High Level Computer Vision - May 3, 2o17

Object Identification
Interest Point Detection & Description

Bernt Schiele - schiele@mpi-inf.mpg.de
Mario Fritz - mfritz@mpi-inf.mpg.de

Overview Today

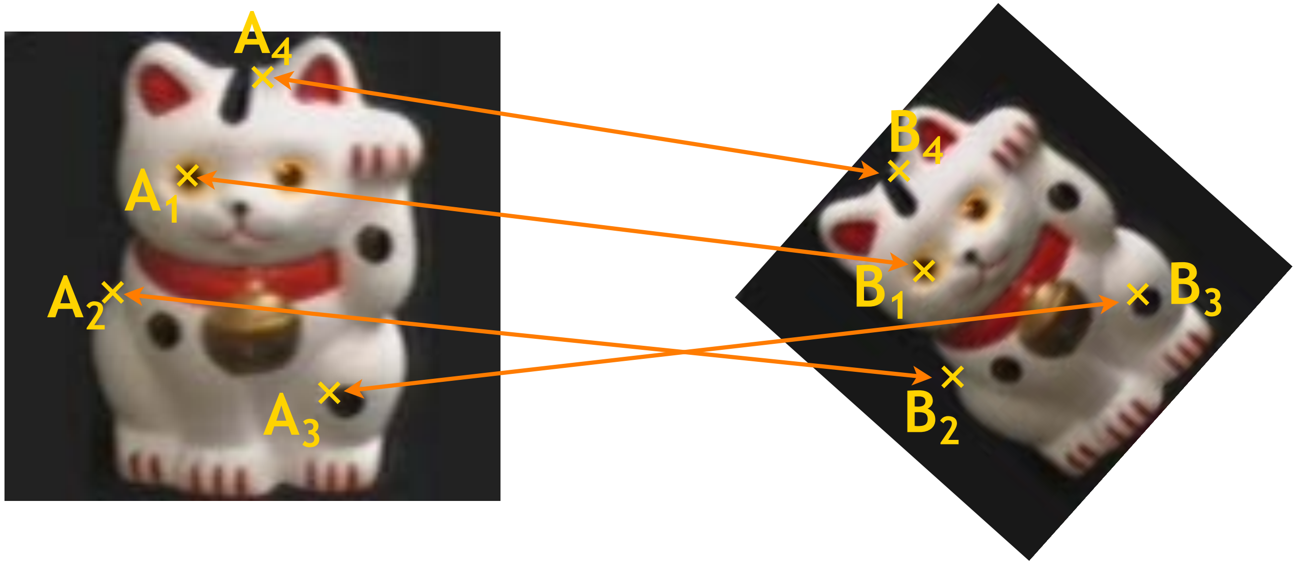
- Object Identification by Point Correspondences
  - general procedure for recognition, stereo, image stitching, ...
- Interest Point Detection & Descriptor
  - local interest point detection
  - scale-invariant interest point detection
  - local image descriptor
- Scaling to Large Numbers of Images and Objects
  - inverted file
  - visual vocabulary

Object Recognition (reminder)

- Different Types of Recognition Problems:
  - Object **Identification**
    - recognize your apple, your cup, your dog
    - sometimes called: “instance recognition”
  - Object **Classification**
    - recognize any apple, any cup, any dog
    - also called: generic object recognition, object categorization, …
    - typical definition: ‘basic level category’

Recognition by "Correspondence" (=Matching)

- General Idea of using Interest Point Detection:
  - Recognition by finding Correspondence between Interest Points



Local Interest Point Detection

- Applications of Local Interest Point Detection
  - recognition by correspondence
  - point correspondence for (sparse) stereo matching
  - (sparse) optical flow - point correspondence
  - ...

- Multiple Goals (somewhat contradicting)
  - **Discriminance**: find points that are discriminant enough to find corresponding points in other images
  - **Invariance to Transformations**: find same set of interest-points regardless of geometric and photometric transformations
    - geometric transformations: translation, scale, rotation, affine, projective
    - photometric transformations: light changes (intensity, color, direction)

Geometric Transformations

- Example of different geometric transformations:
  - (1) original
  - (2) similarity transformation (translation, image plane rotation, scaling)
  - (3) projective transformation

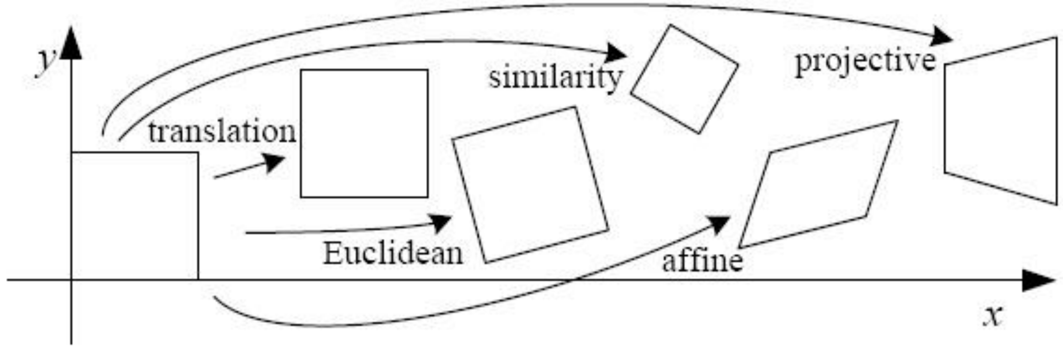
(1)

(2)

(3)

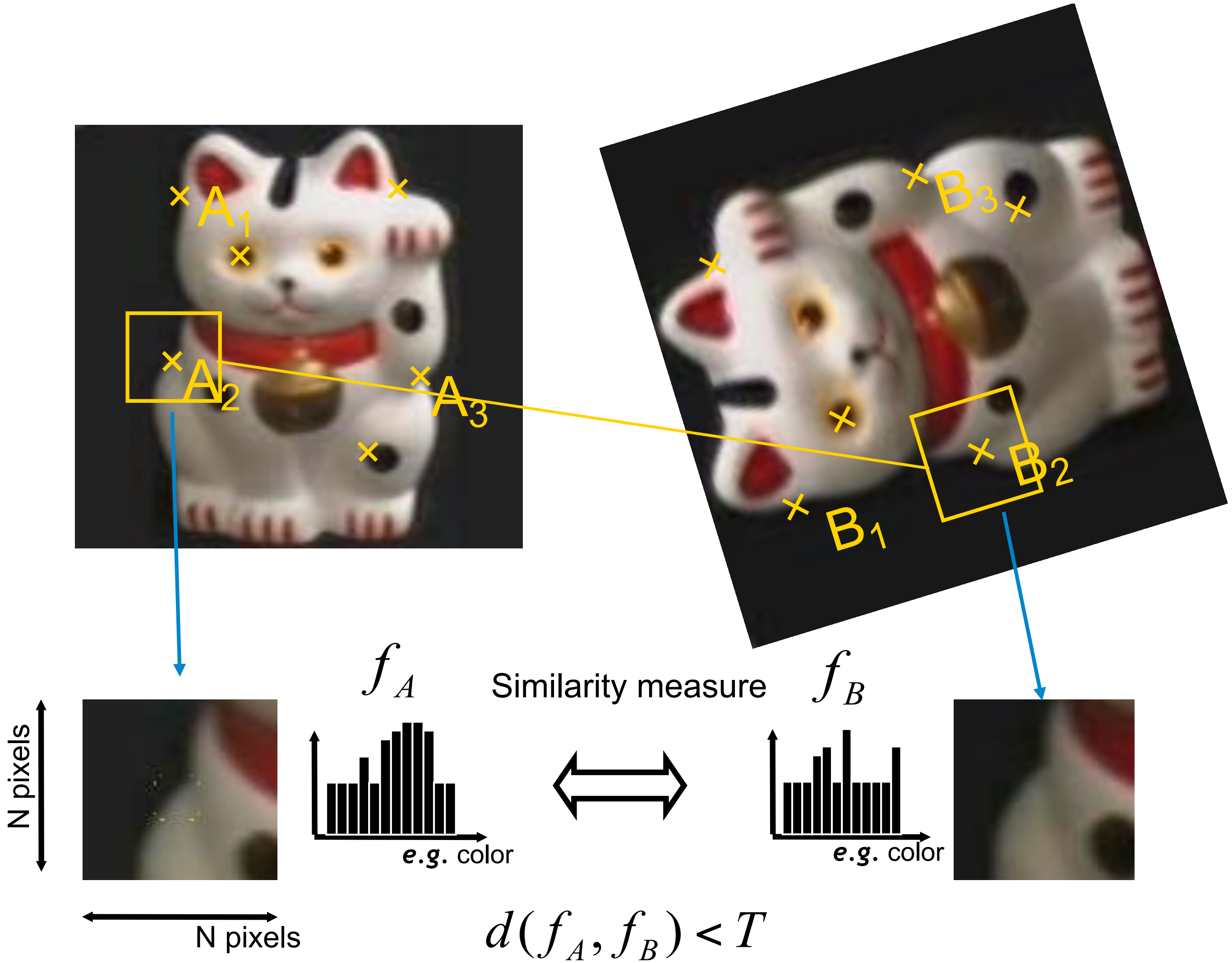
Planar image transformations

- Transformation of planar scenes
  - Fully defined by a 3x3 matrix (in homogeneous coordinates)



Invariants / Transformation	length	angle	length - ratio	parallelism	straight lines
Euclidean (rotation, translation = 3 DoF)	yes	yes	yes	yes	yes
Similarity (rotation, translation, scale = 4 DoF)	no	yes	yes	yes	yes
Affine (similarity+non-uniform scale,sheer =6DoF)	no	no	no	yes	yes
Projective (8 DoF)	no	no	no	no	yes

Point Correspondence for Object Instance Recognition: General Approach

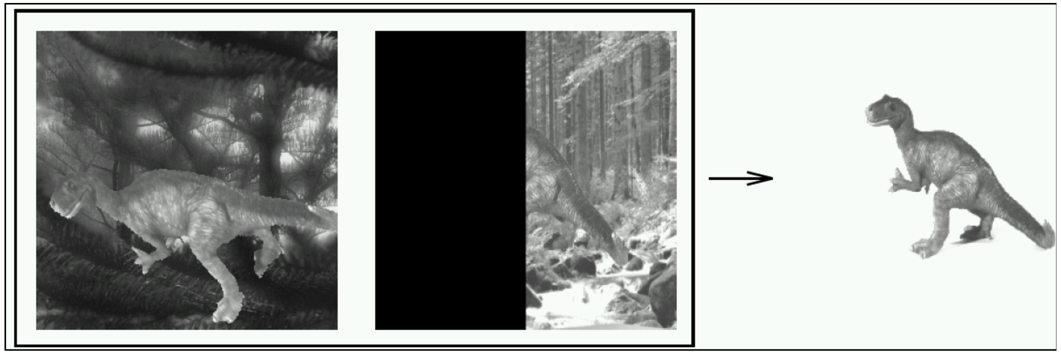


$$d(f_A, f_B) < T$$

1. Interest Point **Detection**: Find a set of distinctive key-points
2. Extract and **normalize** the region content
3. Compute local **descriptor** from the normalized region
4. **Match** local descriptors

(5. Estimate global transformation)

Recognition of Specific Objects, Scenes



Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

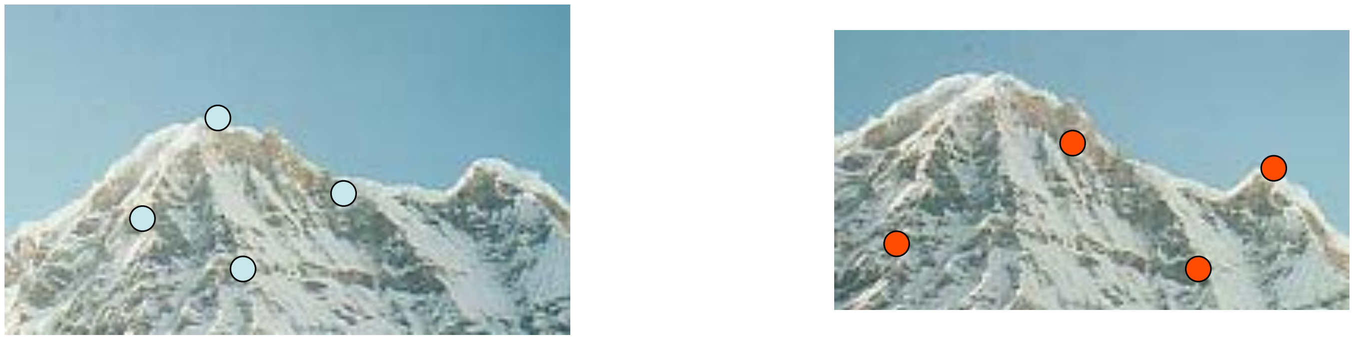
Lowe 2002

General Procedure for Point Correspondence

- 5-Step Procedure
  1. Interest Point Detection
  2. Extract and Normalize Region around Interest Point
  3. Compute Local Descriptor
  4. Match Local Descriptor
  5. Estimate Global Transformation
     - to align images (e.g. image stitching)
     - to verify point correspondence globally (e.g. object recognition)

1. Interest Point Detection
Common Requirements

- Problem 1:
  - Detect the same point independently in both images



No chance to match!

We need a repeatable detector!

Slide credit: Darya Frolova, Denis Simakov

1. Interest Point Detection
Many Existing Detectors Available

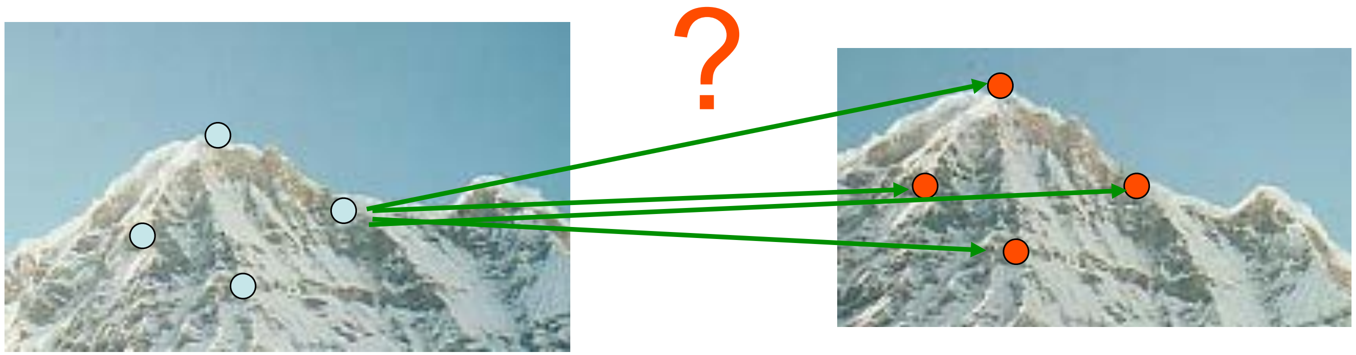
- Hessian & Harris [Beaudet '78], [Harris '88]
- Laplacian, DoG [Lindeberg '98], [Lowe '99]
- Harris-/Hessian-Laplace [Mikolajczyk & Schmid '01]
- Harris-/Hessian-Affine [Mikolajczyk & Schmid '04]
- EBR and IBR [Tuytelaars & Van Gool '04]
- MSER [Matas '02]
- Salient Regions [Kadir & Brady '01]
- Others…

- Those detectors have become a basic building block for many recent applications in Computer Vision.

1. Interest Point Detection
Common Requirements

- Problem 1:
  - Detect the same point independently in both images

- Problem 2:
  - For each point correctly recognize the corresponding one



We need a **reliable** detector to find **distinctive** points/regions!

Slide credit: Darya Frolova, Denis Simakov

1. Interest Point Detection Requirements

- Region extraction needs to be repeatable and accurate
  - Invariant to translation, rotation, scale changes
  - Robust or covariant to out-of-plane (≈affine) transformations
  - Robust to lighting variations, noise, blur, quantization
- Locality: Features are local, therefore robust to occlusion and clutter.
- Quantity: We need a sufficient number of regions to cover the object.
- Distinctiveness: The regions should contain “interesting” structure.
- Efficiency: Close to real-time performance.

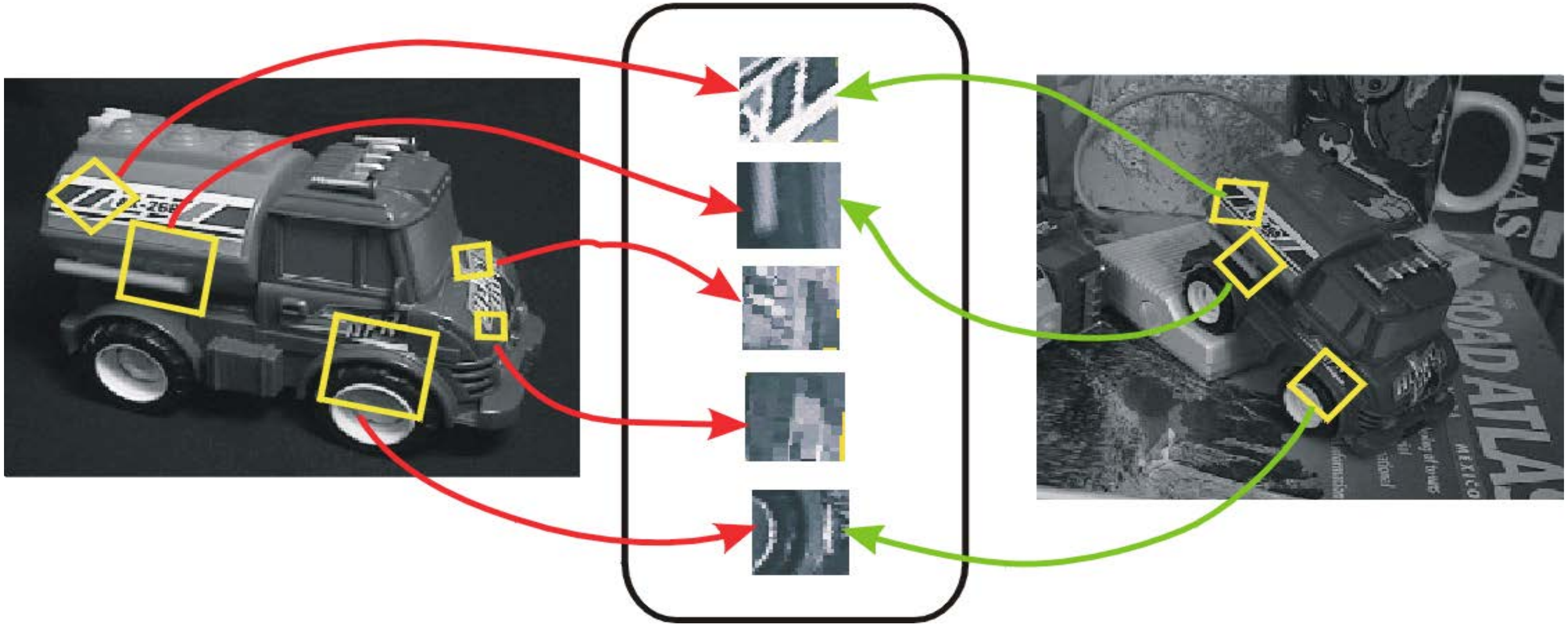
General Procedure for Point Correspondence

- 5-Step Procedure
  1. Interest Point Detection
  2. Extract and Normalize Region around Interest Point
  3. Compute Local Descriptor
  4. Match Local Descriptor
  5. Estimate Global Transformation
     - to align images (e.g. image stitching)
     - to verify point correspondence globally (e.g. object recognition)

Invariance vs. Covariance - or Two Ways to Obtain Invariance

Slide credit: Svetlana Lazebnik, David Lowe

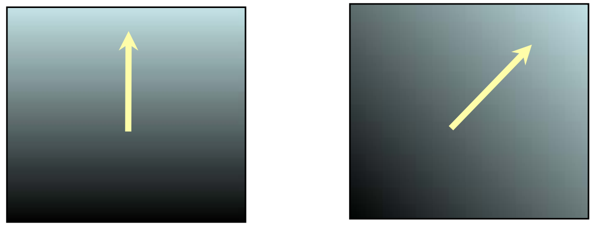
- Invariance:
  - features(transform(image)) = features(image)
- Covariance:
  - features(transform(image)) = transform(features(image))

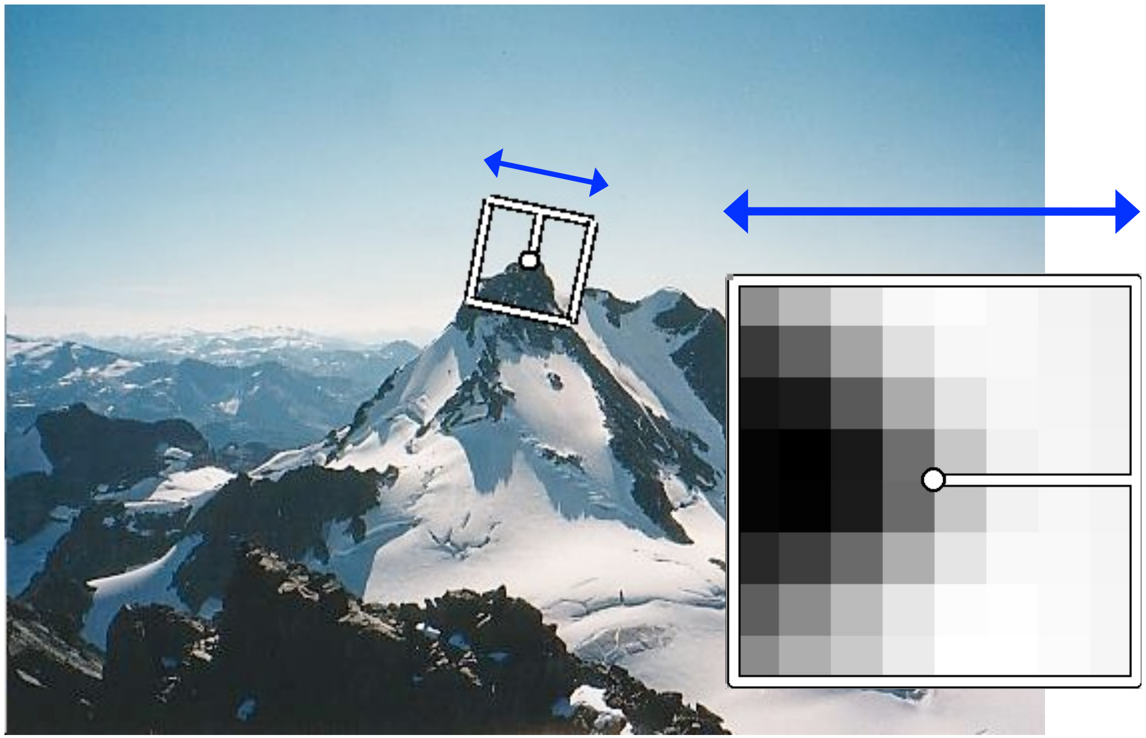


Covariant detection ⇒ invariant description

2. Extract and Normalize Region around Interest Point
Rotation Invariant Descriptors

- Find local orientation
  - Dominant direction of gradient for the image patch
- Rotate patch according to this angle
  - This puts the patches into a canonical orientation.



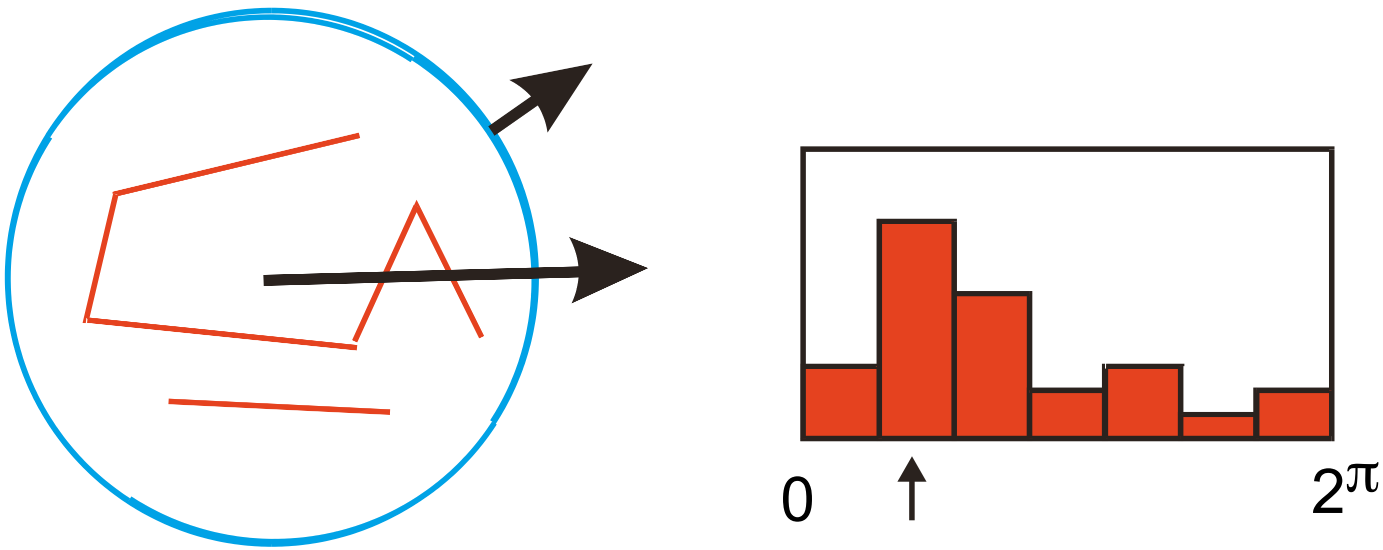


Slide credit: Svetlana Lazebnik, Matthew Brown

2. Extract and Normalize Region around Interest Point
Orientation Normalization: Computation

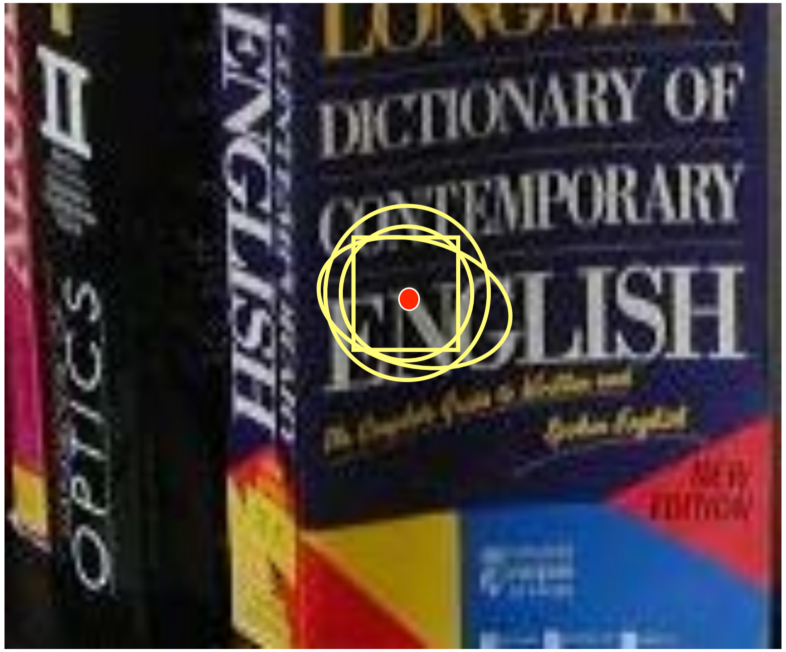
[Lowe, SIFT, 1999]

- Compute orientation histogram
- Select dominant orientation
- Normalize: rotate to fixed orientation



Slide adapted from David Lowe

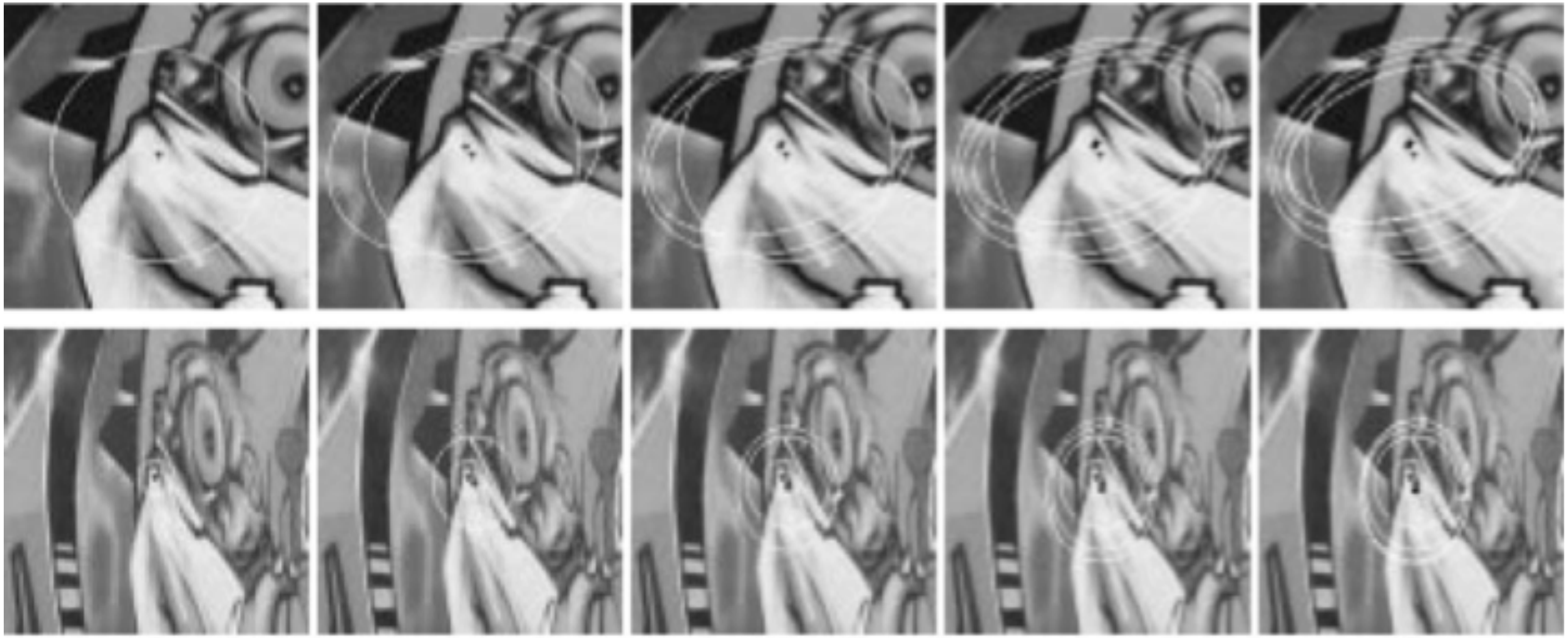
2. Extract and Normalize Region around Interest Point
The Need for Invariance



- Up to now, we had invariance to
  - Translation, Scale, Rotation
- Not sufficient to match regions under viewpoint changes
  - For this, we need also affine adaptation

Slide credit: Tinne Tuytelaars

2. Extract and Normalize Region around Interest Point Iterative Affine Adaptation



1. Detect keypoints, e.g. multi-scale Harris
2. Automatically select the scales
3. Adapt affine shape based on second order moment matrix
4. Refine point location

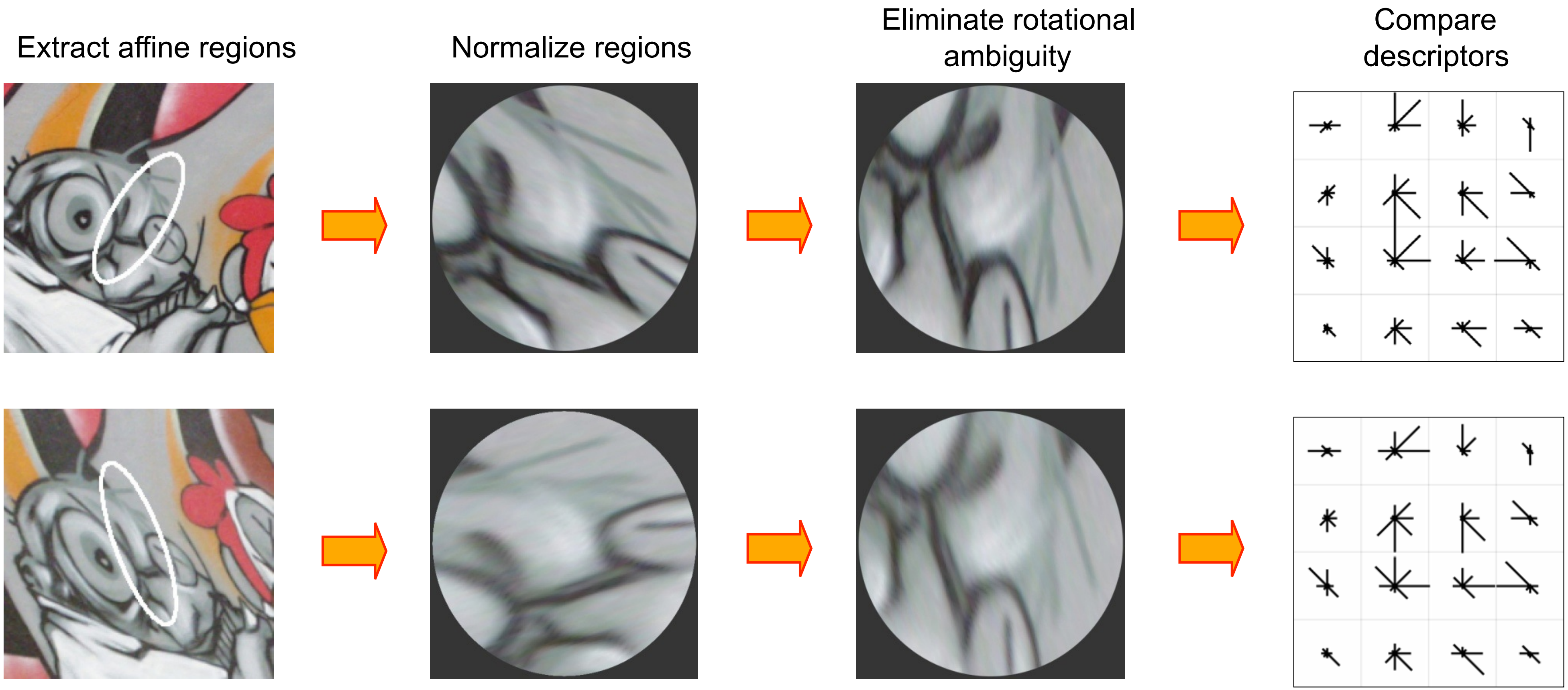
K. Mikolajczyk and C. Schmid, Scale and affine invariant interest point detectors, IJCV 60(1):63-86, 2004.

2. Extract and Normalize Region around Interest Point Affine Normalization/Deskewing

- Steps
  - Rotate the ellipse’s main axis to horizontal
  - Scale the x axis, such that it forms a circle

Slide credit: Tinne Tuytelaars

2. Extract and Normalize Region around Interest Point
Summary: Affine-Inv. Feature Extraction



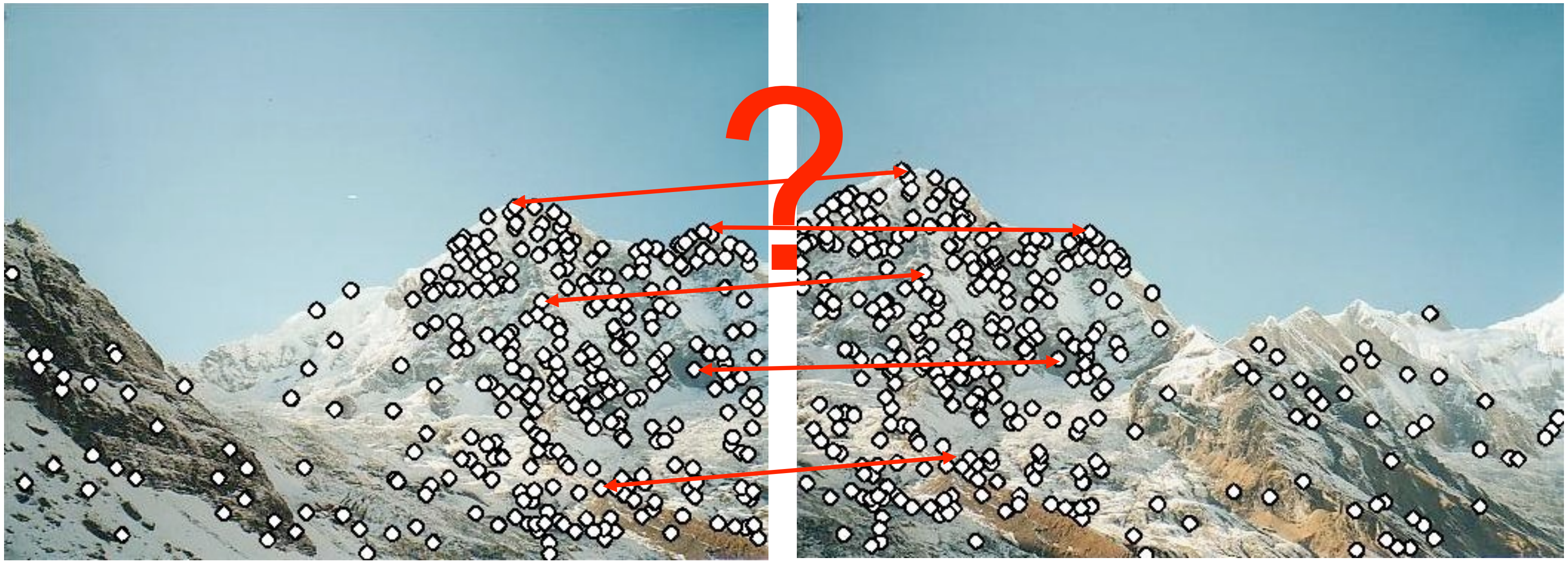
Slide credit: Svetlana Lazebnik

General Procedure for Point Correspondence

- 5-Step Procedure
  1. Interest Point Detection
  2. Extract and Normalize Region around Interest Point
  3. Compute Local Descriptor
  4. Match Local Descriptor
  5. Estimate Global Transformation
     - to align images (e.g. image stitching)
     - to verify point correspondence globally (e.g. object recognition)

Local Descriptors

- Let's assume we know how to detect points
- Next question: How to describe them for matching?

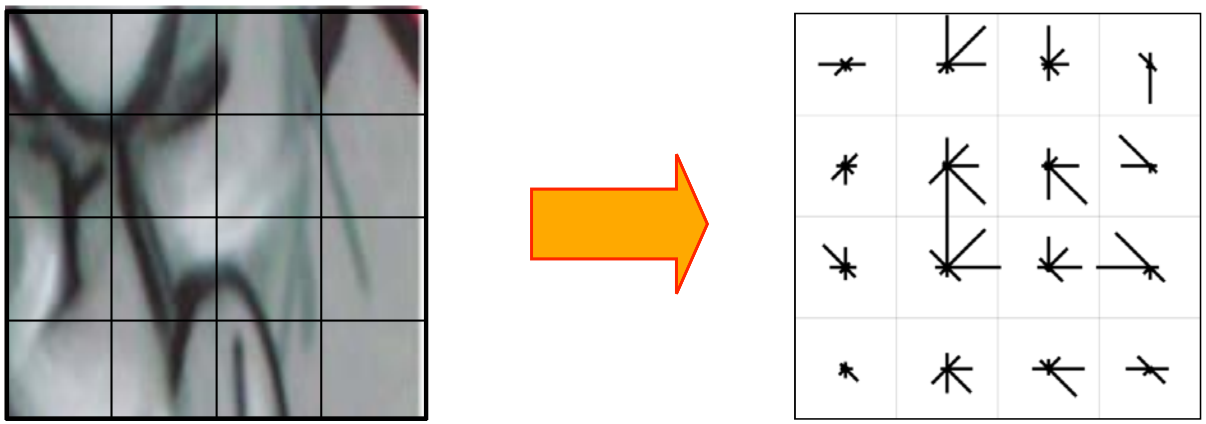


Point descriptor should be:
1. Invariant
2. Distinctive

Slide credit: Kristen Grauman

Feature Descriptors: SIFT

- Scale Invariant Feature Transform
- Descriptor computation:
  - Divide patch into 4x4 sub-patches: 16 cells
  - Compute histogram of gradient orientations (8 reference angles) for all pixels inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions



David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

Slide credit: Svetlana Lazebnik

General Procedure for Point Correspondence

- 5-Step Procedure
  1. Interest Point Detection
  2. Extract and Normalize Region around Interest Point
  3. Compute Local Descriptor
  4. Match Local Descriptor
  5. Estimate Global Transformation
     - to align images (e.g. image stitching)
     - to verify point correspondence globally (e.g. object recognition)

Feature Matching

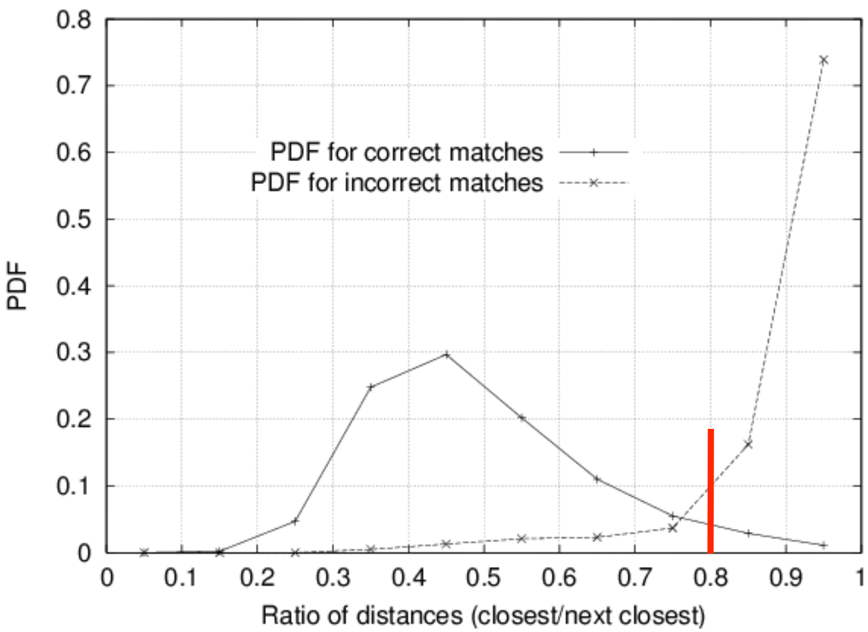
- Generating putative matches:
  - For each patch in one image, find a short list of patches in the other image that could match it based solely on appearance.

- Options
  - Exhaustive search
    - For each feature in one image, compute the distance to all features in the other image and find the "closest" ones (threshold or fixed number of top matches).
  - Fast approximate nearest neighbor search
    - Hierarchical spatial data structures (kd-trees, vocabulary trees)
    - Hashing

Slide credit: Svetlana Lazebnik

Feature Space Outlier Rejection

- How can we tell which putative matches are reliable?
- Heuristic: compare distance of nearest neighbor to that of second nearest neighbor (of another object)
  - Ratio will be high for features that are not distinctive
  - Threshold of 0.8 provides good separation



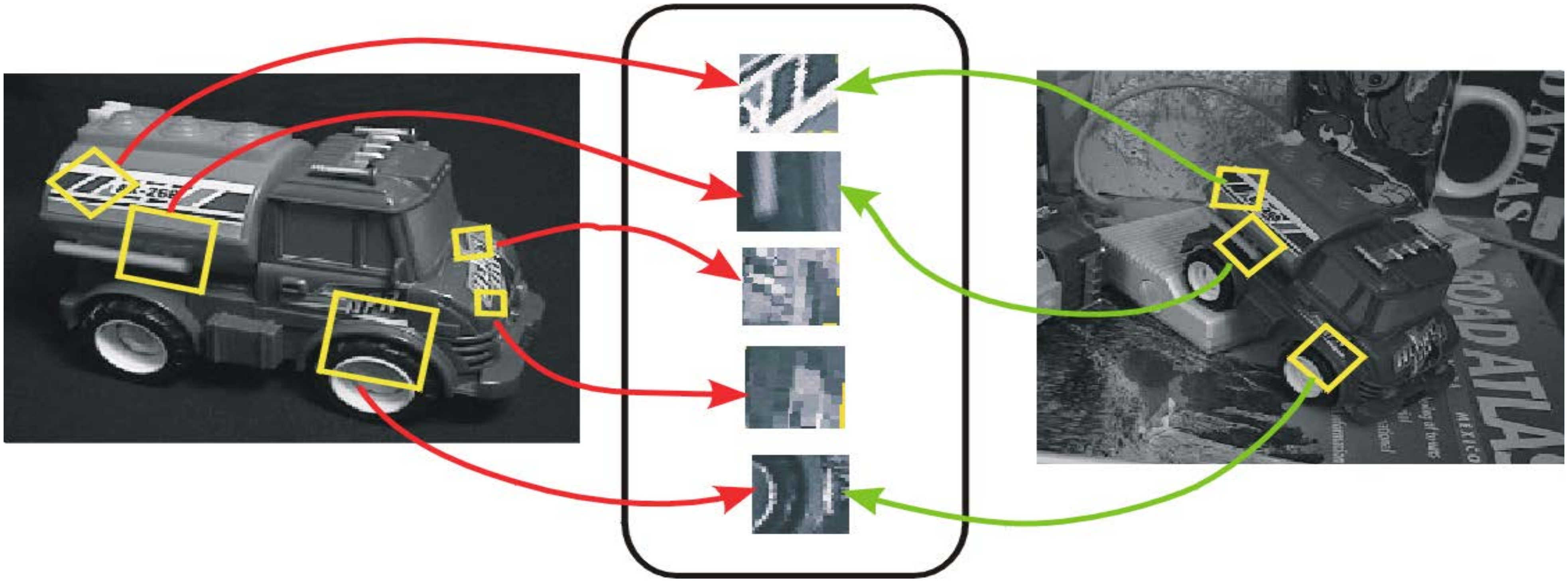
David G. Lowe. "Distinctive image features from scale-invariant keypoints." ***IJCV*** 60 (2), pp. 91-110, 2004.

General Procedure for Point Correspondence

- 5-Step Procedure
  1. Interest Point Detection
  2. Extract and Normalize Region around Interest Point
  3. Compute Local Descriptor
  4. Match Local Descriptor
  5. Estimate Global Transformation
     - to align images (e.g. image stitching)
     - to verify point correspondence globally (e.g. object recognition)

Recognition with Local Features

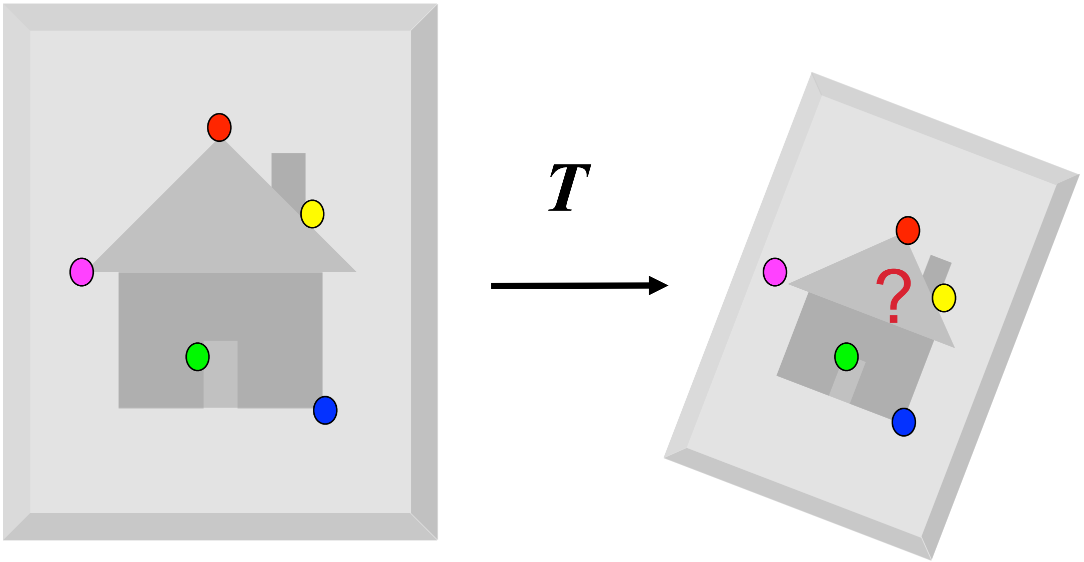
- Image content is transformed into local features that are invariant to translation, rotation, and scale
- Goal: Verify if they belong to a consistent configuration



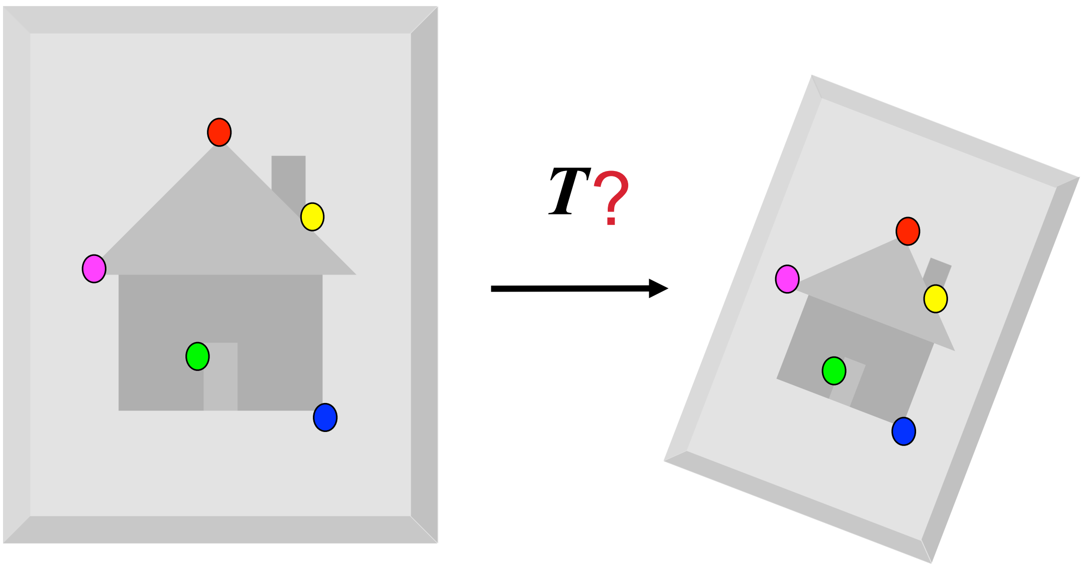
Local Features,
e.g. SIFT

Slide credit: David Lowe

Warping vs. Alignment

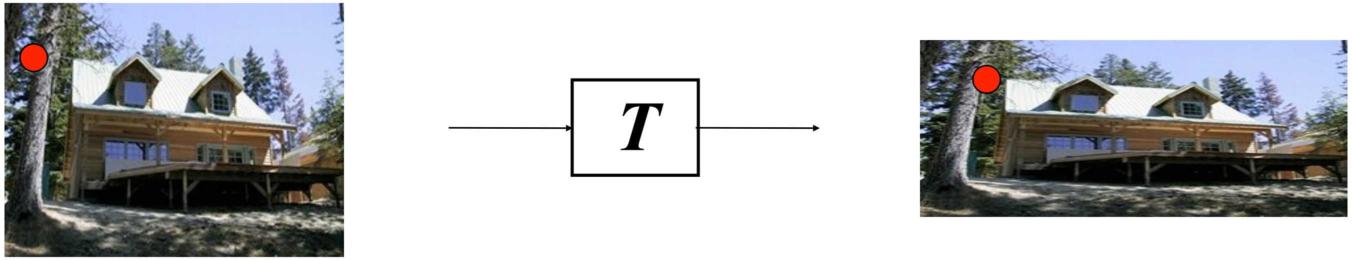


Warping: Given a source image and a transformation T, what does the transformed output look like?



Alignment: Given two images with corresponding points, what is the transformation T between them?

Parametric (Global) Warping



p = (x,y) p’ = (x’,y’)

- Transformation T is a coordinate-changing machine:
  p’ = T(p)

- What does it mean that T is global?
  - It’s the same for any point p
  - It can be described by just a few numbers (parameters)

- Let’s represent T as a matrix:
  p’ = Mp ,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{M} \begin{bmatrix} x \\ y \end{bmatrix}$$

Slide credit: Alexej Efros

What Can be Represented by a 2×2 Matrix?

- 2D Scaling?

$$x' = s_x * x$$
$$y' = s_y * y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- 2D Rotation around (0,0)?

$$x' = \cos\theta * x - \sin\theta * y$$
$$y' = \sin\theta * x + \cos\theta * y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- 2D Shearing?

$$x' = x + sh_x * y$$
$$y' = sh_y * x + y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Slide credit: Alexej Efros

What Can be Represented by a 2×2 Matrix?

- 2D Mirror about y axis?

$$x' = -x$$
$$y' = y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- 2D Mirror over (0,0)?

$$x' = -x$$
$$y' = -y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- 2D Translation?

$$x' = x + t_x$$
$$y' = y + t_y$$

NO!

Slide credit: Alexej Efros

2D Linear Transforms

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Only linear 2D transformations can be represented with a 2x2 matrix.
- Linear transformations are combinations of …
  - Scale,
  - Rotation,
  - Shear, and
  - Mirror

Slide credit: Alexej Efros

Homogeneous Coordinates

- Q: How can we represent translation as a 3x3 matrix using homogeneous coordinates?

$$x' = x + t_x$$

$$y' = y + t_y$$

- A: Using the rightmost column:

$$\text{Translation} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Slide credit: Alexej Efros

Basic 2D Transformations in homogeneous coordinates

- Basic 2D transformations as 3x3 matrices

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translation

$$\begin{bmatrix} \boldsymbol{x'} \\ \boldsymbol{y'} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s_x} & 0 & 0 \\ 0 & \boldsymbol{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Scaling

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shearing

Slide credit: Alexej Efros

2D Affine Transformations

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

- Affine transformations are combinations of …
  - Linear transformations, and
  - Translations

- Parallel lines remain parallel

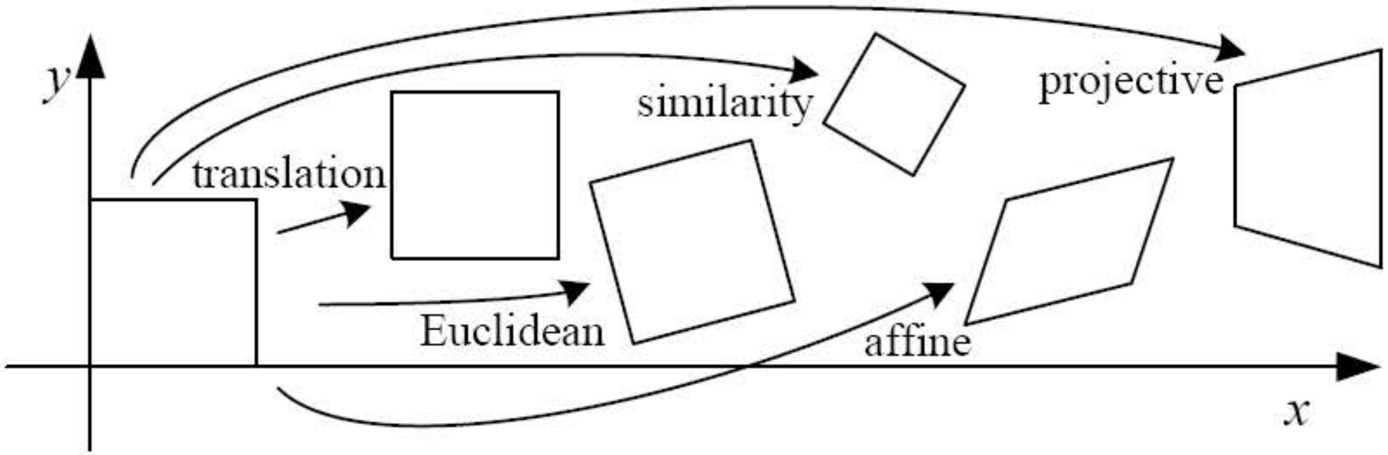
Slide credit: Alexej Efros

Projective Transformations

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

- Projective transformations:
  - Affine transformations, and
  - Projective warps

- Parallel lines do not necessarily remain parallel



Slide credit: Alexej Efros

Overview Today

- Object Identification by Point Correspondences
  - general procedure for recognition, stereo, image stitching, ...
- Interest Point Detection & Descriptor
  - local interest point detection
  - scale-invariant interest point detection
  - local image descriptor
- Scaling to Large Numbers of Images and Objects
  - inverted file
  - visual vocabulary

Overview Interest Point Detection

- Local Interest Point Detection
  - contour based methods
  - intensity based methods
    - Examples: Harris, Hessian

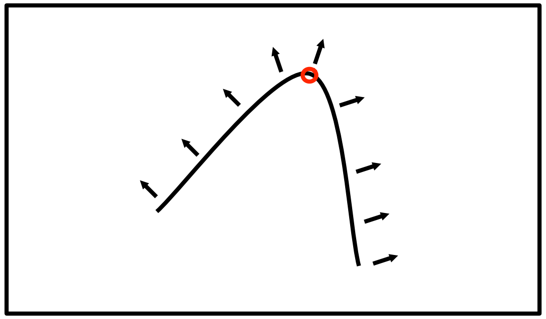
- Scale-Invariant Interest Point Detection
  - matching images of different scales
  - automatic scale selection
  - scale invariant methods for feature extraction
    - Example: Harris-Laplace

Why LOCAL interest points?

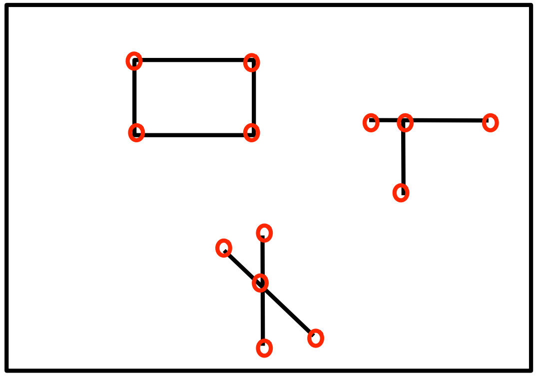
- Global fail in
  - Image transformations
    - e.g. scale change
  - Occlusions and background clutter
    - i.e. segmentation is difficult
  - Color
    - changes in non-uniform lighting
  - Geometric
    - contour based (fail if no shape)

Interest point detectors
Contour based methods

- Detecting curvature change
  - Detecting edges
  - Detecting sudden edge orientation change

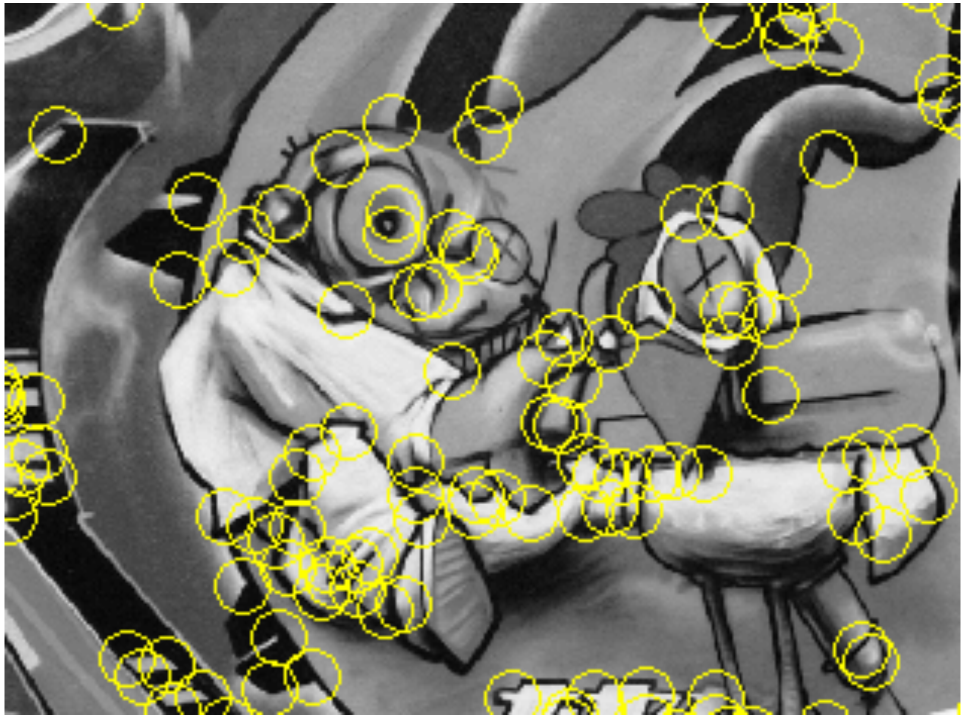


- Detecting intersections of line segments
  - Detecting edges
  - Fitting line segments to the edges
    i.e., Hough transform
  - Finding intersections



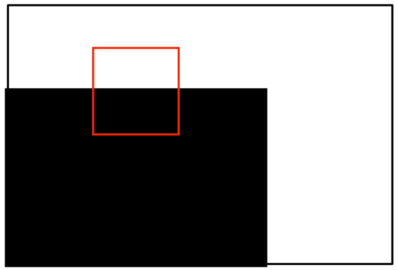
Local Interest Points
Intensity Based Methods

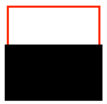
- Interest points
  - Two dimensional signal change
  - More complex local structures



Interest point detectors
Intensity based methods [Moravec'77]

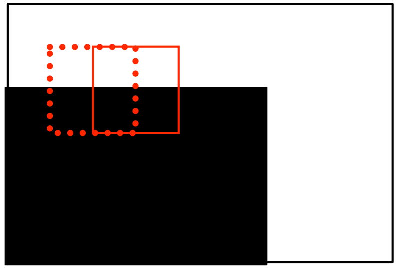
- Autocorrelation function

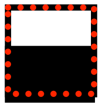




Interest point detectors
Intensity based methods [Moravec'77]

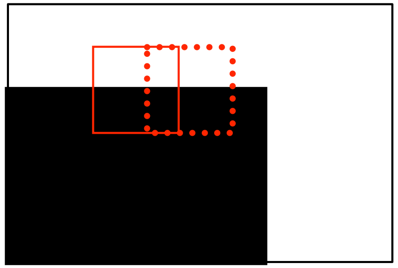
- Autocorrelation function

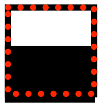
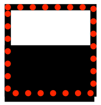


a= ( - )

Interest point detectors
Intensity based methods [Moravec'77]

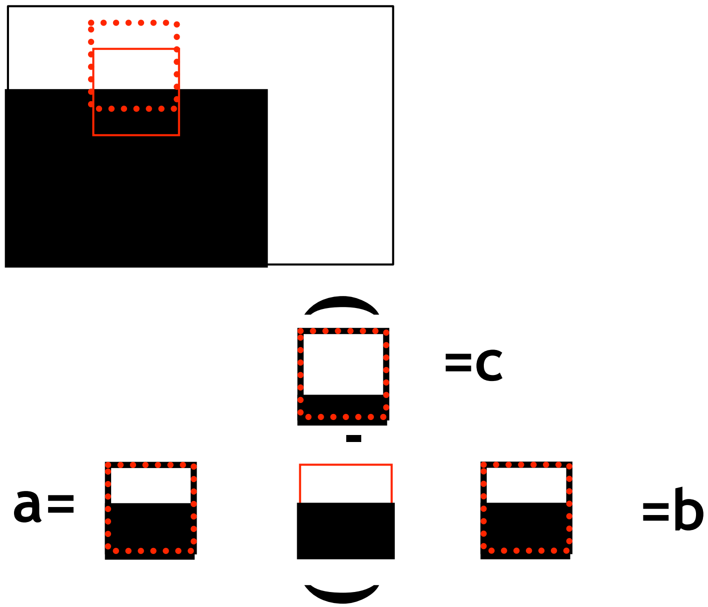
- Autocorrelation function



a=  ( - ) =b

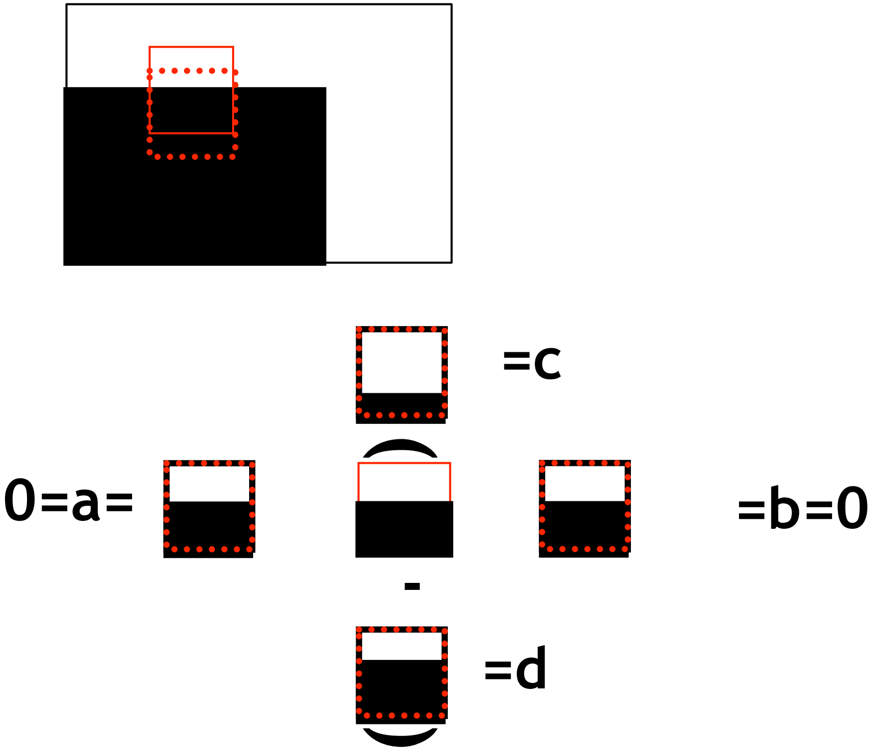
Interest point detectors
Intensity based methods [Moravec’77]

- Autocorrelation function



Interest point detectors
Intensity based methods [Moravec'77]

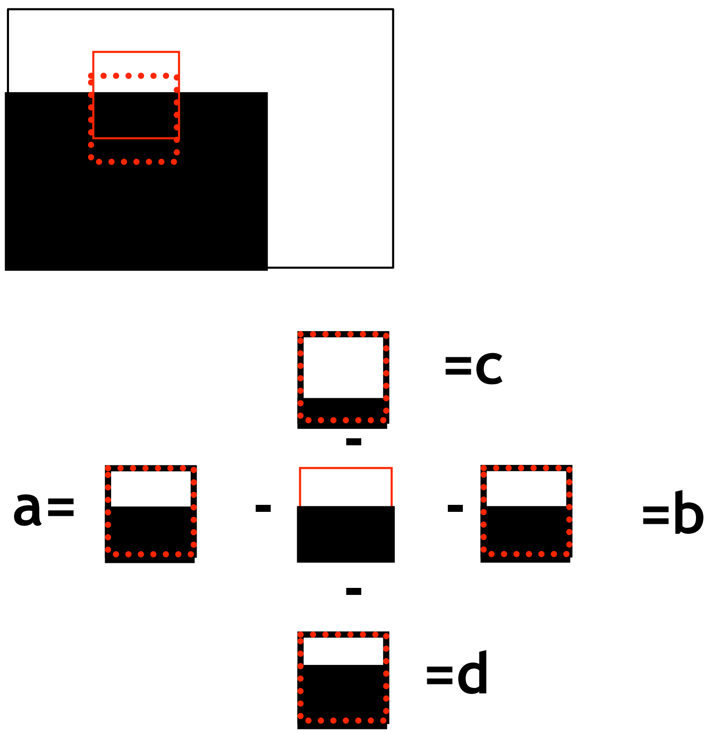
- Autocorrelation function



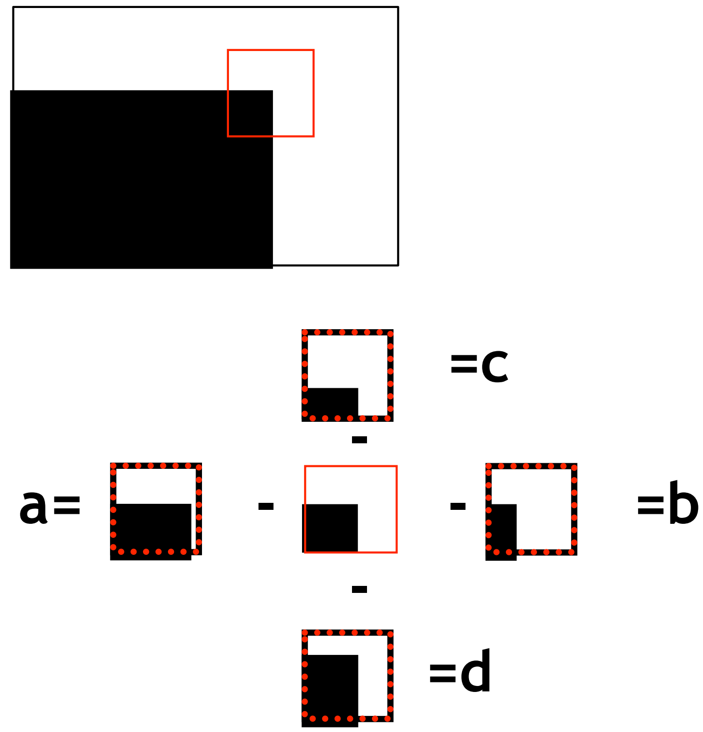
min(a^2,b^2,c^2,d^2) > T

Interest point detectors
Intensity based methods [Moravec'77]

- Autocorrelation function (in this case not product, but sum of squared differences SSD)



min(a^2,b^2,c^2,d^2) > T



min(a^2,b^2,c^2,d^2) > T

Interest point detectors
Intensity based methods [Harris'88]

- More general: auto correlation function:

$$E_{\mathrm{AC}}(\Delta \boldsymbol{u}) = \sum_i w(\boldsymbol{x}_i)[I_0(\boldsymbol{x}_i + \Delta \boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2$$

offset weighting

- Taylor series expansion of image function:

$$I_0(\boldsymbol{x}_i + \Delta \boldsymbol{u}) \approx I_0(\boldsymbol{x}_i) + \nabla I_0(\boldsymbol{x}_i) \cdot \Delta u \qquad \nabla I_0(\boldsymbol{x}_i) = (\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y})(\boldsymbol{x}_i)$$

- Approximation of auto correlation function:

$$\begin{aligned} E_{\mathrm{AC}}(\Delta \boldsymbol{u}) &= \sum_i w(\boldsymbol{x}_i)[I_0(\boldsymbol{x}_i + \Delta \boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2 \\ &\approx \sum_i w(\boldsymbol{x}_i)[I_0(\boldsymbol{x}_i) + \nabla I_0(\boldsymbol{x}_i) \cdot \Delta \boldsymbol{u} - I_0(\boldsymbol{x}_i)]^2 \\ &= \sum_i w(\boldsymbol{x}_i)[\nabla I_0(\boldsymbol{x}_i) \cdot \Delta \boldsymbol{u}]^2 \\ &= \Delta \boldsymbol{u}^T \boldsymbol{A} \Delta \boldsymbol{u}, \end{aligned}$$

$$\boldsymbol{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

<u>in each direction now!</u>

Calculation of Image Derivatives

- 1st derivative:

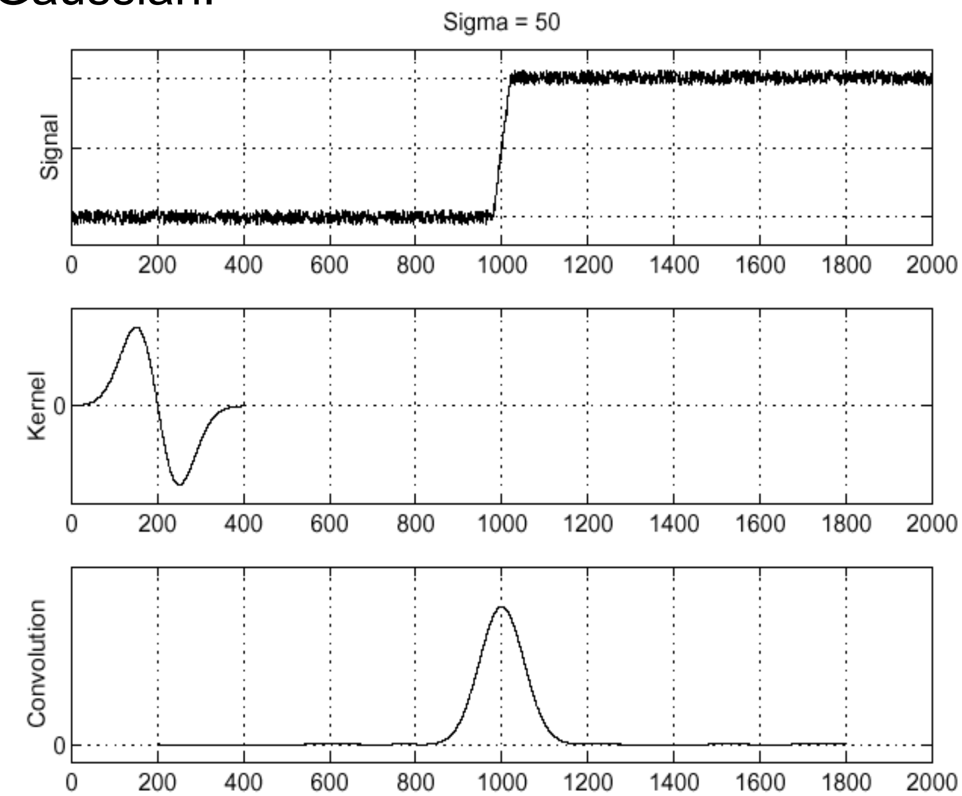
$$\frac{d}{dx}(g \otimes f) = \left(\frac{d}{dx} g\right) \otimes f$$

  - convolution with 1st derivative of a Gaussian:

$$f$$

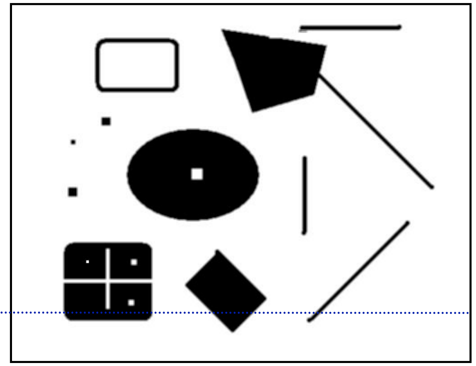
$$\frac{d}{dx} g$$

$$(\frac{d}{dx} g) \otimes f$$



Interest point detectors
Intensity based methods [Harris'88]

- Second moment matrix
  autocorrelation matrix



$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

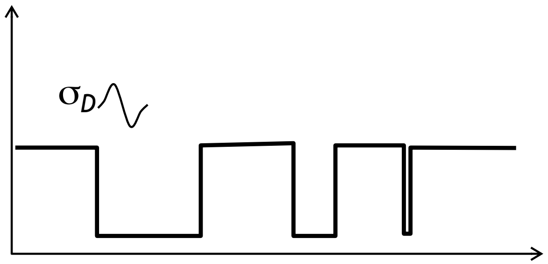
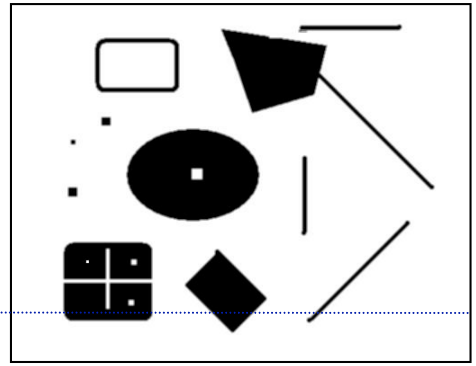


Image row of pixels

Interest point detectors
Intensity based methods [Harris'88]

- Second moment matrix
  autocorrelation matrix



$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_xI_y(\sigma_D) \\ I_xI_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

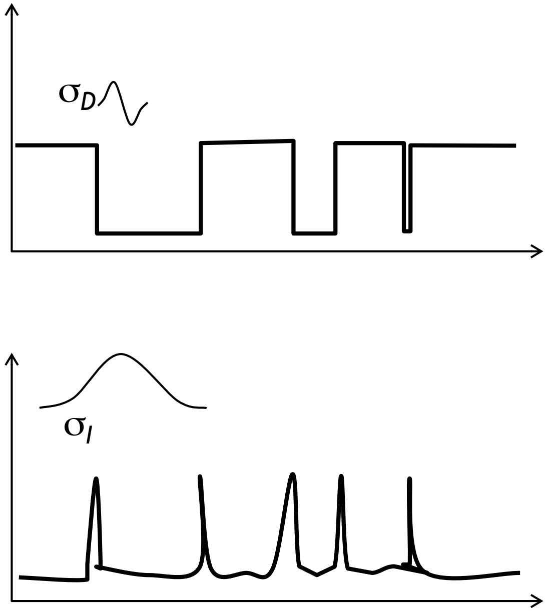


Image row of pixels

Eigenvalues-reminder

- Singular value decomposition

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}^T$$

$$= U \cdot D \cdot V^T$$

eigenvectors $\begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} \begin{bmatrix} v_{12} \\ v_{22} \end{bmatrix}$

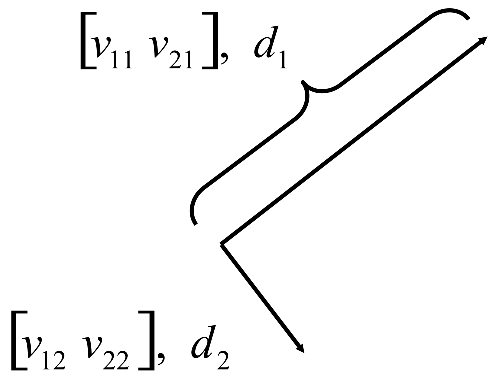
$$U \cdot U^T = V \cdot V^T = I$$

$$U^T = U^{-1} \qquad V^T = V^{-1}$$

eigenvalues $d_1, d_2 \geq 0$

determinant $det(A) = ad - cb = d_1 d_2$

Eigenvector, eigenvalue



Interest point detectors
Intensity based methods [Harris'88]

- Harris Corner Detector
  - looks at eigenvalues d_1 and d_2 of second moment matrix A
    - if d_1 and d_2 are small -> no feature of interest around (x,y)
    - if d_1 small and d_2 is some large value -> then an edge is found at (x,y)
    - if d_1 and d_2 are both large values -> then a corner is found at (x,y)

- criteria:

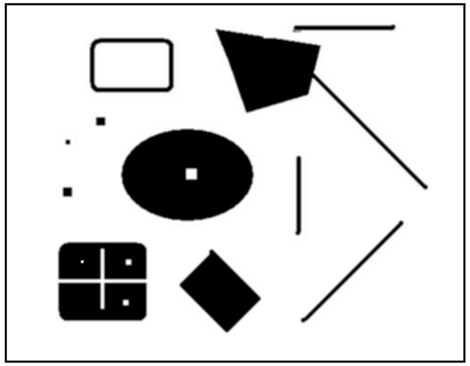
$$cornerness = d_1 d_2 - \alpha(d_1 + d_2)^2$$

$$cornerness = det(A) - \alpha(trace(A))^2$$

- no need to compute eigenvalues: det and trace

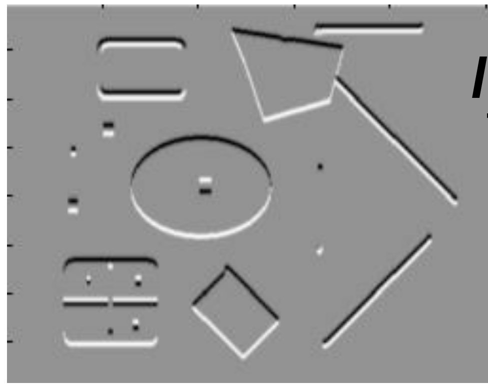
Interest point detectors
Intensity based methods [Harris'88]

- Second moment matrix
  autocorrelation matrix

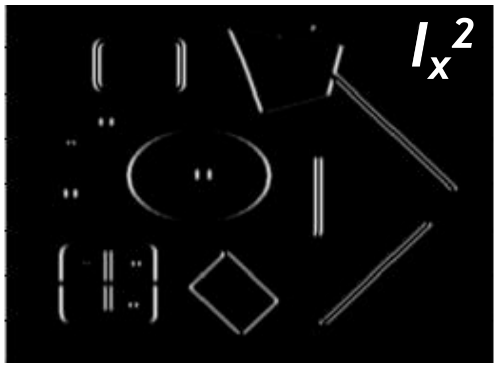


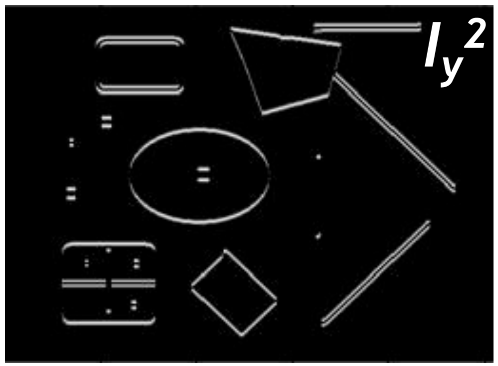
$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

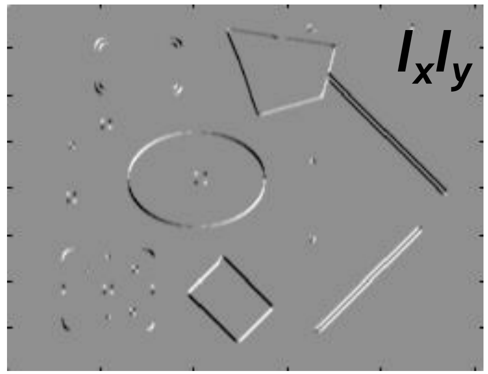
1. Image derivatives
   $I_x(\sigma_D)$, $I_y(\sigma_D)$



2. Square of derivatives



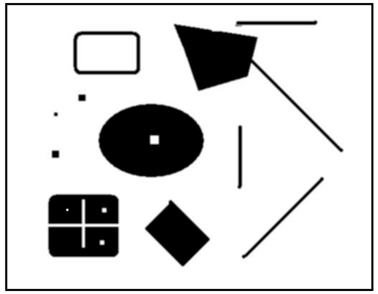




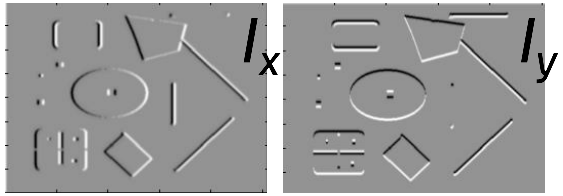
Interest point detectors
Intensity based methods [Harris'88]

- Second moment matrix
  autocorrelation matrix

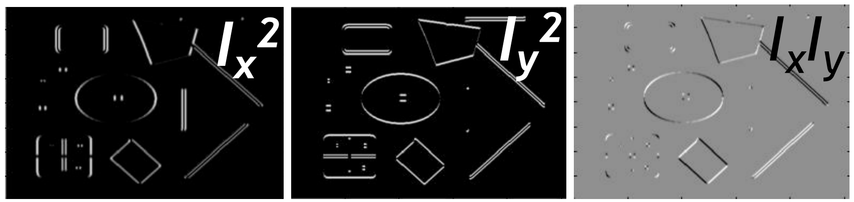
$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



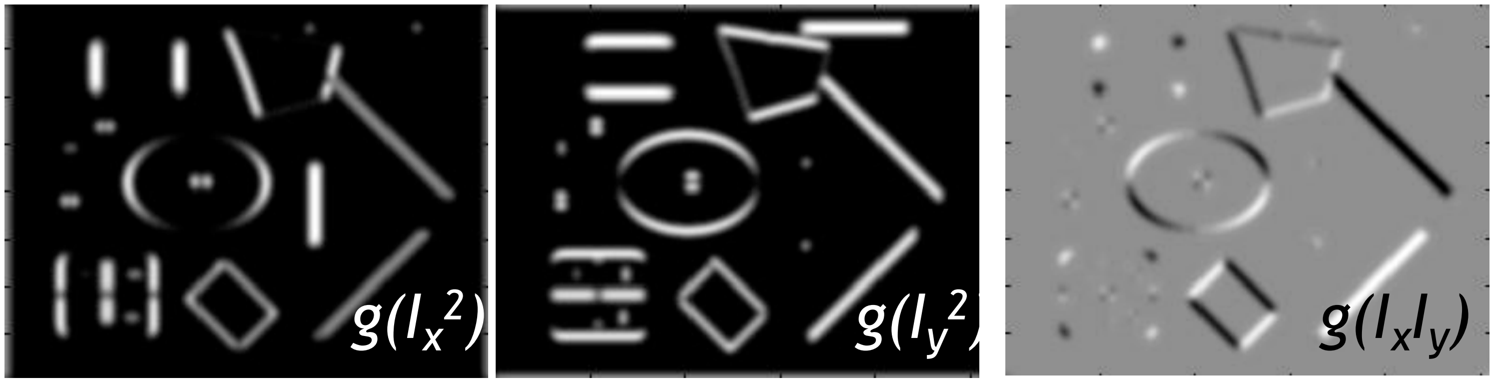
1. Image derivatives



2. Square of derivatives

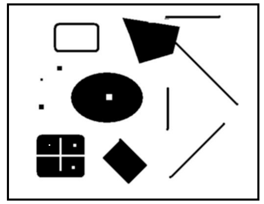


3. Gaussian filter $g(\sigma_I)$



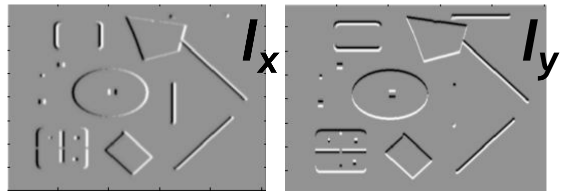
Interest point detectors
Intensity based methods [Harris'88]

- Second moment matrix
  autocorrelation matrix

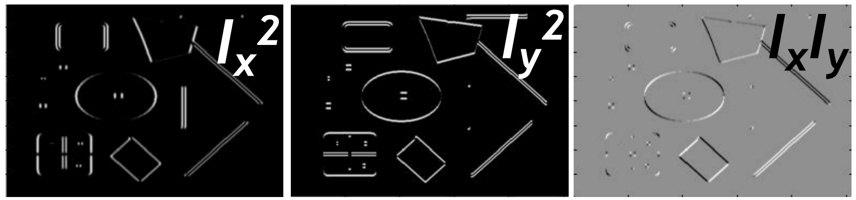


$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

1. Image derivatives



2. Square of derivatives



3. Gaussian filter $g(\sigma_I)$

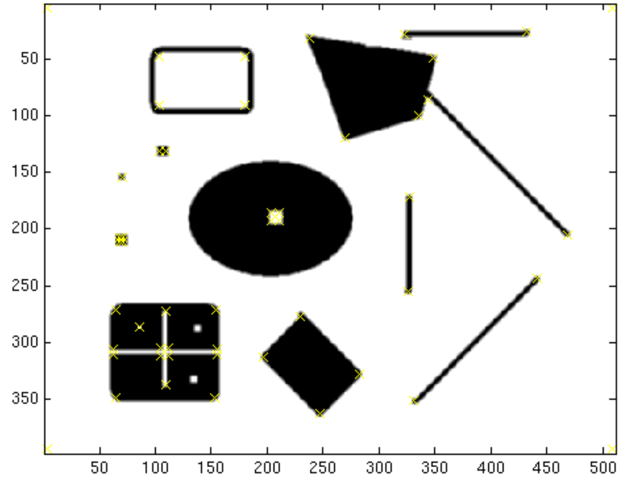
4. Cornerness function - both eigenvalues are strong

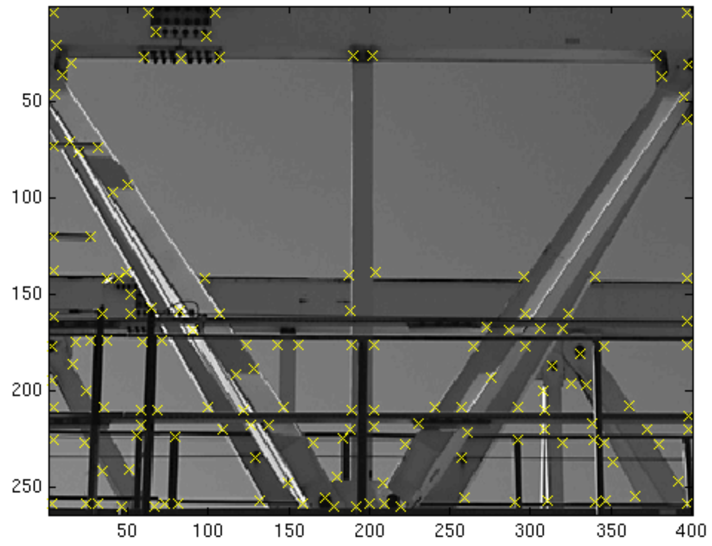
$$har = \det[\mu(\sigma_I,\sigma_D)] - \alpha[\text{trace}^2(\mu(\sigma_I,\sigma_D))] =$$

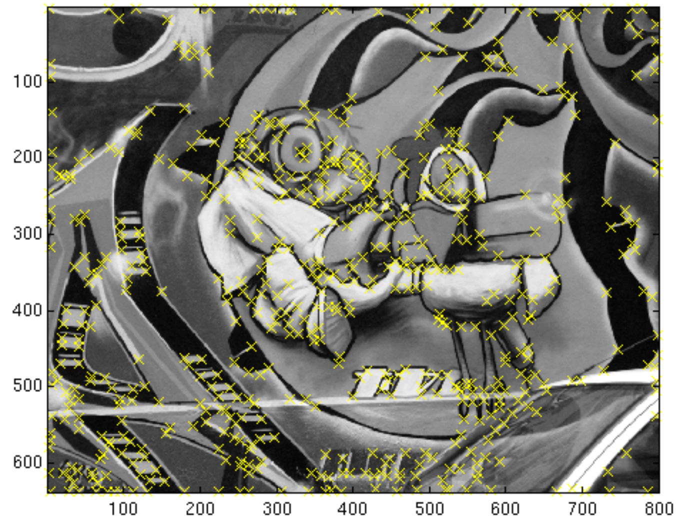
$$g(I_x^2)g(I_y^2) - [g(I_xI_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

5. Non-maxima suppression

Interest point detectors
Intensity based methods [Harris'88]

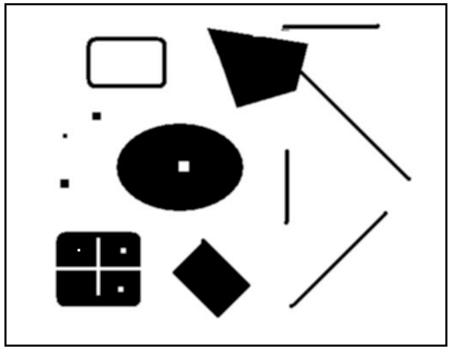






Interest point detectors
Intensity based methods [Beaudet'78]

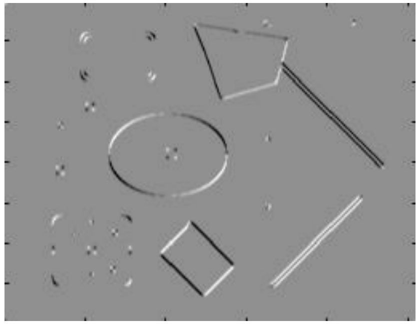
- Hessian determinant



I_{xx}

$$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

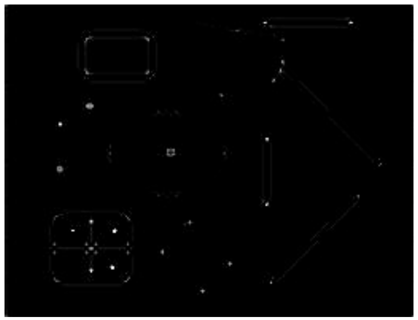
I_{xy}



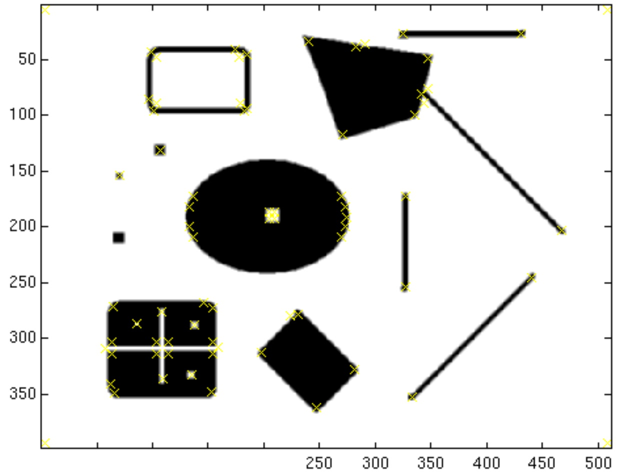
I_{yy}

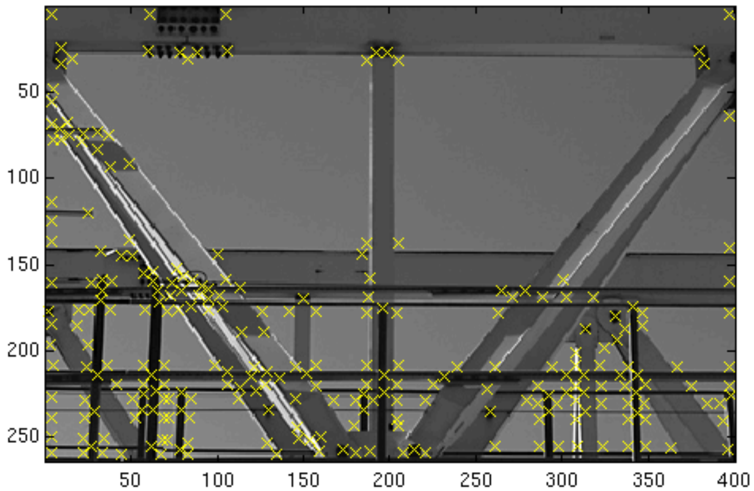
$$\det(Hessian(I)) = I_{xx} I_{yy} - I_{xy}^2$$

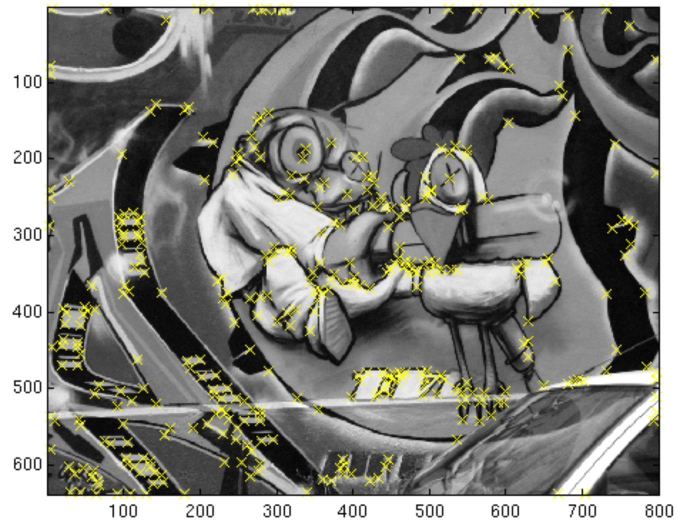
In Matlab: $I_{xx} .* I_{yy} - (I_{xy})^\wedge 2$



Interest point detectors
Intensity based methods [Beaudet'78]







Discussion

- Interest-point Detection so far:
  - using Harris or Hessian-detector
  - finds discriminant points
    (Harris detector was the “de-facto” standard for a long time)
  - used for recognition, correspondence for stereo, sparse optical flow/motion, etc.

- But: remember goals of interest point detection:
  - discriminance vs. invariance to transformation
  - Harris & Hessian find discriminant points - but they are **not** invariant
    to scale, affine and projective transformations

Overview Interest Point Detection

- Local Interest Point Detection
  - parametric model based methods
  - contour based methods
  - intensity based methods
    - Examples: Harris, Hessian

- Scale-Invariant Interest Point Detection
  - matching images of different scales
  - automatic scale selection
  - scale invariant methods for feature extraction
    - Example: Harris-Laplace

Similarity transformation
Matching patches

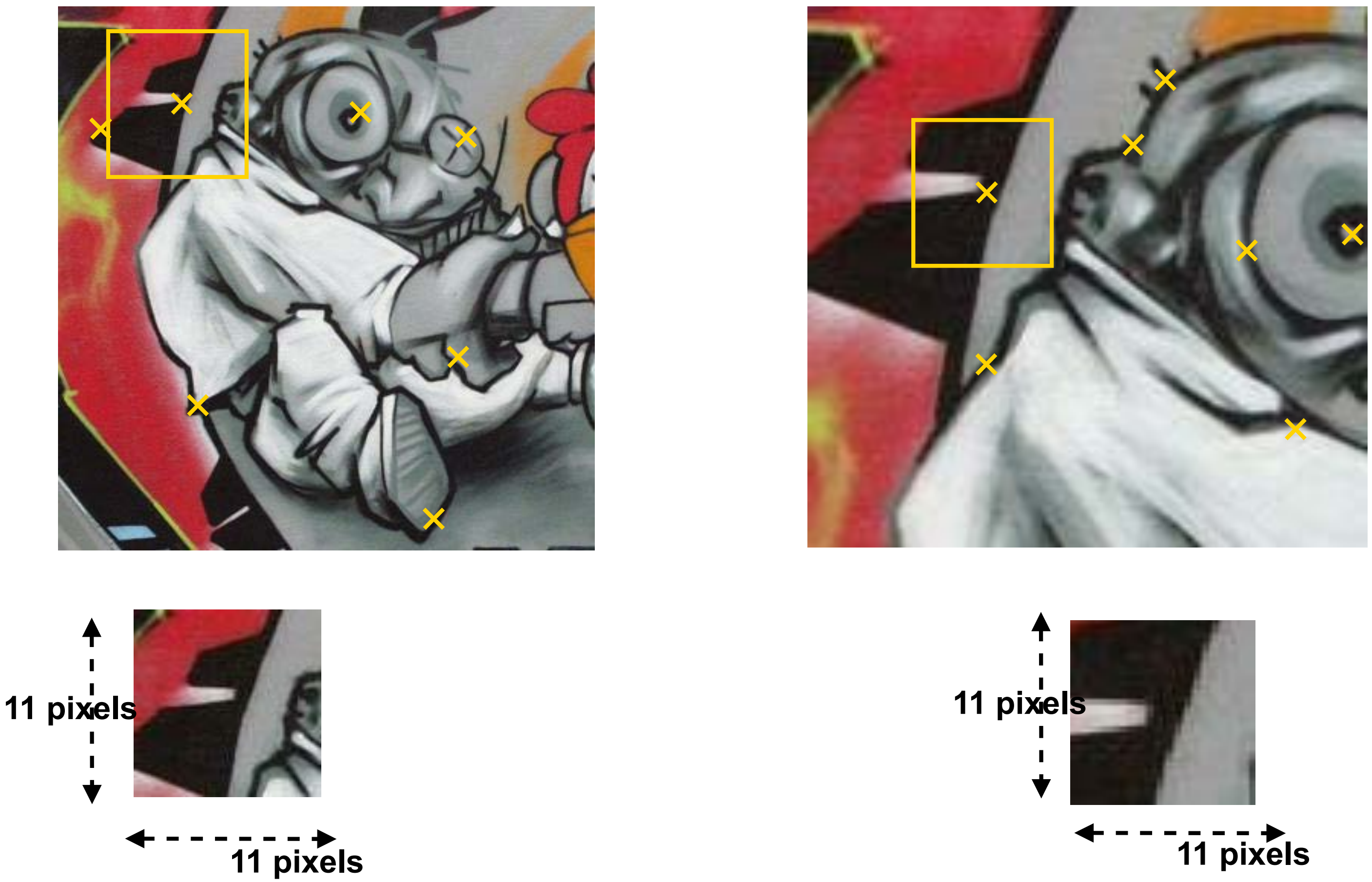
- Matching images of different scales

Similarity transformation
Matching patches

- Detecting interest points

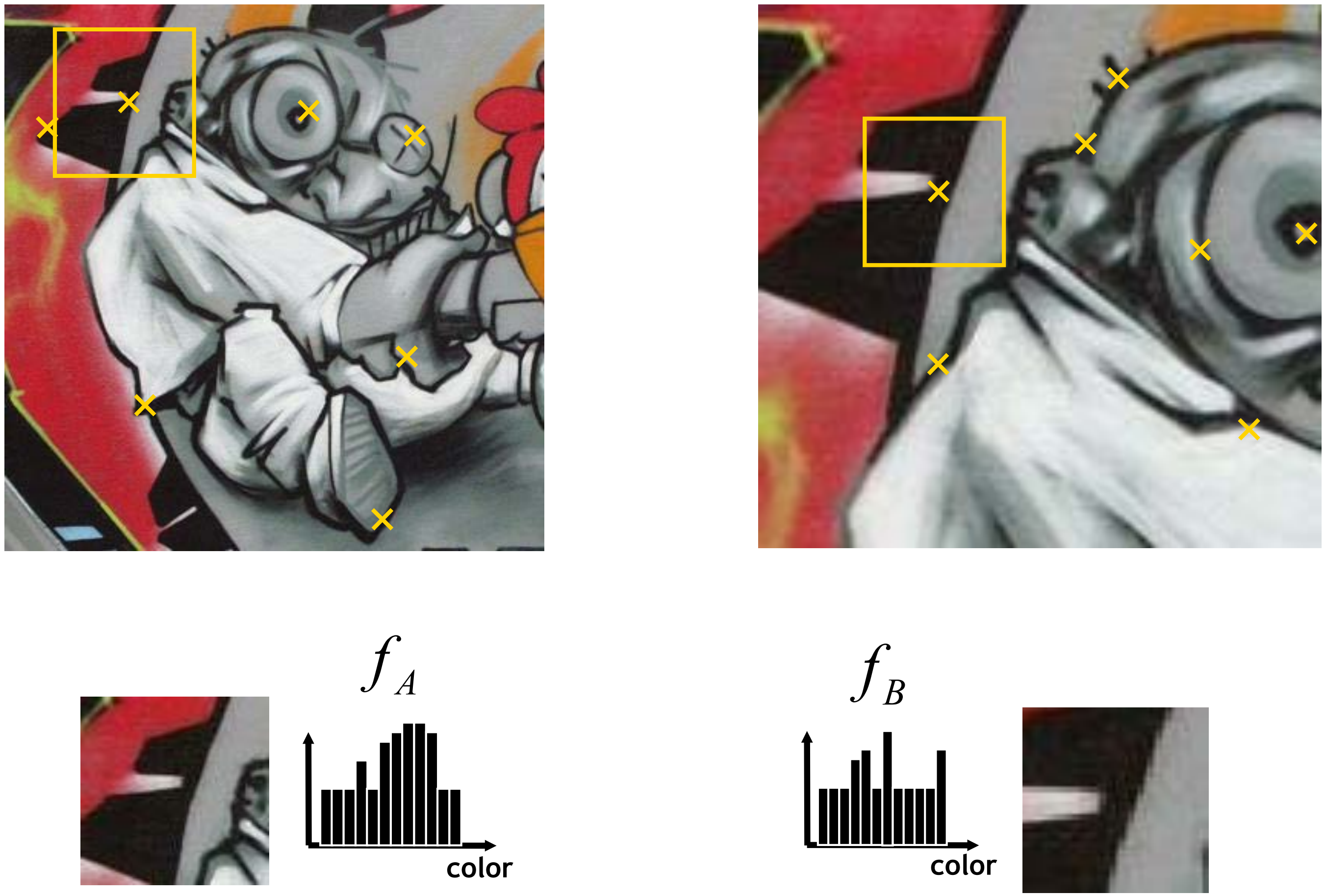
Similarity transformation
Matching patches

- Extracting patches



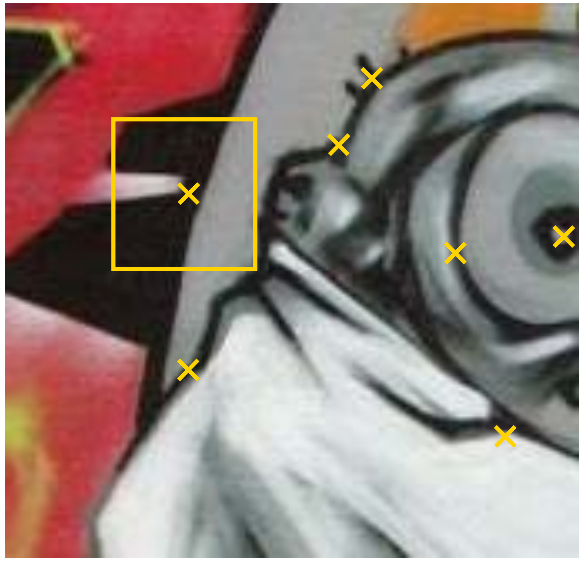
Similarity transformation
Matching patches

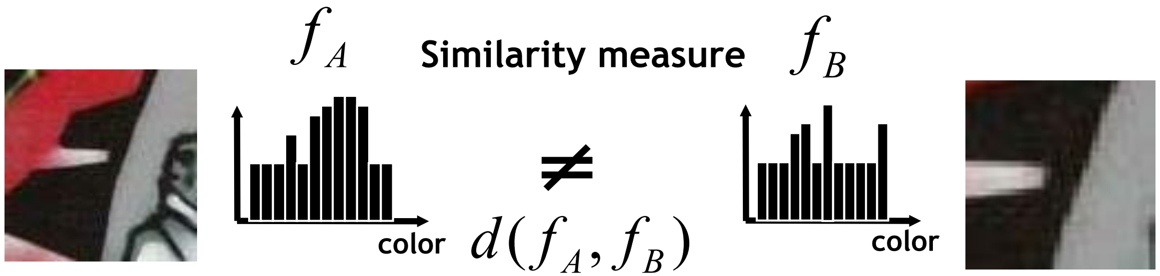
- Computing descriptors



Similarity transformation
Matching patches

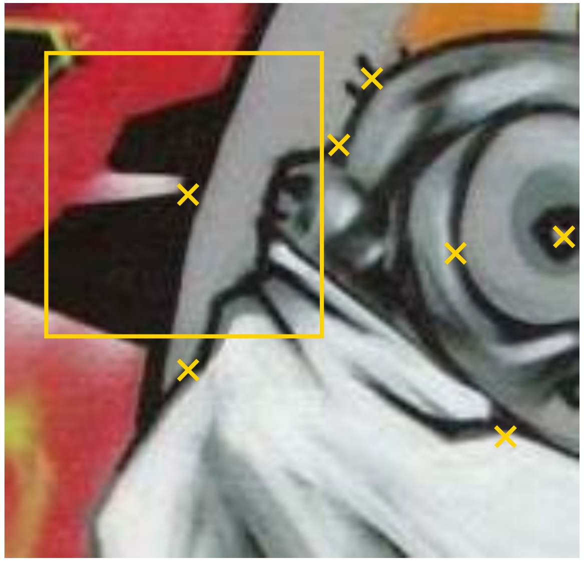
- Comparing descriptors
  - Impossible to match – different histograms due to different patch content

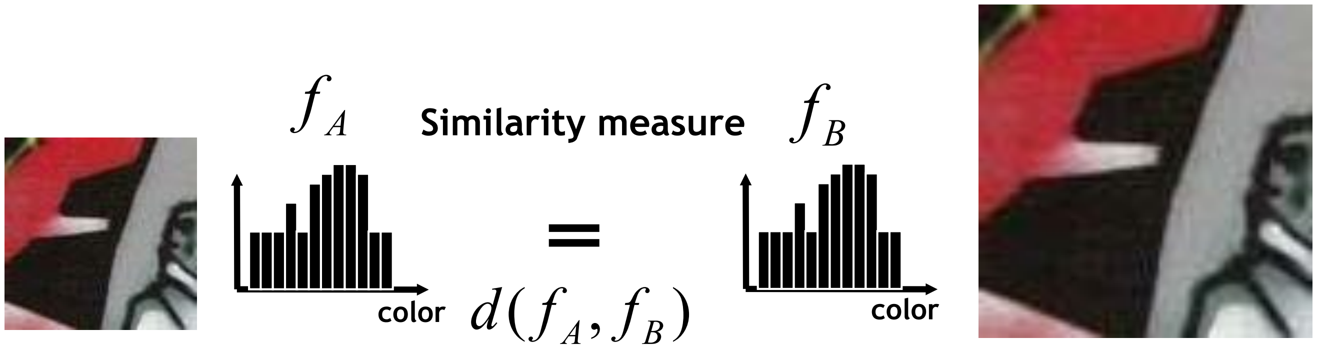





Similarity transformation
Matching patches

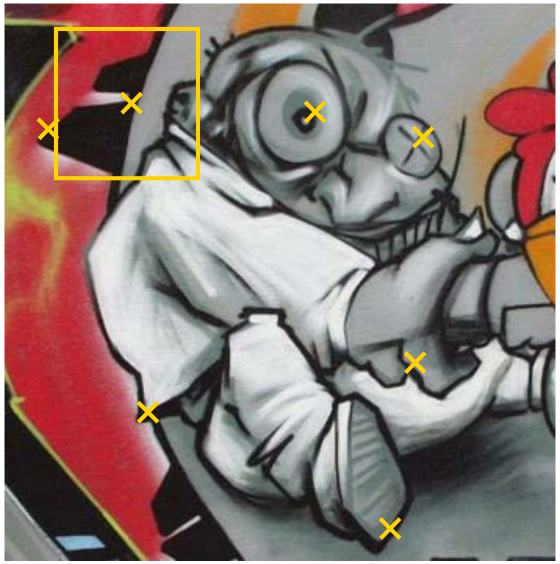
- Comparing descriptors
  - The patch should contain the same image – how to find the correct size?

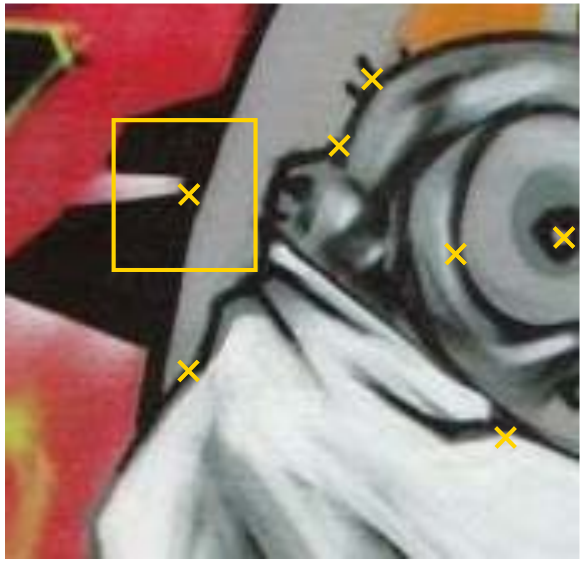


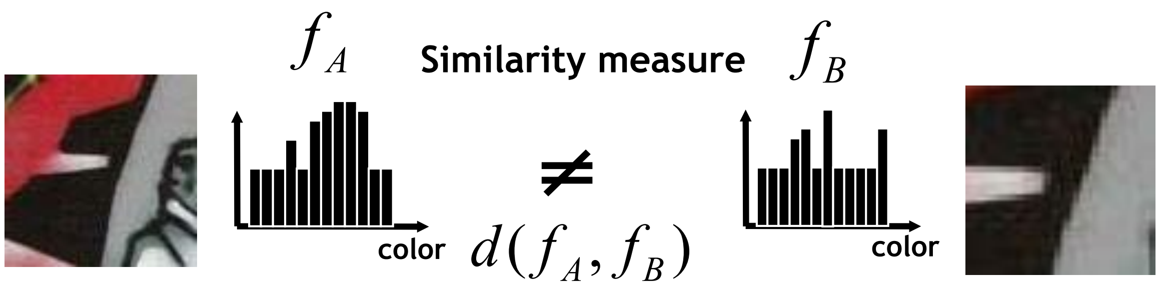


Similarity transformation
Matching patches

- Comparing descriptors while varying the patch size

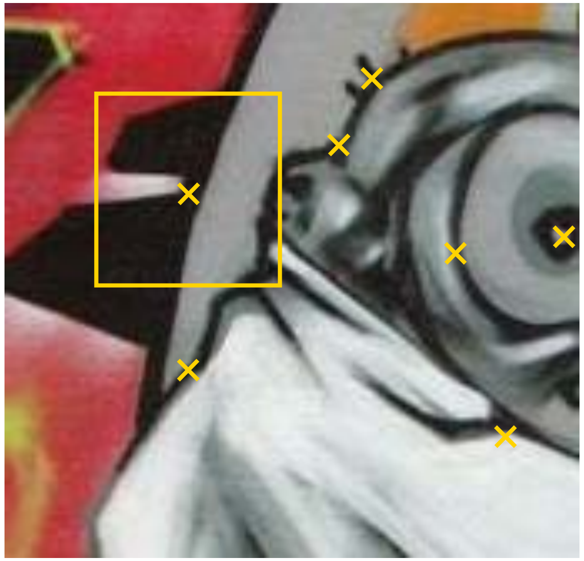


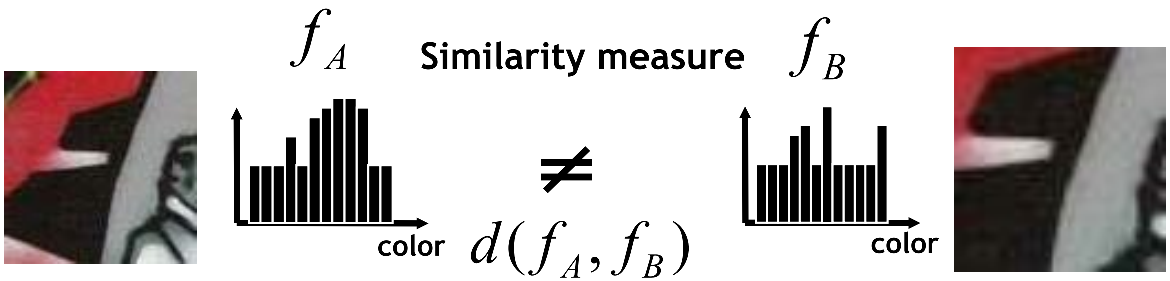




Similarity transformation
Matching patches

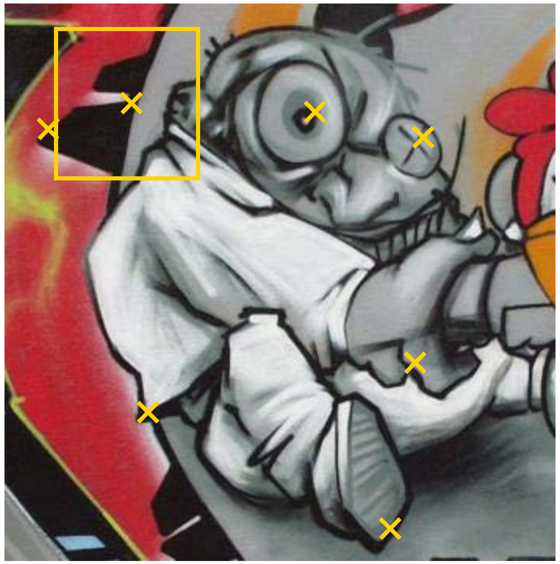
- Comparing descriptors while varying the patch size

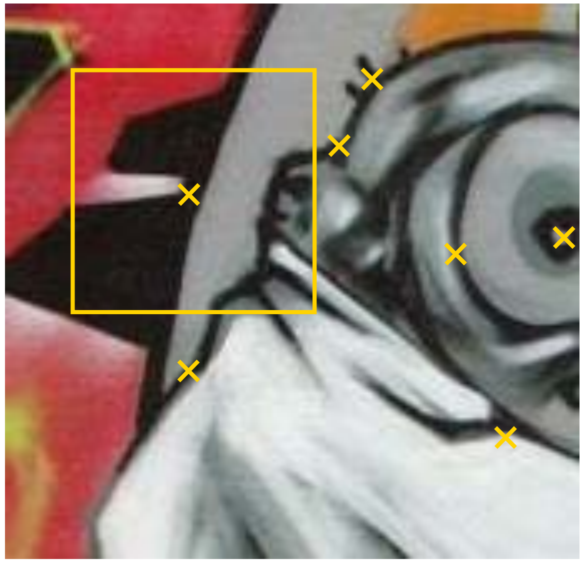


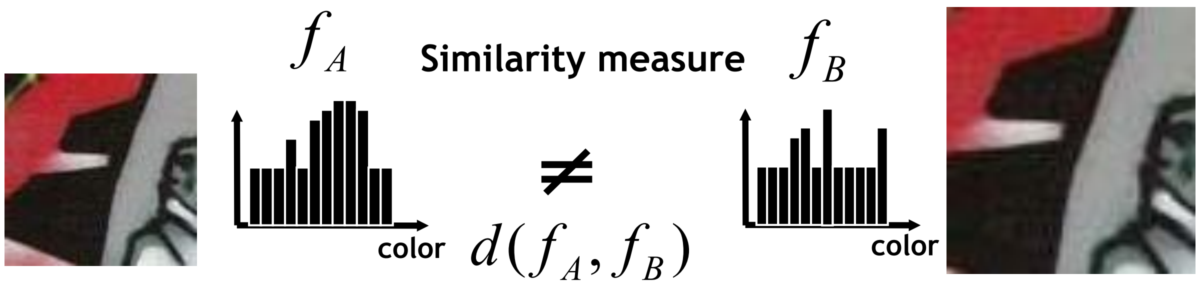


Similarity transformation
Matching patches

- Comparing descriptors while varying the patch size

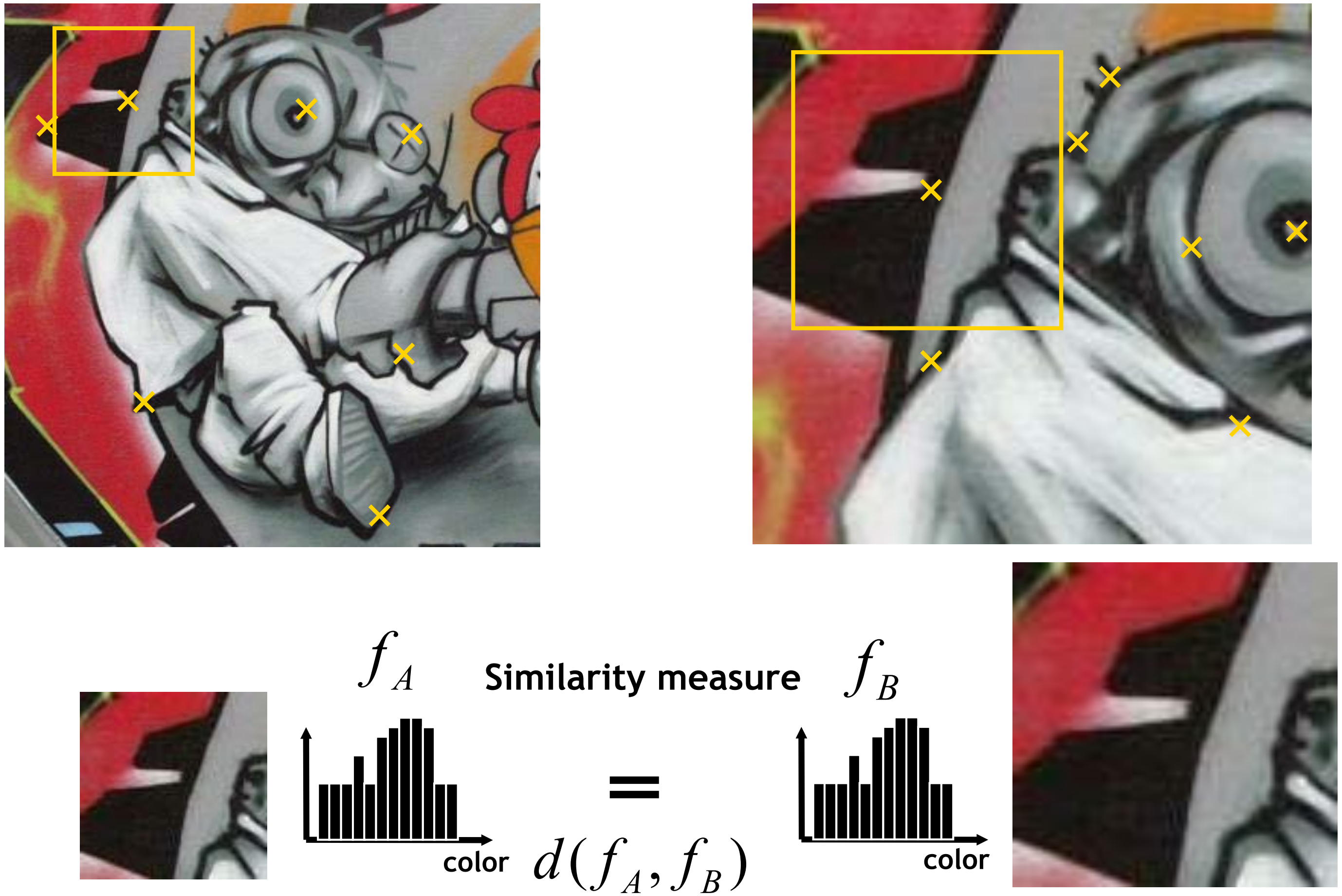






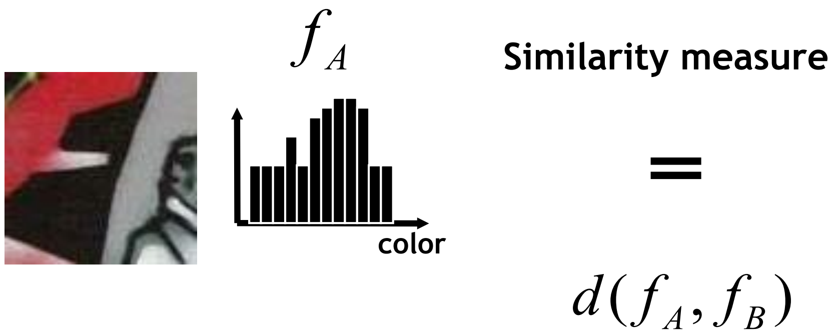
Similarity transformation
Matching patches

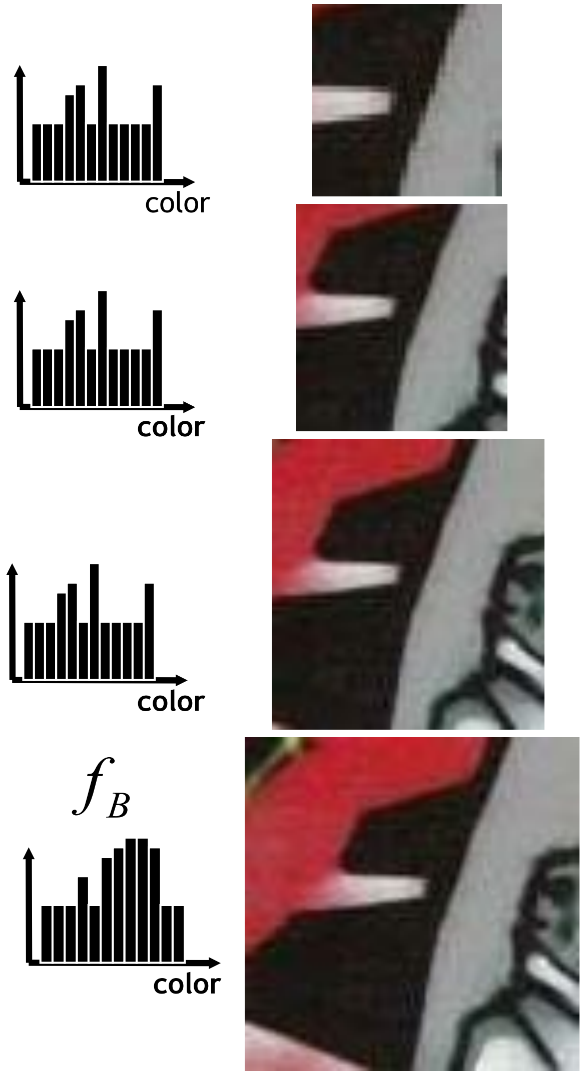
- Comparing descriptors while varying the patch size



Similarity transformation
Matching patches

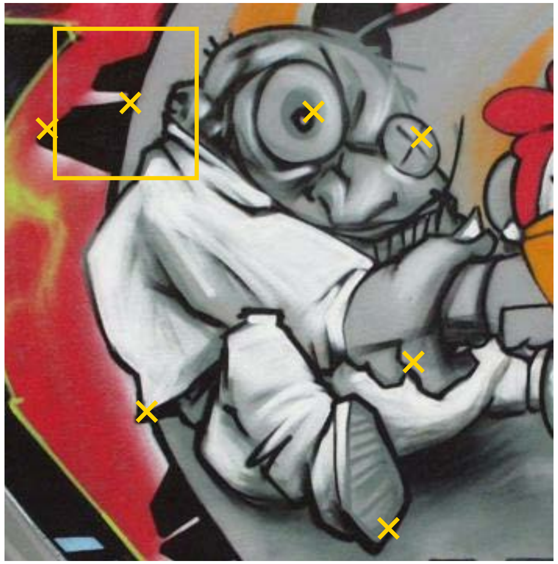
- Comparing descriptors while varying the patch size
  - Computationally inefficient/prohibitive
  - Inefficient but possible for matching
  - Prohibitive for retrieval in large databases
  - Prohibitive for recognition

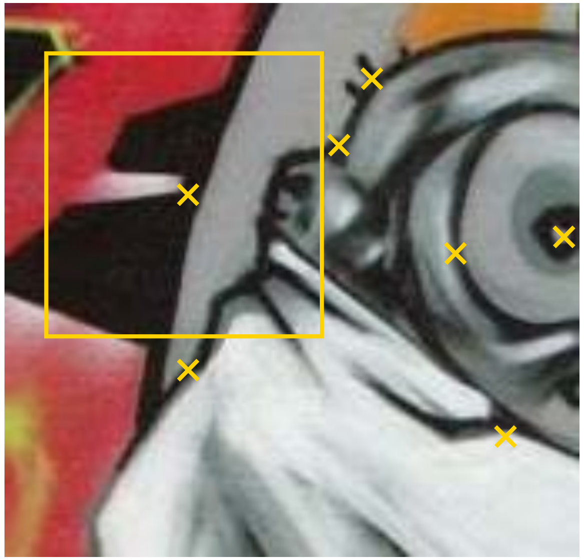


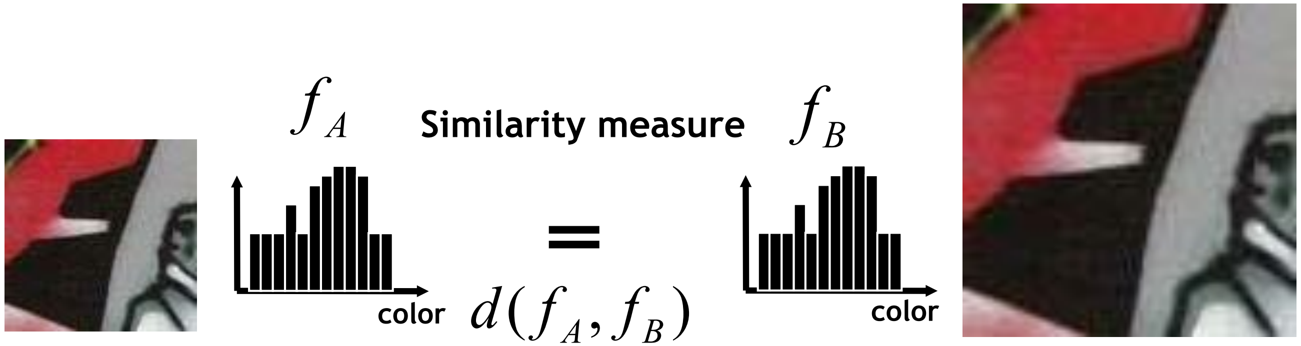


Similarity transformation
Scale invariant detector

- Detector finds location and scale of interest points
  - In both images: **independent** automatic scale detection
  - by finding "characteristic" scale of an interest point







Automatic scale selection

$$I_{i_1 \ldots i_m}(x, \sigma) = I_{i_1 \ldots i_m}(x', \sigma')$$

The same derivative responses if the patch contains the same image up to scale factor

Automatic scale selection

$$f(I_{i_1 \ldots i_m}(x, \sigma)) = f(I_{i_1 \ldots i_m}(x', \sigma'))$$

The same operator responses if the patch contains the same image up to scale factor

How to find corresponding patch sizes?

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)

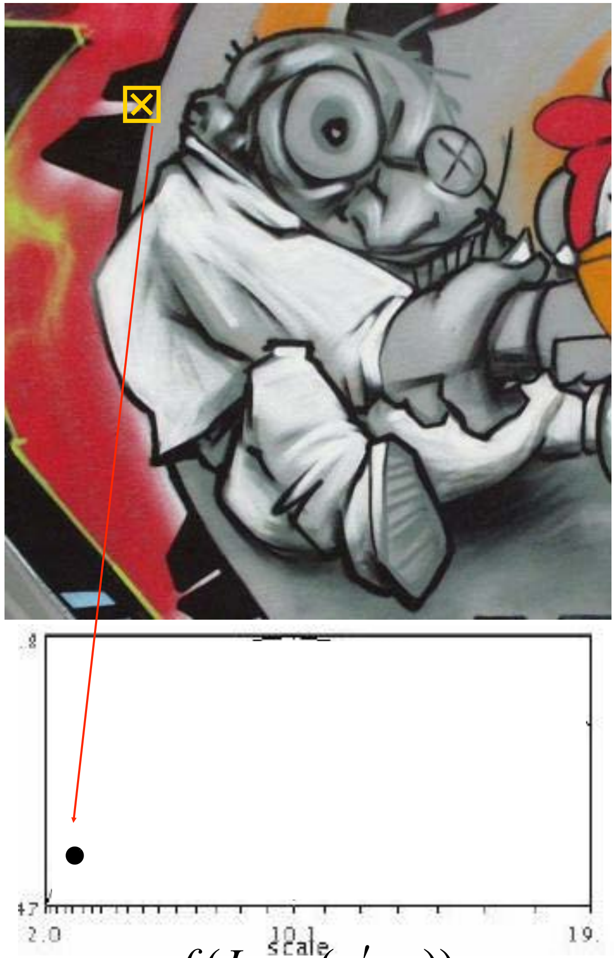
$$f(I_{i_1 \dots i_m}(x,\sigma))$$

$$f(I_{i_1 \dots i_m}(x',\sigma))$$

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)

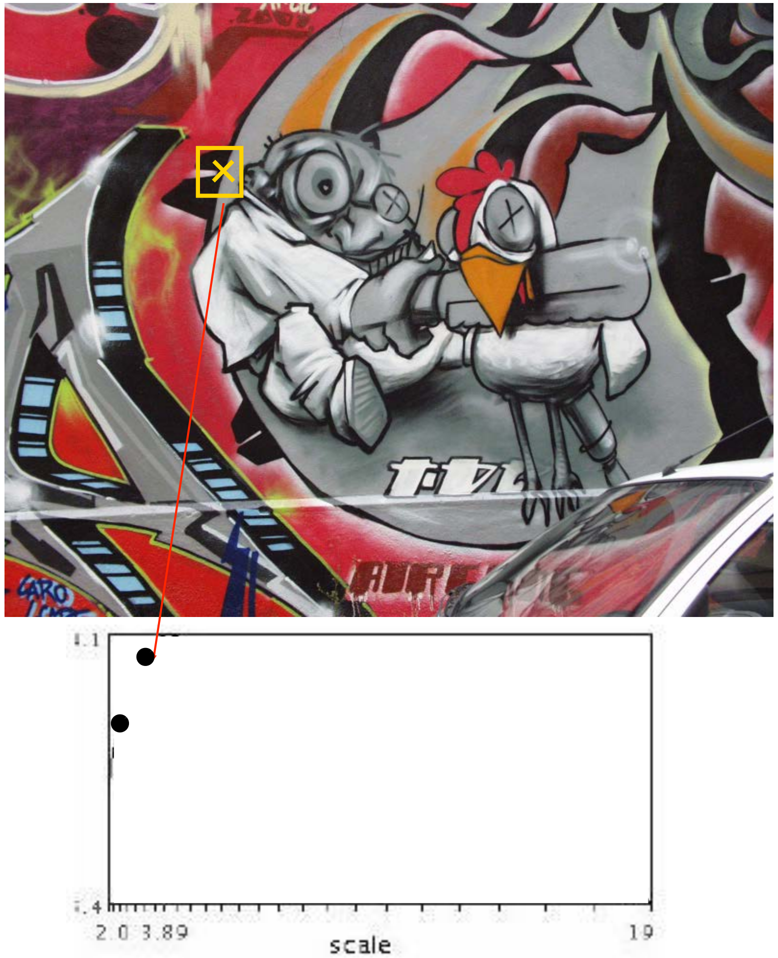
$$f(I_{i_1\ldots i_m}(x,\sigma))$$



$$f(I_{i_1\ldots i_m}(x',\sigma))$$

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)

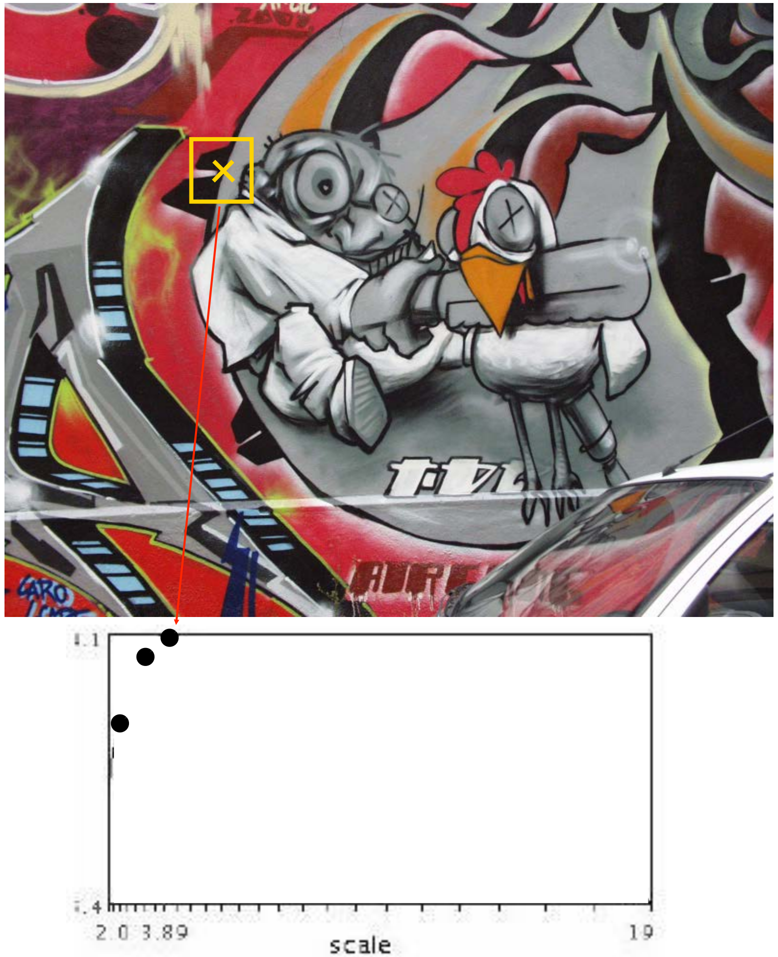


$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

$$f(I_{i_1 \ldots i_m}(x',\sigma))$$

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)

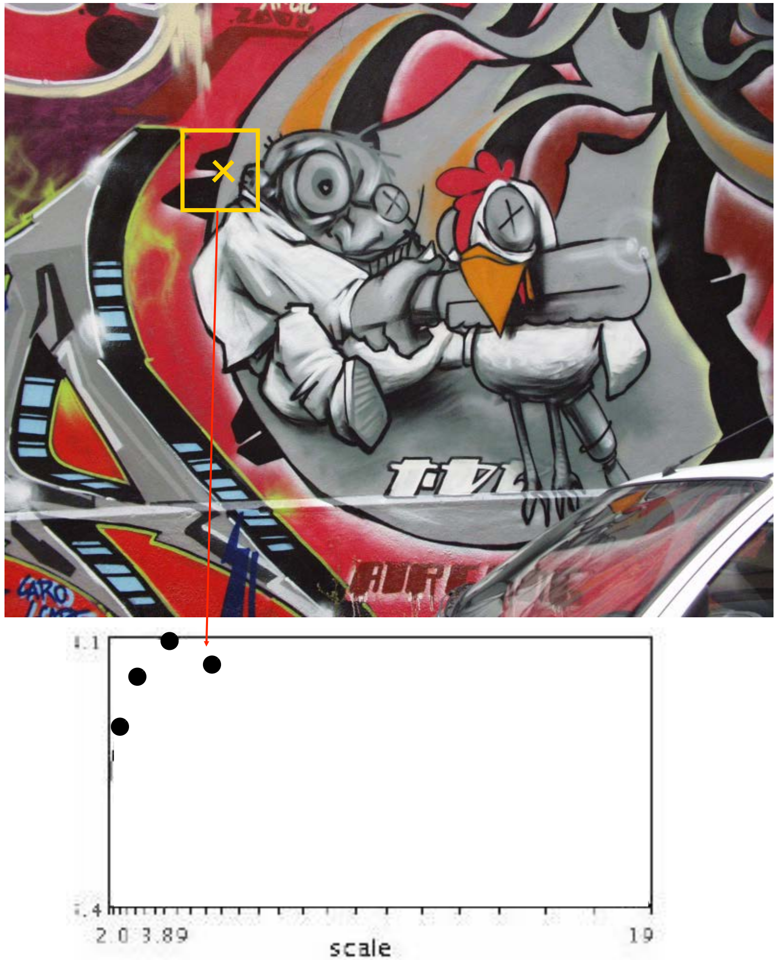


$f(I_{i_1 \ldots i_m}(x, \sigma))$

$f(I_{i_1 \ldots i_m}(x', \sigma))$

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)

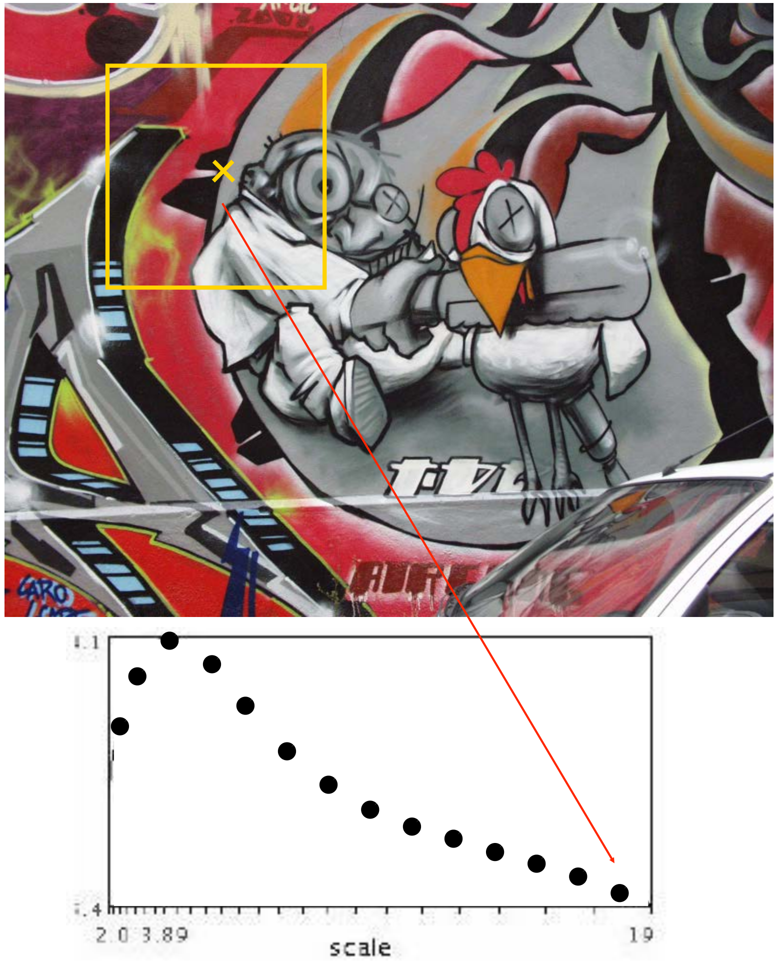


$$f(I_{i_1 \ldots i_m}(x, \sigma))$$

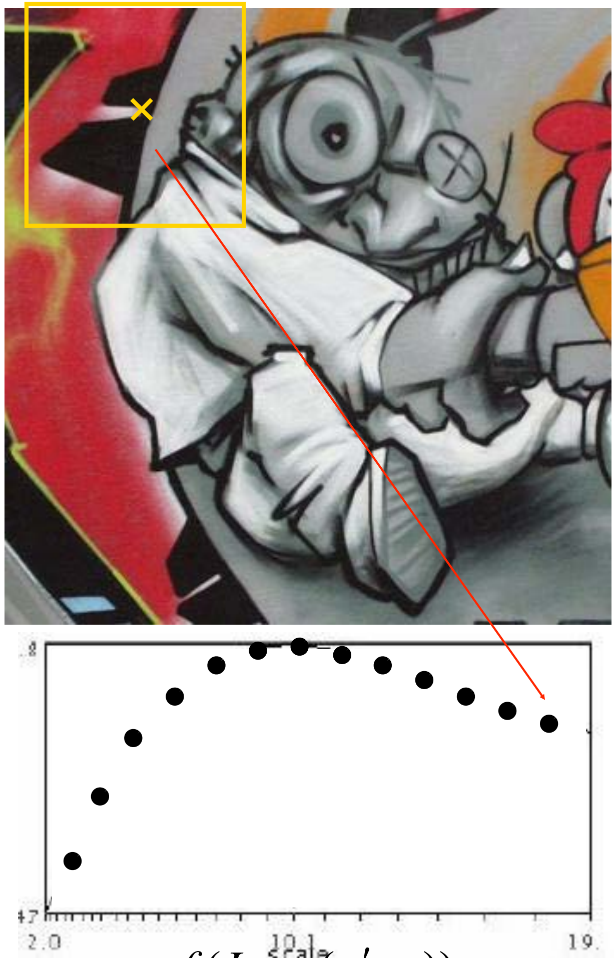
$$f(I_{i_1 \ldots i_m}(x', \sigma))$$

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)



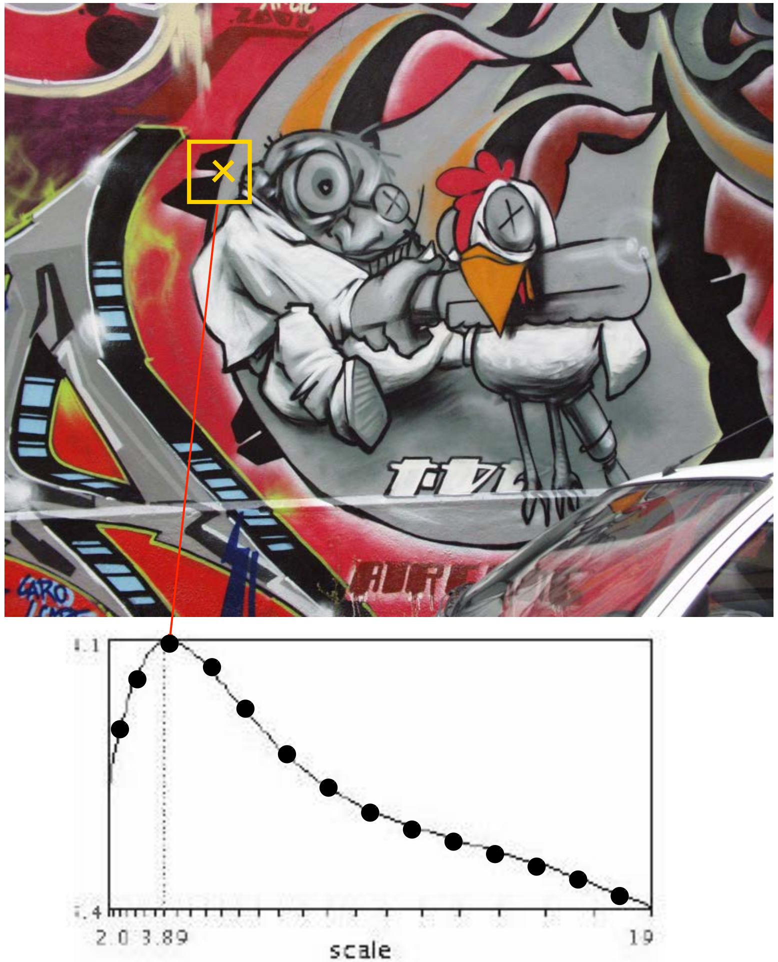
$f(I_{i_1 \ldots i_m}(x,\sigma))$



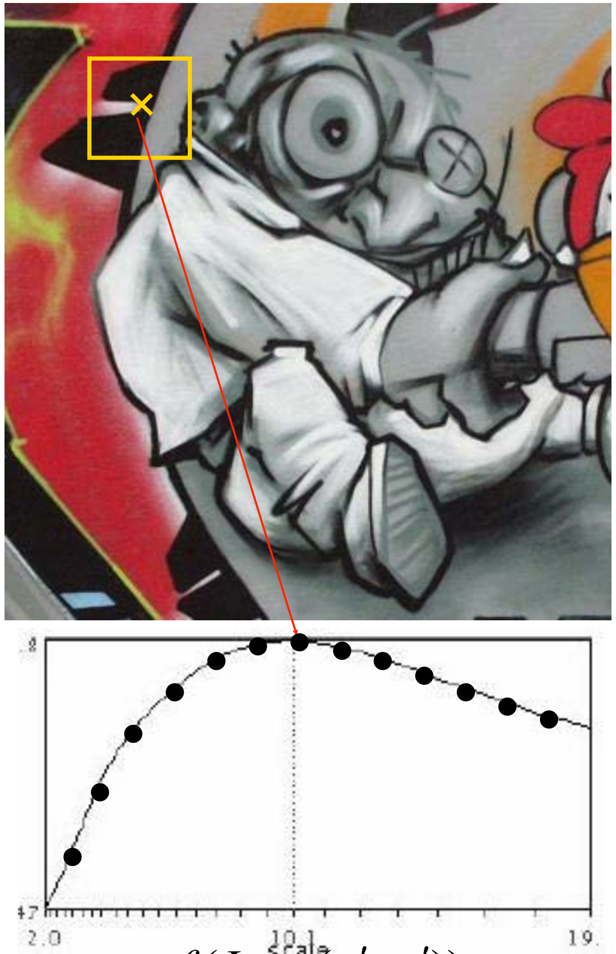
$f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)



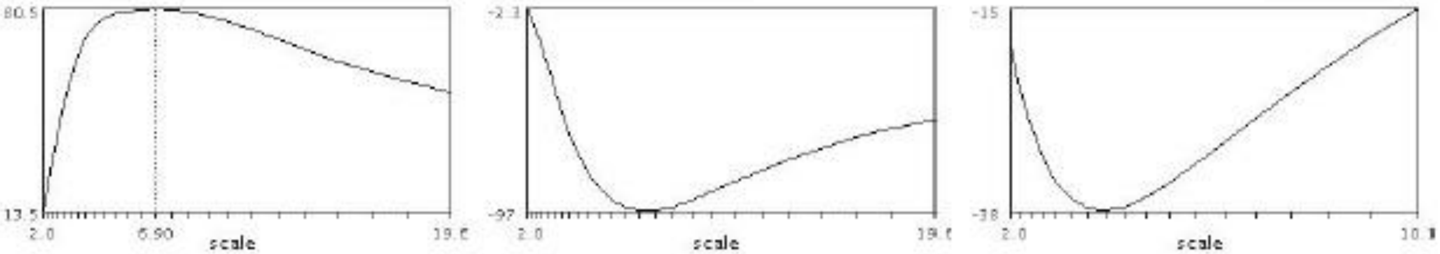
$f(I_{i_1 \ldots i_m}(x, \sigma))$



$f(I_{i_1 \ldots i_m}(x', \sigma'))$

Automatic scale selection

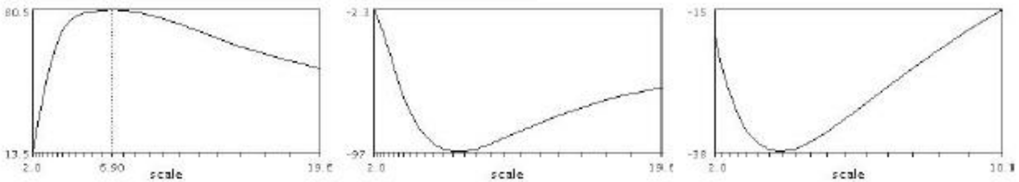
$$\text{Laplacian} = \text{trace}(\mathcal{H})$$



$$f(I_{i_1 \ldots i_m}(x,\sigma)) = L_{xx} + L_{yy}$$

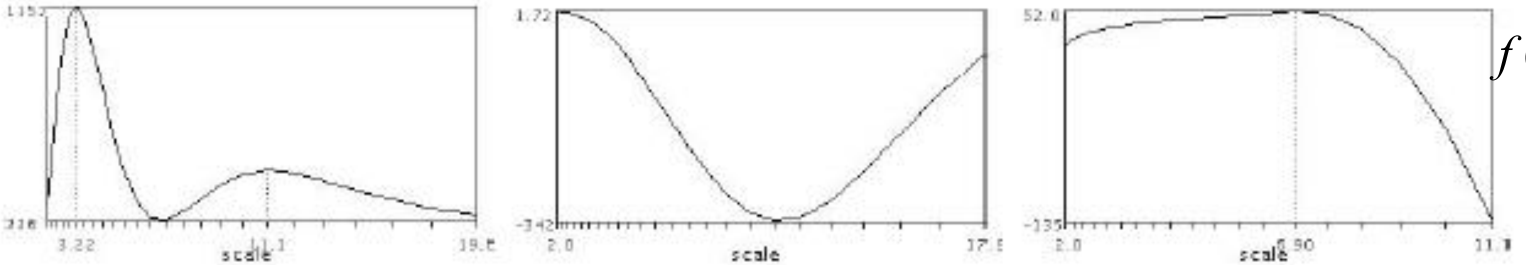
Automatic scale selection

Laplacian = trace($\mathcal{H}$)



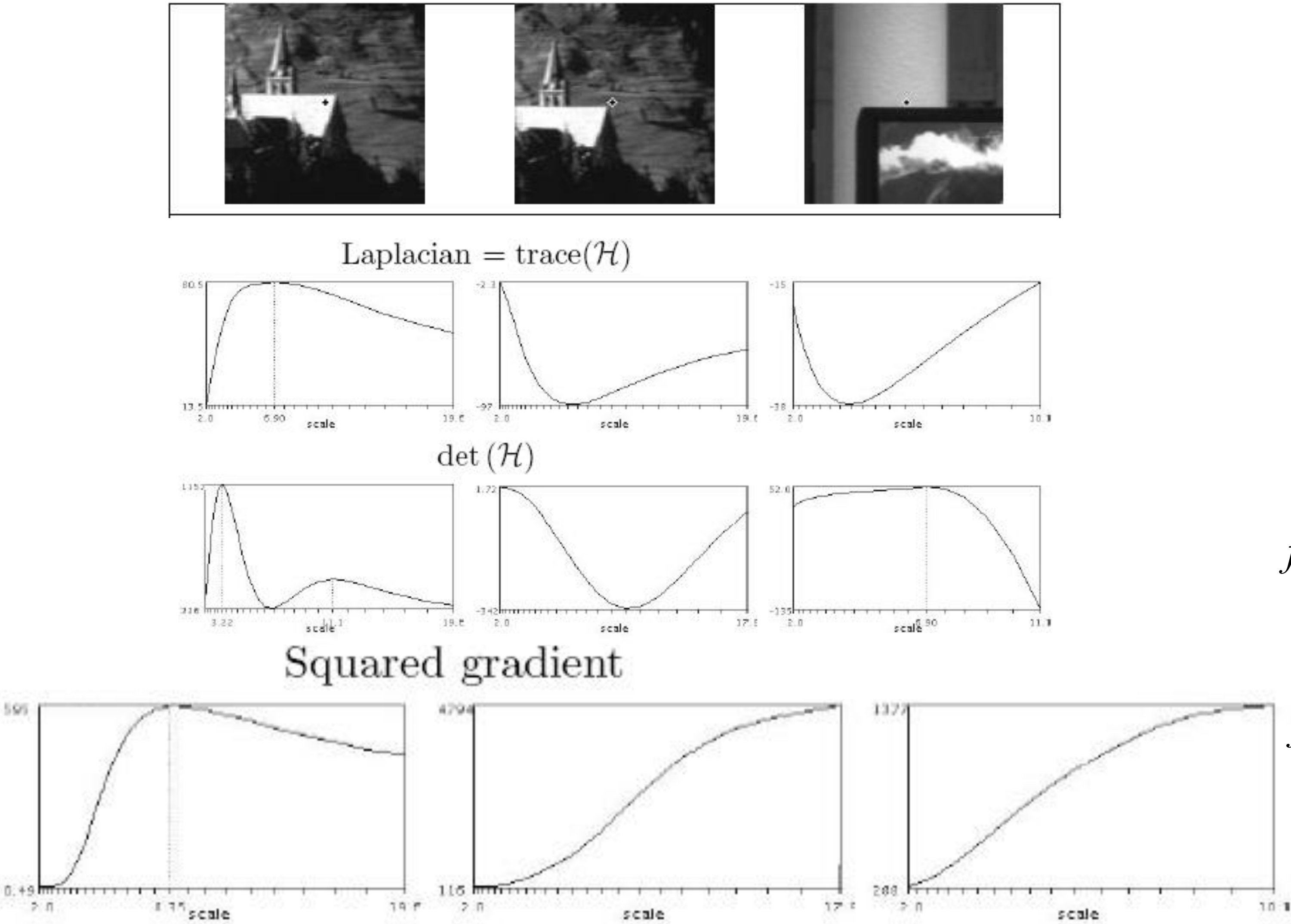
$$f(I_{i_1 \ldots i_m}(x,\sigma)) = L_{xx} + L_{yy}$$

det ($\mathcal{H}$)



$$f(I_{i_1 \ldots i_m}(x,\sigma)) = L_{xx}L_{yy} - L_{xy}^2$$

Automatic scale selection

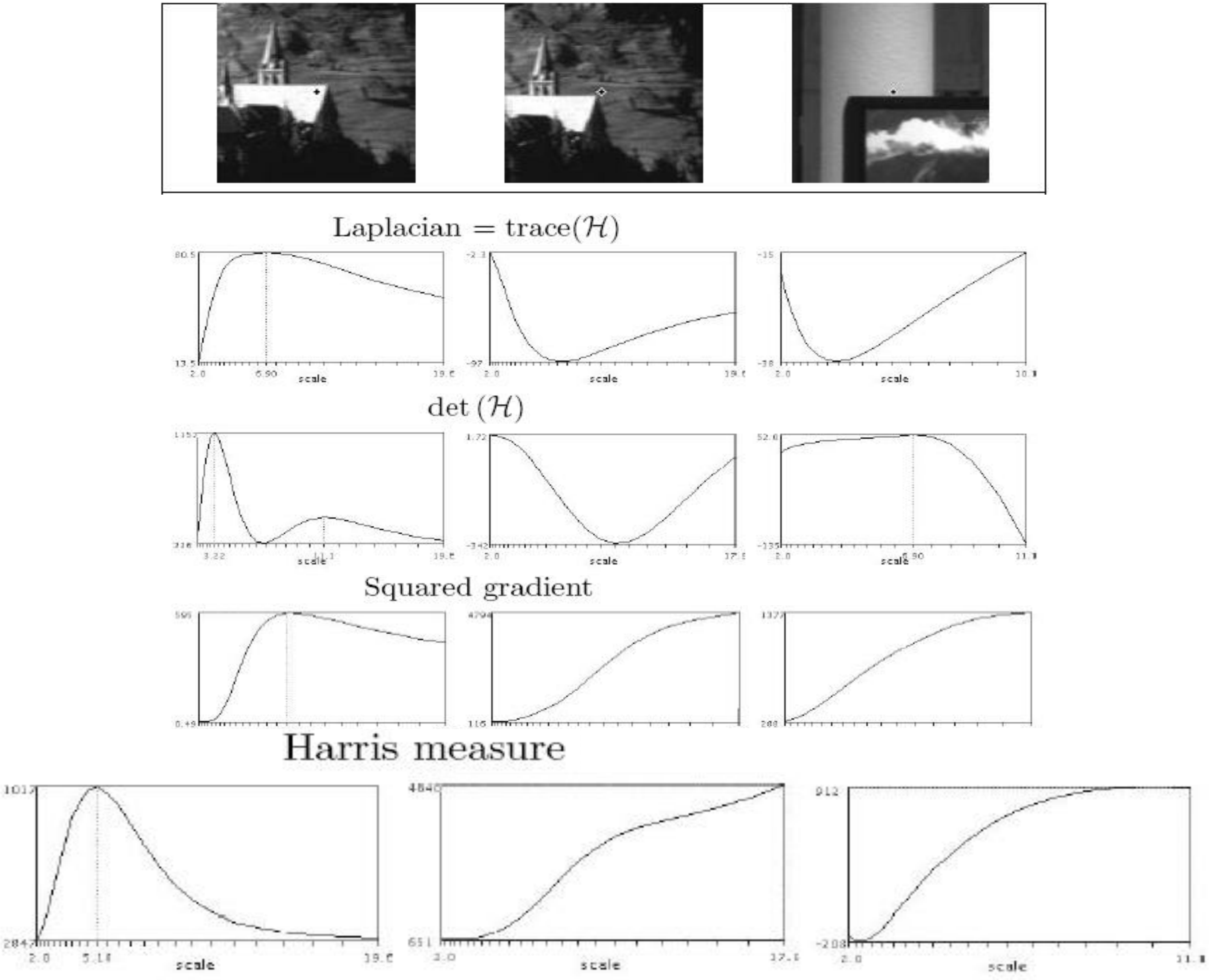


$$f(I_{i_1 \ldots i_m}(x,\sigma)) = L_{xx} + L_{yy}$$

$$f(I_{i_1 \ldots i_m}(x,\sigma)) = L_{xx}L_{yy} - L_{xy}^2$$

$$f(I_{i_1 \ldots i_m}(x,\sigma)) = L_x^2 + L_y^2$$

Automatic scale selection



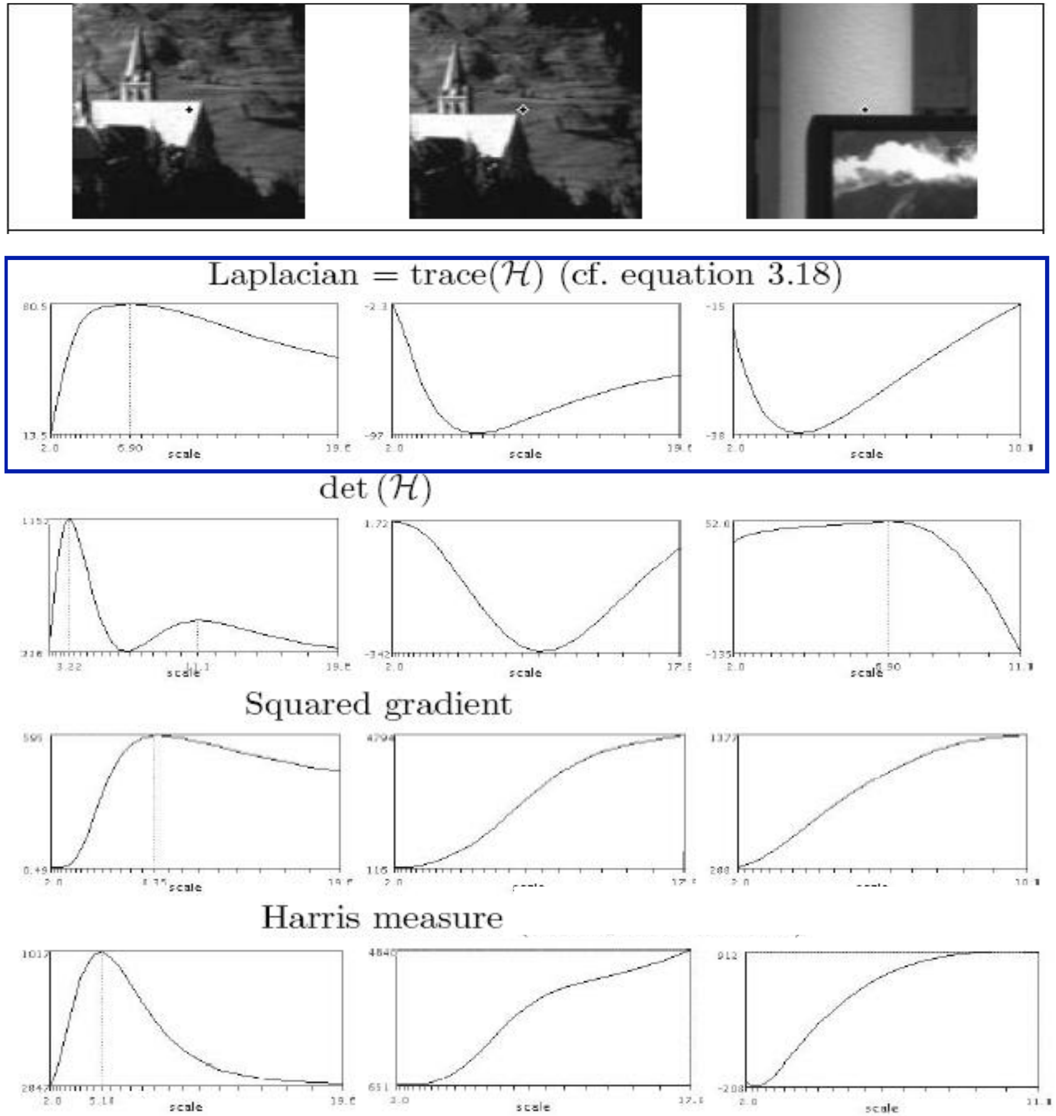
$$f(I_{i_1 \ldots i_m}(x,\sigma)) = L_{xx} + L_{yy}$$

$$f(I_{i_1 \ldots i_m}(x,\sigma)) = L_{xx}L_{yy} - L_{xy}^2$$

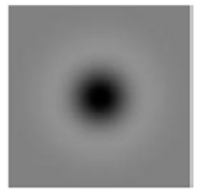
$$f(I_{i_1 \ldots i_m}(x,\sigma)) = L_x^2 + L_y^2$$

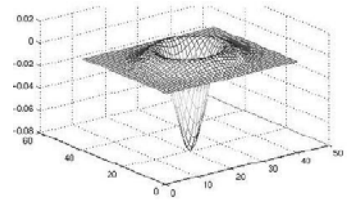
$$f(I_{i_1 \ldots i_m}(x,\sigma)) = \det(\mu) - \alpha \operatorname{trace}^2(\mu)$$

Automatic scale selection



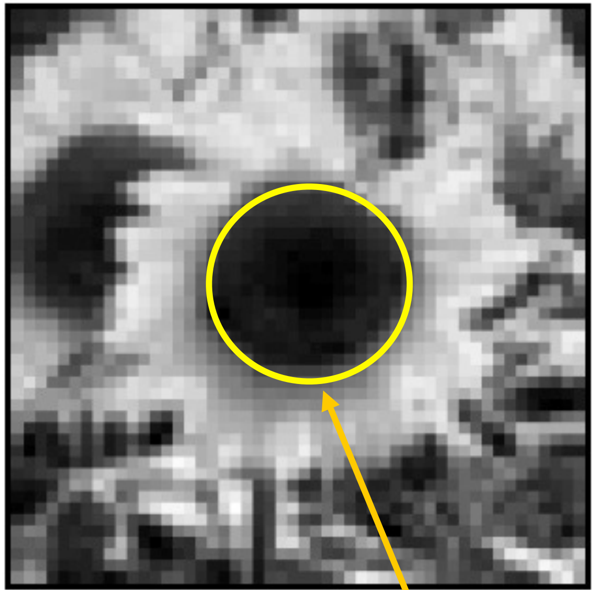
$$f(I_{i_1\ldots i_m}(x,\sigma)) = L_{xx} + L_{yy}$$

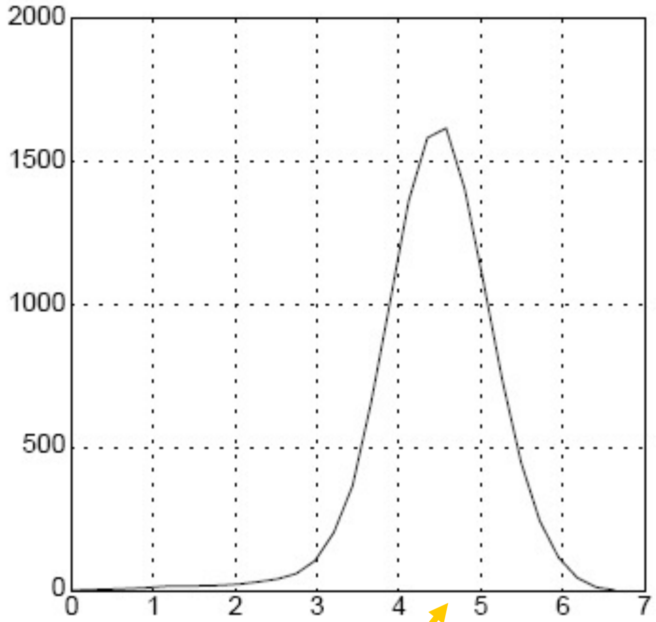




1. Interest Point Detection
Characteristic Scale

- We define the ***characteristic scale*** as the scale that produces peak of Laplacian response





Characteristic scale

T. Lindeberg (1998). "Feature detection with automatic scale selection." ***International Journal of Computer Vision*** 30 (2): pp 77--116.

Slide credit: Svetlana Lazebnik

Scale invariant detectors
Laplacian of Gaussian

- Local maxima in scale space of Laplacian of Gaussian LoG

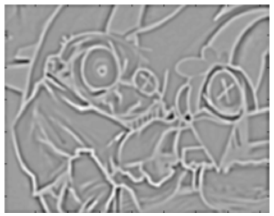
$$L_{xx}(\sigma) + L_{yy}(\sigma)$$

Scale invariant detectors
Laplacian of Gaussian

- Local maxima in scale space of Laplacian of Gaussian LoG

$$L_{xx}(\sigma) + L_{yy}(\sigma)$$

σ



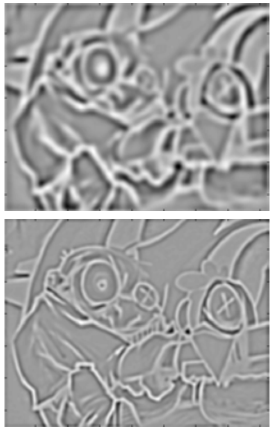
Scale invariant detectors
Laplacian of Gaussian

- Local maxima in scale space of Laplacian of Gaussian LoG

$$L_{xx}(\sigma) + L_{yy}(\sigma)$$

σ^2

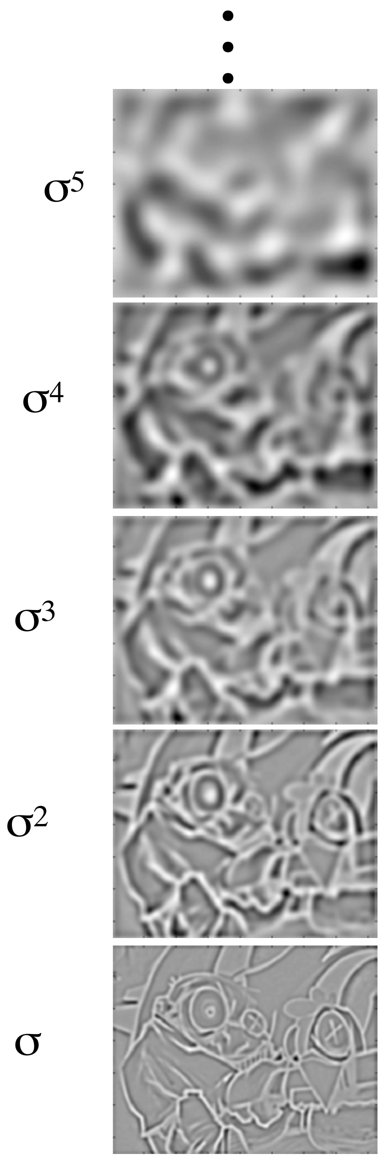
σ



Scale invariant detectors
Laplacian of Gaussian

- Local maxima in scale space of Laplacian of Gaussian LoG

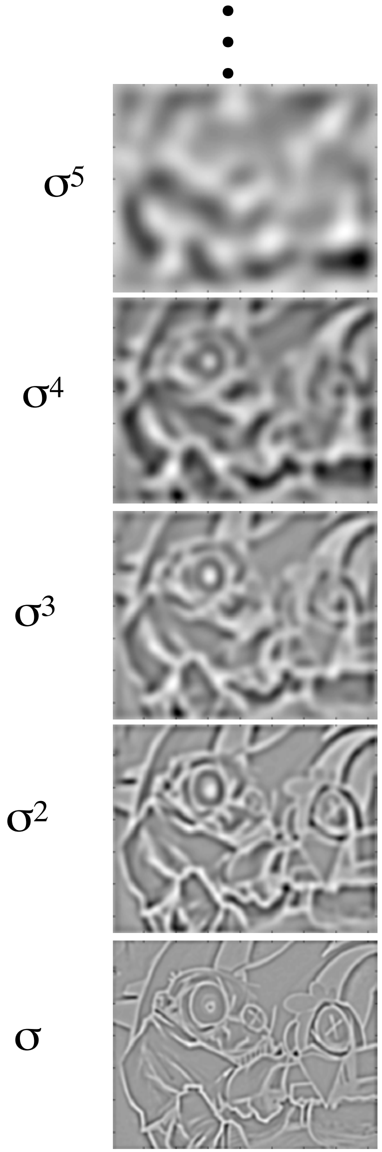
$$L_{xx}(\sigma) + L_{yy}(\sigma)$$

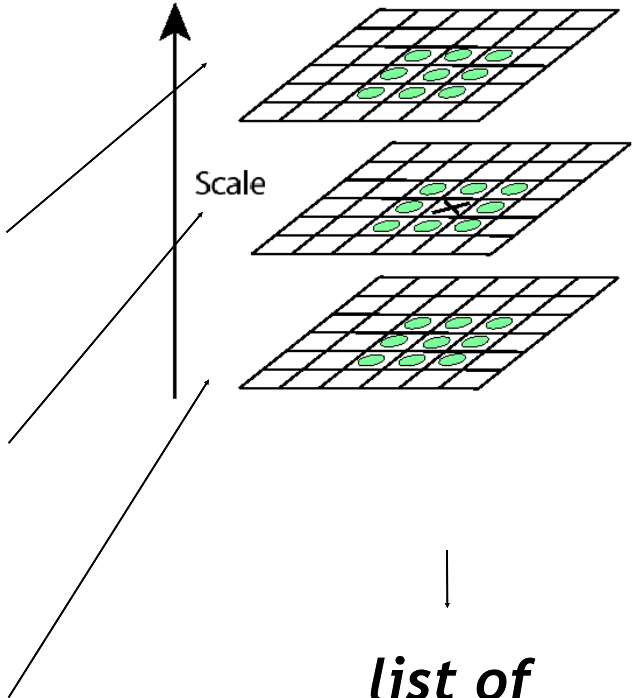


Scale invariant detectors
Laplacian of Gaussian

- Local maxima in scale space of Laplacian of Gaussian LoG

$$L_{xx}(\sigma) + L_{yy}(\sigma)$$



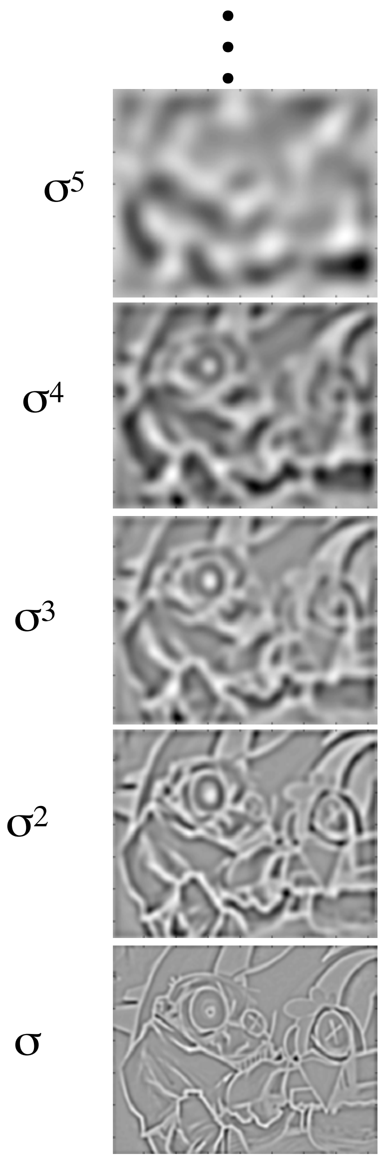


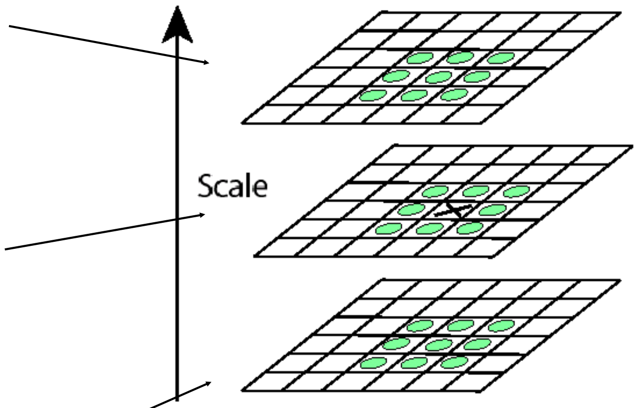
list of
(*x*, *y*, σ)

Scale invariant detectors
Laplacian of Gaussian

- Local maxima in scale space of Laplacian of Gaussian LoG

$$L_{xx}(\sigma) + L_{yy}(\sigma)$$



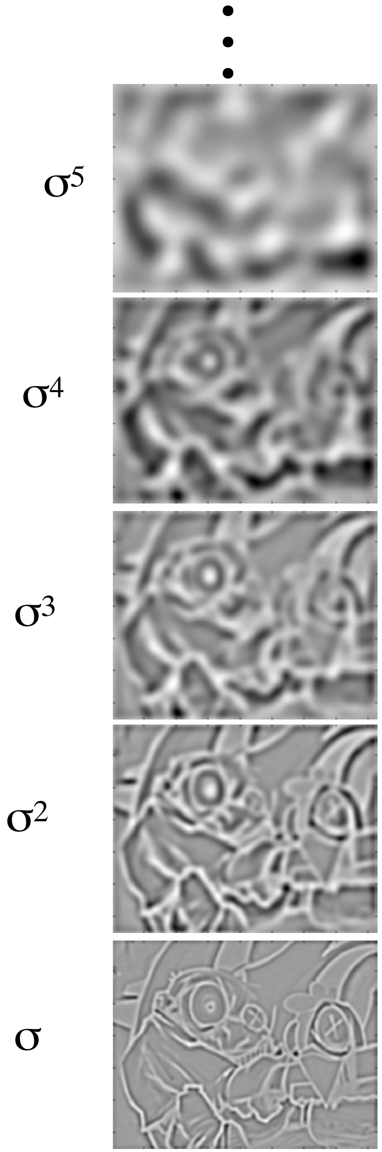


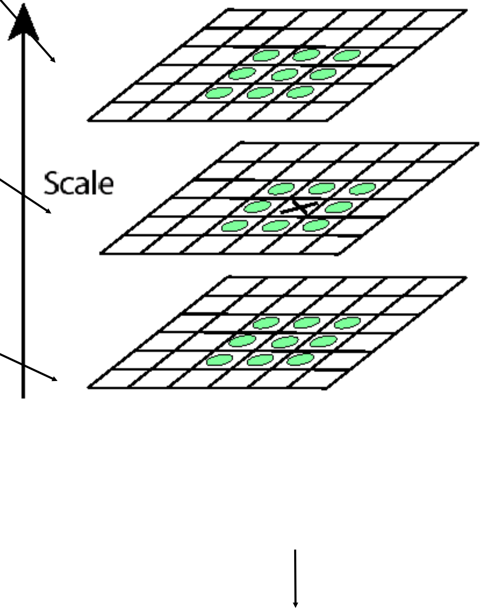
list of
$(\boldsymbol{x}, \boldsymbol{y}, \sigma)$

Scale invariant detectors
Laplacian of Gaussian

- Local maxima in scale space of Laplacian of Gaussian LoG

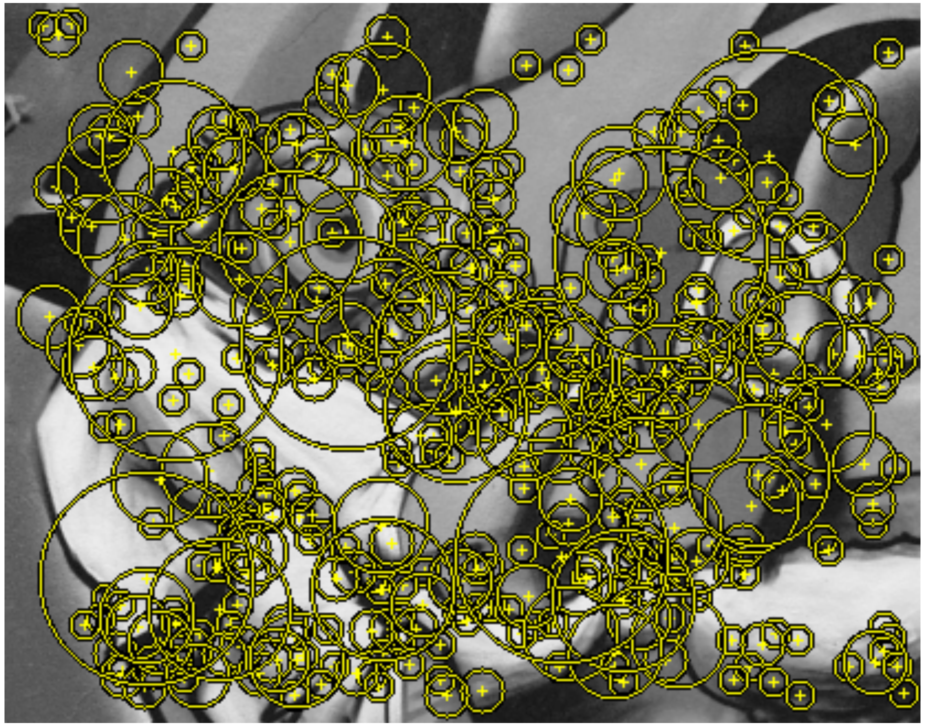
$$L_{xx}(\sigma) + L_{yy}(\sigma)$$





list of
(***x***, ***y***, σ)

Scale invariant detectors
Laplacian of Gaussian



Scale invariant detectors
Harris-Laplace (HarLap)

- Detecting multiscale
  Harris points

$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

$$har = \det[\mu(\sigma_I,\sigma_D)] - \alpha[\mathrm{trace}(\mu(\sigma_I,\sigma_D))]$$

$$\sigma_I = 1.6 \cdot \sigma_D$$

Scale invariant detectors
Harris-Laplace (HarLap)

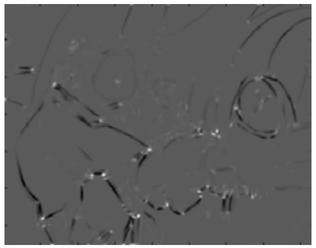
- Detecting multiscale Harris points

$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_xI_y(\sigma_D) \\ I_xI_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

$$har = \det[\mu(\sigma_I,\sigma_D)] - \alpha[\mathrm{trace}(\mu(\sigma_I,\sigma_D))]$$

$$\sigma_I = 1.6 \cdot \sigma_D$$

σ



Computing Harris function

Scale invariant detectors
Harris-Laplace (HarLap)

- Detecting multiscale
  Harris points

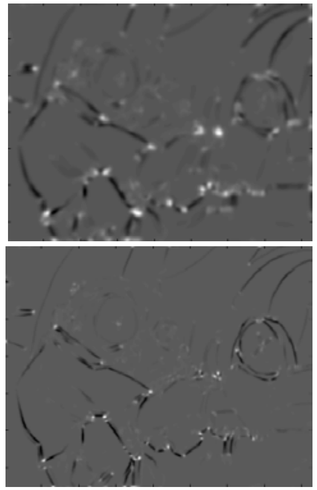
$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

$$har = \det[\mu(\sigma_I, \sigma_D)] - \alpha[\mathrm{trace}(\mu(\sigma_I, \sigma_D))]$$

$$\sigma_I = 1.6 \cdot \sigma_D$$

σ^2

σ



Computing Harris function

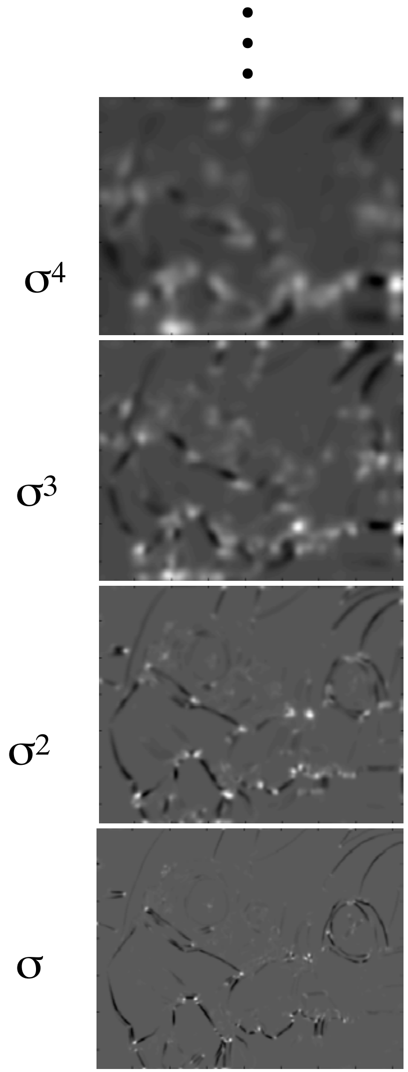
Scale invariant detectors
Harris-Laplace (HarLap)

- Detecting multiscale Harris points – thousands of interest points

$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

$$har = \det[\mu(\sigma_I,\sigma_D)] - \alpha[\mathrm{trace}(\mu(\sigma_I,\sigma_D))]$$

$$\sigma_I = 1.6 \cdot \sigma_D$$



Computing Harris function

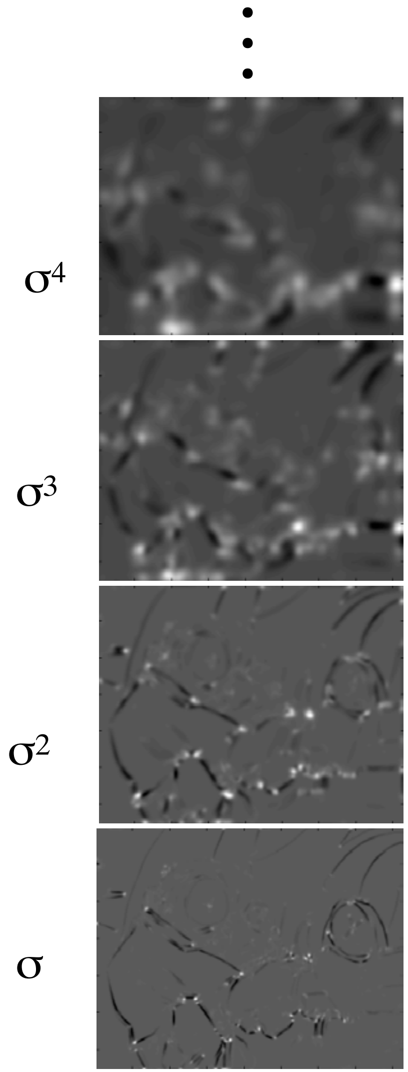
Scale invariant detectors
Harris-Laplace (HarLap)

- Detecting multiscale Harris points – thousands of interest points

$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

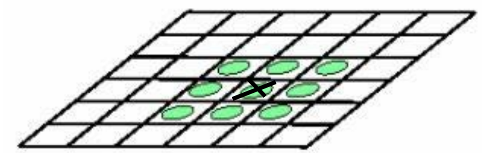
$$har = \det[\mu(\sigma_I, \sigma_D)] - \alpha[\mathrm{trace}(\mu(\sigma_I, \sigma_D))]$$

$$\sigma_I = 1.6 \cdot \sigma_D$$



Computing Harris function

list of
$(\boldsymbol{x}, \boldsymbol{y}, \sigma)$



Detecting local maxima

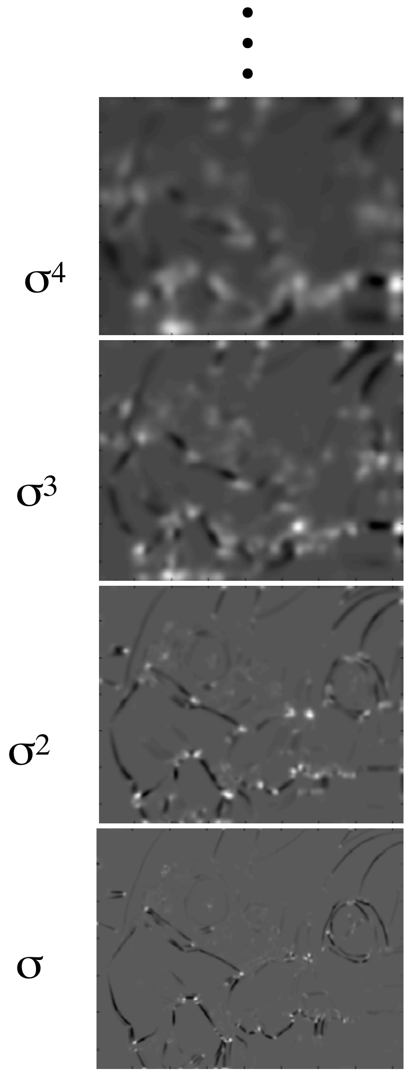
Scale invariant detectors Harris-Laplace (HarLap)

- Detecting multiscale Harris points – thousands of interest points

$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

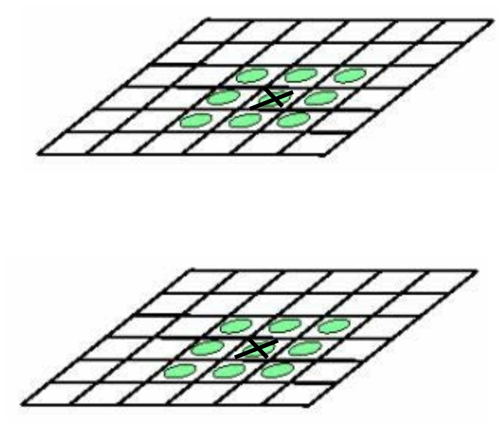
$$har = \det[\mu(\sigma_I,\sigma_D)] - \alpha[\mathrm{trace}(\mu(\sigma_I,\sigma_D))]$$

$$\sigma_I = 1.6 \cdot \sigma_D$$



Computing Harris function

list of
(*x*, *y*, σ)



Detecting local maxima

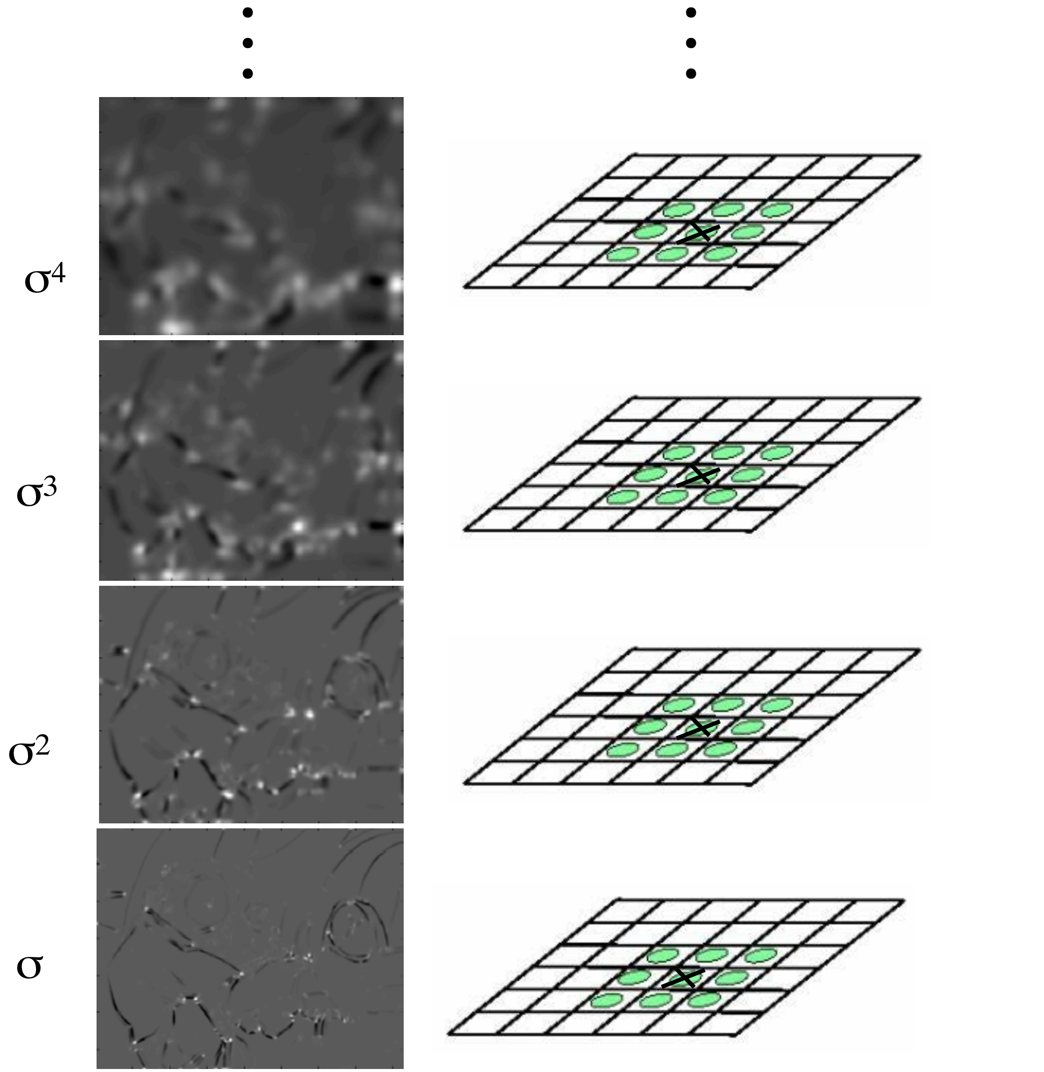
Scale invariant detectors
Harris-Laplace (HarLap)

- Detecting multiscale Harris points – thousands of interest points

$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

$$har = \det[\mu(\sigma_I, \sigma_D)] - \alpha[\mathrm{trace}(\mu(\sigma_I, \sigma_D))]$$

$$\sigma_I = 1.6 \cdot \sigma_D$$



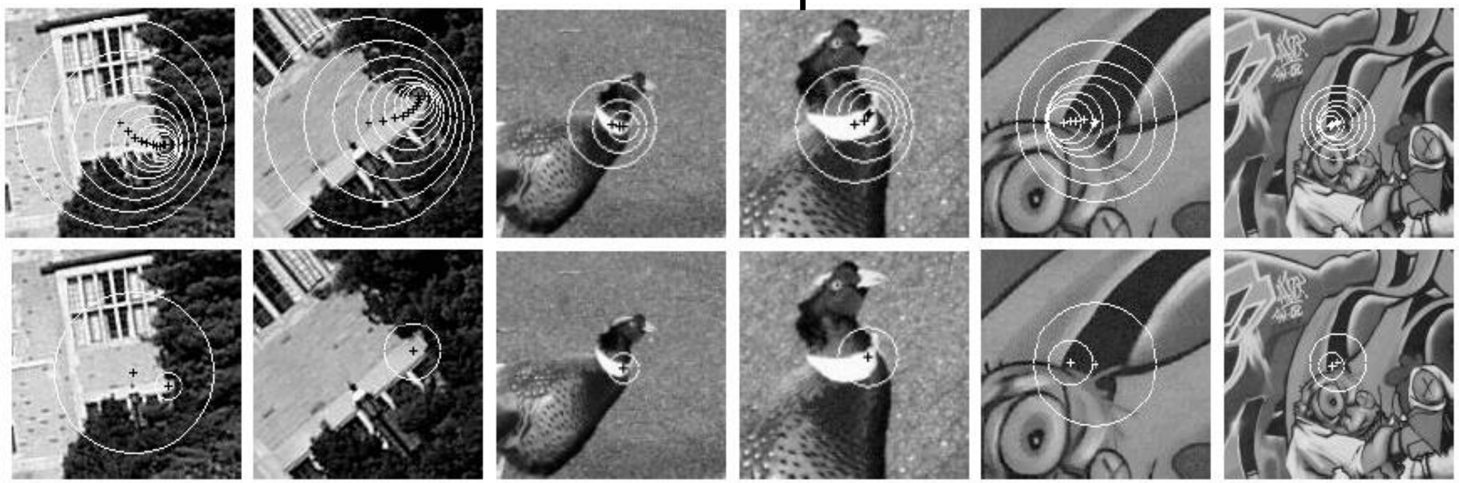
Computing Harris function

Detecting local maxima

Scale invariant detectors
Harris-Laplace (HarLap)

- Detecting multiscale Harris points
- Selecting points which maximize the Laplacian
  - Automatic scale selection
    - Given a point (x, y, σ_n)
    - If $(L_{xx}(\sigma_n) + L_{yy}(\sigma_n)) > (L_{xx}(\sigma_{n-1}) + L_{yy}(\sigma_{n-1}))$ and
      $(L_{xx}(\sigma_n) + L_{yy}(\sigma_n)) > (L_{xx}(\sigma_{n+1}) + L_{yy}(\sigma_{n+1}))$ keep the point,
      otherwise reject

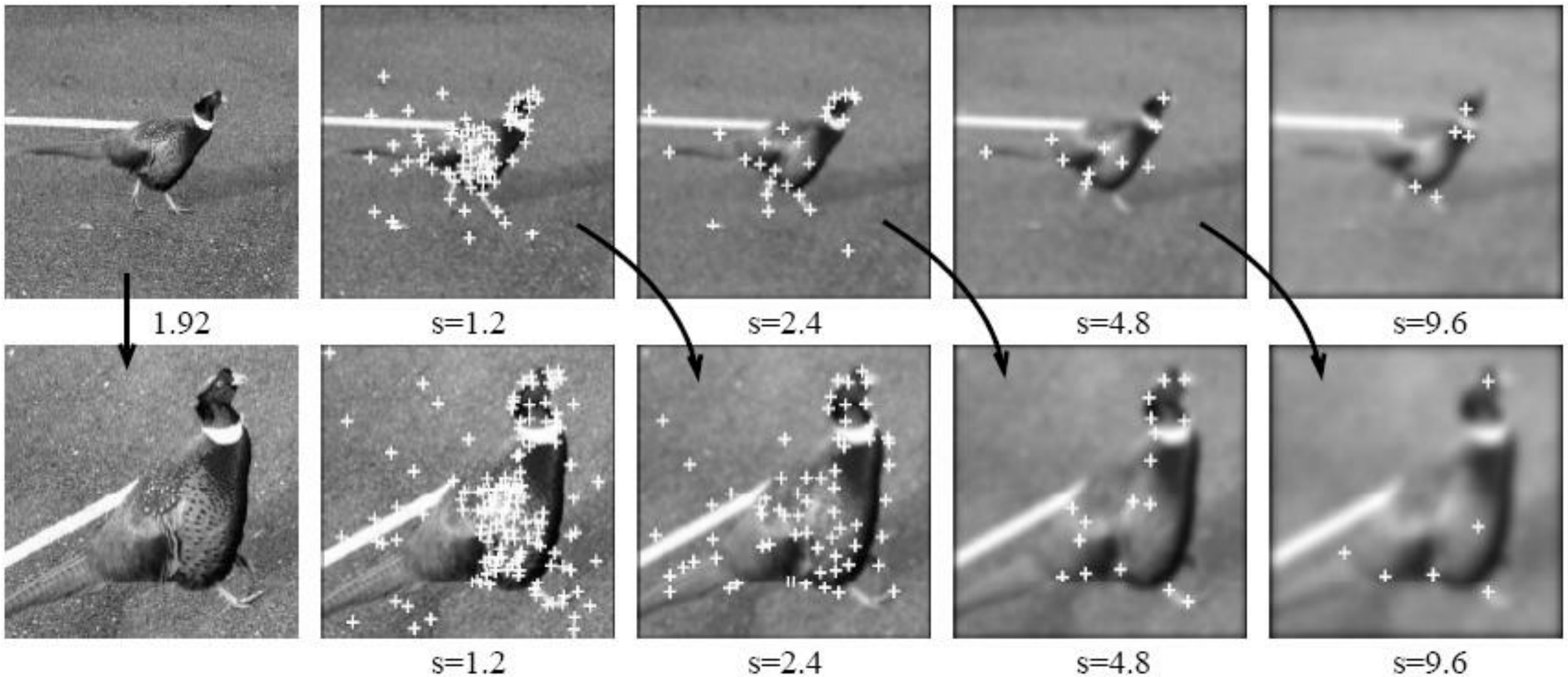
Harris points



Harris-Laplace points

Scale invariant detectors
Harris-Laplace

- Detecting multiscale Harris points
- Selecting Harris points which maximize the Laplacian
- Automatic scale selection



Harris-Laplace points

Local Descriptors / Features

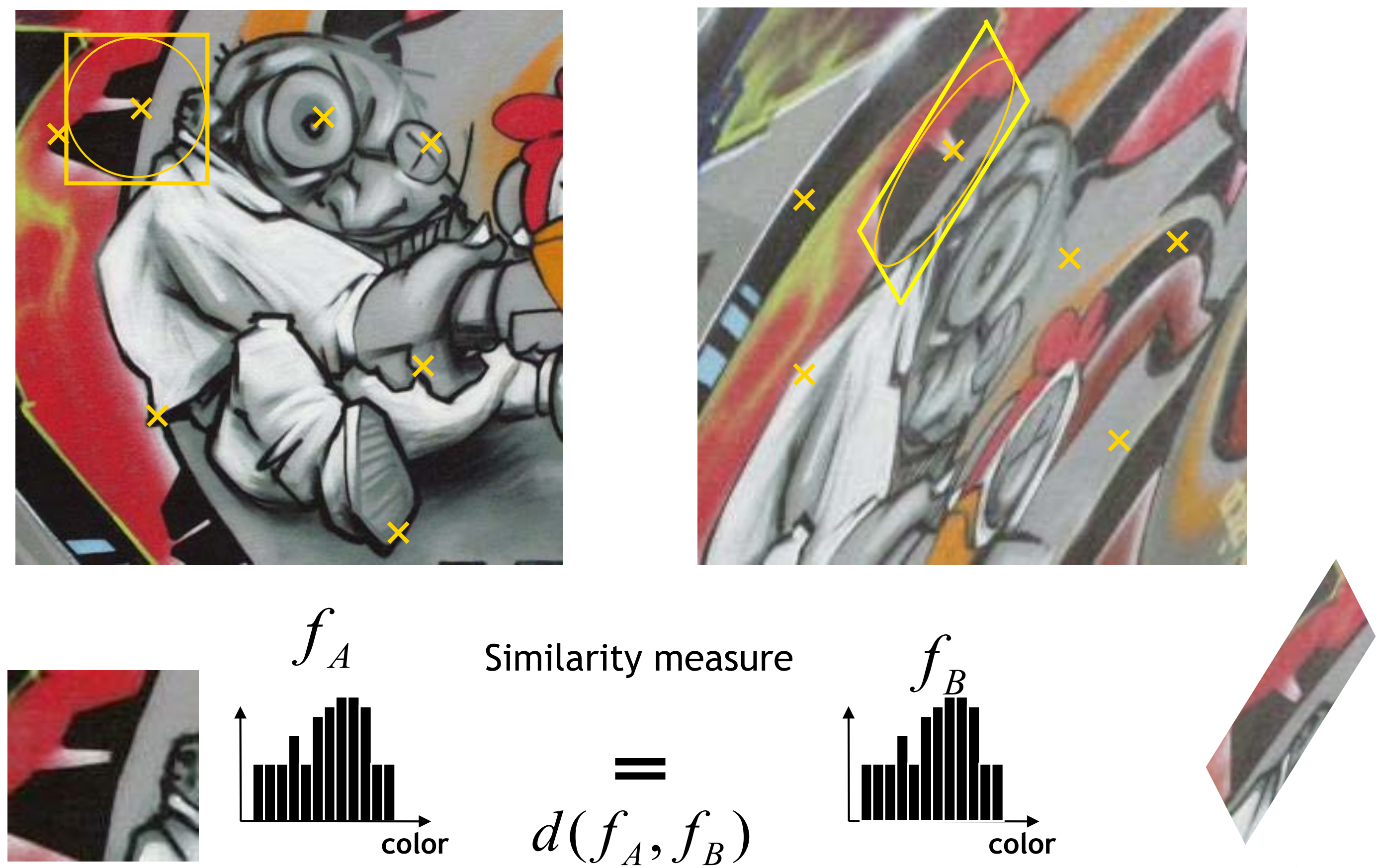
- Important properties of local descriptors
  - Distinctiveness, invariance, robustness, dimensionality, etc...

- Local descriptors
  - Differential invariants, steerable filters, complex filters
  - PCA,
  - Moment invariants,
  - Shape context,
  - Gradient orientation histogram

- Evaluation criteria

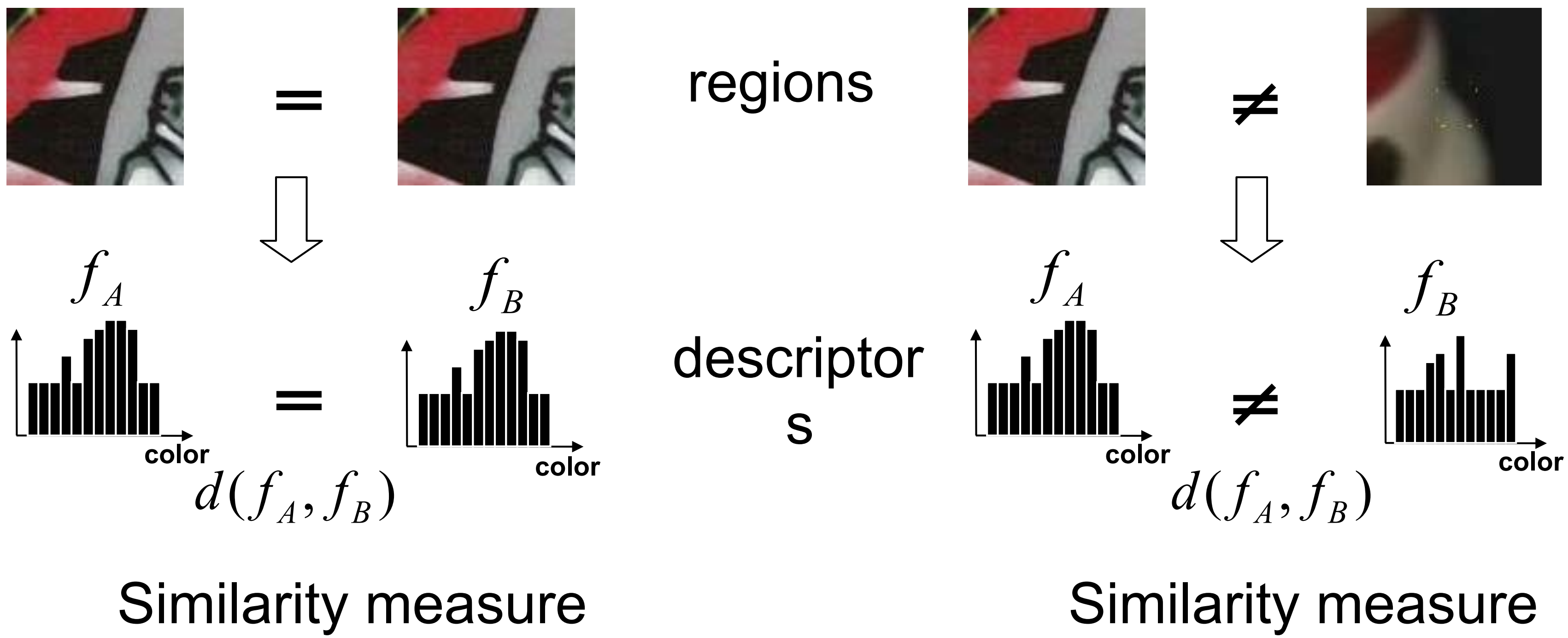
Local descriptors

- Detector finds location, scale and shape of interest regions
- Local descriptors are computed for interest regions



Local descriptors

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors



Local descriptors

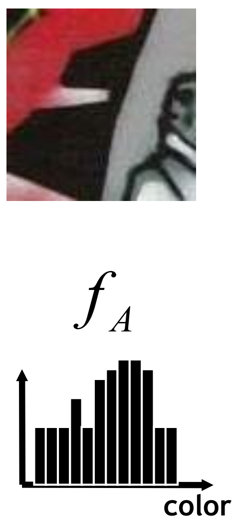
- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness

Two ways to obtain invariance

Local descriptors

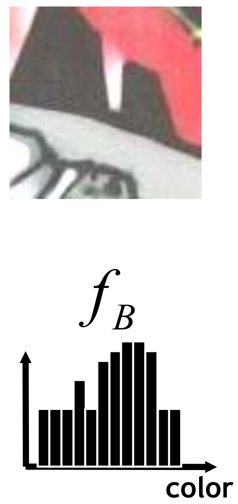
- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness

Two ways to obtain invariance



=

=

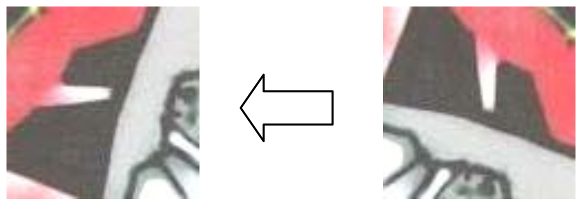


1. Computing invariant descriptor

Local descriptors

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness

Two ways to obtain invariance

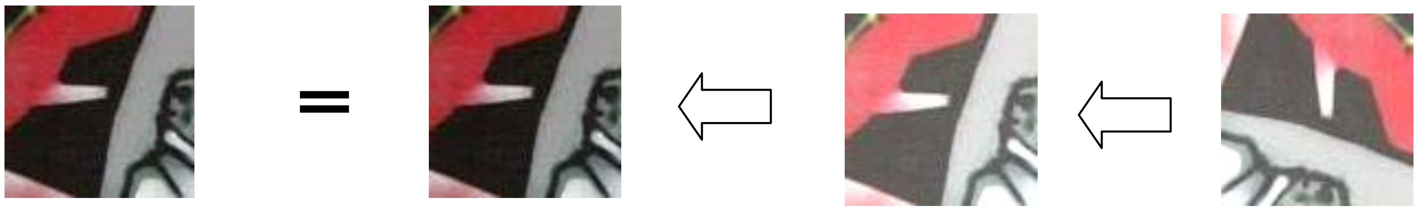


2. **Geometric normalization**

Local descriptors

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness

Two ways to obtain invariance

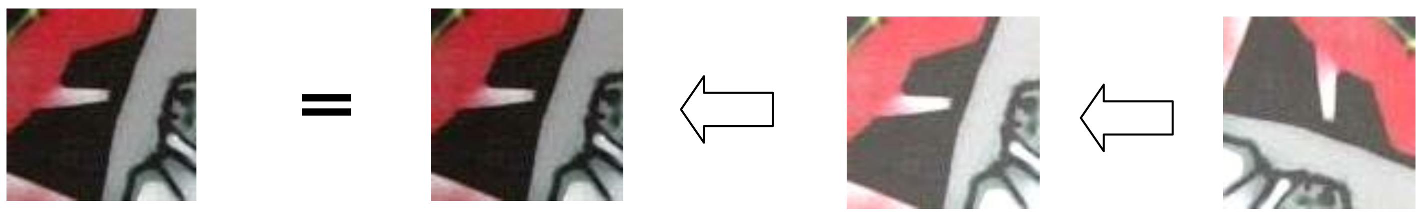


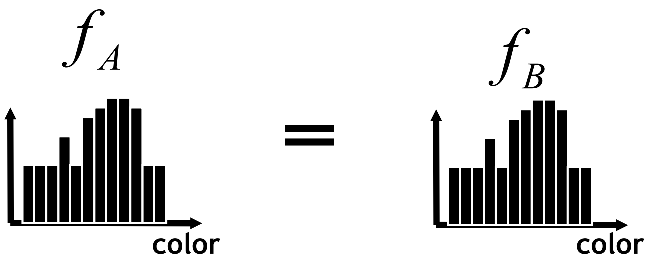
2. **Geometric normalization + photometric normalization**

Local descriptors

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness

Two ways to obtain invariance

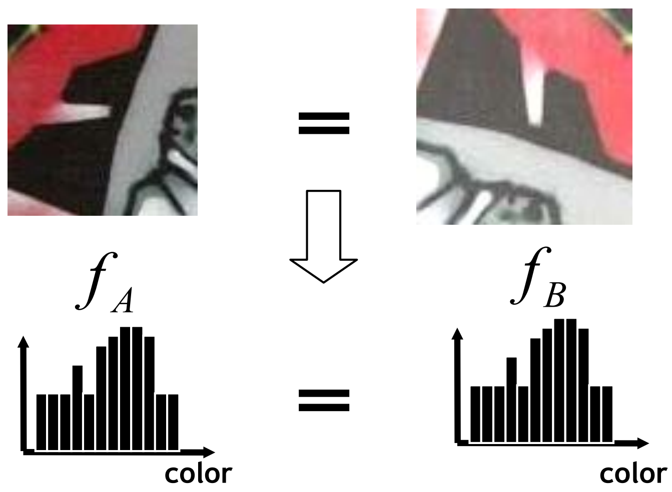




2. **Geometric normalization + photometric normalization + computing descriptors**

Local descriptors

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness
  - Robustness
    - Visually similar regions should have similar descriptors despite the noise (geometric, photometric)



Local descriptors

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness
  - Robustness
    - Visually similar regions should have similar descriptors despite the noise (geometric, photometric)
  - Dimensionality
    - Descriptors should be low dimensional i.e., small number of histogram bins.
      - Efficiency (large databases)
      - Generalization property

Local Descriptors / Features

- Important properties of local descriptors
  - Distinctiveness, invariance, robustness, dimensionality, etc...

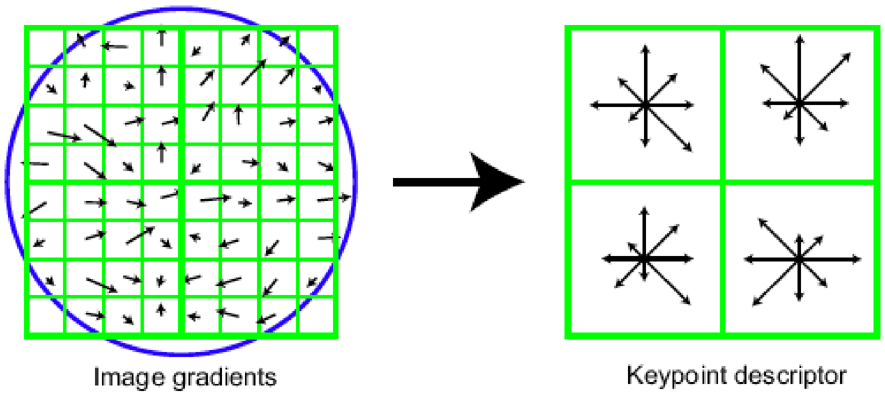
- Local descriptors
  - Differential invariants, steerable filters, complex filters
  - PCA,
  - Moment invariants,
  - Shape context,
  - Gradient orientation histogram

- Evaluation criteria

SIFT - Scale Invariant Feature Transform [Lowe]

- Interest Points:
  - Difference of Gaussians

- Feature Descriptor:
  - local histogram of 4x4 local orientation histograms (each over 16x16 pixels),
    - 8 orientations x 4 x 4 = 128 dimensions
  - example: 2x2 local orientation histogram (each of 4x4 pixels):



Local Descriptors

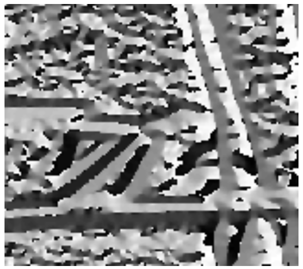
- Gradient location-orientation histogram (GLOH)
  - Invariant – only when computed on normalized patches

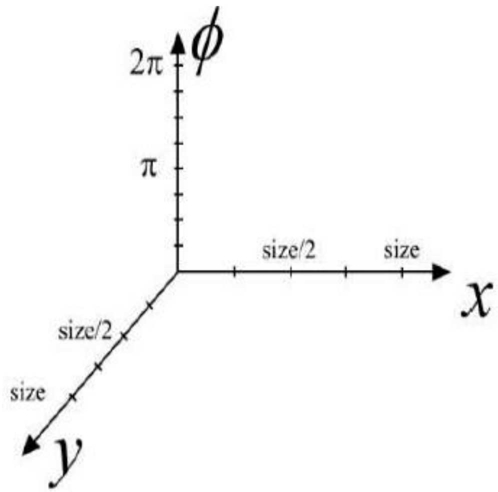
derivatives

$L_x \; L_y$

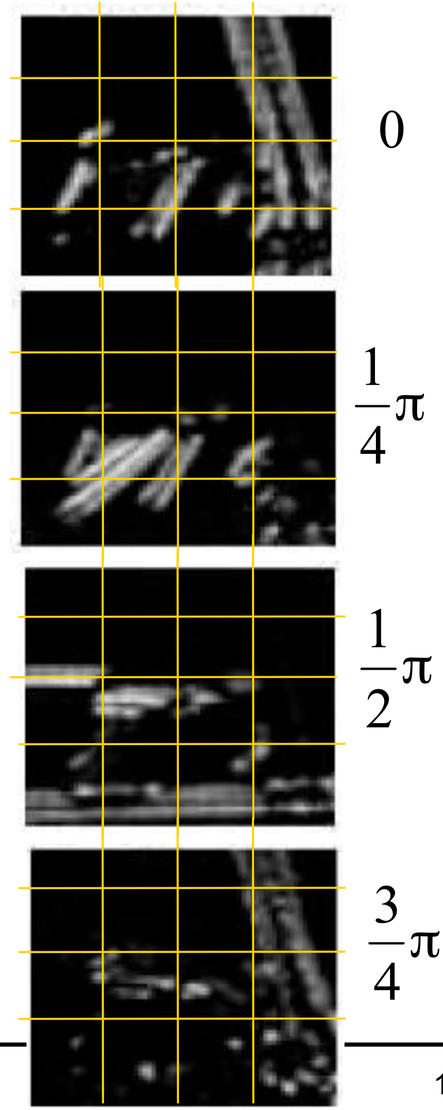
magnitude

orientation





Histogram of gradient locations and orientations

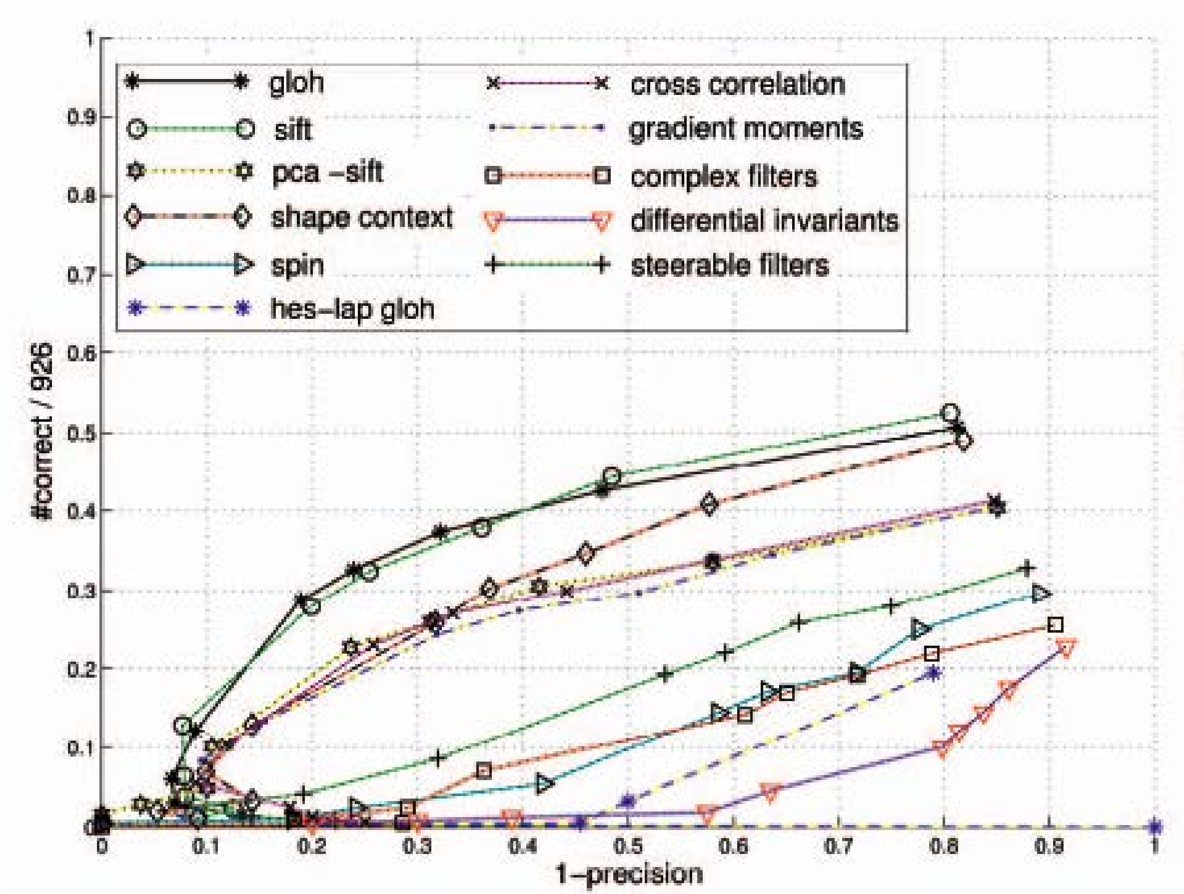


Evaluation & Comparison

- Sample Images
  - (Mikolajczyk & Schmid, PAMI o5)
  - (a,b) rotation
  - (c,d) zoom & rotation
  - (e,f) viewpoint
  - (g) blur
  - (i) JPEG
  - (j) light change

Sample Results for viewpoint changes (e)

- Interest Points: Hes-Affine
- Image Descriptors: varied...
- Nearest Neighbor Matching
  - best:
    - GLOH
    - SIFT
  - second:
    - Shape Context
    - cross correlation
    - PCA-SIFT
  - not so good:
    - steerable filters
    - spin
    - gradient moments
    - differential invariants



- similar results for other test images (scale, blur, ...)

Local Interest Points and Features

- So far talked about:
  - local interest points (Harris, Hessian)
  - local scale selection (e.g. Laplacian)
  - local features (e.g. SIFT, Shape Context)

- Application: find corresponding points
  - recognition by point correspondence
  - point correspondence for (sparse) stereo matching
  - point correspondence for (sparse) optical flow
  - point correspondence for image matching
  - ...

Wide-Baseline Stereo

Image from T. Tuytelaars ECCV 2006 tutorial

Application of Point Correspondence: Image Matching

by Diva Sian

by swashford

Slide credit: Steve Seitz

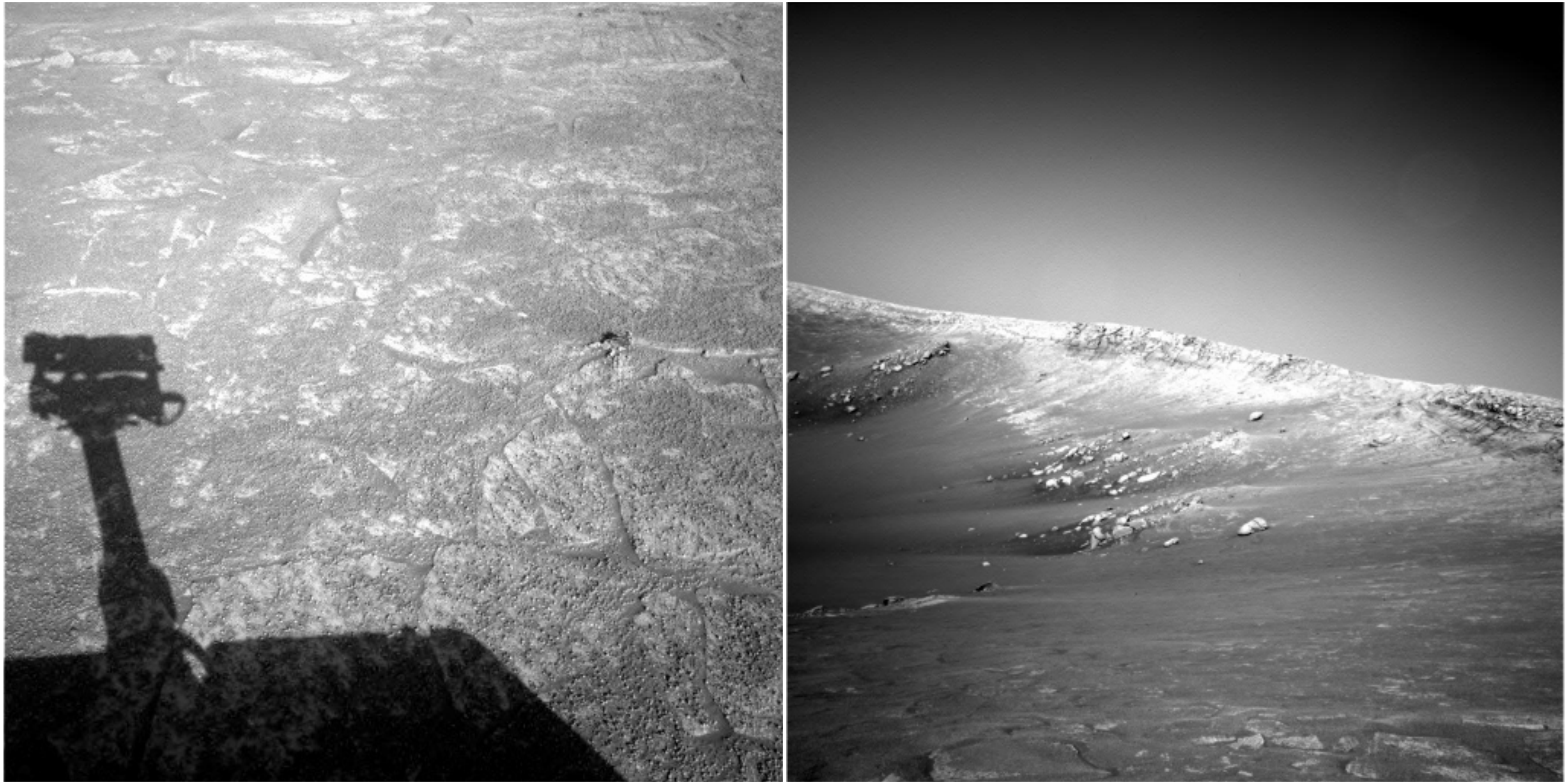
Harder Case

by [Diva Sian]

by [scgbt]

Slide credit: Steve Seitz

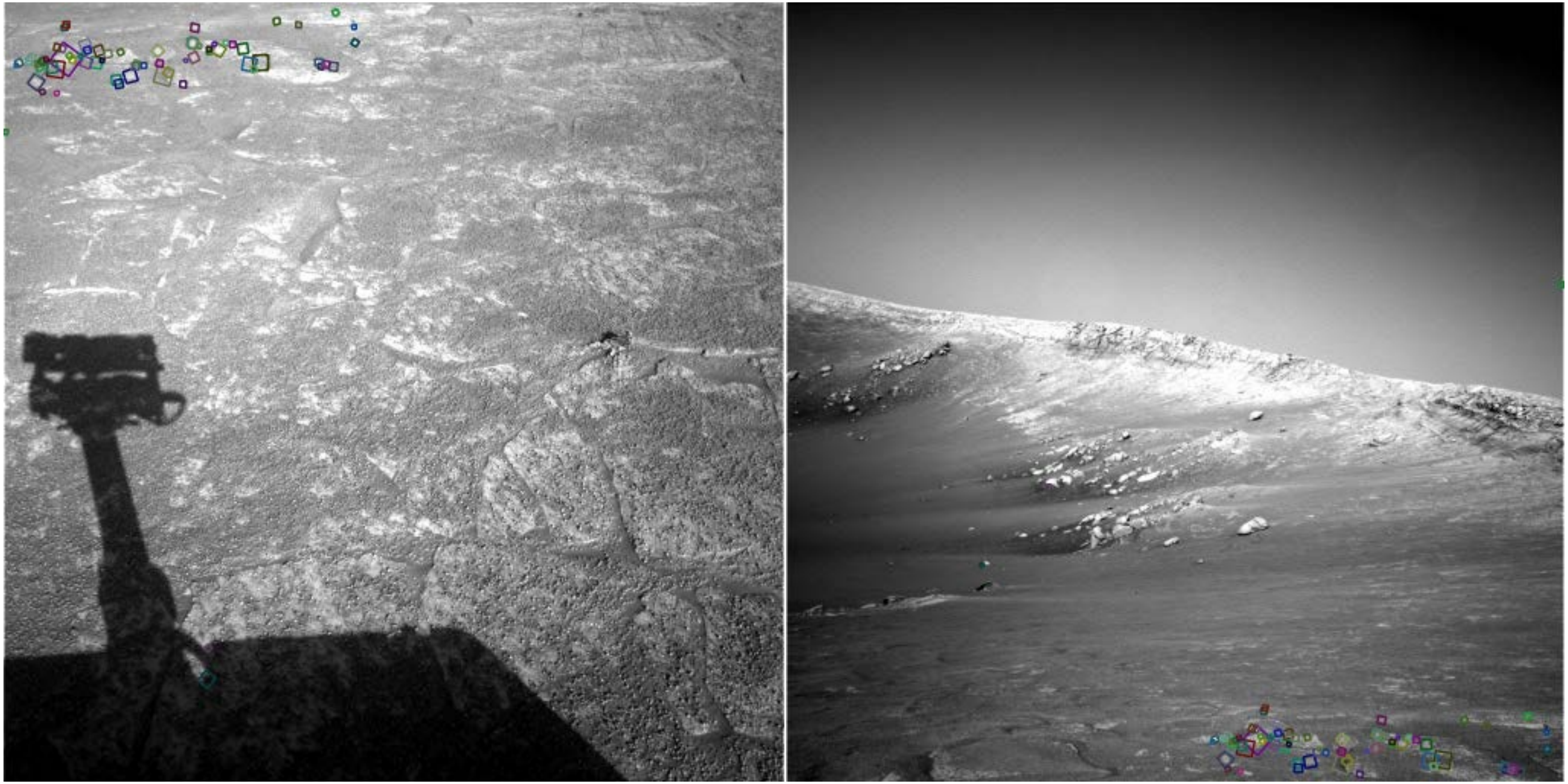
Harder Still?



NASA Mars Rover images

Slide credit: Steve Seitz

Answer Below (Look for tiny colored squares)



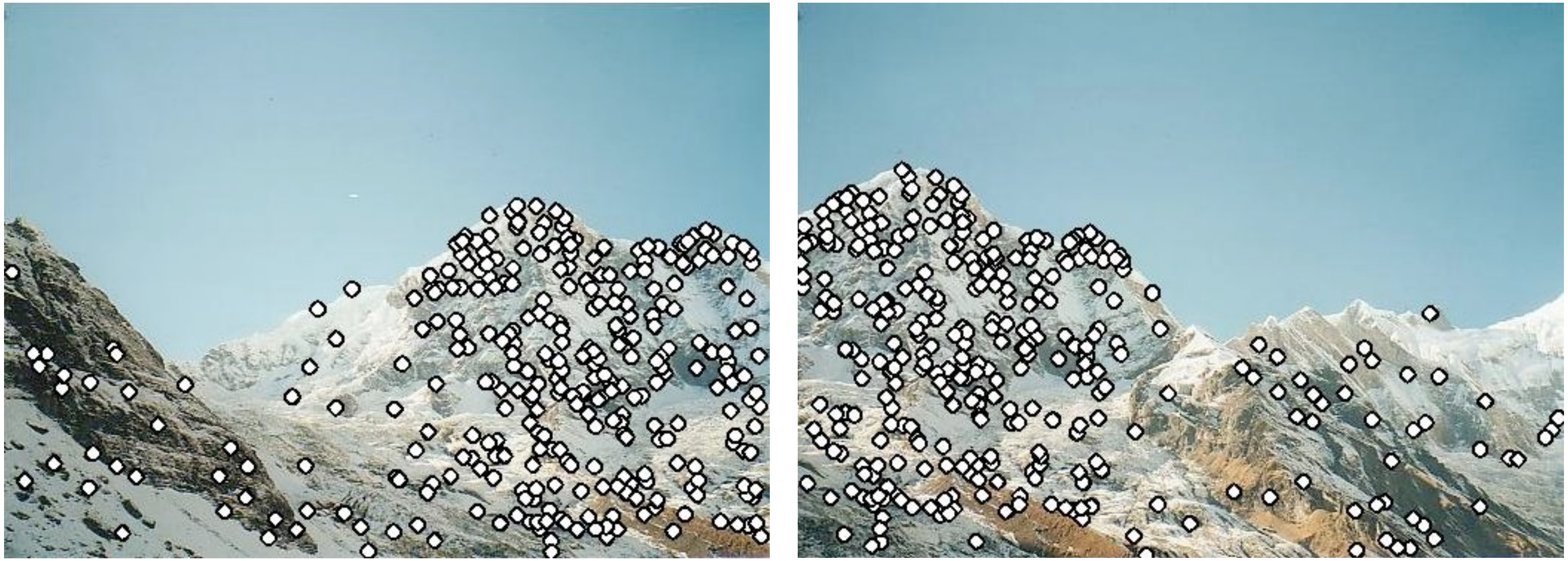
NASA Mars Rover images with SIFT feature matches
(Figure by Noah Snavely)

Slide credit: Steve Seitz

Application: Image Stitching

Slide credit: Darya Frolova, Denis Simakov

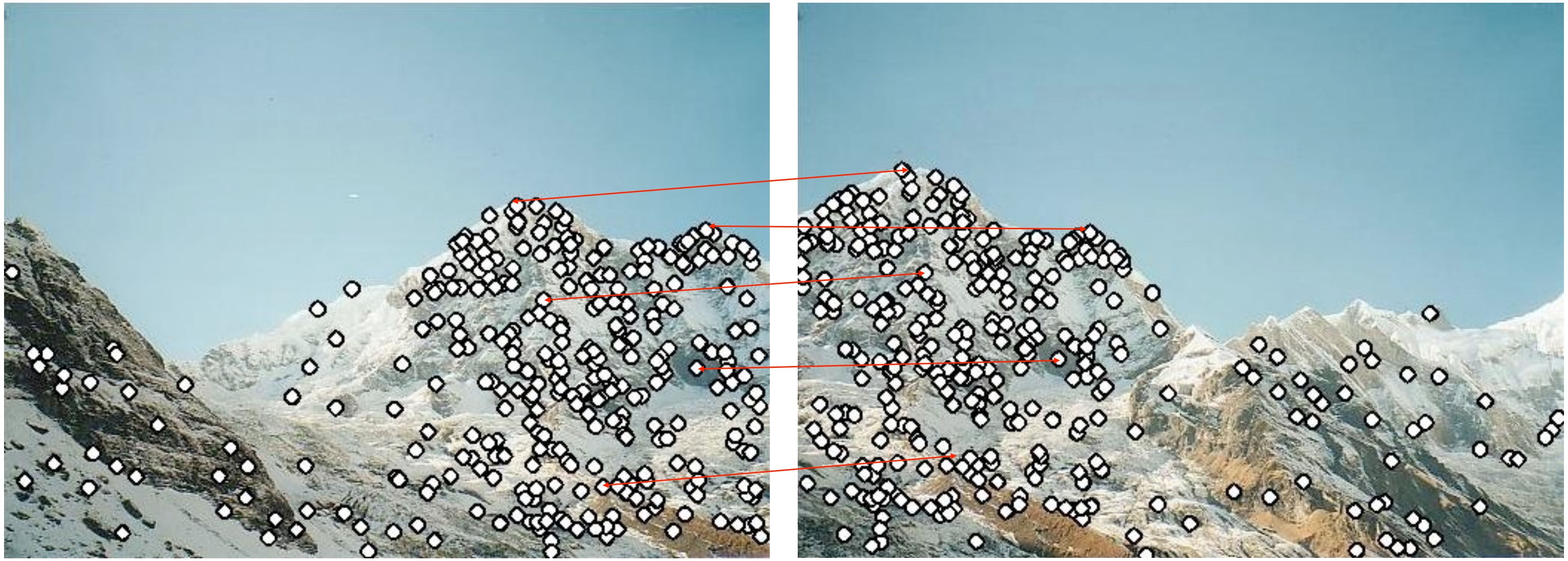
Application: Image Stitching



- Procedure:
  - Detect feature points in both images (step 1 & 2)

Slide credit: Darya Frolova, Denis Simakov

Application: Image Stitching



- Procedure:
  - Detect feature points in both images (step 1 & 2)
  - Find corresponding pairs (step 3 & 4)

Slide credit: Darya Frolova, Denis Simakov

Application: Image Stitching

- Procedure:
  - Detect feature points in both images (step 1 & 2)
  - Find corresponding pairs (step 3 & 4)
  - Use these pairs to align the images (step 5)

Slide credit: Darya Frolova, Denis Simakov

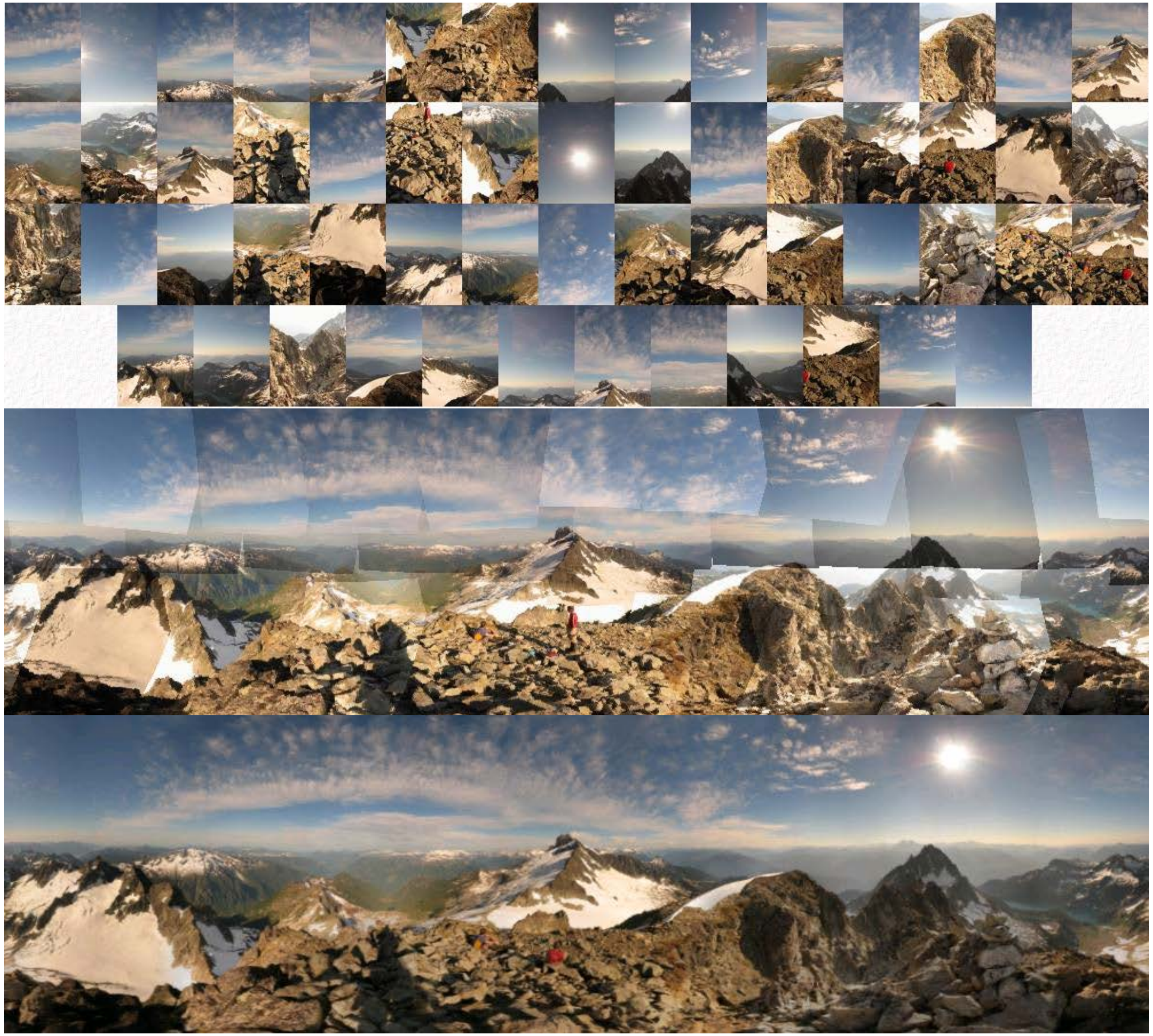
Application: Image Stitching

- Procedure:
  - Detect feature points in both images (step 1 &2)
  - Find corresponding pairs (step 3 & 4)
  - Use these pairs to align the images (step 5)

Slide credit: Darya Frolova, Denis Simakov

Automatic Mosaicing

[Brown & Lowe, ICCV'03]



Panorama Stitching

[Brown, Szeliski, and Winder, 2005]

(a) Matier data set (7 images)

(b) Matier final stitch

iPhone version available

http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

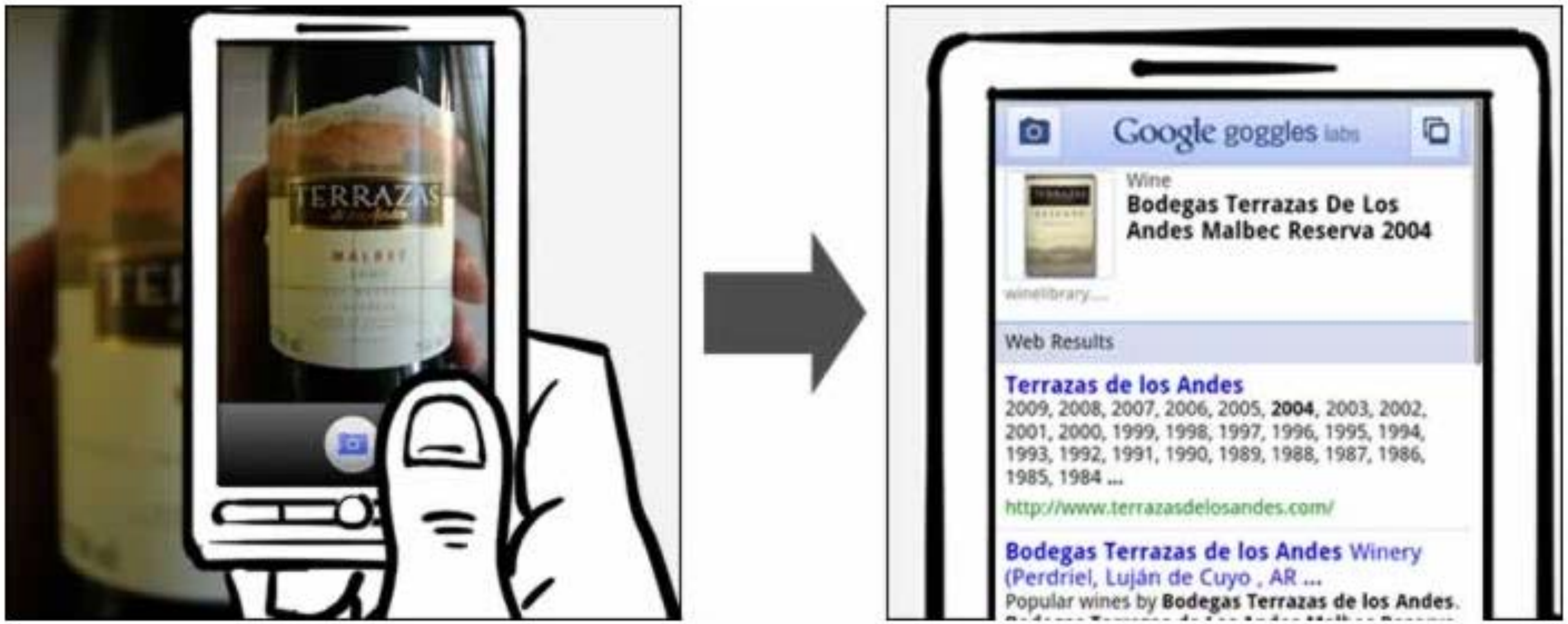
Overview Today

- Object Identification by Point Correspondences
  - general procedure for recognition, stereo, image stitching, ...
- Interest Point Detection & Descriptor
  - local interest point detection
  - scale-invariant interest point detection
  - local image descriptor
- Scaling to Large Numbers of Images and Objects
  - inverted file
  - visual vocabulary

Application: Mobile Visual Search

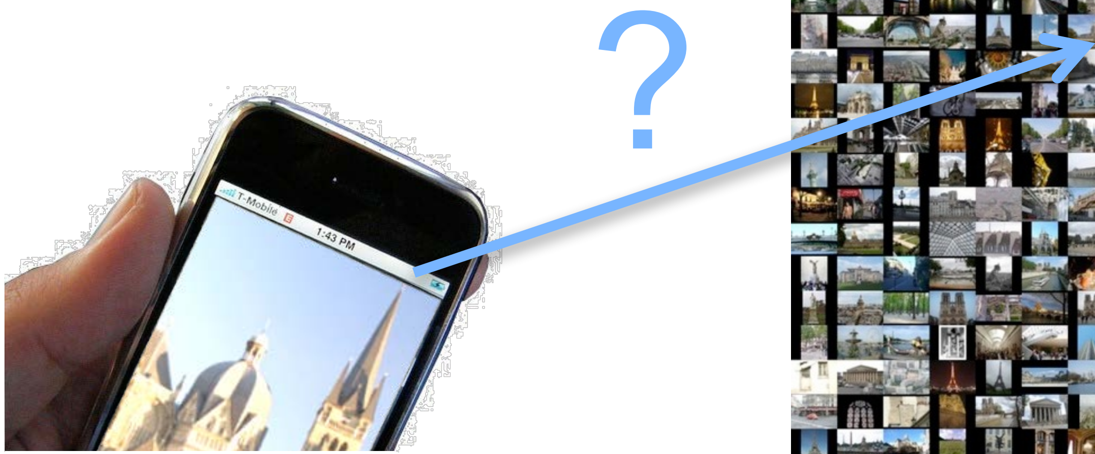
Google Goggles in Action

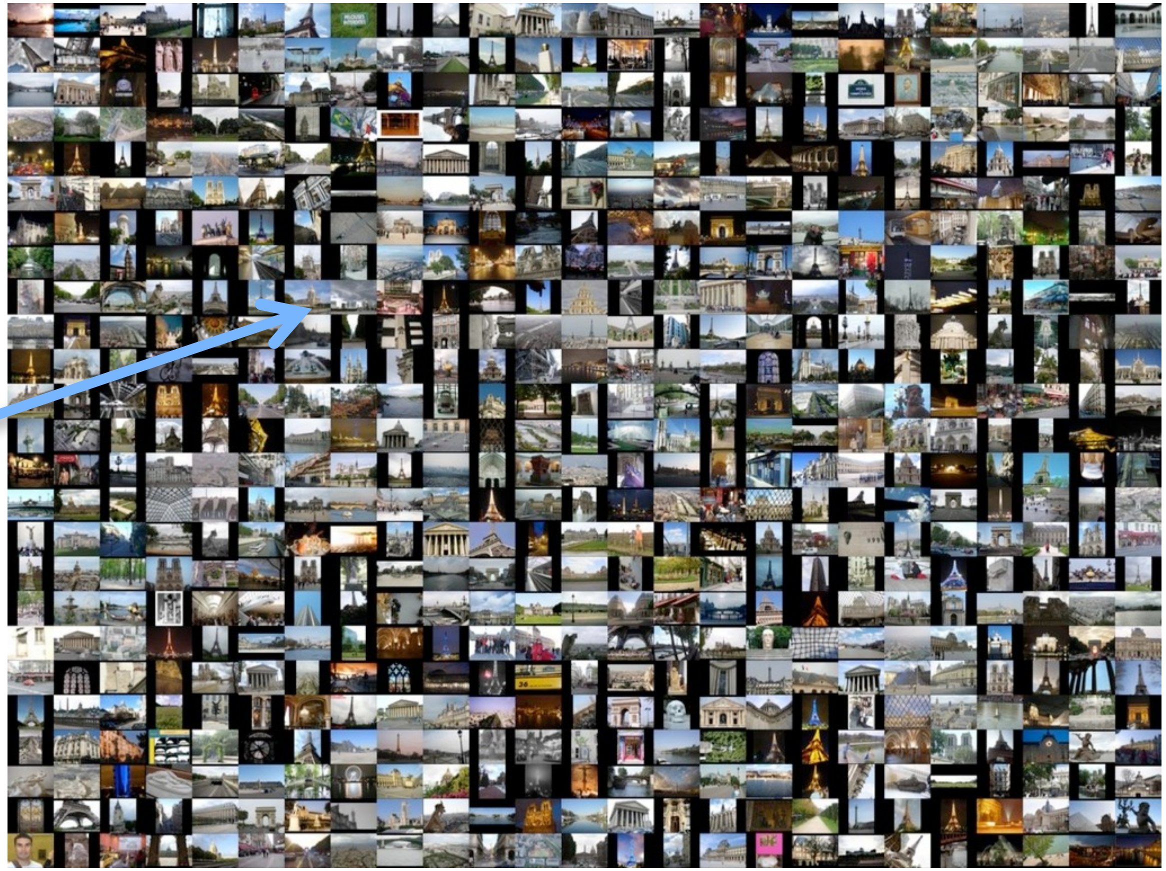
Click the icons below to see the different ways Google Goggles can be used.



- Take photos of objects as queries for visual search

Large-Scale Image Matching Problem





Database with thousands (millions) of images

- How can we perform this matching step efficiently?

Indexing Local Features

- Each patch / region has a descriptor, which is a point in some high-dimensional feature space (e.g., SIFT)

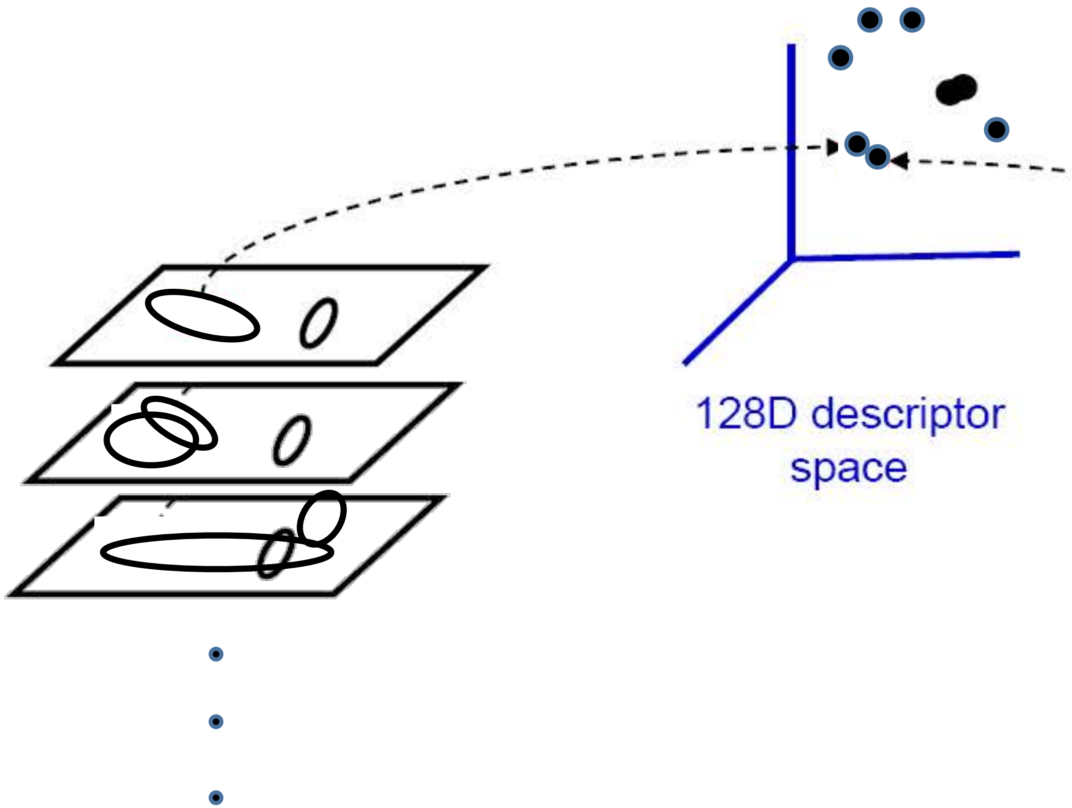
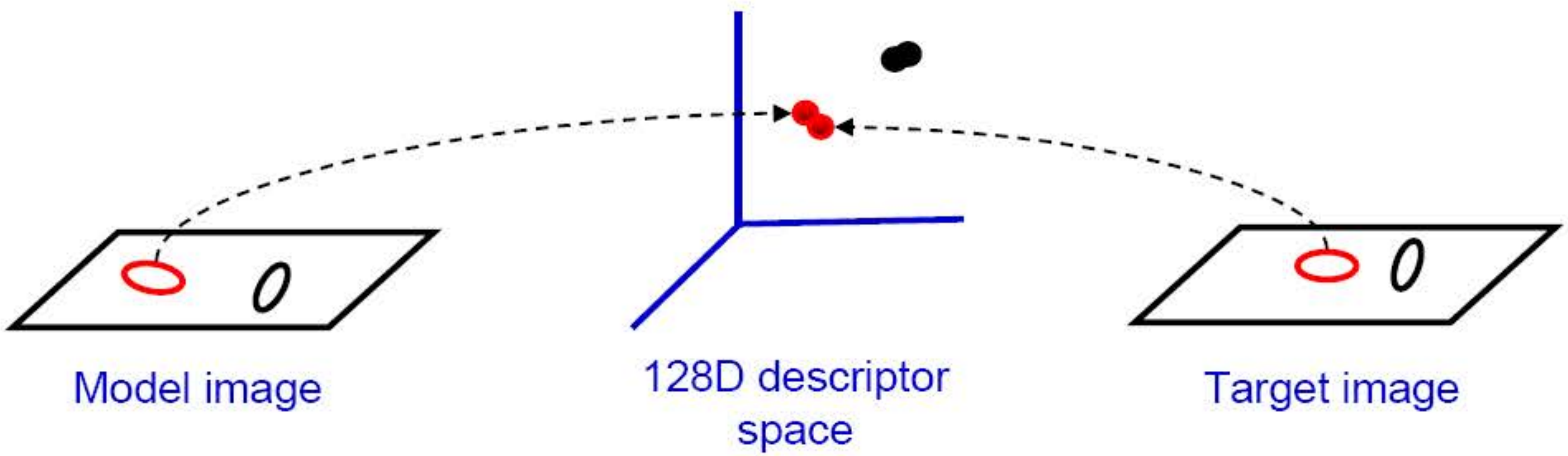


Figure credit: A. Zisserman

Indexing Local Features

- When we see close points in feature space, we have similar descriptors, which indicates similar local content.

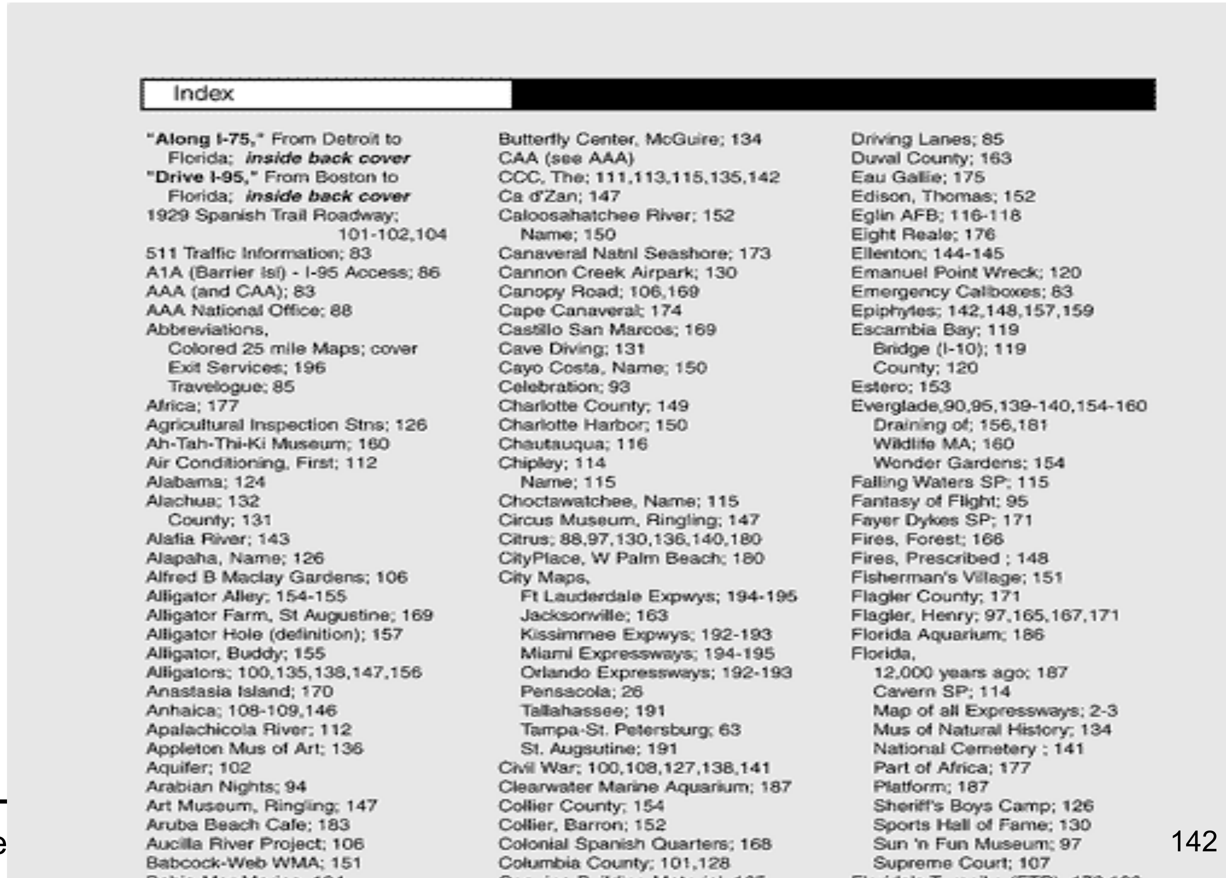


- This is of interest for many applications
  - E.g. Image matching,
  - E.g. Retrieving images of similar objects,
  - E.g. Object recognition, categorization, 3d Reconstruction,…

Figure credit: A. Zisserman

Indexing Local Features: Inverted File Index

- For text documents, an efficient way to find all ***pages*** on which a ***word*** occurs is to use an index…
- We want to find all ***images*** in which a ***feature*** occurs.
- To use this idea, we'll need to map our features to "visual words".



Index

slide credit:
K. Grauman, B. Leibe

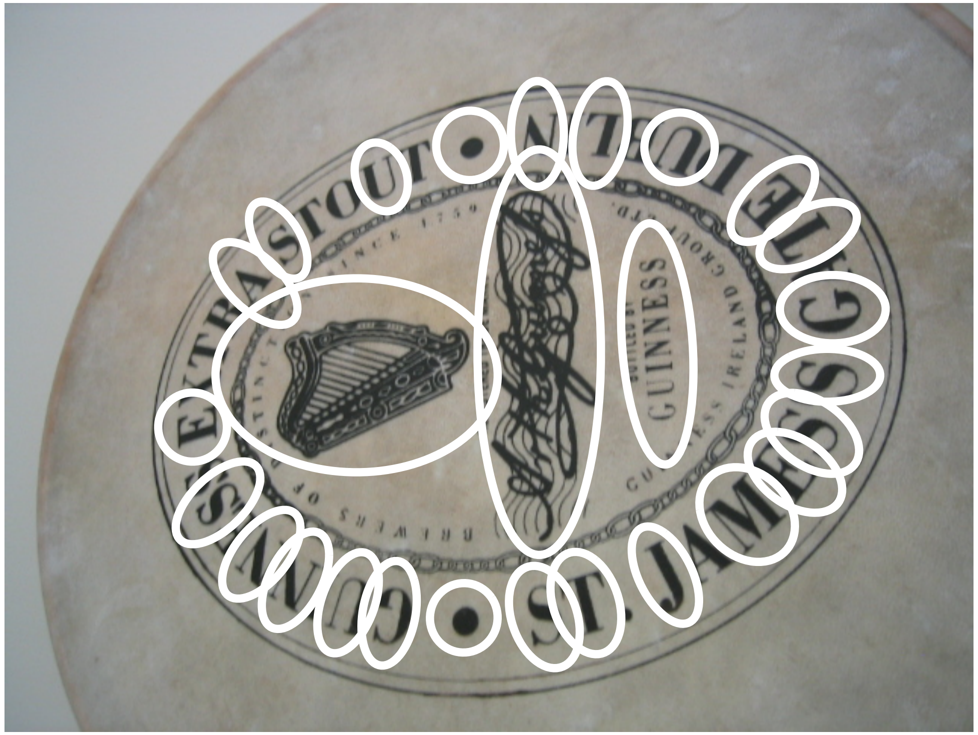
Visual Words: Main Idea

- Extract some local features from a number of images …

Slide credit: David Nister

Visual Words: Main Idea

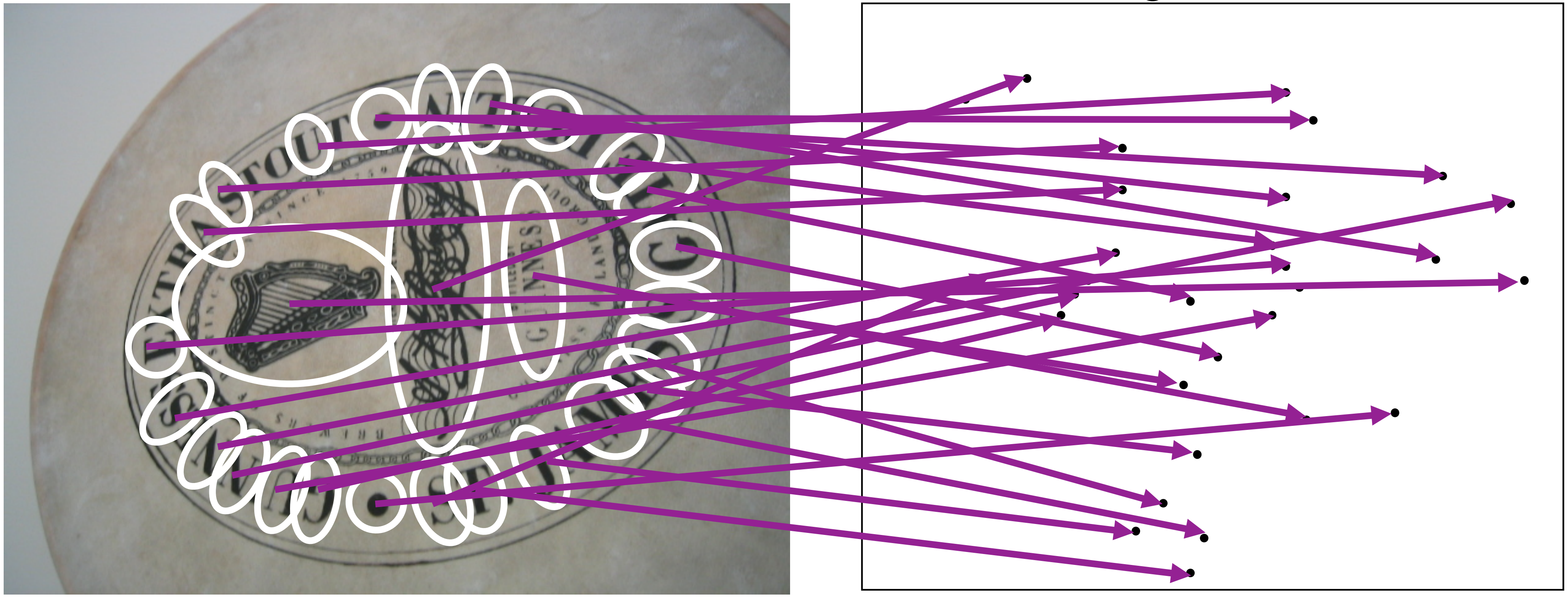
- Extract some local features from a number of images …



Slide credit: David Nister

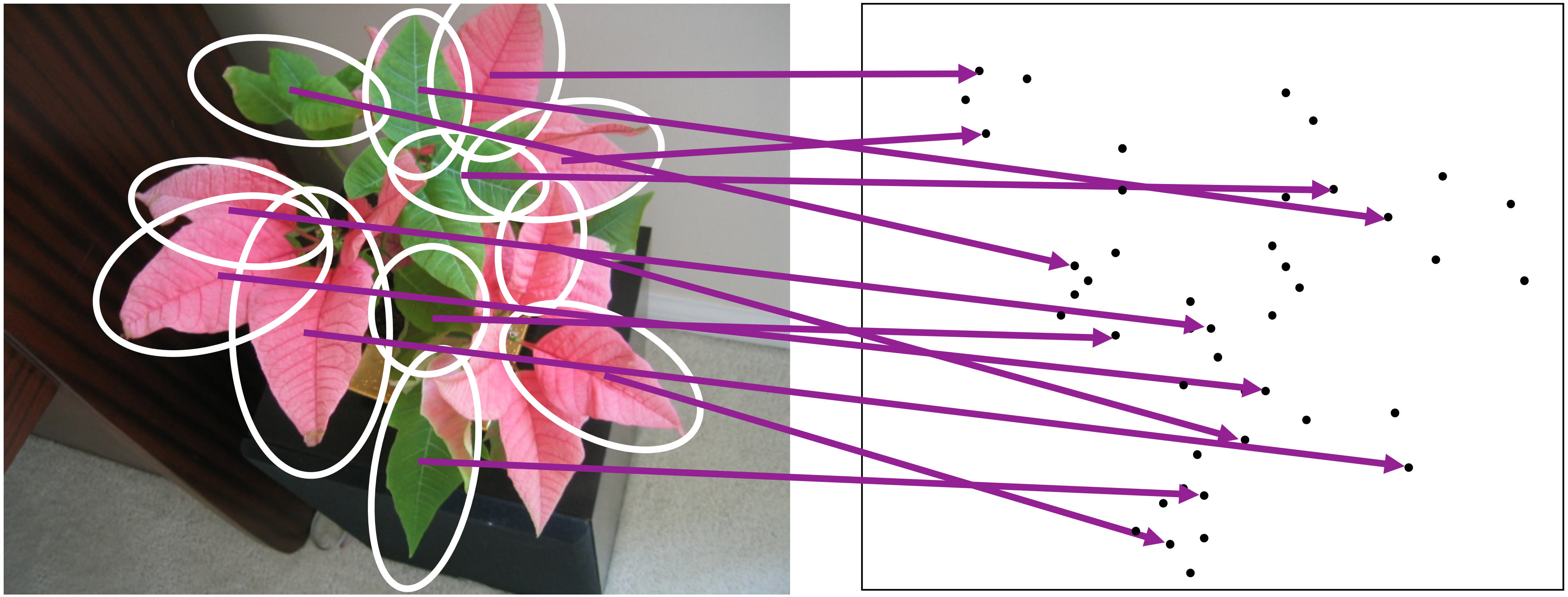
Visual Words: Main Idea

- Extract some local features from a number of images …



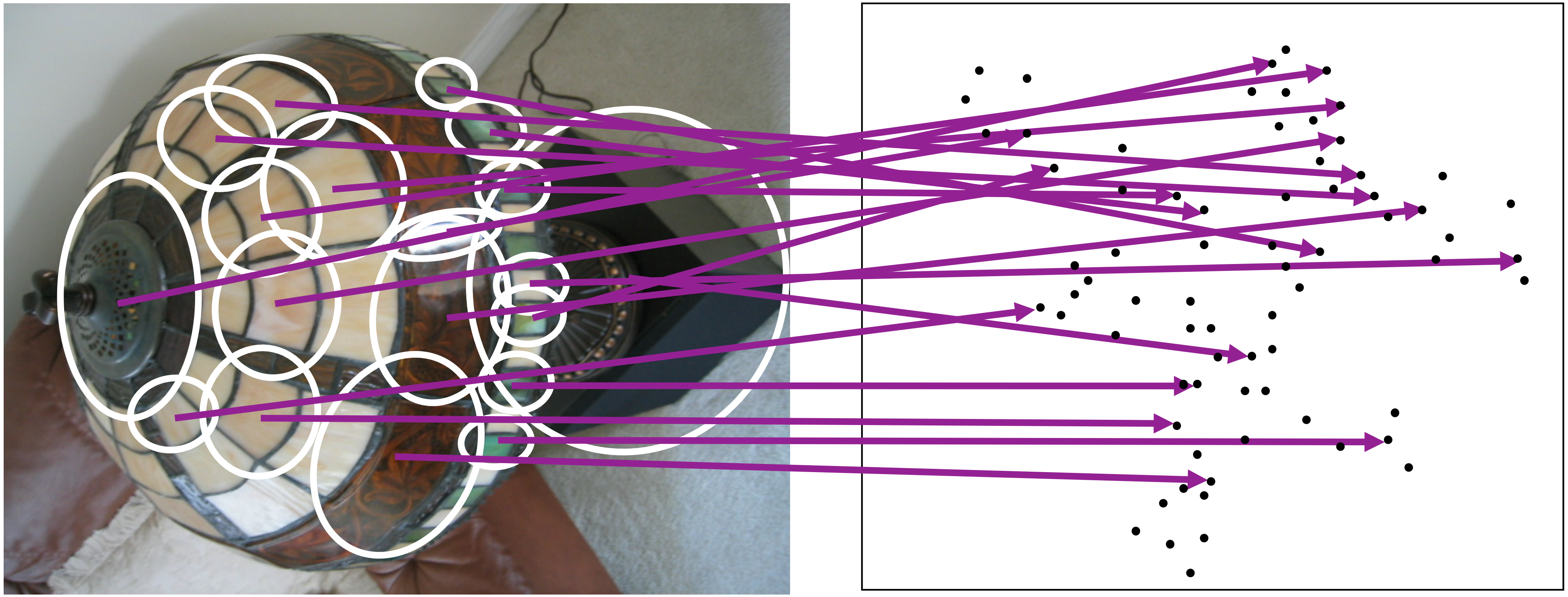
Slide credit: David Nister

Visual Words: Main Idea



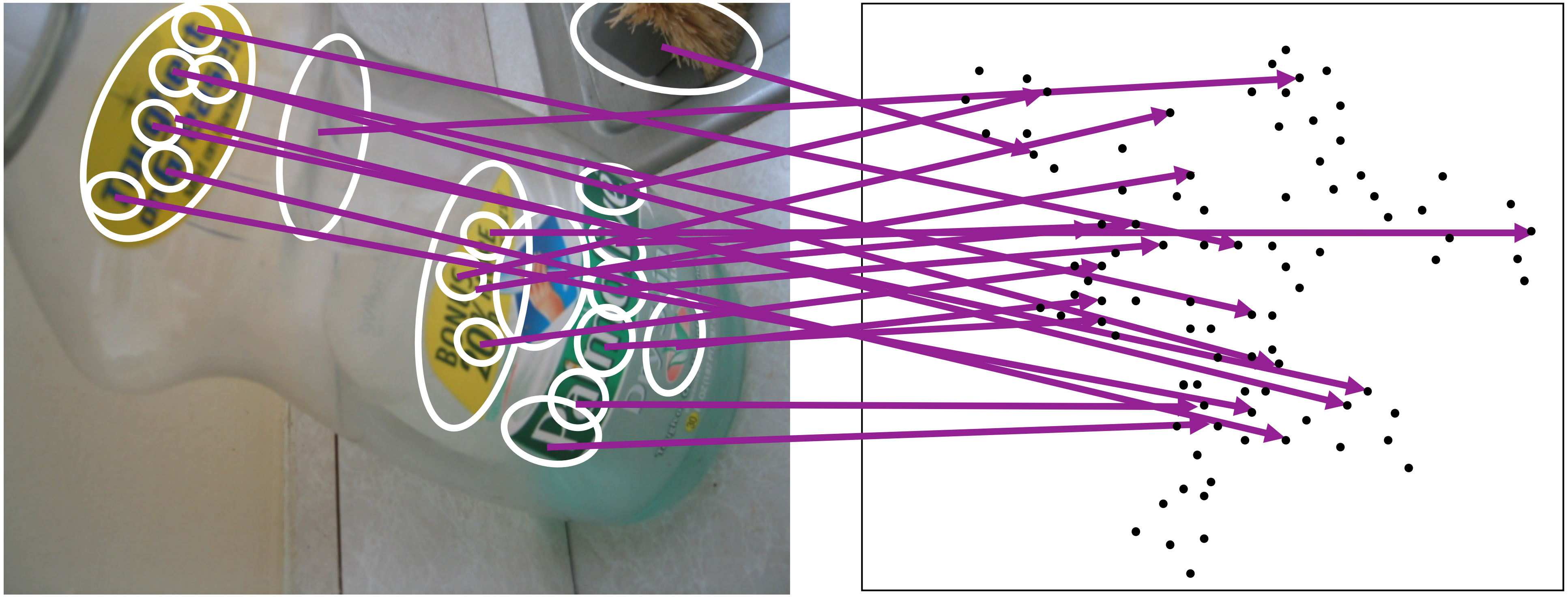
Slide credit: David Nister

Visual Words: Main Idea

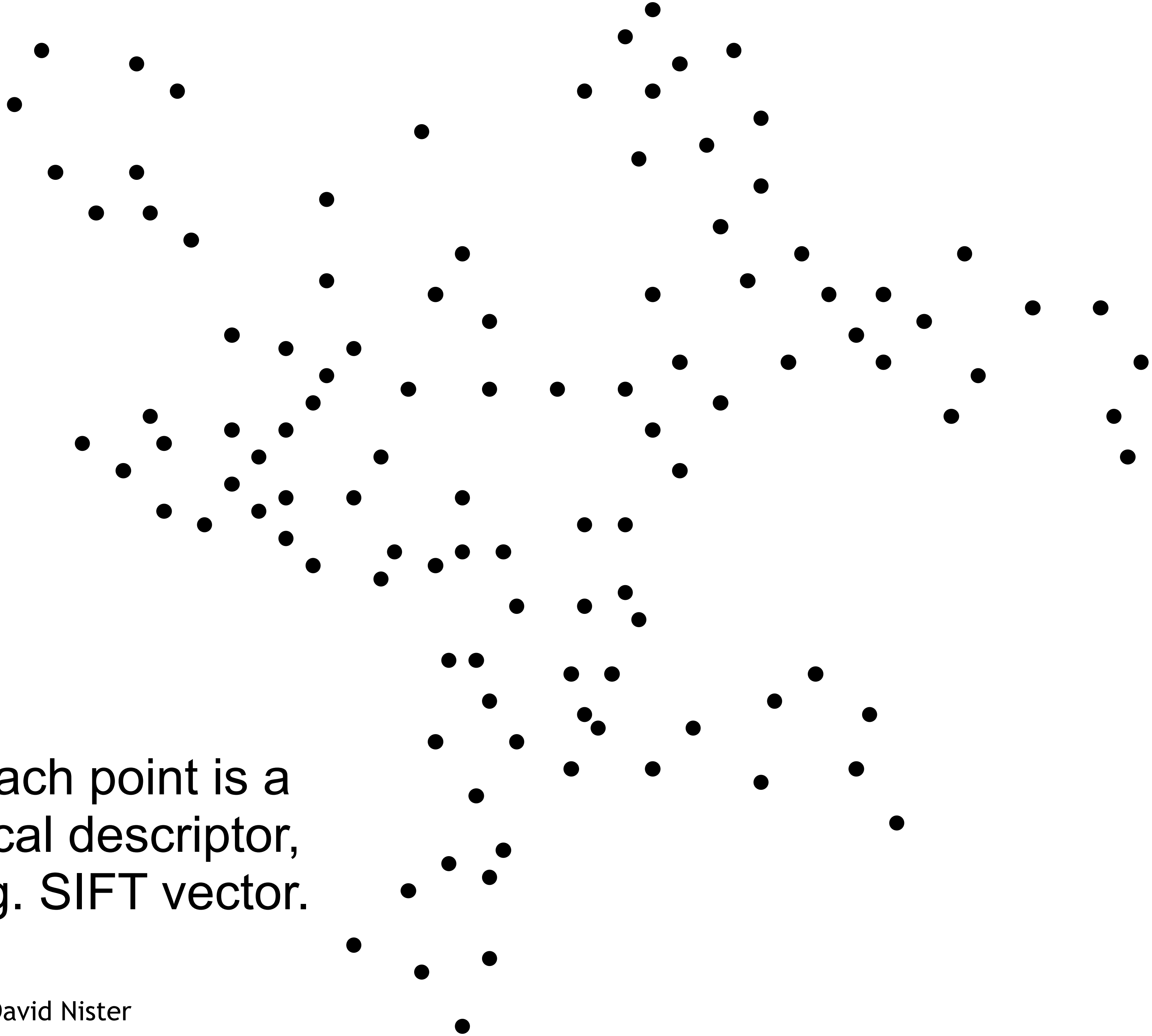


Slide credit: David Nister

Visual Words: Main Idea

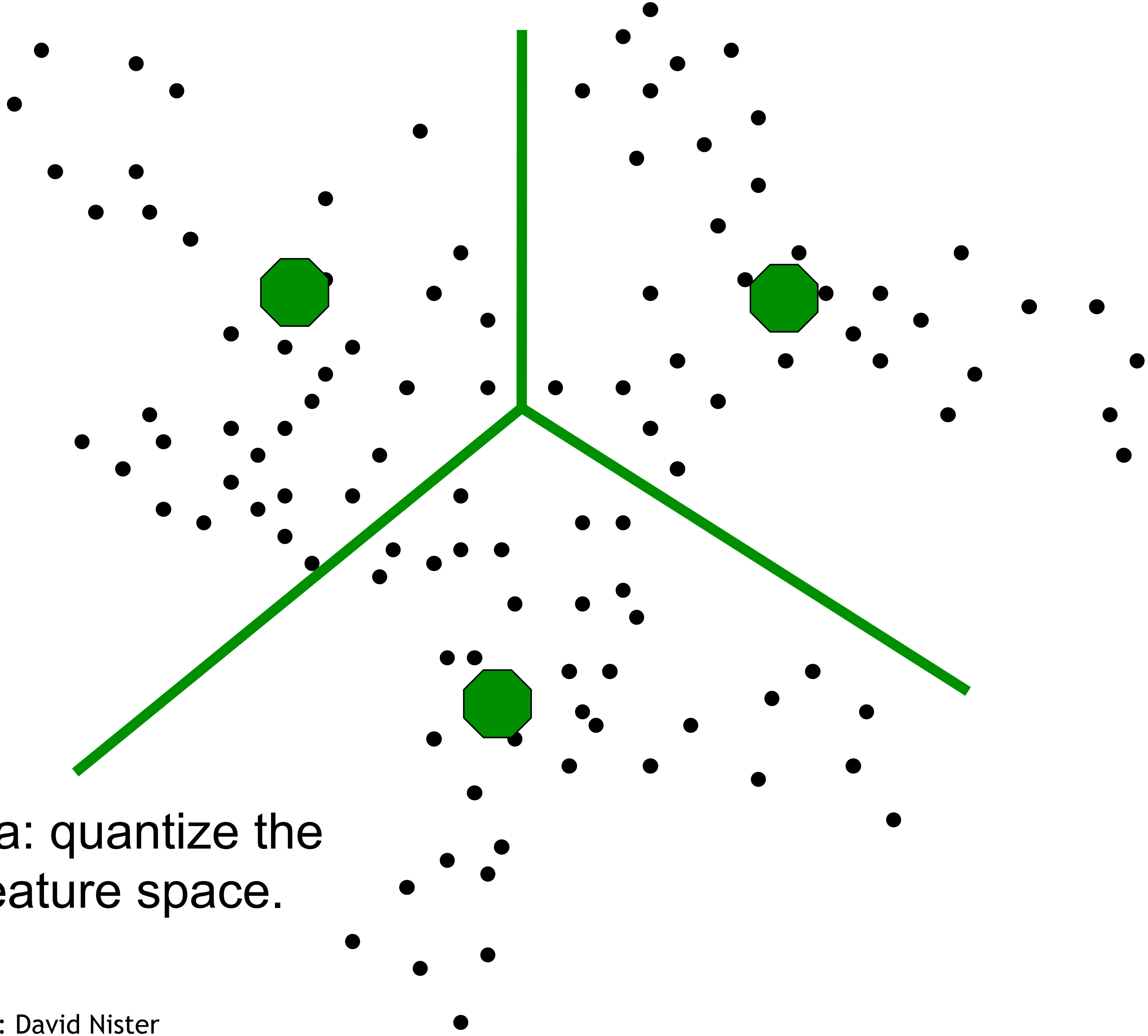


Slide credit: David Nister



Each point is a local descriptor, e.g. SIFT vector.

Slide credit: David Nister

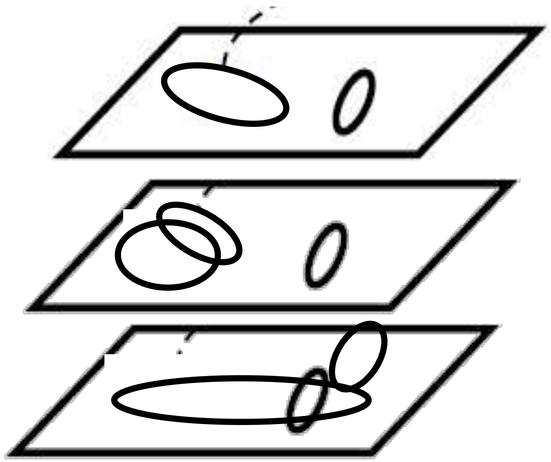


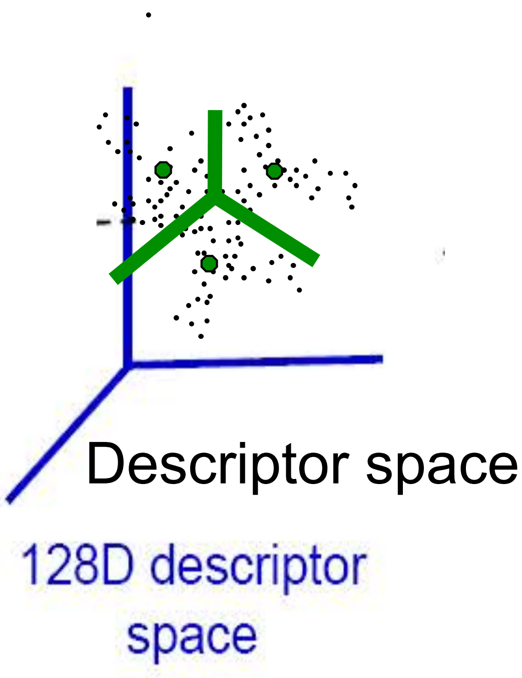
Idea: quantize the feature space.

Slide credit: David Nister

Indexing with Visual Words

Map high-dimensional descriptors to tokens/words by quantizing the feature space



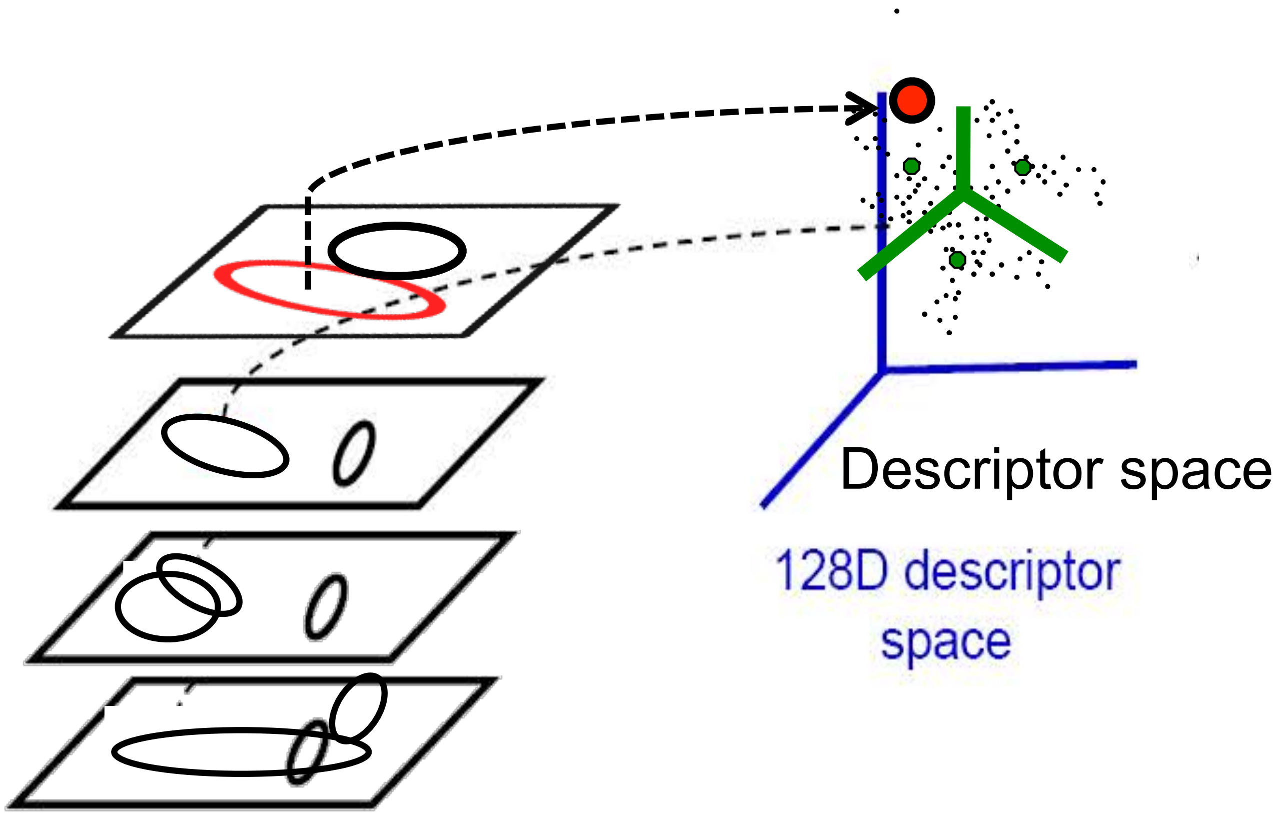


Quantize via clustering, let cluster centers be the prototype "words"

Slide credit: Kristen Grauman

Indexing with Visual Words

Map high-dimensional descriptors to tokens/words by quantizing the feature space

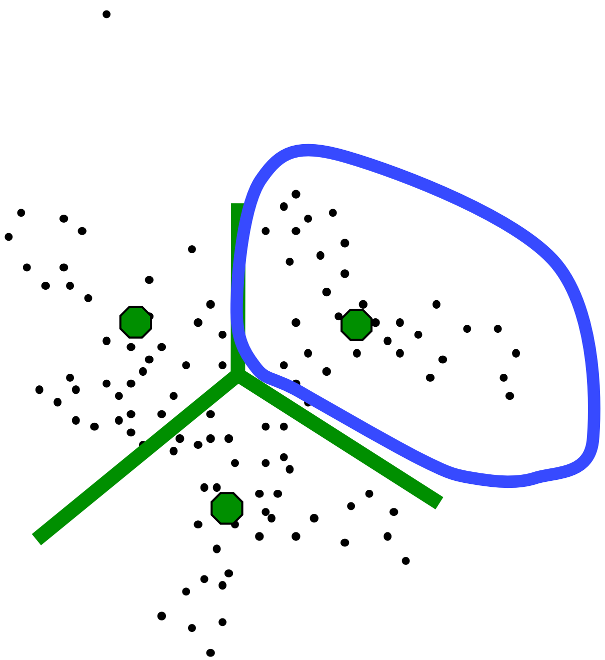


Determine which word to assign to each new image region by finding the closest cluster center.

Slide credit: Kristen Grauman

Visual Words

- Example: each group of patches belongs to the same visual word



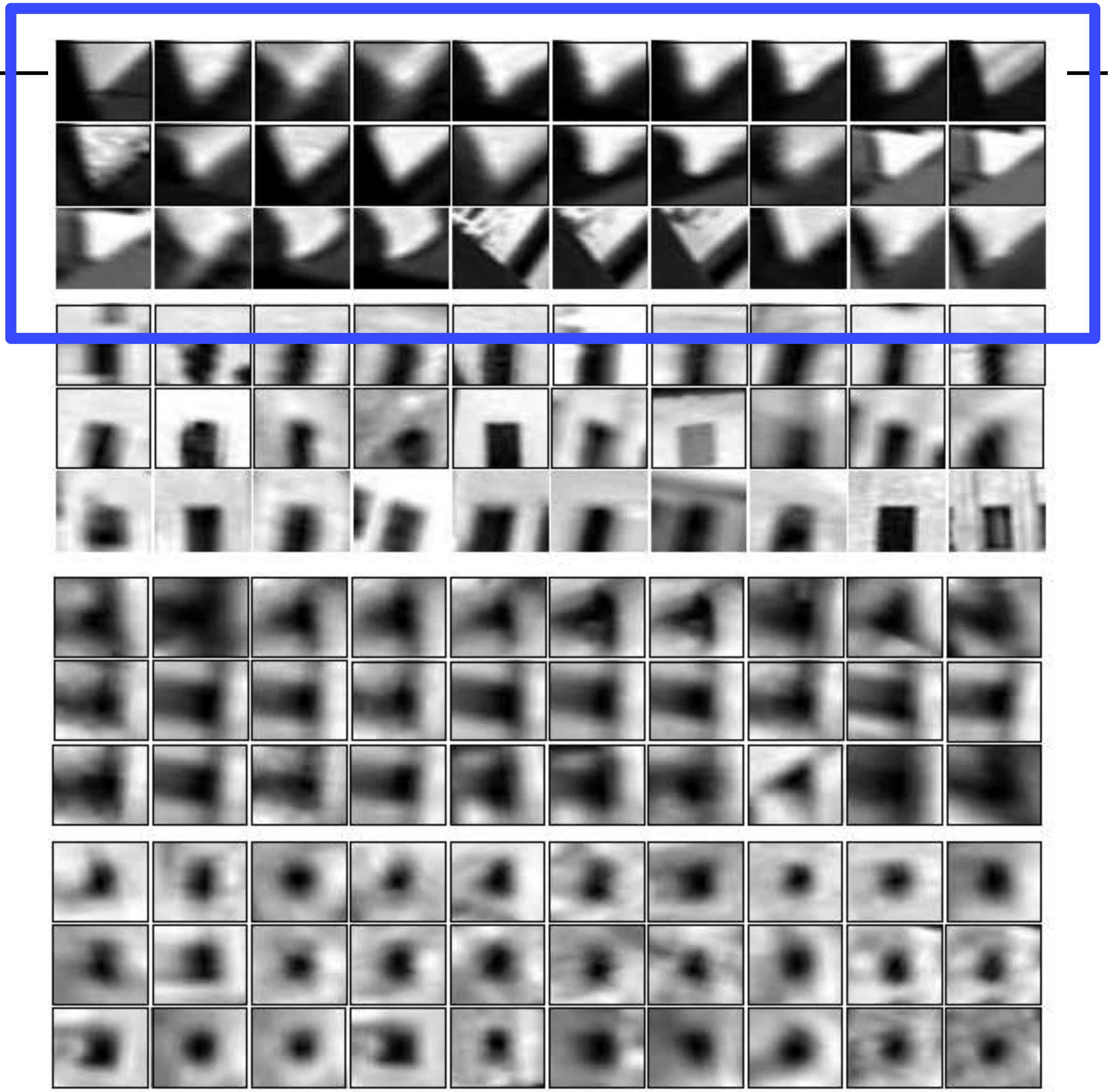
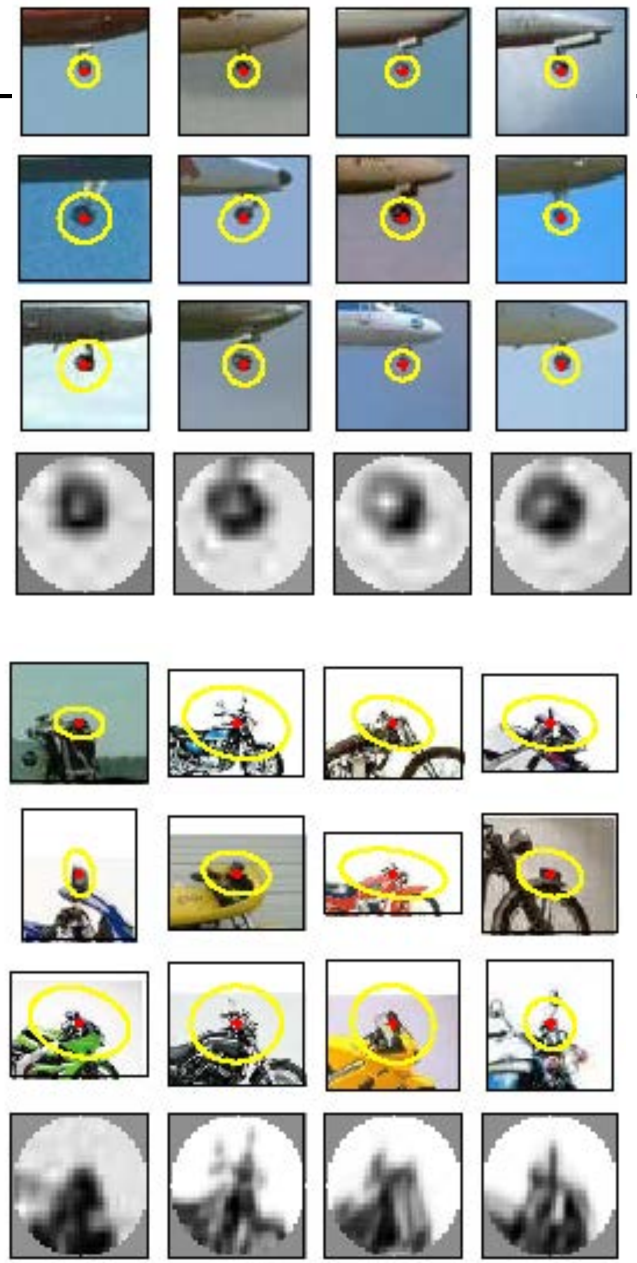


Figure from Sivic & Zisserman, ICCV 2003

Slide credit: Kristen Grauman

Visual Words

- More recently used for describing scenes and objects for the sake of indexing or classification.



Sivic & Zisserman 2003; Csurka, Bray, Dance, & Fan 2004; many others.

Slide credit: Kristen Grauman

Inverted File for Images of Visual Words

frame #5

frame #10

Word number		List of image numbers
1	→	5,10, ...
2	→	10,...
...		...

When will this give us a significant gain in efficiency?

Slide credit: Kristen Grauman

Image credit: A. Zisserman

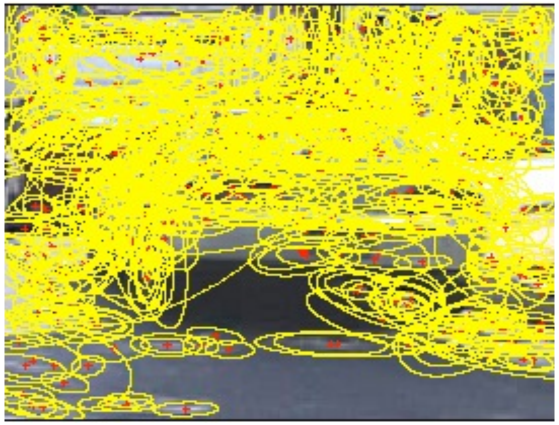
Visual Vocabulary Formation

Design choices:

- Sampling strategy: where to extract features?
- Clustering / quantization algorithm
- Unsupervised vs. supervised
- What corpus provides features (universal vocabulary?)
- Vocabulary size, number of words

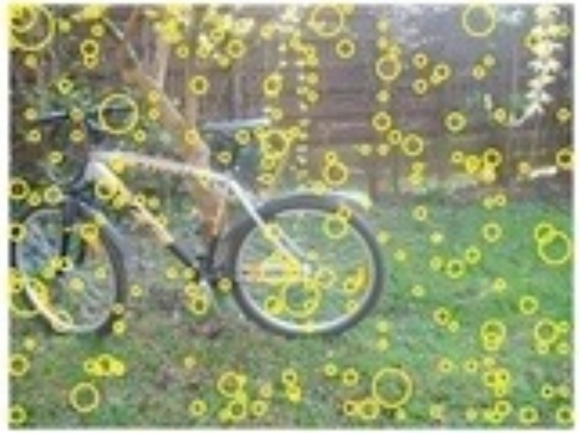
Slide credit: Kristen Grauman

Sampling Strategies

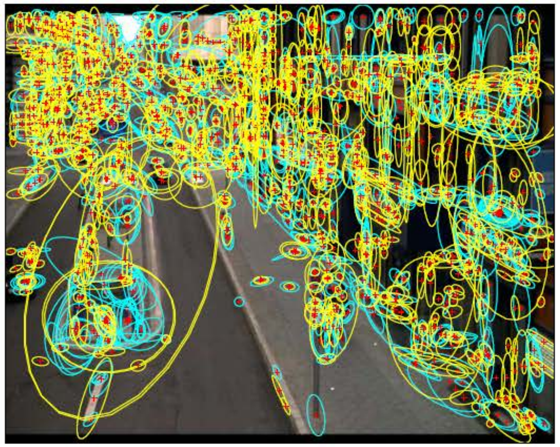


Sparse, at
interest points

Dense, uniformly



Randomly



Multiple interest
operators

- To find specific, textured objects, sparse sampling from interest points often more reliable.
- Multiple complementary interest operators offer more image coverage.
- For object categorization, dense sampling offers better coverage.

[See Nowak, Jurie & Triggs, ECCV 2006]

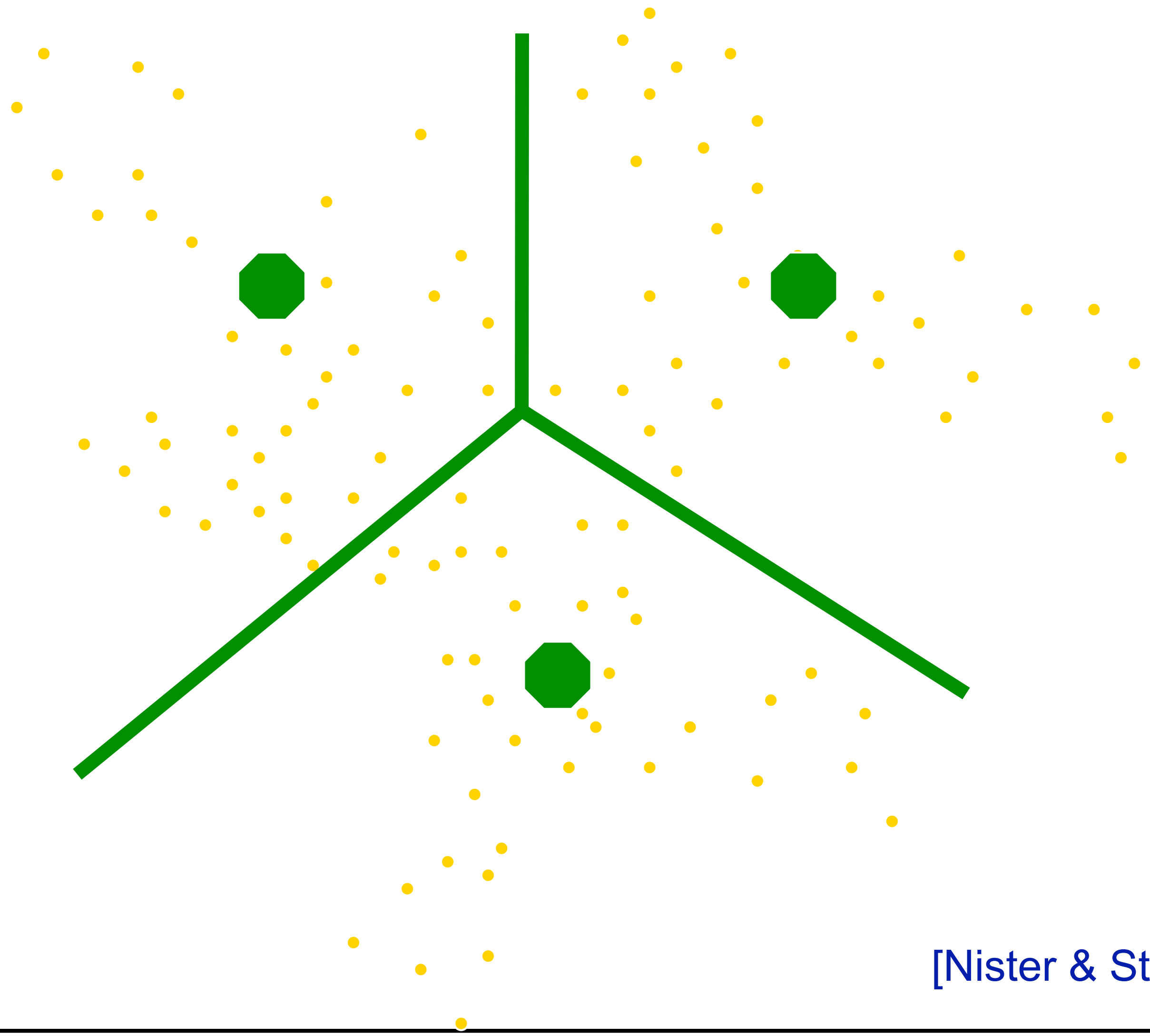
Clustering / Quantization Methods

- k-means (typical choice), agglomerative clustering, mean-shift,…
- Hierarchical clustering: allows faster insertion / word assignment while still allowing large vocabularies
  - Vocabulary tree [Nister & Stewenius, CVPR 2006]

Slide credit: Kristen Grauman

Example: Recognition with Vocabulary Tree

- Tree construction:

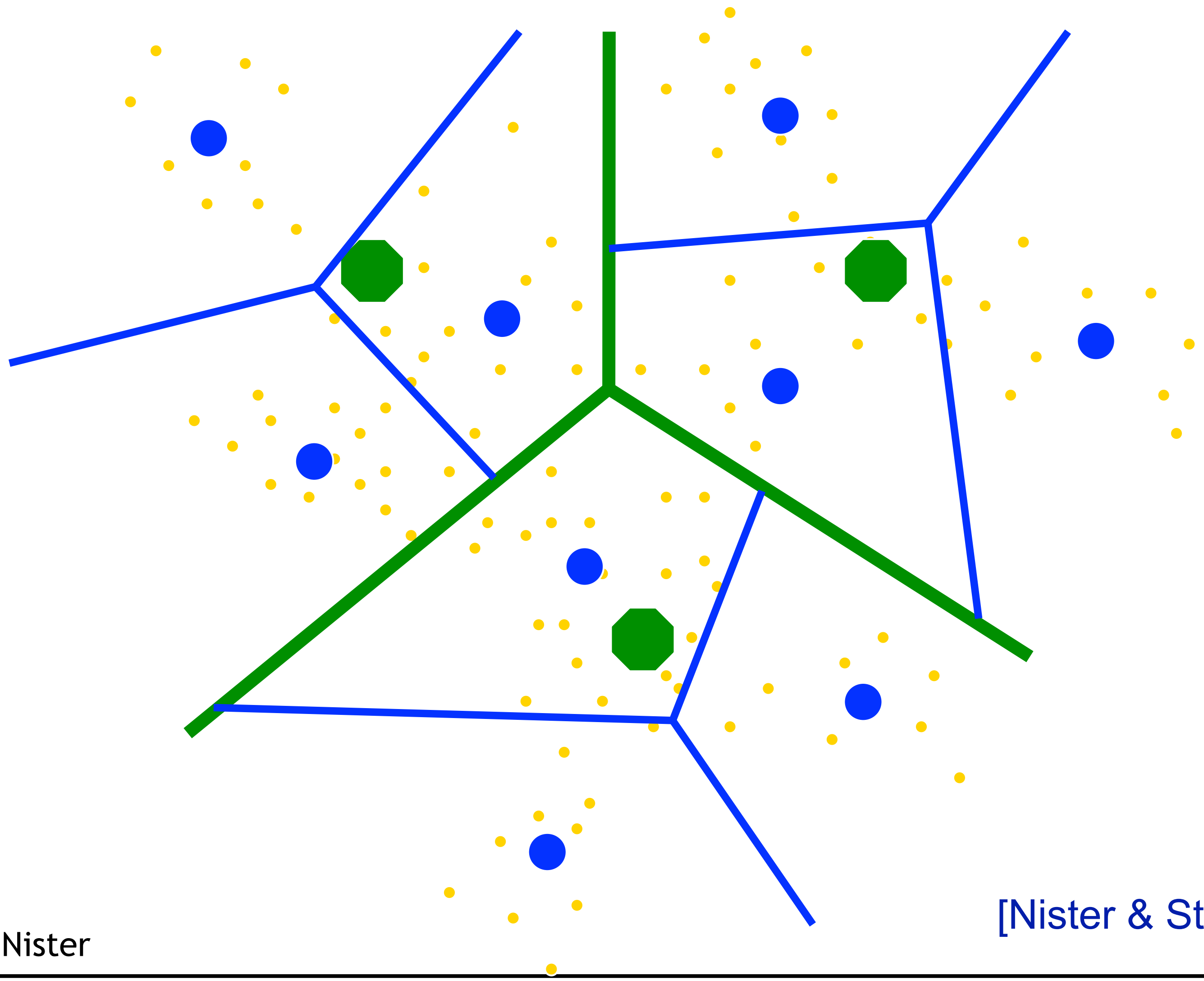


Slide credit: David Nister

[Nister & Stewenius, CVPR'06]

Example: Recognition with Vocabulary Tree

- Tree construction:

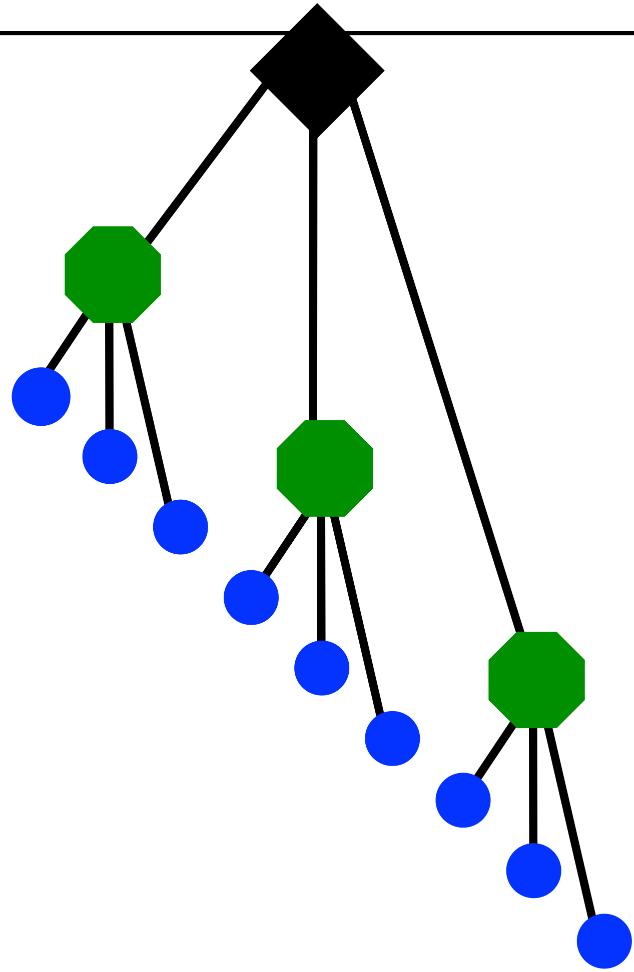


Slide credit: David Nister

[Nister & Stewenius, CVPR'06]

Vocabulary Tree

- Training: Filling the tree

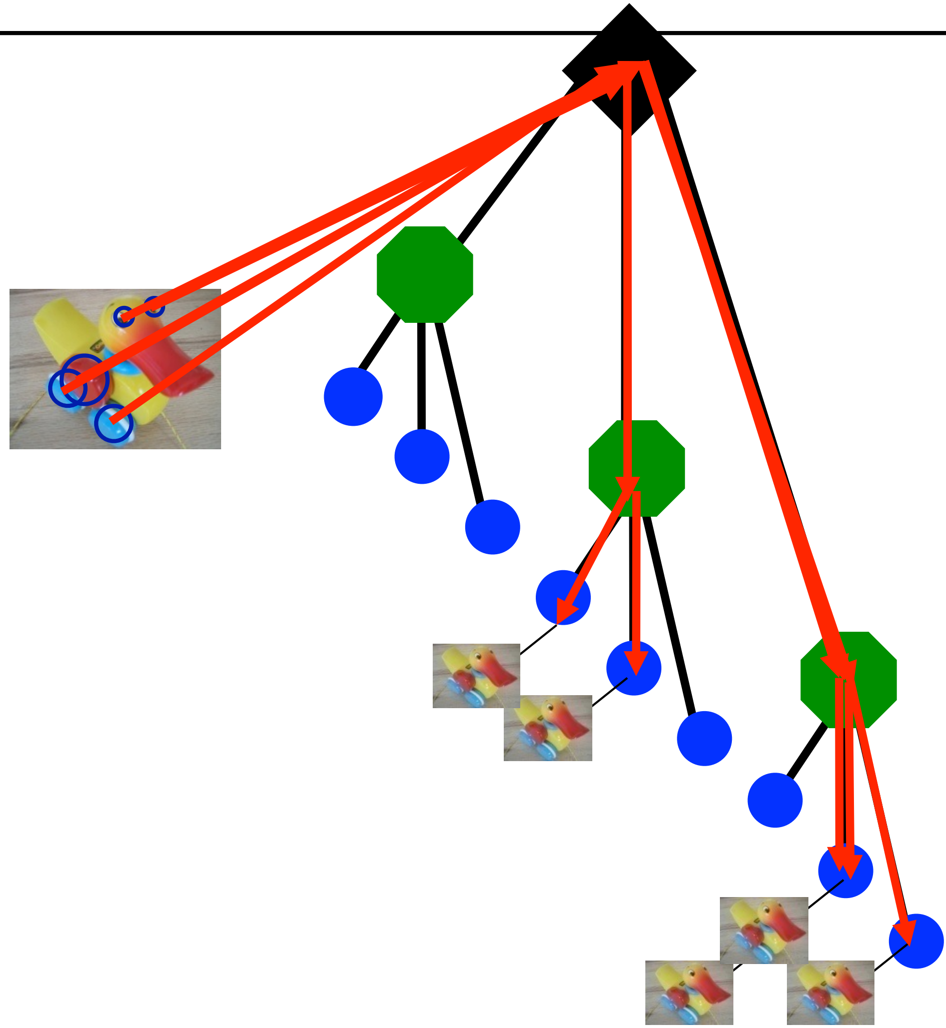


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

- Training: Filling the tree

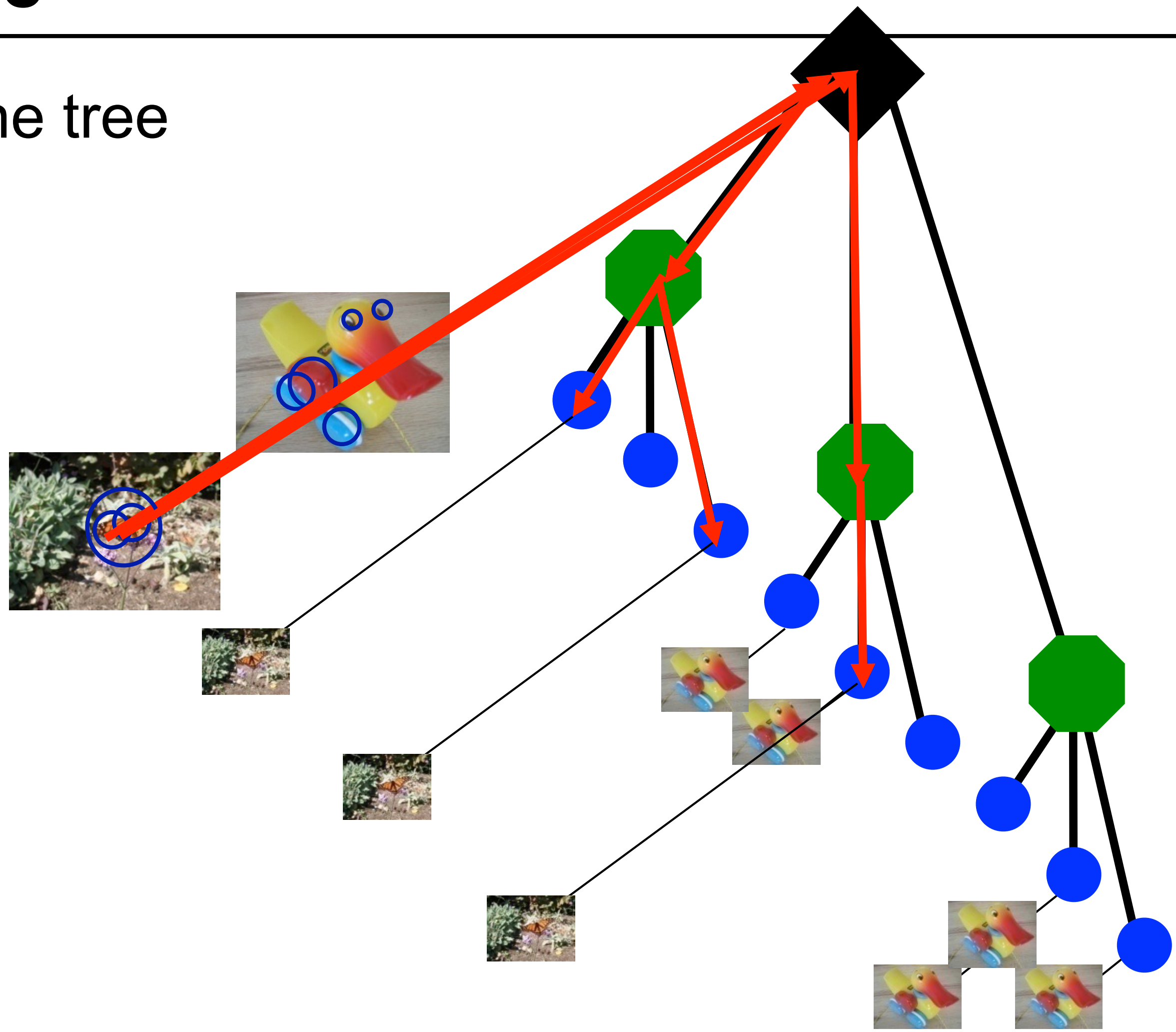


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

- Training: Filling the tree

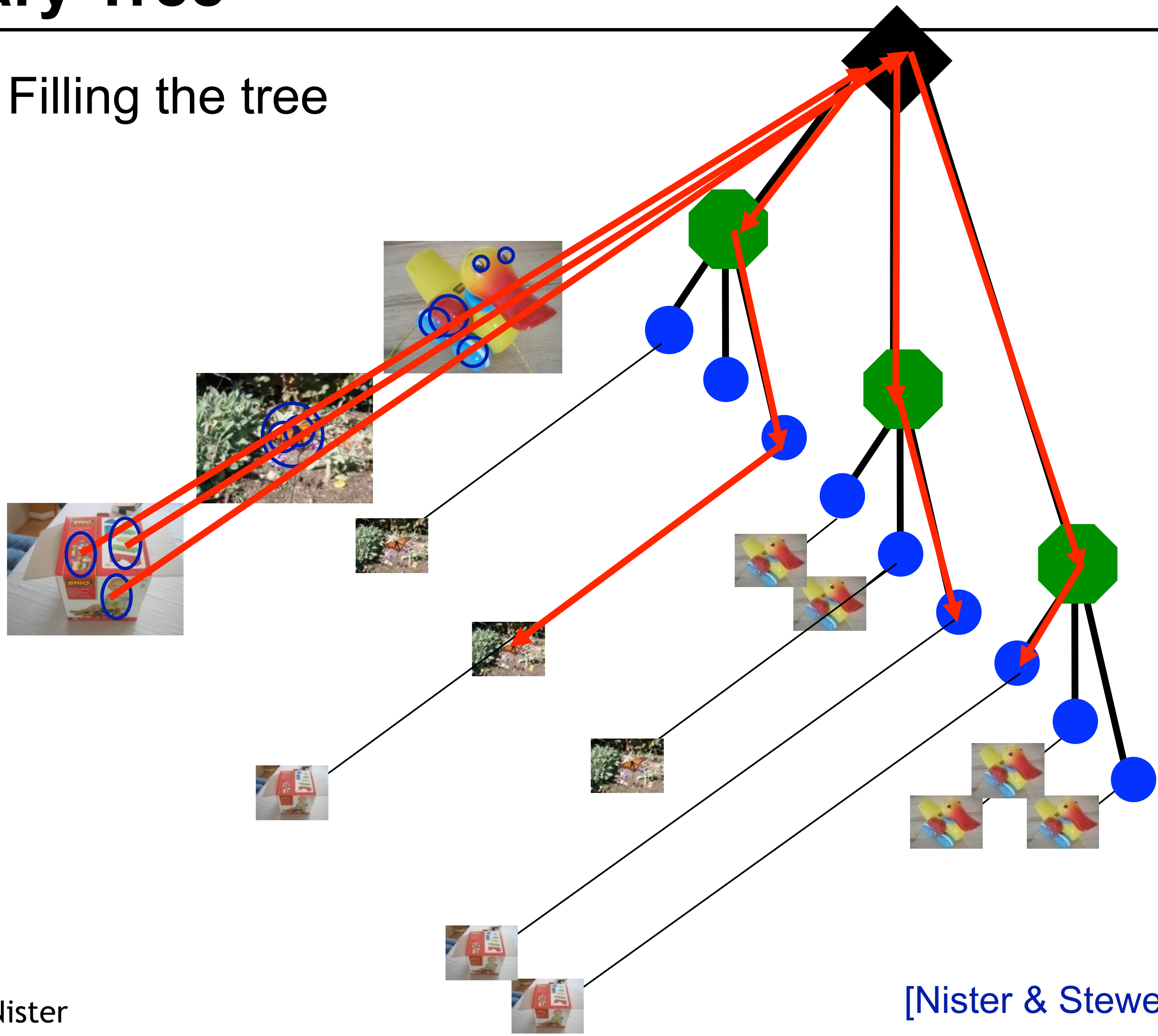


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

Vocabulary Tree

- Training: Filling the tree

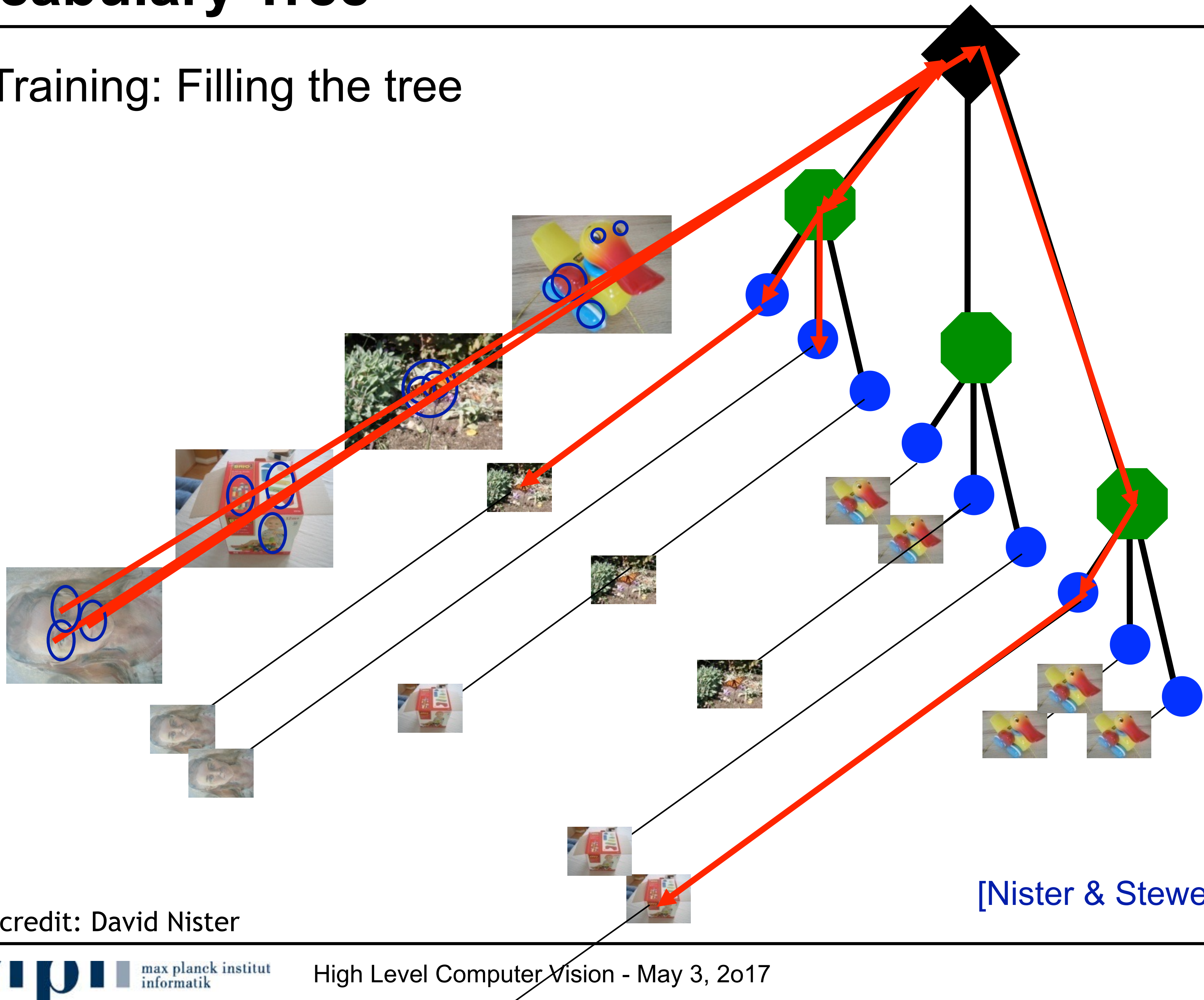


Slide credit: David Nister

[Nister & Stewenius, CVPR'06]

Vocabulary Tree

- Training: Filling the tree



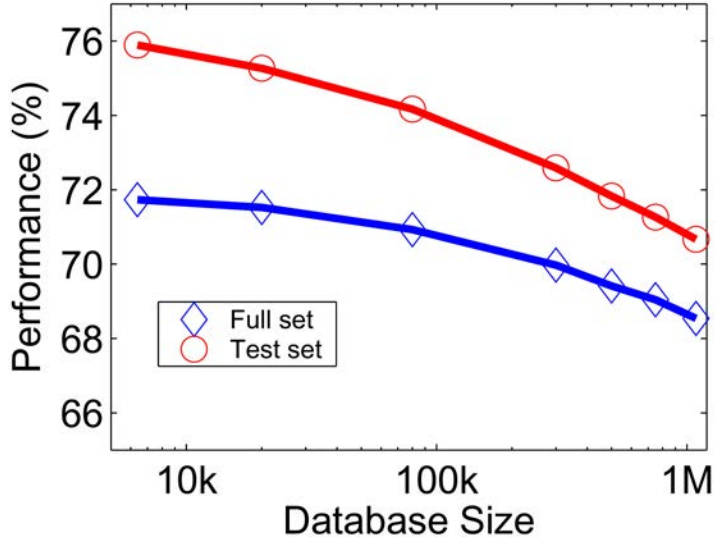
[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

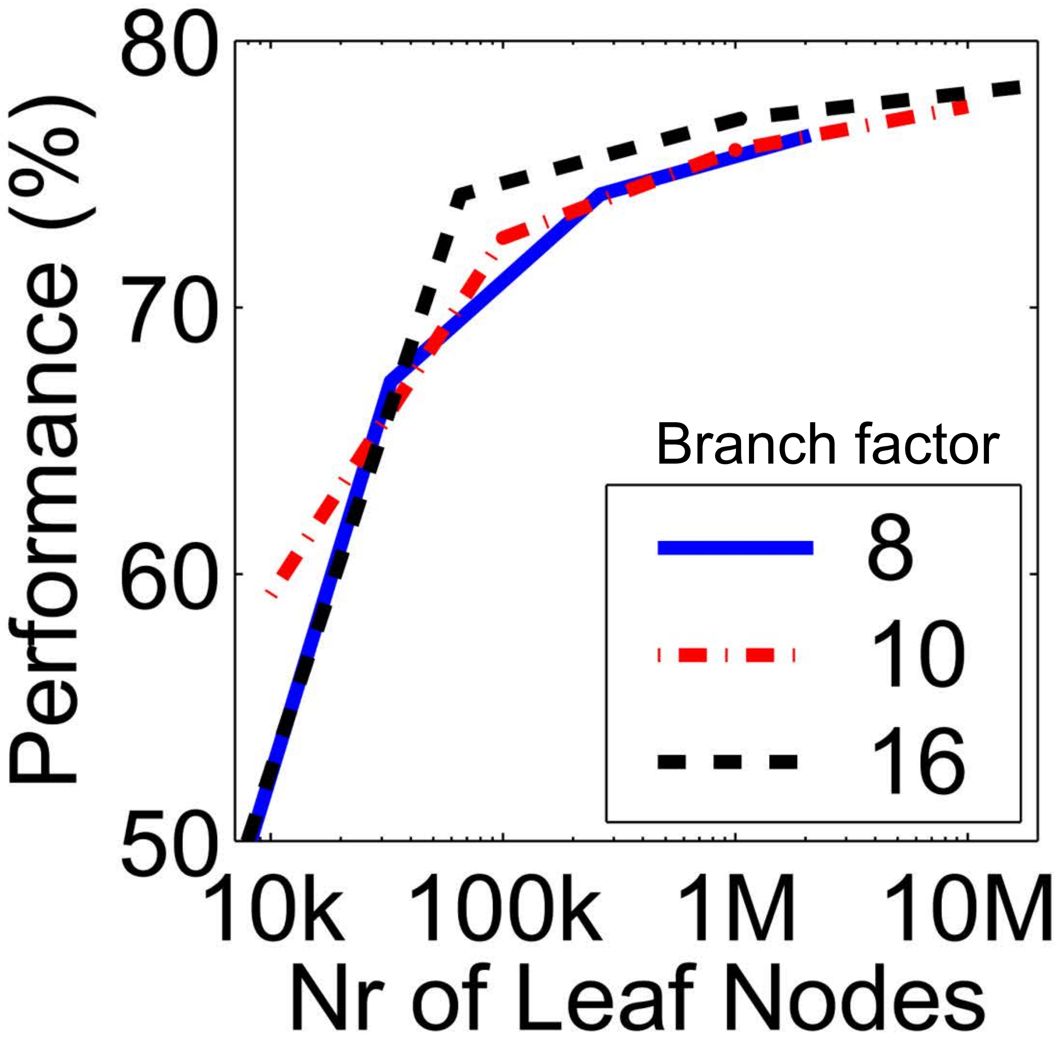
Vocabulary Tree: Performance

- Evaluated on large databases
  - Indexing with up to 1M images
- Online recognition for database of 50,000 CD covers
  - Retrieval in ~1s
- Experimental finding that large vocabularies can be beneficial for recognition

[Nister & Stewenius, CVPR'06]



Vocabulary Size



- Larger vocabularies can be advantageous…
- But what happens when the vocabulary gets too large?
  - Efficiency?
  - Robustness?

Figure from [Nister & Stewenius, CVPR'06]

tf-idf Weighting

- Term frequency – inverse document frequency
- Describe frame by frequency of each word within it, downweight words that appear often in the database
- (Standard weighting for text retrieval)

Number of occurrences of word i in document d

Total number of documents in database

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

Number of words in document d

Number of occurrences of word i in whole database

Slide credit: Kristen Grauman

Applications

- Applications
  - Content based image/video retrieval
  - Specific object recognition
  - Mobile visual search
  - Mobile augmented reality

Application for Content Based Image Retrieval

- What if query of interest is a portion of a frame?

Visually defined query

"Groundhog Day" [Rammis, 1993]

"Find this clock"

"Find this place"

Slide credit: Andrew Zisserman

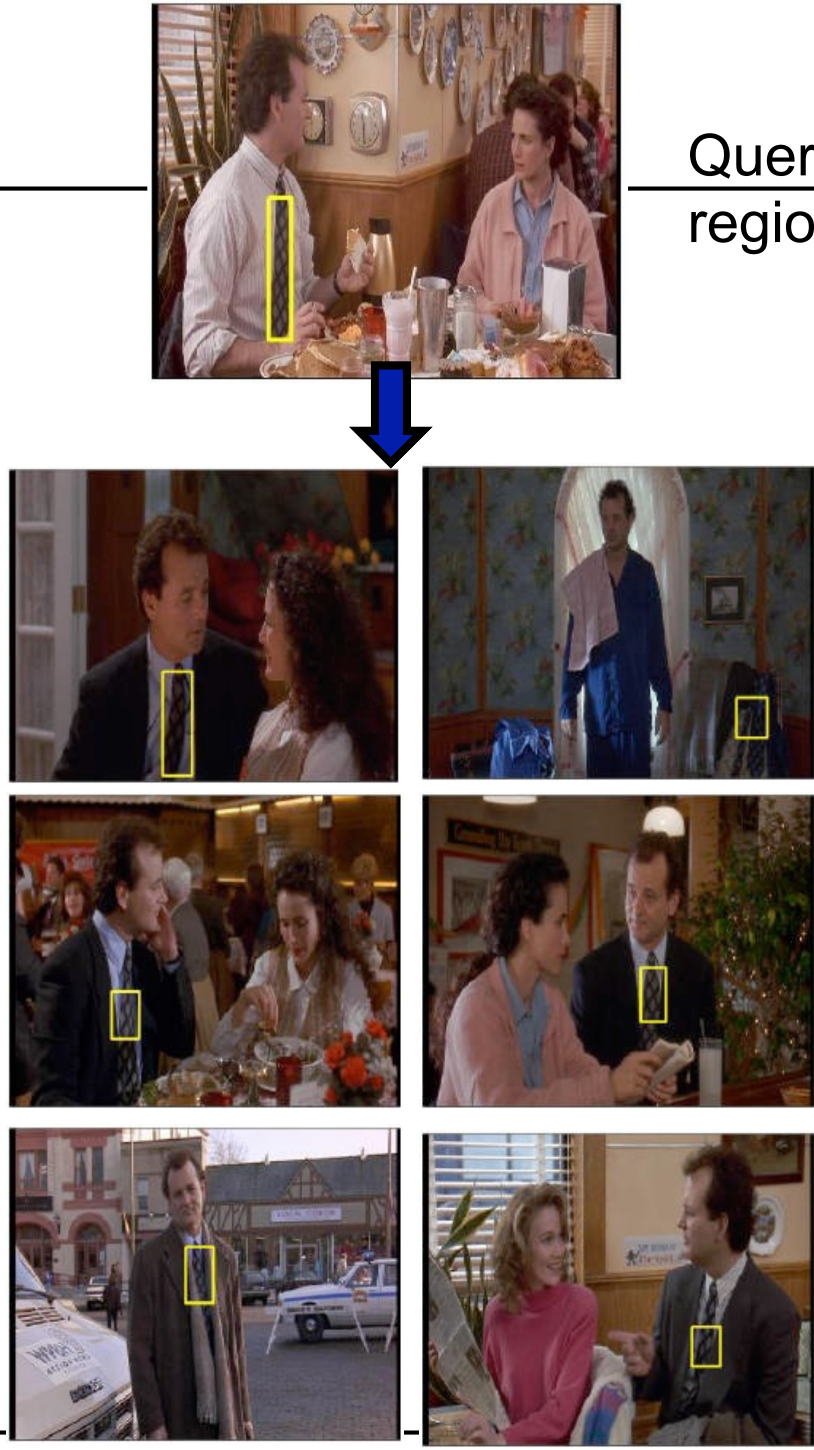
[Sivic & Zisserman, ICCV'03]

Video Google System

1. Collect all words within query region
2. Inverted file index to find relevant frames
3. Compare word counts
4. Spatial verification

Sivic & Zisserman, ICCV 2003

- Demo online at : http://www.robots.ox.ac.uk/~vgg/research/vgoogle/index.html



Query region

Retrieved frames

Slide credit: Kristen Grauman

Example Results

Query

Retrieved shots

Slide credit: Andrew Zisserman

Collecting Words Within a Query Region

- Example: Friends

Query region:
pull out only the SIFT descriptors whose positions are within the polygon

Example Results

Query

raw nn 1sim=0.56697

raw nn 2sim=0.56163

raw nn 5sim=0.54917

More Results

Query

Retrieved shots

Slide credit: Kristen Grauman

More Results

Query

Retrieved shots

Slide credit: Kristen Grauman

Applications: Specific Object Recognition

- Commercial services:

Works well for mostly planar objects:
- Movie posters,
- Book covers,
- CD/DVD covers,
- Video games,
- …

(~20M images indexed)

References and Further Reading

- David Lowe's SIFT paper
  - D. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60(2), pp. 91-110, 2004
- Details about the inverse file idea (e.g. video google)
  - J. Sivic, A. Zisserman, Video Google: A Text Retrieval Approach to Object Matching in Videos, ICCV'03, 2003.
  - D. Nistér and H. Stewénius, Scalable Recognition with a Vocabulary Tree, accepted for oral presentation at CVPR 2006.
- Good survey paper on Int. Pt. detectors and descriptors
  - T. Tuytelaars, K. Mikolajczyk, Local Invariant Feature Detectors: A Survey, Foundations and Trends in Computer Graphics and Vision, Vol. 3, No. 3, pp 177-280, 2008.