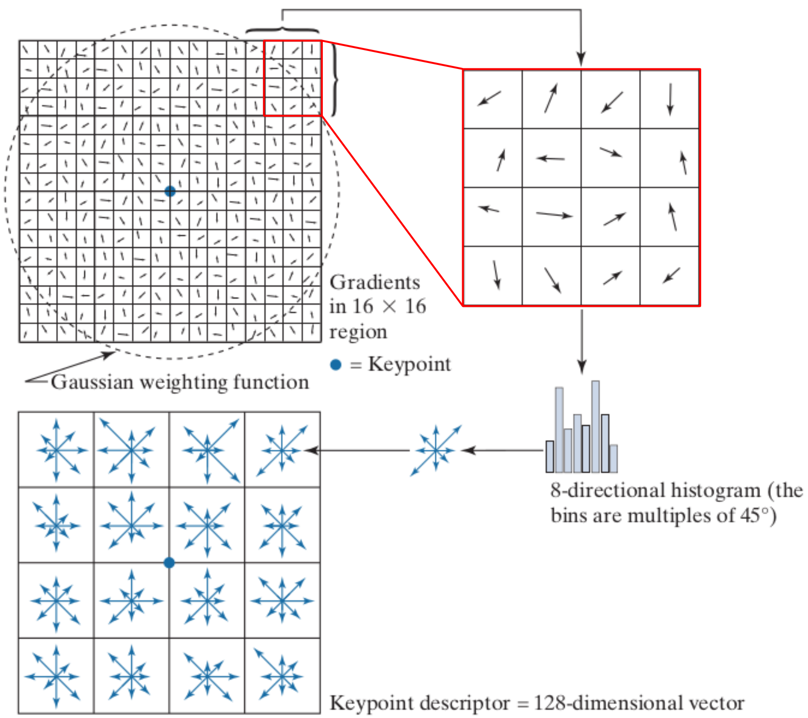
Descriptor

- Compute gradient magnitude and direction
  - neighborhood: 16×16
- Values weighted with a gaussian function centered on the keypoint



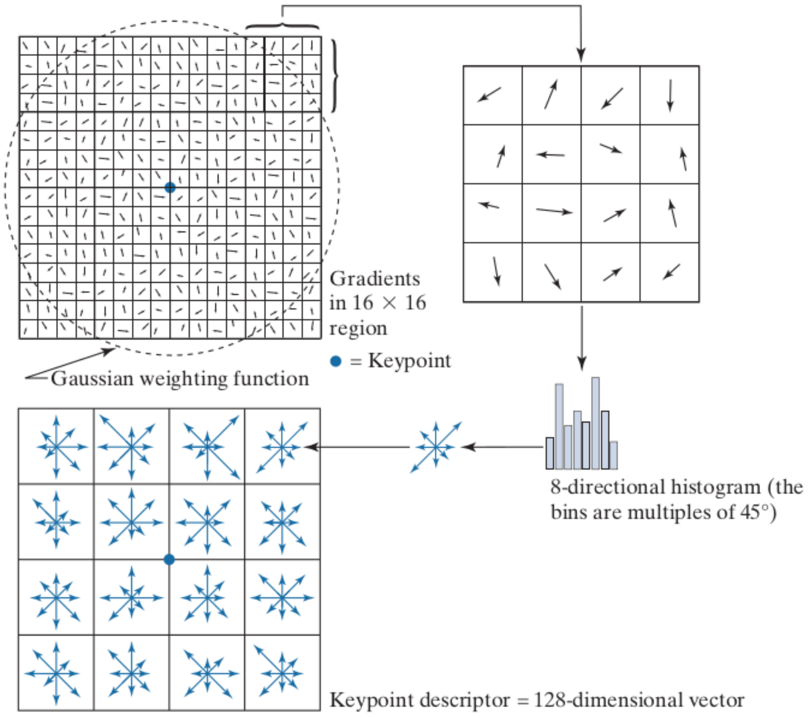
Descriptor

- 16x16 neighborhood divided into regions of 4×4 pixels
- Evaluate histogram of each region
  - 8 bins (45° each)
- Save the values



Descriptor size

- Descriptor size:
  - 16 regions
  - 8 histogram values per region
- 128 elements
  - 1 byte per element

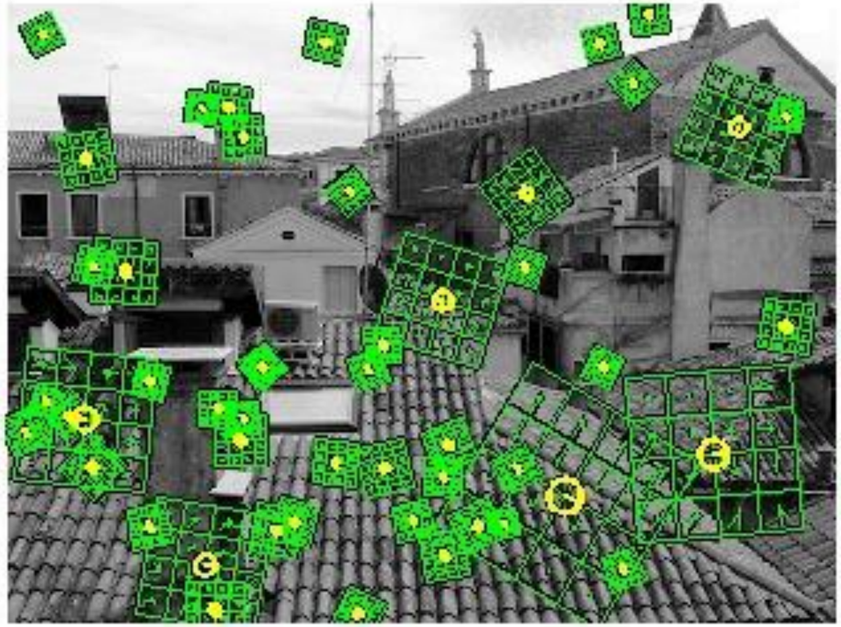


Descriptor refinements

- Additional details are considered by the SIFT algorithm
- Enhanced **invariance to illumination**: feature vector normalized to unit length
- Enhanced **robustness to large gradient magnitudes**: threshold to 0.2 on all the components
  - Nonlinear illumination effect – e.g., camera saturation
  - Further normalization to 1 after thresholding

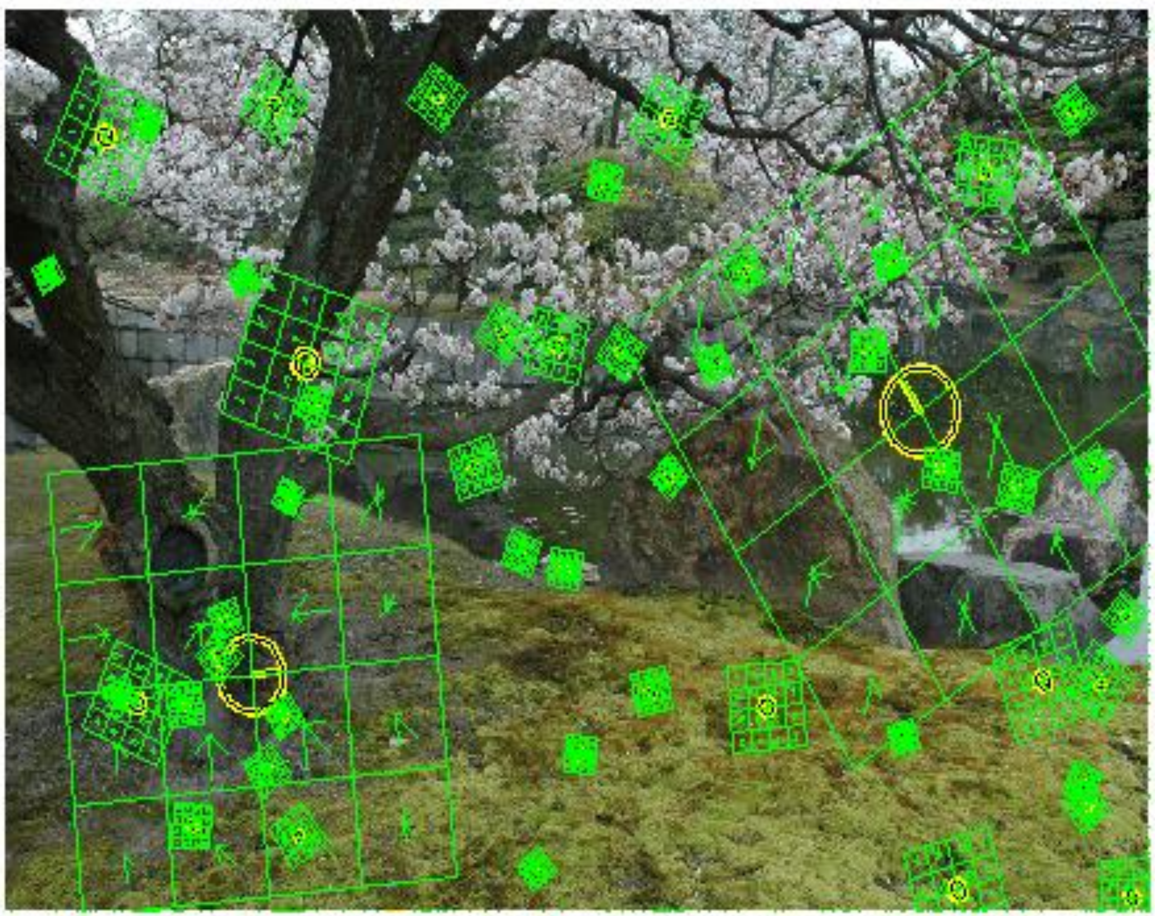
Example

- SIFT (VLFeat library)

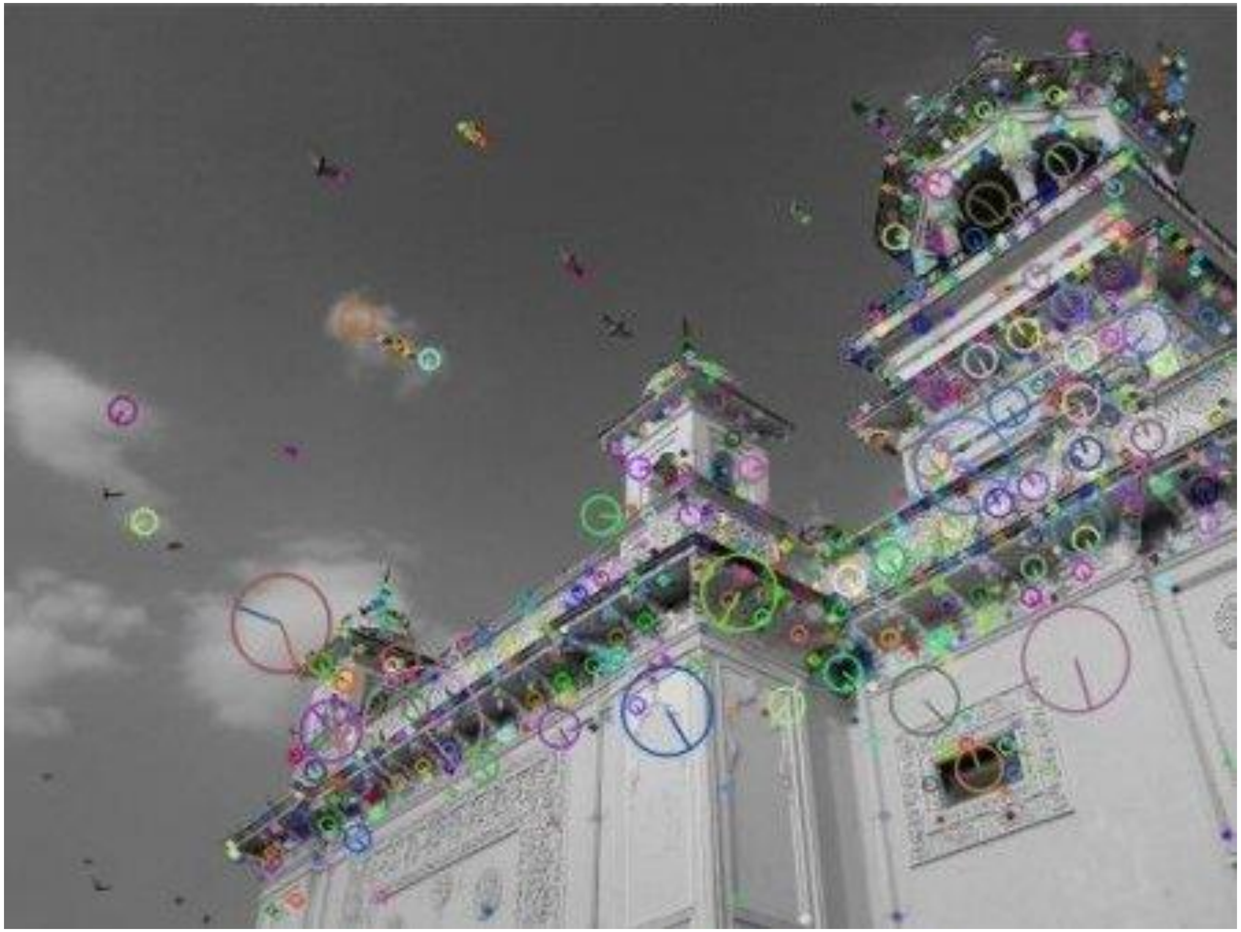


Example

- SIFT (VLFeat library)



- SIFT (OpenCV)

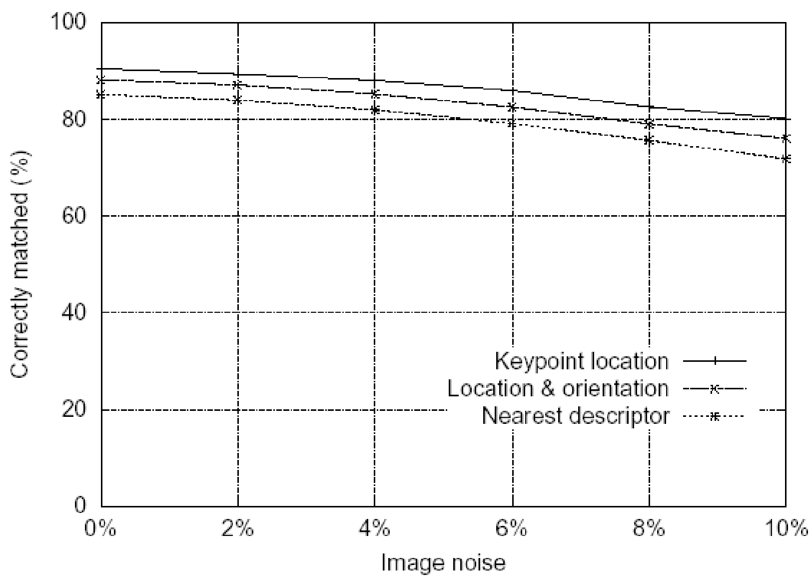


SIFT robustness

- The robustness of the SIFT features to noise factor was accurately measured

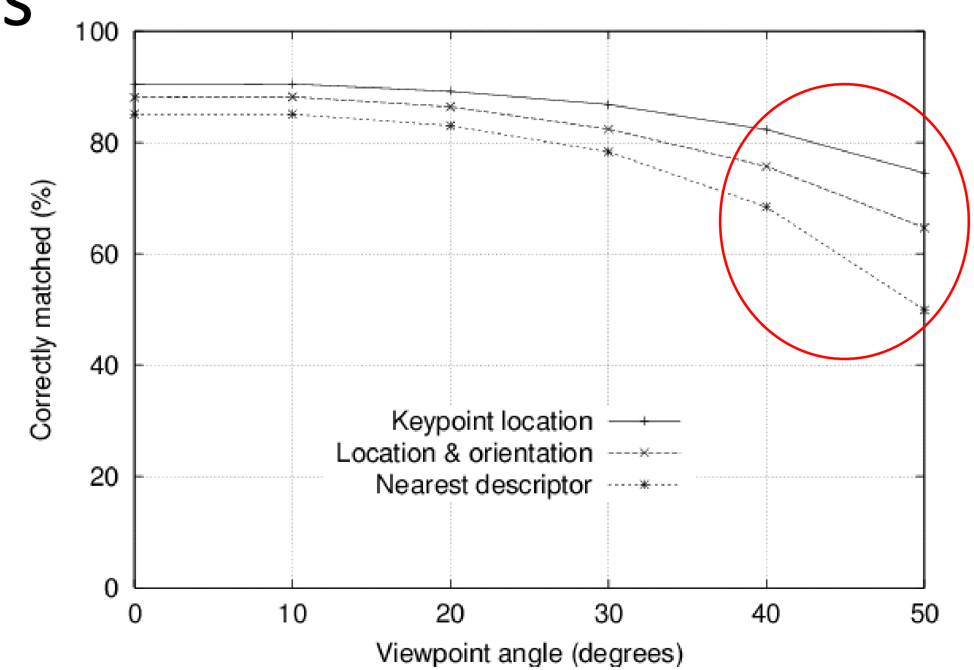
Robustness to noise

- Random rotation and scale change with different noise levels
  - Nearest neighbor on 30,000 feature database



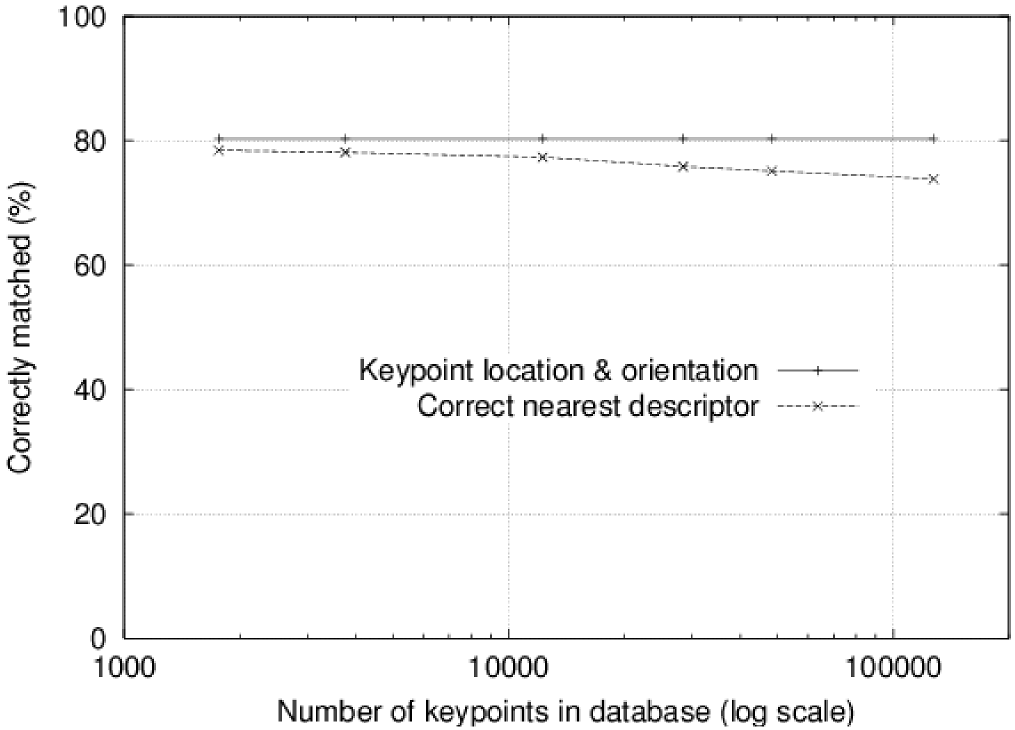
Robustness to affine transform

- Random scale change + rotation with 2% noise and affine distortion
  - Nearest neighbor on 30,000 feature database
- Strong for small angles



Feature distinctiveness

- 2% Noise and 30° viewpoint change



Discussion

- Why is SIFT invariant to scale?
- Why is SIFT invariant to illumination changes?
- Why is SIFT invariant to orientation?

UNIVERSITÀ DEGLI STUDI DI PADOVA

The SIFT feature

Stefano Ghidoni