SIFT

Presented by Xu Miao

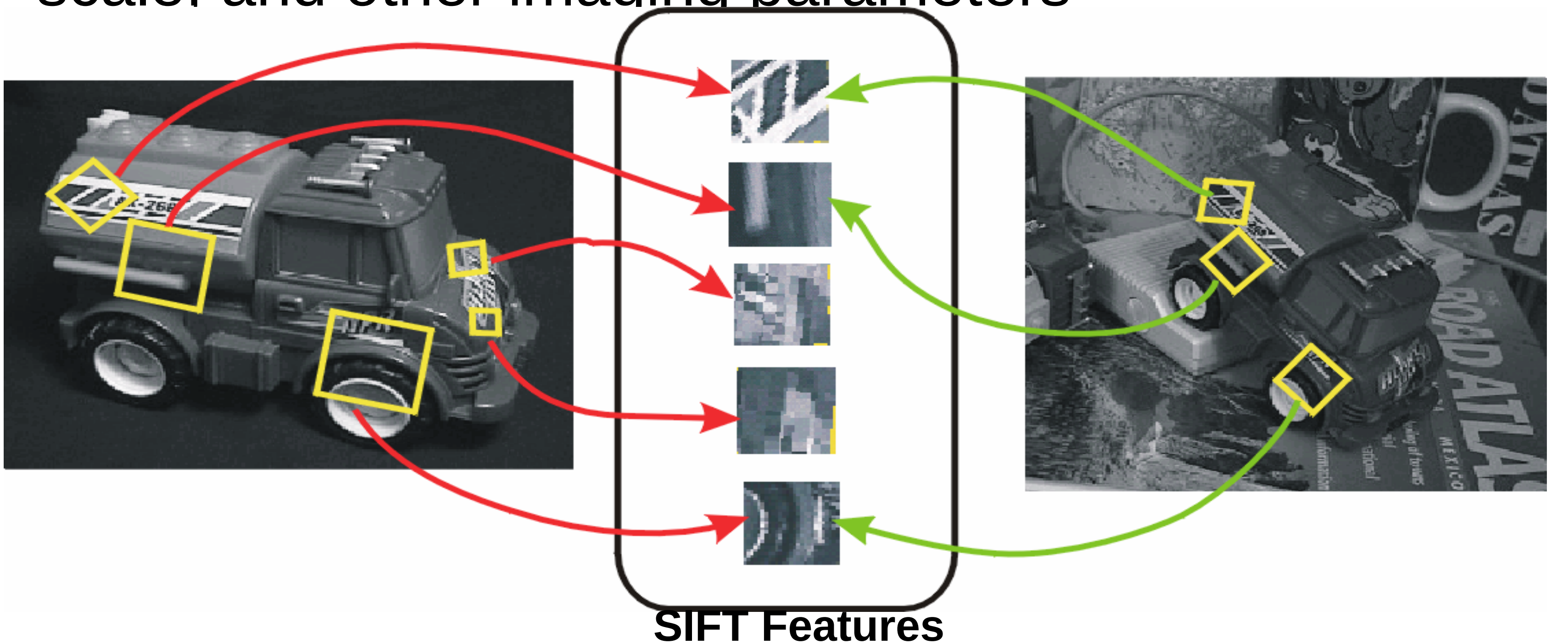
April 20, 2005

Outline

- Motivation of SIFT
- Scale-space extrema detection
  - Scale-space function
  - Local extrema detection
  - Detection sampling
- Keypoint localization
- Orientation assignment
- Keypoint descriptor
- Comparison of Harris-Laplacian and SIFT
- Image matching

Motivation of SIFT (copy from 576 slides)

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



Motivation of SIFT
--Advantages of local features (copy)

- **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

More motivation… (copy)

- Feature points are used also for:
  - Image alignment (homography, fundamental matrix)
  - 3D reconstruction
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation
  - … other

Scale-space extrema detection

- Scale-space function
  - The only reasonable one:

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

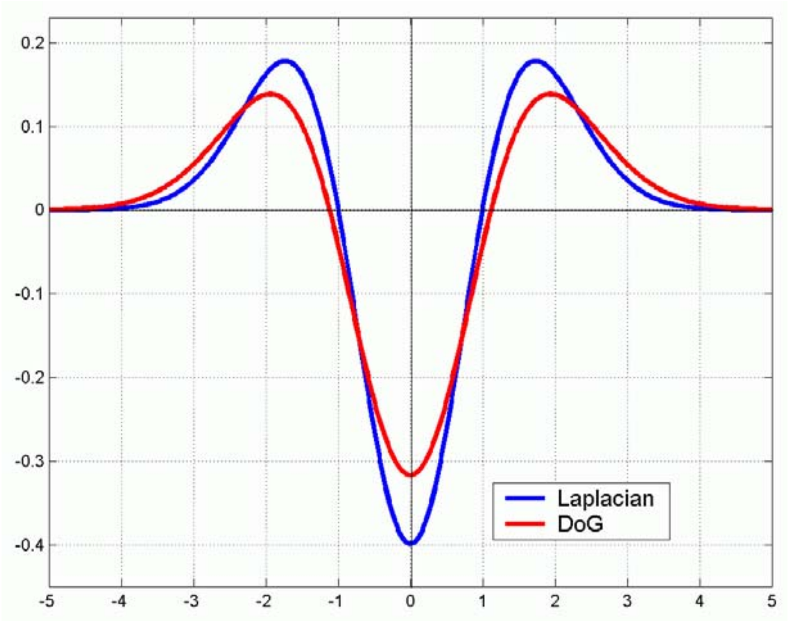
  - Laplacian of Gaussian kernel is a good choice of scale invariance

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

  - Difference of Gaussian kernel is a close approximate to scale

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

Scale-space extrema detection



- Gaussian is an ad hoc solution of heat diffusion equation

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

- Hence

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

- k is not necessarily very small in practice

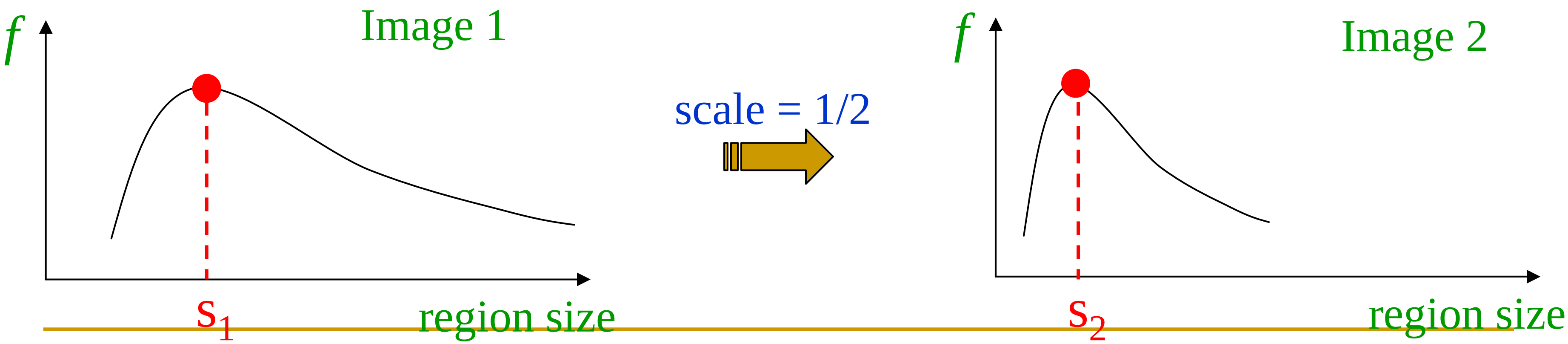
Scale Invariant Detection (Copy)

- Common approach:

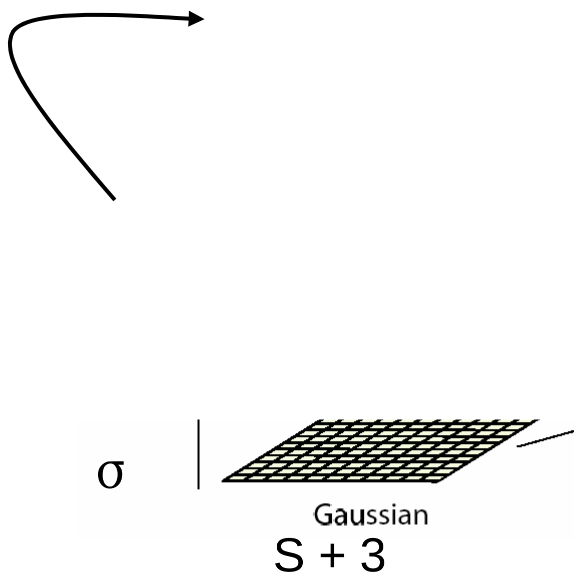
  Take a local maximum of this function (convolution of kernel and image)

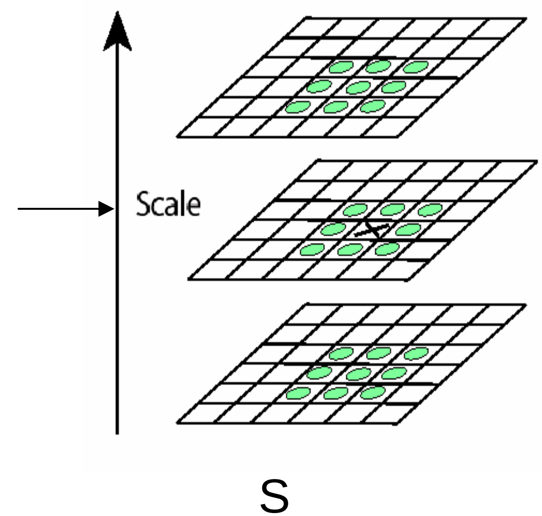
  Observation: region size, for which the maximum is achieved, should be *invariant* to image scale.

Important: this scale invariant region size is found in each image independently!

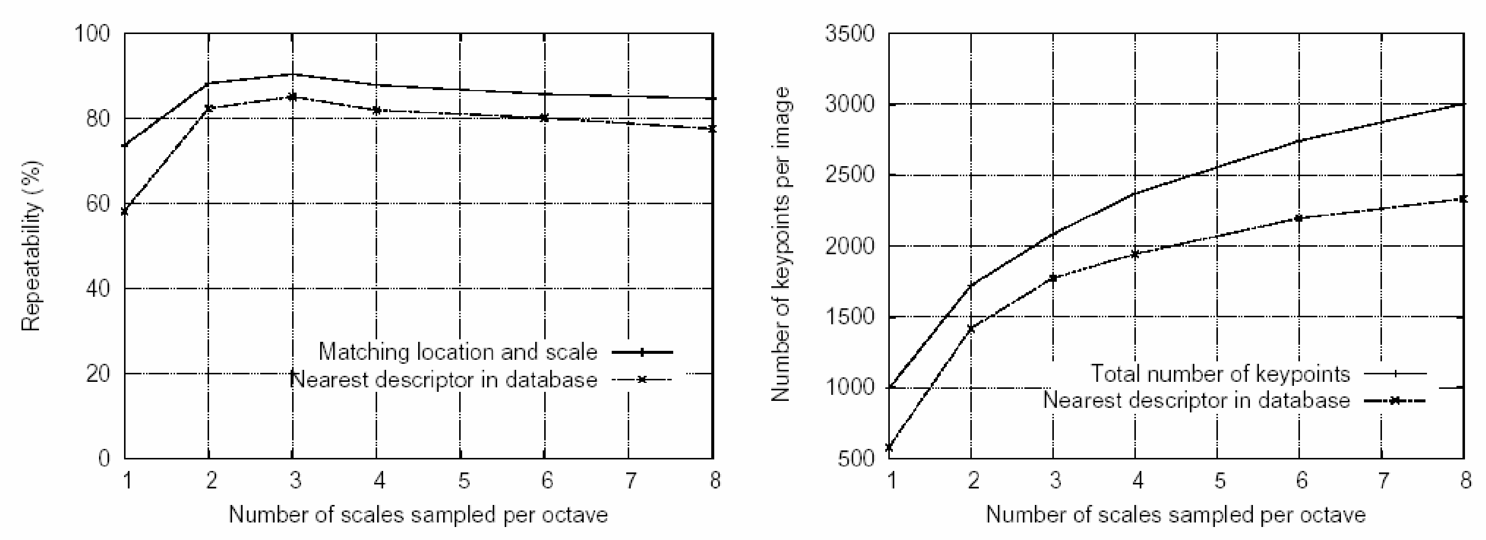


Scale-space extrema detection



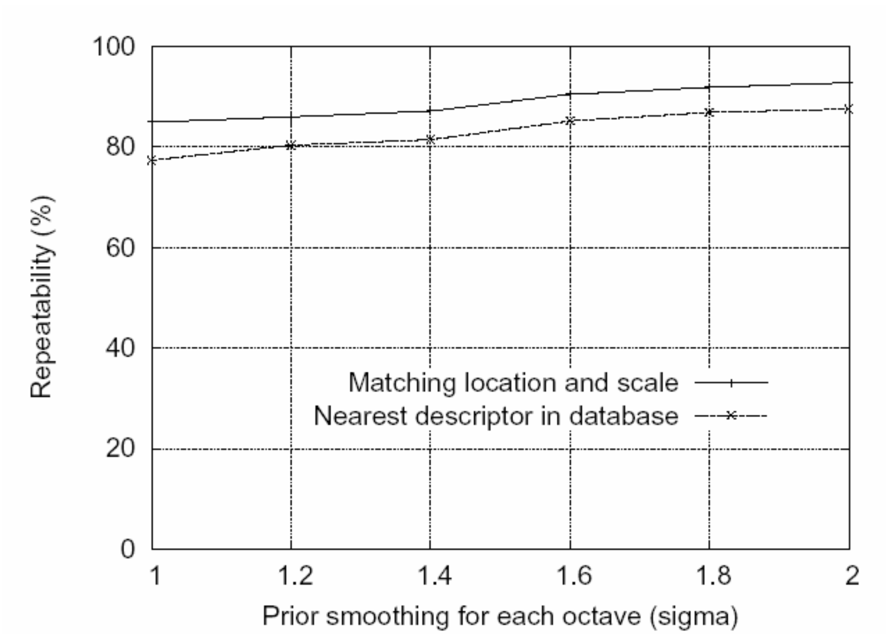


Scale-space extrema detection



- 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

Scale-space extrema detection



- Pre-smooth before extrema detection is equivalent to spatial sampling
  - Sigma is higher, # of stable keypoints is larger
  - Lower sampling frequency preferred?
  - To utilize the information smoothed off, first double the original image, which also increases the stable kepoints found by 4

Keypoint localization

- Detailed keypoint determination
  - Sub-pixel and sub-scale location scale determination
  - Ratio of principal curvature to reject edges and flats (detect corners?)

Keypoint localization

- Sub-pixel and sub-scale location scale determination

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathbf{T}} \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}$$

$$\mathbf{x} = (x, y, \sigma)^T$$

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

Keypoint localization

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

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

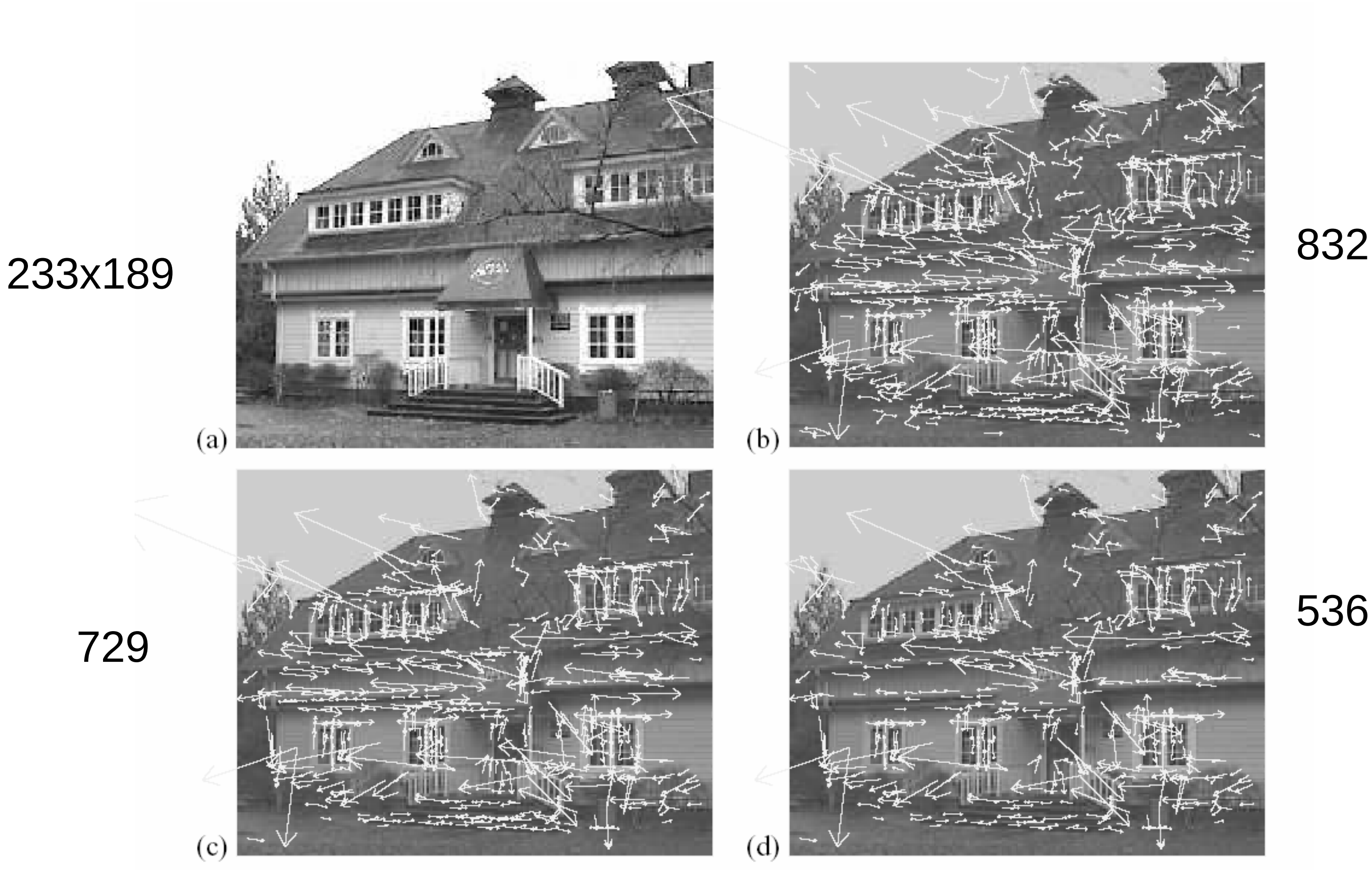
$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

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

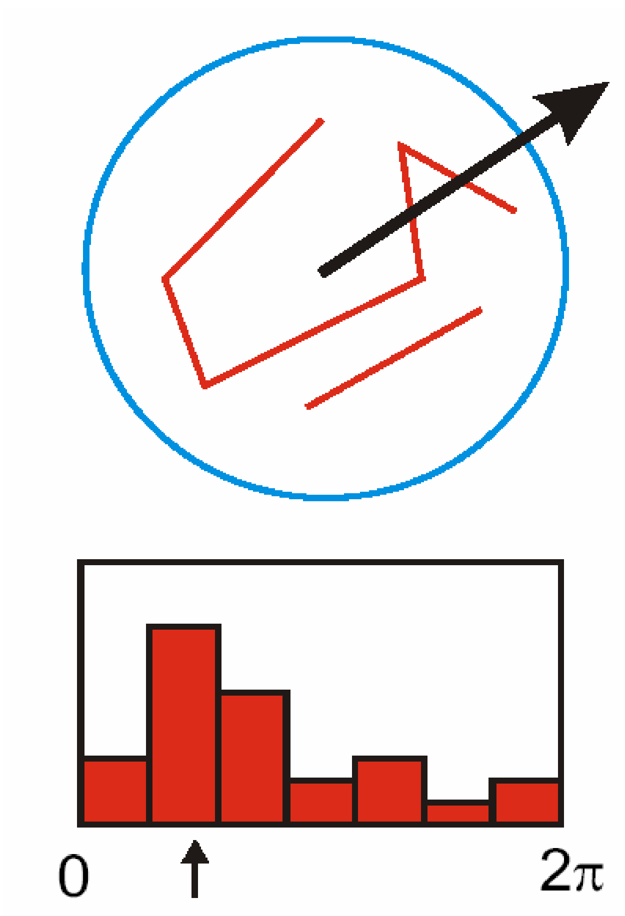
$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r},$$

  - r < 10
- Is it Harris corner detector?

Keypoint localization



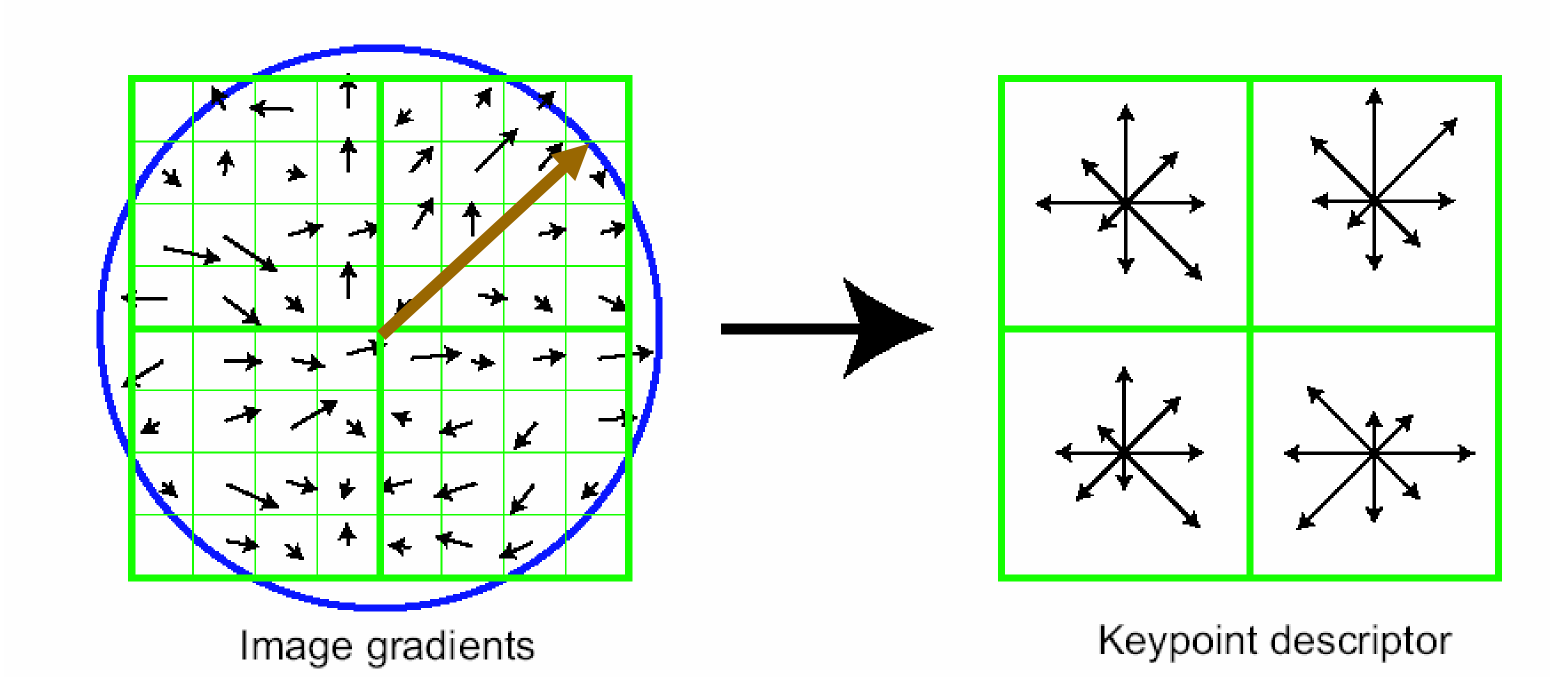
Orientation assignment



- Create histogram of local gradient directions at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale,orientation)

Keypoint descriptor

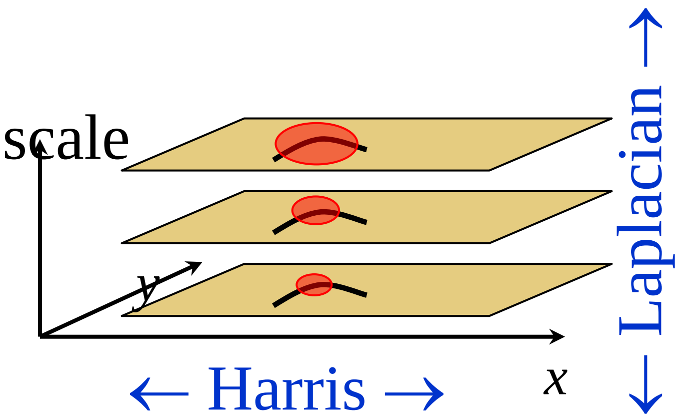
- Invariant to other changes (Complex Cell)



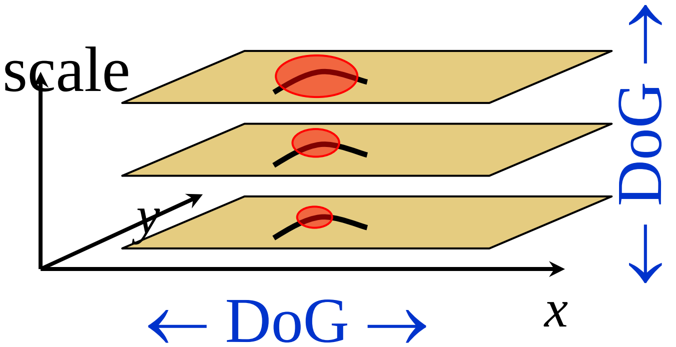
In experiment, 4x4 arrays of 8 bin histogram is used, in total of 128 features for one keypoint

Scale Invariant Detectors (copy)

- Harris-Laplacian[1]
  Find local maximum of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale



- SIFT (Lowe)[2]
  Find local maximum of:
  - Difference of Gaussians in space and scale



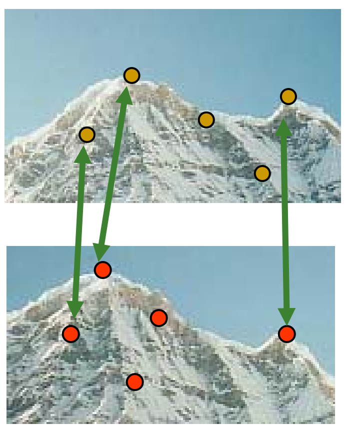
[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". Accepted to IJCV 200

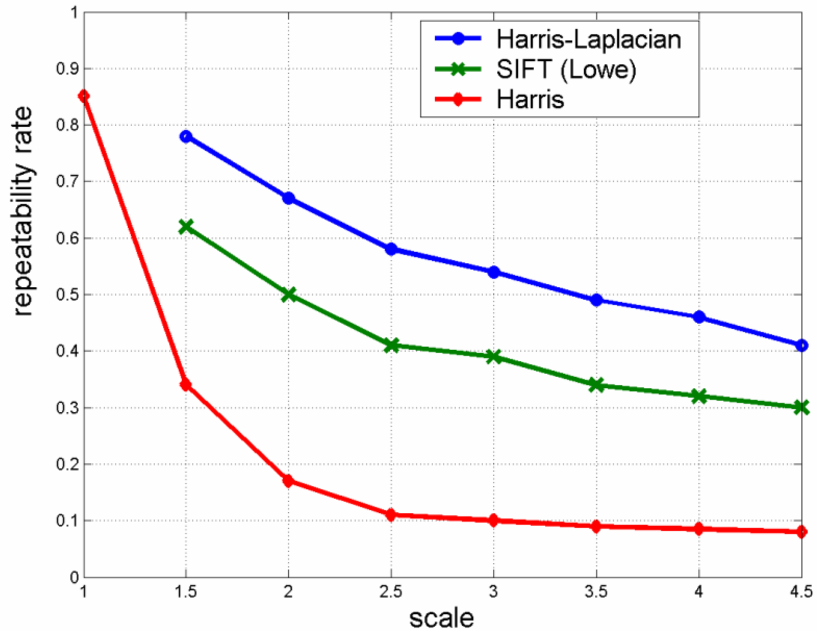
Scale Invariant Detectors(copy)

- Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:

$$\frac{\#\text{ correspondences}}{\#\text{ possible correspondences}}$$





K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

Comparison with Harrison-Laplacian

- Affine-invariant comparison
  - Translation-invariant – local features
  - Rotation-invariant
    - Harrison-Laplacian
      - PCA
    - SIFT
      - Orientation
  - Shear-invariant
    - Harrison-Laplacian
      - Eigen values
    - SIFT
      - No
- Within 50 degree of viewpoint, SIFT is better than HL, after 70 degree, HL is better.

Comparison with Harrison-Laplacian

- Computational time:
  - SIFT is few floating point calculation
  - HL uses iterative calculation which costs much more

Object recognition by SIFT keypoint matching

- Efficient nearest neighbor algorithm
  - Best-Bin-First (modification of k-d tree)
- Hough transformation to cluster features into 3-feature groups
- Solving affine parameters by pseudo-inversion to verify the matching model
- Final decision is made by Bayesian approach

$$p = dlrs$$

$$P(f|\neg m) = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j}$$

$$P(m|f) \approx \frac{P(m)}{P(m) + P(f|\neg m)}$$