# SIFT

Mohammad Nayeem Teli

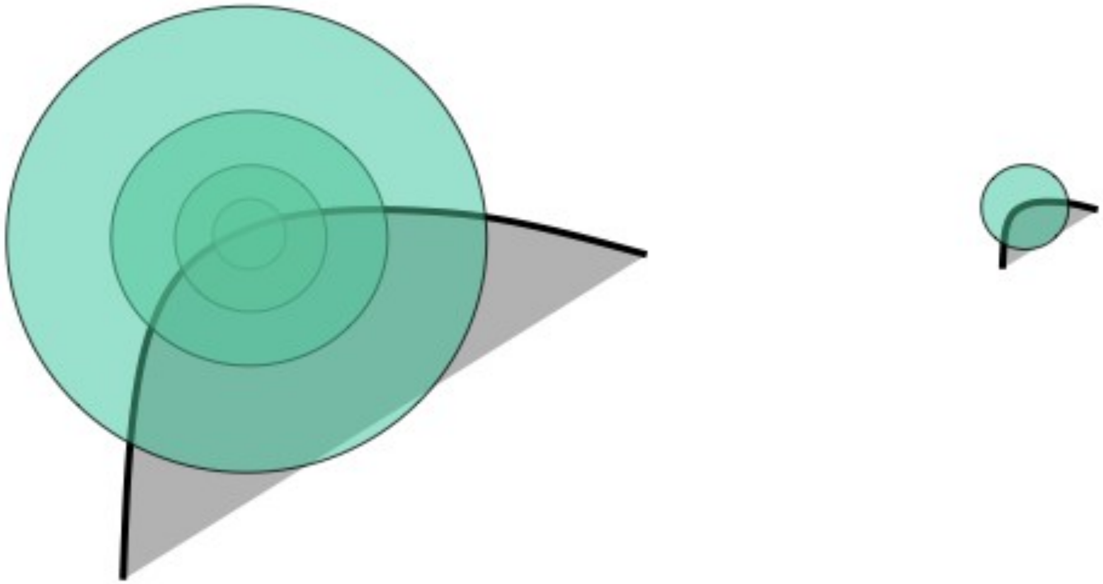
# Feature detectors

should be invariant or at least robust

to affine changes
translation
rotation
scale change

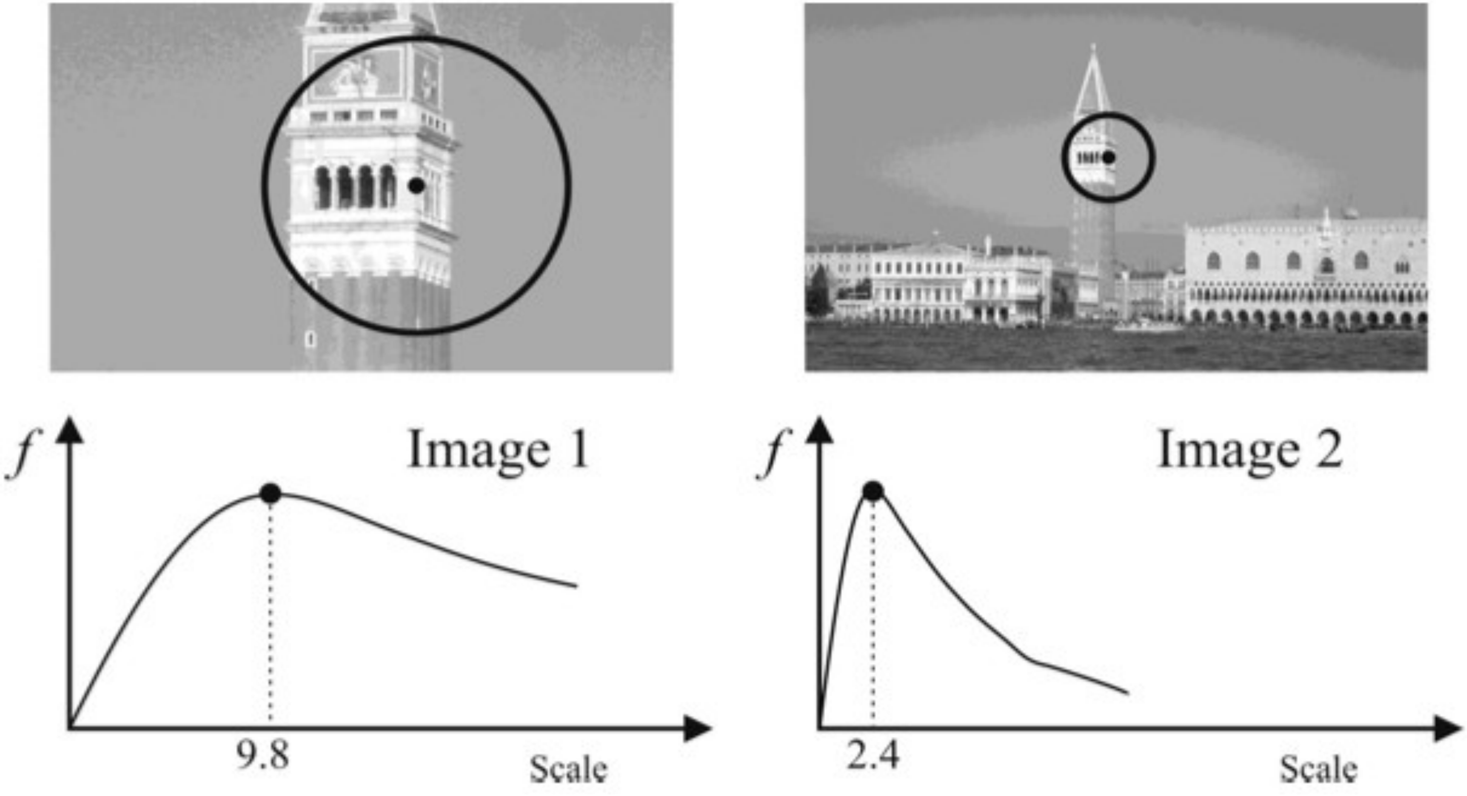
# Scale Invariant Detection

- Consider regions of different size
- Select regions to subtend the same content



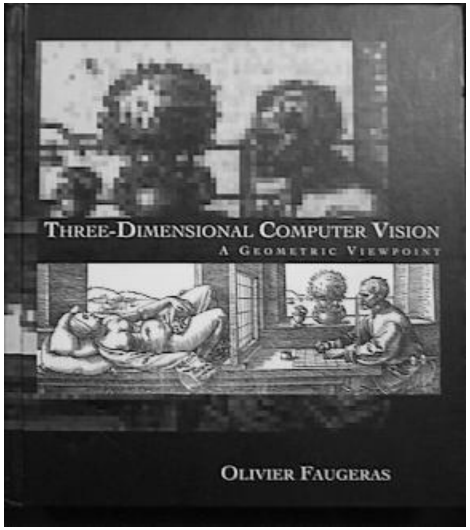
# Scale Invariant detection

- Sharp local intensity changes are good functions for identifying relative scale of the region
- Response of Laplacian of Gaussians (LoG) at a point

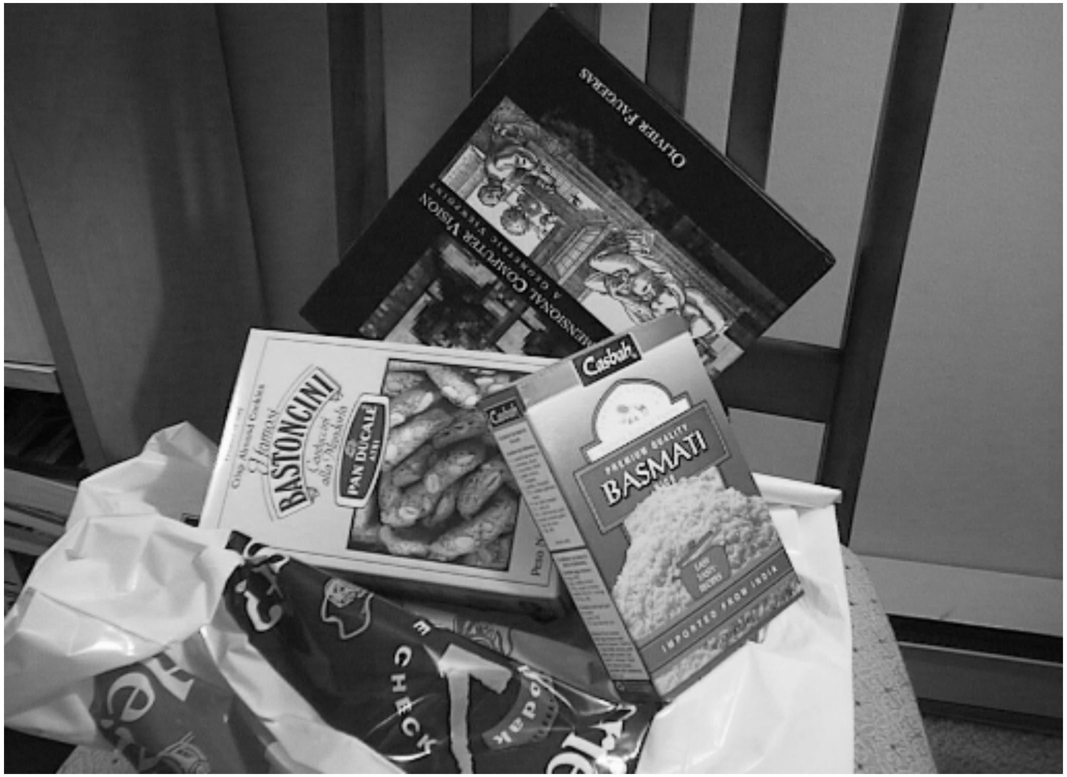


# Improved Invariance Handling

Want to find



… in here

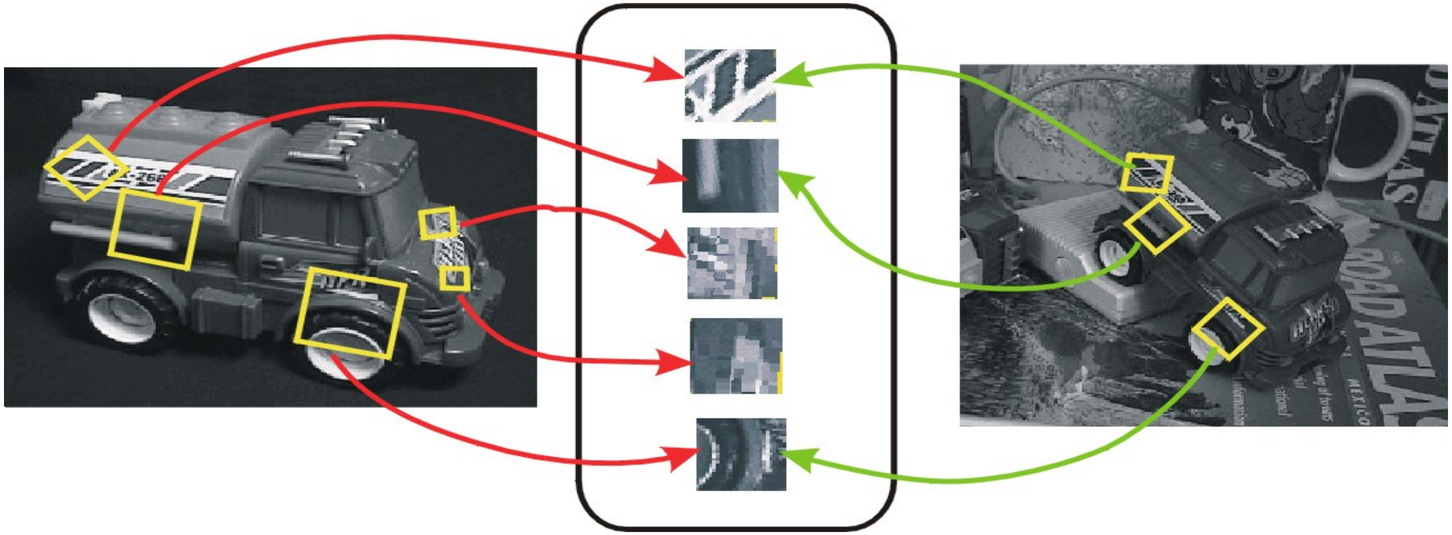


# SIFT

- Scale-Invariant Feature Transform
- David Lowe
- Scale/rotation invariant
- A very well known feature descriptor
- Applications
  - Object recognition, Robot localization

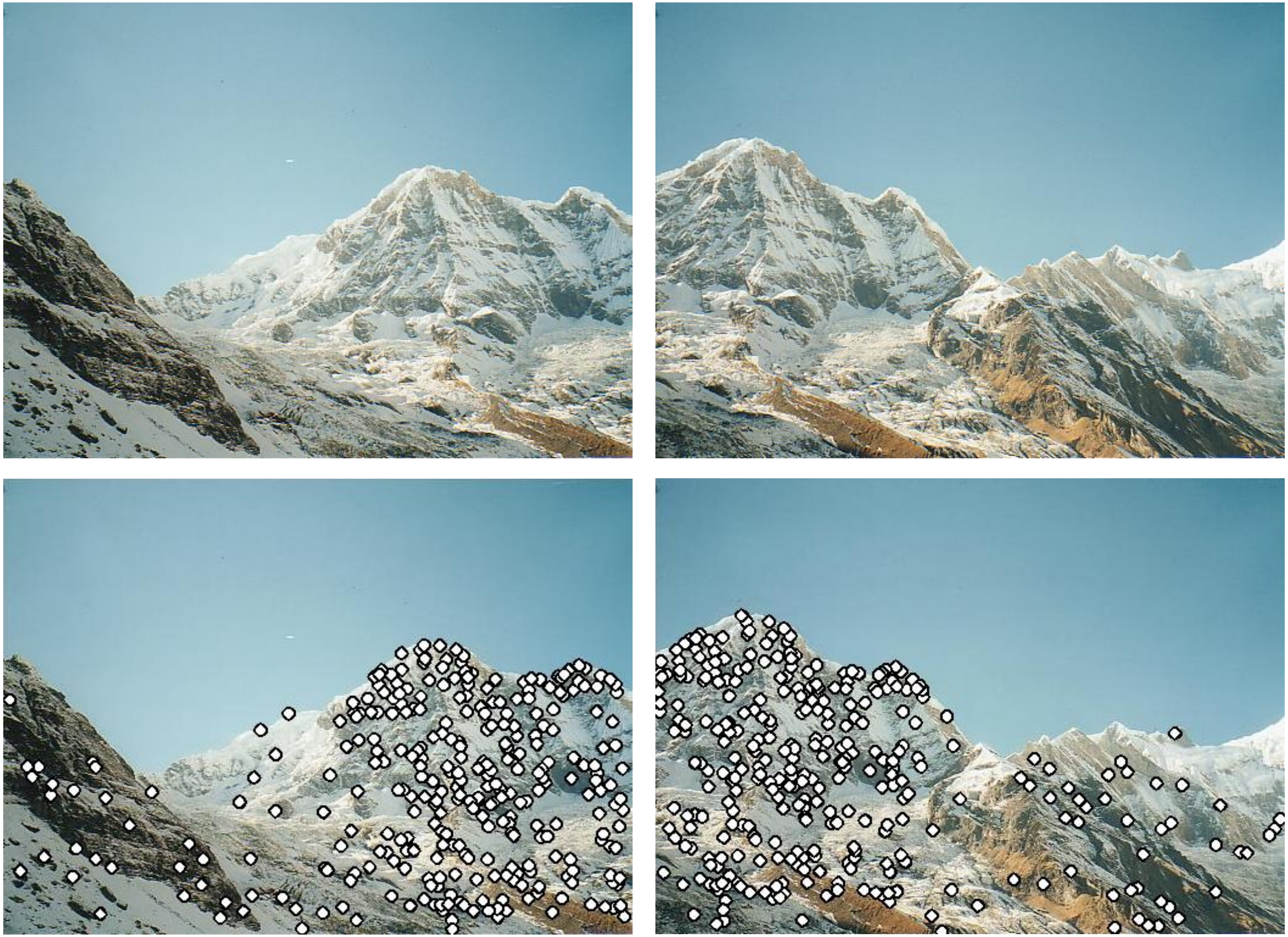
# Invariant Local Features

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

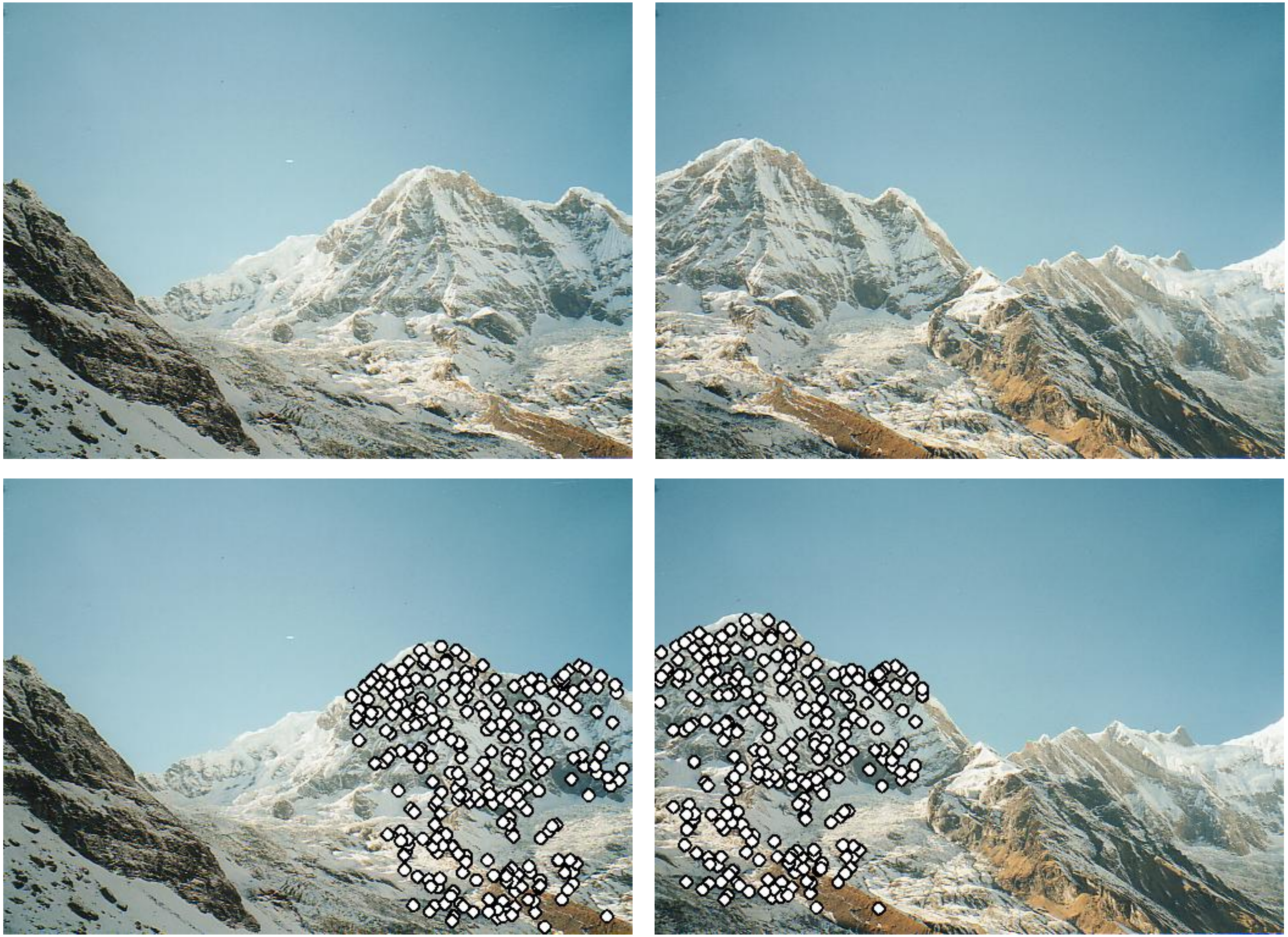


**SIFT Features**

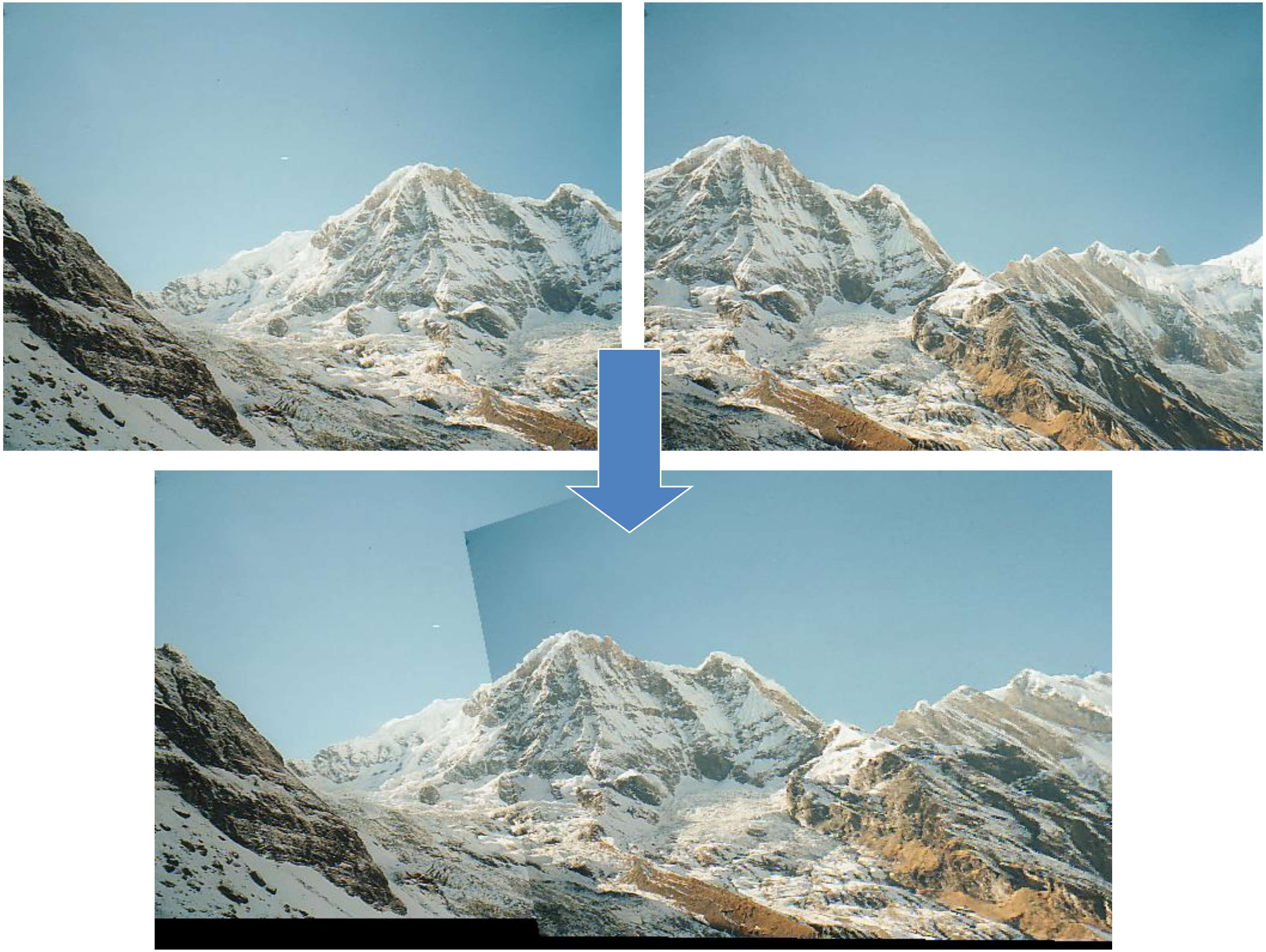
# Example I: mosaicking
# Using SIFT features we match the different images



# Using those matches we estimate the homography relating the two images



# And we can “stitch” the images



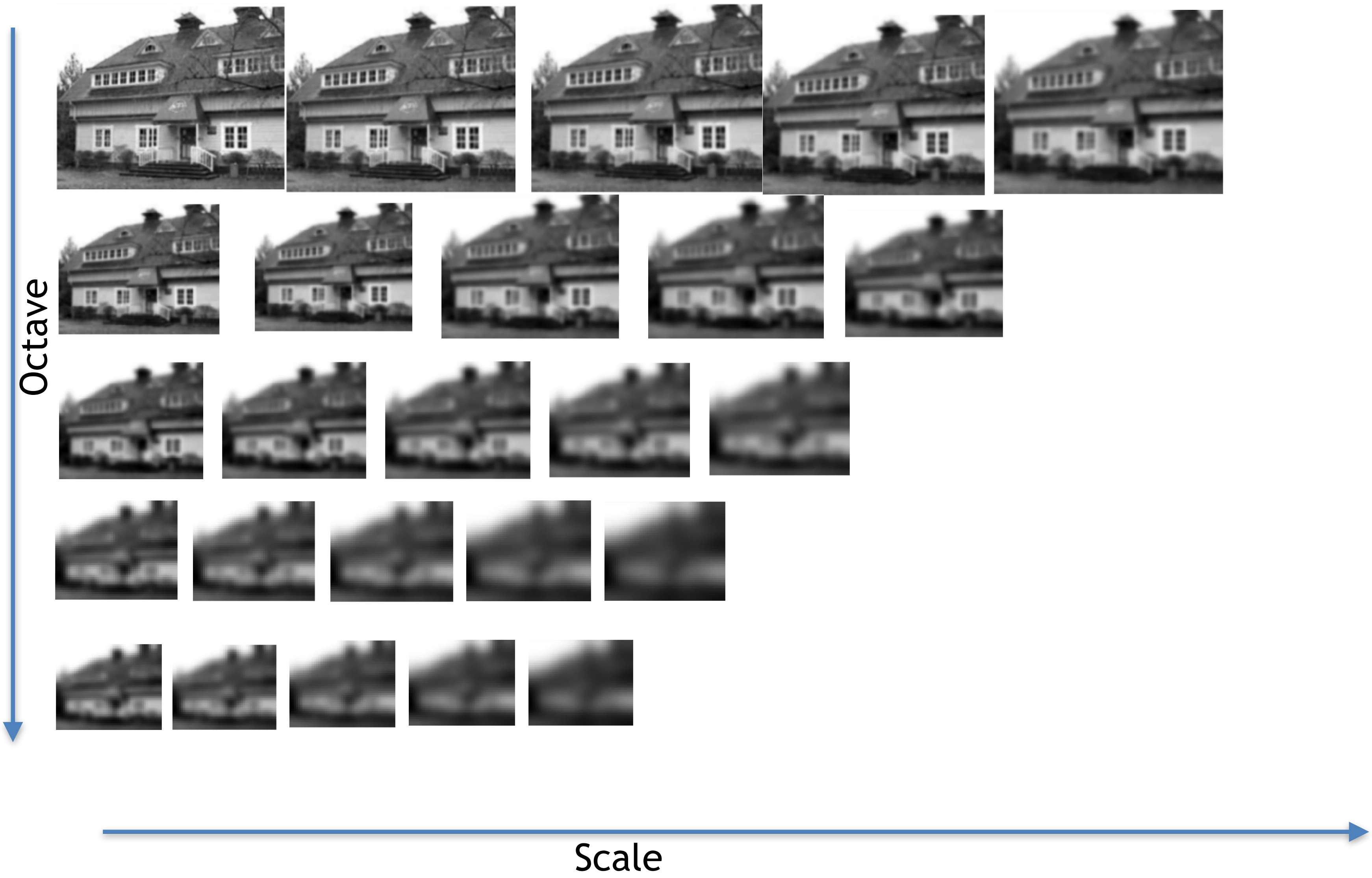
# SIFT Algorithm

1. **Detection**
   – Detect points that can be repeatably selected under location/scale change
2. **Description**
   – Assign orientation to detected feature points
   – Construct a descriptor for image patch around each feature point
3. **Matching**

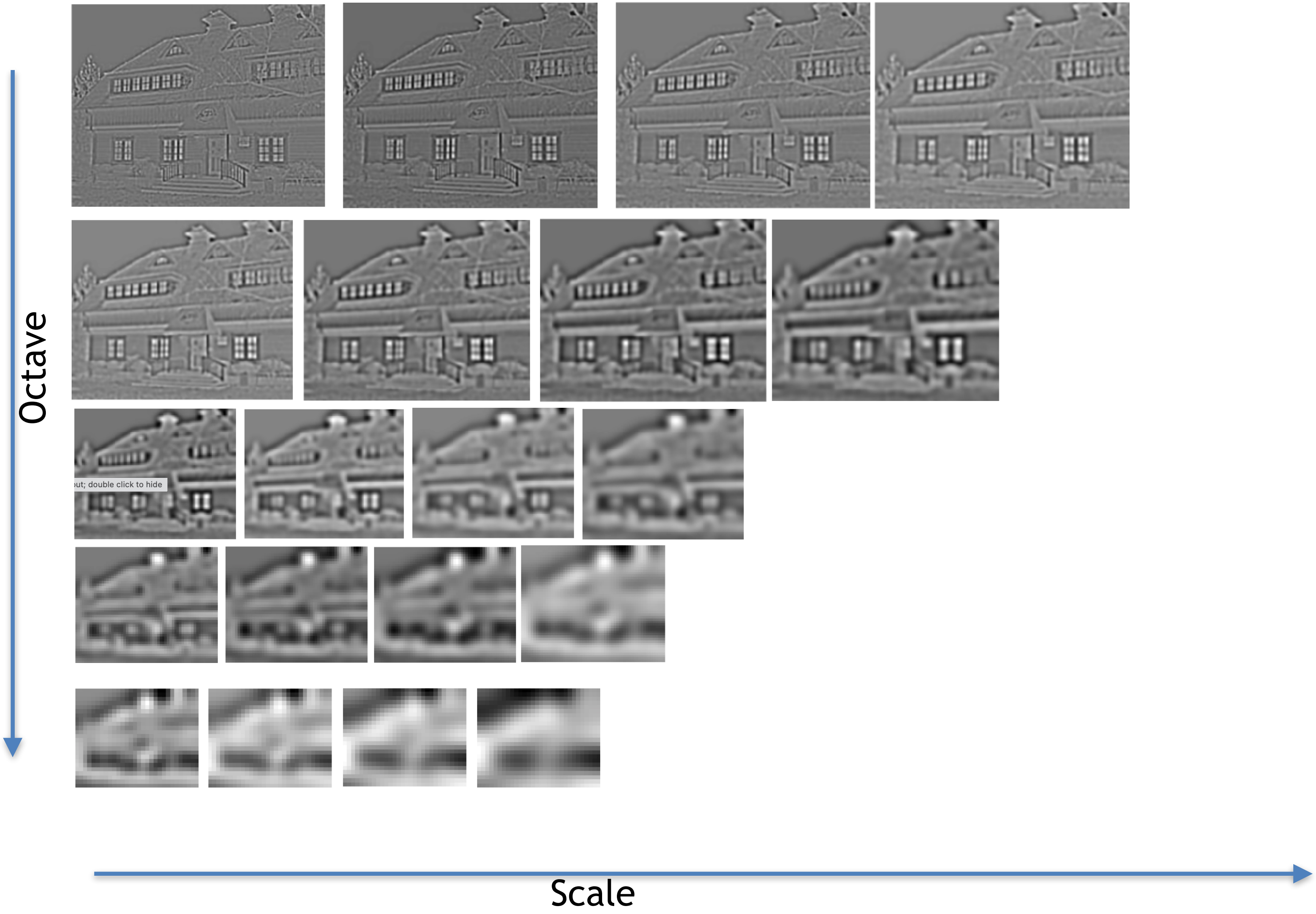
# 1. Keypoint Detection

- This is the stage where the interest points, which are called keypoints in the SIFT framework, are detected.
- For this, the image is convolved with Gaussian filters at different scales, and then the difference of successive Gaussian-blurred images are taken.
- Keypoints are then taken as maxima/minima of the Difference of Gaussians (DoG) that occur at multiple scales.
- This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales.
- If the pixel value is the maximum or minimum among all compared pixels, it is selected as a candidate keypoint.

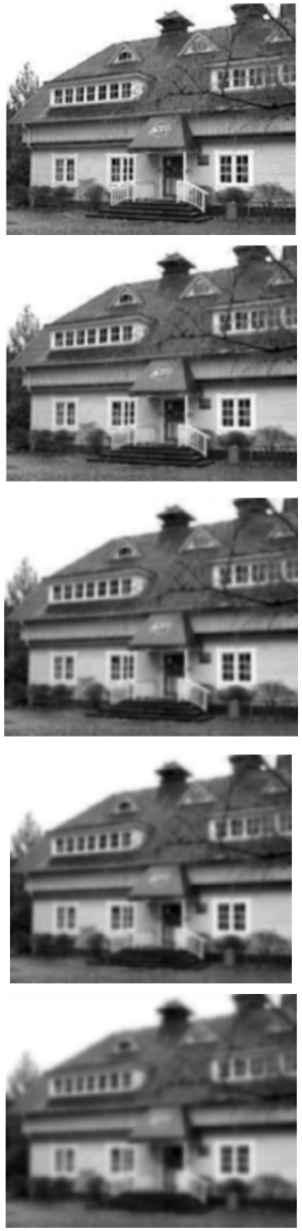
# 1. Keypoint Detection - Gaussians

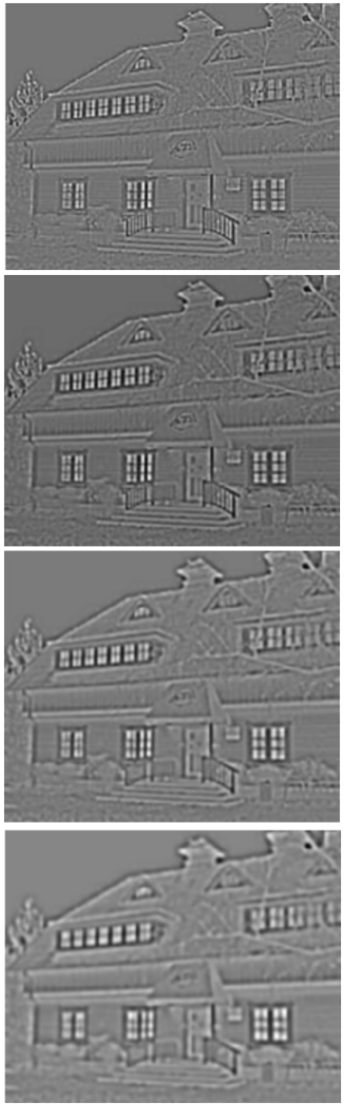


# 1. Keypoint Detection - Difference of Gaussians

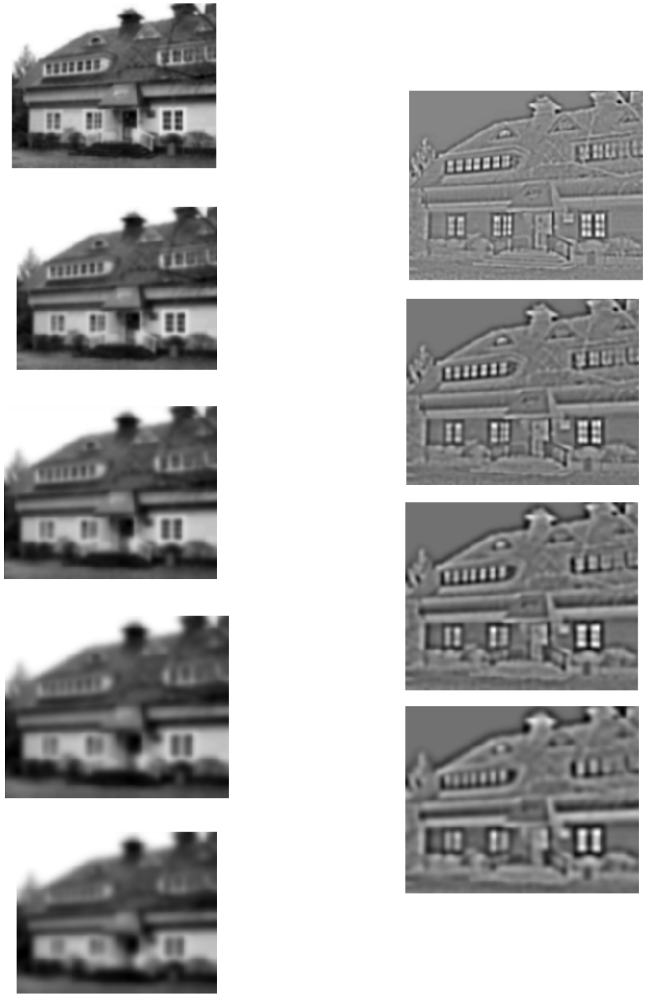


# Difference of Gaussians for the Ist Octave

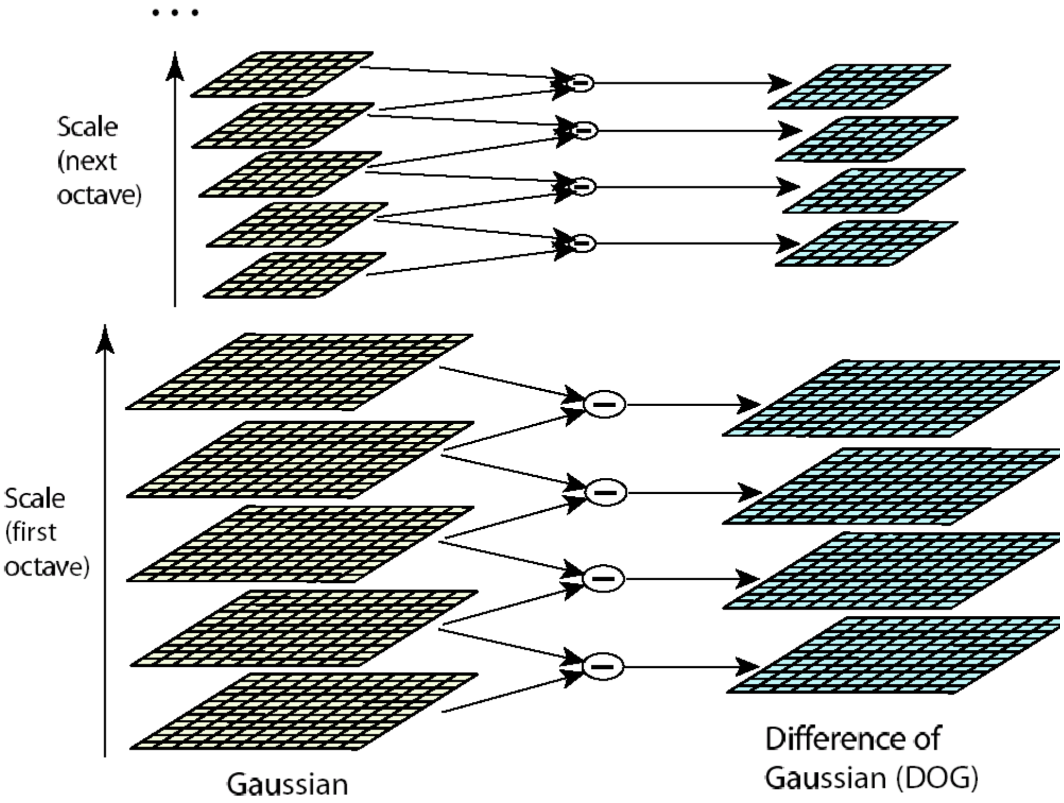


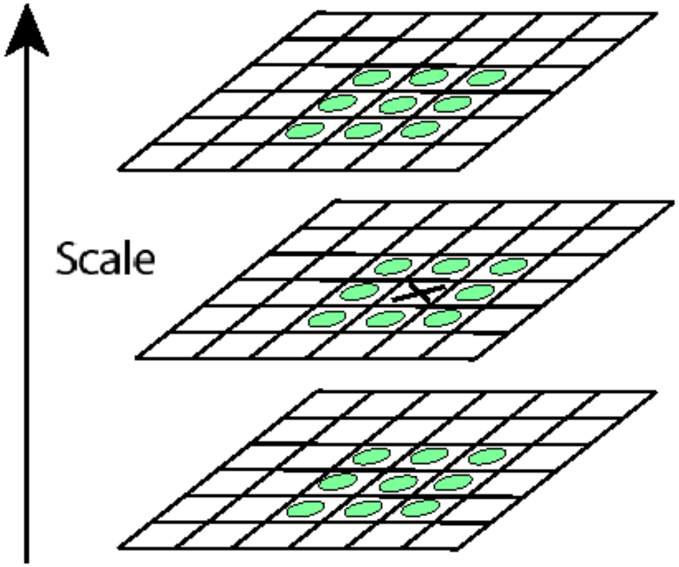


# Difference of Gaussians for the 2nd Octave

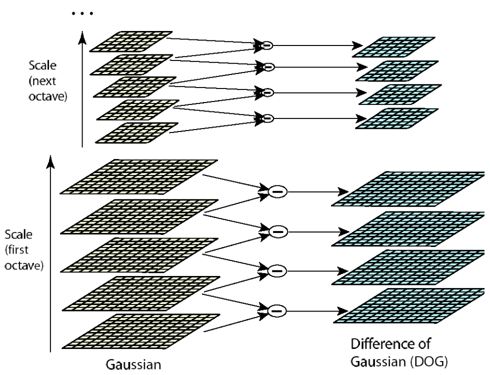


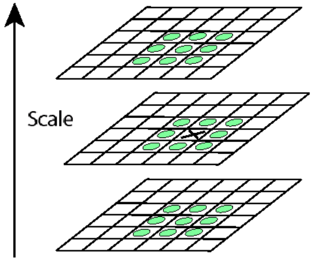
# 1. Feature detection - Key point detection





# 1. Feature detection - Key point detection





Scale of an Image: $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$

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

Difference-of-Gaussian function:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

# 1. Feature detection - Key point detection

Difference-of-Gaussian function:

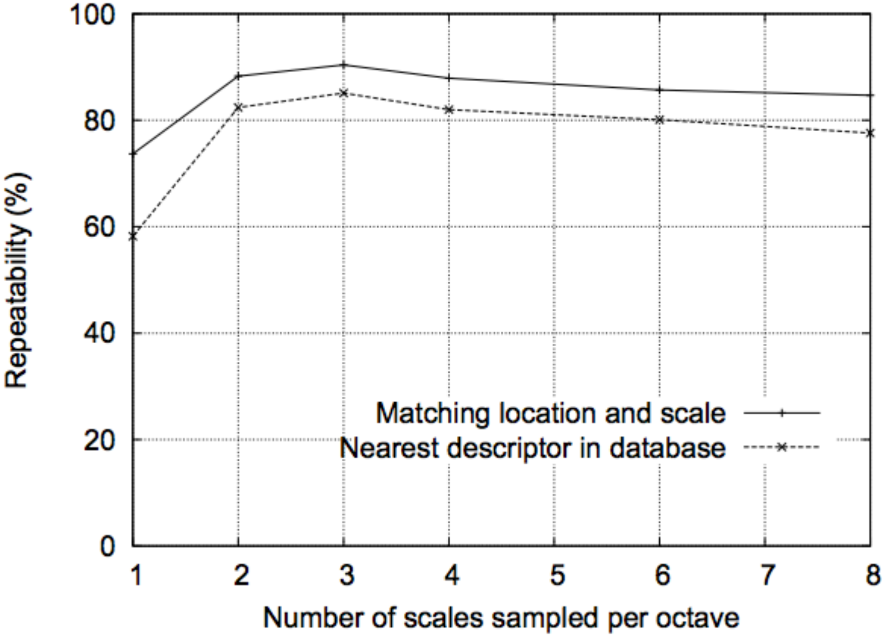
$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

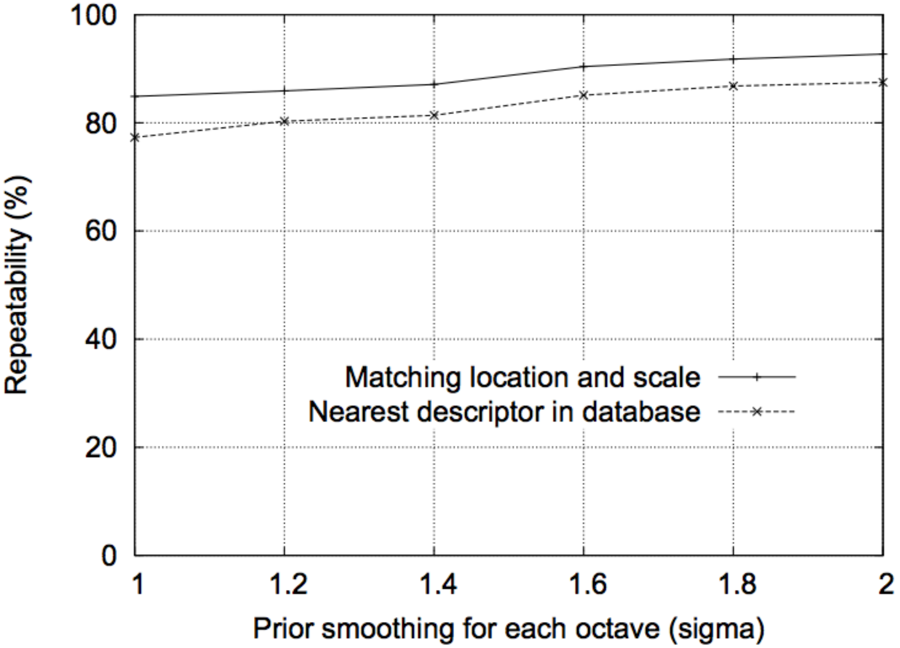
Each octave is divided into, s, intervals, where s is an integer,

$$k = 2^{1/s}$$

Produce s + 3 images in each octave

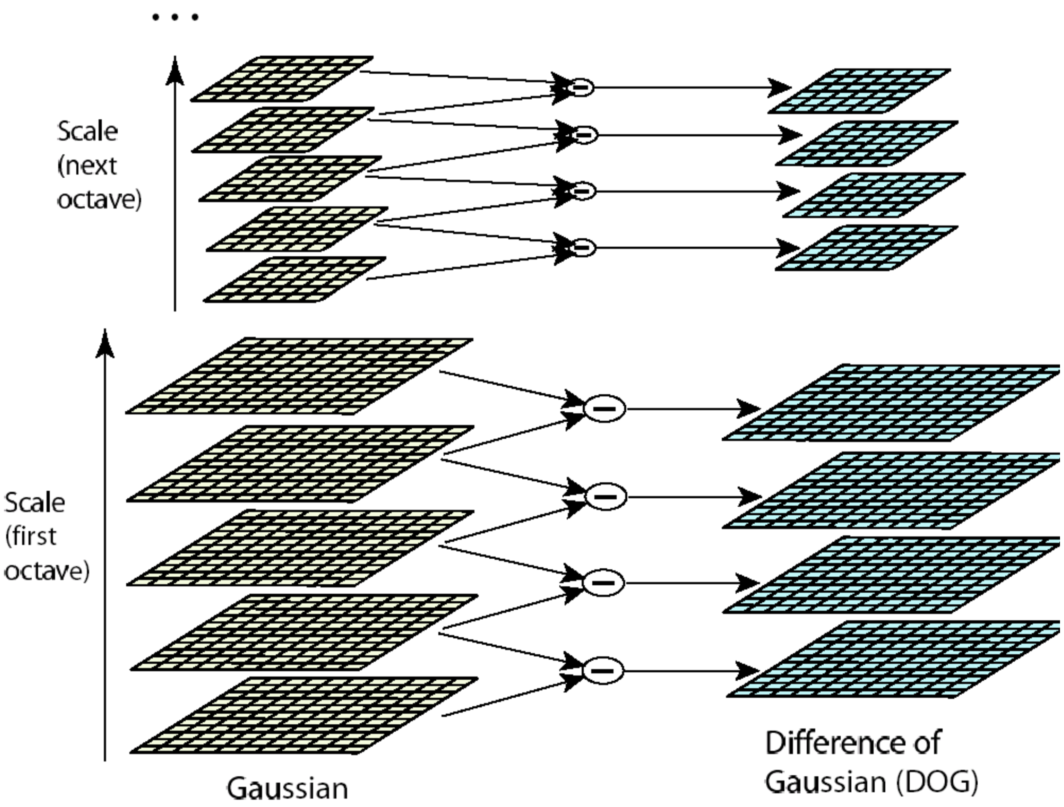
# Repeatability

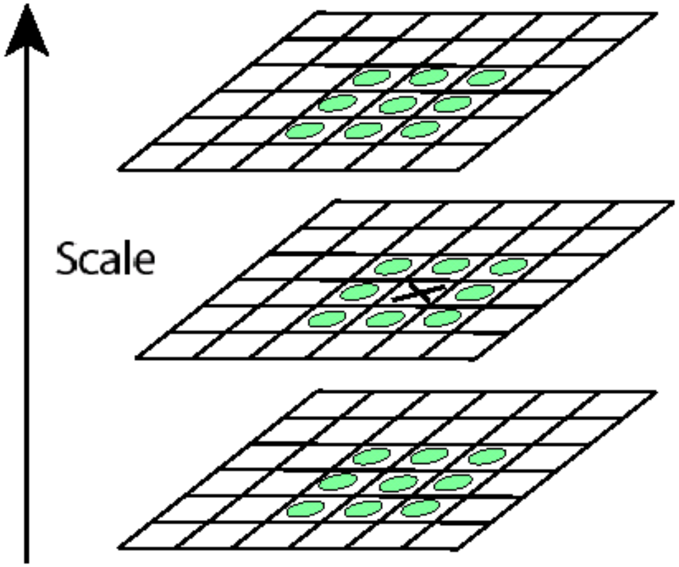




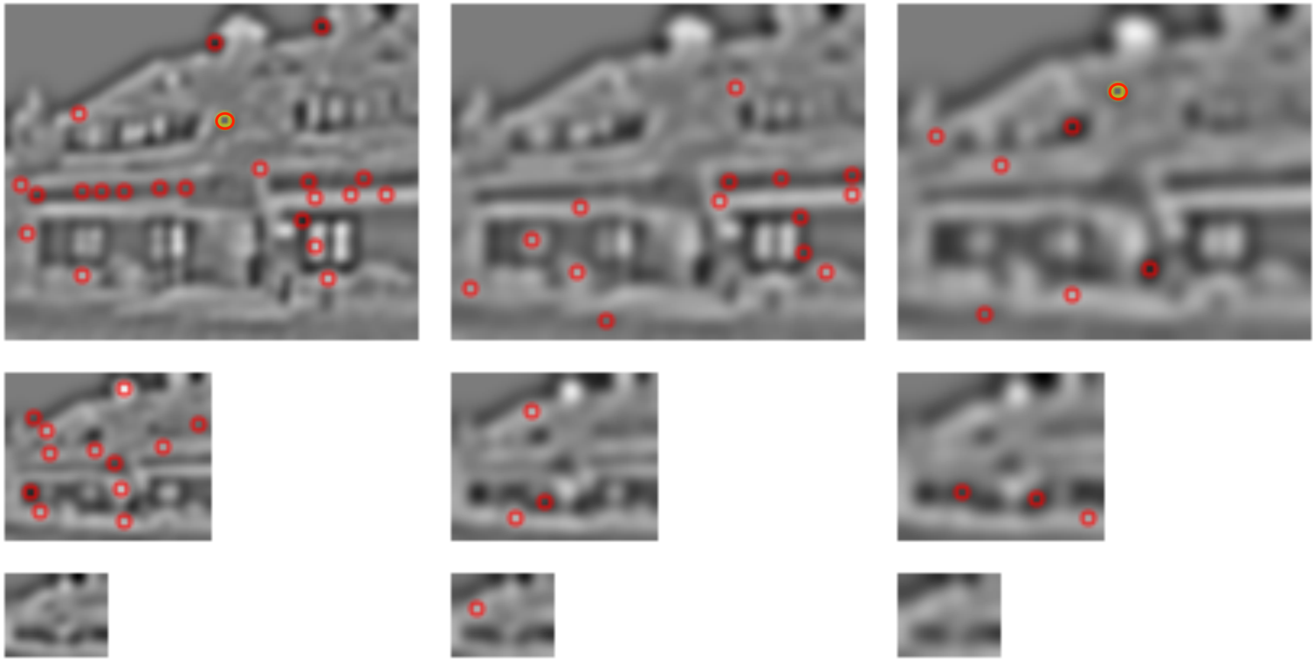
Lowe, 04

# 1. Feature detection - Key point detection

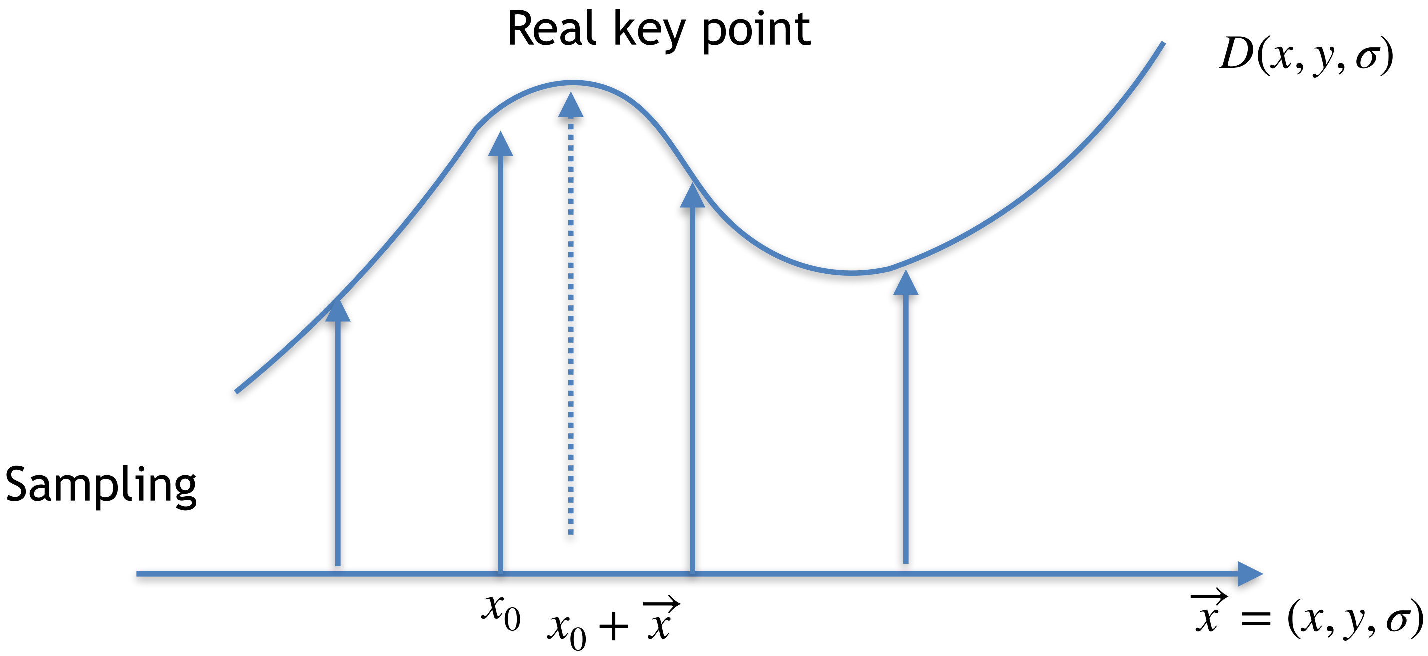




# 1. Feature detection - Key point detection



# 1. Key point localization



# 1. Key point localization

- Detailed fit using data surrounding the keypoint to Localize extrema by fitting a quadratic, to nearby data for location, scale and ratio of principal curvatures

  1) Sub-pixel/sub-scale interpolation using Taylor expansion (D - difference of Gaussian Image)

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

Location of the extrema, $\hat{x} = -\frac{\partial^2 D}{\partial x^2}^{-1} \frac{\partial D}{\partial x}$

by taking a derivative and setting it to zero

# 1. Key point localization

1) Sub-pixel/sub-scale interpolation using Taylor expansion

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

Location of the extrema, $\hat{x} = -\frac{\partial^2 D}{\partial x^2}^{-1} \frac{\partial D}{\partial x}$

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

$$D(\hat{x}) = D + \frac{1}{2}\frac{\partial D}{\partial x}\hat{x}$$

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

key points with low contrast

# 1. Key point localization - Eliminating edge response

1) Principal curvatures can be computed from a 3 x 3 Hessian matrix

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

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

$$D_{xx} = D[x+1, y, \sigma] - 2 \times [x, y, \sigma] + D[x-1, y, \sigma]$$

$$D_{yy} = D[x, y+1, \sigma] - 2 \times D[x, y, \sigma] + D[x, y-1, \sigma]$$

$$D_{\sigma\sigma} = D[x, y, \sigma+1] - 2 \times D[x, y, \sigma] + D[x, y, \sigma-1]$$

$$\hat{x} = -\frac{\partial^2 D}{\partial x^2}^{-1} \frac{\partial D}{\partial x}$$

$$D_{xy} = \frac{(D(x+1, y+1, \sigma) - D(x-1, y+1, \sigma)) - (D(x+1, y-1, \sigma) - D(x-1, y-1, \sigma))}{4}$$

$$D_{x\sigma} = \frac{(D(x+1, y, \sigma+1) - D(x-1, y, \sigma+1)) - (D(x+1, y, \sigma-1) - D(x-1, y, \sigma-1))}{4}$$

$$D_{y\sigma} = \frac{(D(x, y+1, \sigma+1) - D(x, y-1, \sigma+1)) - (D(x, y+1, \sigma-1) - D(x, y-1, \sigma-1))}{4}$$

# 1. Feature detection - Keypoint localization

- Discard low-contrast/edge points
  1) Low contrast: discard keypoints with threshold < 0.03
  2) Edge points: high contrast in one direction, low in the other → compute principal curvatures from eigenvalues of 3x3 Hessian matrix, and limit ratio (discard when r > 10)

$$\mathbf{H} = \left[ \begin{array}{cc} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{array} \right]$$

$$Tr(H) = D_{xx} + D_{yy} = \lambda_1 + \lambda_2$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2$$

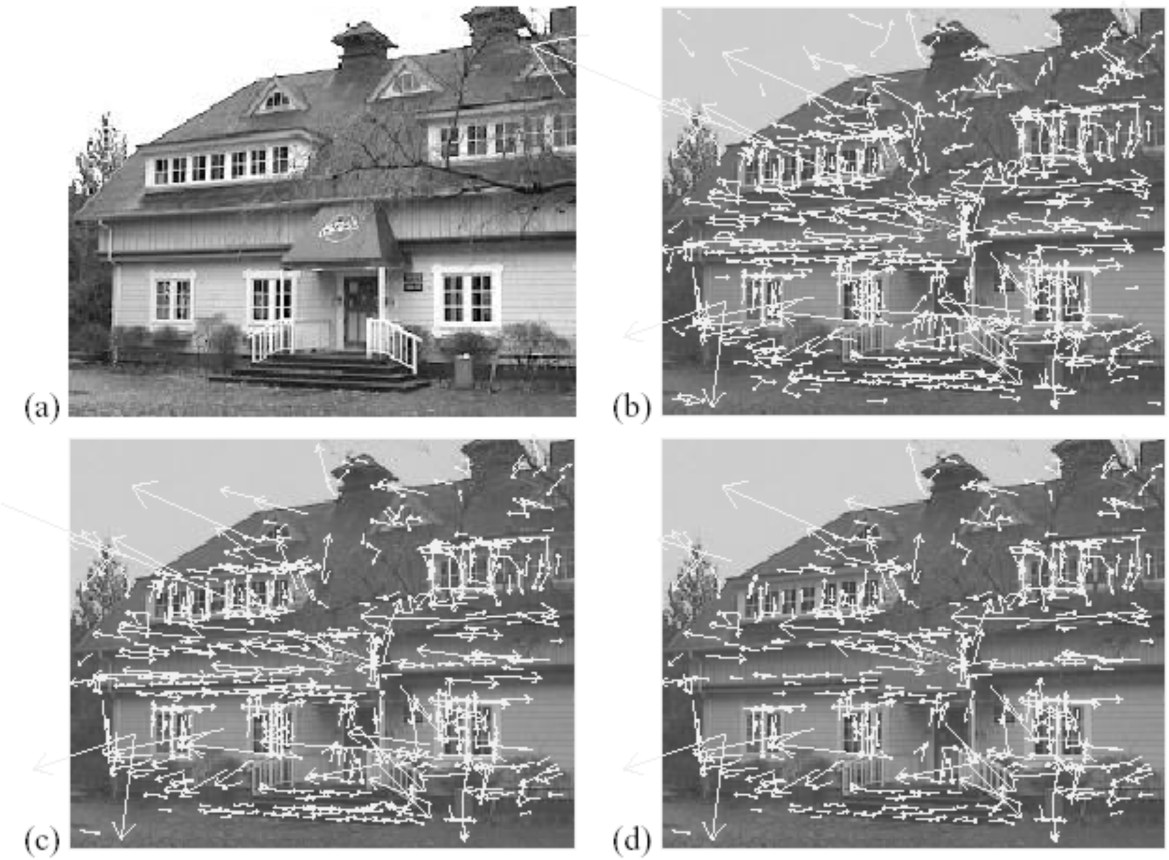
$$r = \frac{\lambda_1}{\lambda_2}$$

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$

r= 10

# 1. Keypoint detection - scale and location

- Example



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

# 2. Orientation Assignment

- Assign orientation to keypoints
  - Assign canonical orientation at peak of smoothed histogram ( L is Gaussian smoothed image)

Gradient magnitude,

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

Orientation,

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

# 2. Orientation Assignment

- Assign orientation to keypoints
  - Create histogram of local gradient directions computed at selected scale

Orientation histogram has 36 bins each covering 10 degrees

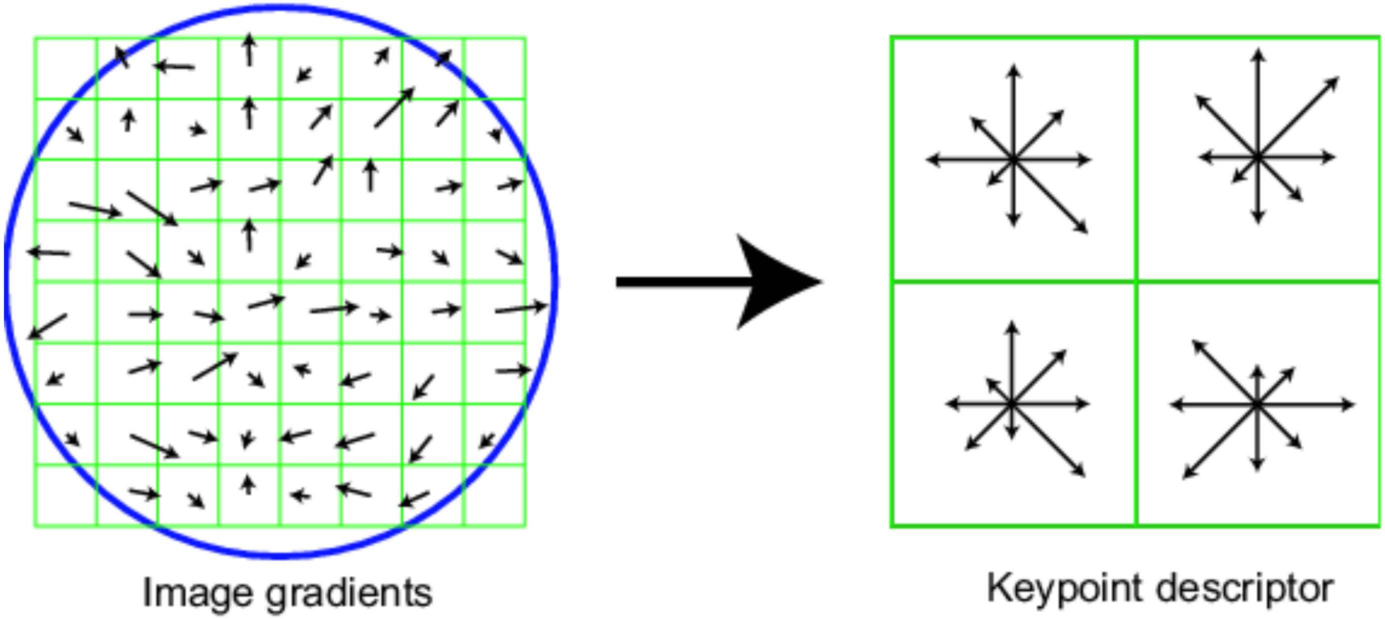
Peaks in the orientation histogram correspond to dominant directions of local gradients.

Any other local peak, within 80% of the highest peak is also used to create a key point with that orientation.

There may be multiple key points with same location and scale but different orientation.

# 2. Feature description

- Construct SIFT descriptor
  - Create array of orientation histograms
  - 8 orientations x 4x4 histogram array = 128 dimensions



# 2. Feature description

- Advantage over simple correlation
  - less sensitive to illumination change
  - robust to deformation, viewpoint change

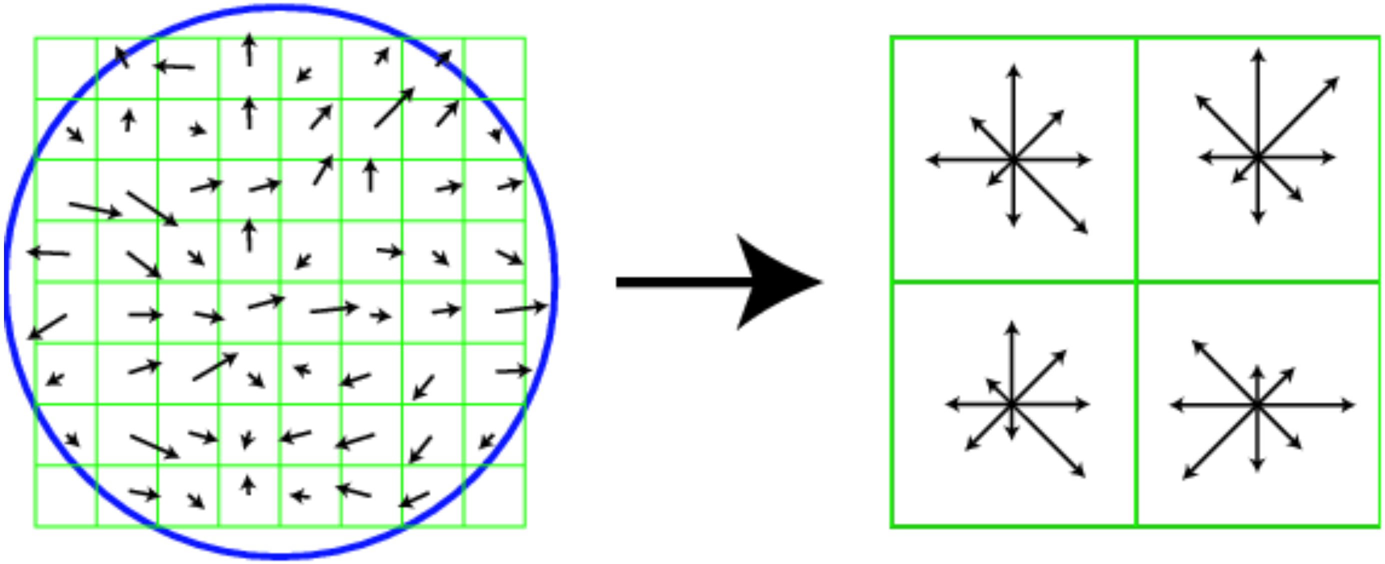


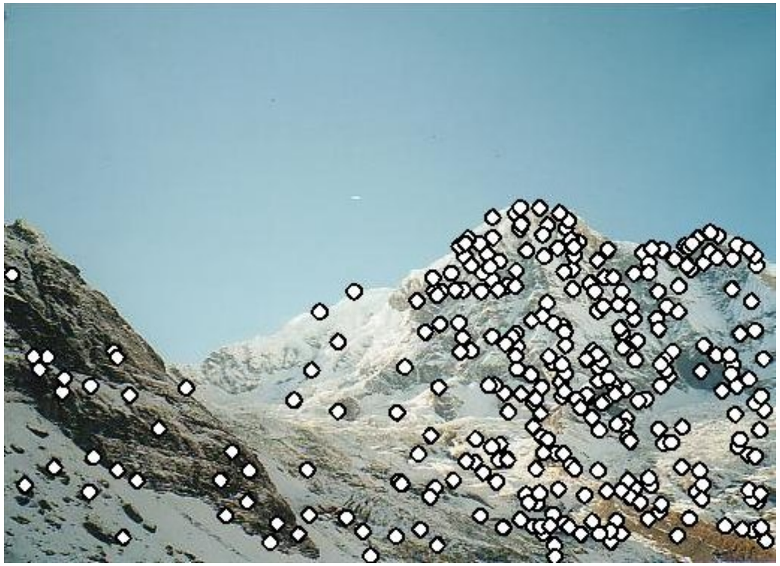
Image gradients

Keypoint descriptor

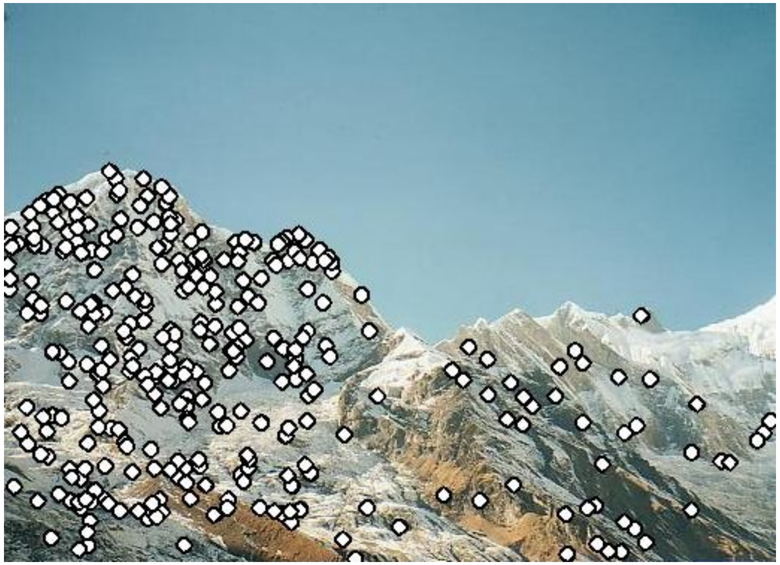
# 3. Feature matching

- For each feature in A, find nearest neighbor in B

**A**



**B**



# 3. Feature matching

- Nearest neighbor search too slow for large database of 128-dimensional data
- **Approximate** nearest neighbor search:
- **Result:** Can give speedup by factor of 1000 while finding nearest neighbor (of interest) 95% of the time

# 3. Feature matching

- Example: 3D object recognition