Distinctive Image Feature from scale-Invariant Keypoints

Soongsil University
Computer Vision Lab
Joo Sung il.

Contents

- Algorithm overview
- Detection of scale-space extrema
- Accurate keypoint localization
- Orientation assignment
- The local image descriptor
- Application to object recognition

Algorithm overview

- Keypoint Detection
  - Make Scale-space (Gaussian)
  - DOG
  - Local extrema detection(detection Candidate Keypoint)
  - reject Low contrast
  - reject edge (ratio between the largest eigenvalue and the smaller one)
- Make Descriptor
  - Orientation and Gradient Magnitude assignment
  - Descriptor representation
- Matching

Detection of scale-space extrema

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

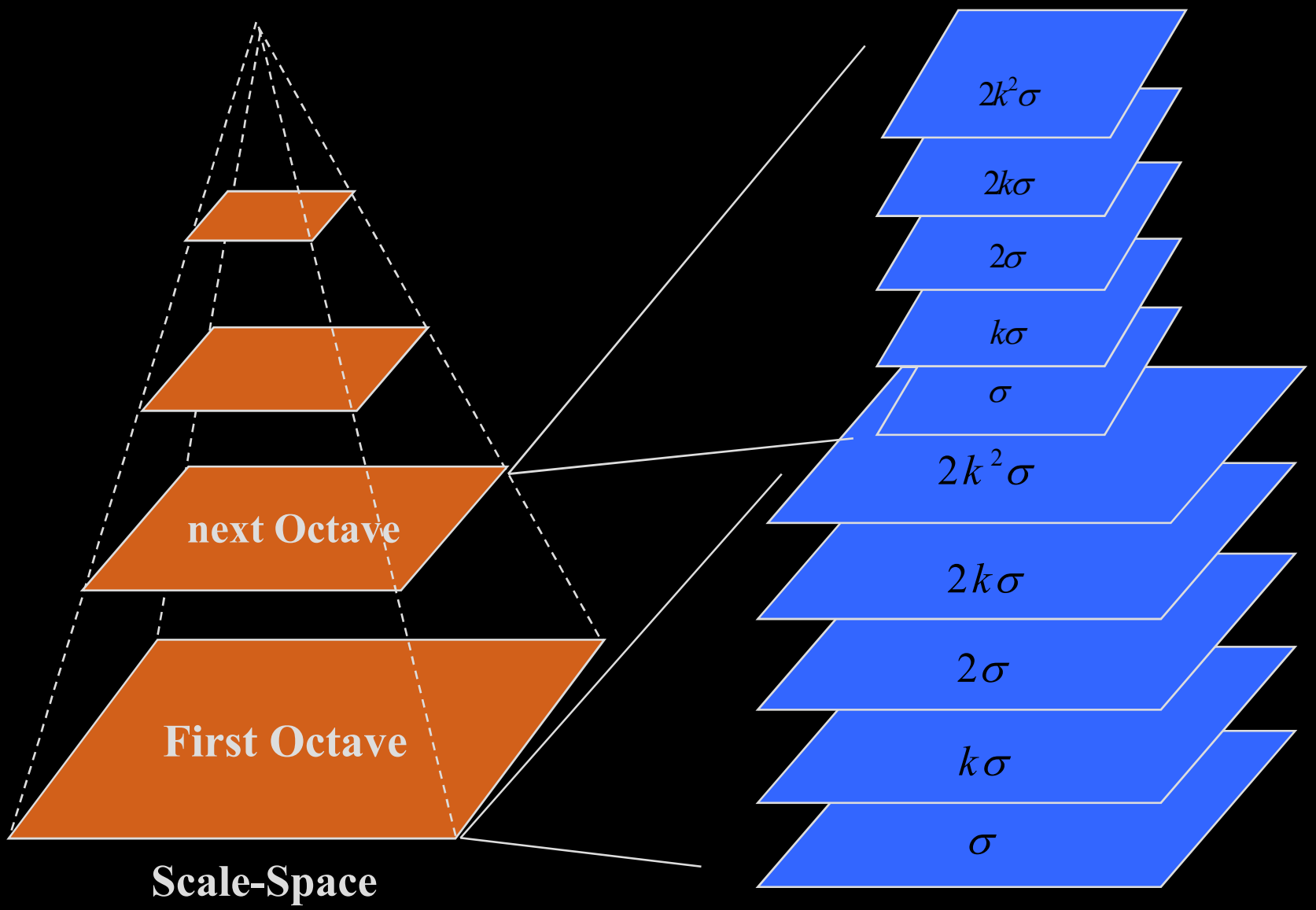
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

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

- Why?

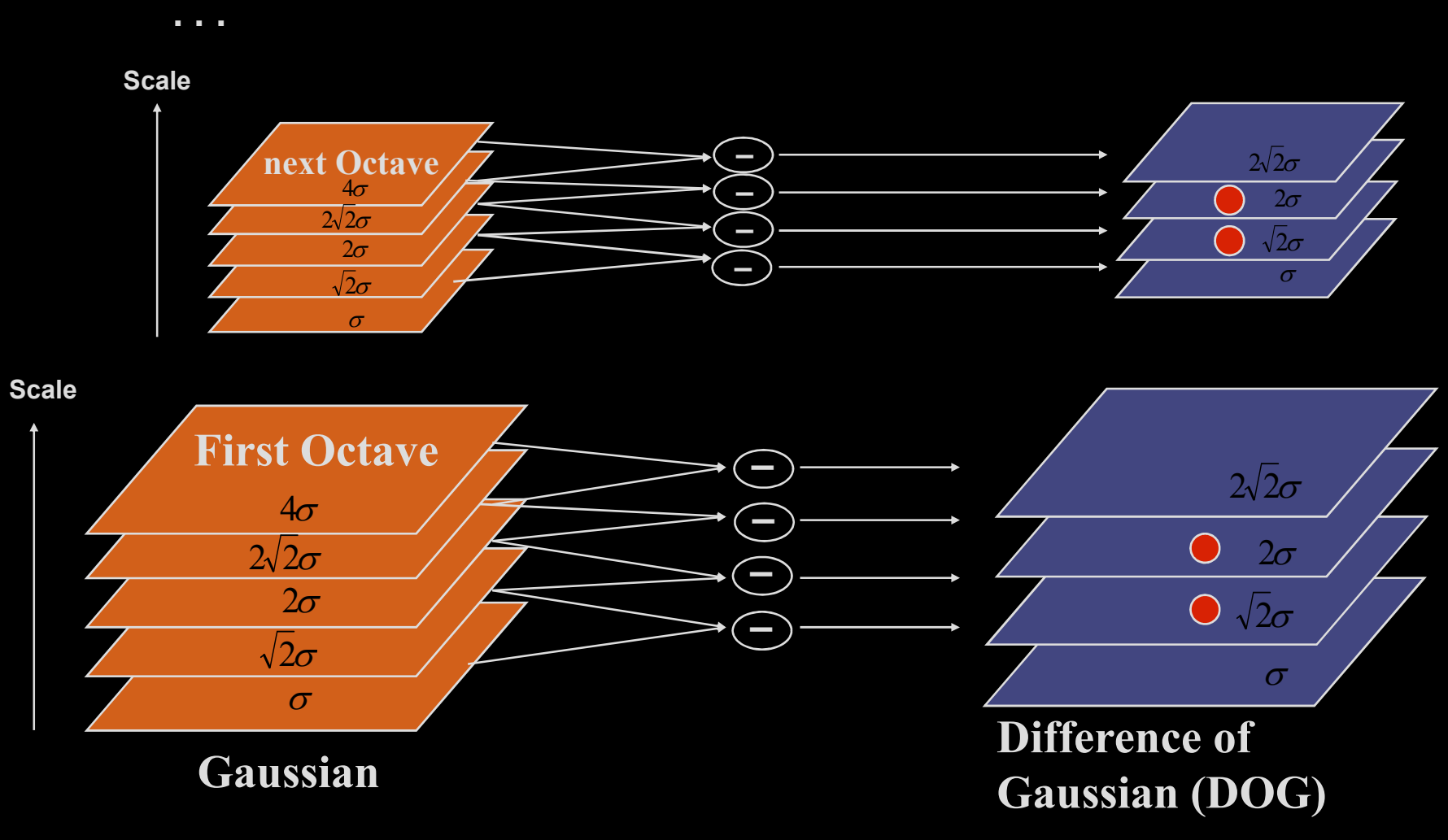
L need to be computed for scale space feature description

Detection of scale-space extrema



Detection of scale-space extrema(cont)

S+3 : S is number of keypoint group



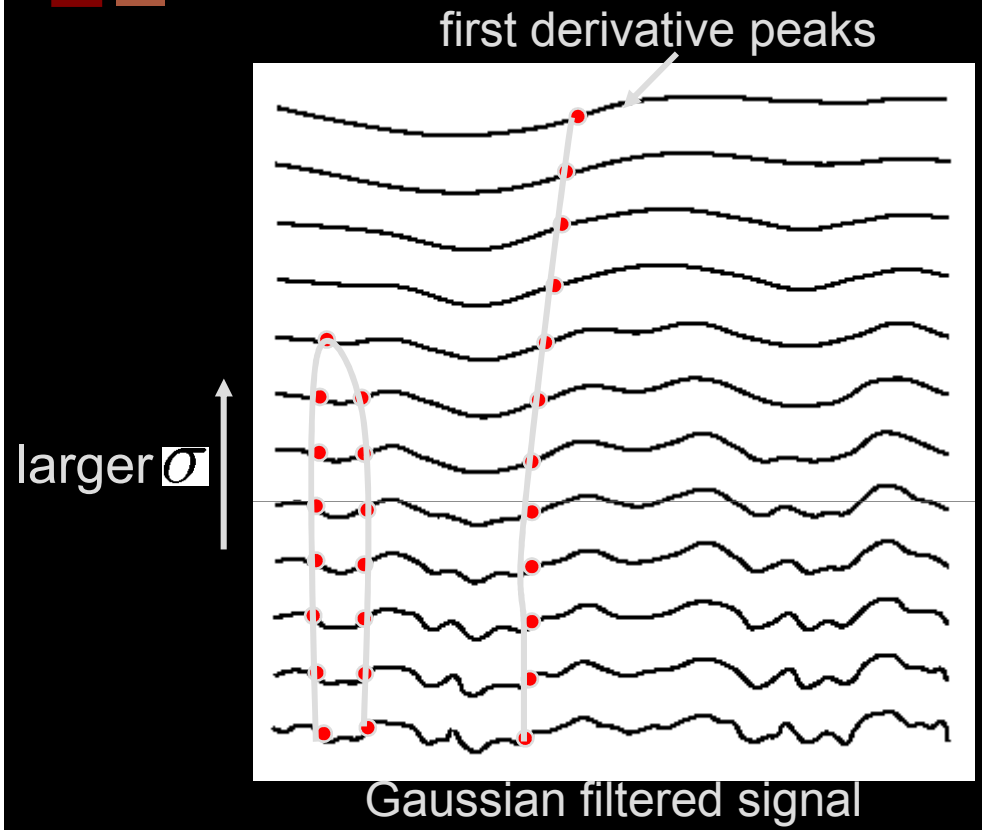
Detection of scale-space extrema(cont)

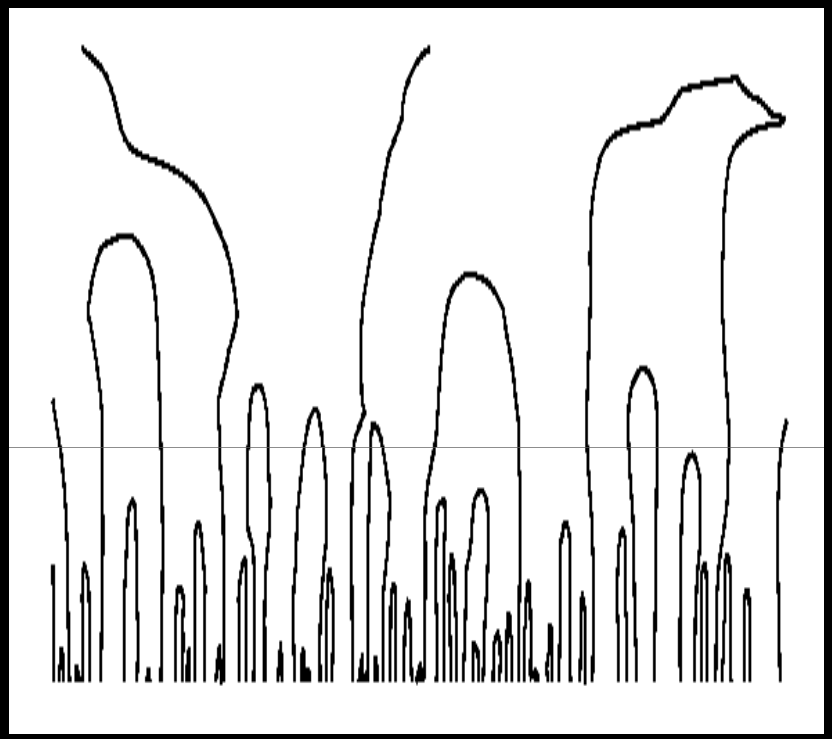
$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G$$

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2 \nabla^2 G$$

Why make Scale-Space ?





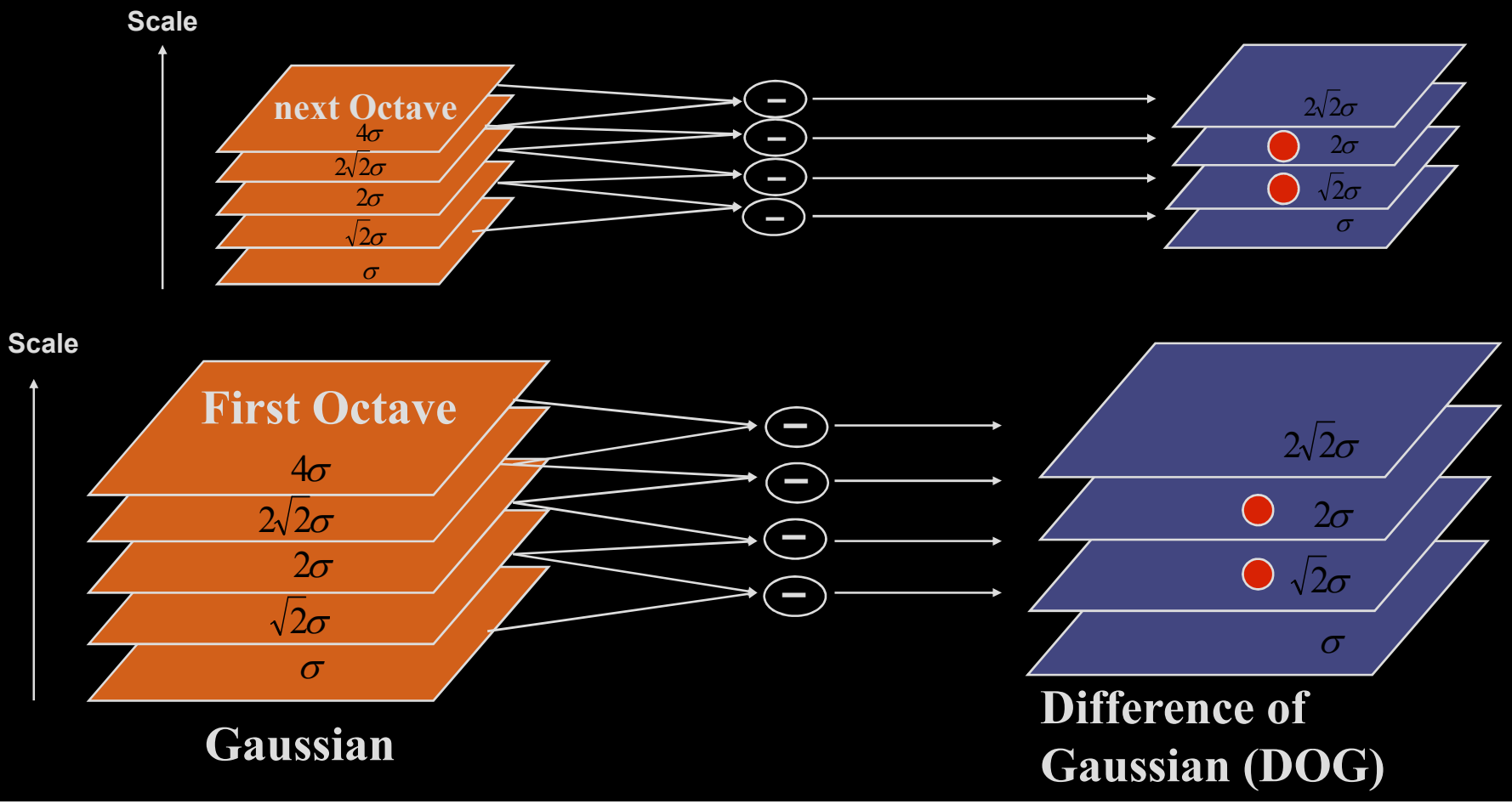
- Properties of scale space (w/ Gaussian smoothing)
  - edge position may shift with increasing scale (σ)
  - two edges may merge with increasing scale
  - an edge may ***not*** split into two with increasing scale

Why make Octave ?

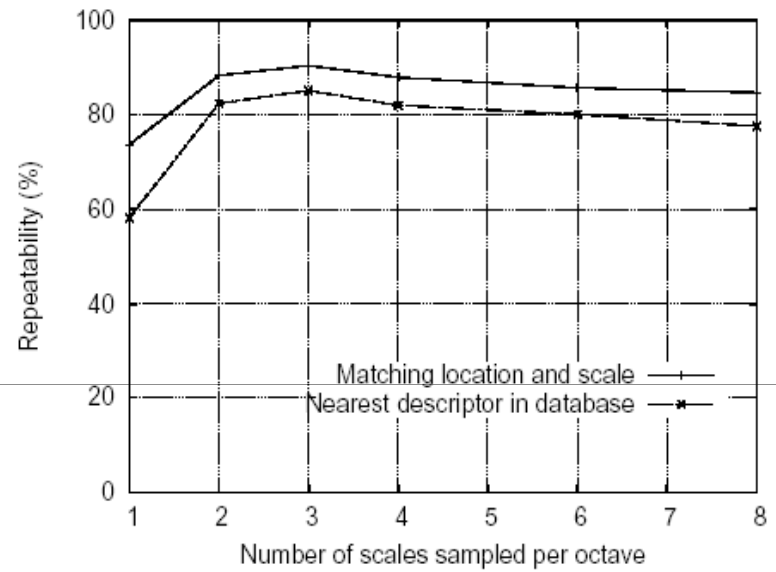
$$k = 2^{\frac{1}{s}}$$

Why?

keypoints



Frequency of sampling in scale



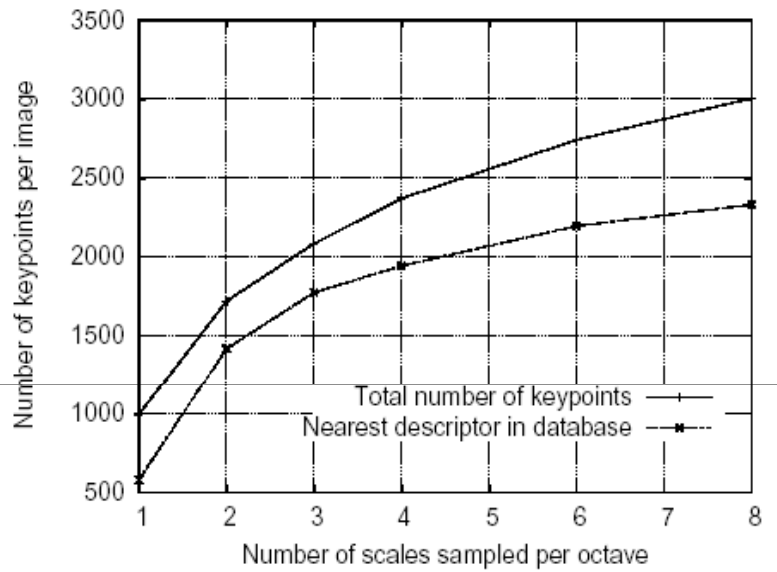


Figure 3: The top line of the first graph shows the percent of keypoints that are repeatably detected at the same location and scale in a transformed image as a function of the number of scales sampled per octave. The lower line shows the percent of keypoints that have their descriptors correctly matched to a large database. The second graph shows the total number of keypoints detected in a typical image as a function of the number of scale samples.

Frequency of sampling in the spatial domain

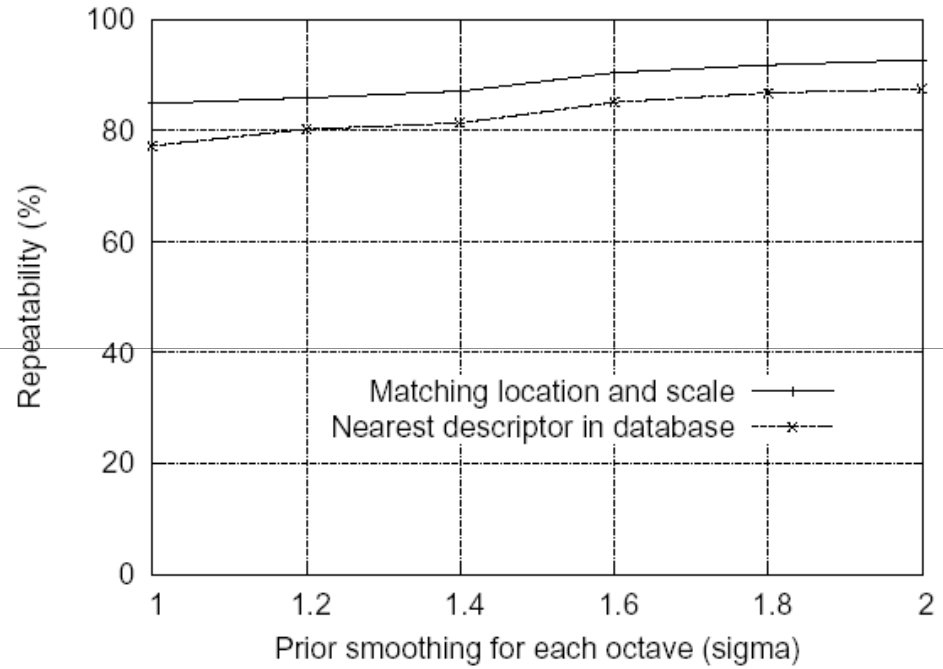
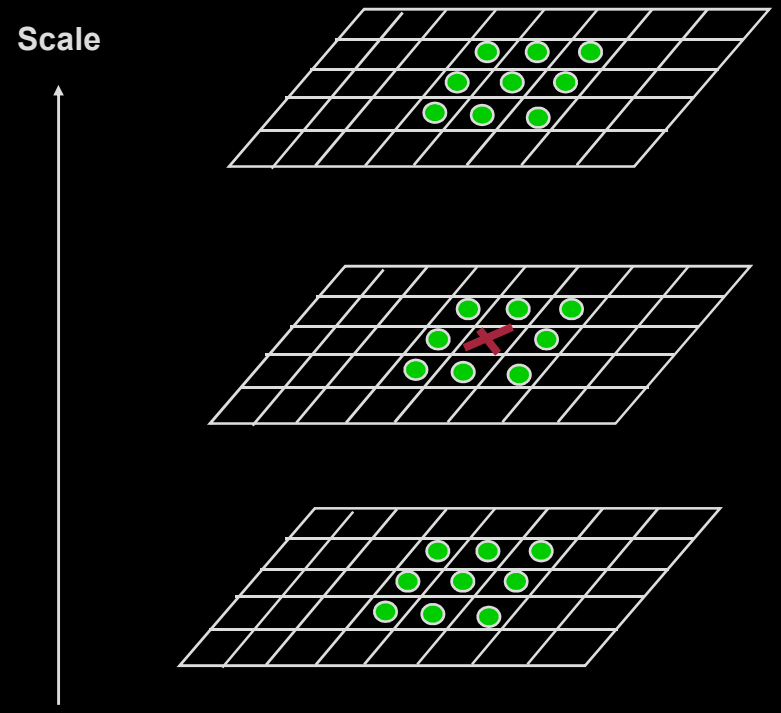


Figure 4: The top line in the graph shows the percent of keypoint locations that are repeatably detected in a transformed image as a function of the prior image smoothing for the first level of each octave. The lower line shows the percent of descriptors correctly matched against a large database.

Local Extrema Detection



- **Local Extrama detection**
  - each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below

Accurate Keypoint Localization(1)

❖ Low Contrast

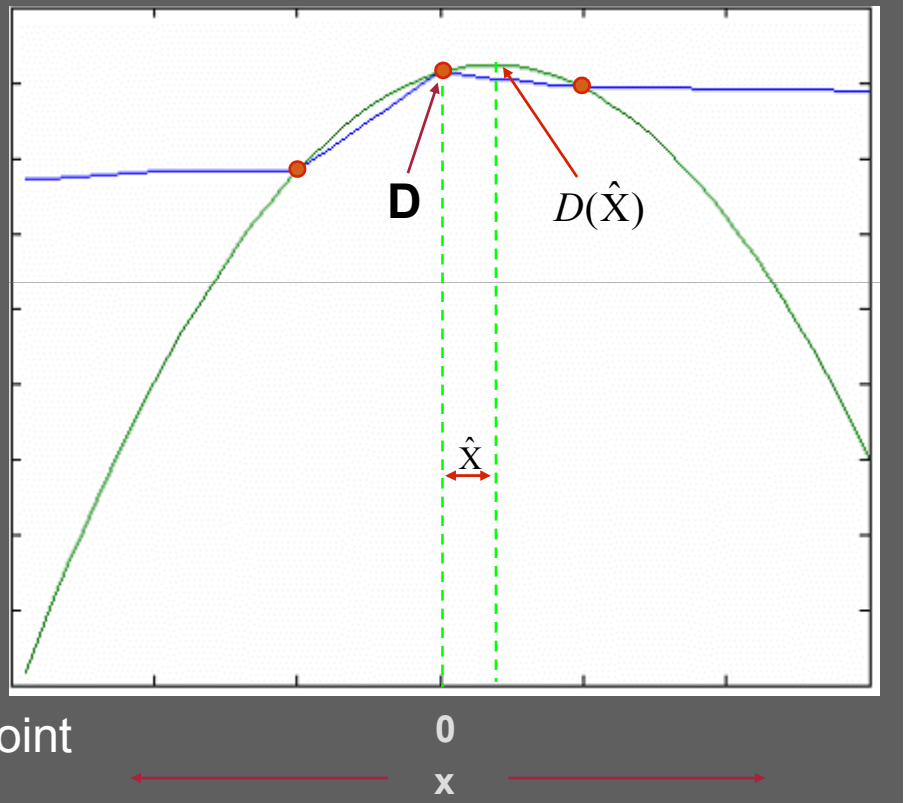
- 3D quadratic function is fit to the local sample points

$$X = (x, y, \sigma)^T$$

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \qquad (2)$$

$$\hat{X} = -\frac{\partial^2 D^{-1}}{\partial X^2} \frac{\partial D}{\partial X} \qquad (3)$$

$$D(\hat{X}) = D + \frac{1}{2} \frac{\partial D^T}{\partial X} \hat{X}$$



$0 = \frac{\partial D}{\partial X} + \frac{\partial^2 D}{\partial X^2} X$ is the location of the keypoint

If X' is > 0.5 in any dimension, process repeated

All extrema with a value of $|D(\hat{X})|$ less than 0.03 were discarded (as before, we assume image pixel values in the range[0,1]).

Accurate Keypoint Localization(2)

❖ Low Contrast

- This is a 3x3 linear system

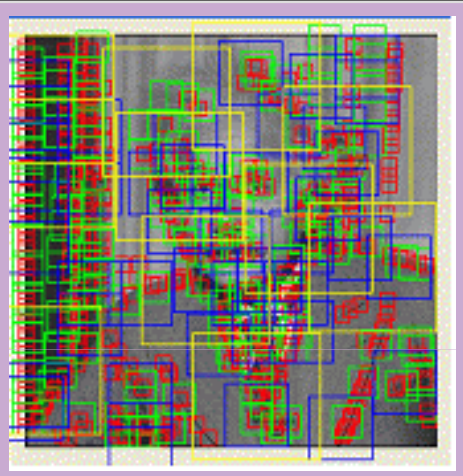
$$\hat{\mathrm{X}} = -\frac{\partial^2 D^{-1}}{\partial \mathrm{X}^2}\frac{\partial D}{\partial \mathrm{X}}$$

$$\begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial xy} & \frac{\partial^2 D}{\partial x\sigma} \\ \frac{\partial^2 D}{\partial xy} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y\sigma} \\ \frac{\partial^2 D}{\partial x\sigma} & \frac{\partial^2 D}{\partial y\sigma} & \frac{\partial^2 D}{\partial \sigma^2} \end{bmatrix} \begin{bmatrix} x \\ y \\ \sigma \end{bmatrix} = -\begin{bmatrix} \frac{\partial D}{\partial x} \\ \frac{\partial D}{\partial y} \\ \frac{\partial D}{\partial \sigma} \end{bmatrix}$$

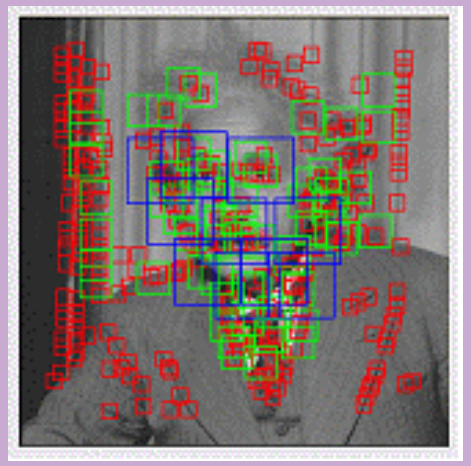
- Derivatives approximated by finite differences,

$$\frac{\partial^2 D}{\partial x^2} = \frac{D_k^{x+1,y} - 2D_k^{x,y} + D_k^{x-1,y}}{1} \qquad \frac{\partial D}{\partial x} = \frac{D_k^{x+1,y} - D_k^{x-1,y}}{2}$$

$$\frac{\partial^2 D}{\partial xy} = \frac{(D_k^{x+1,y+1} - D_k^{x-1,y+1}) - (D_k^{x+1,y-1} - D_k^{x-1,y-1})}{4}$$



583 Keypoints



335 Keypoints

Accurate Keypoint Localization(3)

❖ **Eliminating edge responses**

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \qquad (4)$$

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

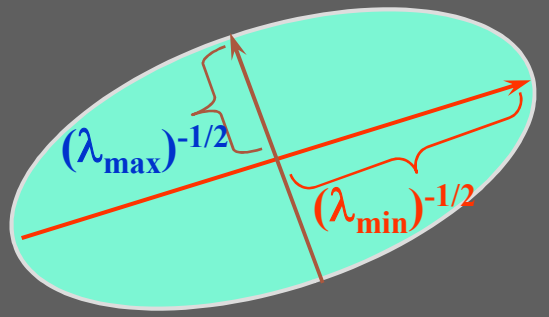
$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$

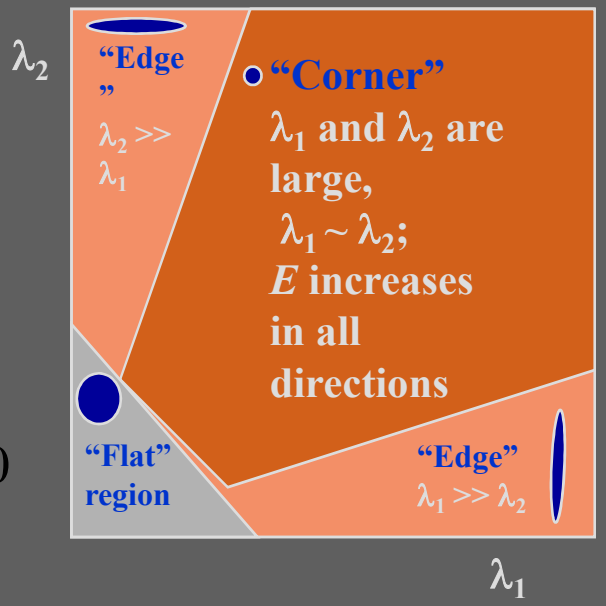
$$\alpha = r\beta$$

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$

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

The experimental value $r = 10$





Accurate Keypoint Localization(4)

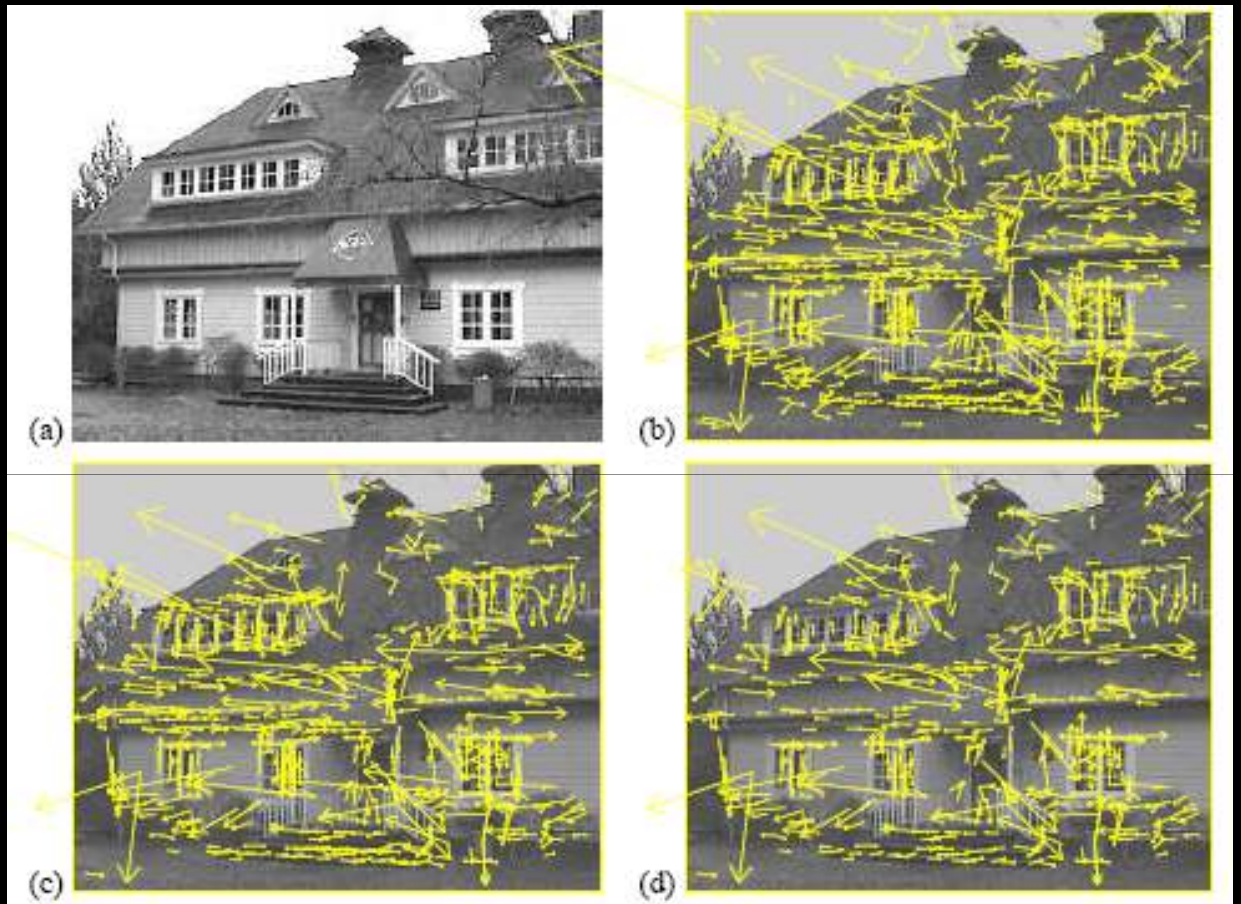
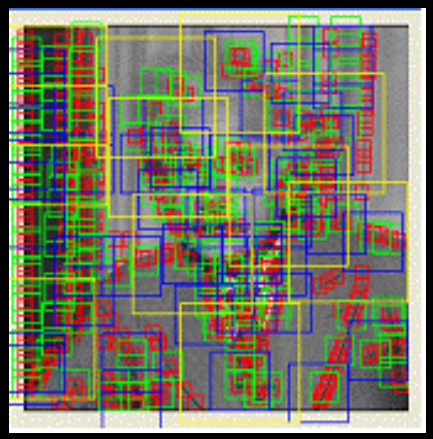
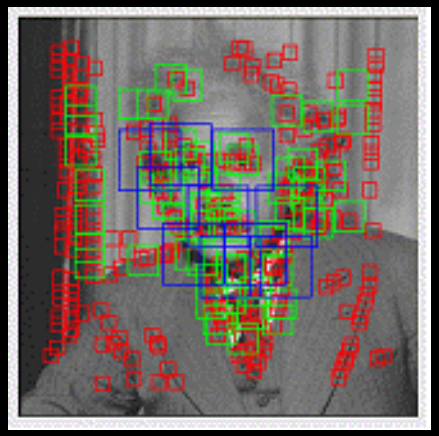


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

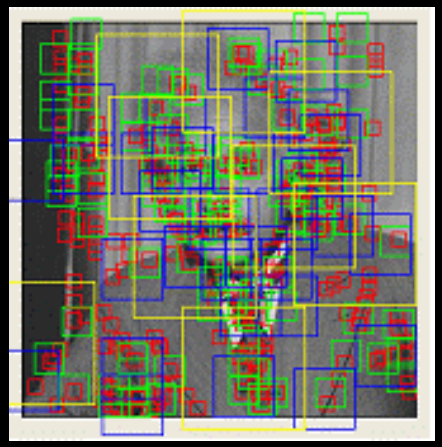
Accurate Keypoint Localization(5)



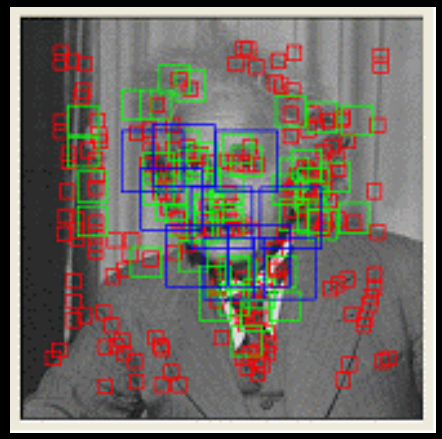
Local Maximum or Minimum
-> 583 Keypoints



Local Maximum or Minimum
Low Contrart
-> 335 Keypoints

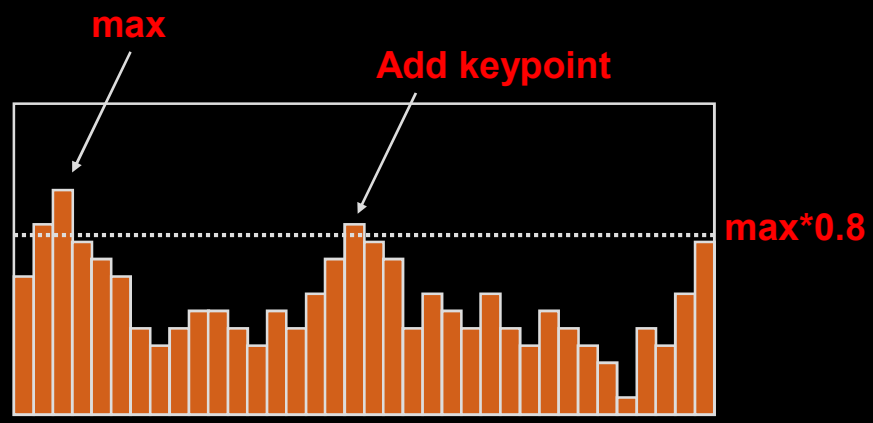


Local Maximum or Minimum
Ratio Principal curvature
-> 370 Keypoints



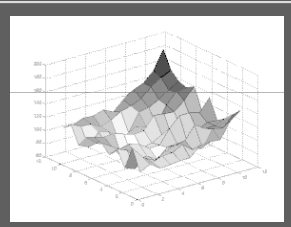
Local Maximum or Minimum
Ratio Principal curvature
Low Contrast
-> 244 Keypoints

Orientation assignment



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

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



- **Make histogram(**36 bins covering the 360 degree Range)
- **Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the keypoint.**

$$\exp\left(\frac{-(i^2 + j^2)}{2(1.5\sigma)^2}\right)$$

- **Histogram smooth**

$$hist[i] = 0.25 * hist[i-1] + 0.5 * hist[i] + 0.25 * hist[i+1]$$

- **Find Maximum**
- **Local peak that is within 80% of the highest peak is used to also create a keypoint with that orientation**

Descriptor summary

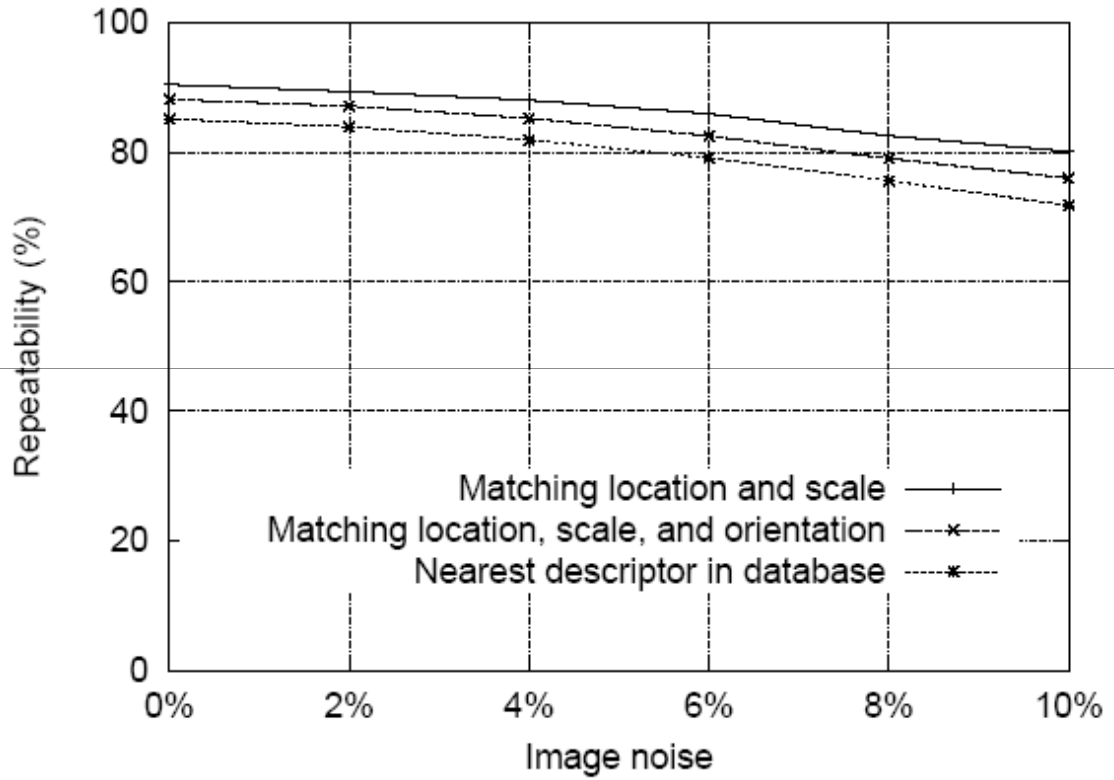
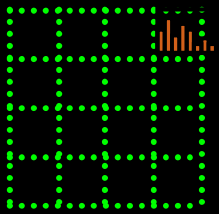


Figure 6: The top line in the graph shows the percent of keypoint locations and scales that are repeatably detected as a function of pixel noise. The second line shows the repeatability after also requiring agreement in orientation. The bottom line shows the final percent of descriptors correctly matched to a large database.

Descriptor summary

Descriptor make by max orientation.([4][4][8])



Converts the array of orientation histograms into a feature's descriptor vector.([4][4][8] -> [128])

The vector is normalized to unit length.

$Sum = 1$

Thresholding the values in the unit feature vector to each be no larger than 0.2.

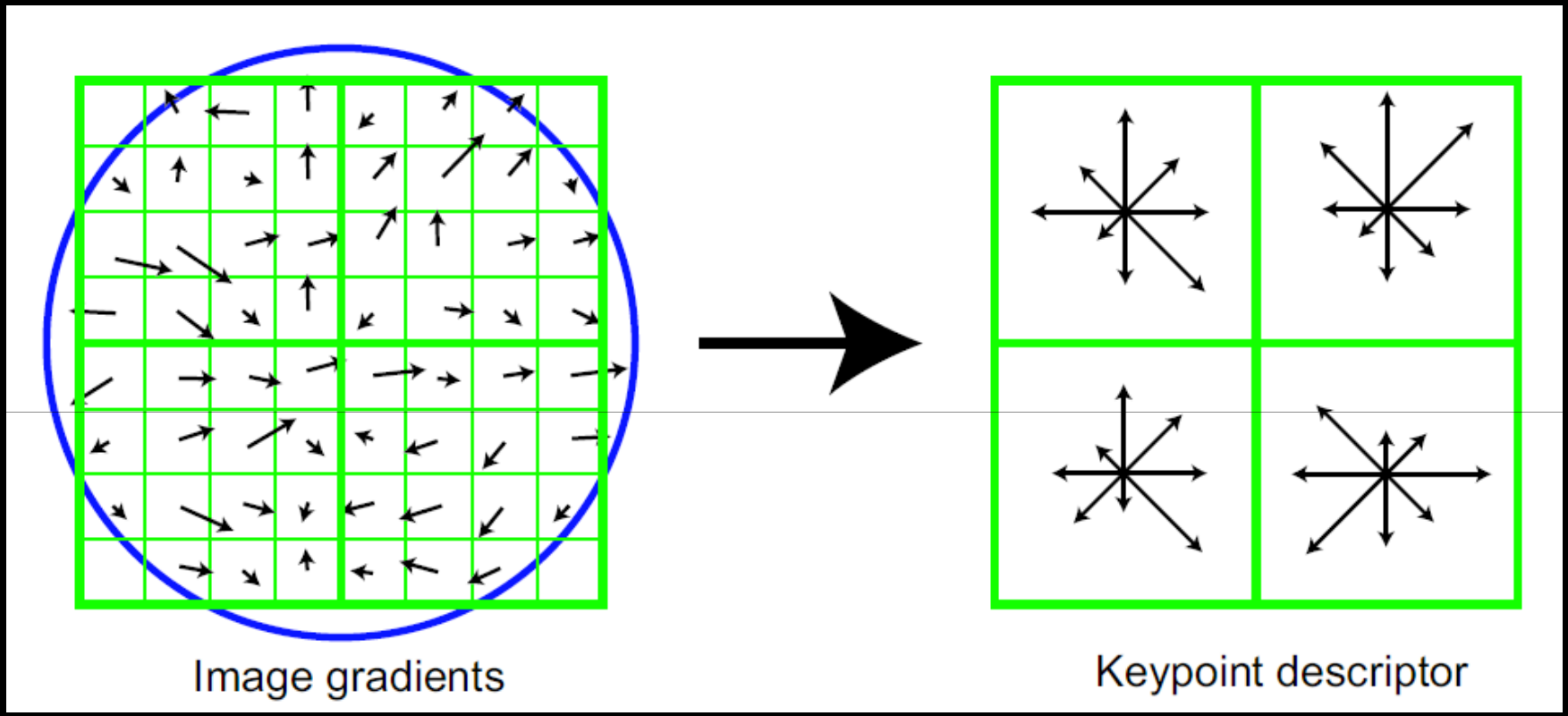
renormalizing to unit length.

$Sum = 1$

convert floating-point descriptor to integer valued descriptor .(Implement)

0.073788 ⟶ 12

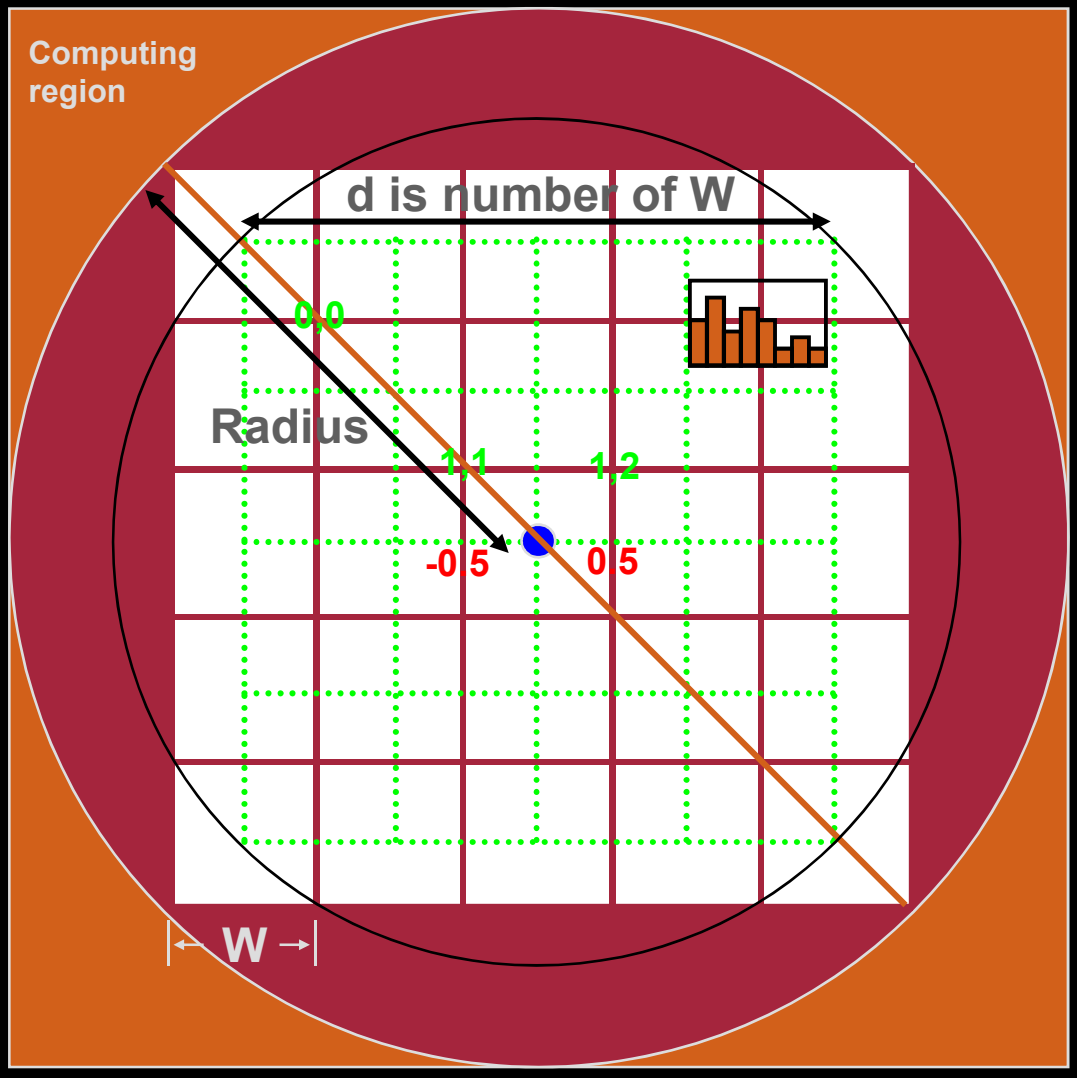
Descriptor Generation



- Weight magnitude of each sample point by Gaussian weighting function, σ=0.5*width
- Distribute each sample to adjacent bins by trilinear interpolation (avoids boundary effects)

Implement Descriptor

Descriptor=[4][4][8]



$$W = 3\sigma$$

$$W(d+1)\sqrt{2}$$

$$Radius = \frac{W(d+1)\sqrt{2}}{2} + 0.5$$

- **Keypoint**

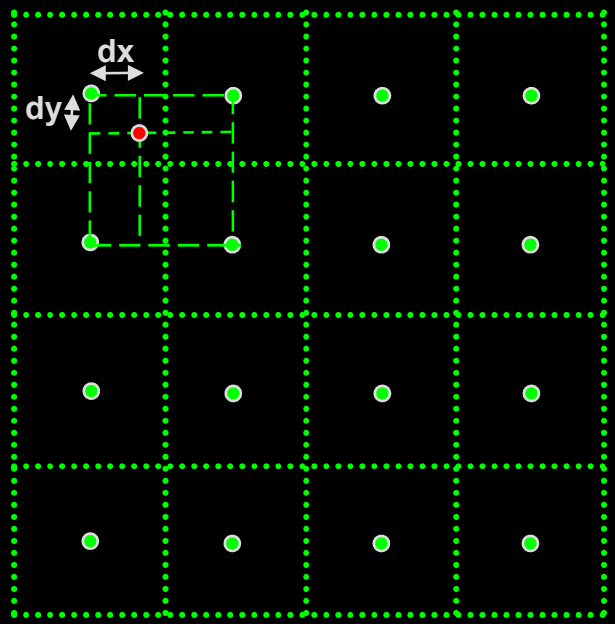
$$\theta = Orientation \quad Peak$$

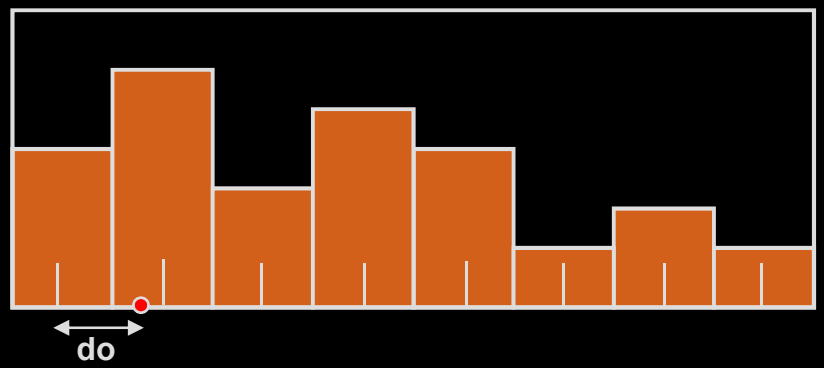
$$\begin{bmatrix} u & v \end{bmatrix} = \frac{1}{W}\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

$$c = u + \frac{d}{2} - 0.5 \qquad r = v + \frac{d}{2} - 0.5$$

$$weight = \frac{-(u^2 + v^2)}{2(d * 0.5)^2}$$

Trilinear interpolation





V_R = mag * (1-dy)
V_C = V_R * (1-dx)
V_O = V_C * (1-do)
[0][0][obin]+=V_O

V_R = mag * (1-dy)
V_C = V_R * (1-dx)
V_O = V_C * (do)
[0][0][obin+1]+=V_O

.
.
.

Normalization and Thresholding

❖ **Normalization**

$$Descriptor[i] = Descriptor[i] \times \frac{1}{\sqrt{\sum_k Descriptor[k]^2}}$$

➢ **Contrast Change will be canceled by vector normalization**

❖ **Thresholding**

$$if\,(Descriptor[i] > 0.2)$$

$$Descriptor[i] = 0.2$$

➢ **Reduce the influence of large gradient magnitudes**
➢ **Means that matching the magnitudes for large gradients is no longer as important, and that the distribution of orientations has greater emphasis.**

❖ **Renormalization**

Descriptor Testing

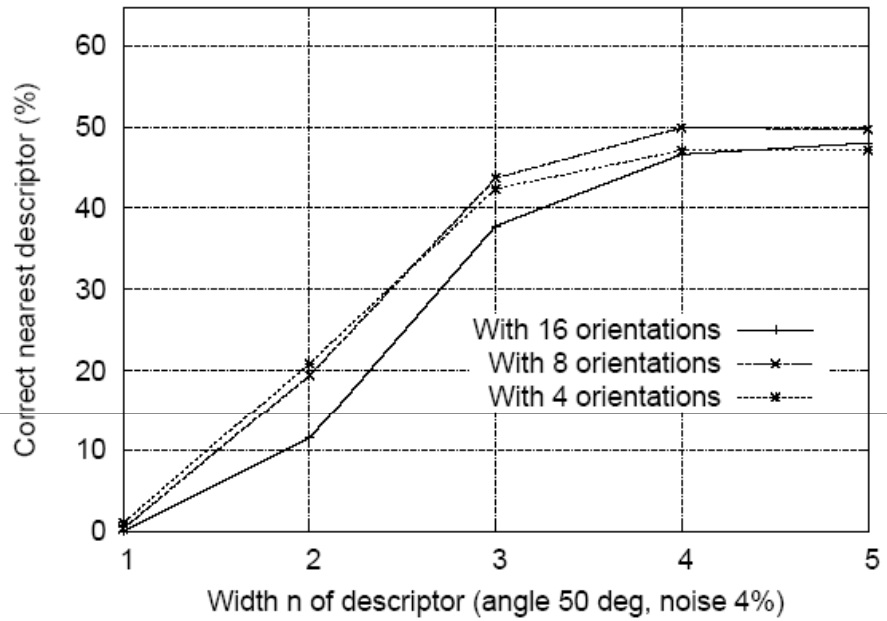


Figure 8: This graph shows the percent of keypoints giving the correct match to a database of 40,000 keypoints as a function of width of the $n \times n$ keypoint descriptor and the number of orientations in each histogram. The graph is computed for images with affi ne viewpoint change of 50 degrees and addition of 4% noise.

Viewpoint transformation : 50 degrees
Noise : 4%
Total Keypoints : 40,000

Sensitivity to affine change

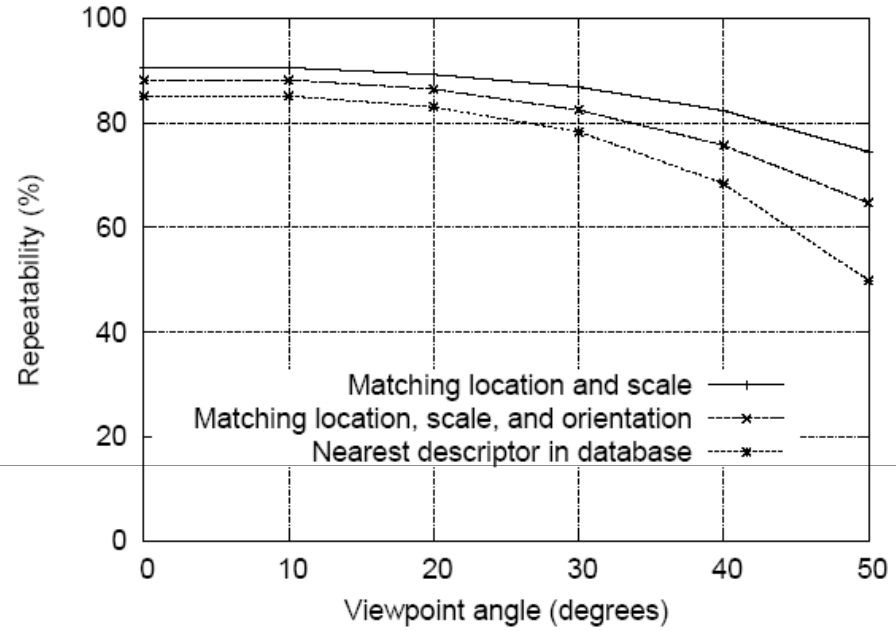


Figure 9: This graph shows the stability of detection for keypoint location, orientation, and final matching to a database as a function of affine distortion. The degree of affine distortion is expressed in terms of the equivalent viewpoint rotation in depth for a planar surface.

Viewpoint transformation : 50 degrees
Noise : 4%
Total Keypoints : 40,000

Matching to large database

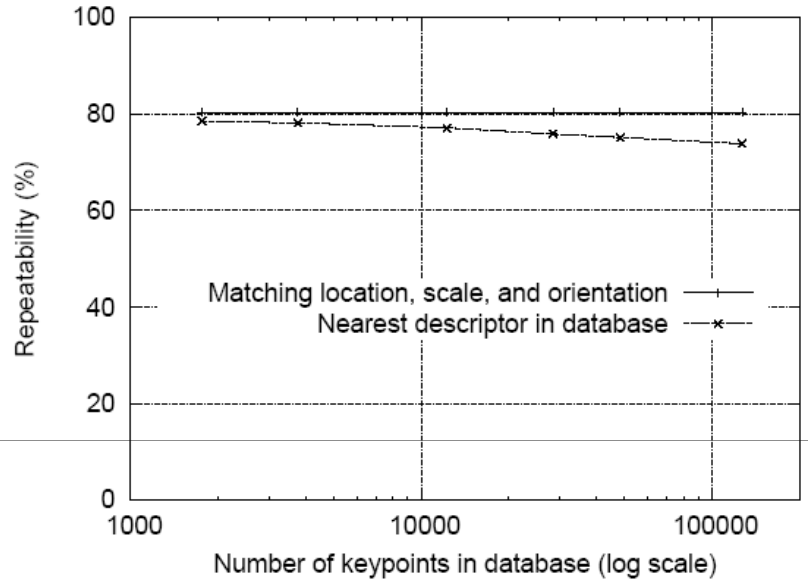


Figure 10: The dashed line shows the percent of keypoints correctly matched to a database as a function of database size (using a logarithmic scale). The solid line shows the percent of keypoints assigned the correct location, scale, and orientation. Images had random scale and rotation changes, an affi ne transform of 30 degrees, and image noise of 2% added prior to matching.

Viewpoint transformation : 50 degrees
Noise : 4%
Total Keypoints : 40,000

Keypoint matching

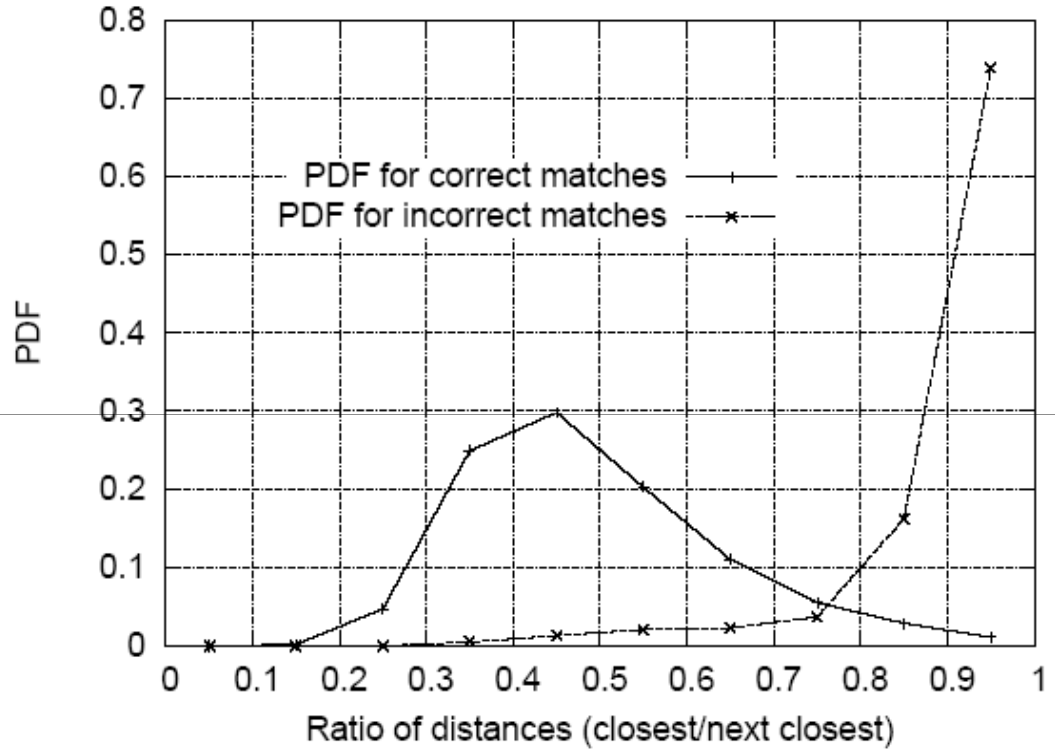


Figure 11: The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest. Using a database of 40,000 keypoints, the solid line shows the PDF of this ratio for correct matches, while the dotted line is for matches that were incorrect.

Solution for affine parameters

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \dots & \dots & & \\ & & \dots & \dots & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \dots \end{bmatrix}$$

Solution for affine parameters

Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affine transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

Recognition Example

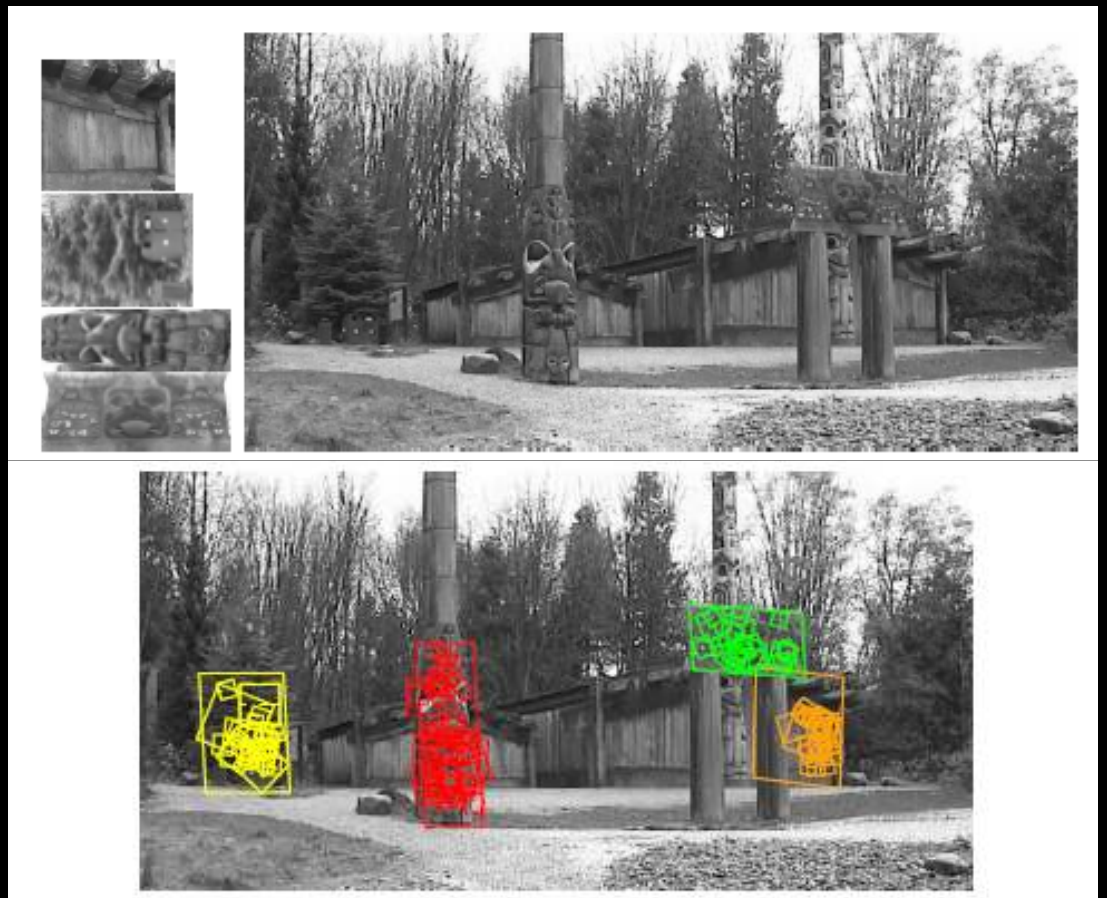
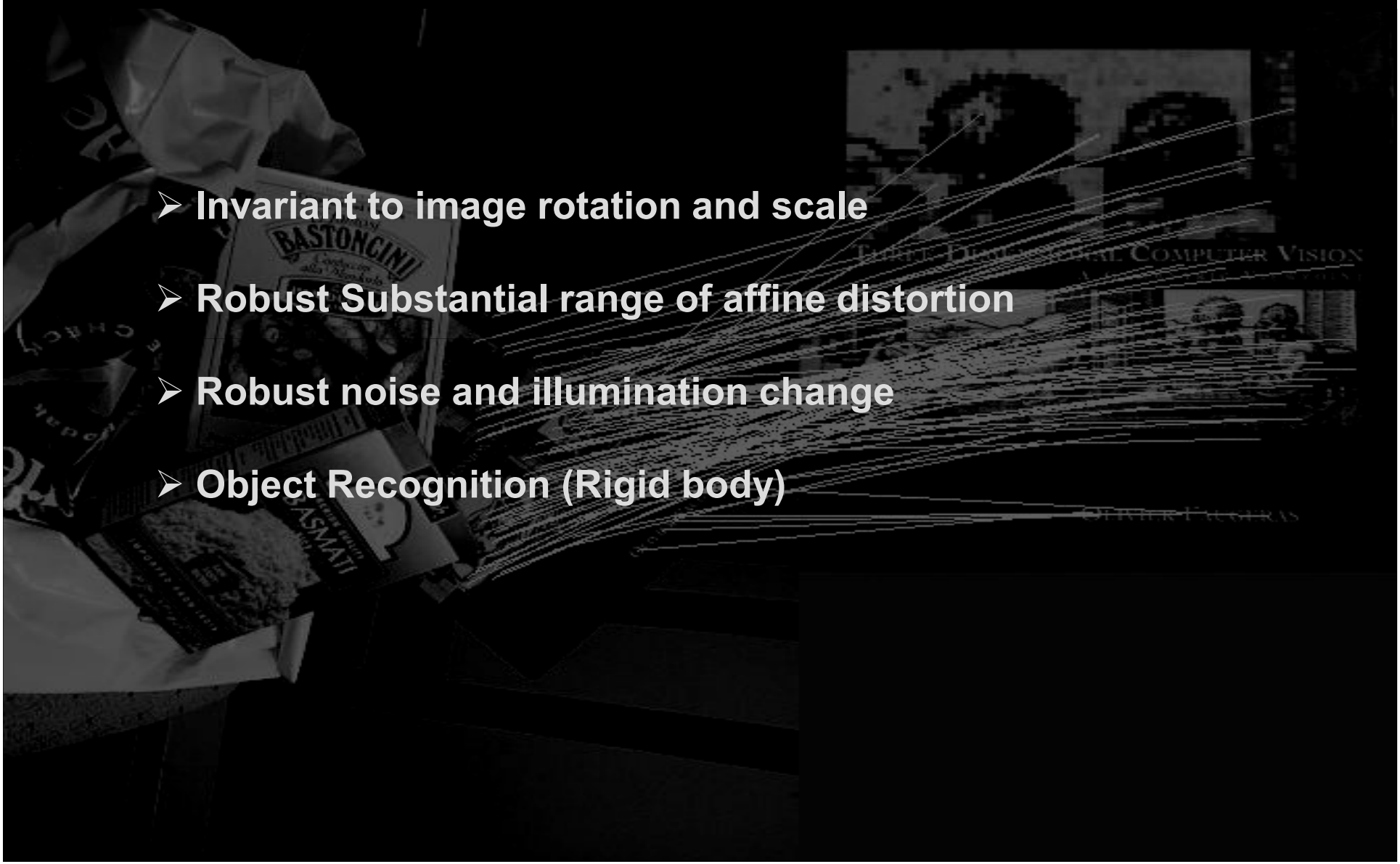


Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affine transform used for recognition.

Conclusions



- Invariant to image rotation and scale
- Robust Substantial range of affine distortion
- Robust noise and illumination change
- Object Recognition (Rigid body)