Analyse et indexation d'images et de vidéos dans des grandes bases Multimédia

Cours n°1

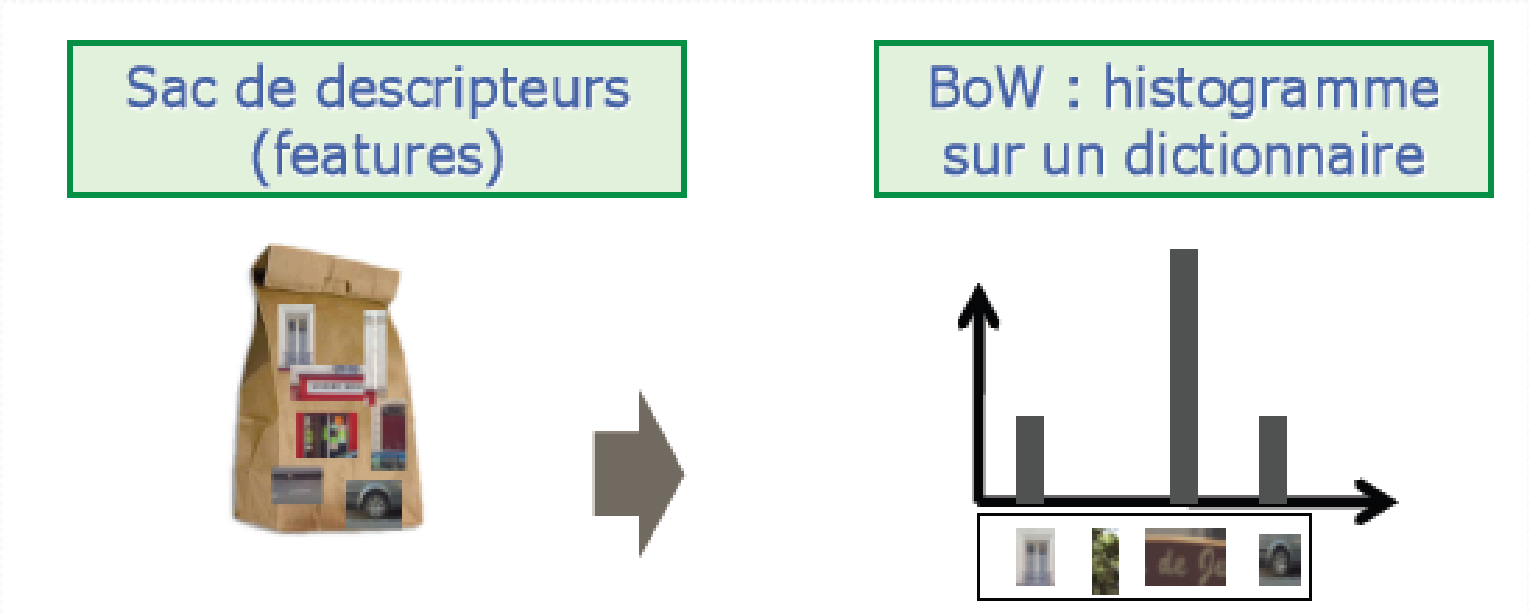
2013-2014

Frederic Precioso

Ce cours s'appuie sur les cours de George Bebis et de Derek Hoiem et d'autres collègues

From Bag of Features (BoF) to (BoW)

- BoF: Local descriptor ensembles
  - How to compare two images ?
    => vector signature
  - Trade-off discrimination / generalization

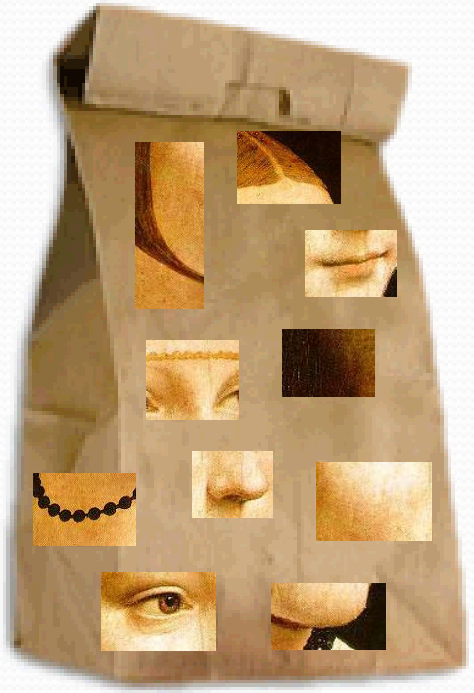


Slide credit L. Fei-Fei

Bag of Visual Words (BoW)

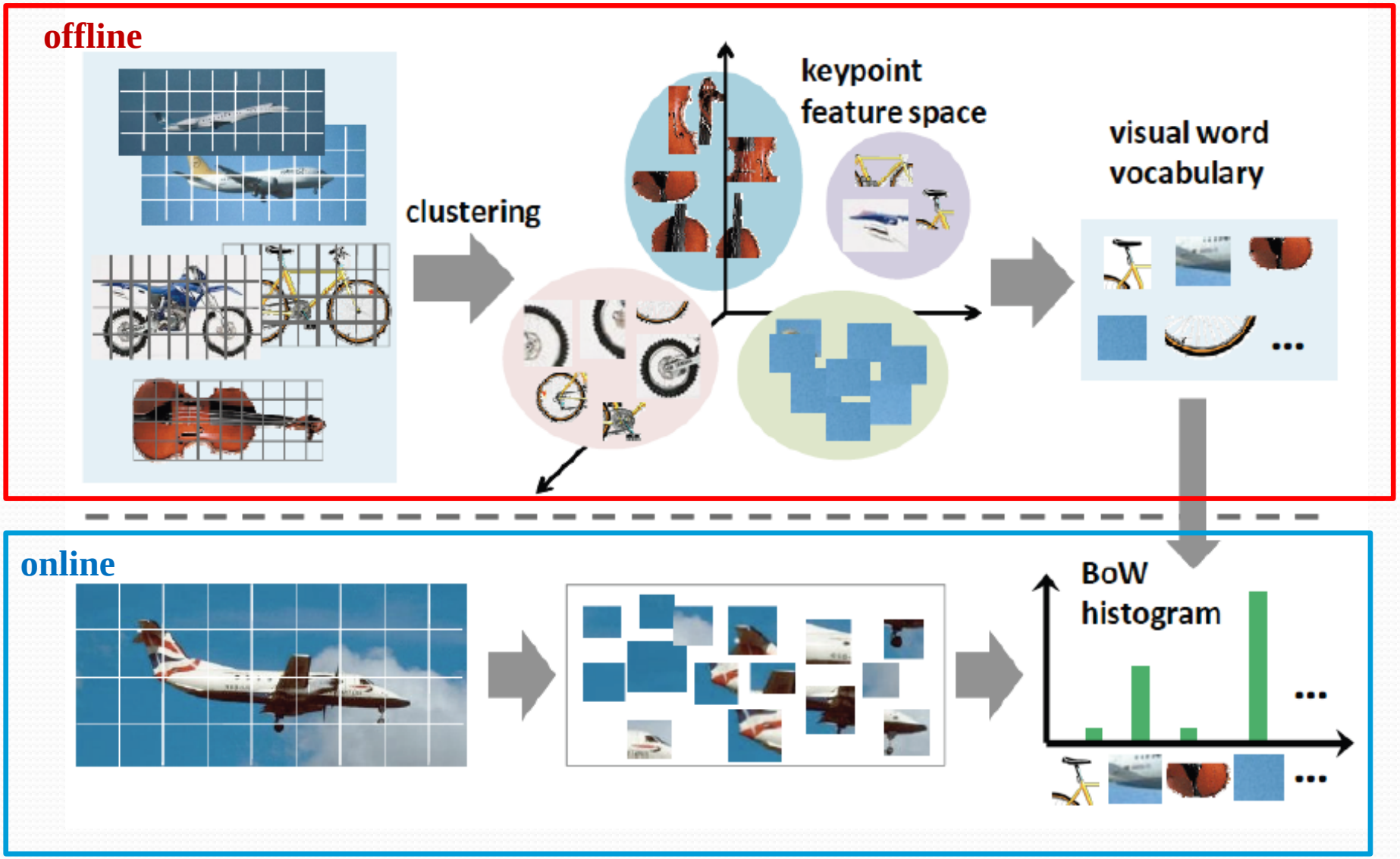
Image

Bag of Visual 'Words'



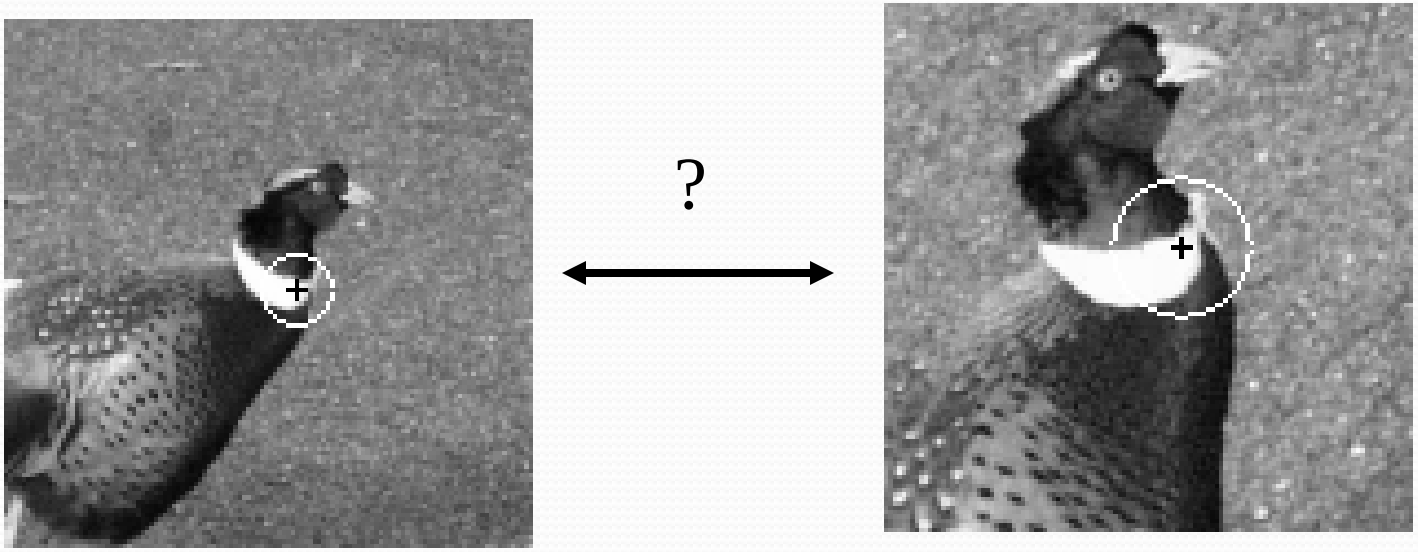
Slide credit L. Fei-Fei

Bag of Visual Words (BoW)



Matching

- Given covariant region descriptors, what information should we use for matching corresponding regions in different images?



Simplest approach: correlation

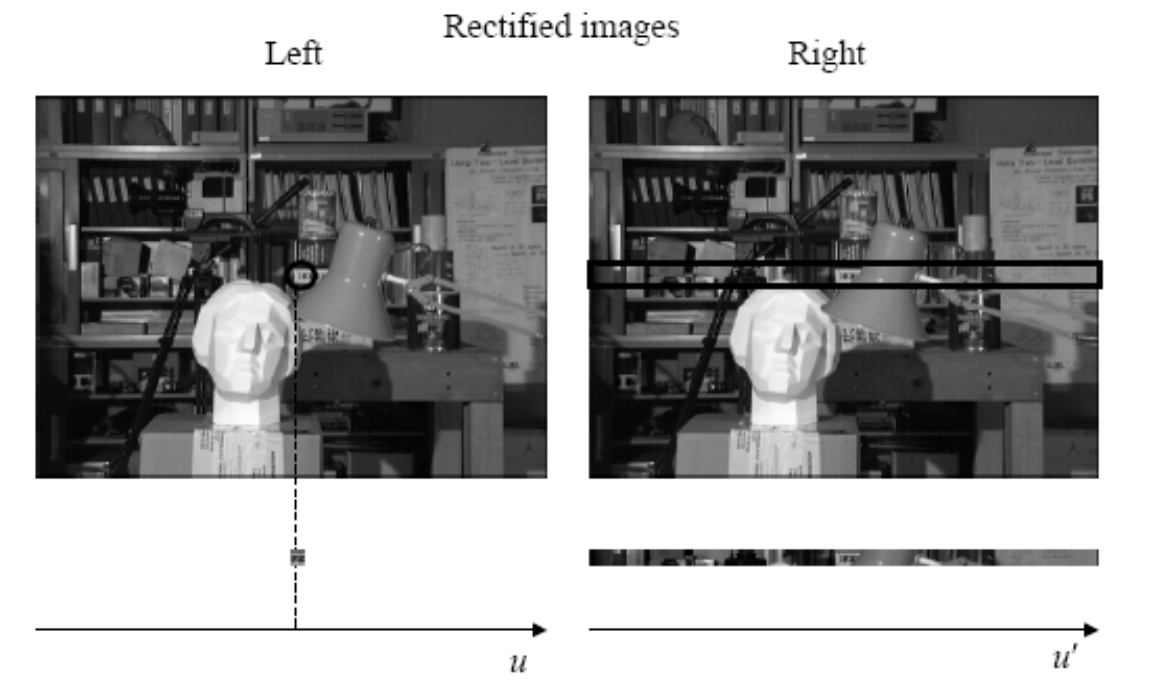
1 x NM vector of pixel intensities

w= […]

- Directly compare intensities using “sum of squared differences” or “normalized cross-correlation”

Simplest approach: correlation (cont'd)

- Works satisfactorily when we matching corresponding regions related mostly by translation.
  - e.g., stereo pairs, video sequence assuming small camera motion

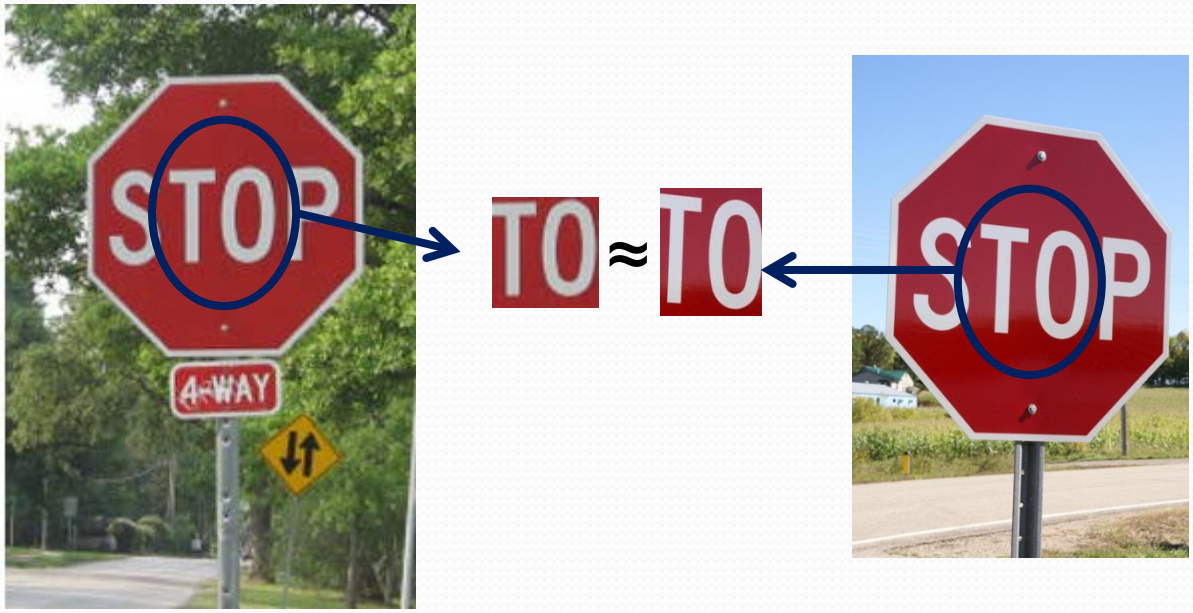


Simplest approach: correlation (cont'd)

- Sensitive to small variations with respect to:
  - Location
  - Pose
  - Scale
  - Intra-class variability
- Poorly distinctive!
- We will discuss a powerful descriptor called SIFT

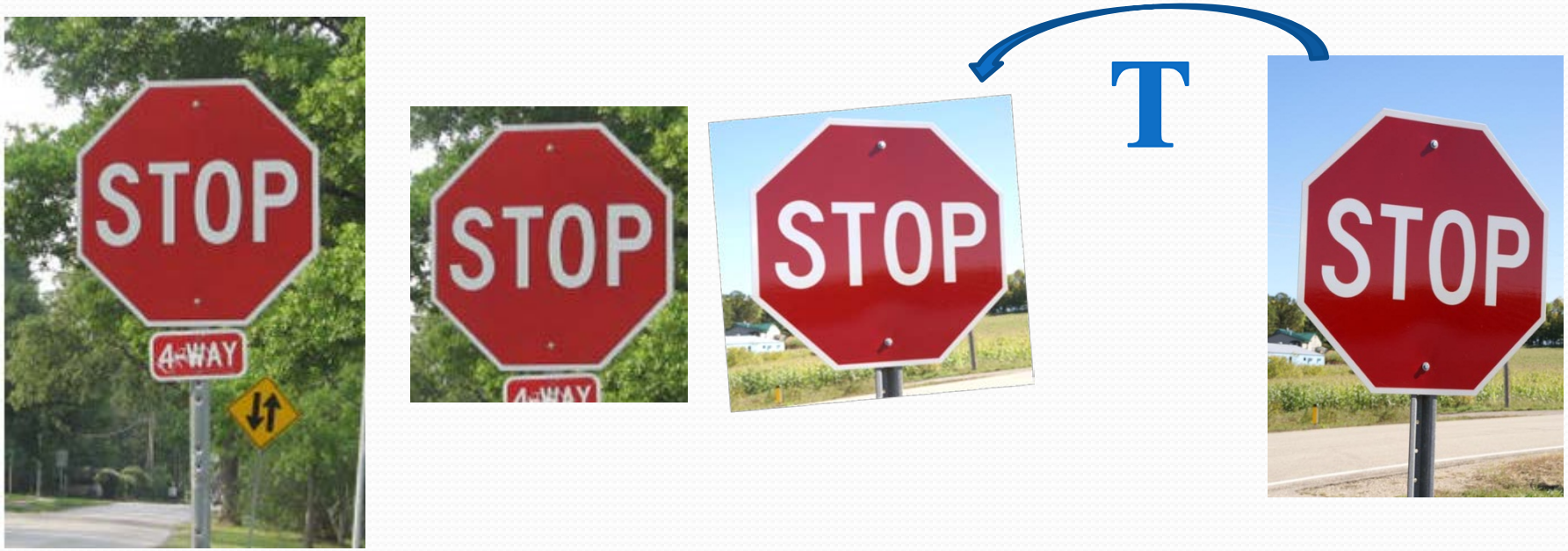
Correspondence and Alignment

- Correspondence: matching points, patches, edges, or regions across images



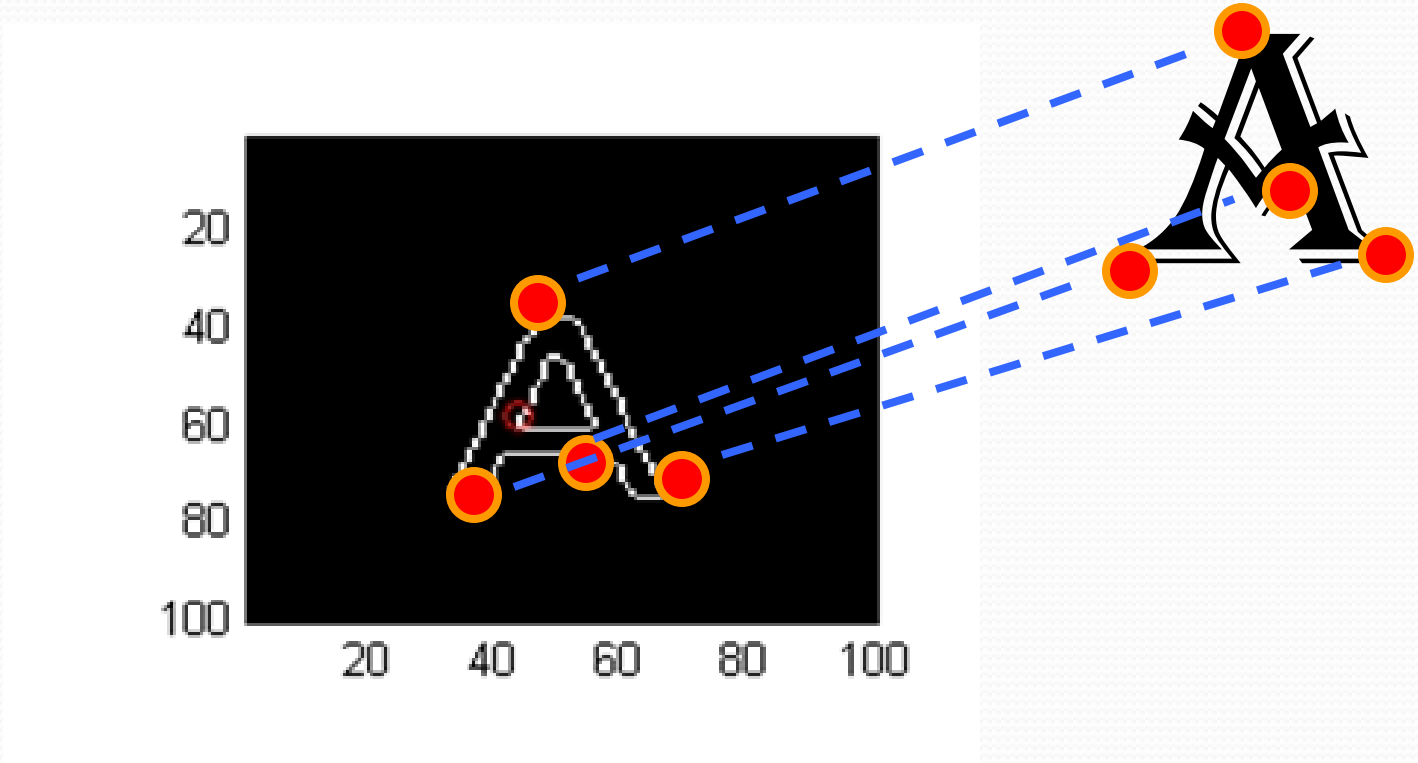
Correspondence and Alignment

- Alignment: solving the transformation that makes two things match better



Correspondence and Alignment

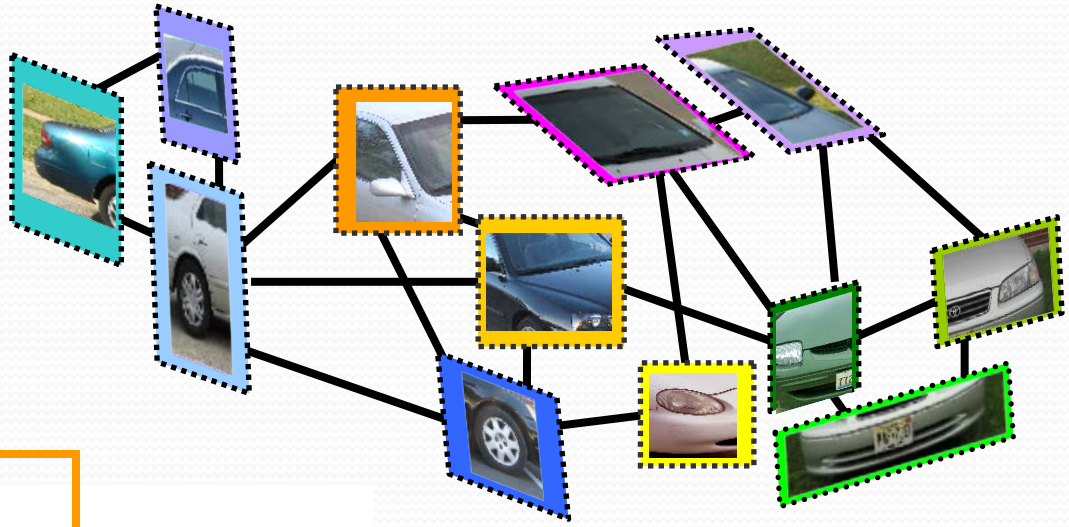
- Example: fitting an 2D shape template



Slide from Silvio Savarese

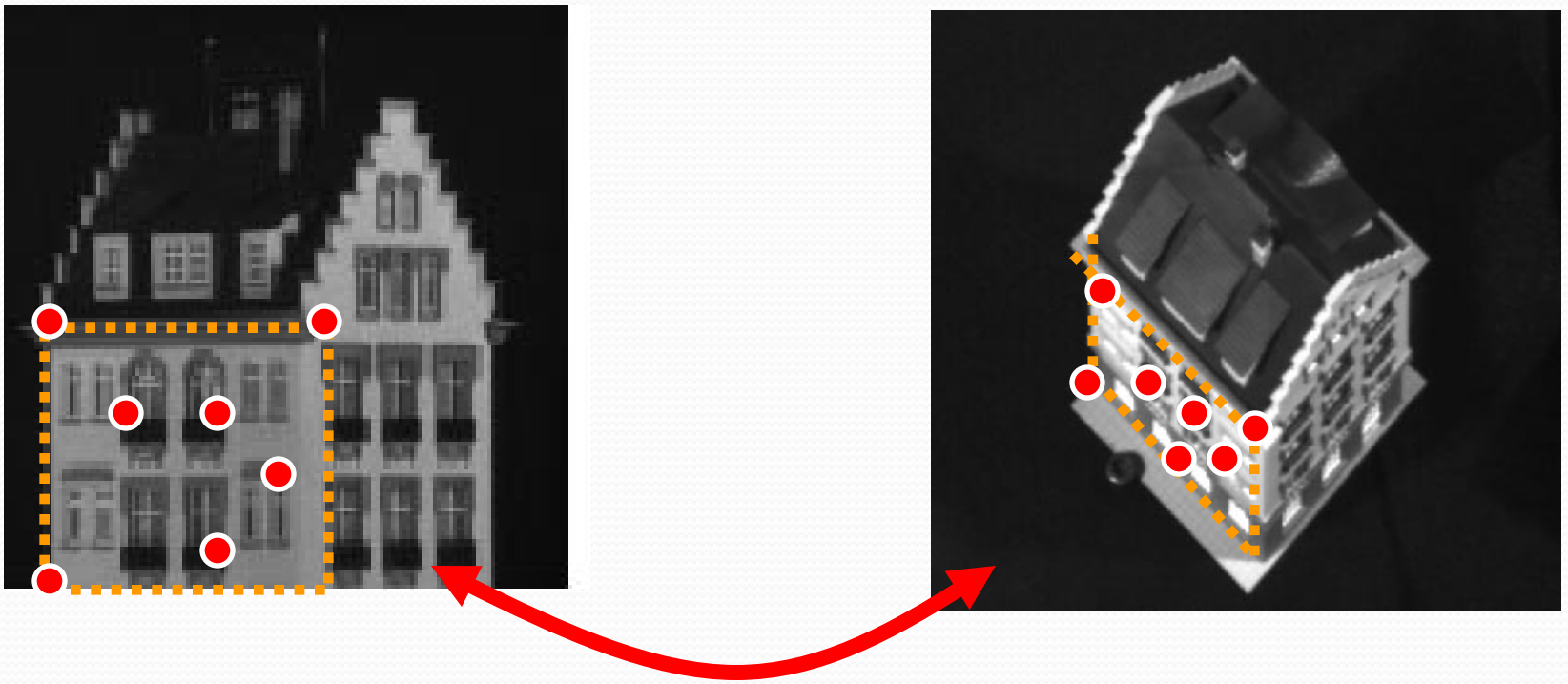
Correspondence and Alignment

- Example: fitting a 3D object model



Slide from Silvio Savarese

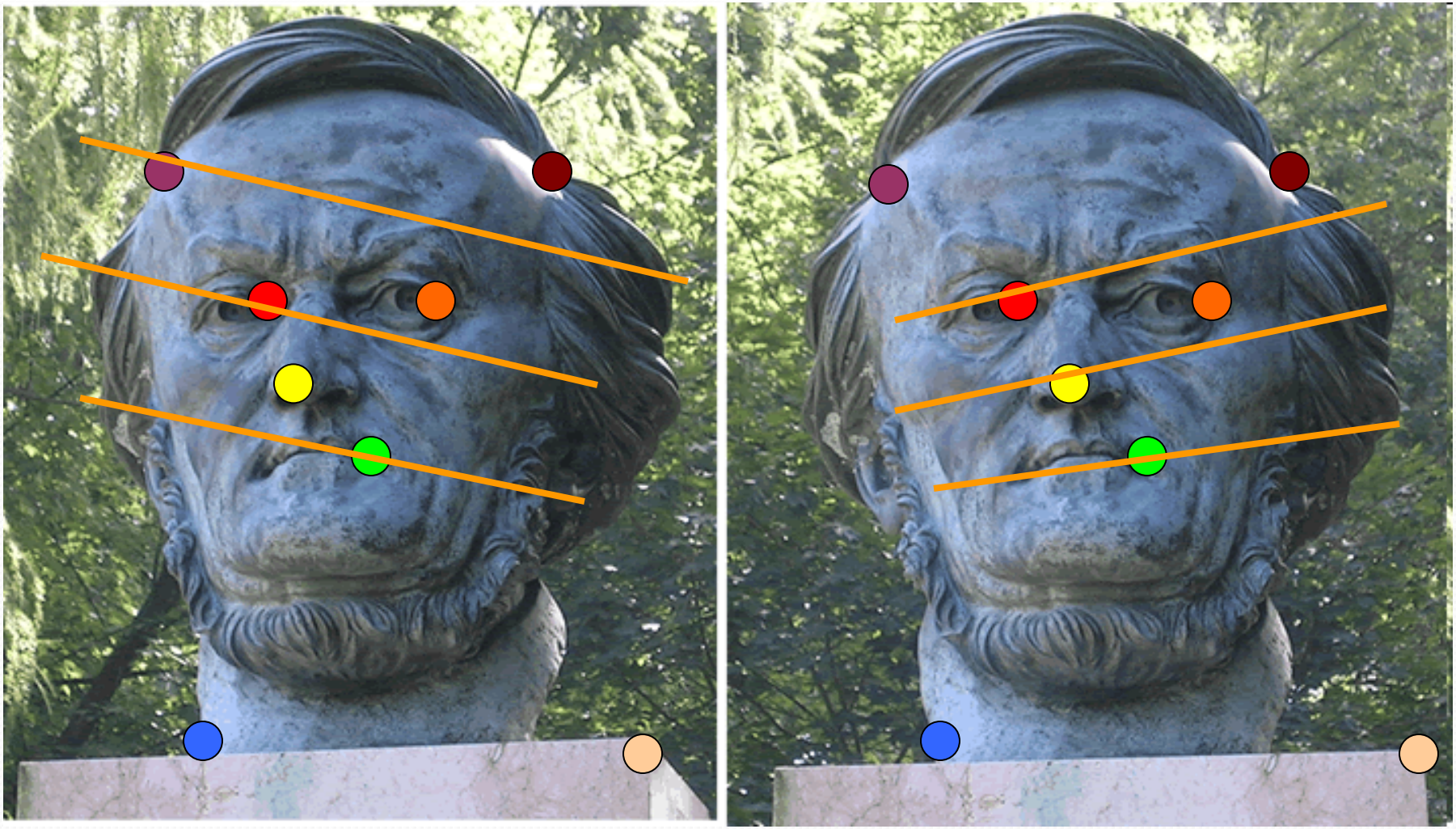
Example: Estimating an homographic transformation



H

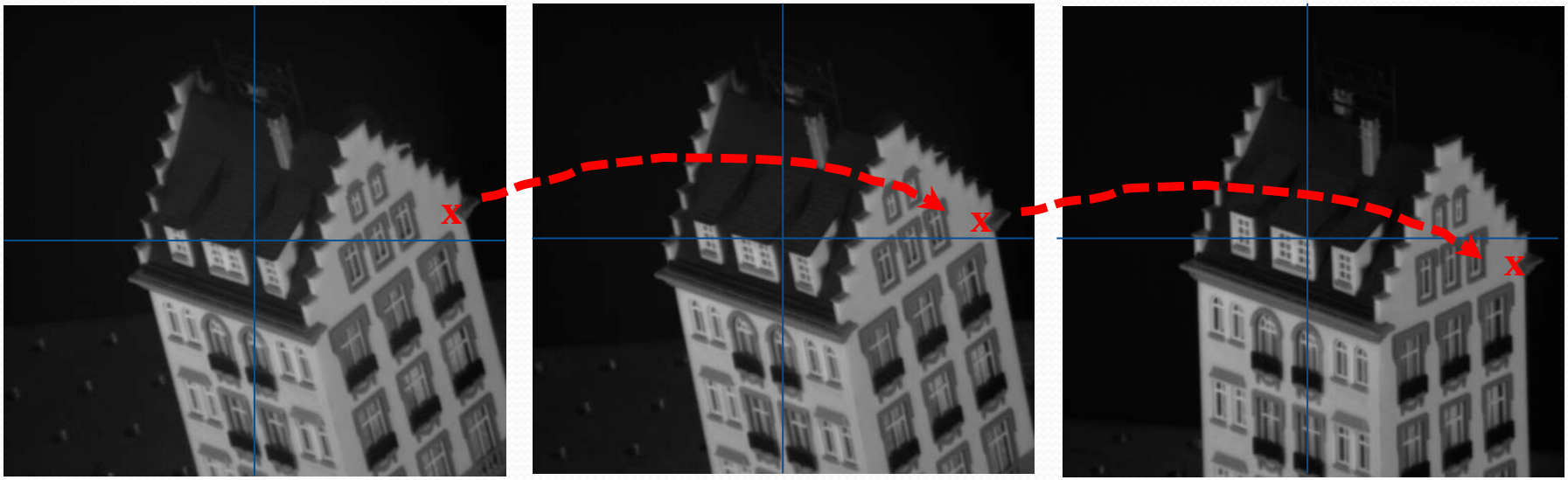
Slide from Silvio Savarese

Example: estimating “fundamental matrix” that corresponds two views



Slide from Silvio Savarese

Example: tracking points

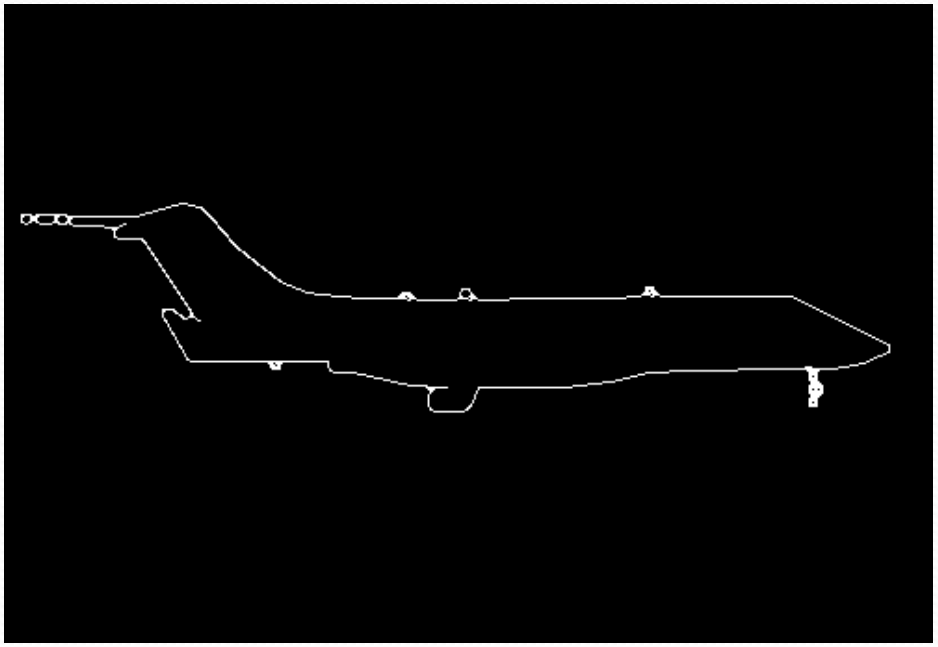


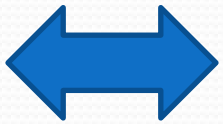
frame 0 frame 22 frame 49

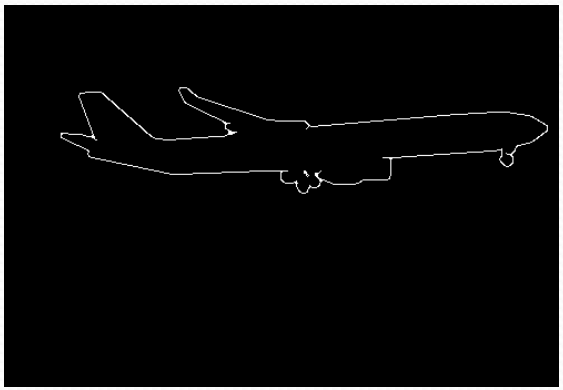
Reverse example of structure from motion

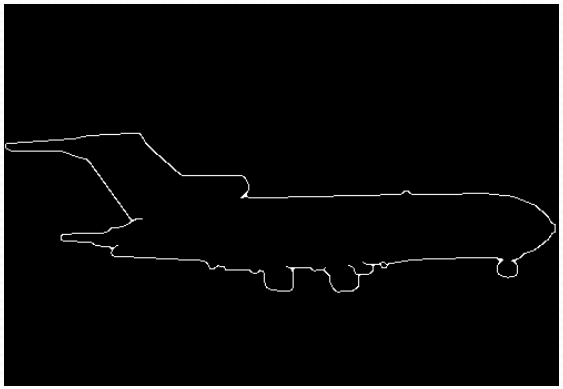
Correspondence and Alignment

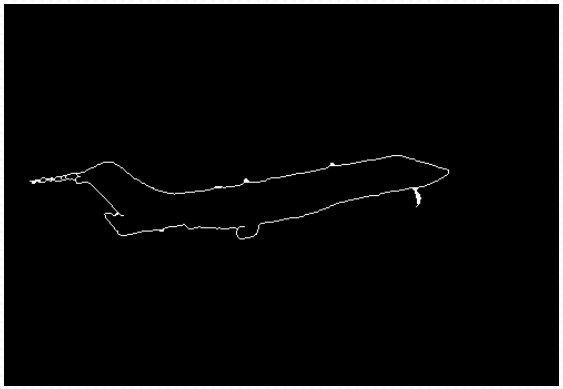
- Alignment of object edge images
  - Compute a transformation that aligns two edge maps









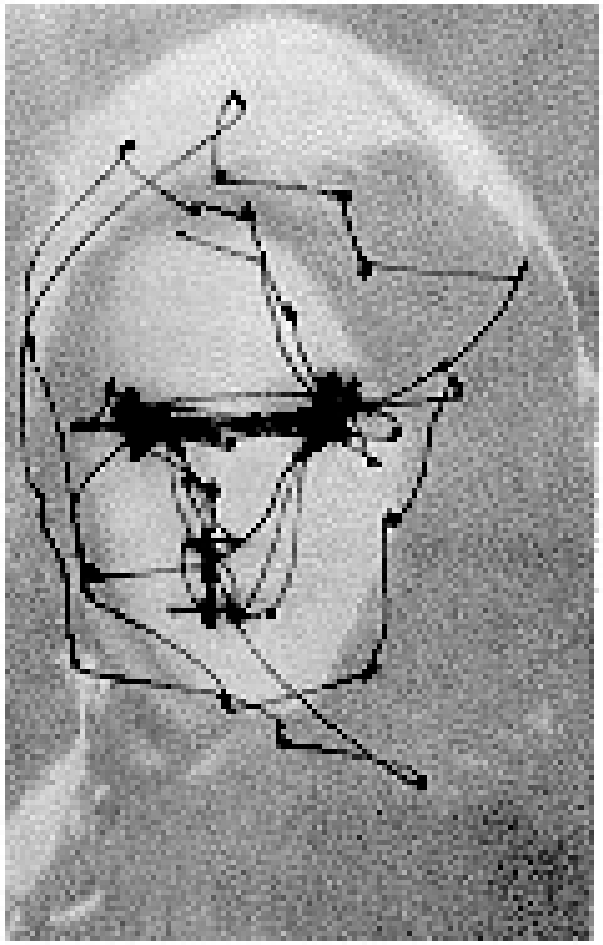


Interest points

- Note: “interest points” = “keypoints”, also sometimes called “features”

- Many applications
  - tracking: which points are good to track?
  - recognition: find patches likely to tell us something about object category
  - 3D reconstruction: find correspondences across different views

Human eye movements

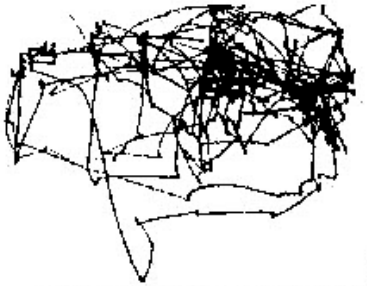


Yarbus eye tracking

Human eye movements

1

Free examination.



2

Estimate material circumstances of the family

3

Give the ages of the people.

4

Surmise what the family had been doing before the arrival of the unexpected visitor.

5

Remember the clothes worn by the people.

6

Remember positions of people and objects in the room.

7

Estimate how long the visitor had been away from the family.

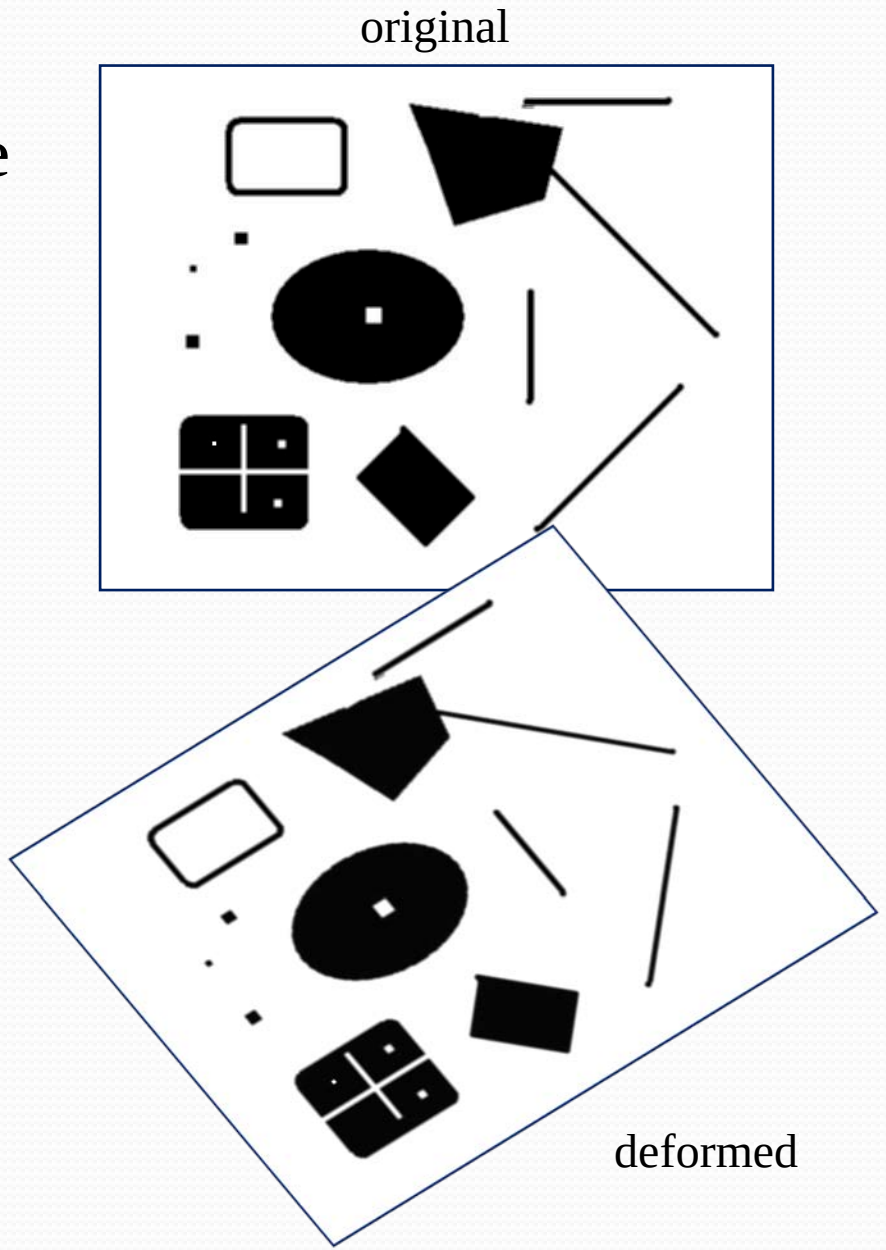
3 min. recordings of the same subject

Study by Yarbus

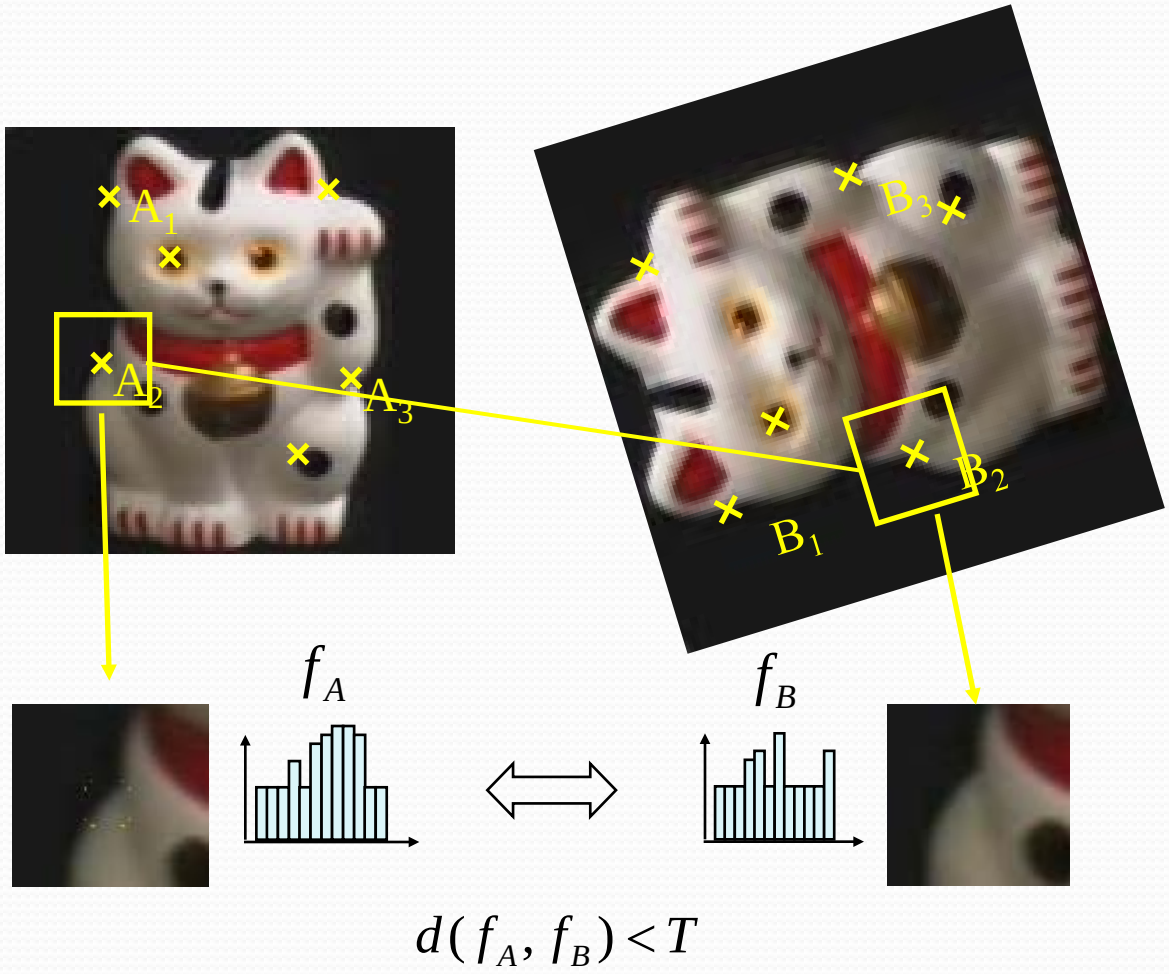
Change blindness: http://www.simonslab.com/videos.html

Interest points

- Suppose you have to click on some point, go away and come back after I deform the image, and click on the same points again.
  - Which points would you choose?



Overview of Keypoint Matching

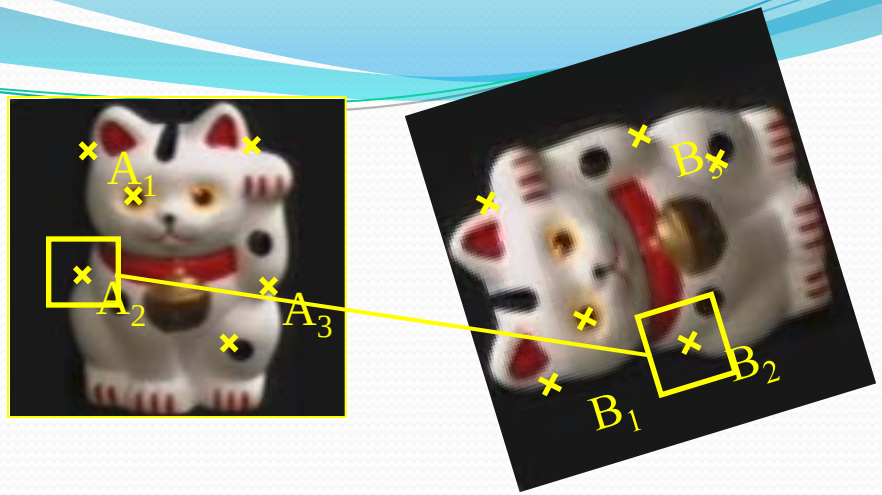


1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

Goals for Keypoints

Detect points that are *repeatable* and *distinctive*

Key trade-offs



Detection

More Repeatable

Robust detection
Precise localization

More Points

Robust to occlusion
Works with less texture

Description

More Distinctive

Minimize wrong matches

More Flexible

Robust to expected variations
Maximize correct matches

Choosing interest points

Where would you tell
your friend to meet you?

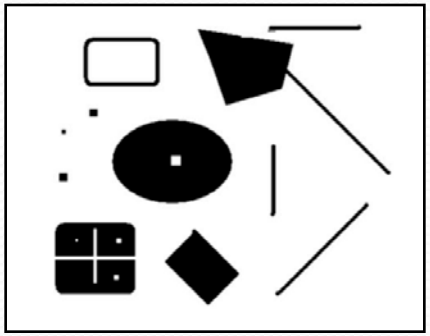
Choosing interest points

Where would you tell
your friend to meet you?

Many Existing Detectors Available

Hessian & Harris	[Beaudet '78], [Harris '88]
Laplacian, DoG	[Lindeberg '98], [Lowe 1999]
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…	

Harris Detector [Harris88]



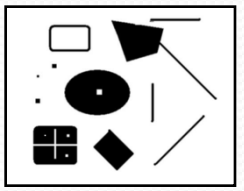
- Second moment matrix (autocorrelation)

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

Intuition: **Search for local neighborhoods where the image content has two main directions (eigenvectors).**

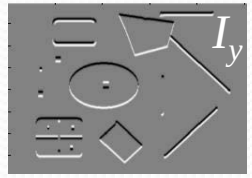
Harris Detector [Harris88]

- Second moment matrix (autocorrelation)

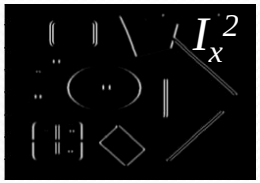


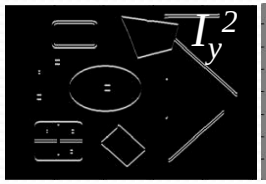
$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

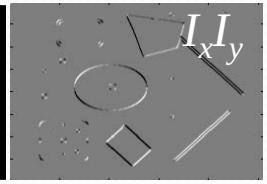
1. Image derivatives (***optionally, blur first***)



2. Square of derivatives

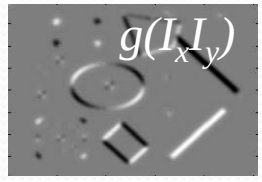






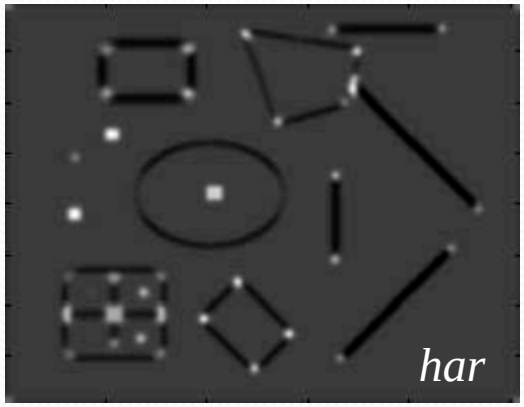
$$\det M = \lambda_1\lambda_2$$
$$\text{trace}\, M = \lambda_1 + \lambda_2$$

3. Gaussian filter $g(\sigma_I)$



4. **Cornerness function – both eigenvalues are strong**

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



5. Non-maxima suppression

Harris Detector: Mathematics

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

1. Empirically, standard deviation $\sigma_I = 2\,\sigma_D$
2. Want large eigenvalues, and small ratio $\frac{\lambda_1}{\lambda_2} < t$
3. We know $\det M = \lambda_1 \lambda_2$

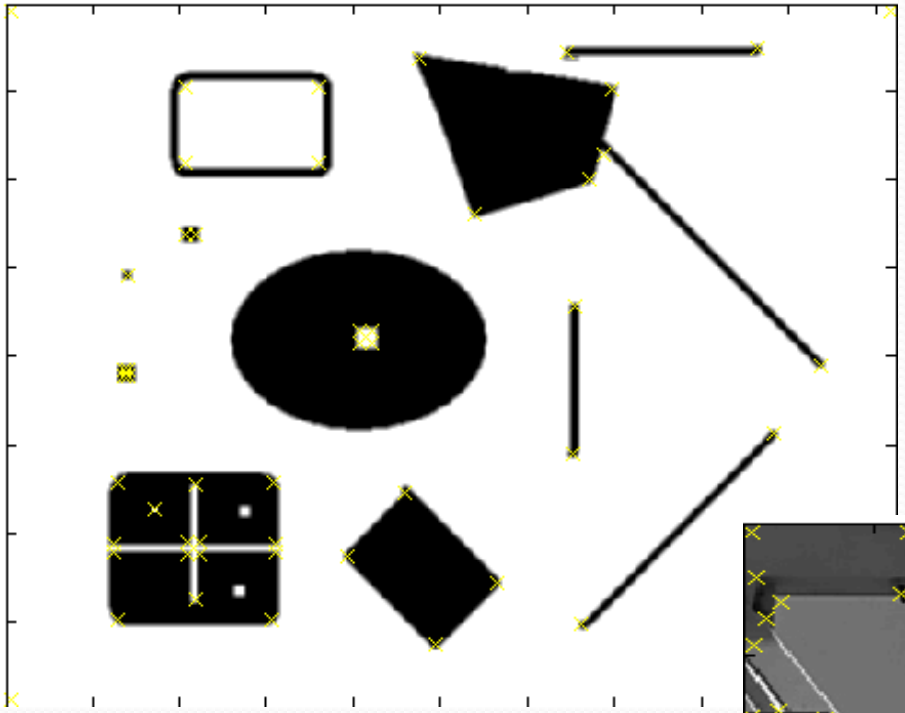
   $\text{trace}\, M = \lambda_1 + \lambda_2$
4. Leads to

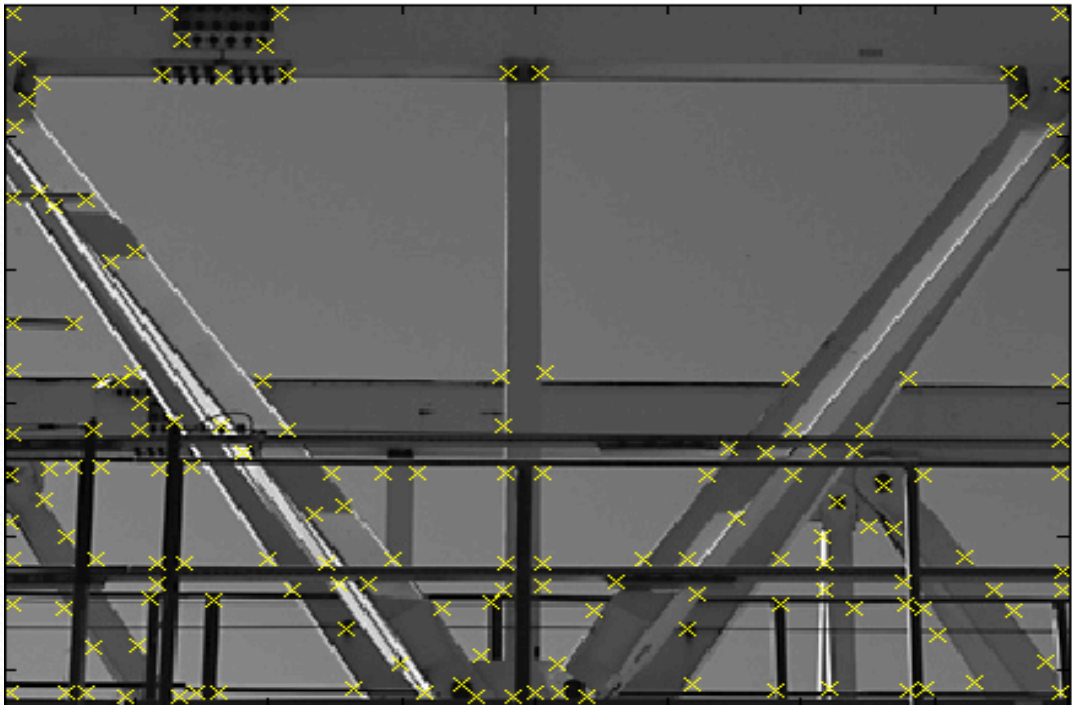
$$\det M - k \cdot \text{trace}^2(M) > t$$

(k :empirical constant, k = 0.04-0.06)

Nice brief derivation on wikipedia

Harris Detector – Responses [Harris88]





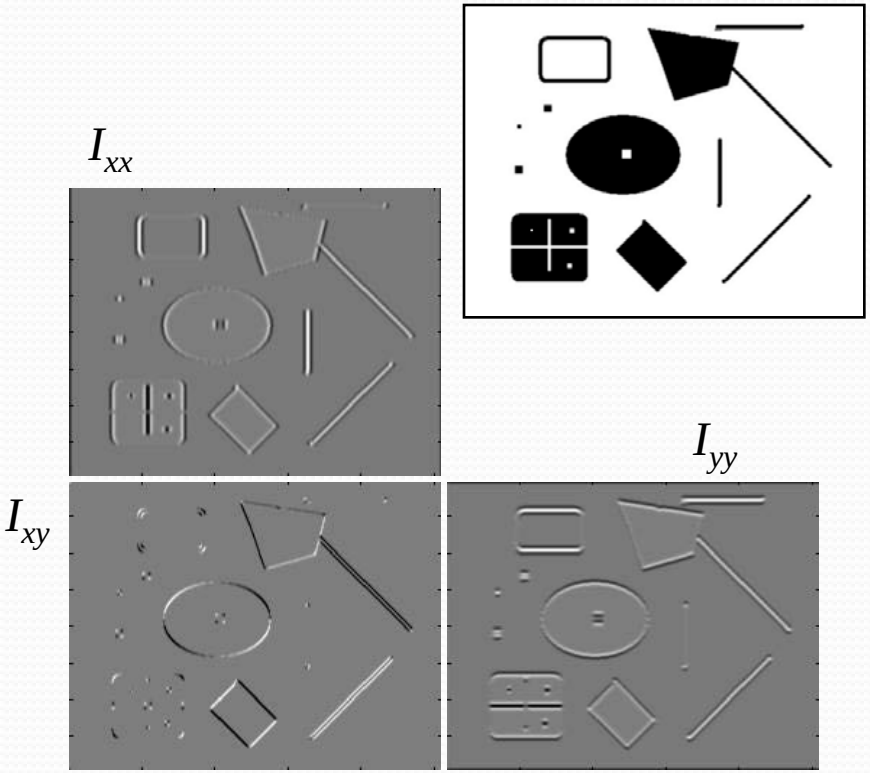
Effect: **A very precise corner detector.**

Harris Detector - Responses [Harris88]

Hessian Detector [Beaudet78]

- Hessian determinant

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



Intuition: **Search for strong curvature in two orthogonal directions**

Hessian Detector [Beaudet78]

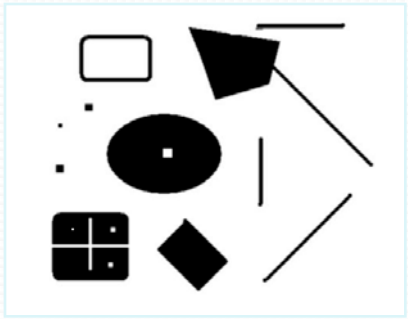
- Hessian determinant

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

$$\det M = \lambda_1 \lambda_2$$

$$\text{trace}\, M = \lambda_1 + \lambda_2$$

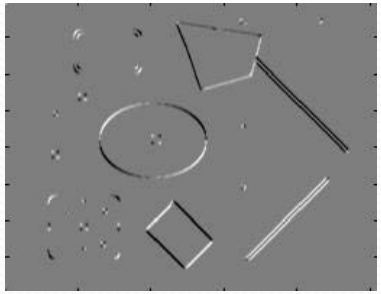
$$\det(Hessian(I)) = I_{xx} I_{yy} - I_{xy}^2$$

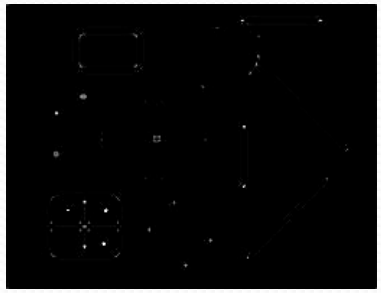


I_{xx}

I_{yy}

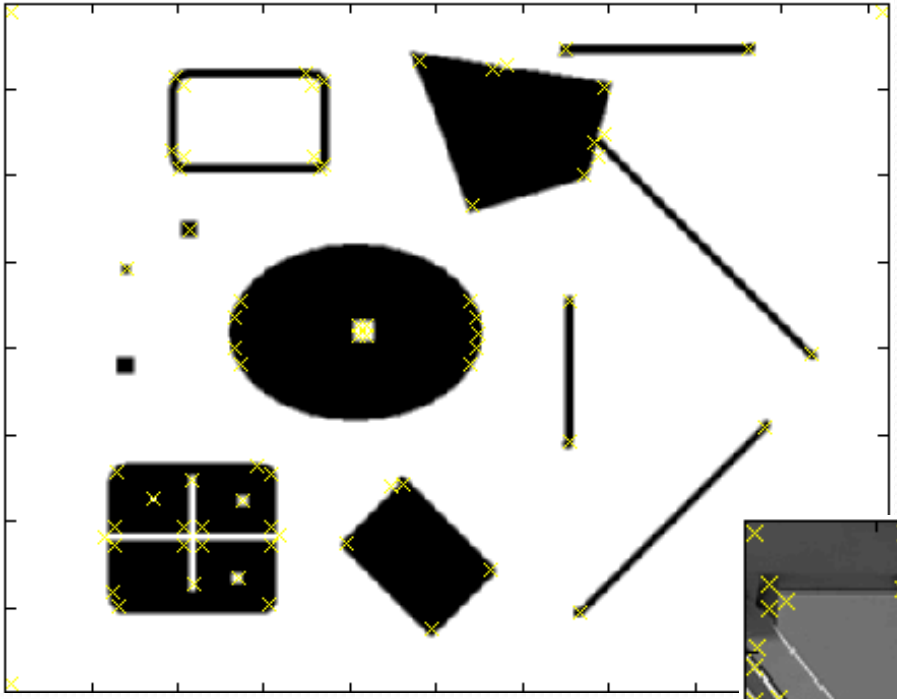
I_{xy}

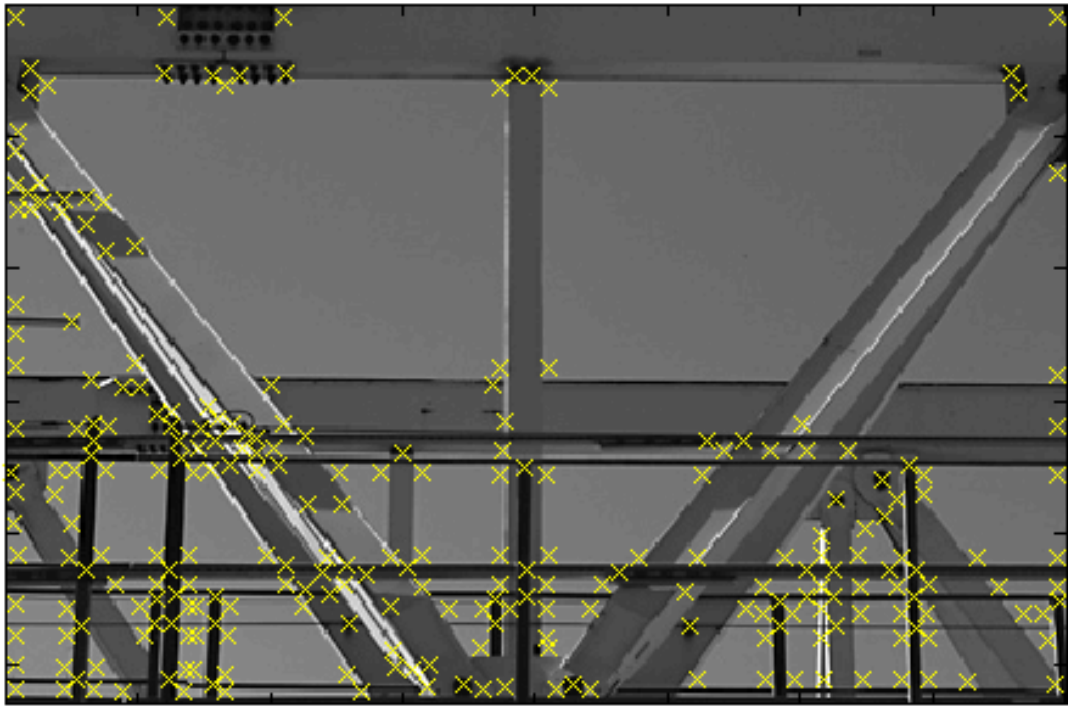




Hessian Detector – Responses

[Beaudet78]

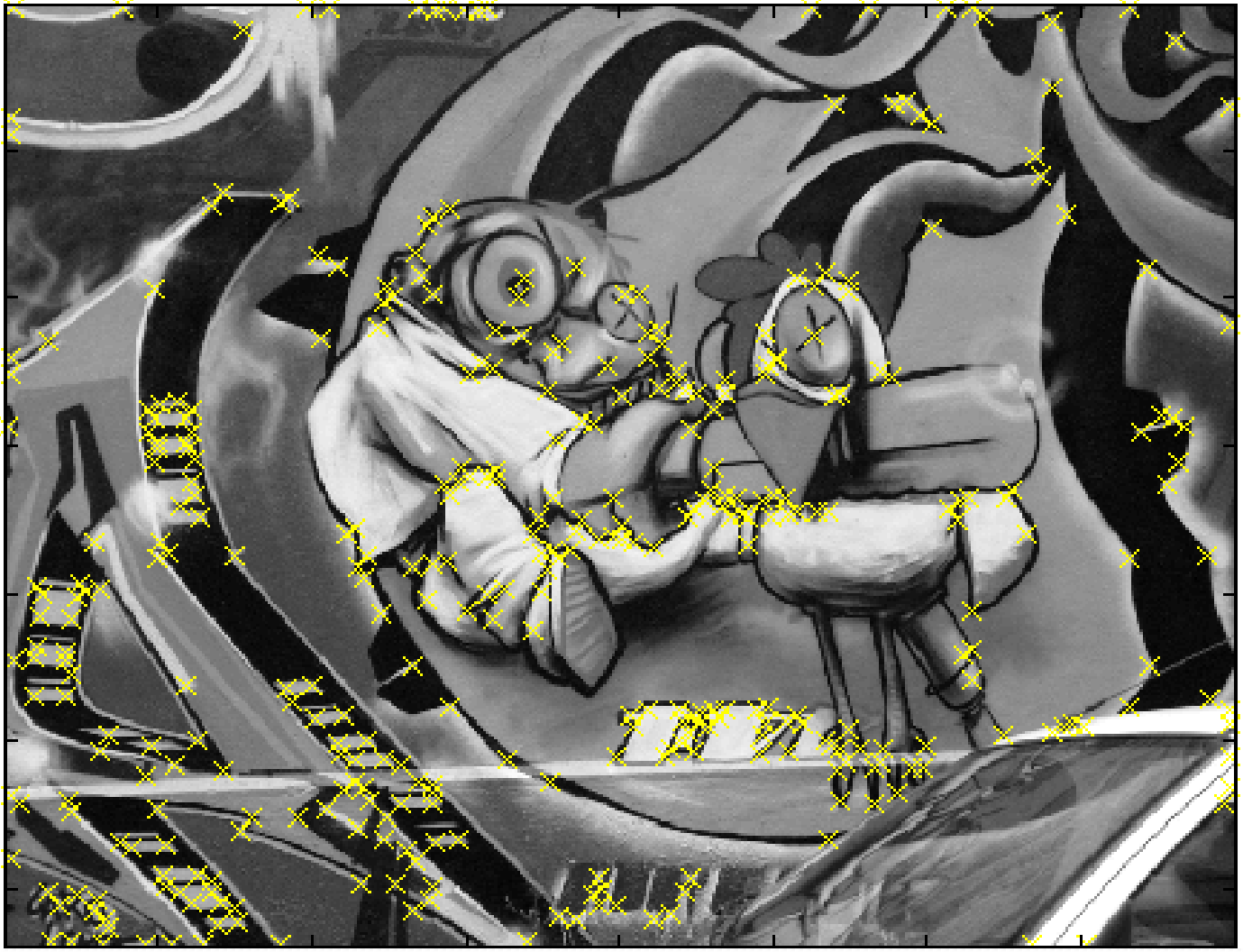




Effect: **Responses mainly on corners and strongly textured areas.**

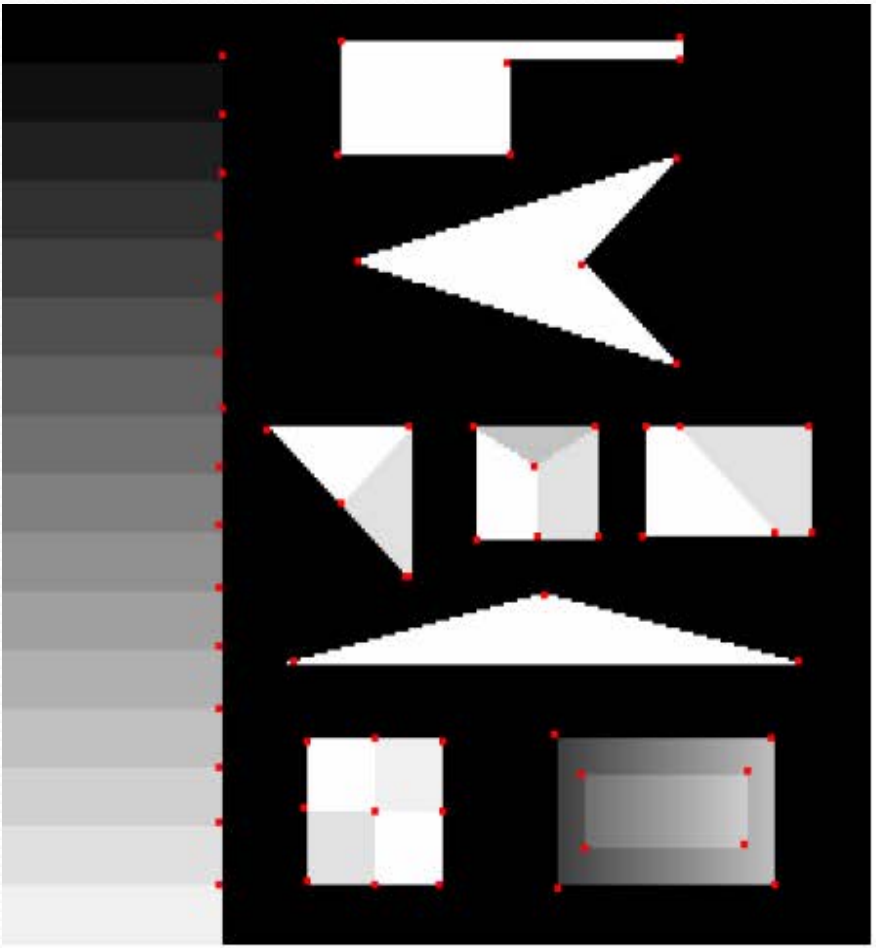
Hessian Detector – Responses

[Beaudet78]



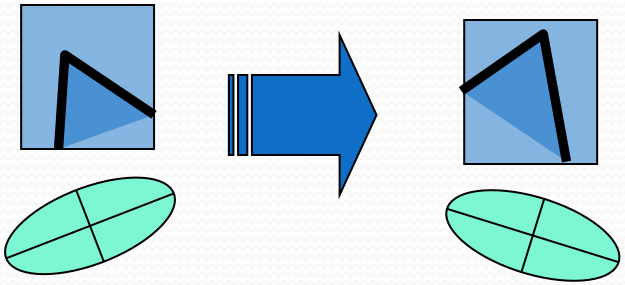
So far: can localize in x-y, but not scale

Harris Detector [Harris88]



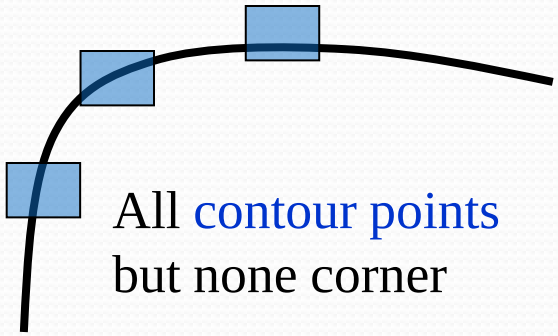
Harris Invariances

- Invariant to translation, rotations, affine illumination changes

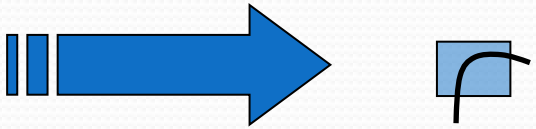


The shape of the ellipse remains the same

- BUT no scale invariance !



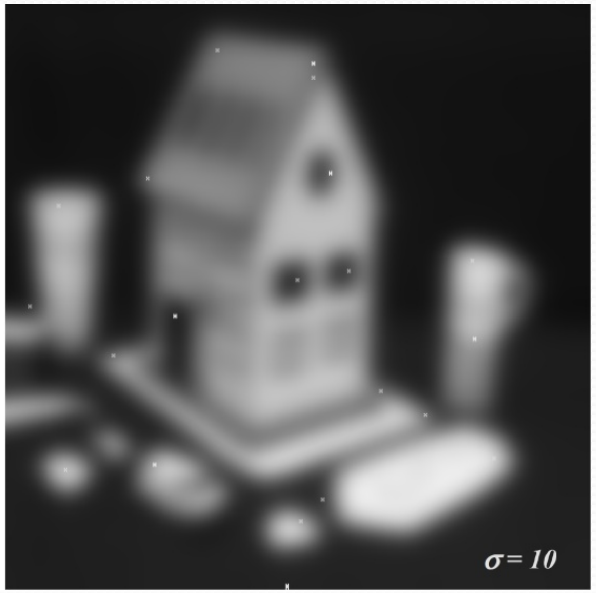
All contour points
but none corner



corner

Multi-scale Harris detector

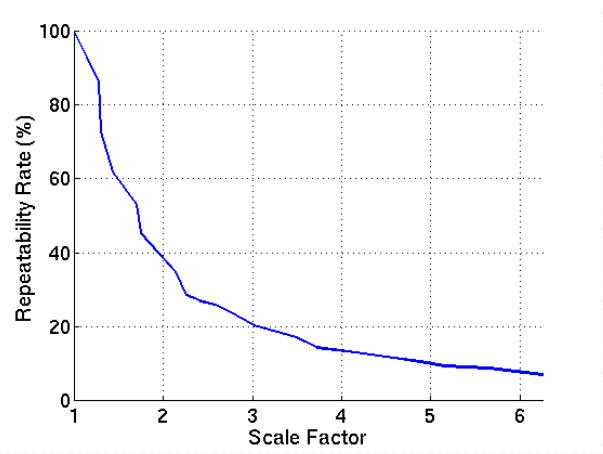
Harris points obtained by computing derivatives through convolution with gaussian derivative of standard deviation σ



Harris detector and scale changes

- How does the Harris dectetor reacts when the scale of the image is changed ?

- Example of direct detector application:
  - Detector application on each image
  - Are the same detected points ?
  - At the same location ?

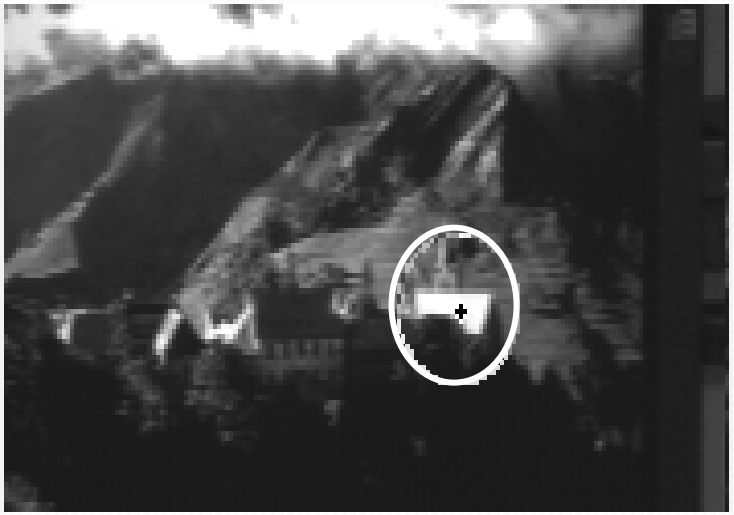


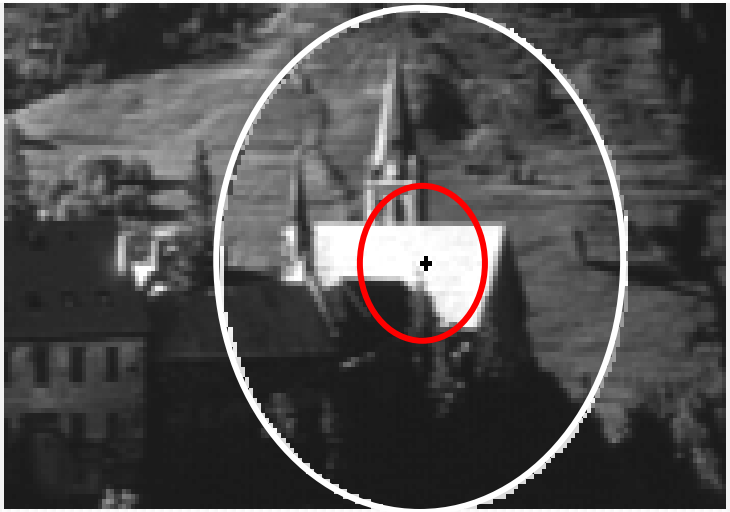
- Consistency evaluation between detections: Repeatability Rate which provides for each pair of images the percentage of detected that are correctly localized (up to a distance threshold)

=> Quick degradation of the results, scale adaptation required

Multi-scale detection problem

Multi-scale detection problem





Automatic Scale Selection

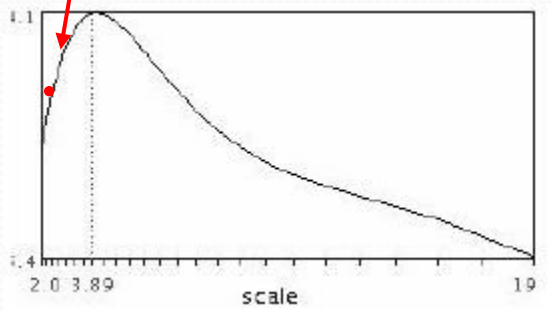
$$f(I_{i_1 \ldots i_m}(x,\sigma)) \;=\; f(I_{i_1 \ldots i_m}(x',\sigma'))$$

Can 2 reponses of the same function f be identical if the patch contains the same image up to a scale factor of proportionality?

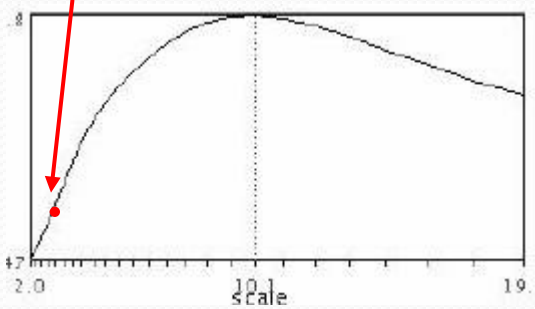
How to find corresponding patch sizes?

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



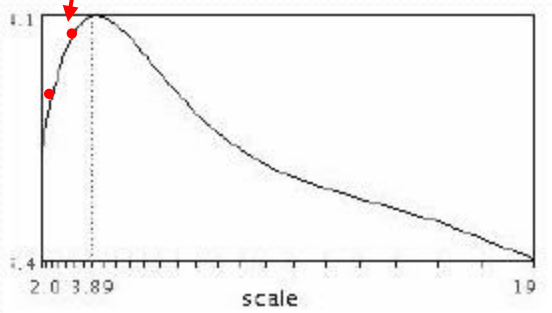
$f(I_{i_1 \ldots i_m}(x,\sigma))$



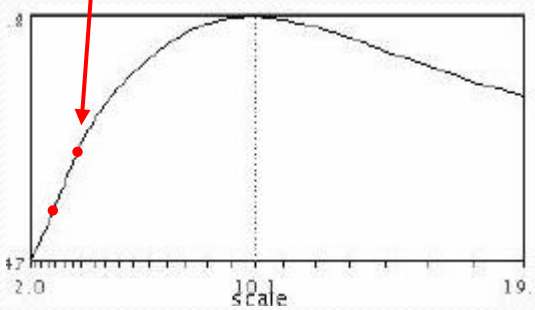
$f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



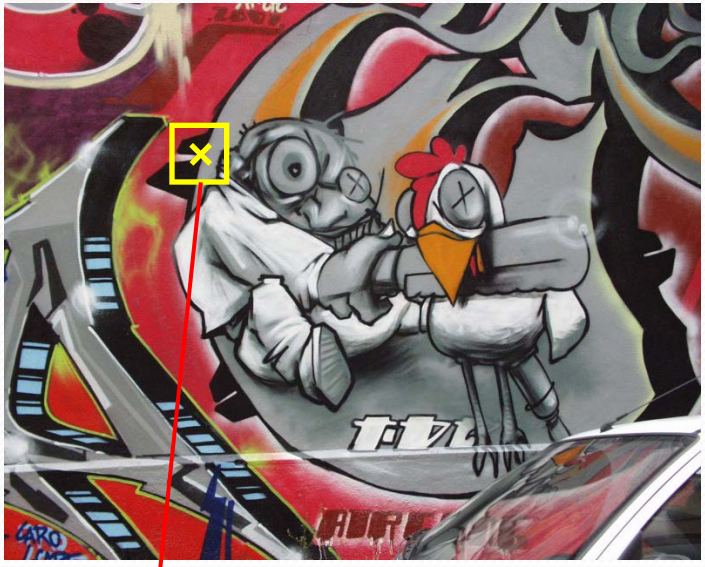
$f(I_{i_1 \ldots i_m}(x,\sigma))$

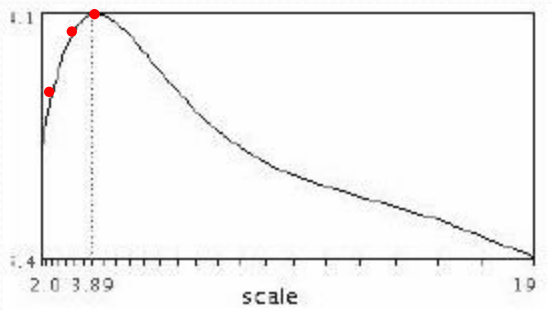


$f(I_{i_1 \ldots i_m}(x',\sigma))$

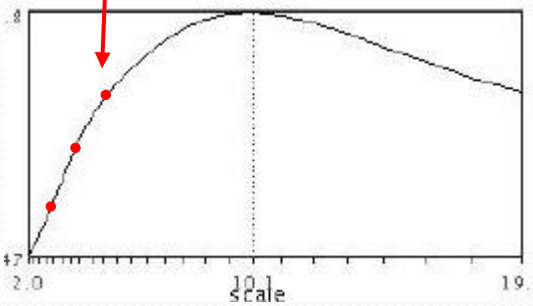
Automatic Scale Selection

- Function responses for increasing scale (scale signature)





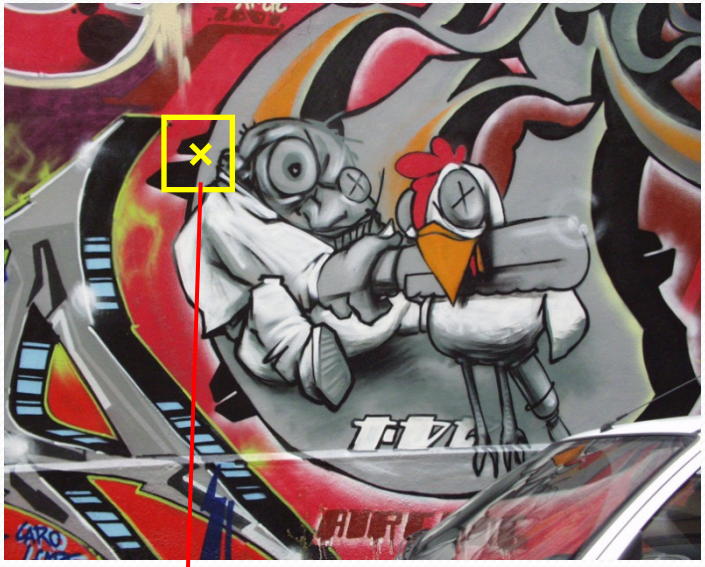
$f(I_{i_1 \ldots i_m}(x,\sigma))$

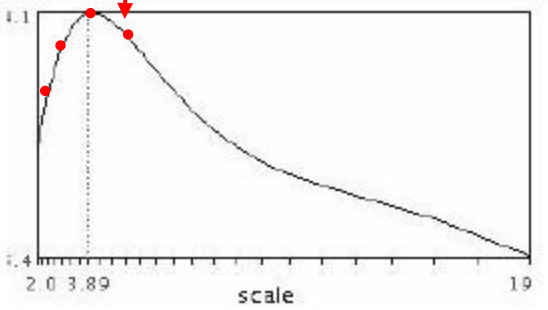


$f(I_{i_1 \ldots i_m}(x',\sigma))$

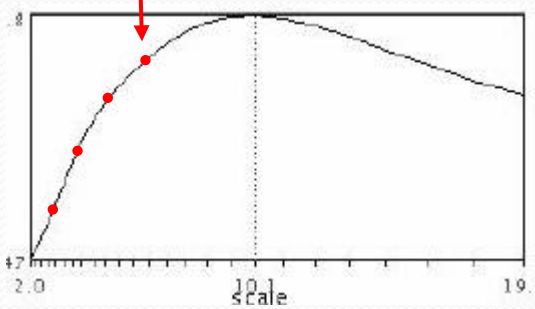
Automatic Scale Selection

- Function responses for increasing scale (scale signature)





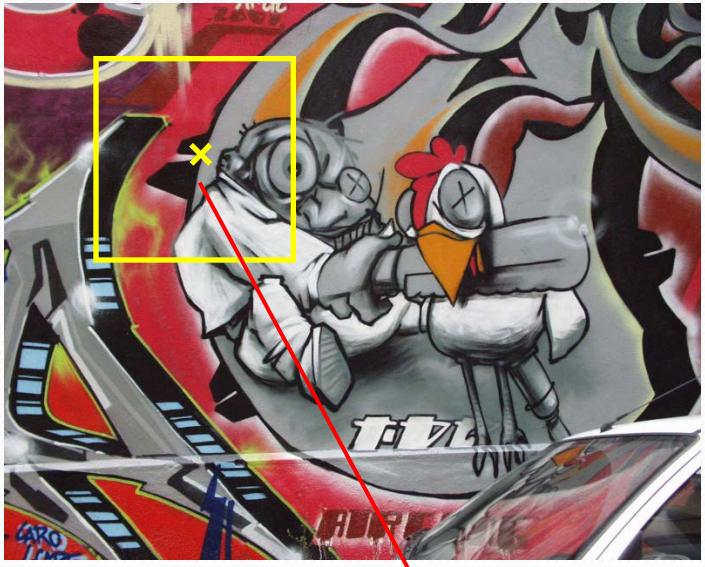
$f(I_{i_1 \ldots i_m}(x,\sigma))$

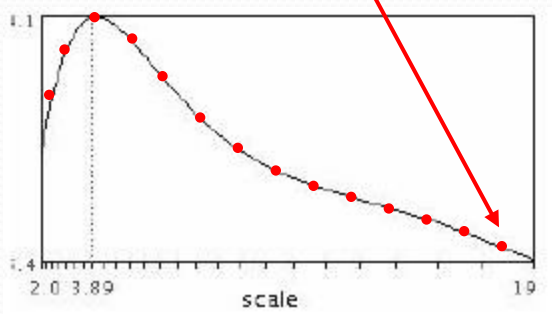


$f(I_{i_1 \ldots i_m}(x',\sigma))$

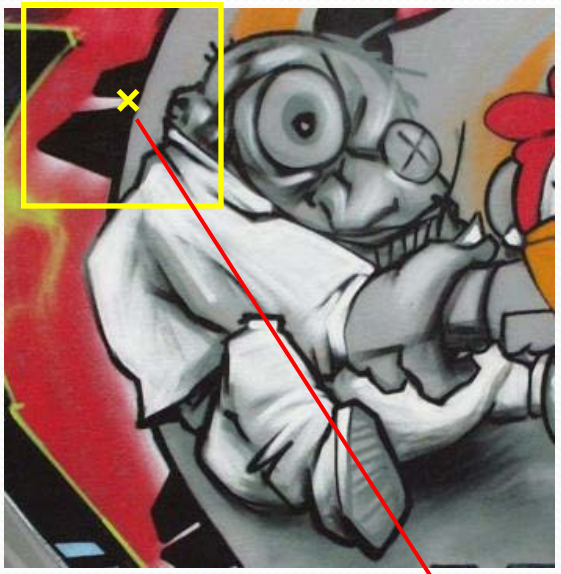
Automatic Scale Selection

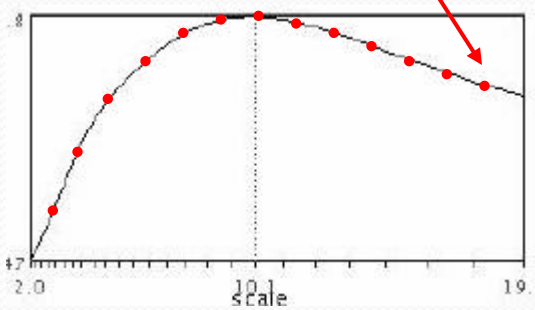
- Function responses for increasing scale (scale signature)





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

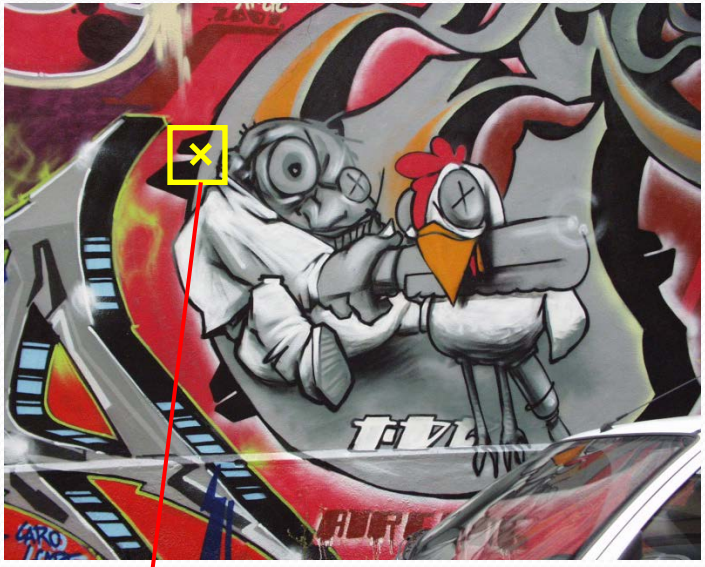


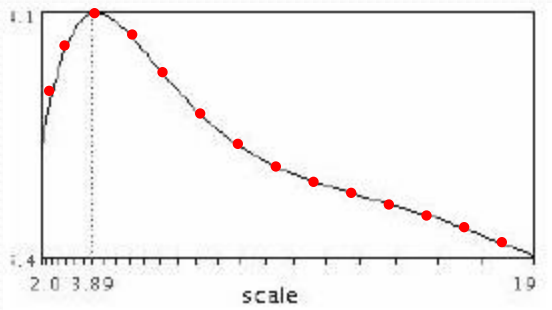


$f(I_{i_1 \ldots i_m}(x',\sigma))$

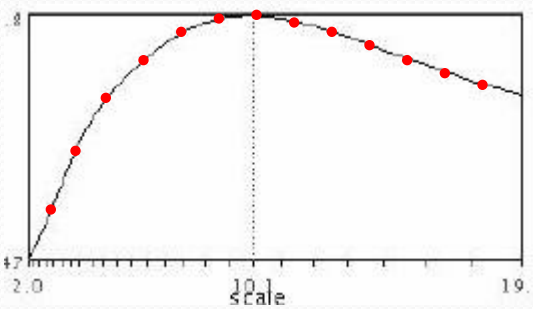
Automatic Scale Selection

- Function responses for increasing scale (scale signature)





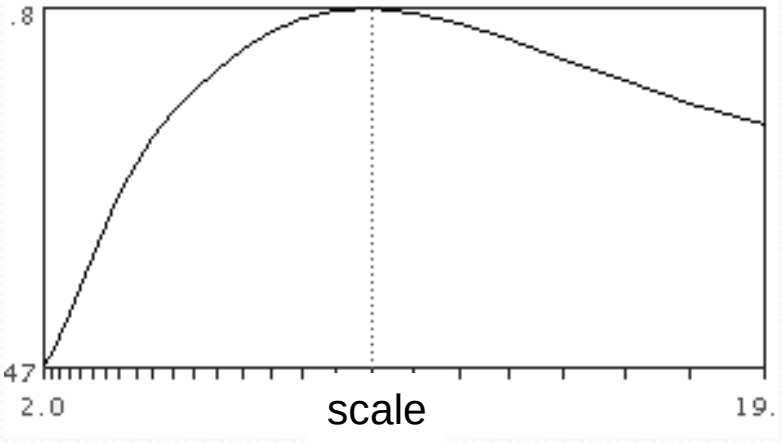
$f(I_{i_1 \ldots i_m}(x,\sigma))$



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

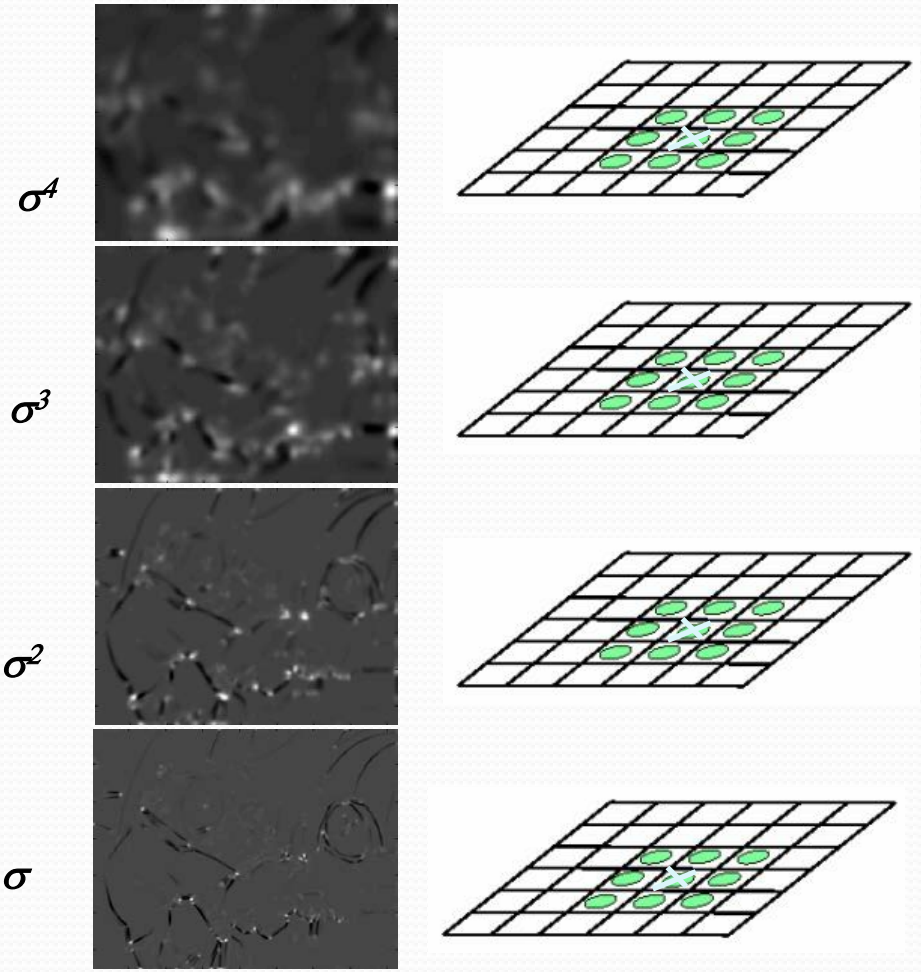
Principles of multi-scale detectors

1. For each point compute *f* (i.e. *gradient*, *laplacien*, etc.) for several scales
2. Normalize values with respect to the scale
3. Select optimal scale: detection of the maximum



Harris-Laplace [Mikolajczyk '01]

1. Initialization: Multiscale Harris corner detection

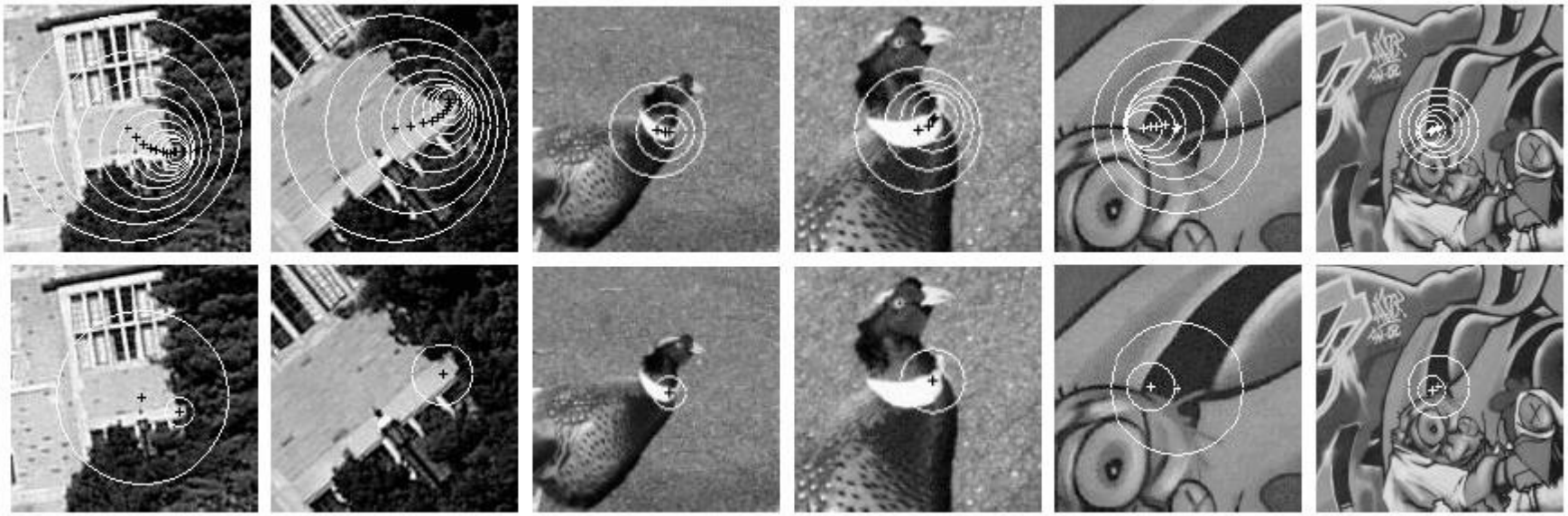


Computing Harris function **Detecting local maxima**

Harris-Laplace [Mikolajczyk '01]

1. Initialization: Multiscale Harris corner detection
2. Scale selection based on Laplacian
   (same procedure with Hessian ⇒ Hessian-Laplace)

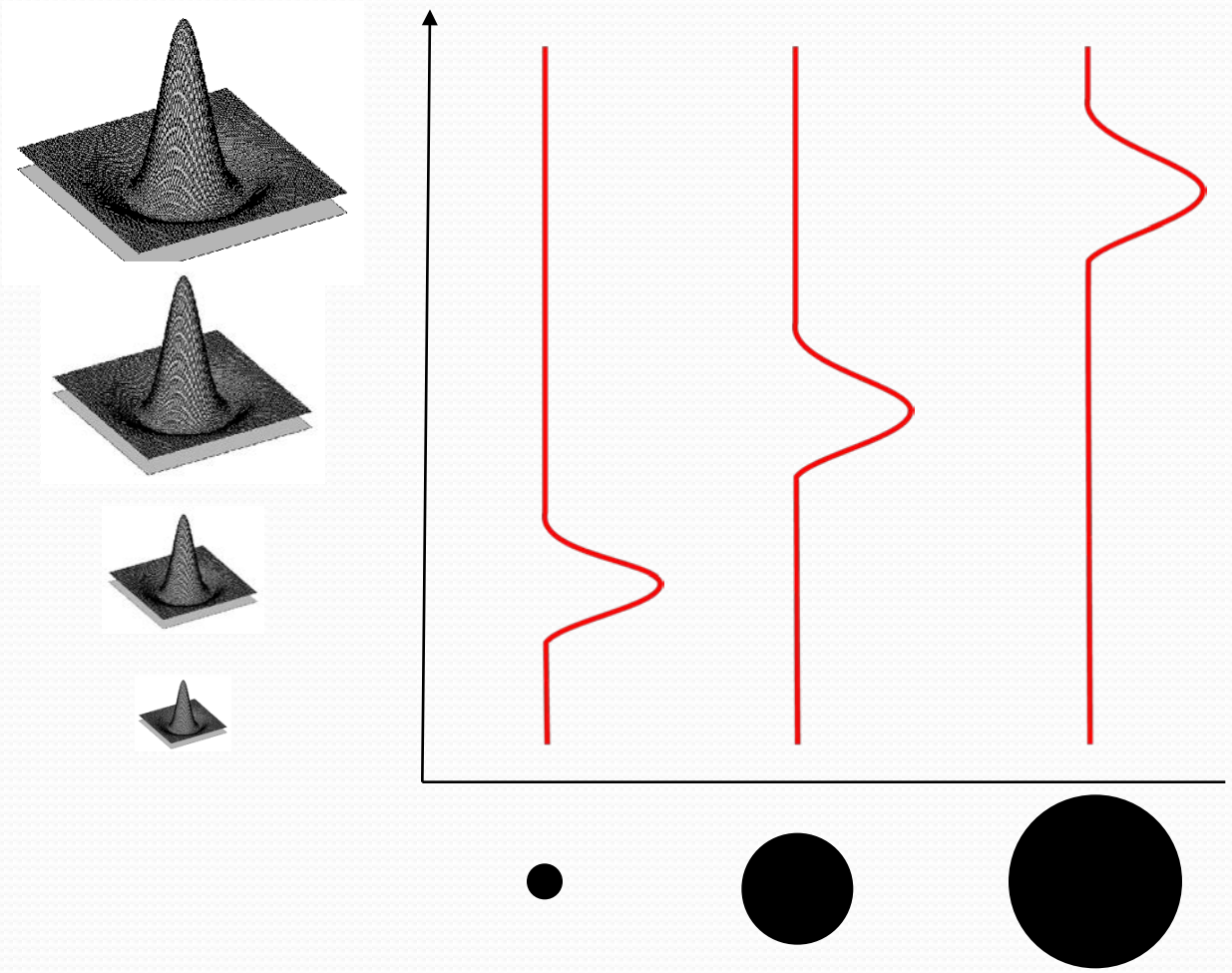
Harris points



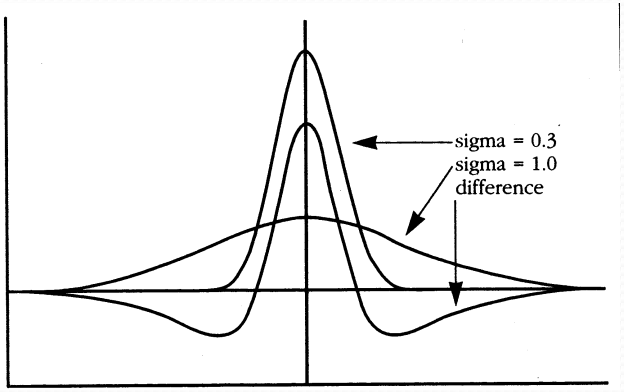
Harris-Laplace points

Other mutli-scale detectors?

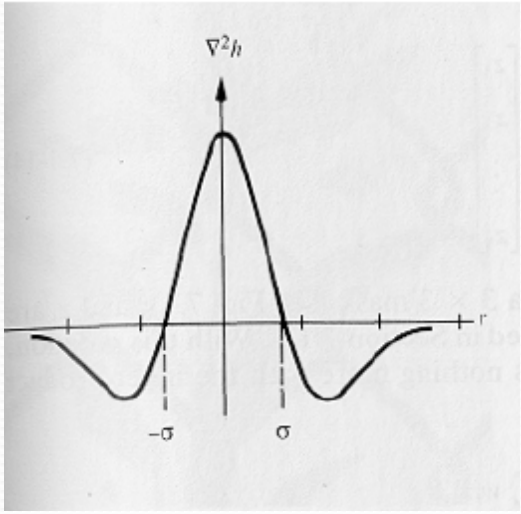
- Difference-of-Gaussian = “blob” detector



Difference-of-Gaussian (DoG)



Difference of Gaussian (DoG)
Approximation of LoG



LoG

$$\nabla^2 G \approx G(x, y; \sigma_1) - G(x, y; \sigma_2)$$

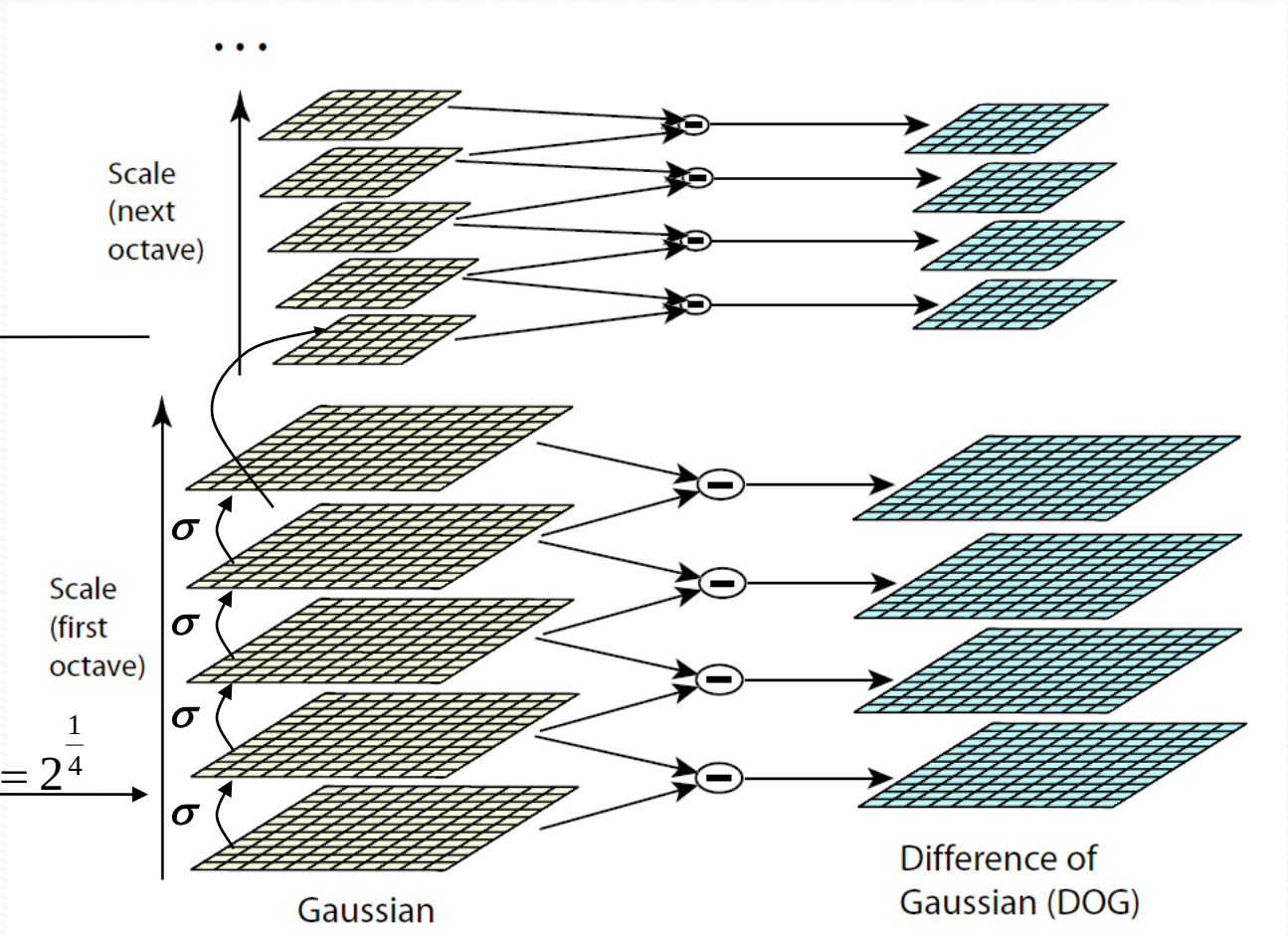
DoG – Efficient Computation

- Computation in Gaussian scale pyramid

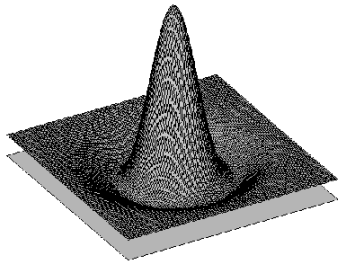
Sampling with step $\sigma^4 = 2$

Original image

$\sigma = 2^{\frac{1}{4}}$



Find local maxima in position-scale space of Difference-of-Gaussian



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

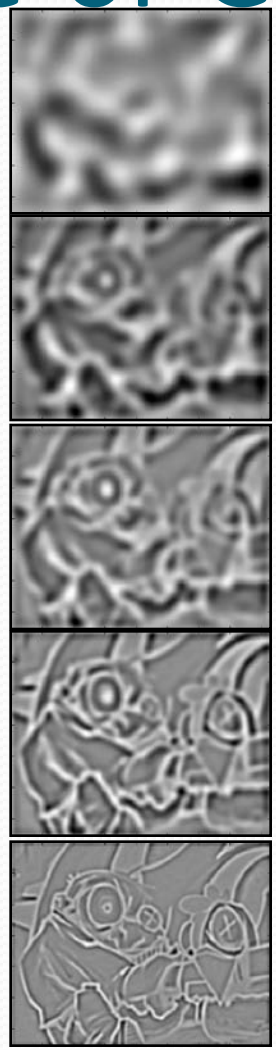
σ^5

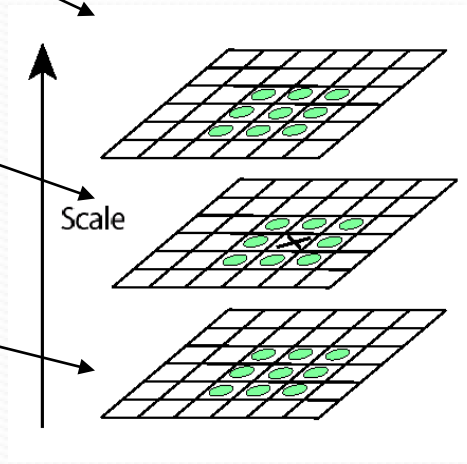
σ^4

σ^3

σ^2

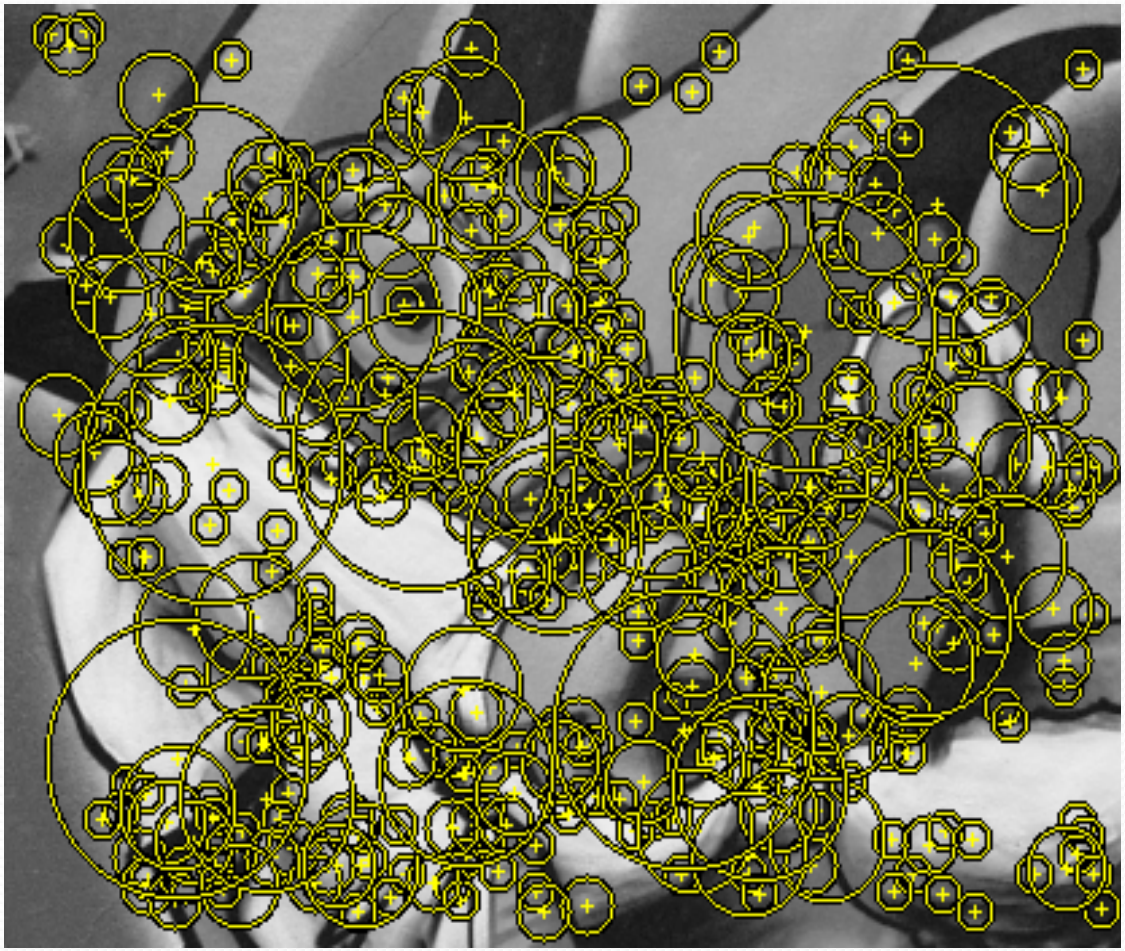
σ





⇒ **List of**
(x, y, s)

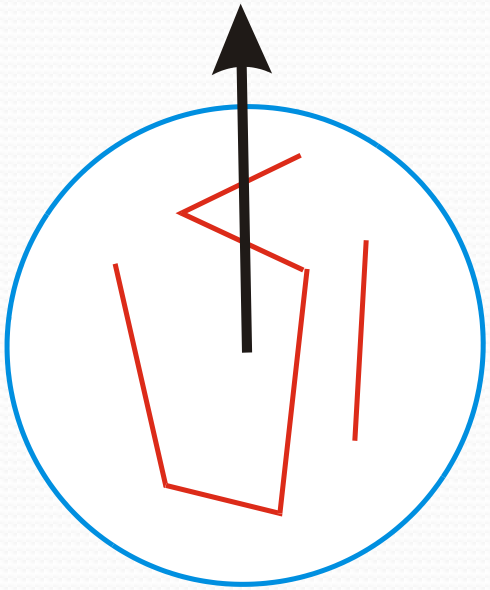
Results: Difference-of-Gaussian

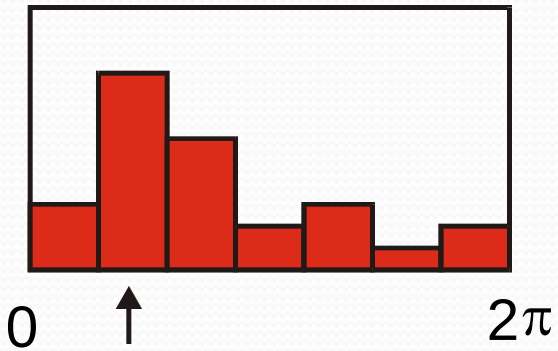


Orientation Normalization

[Lowe, SIFT, 1999]

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

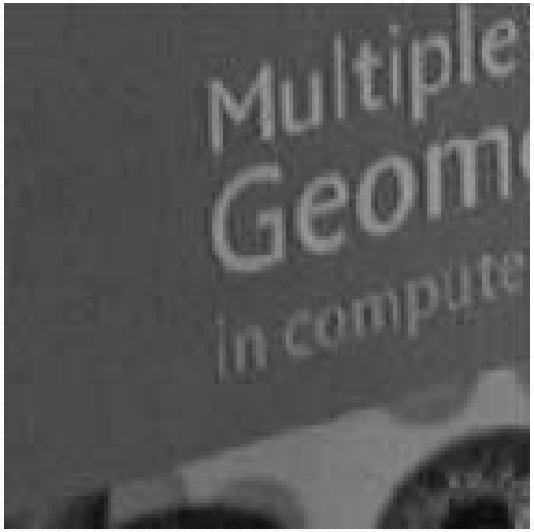


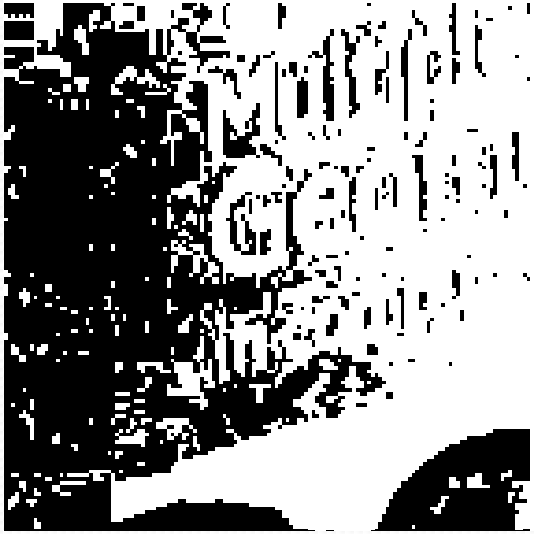


Maximally Stable Extremal Regions

[Matas '02]

- Based on Watershed segmentation algorithm
- Select regions that stay stable over a large parameter range





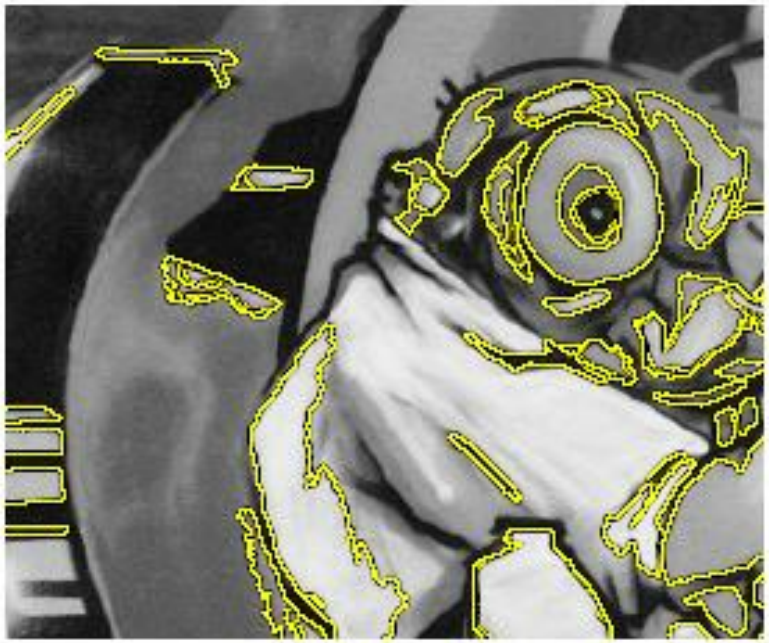
Maximally Stable Extremal Regions

[Matas '02]

$$\forall p \in R_i \, , \forall q \in boundary(R_i) \rightarrow I_{in}(p) \geq I_{in}(q)$$

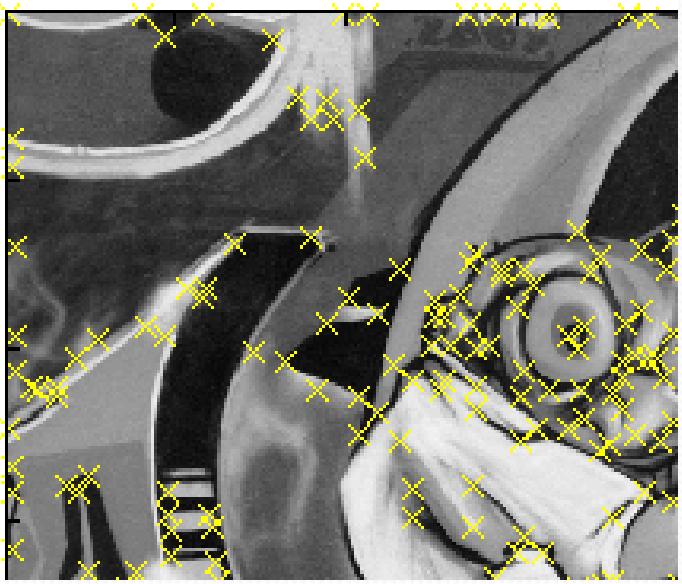
$$I_{bin}^{g} = \begin{cases} 1 & I_{in} \geq g \\ 0 & otherwise \end{cases}$$

Example Results: MSER

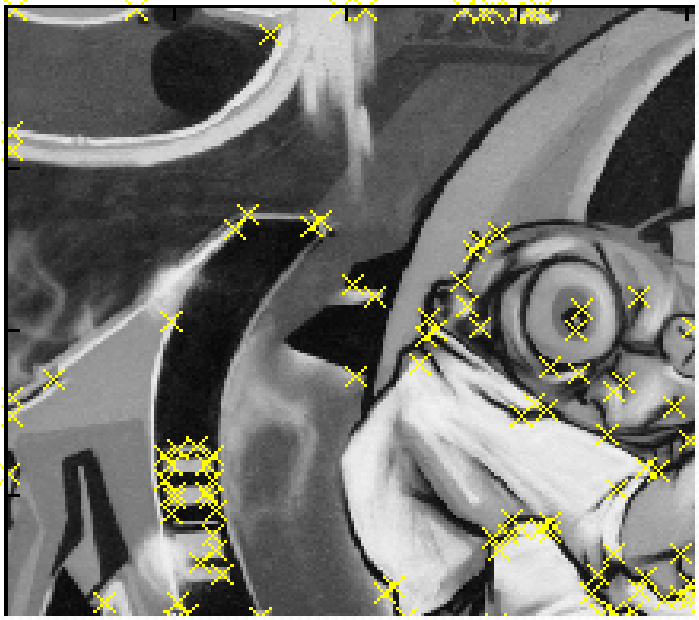


Comparison

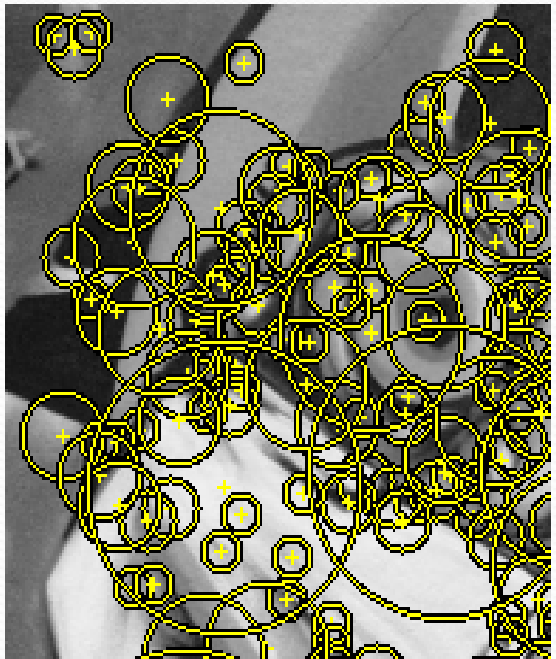
Harris



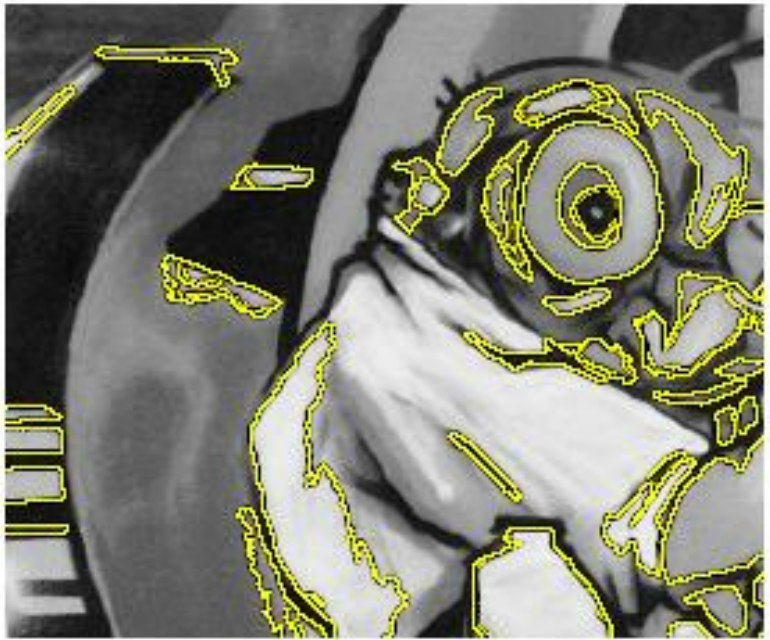
Hessian



LoG



MSER



Available at…

- For most local feature detectors, executables are available online:
  - http://www.robots.ox.ac.uk/~vgg/research/affine
  - http://www.cs.ubc.ca/~lowe/keypoints/
  - http://www.vision.ee.ethz.ch/~surf

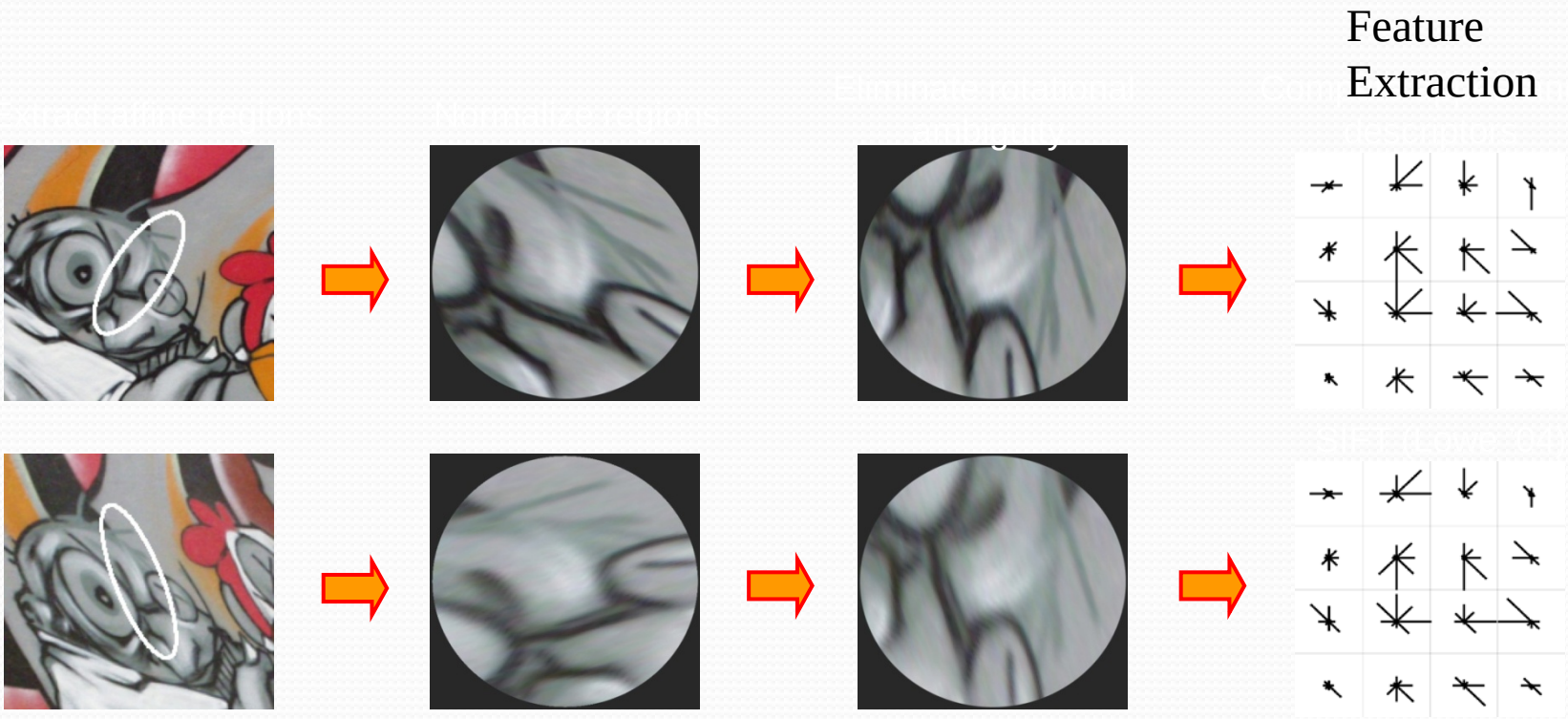
Detectors: Conclusion

- Nowadays many descriptors (PoI/RoI) which are robust to important scale and viewpoint changes.
- The best choice will depend on the type of deformation expected.
- MSER well adapted to structured scenes
- Harris and Hessian well adapted to textured scenes
- SIFT (DoG), and extension as SURF very powerful

- The processing chain:
  1. Extremum detection in a scale-space volume with possibly ***1bis***: Refined localization of PoI
  2. Extraction of area of interest (affine inv, rotation inv, …)
  3. Computation of the descriptor

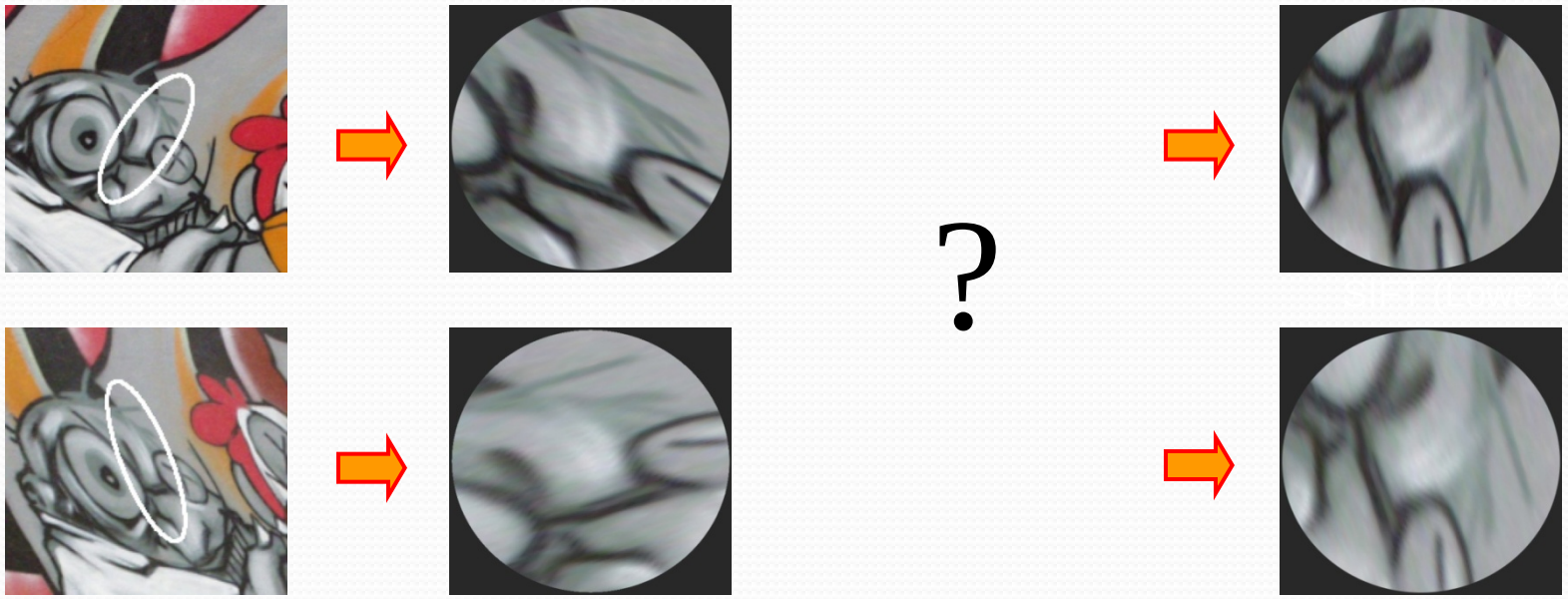
SIFT in details…

Region Detection Steps: Review



Scale Invariant Feature Transform (SIFT)

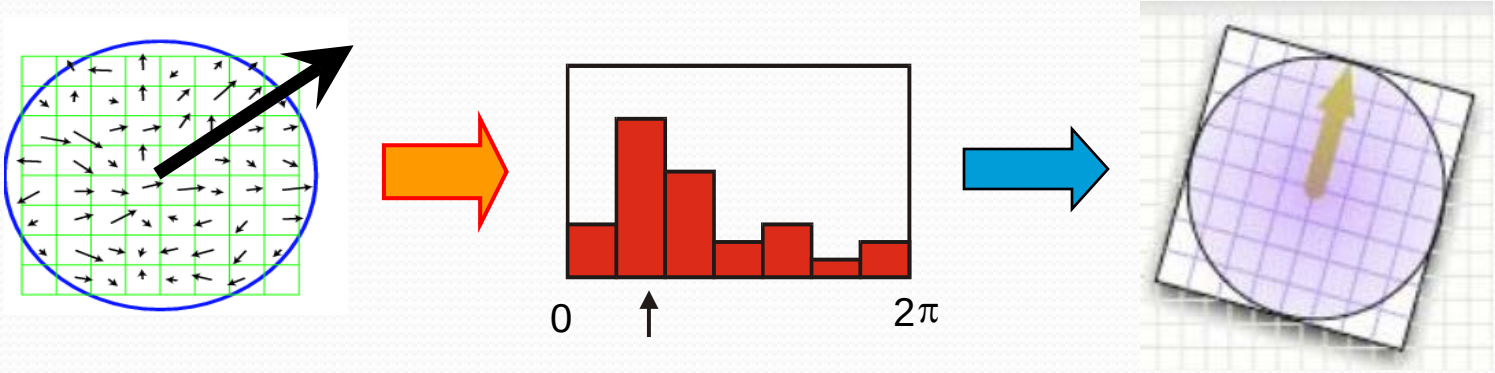
- Remember how we resolved the orientation ambiguity?



Scale Invariant Feature Transform (SIFT)

- Find dominant gradient direction using the histograms of gradient direction.

Dominant direction of gradient



(36 bins)

Scale Invariant Feature Transform (SIFT)

- Same theory, except that we use 16 histograms (8 bins each).

Main idea:

1. Take a 16 x16 window around detected interest point
2. Divide into a 4x4 grid of cells
3. Compute histogram in each cell

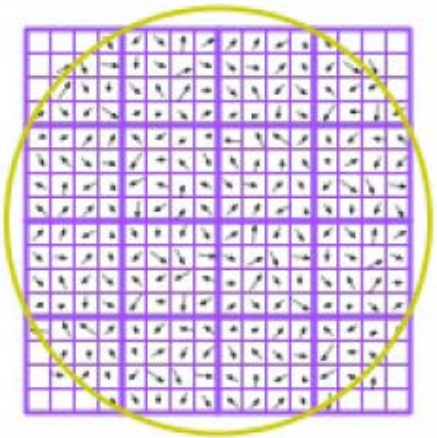
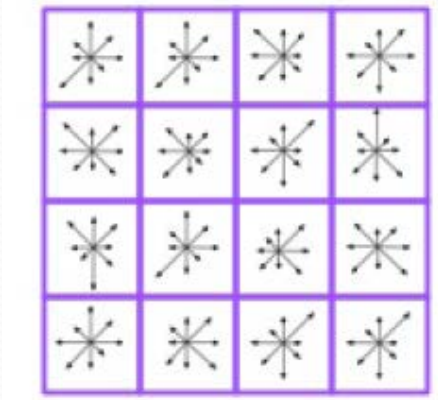


Image gradients

16 histograms x 8 orientations
= 128 features



Keypoint descriptor

Properties of SIFT

- Highly distinctive!
  - A single feature can be correctly matched with high probability against a large database of features from many images.
- Scale and rotation invariant.
- Partially invariant to 3D camera viewpoint
  - Can tolerate up to about 60 degree out of plane rotation
- Can be computed fast and efficiently

Properties of SIFT (cont'd)

- Partially invariant to changes in illumination

http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT

SIFT – Main Steps

(1) Scale-space extrema detection

- Extract scale and rotation invariant interest points (i.e., keypoints).

(2) Keypoint localization

- Determine location and scale for each interest point.

(3) Orientation assignment

- Assign one or more orientations to each keypoint.

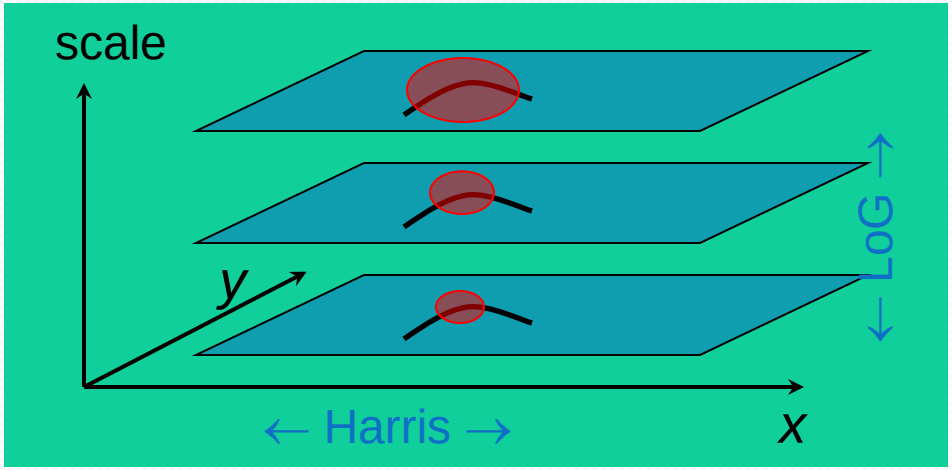
(4) Keypoint descriptor

- Use local image gradients at the selected scale.

D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", **International Journal of Computer Vision**, 60(2):91-110, 2004. Cited 9589 times (as of 3/7/2011)

1. Scale-space Extrema Detection

- Harris-Laplacian
- *Find local maxima of:*
  - Harris detector in space
  - LoG in scale

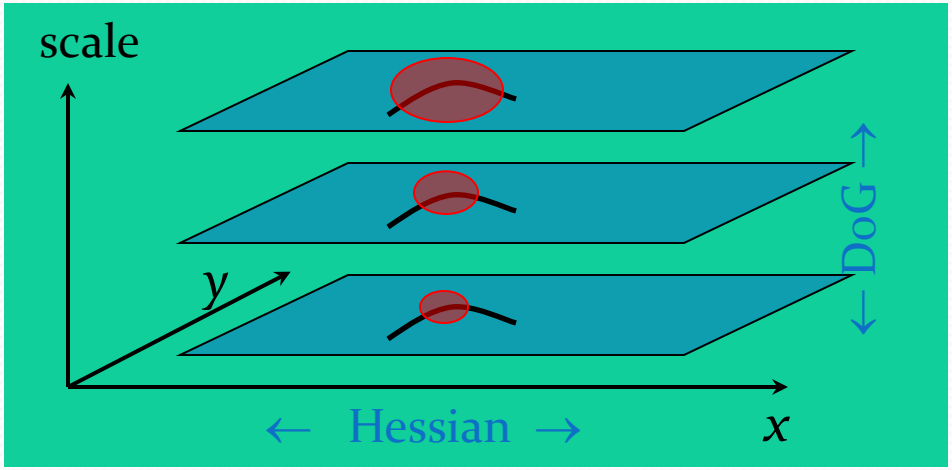


- SIFT

Find local maxima of:
- DoG in space
- DoG in scale

$\sigma_n = k^n \sigma_0$

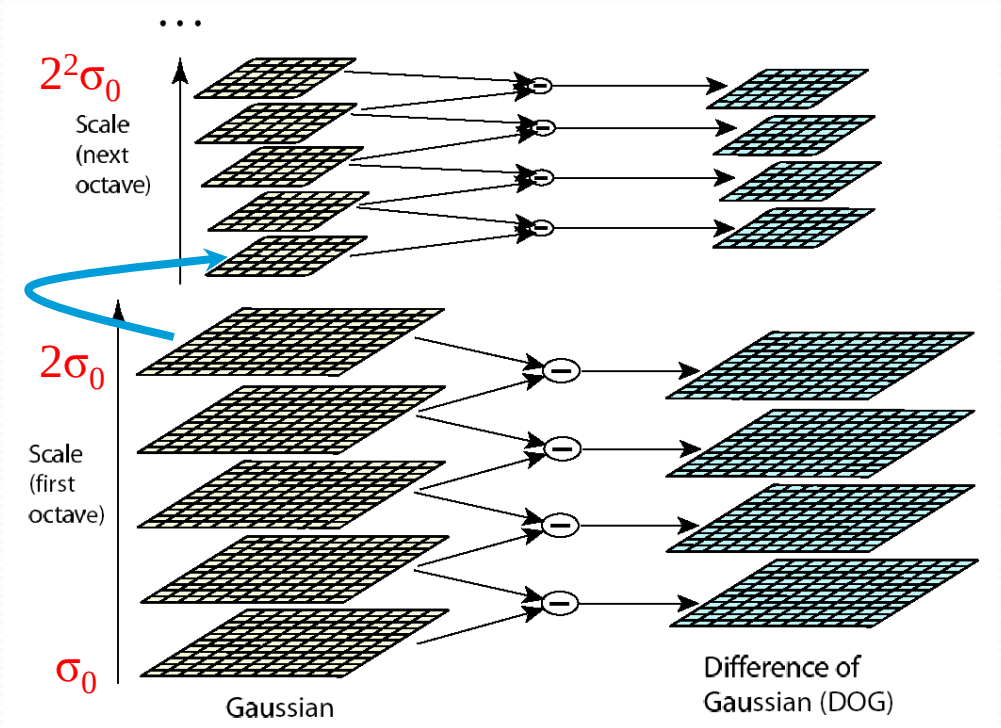
$(k=2)$



1. Scale-space Extrema Detection (cont'd)

- DoG images are grouped by octaves
  - -An octave corresponds to doubling the value of σ
- Fixed number of scales (i.e., levels) per octave

Down-sample

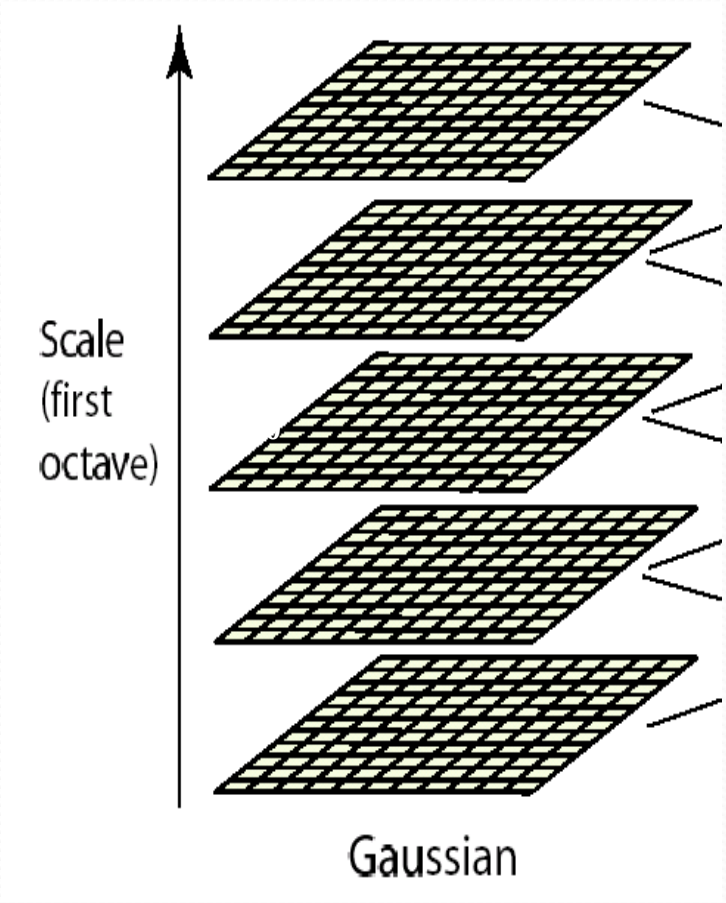


$$G(x,y,k\sigma) - G(x,y,\sigma)$$
$$\approx (k-1)\sigma^2\nabla^2 G$$

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

$$D(x,y,\sigma) =$$
$$L(x,y,k\sigma) - L(x,y,\sigma)$$

1. Scale-space Extrema Detection (cont'd)



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

- Images separated by a constant factor **k**
- If each octave is divided in **s** intervals, we have s+1 DoG images/octave where:

$$\mathbf{k^s = 2 \quad or \quad k = 2^{1/s}}$$

Note: need (s+1)+2 = s+3 blurred images

Choosing SIFT parameters

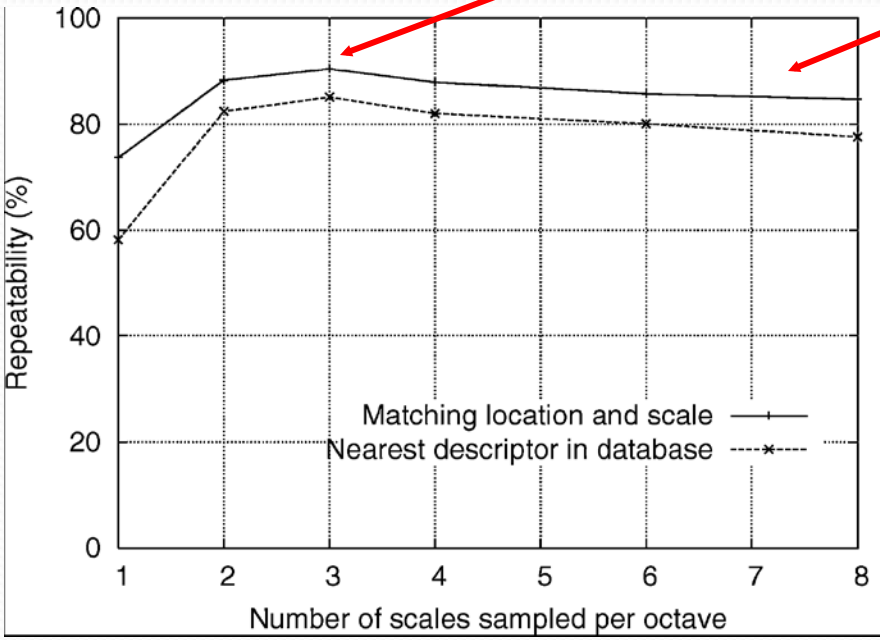
- Experimentally using a matching task:
  - 32 real images (outdoor, faces, aerial etc.)
  - Images subjected to a wide range of transformations (i.e., rotation, scaling, shear, change in brightness, noise).
  - Keypoints are detected in each image.
  - Parameters are chosen based on keypoint repeatability, localization, and matching accuracy.

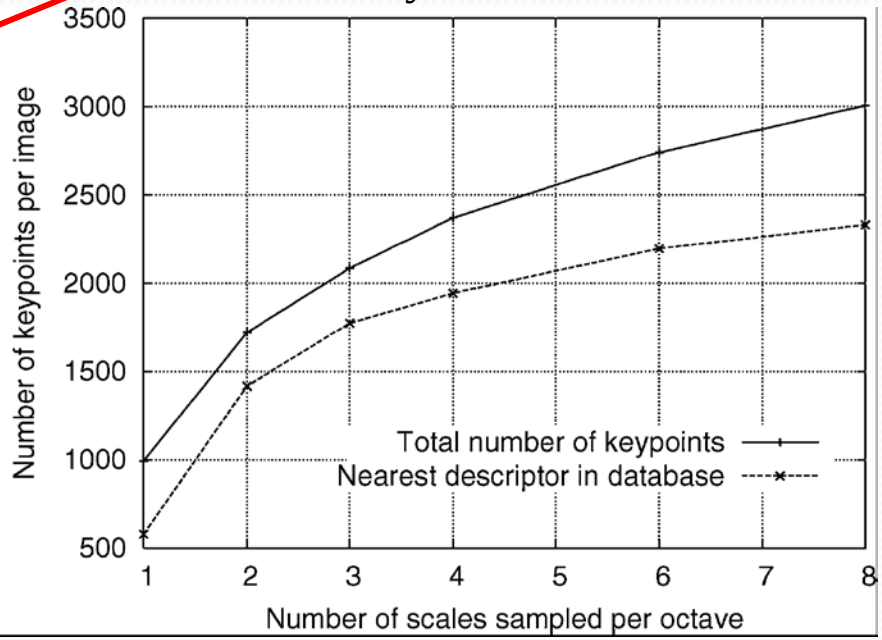
1. Scale-space Extrema Detection (cont'd)

- How many scales sampled per octave?

3 scales

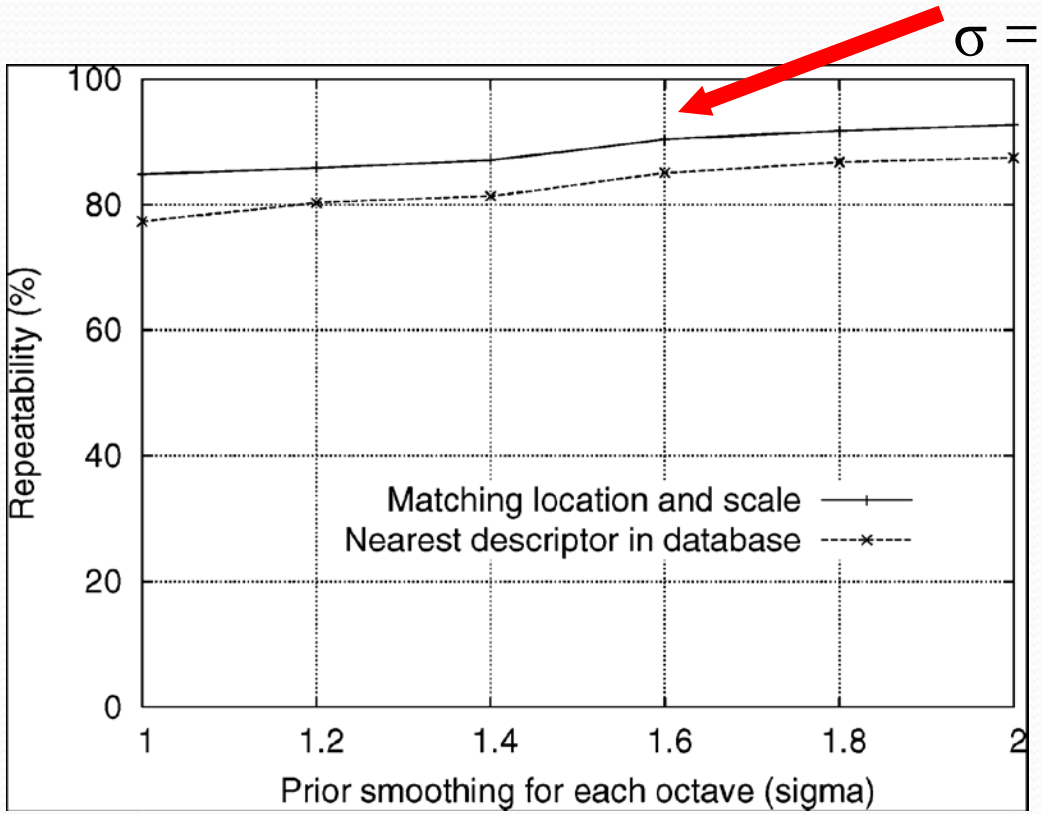
# of keypoints increases but they are not stable!



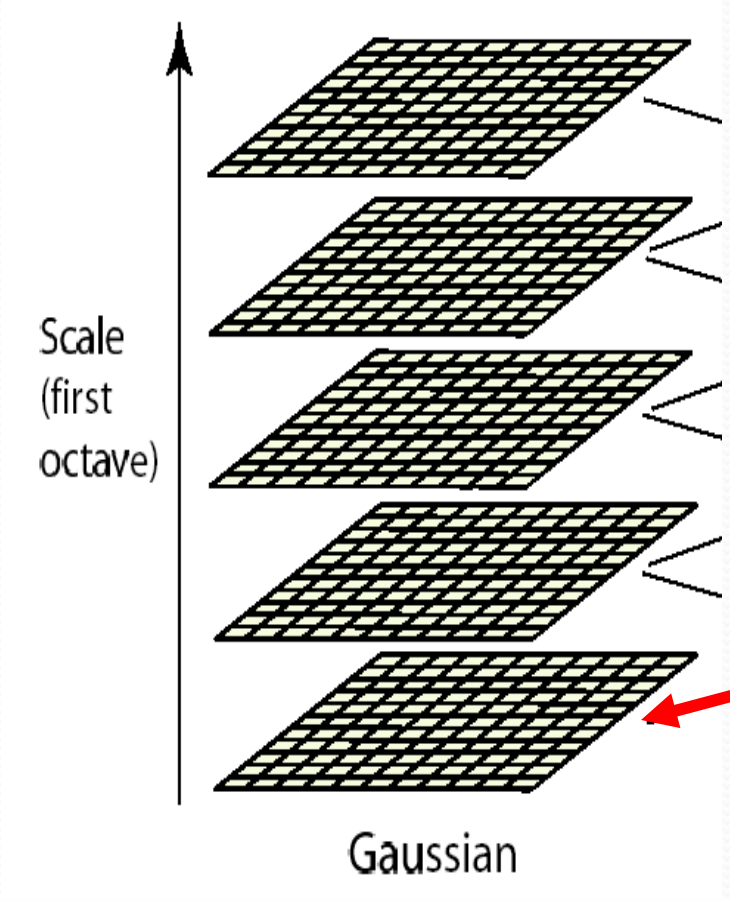


1. Scale-space Extrema Detection (cont'd)

- Smoothing is applied to the first level of each octave.
- How to choose σ? (i.e., integration scale)



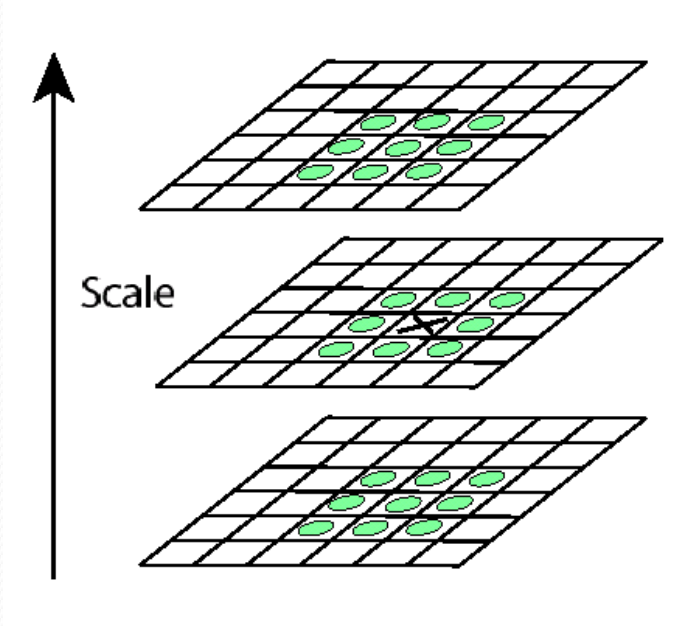
1. Scale-space Extrema Detection (cont'd)



- Pre-smoothing discards high frequencies.
- Double the size of the input image (i.e., using linear interpolation) prior to building the first level of the DoG pyramid.
- Increases the number of stable keypoints by a factor of 4.

1. Scale-space Extrema Detection (cont'd)

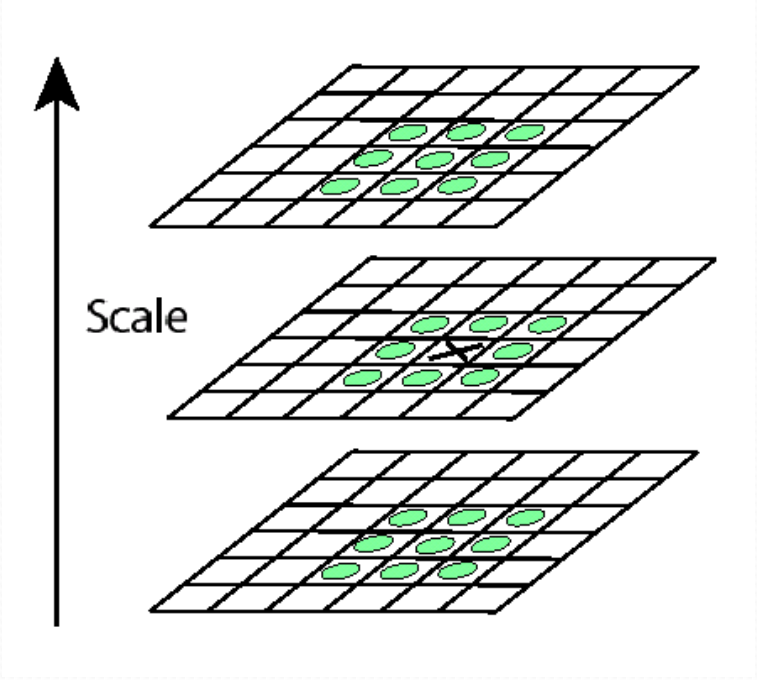
- Extract local extrema (i.e., minima or maxima) in DoG pyramid.
  - Compare each point to its 8 neighbors at the same level, 9 neighbors in the level above, and 9 neighbors in the level below (i.e., 26 total).



2. Keypoint Localization

- Determine the location and scale of keypoints to **sub-pixel** and **sub-scale** accuracy by fitting a 3D quadratic function at each keypoint.

$$X_i = (x_i, y_i, \sigma_i) \rightarrow X_i + \Delta X$$



- Substantial improvement to matching and stability!

2. Keypoint Localization

- Use Taylor expansion of D(x,y,σ) (i.e., DoG function) around the sample point $X_i = (x_i, y_i, \sigma_i)$:

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

where

$\Delta X = (x - x_i, y - y_i, \sigma - \sigma_i)$ is the offset from this point.

2. Keypoint Localization

- To find the extrema of D(ΔX): $\frac{\partial D(X_i)}{\partial X} + \frac{\partial^2 D(X_i)}{\partial X^2}\Delta X = 0$

$$\frac{\partial^2 D(X_i)}{\partial X^2}\Delta X = -\frac{\partial D(X_i)}{\partial X} \quad \rightarrow \quad \Delta X = -\frac{\partial^2 D^{-1}(X_i)}{\partial X^2}\frac{\partial D(X_i)}{\partial X}$$

- ΔX can be computed by solving a 3x3 linear system:

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

use finite differences:

$$\frac{\partial D}{\partial \sigma} = \frac{D_{k+1}^{i,j} - D_{k-1}^{i,j}}{2}$$

$$\frac{\partial^2 D}{\partial \sigma^2} = \frac{D_{k-1}^{i,j} - 2D_k^{i,j} + D_{k+1}^{i,j}}{1}$$

$$\frac{\partial^2 D}{\partial \sigma y} = \frac{(D_{k+1}^{i+1,j} - D_{k-1}^{i+1,j}) - (D_{k+1}^{i-1,j} - D_{k-1}^{i-1,j})}{4}$$

2. Keypoint Localization (cont'd)

- Sub-pixel, sub-scale interpolated estimate:

$$X_i = X_i + \Delta X$$

If $\Delta X > 0.5$ in any dimension, repeat.

- Reject keypoints having low contrast.
  - i.e., sensitive to noise

If $|D(X_i + \Delta X)| < 0.03$ reject keypoint

– i.e., assumes that image values have been normalized in [0,1]

2. Keypoint Localization (cont'd)

- Reject points lying on edges (or being close to edges)

- Harris uses the 2nd order moment matrix:

$$A_W(x,y) = \sum_{x \in W, y \in W} \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix}$$

$$\mathbf{R(A_W) = det(A_W) - \alpha\ trace^2(A_W)}$$

or $$\mathbf{R(A_W) = \lambda_1 \lambda_2 - \alpha\ (\lambda_1 + \lambda_2)^2}$$

2. Keypoint Localization (cont'd)

- SIFT uses the Hessian matrix for efficiency.
  - i.e., encodes principal curvatures

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

α: largest eigenvalue (λ_{max})
β: smallest eigenvalue (λ_{min})
(proportional to principal curvatures)

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

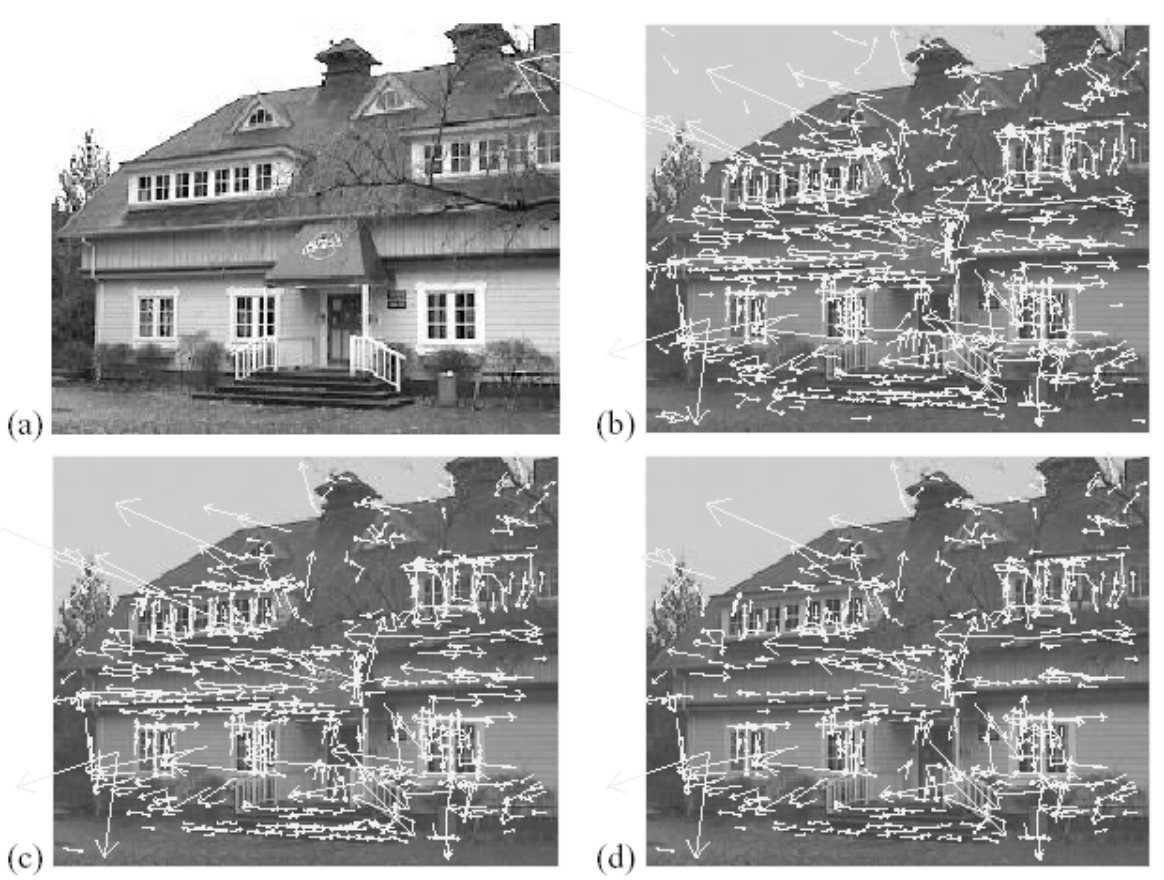
$$\rightarrow \quad \frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r},$$

$(r = \alpha/\beta)$

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

(SIFT uses r = 10)

2. Keypoint Localization (cont'd)



(a) 233x189 image

(b) 832 DoG extrema

(c) 729 left after low contrast threshold

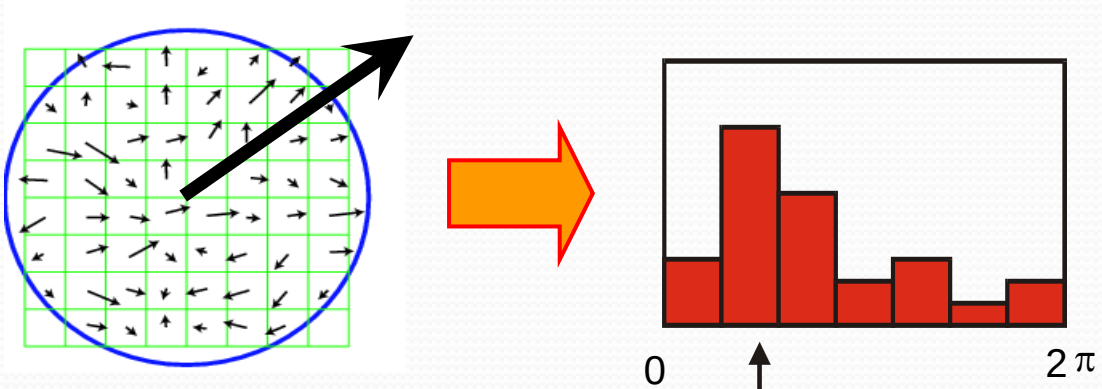
(d) 536 left after testing ratio based on Hessian

3. Orientation Assignment

- Create histogram of gradient directions, within a region around the keypoint, at selected scale (i.e., scale invariance): $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$

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

$$\theta(x, y) = a\tan 2((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$

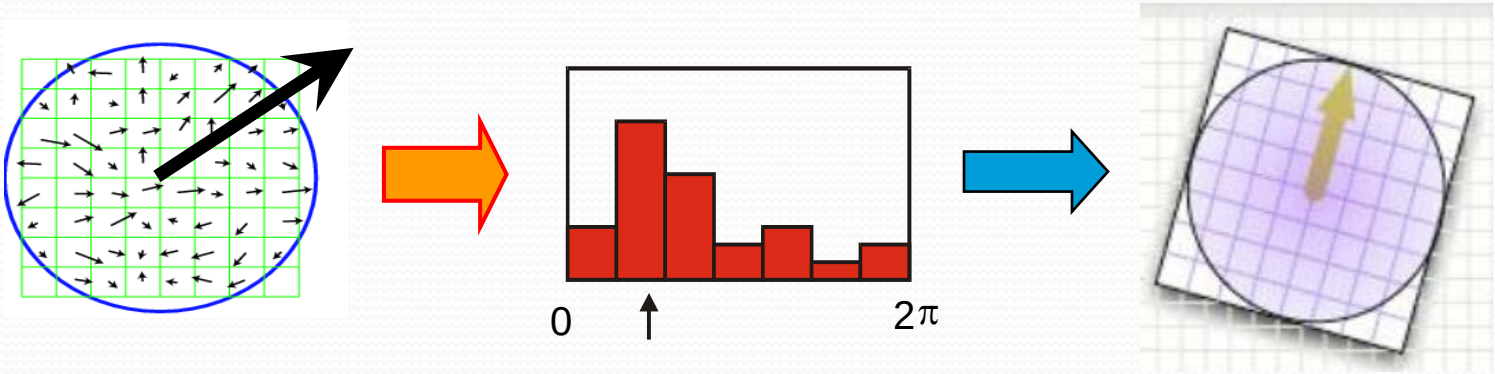


36 bins (i.e., 10° per bin)

- Histogram entries **are weighted** by (i) gradient magnitude and (ii) a Gaussian function with σ equal to 1.5 times the scale of the keypoint.

3. Orientation Assignment (cont'd)

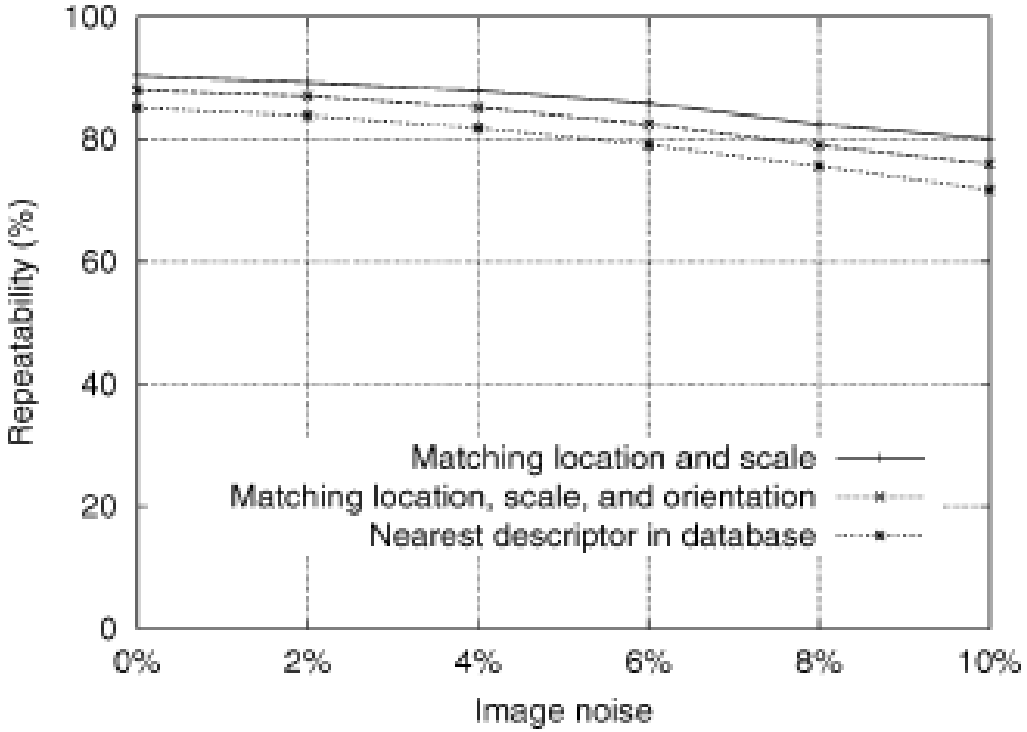
- Assign canonical orientation at peak of smoothed histogram (fit parabola to better localize peak).



- In case of peaks within 80% of highest peak, multiple orientations assigned to keypoints.
  - About 15% of keypoints has multiple orientations assigned.
  - Significantly improves stability of matching.

3. Orientation Assignment (cont'd)

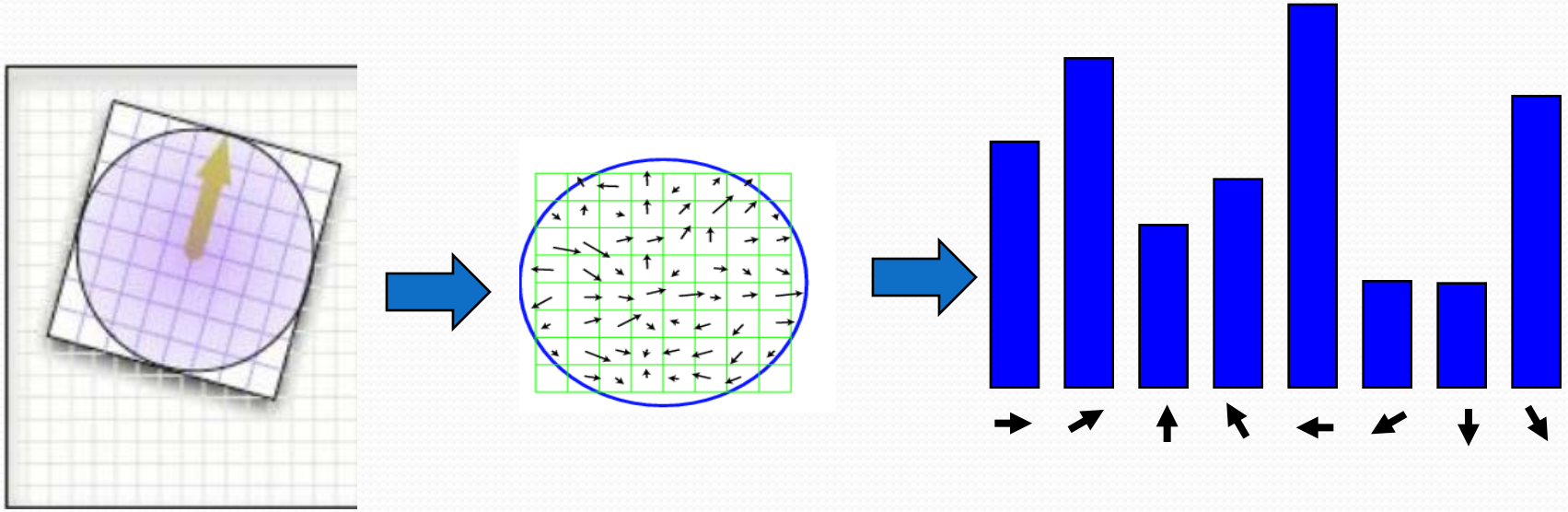
- Stability of location, scale, and orientation (within 15 degrees) under noise.



4. Keypoint Descriptor

- Have achieved invariance to location, scale, and orientation.
- Next, tolerate illumination and viewpoint changes.

Orientation histogram of gradient magnitudes



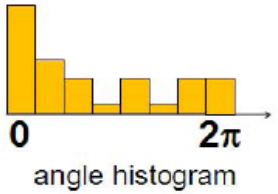
8 bins

4. Keypoint Descriptor (cont'd)

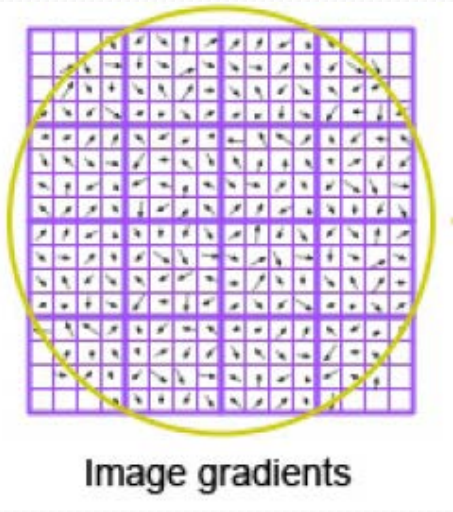
1. Take a 16 x16 window around detected interest point.

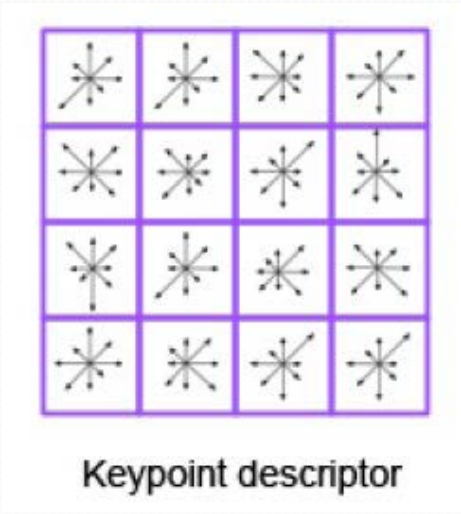
2. Divide into a 4x4 grid of cells.

3. Compute histogram in each cell.



(8 bins)

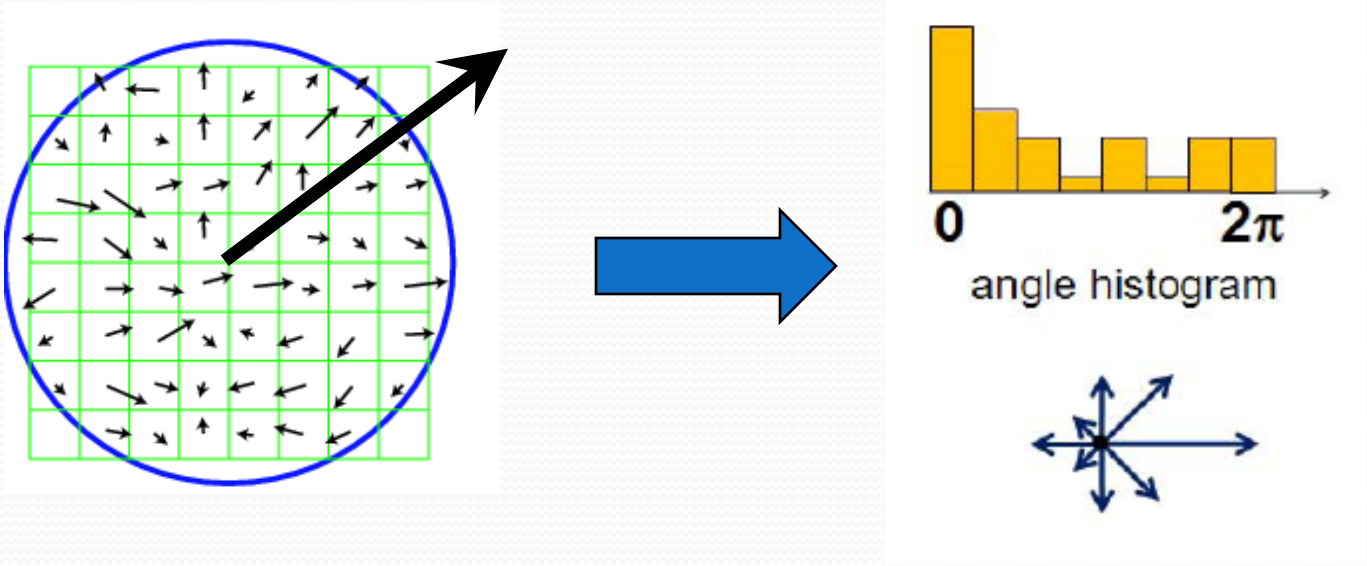




16 histograms x 8 orientations
= 128 features

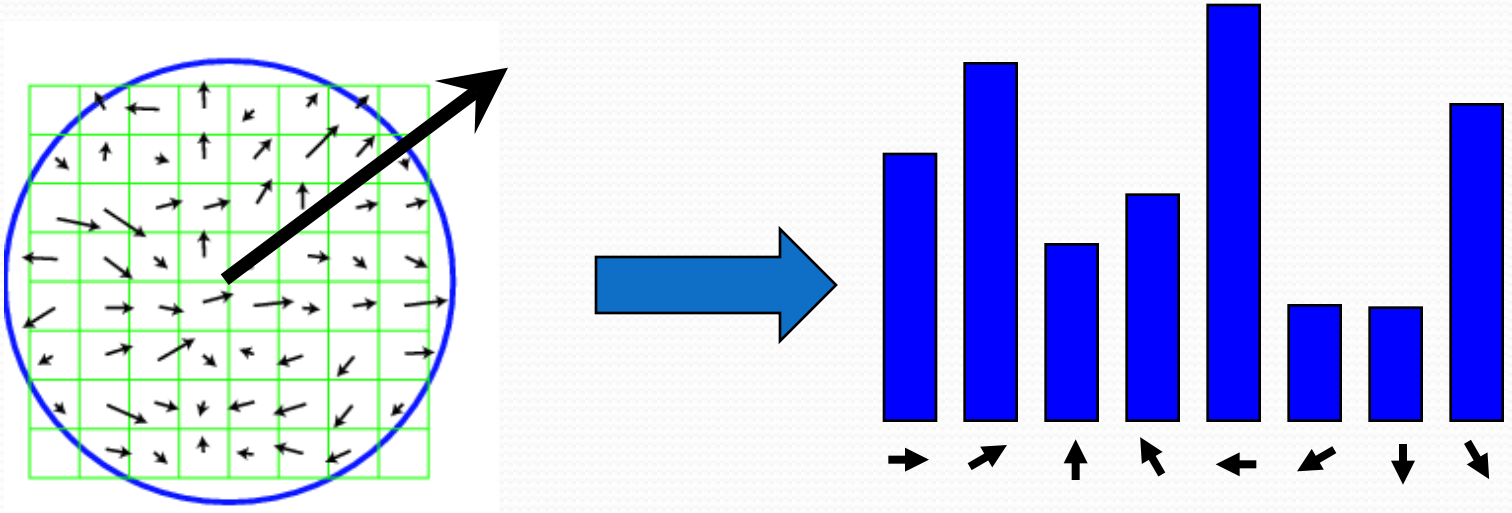
4. Keypoint Descriptor (cont'd)

- Each histogram entry **is weighted** by (i) gradient magnitude and (ii) a Gaussian function with σ equal to 0.5 times the width of the descriptor window.



4. Keypoint Descriptor (cont'd)

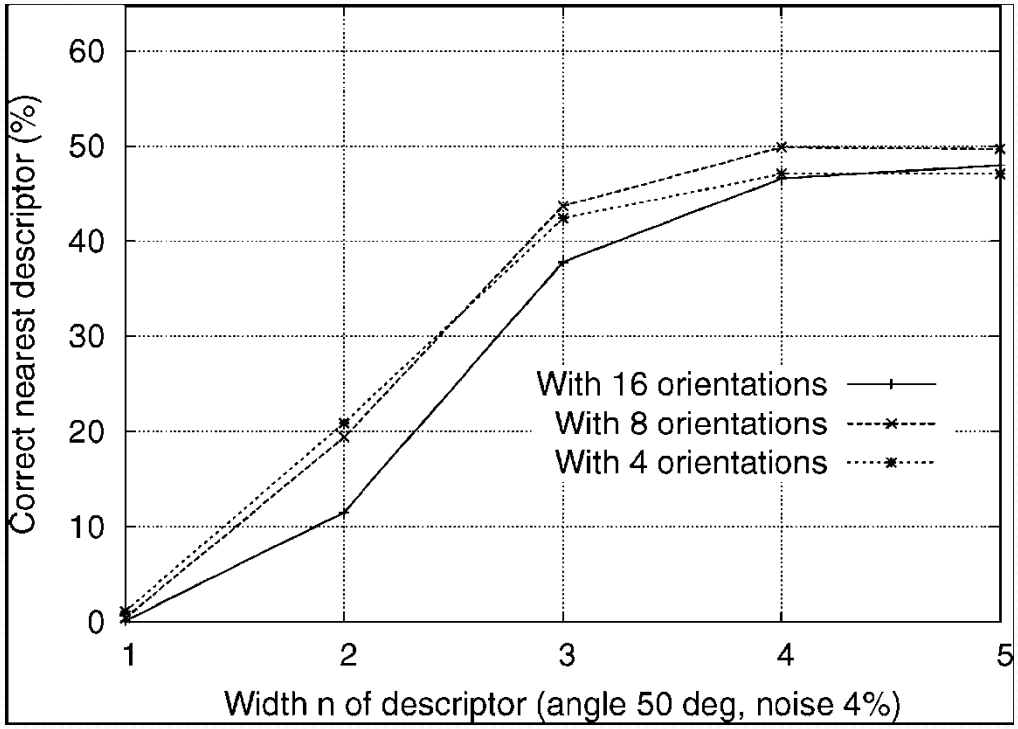
- Partial Voting: distribute histogram entries into adjacent bins (i.e., additional robustness to shifts)
  - Each entry is added to all bins, multiplied by a weight of 1-d, where d is the distance from the bin it belongs.



4. Keypoint Descriptor (cont'd)

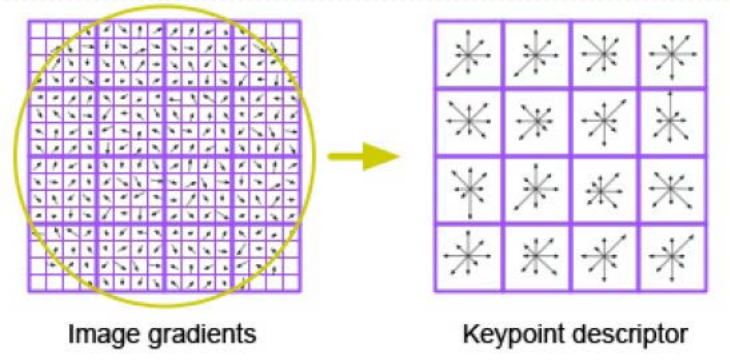
- Descriptor depends on two parameters:
  (1) number of orientations r
  (2) n x n array of orientation histograms

} $r\ n^2$ features



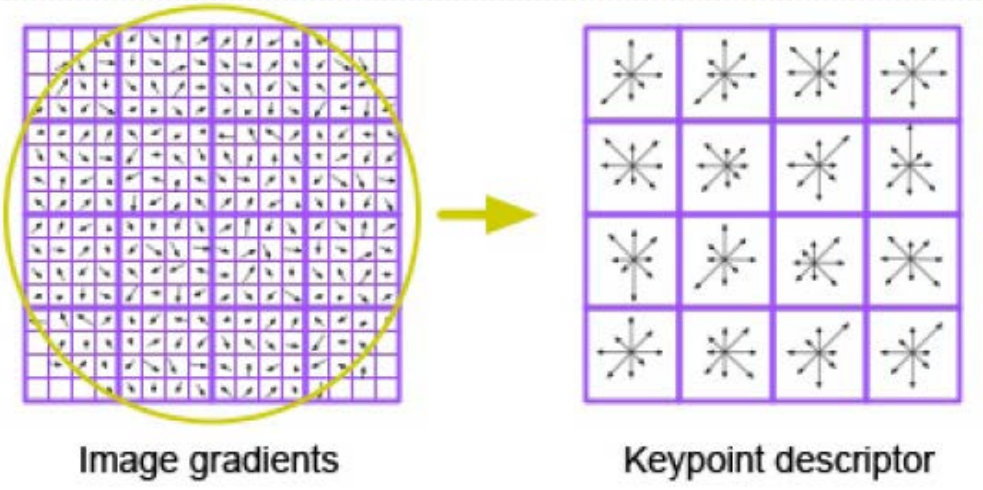
SIFT: r = 8, n = 4

128 features



4. Keypoint Descriptor (cont'd)

- Invariance to affine (linear) illumination changes:
  - Normalization to unit length is sufficient.

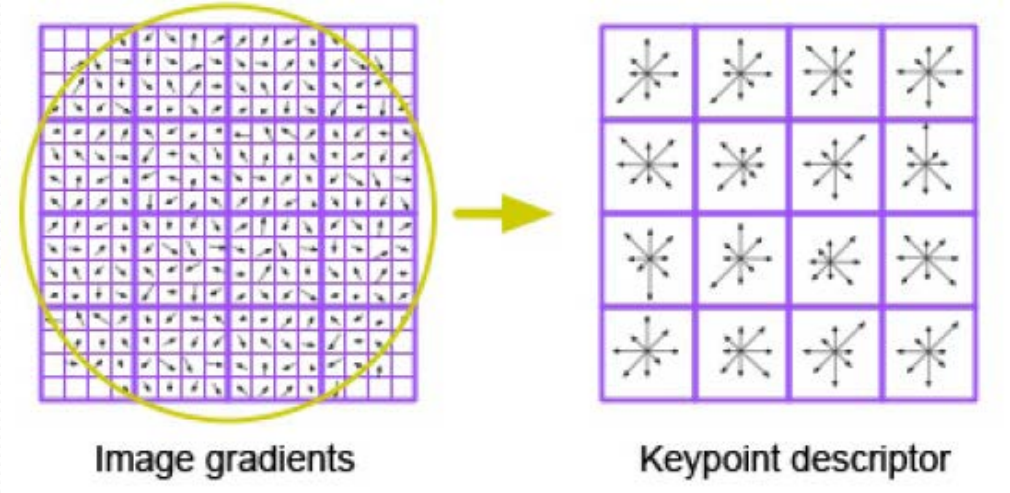


128 features

4. Keypoint Descriptor (cont'd)

- Non-linear illumination changes :
  - Saturation affects gradient magnitudes more than orientations
  - Threshold gradient magnitudes to be no larger than 0.2 and renormalize to unit length

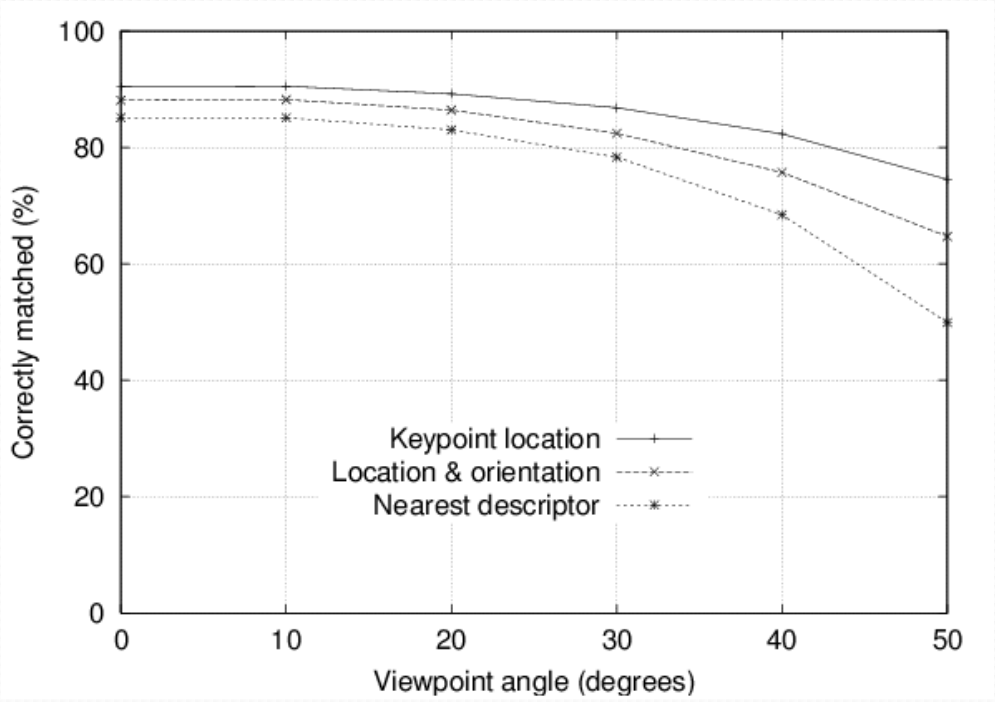
    (i.e., emphasizes gradient orientations than magnitudes)



128 features

Robustness to viewpoint changes

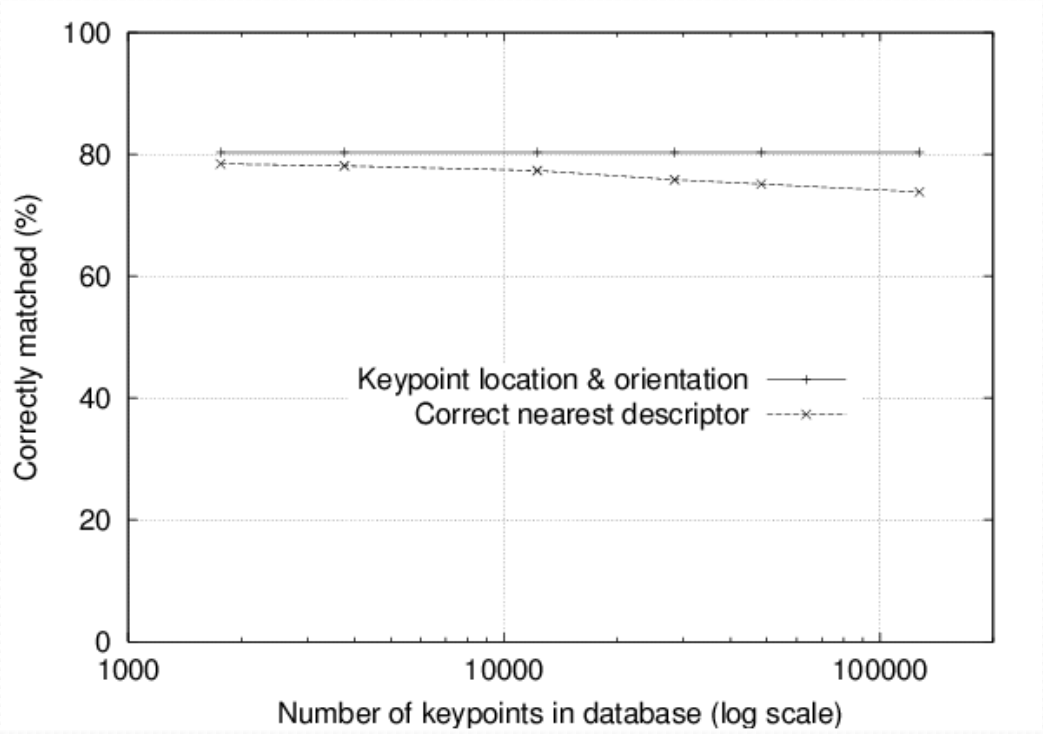
- Match features after random change in image scale and orientation, with 2% image noise, and affine distortion.
- Find nearest neighbor in database of 30,000 features.



Additional robustness can be achieved using affine invariant region detectors.

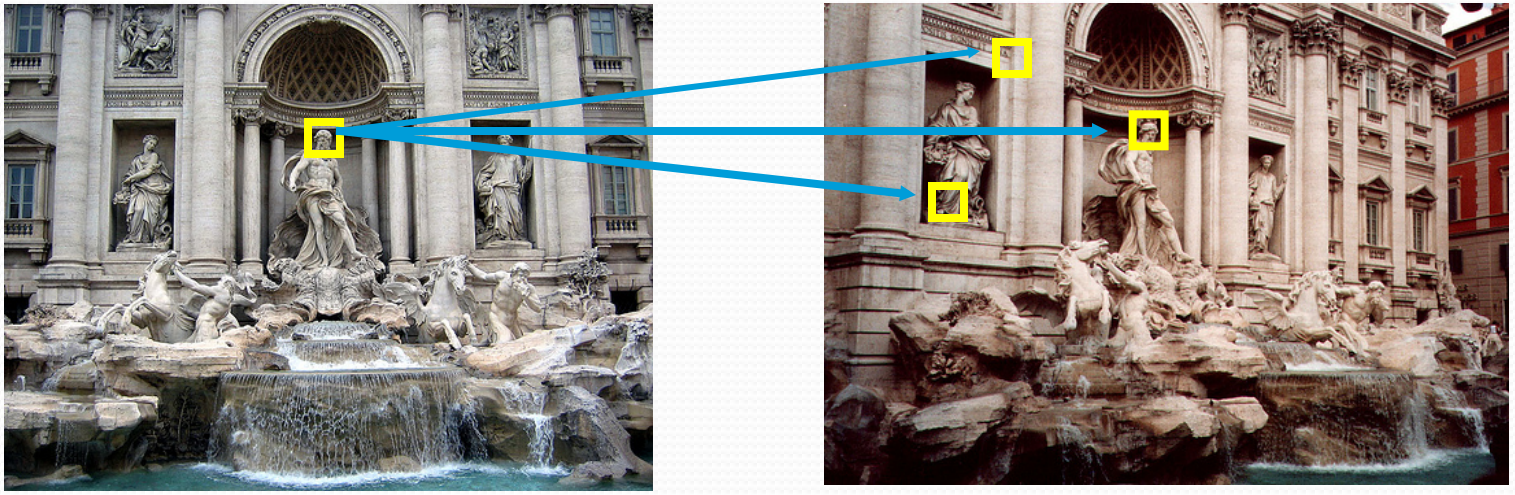
Distinctiveness

- Vary size of database of features, with 30 degree affine change, 2% image noise.
- Measure % correct for single nearest neighbor match.



Matching SIFT features

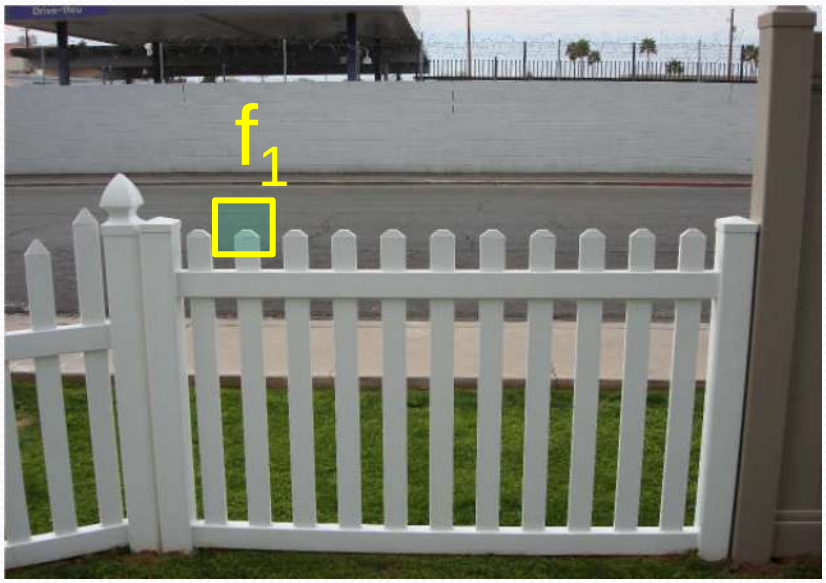
- Given a feature in I_1, how to find the best match in I_2?
  1. Define distance function that compares two descriptors.
  2. Test all the features in I_2, find the one with min distance.



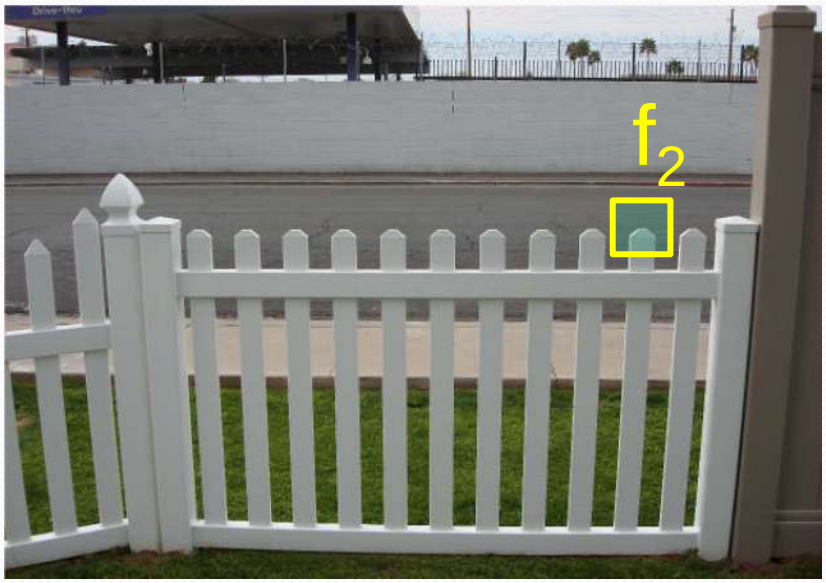
I_1 I_2

Matching SIFT features (cont'd)

- How to define the distance between two features f_1, f_2?
  - Simple approach: SSD(f_1, f_2) (i.e., sum of squared differences)
  - Can give good scores to very ambiguous (bad) matches ?



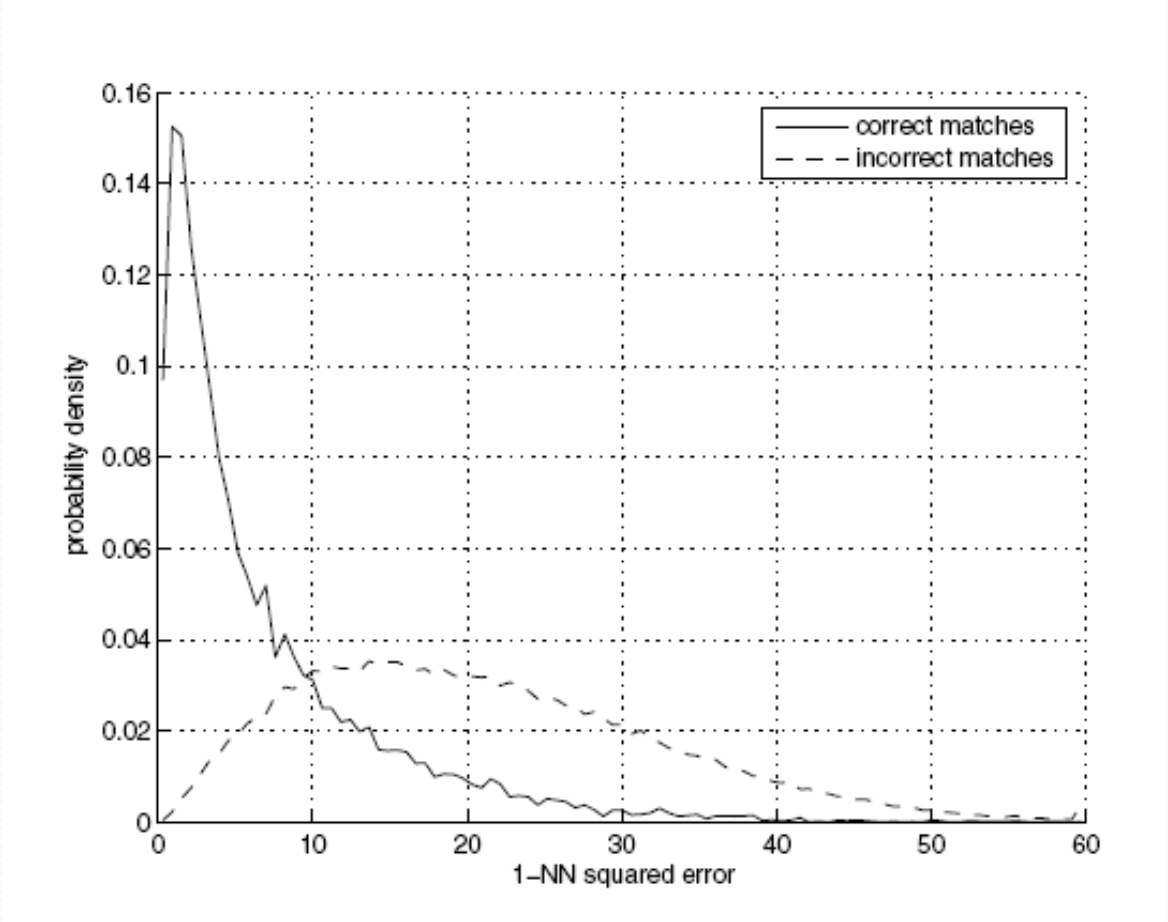
I_1



I_2

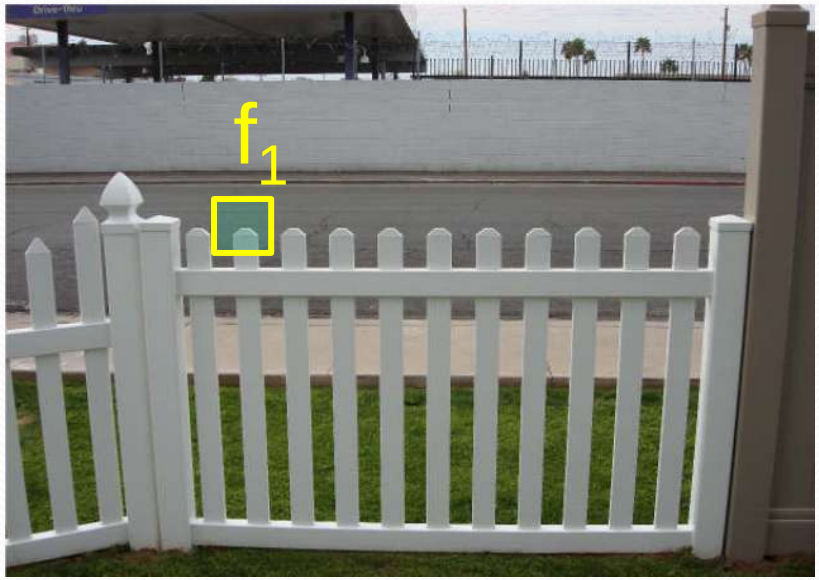
Matching SIFT features (cont'd)

SSD(f1,f2) < t How to set t?

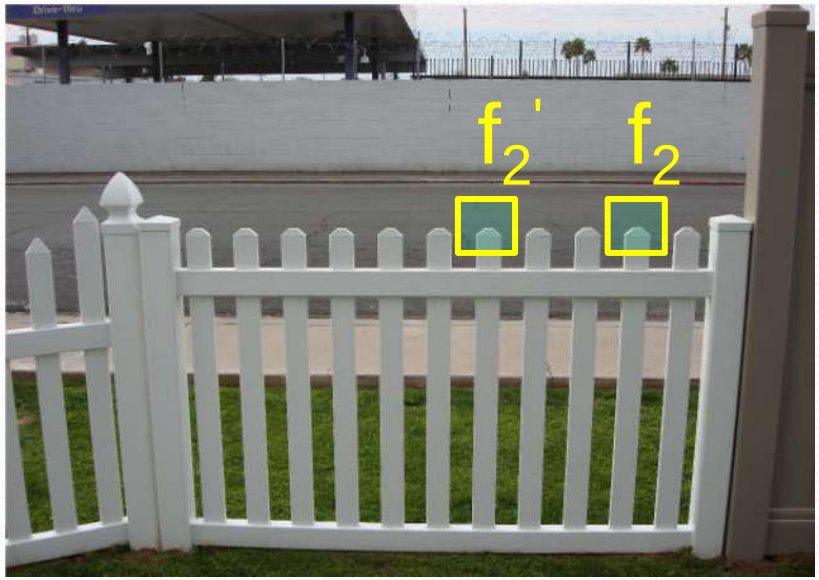


Matching SIFT features (cont'd)

- How to define the difference between two features f_1, f_2?
  - Better approach: $SSD(f_1, f_2) / SSD(f_1, f_2')$
    - f_2 is best SSD match to f_1 in I_2
    - f_2' is 2nd best SSD match to f_1 in I_2



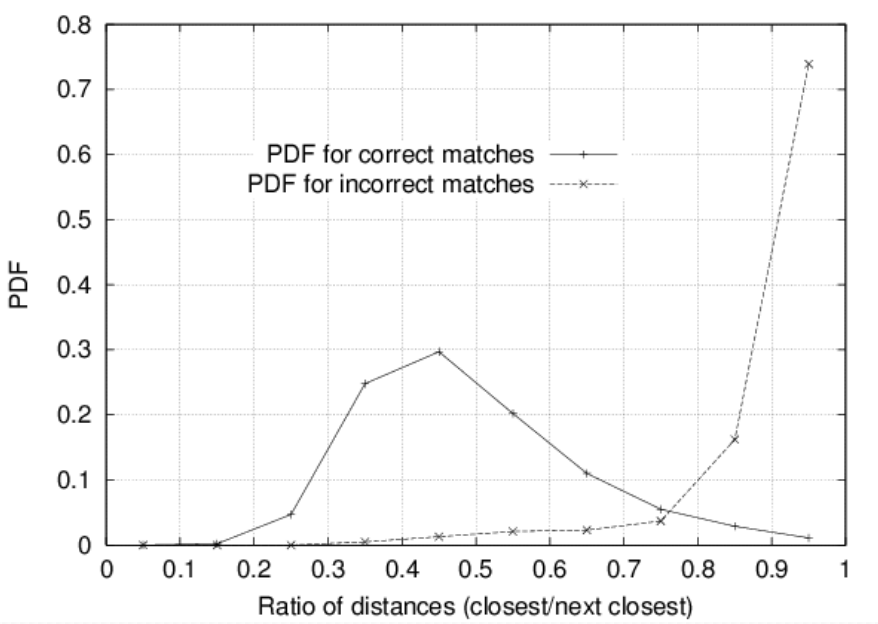
I_1



I_2

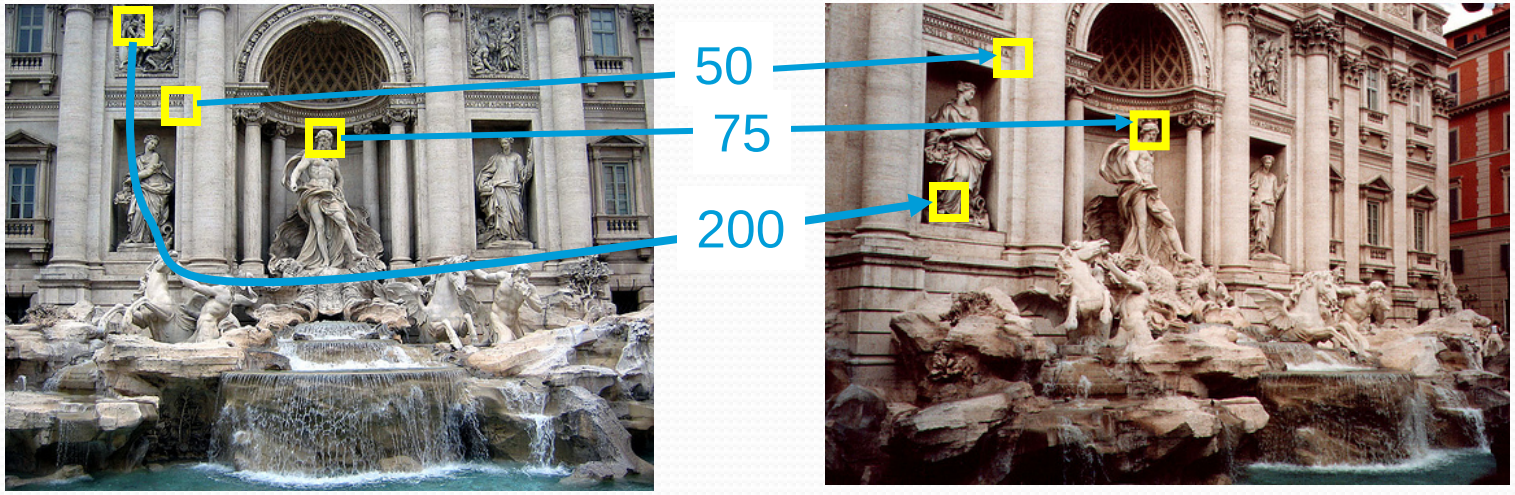
Matching SIFT features (cont'd)

- Accept a match if SSD(f_1, f_2) / SSD(f_1, f_2') < t
- t=0.8 has given good results in object recognition.
  - Eliminated 90% of false matches.
  - Discarded less than 5% of correct matches



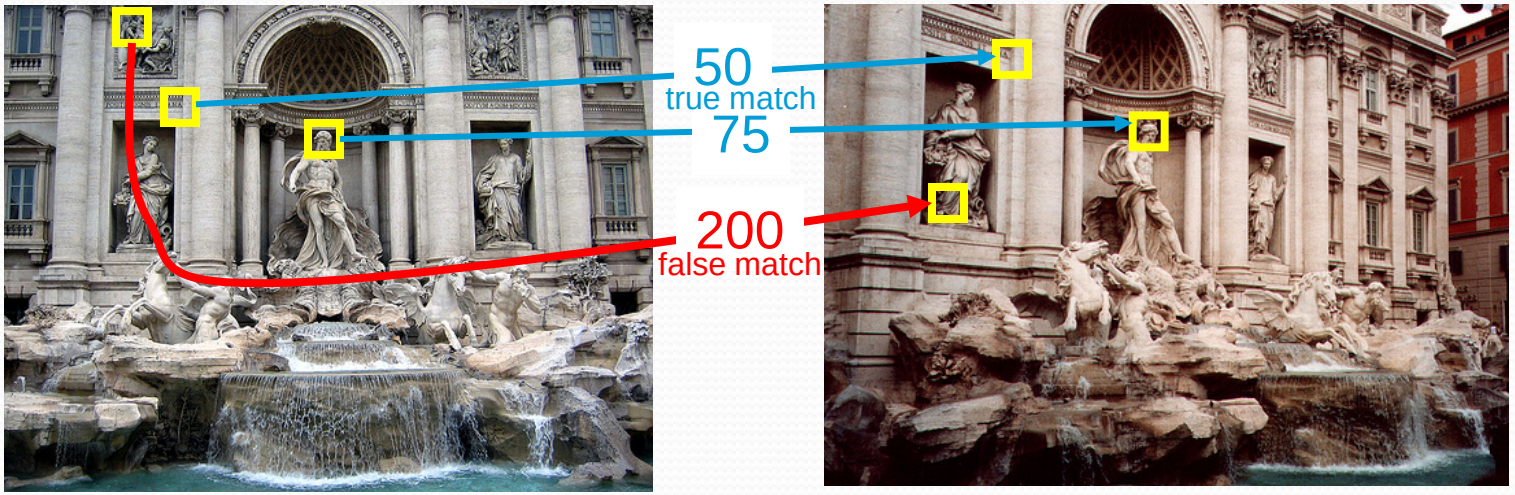
Matching SIFT features (cont'd)

- How to evaluate the performance of a feature matcher?



Matching SIFT features (cont'd)

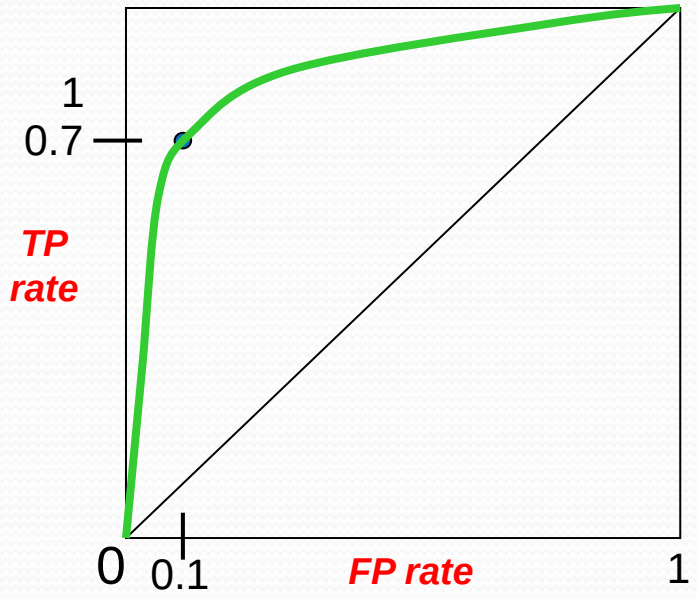
- Threshold t affects # of correct/false matches



- True positives (TP) = # of detected matches that are correct
- False positives (FP) = # of detected matches that are incorrect

Matching SIFT features(cont'd)

- ROC Curve
  - Generated by computing (FP, TP) for different thresholds.
  - Need to maximize area under the curve (AUC)



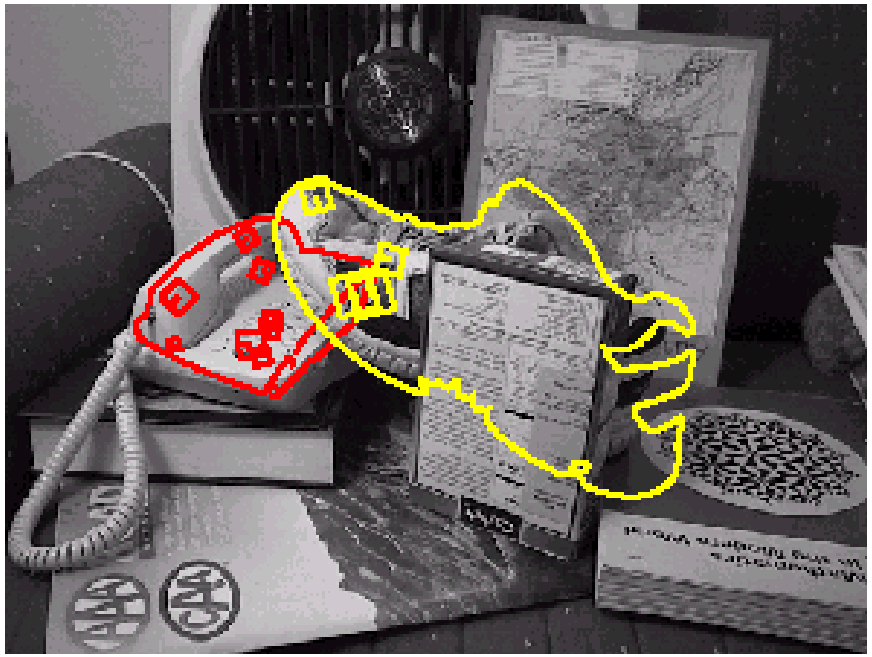
http://en.wikipedia.org/wiki/Receiver_operating_characteristic

Applications of SIFT

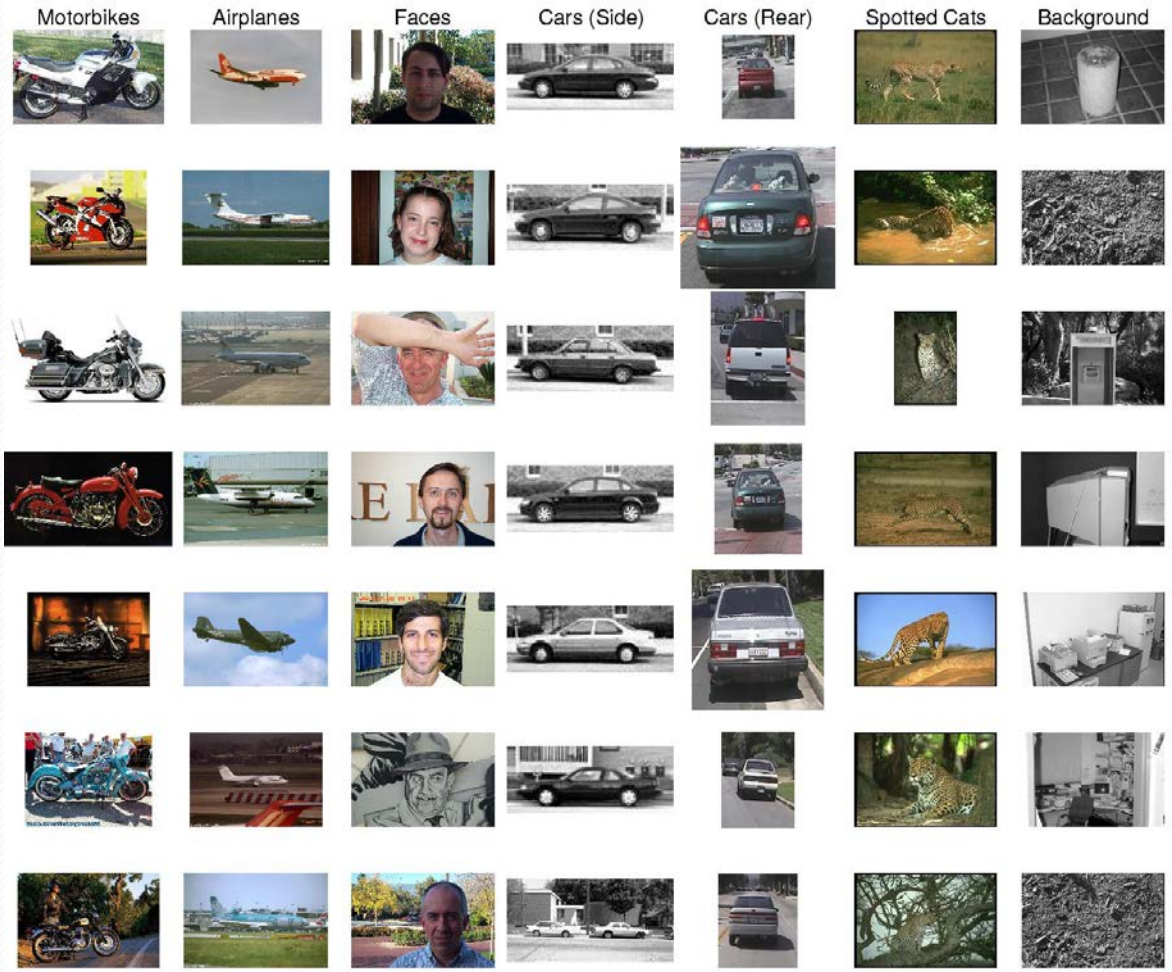
- Object recognition
- Object categorization
- Location recognition
- Robot localization
- Image retrieval
- Image panoramas

Object Recognition

Object Models

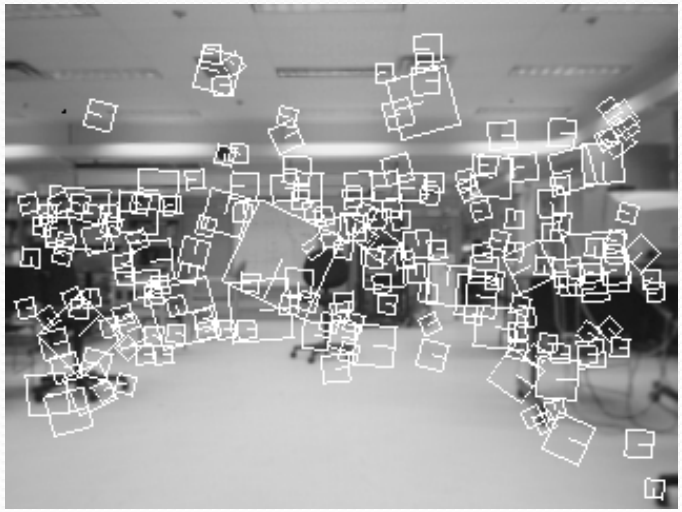


Object Categorization



Location recognition

Robot Localization



Map continuously built over time

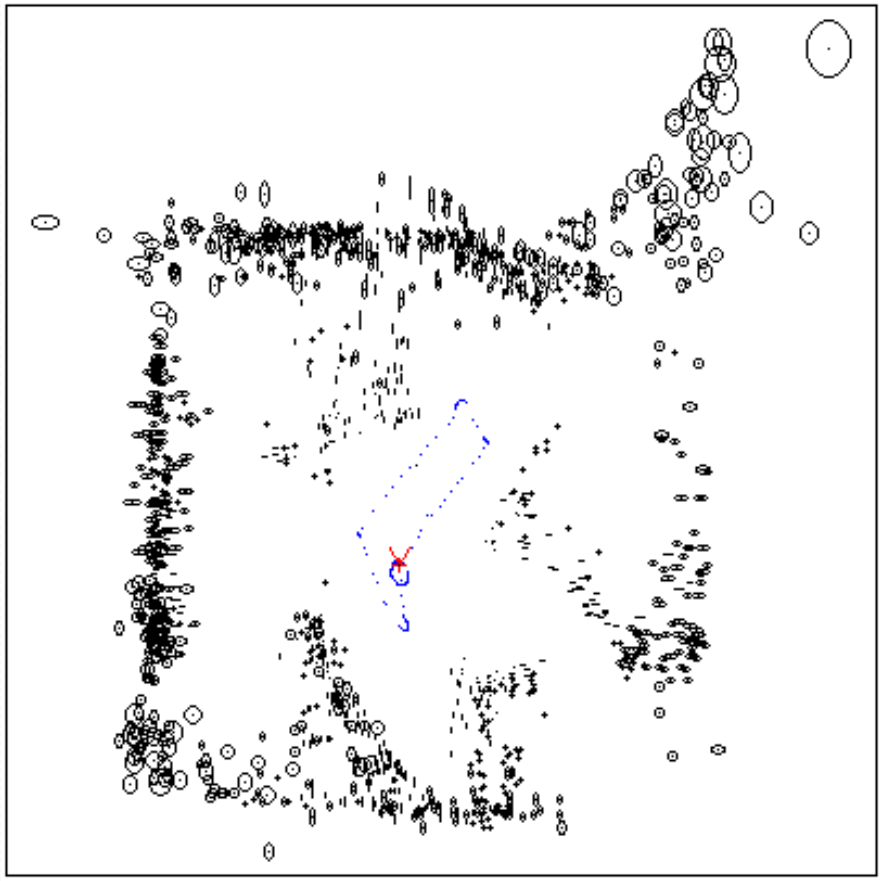


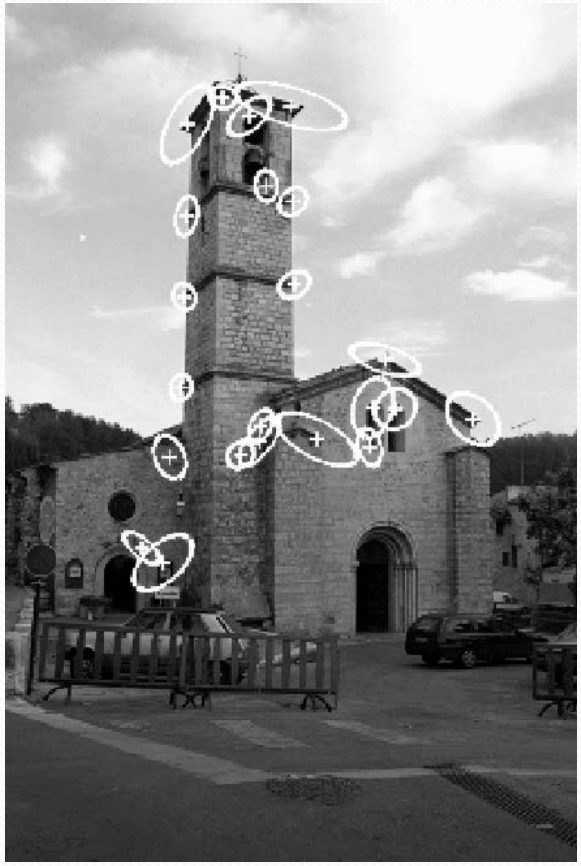
Image retrieval – Example 1

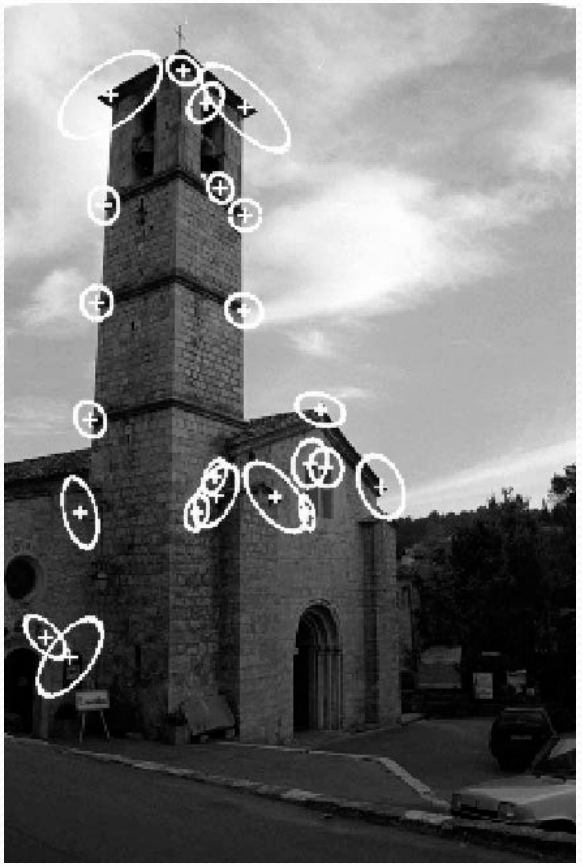
change in viewing angle

• • •

> 5000 images

Matches





22 correct matches

Image retrieval – Example 2

change in viewing angle
+ scale change

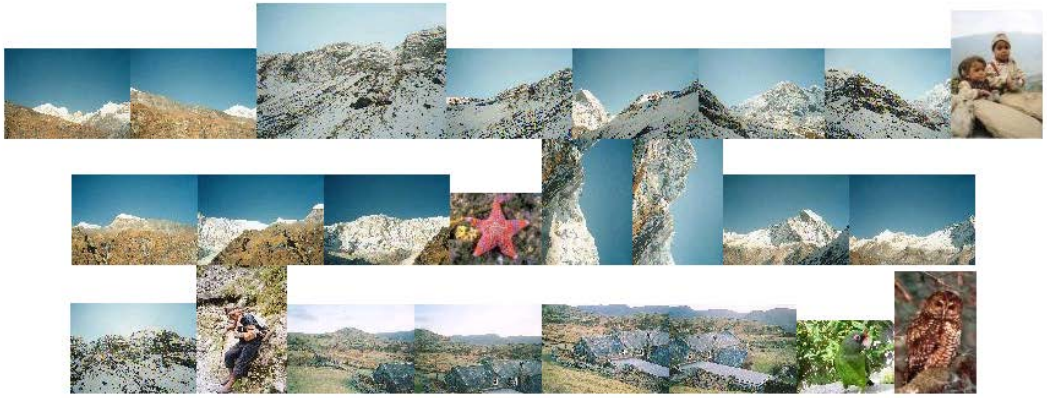
• • •

> 5000 images

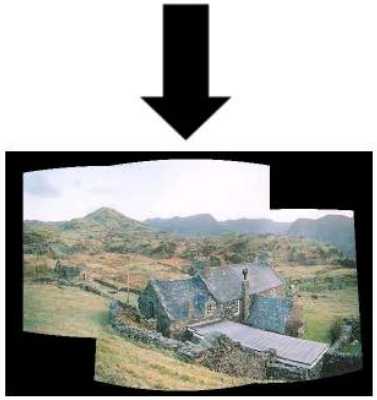
Matches

33 correct matches

Image panoramas from an unordered image set



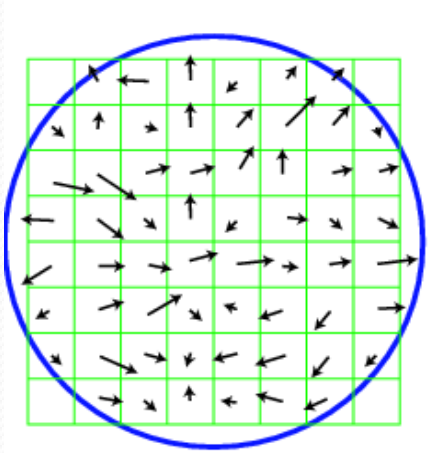
Input images



Output panorama 1

PCA-SIFT

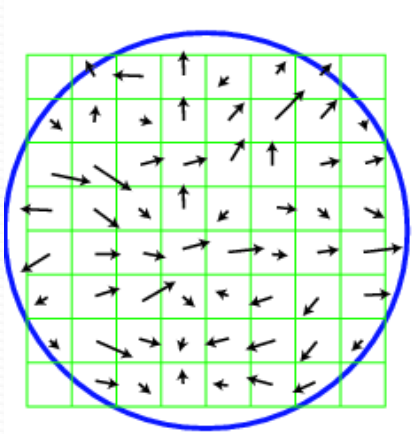
- Steps 1-3 are the same; Step 4 is modified.
- Take a 41 x 41 patch at the given scale, centered at the keypoint, and normalized to a canonical direction.



*Yan Ke and Rahul Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors", **Computer Vision and Pattern Recognition**, 2004*

PCA-SIFT

- Instead of using weighted histograms, concatenate the horizontal and vertical gradients into a long vector.
- Normalize vector to unit length.

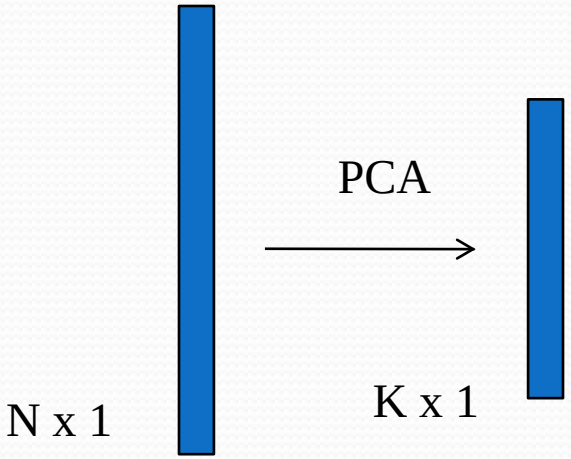


2 x 39 x 39 = 3042 vector

*Yan Ke and Rahul Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors", **Computer Vision and Pattern Recognition**, 2004*

PCA-SIFT

- Reduce the dimensionality of the vector using Principal Component Analysis (PCA)
  - e.g., from 3042 to 36



$$A_{KxN} I_{Nx1} = I'_{Kx1}$$

- Proved to be less discriminatory than SIFT.

SURF: Speeded Up Robust Features

- Fast implementation of a variation of the SIFT descriptor.
- Key idea: fast approximation of (i) Hessian matrix and (ii) descriptor using "integral images".

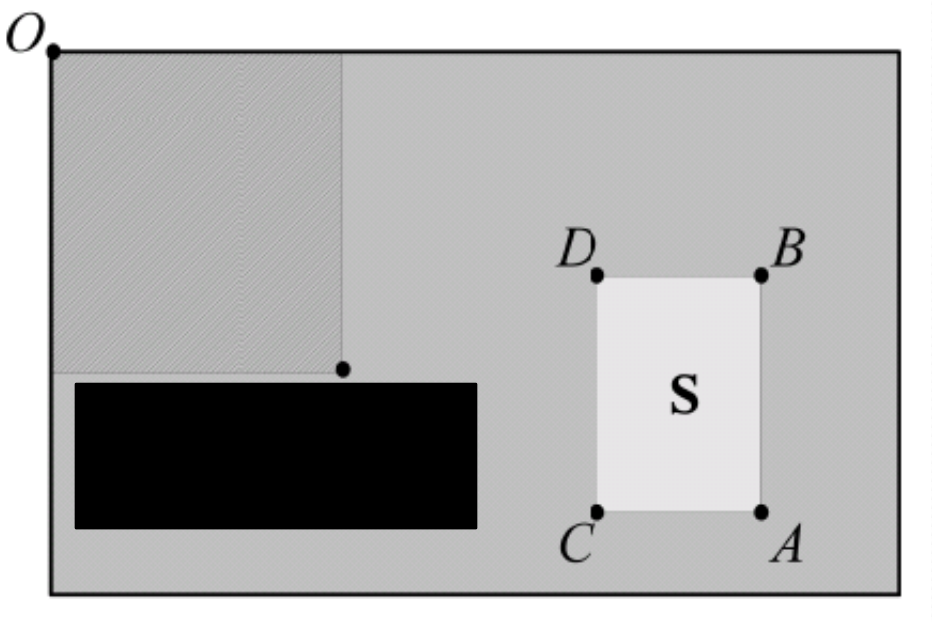
$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix},$$

- What is an "integral image"?

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "SURF: Speeded Up Robust Features", ***European Computer Vision Conference (ECCV),*** *2006.*

Integral Image

- The integral image $I_{\Sigma}(x,y)$ of an image I(x, y) represents the sum of all pixels in I(x,y) of a rectangular region formed by (0,0) and (x,y).



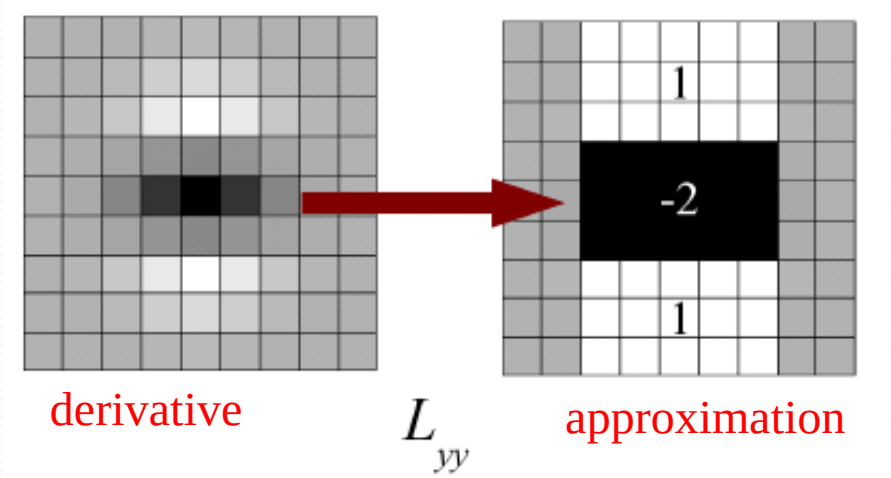
Using integral images, it takes only four array references to calculate the sum of pixels over a rectangular region of any size.

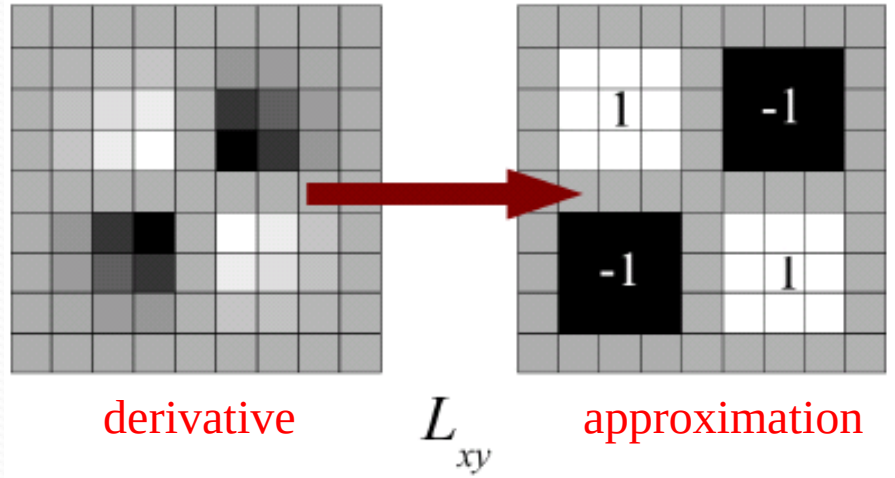
$$\mathbf{S}=A-B-C+D$$

SURF: Speeded Up Robust Features (cont'd)

- Approximate L_{xx}, L_{yy}, and L_{xy} using box filters.

(box filters shown are 9 x 9 – good approximations for a Gaussian with σ=1.2)

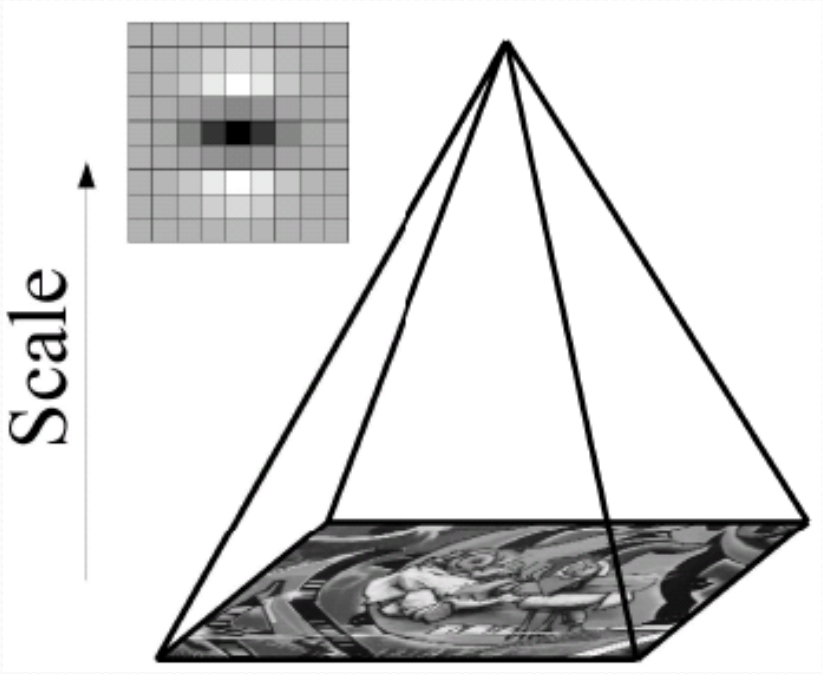




- Can be computed very fast using integral images!

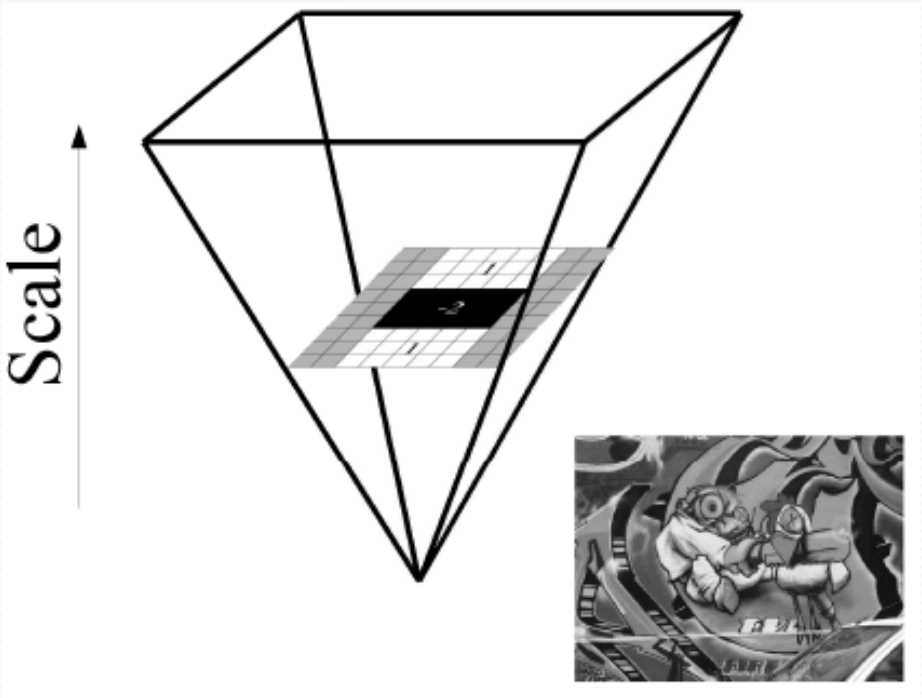
SURF: Speeded Up Robust Features (cont'd)

- In SIFT, images are repeatedly smoothed with a Gaussian and subsequently sub-sampled in order to achieve a higher level of the pyramid.



SURF: Speeded Up Robust Features (cont'd)

- Alternatively, we can use filters of larger size on the original image.
- Due to using integral images, filters of any size can be applied at exactly the same speed!



(see paper for details on how the filter size and octaves are computed)

SURF: Speeded Up Robust Features (cont'd)

- Approximation of H:

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix},$$

Using DoG

$$SIFT: \; H_{approx}^{SIFT} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}$$

Using box filters

$$SURF: \; H_{approx}^{SURF} = \begin{bmatrix} \hat{L}_{xx} & \hat{L}_{xy} \\ \hat{L}_{yx} & \hat{L}_{yy} \end{bmatrix}$$

SURF: Speeded Up Robust Features (cont'd)

- Instead of using a different measure for selecting the location and scale of interest points (like in SIFT(, SURF uses the determinant of H_{approx}^{SURF} to find both.

- Determinant elements must be weighted to obtain a good approximation of det(H):

$$\det(H_{approx}^{SURF}) = \hat{L}_{xx}\hat{L}_{yy} - (0.9\hat{L}_{xy})^2$$

SURF: Speeded Up Robust Features (cont'd)

- Once interest points have been localized both in space and scale, the next steps are:

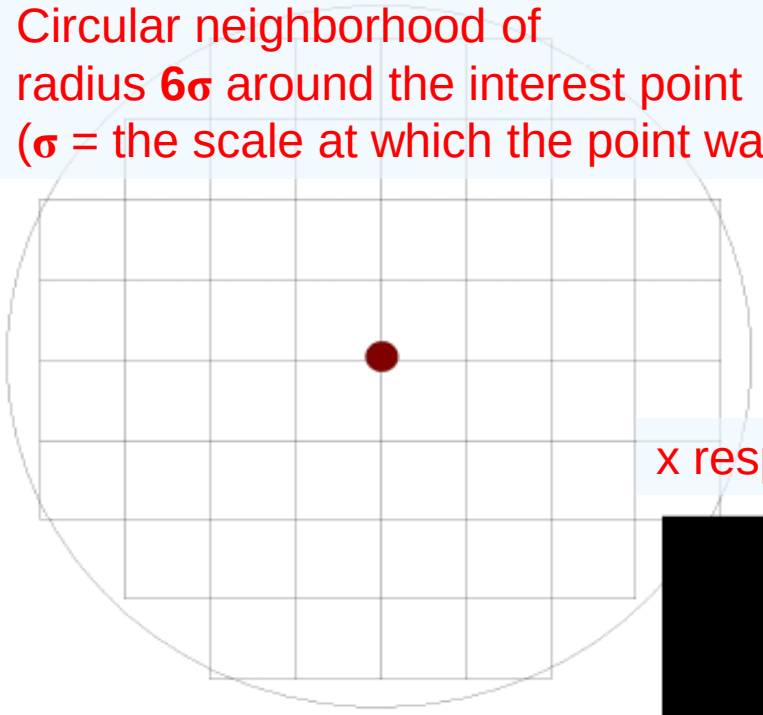
  (1) Orientation assignment

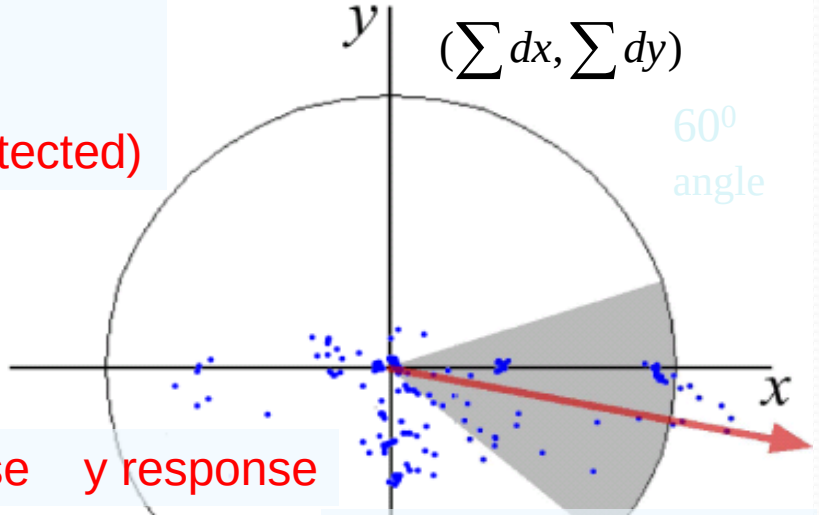
  (2) Keypoint descriptor

SURF: Speeded Up Robust Features (cont'd)

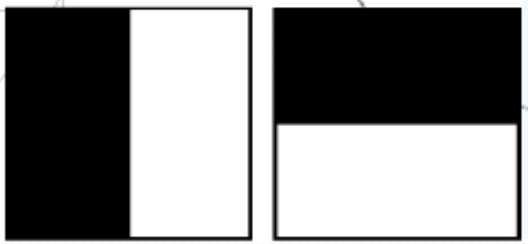
- Orientation assignment

Circular neighborhood of
radius **6σ** around the interest point
(**σ** = the scale at which the point was detected)





x response y response

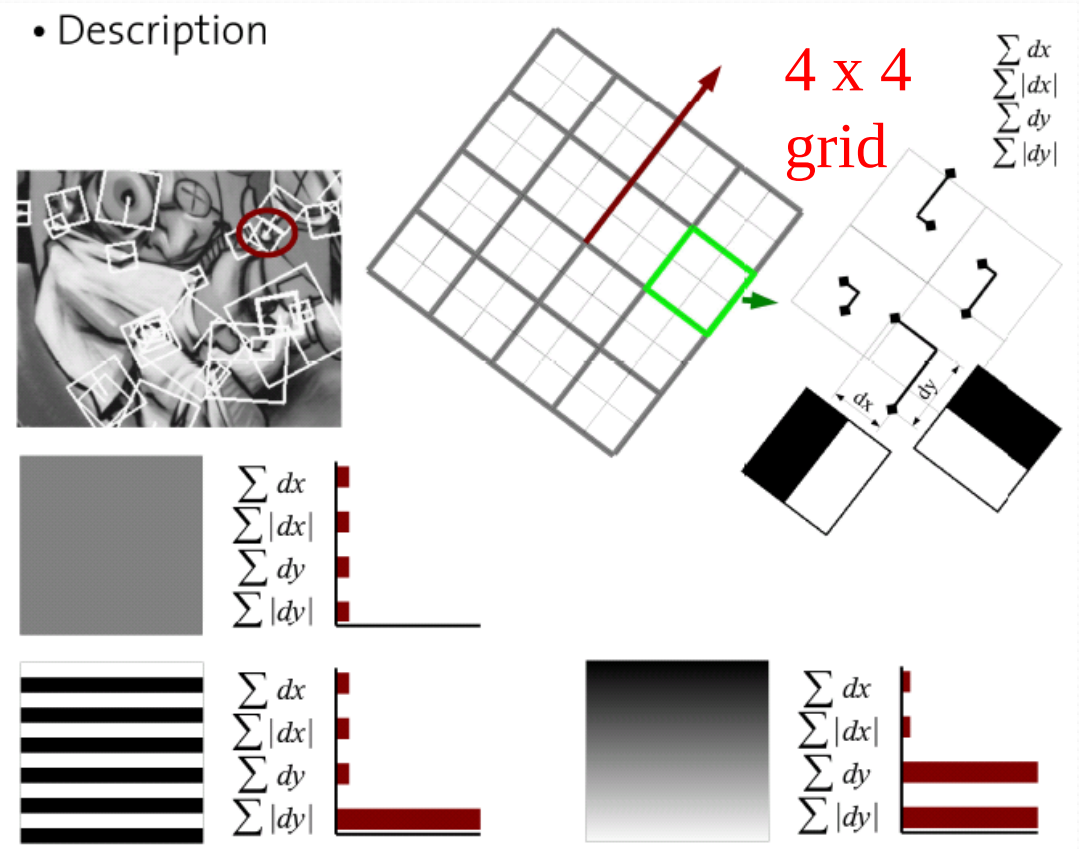


Haar wavelets
(responses weighted
with Gaussian)
Side length = **4σ**

Can be computed very fast using integral images!

SURF: Speeded Up Robust Features (cont'd)

- Keypoint descriptor (square region of size 20σ)

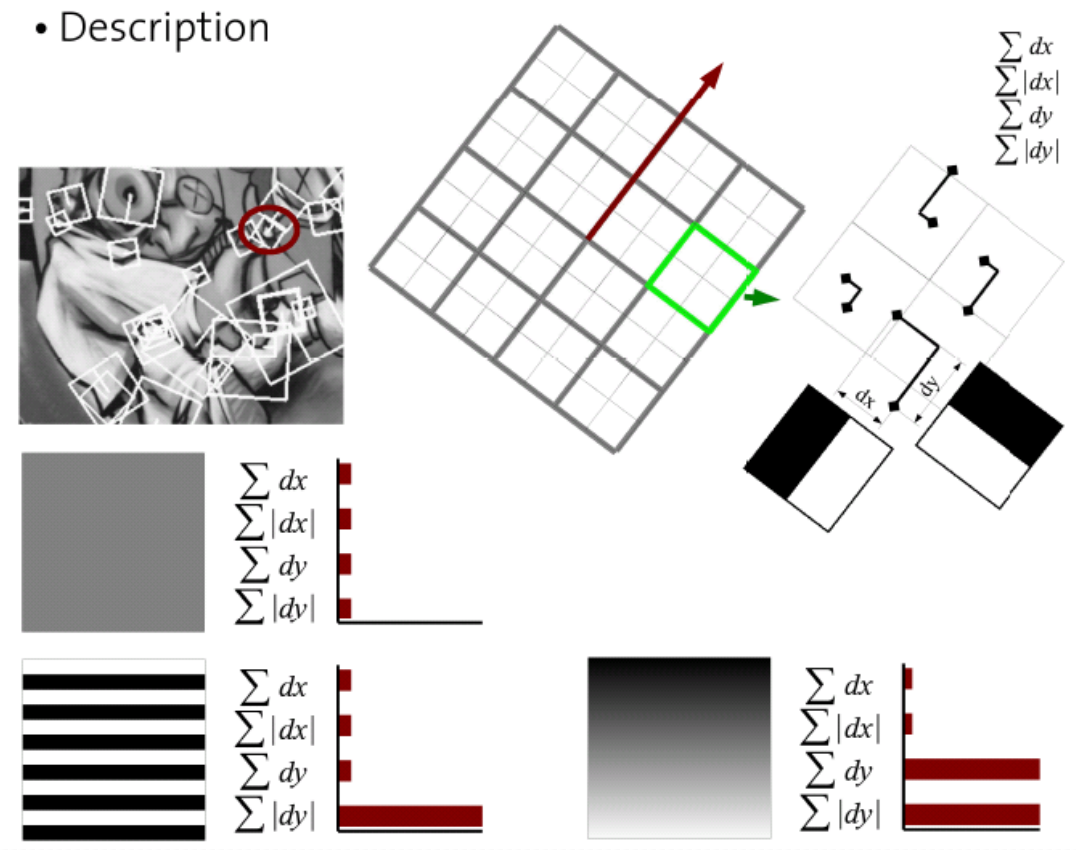


- Sum the response over each sub-region for d_x and d_y separately.
- To bring in information about the polarity of the intensity changes, extract the sum of absolute value of the responses too.

Feature vector size:

4 x 16 = 64

SURF: Speeded Up Robust Features (cont'd)



- SURF-128
  - The sum of d_x and $|d_x|$ are computed separately for points where $d_y < 0$ and $d_y > 0$
  - Similarly for the sum of d_y and $|d_y|$
  - **More discriminatory!**

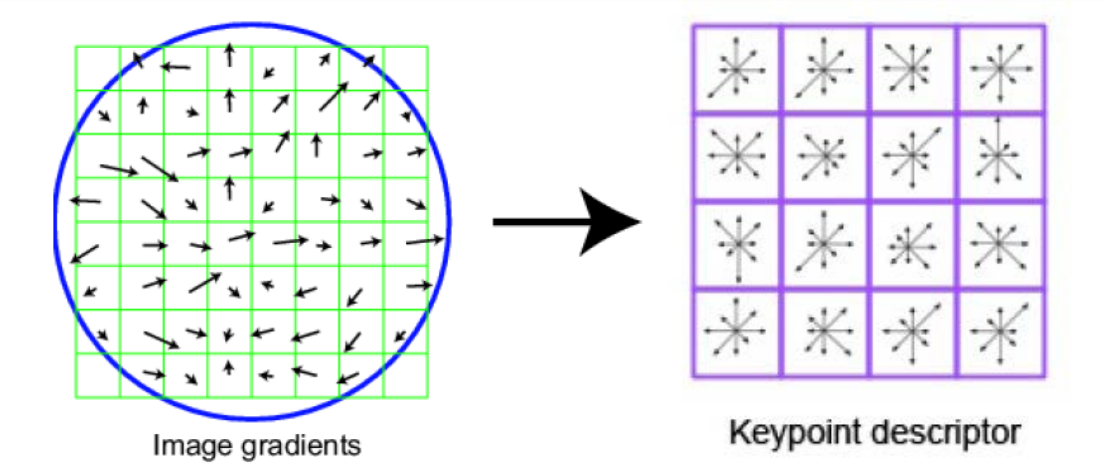
SURF: Speeded Up Robust Features

- Has been reported to be 3 times faster than SIFT.
- Less robust to illumination and viewpoint changes compared to SIFT.

K. Mikolajczyk and C. Schmid,"A Performance Evaluation of Local Descriptors", ***IEEE Transactions on Pattern Analysis and Machine Intelligence****, vol. 27, no. 10, pp. 1615-1630, 2005.*

Gradient location-orientation histogram (GLOH)

- Compute SIFT using a log-polar location grid:
  - 3 bins in radial direction (i.e., radius 6, 11, and 15)
  - 8 bins in angular direction
- Gradient orientation quantized in 16 bins.
- Total: (2x8+1)*16=272 bins →PCA.



*K. Mikolajczyk and C. Schmid,"A Performance Evaluation of Local Descriptors", **IEEE Trans. on Pattern Analysis and Machine Intelligence**, vol. 27, no. 10, pp. 1615-1630, 2005.*

Other descriptors

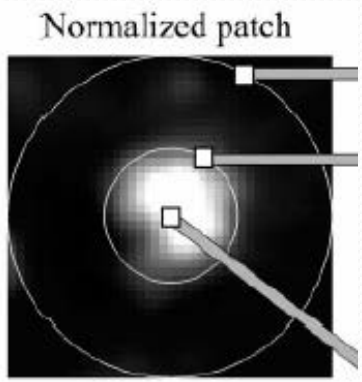
Shape Context

- A 3D histogram of edge point locations and orientations.
  - Edges are extracted by the Canny edge detector.
  - Location is quantized into 9 bins (using a log-polar coordinate system).
  - Orientation is quantized in 4 bins (i.e., (horizontal, vertical, and two diagonals).
- Total number of features: 4 x 9 = 36

K. Mikolajczyk and C. Schmid,"A Performance Evaluation of Local Descriptors", ***IEEE Transactions on Pattern Analysis and Machine Intelligence****, vol. 27, no. 10, pp. 1615-1630, 2005.*

Spin image

- A histogram of quantized pixel locations and intensity values.
  - A normalized histogram is computed for each of five rings centered on the region.
  - The intensity of a normalized patch is quantized into 10 bins.
- Total number of features: 5 x 10 = 50

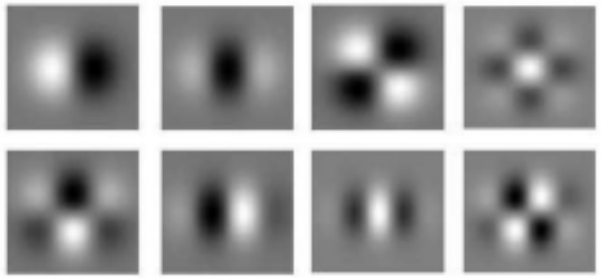


K. Mikolajczyk and C. Schmid,"A Performance Evaluation of Local Descriptors", ***IEEE Transactions on Pattern Analysis and Machine Intelligence****, vol. 27, no. 10, pp. 1615-1630, 2005.*

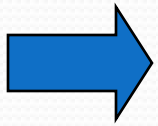
Differential Invariants

- "Local jets" of derivatives obtained by convolving the image with Gaussian derivates.
- Derivates are computed at different orientations by rotating the image patches.

Example: some Gaussian derivatives up to fourth order



$$\mathbf{v}(x,y) = \begin{pmatrix} I(x,y) * G(\sigma) \\ I(x,y) * G_x(\sigma) \\ I(x,y) * G_y(\sigma) \\ I(x,y) * G_{xx}(\sigma) \\ I(x,y) * G_{xy}(\sigma) \\ I(x,y) * G_{yy}(\sigma) \\ \vdots \end{pmatrix} = \begin{pmatrix} L(x,y) \\ L_x(x,y) \\ L_y(x,y) \\ L_{xx}(x,y) \\ L_{xy}(x,y) \\ L_{yy}(x,y) \\ \vdots \end{pmatrix}$$



invariants

Differential Invariants

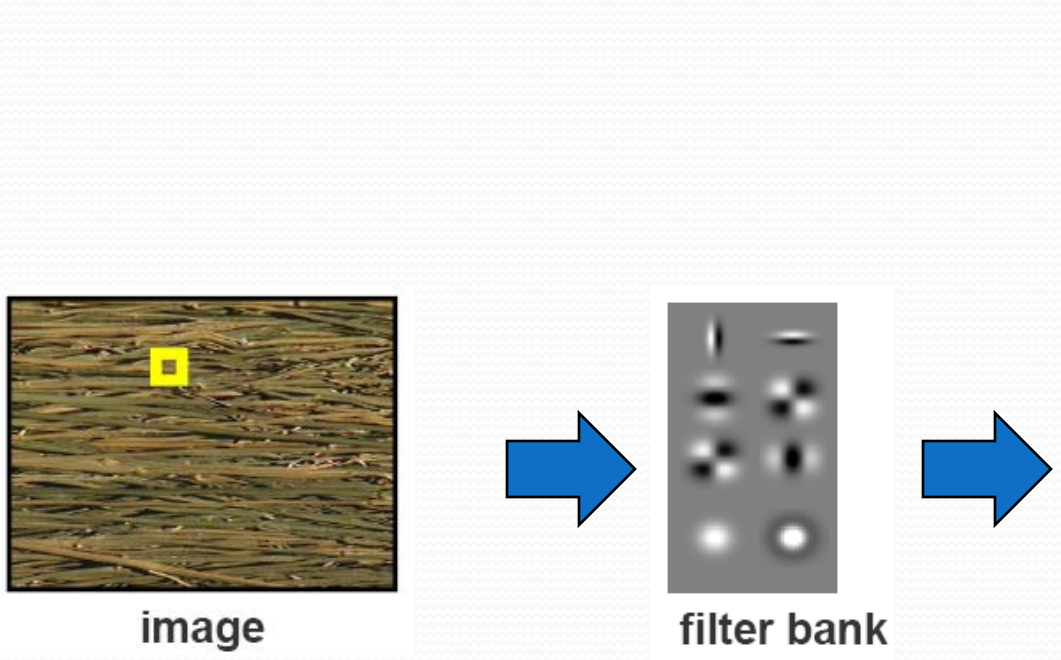
Invariance to image rotations: differential invariants

norme du gradient

Laplacien

$$\begin{bmatrix} L \\ L_x L_x + L_y L_y \\ L_{xx} L_x L_x + 2 L_{xy} L_x L_y + L_{yy} L_{yy} \\ L_{xx} + L_{yy} \\ L_{xx} L_{xx} + 2 L_{xy} L_{xy} + L_{yy} L_{yy} \\ \dots \\ \dots \\ \dots \\ \dots \end{bmatrix}$$

Bank of Filters



(e.g., Gabor filters)

Moment Invariants

- Moments are computed for derivatives of an image patch using:

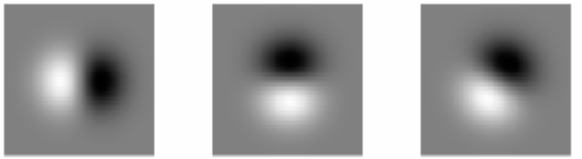
$$M_{pq}^{a} = \frac{1}{xy} \sum_{x,y} x^{p} y^{q} [I_{d}(x, y)]^{a}$$

where p and q is the order, a is the degree, and I_d is the image gradient in direction d.

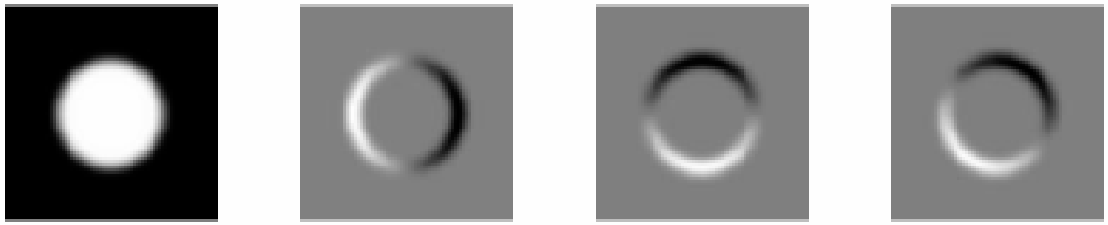
- Derivatives are computed in x and y directions.

Bank of Filters: Steerable Filters

$$
\begin{aligned}
R_1^{0^\circ} &= G_1^0 * I \\
R_1^{90^\circ} &= G_1^{90} * I \\
\text{then} & \\
R_1^{\theta} &= \cos(\theta) R_1^{0^\circ} + \sin(\theta) R_1^{90^\circ}
\end{aligned}
$$



a b c



d e f g

Figure 1: Example of steerable filters. (a) $G_1^{0^\circ}$, first derivative with respect to x (horizontal) of a Gaussian. (b) $G_1^{90^\circ}$, which is $G_1^{0^\circ}$, rotated by 90°. From a linear combination of these two filters, one can create G_1^{θ}, an arbitrary rotation of the first derivative of a Gaussian. (c) $G_1^{30^\circ}$, formed by $\frac{1}{2}G_1^{0^\circ} + \frac{\sqrt{3}}{2}G_1^{90^\circ}$. The same linear combinations used to synthesize G_1^{θ} from the basis filters will also synthesize the response of an image to G_1^{θ} from the responses of the image to the basis filters: (d) Image of circular disk. (e) $G_1^{0^\circ}$ (at a smaller scale than pictured above) convolved with the disk, (d). (f) $G_1^{90^\circ}$ convolved with (d). (g) $G_1^{30^\circ}$ convolved with (d), obtained from $\frac{1}{2}$ [image e] $+\frac{\sqrt{3}}{2}$ [image f].

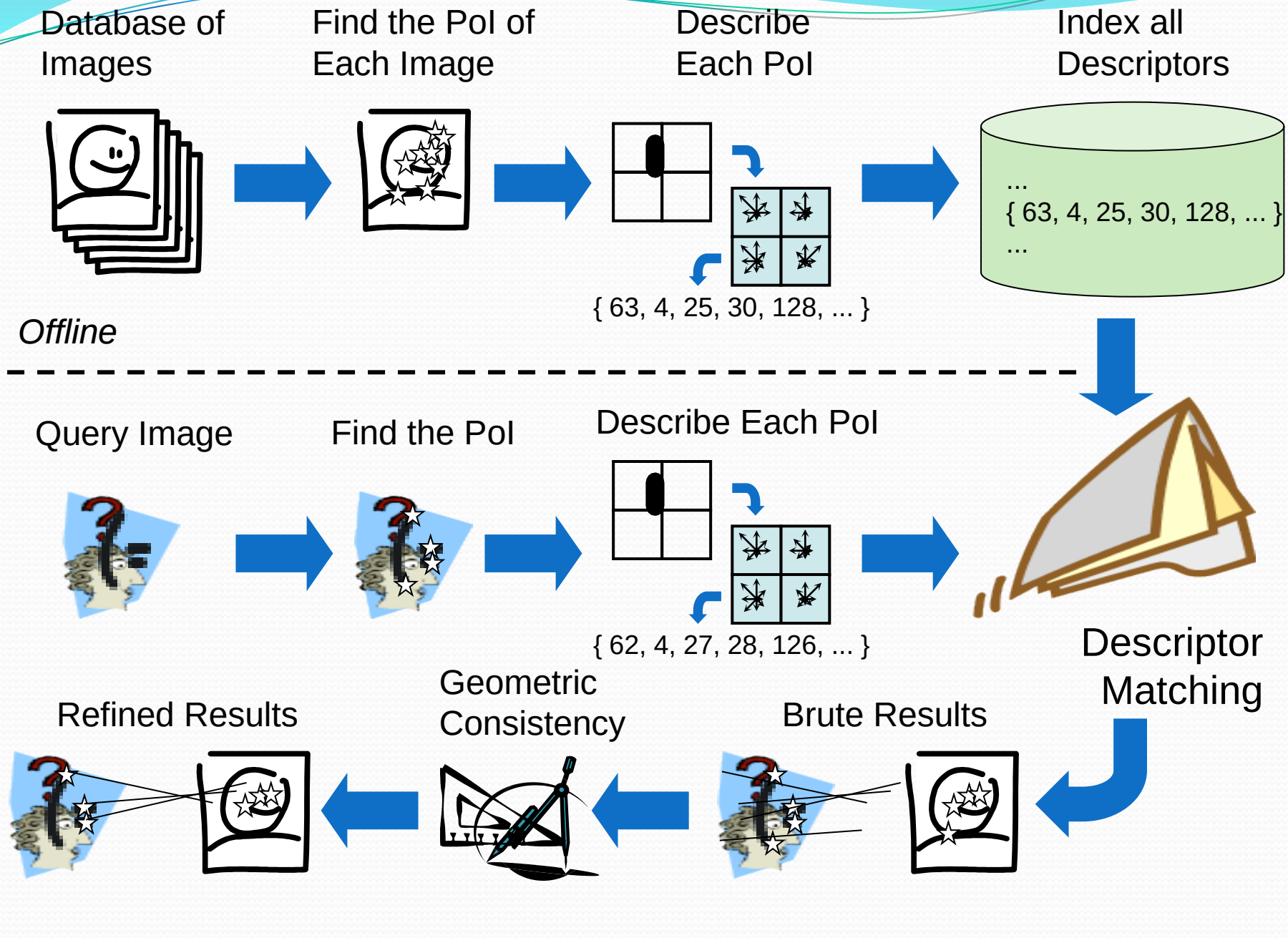
Descriptors : Conclusion

- Many descriptors
- Many evaluations/comparisons

  Details on the most powerful => SIFT
- Best choice depend often on the application
- Often better results by combining descriptors
- Question: how to exploit these descriptions ?

Bag of Word : introduction

- Back to our target retrieval problem
  - Requires to find the same objects with deformations
  - Winning combination: sparse detector PoI + discriminant and invariant descriptors, such as SIFT
  - Local => so intrinsically robust to occlusions and noise (of the background)
  - Challenge: fast search of nearest neighbors in very large datasets
  - Geometrical consistency to constrain correspondances


Database of Images
Find the PoI of Each Image
Describe Each PoI
Index all Descriptors
...
{ 63, 4, 25, 30, 128, ... }
...
{ 63, 4, 25, 30, 128, ... }
Offline
Query Image
Find the PoI
Describe Each PoI
{ 62, 4, 27, 28, 126, ... }
Descriptor Matching
Brute Results
Geometric Consistency
Refined Results

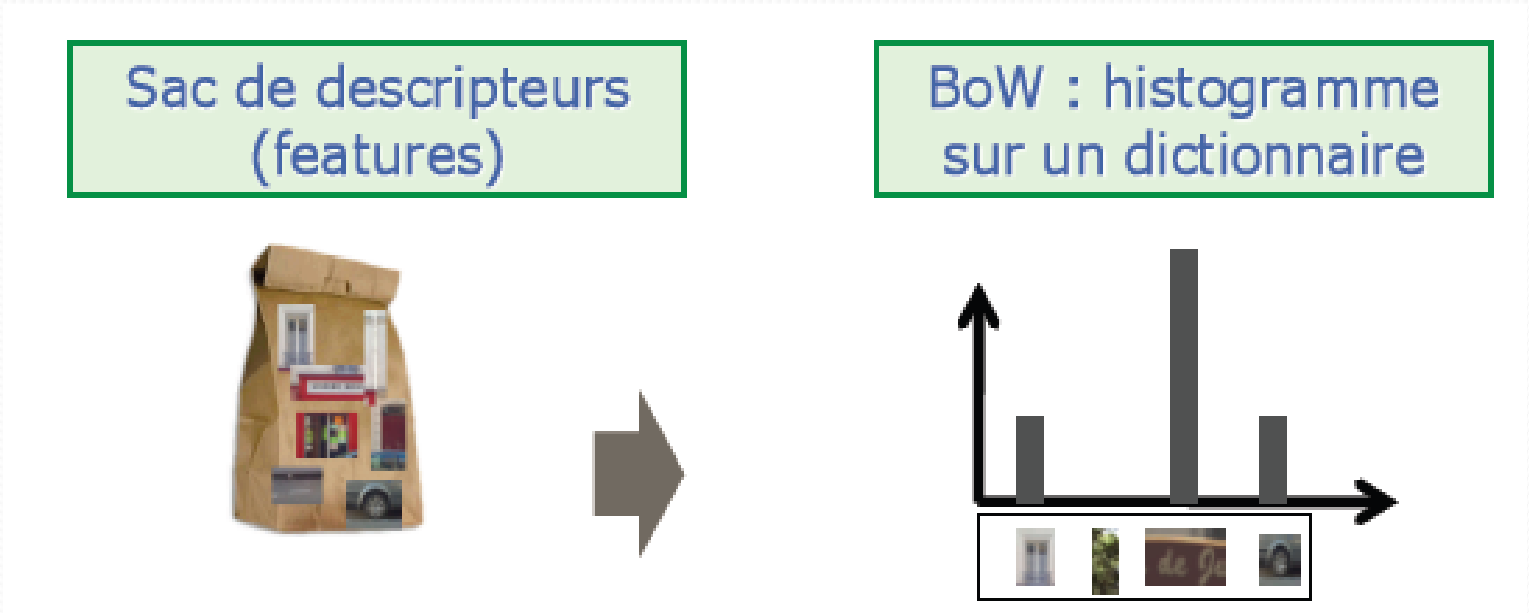
Bag of Word: introduction

- Target retrieval efficient with local descriptor matching
  - But not scalable for (very) large datasets or very large images
  - Approximation required

- Categorization: we do not look for “identical” images
  - Model providing state-of-art results: “**Bag of Word”**
  - Text-mining approach

From Bag of Features (BoF) to (BoW)

- BoF: Local descriptor ensembles
  - How to compare two images ? => signature vectorielle
  - Trade-off discrimination / generalization

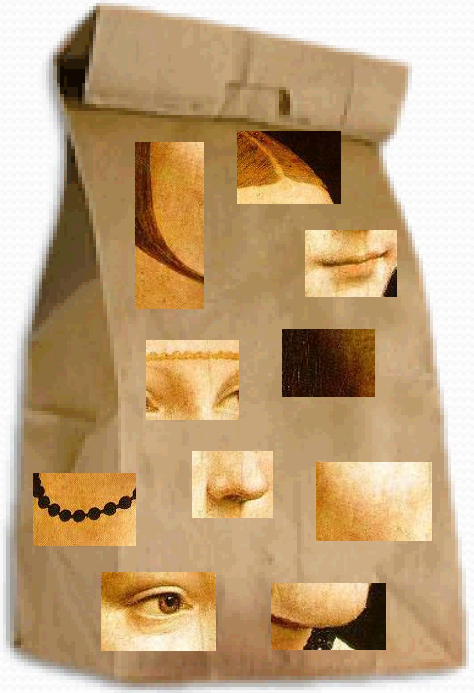


Slide credit L. Fei-Fei

Bag of Visual Words (BoW)

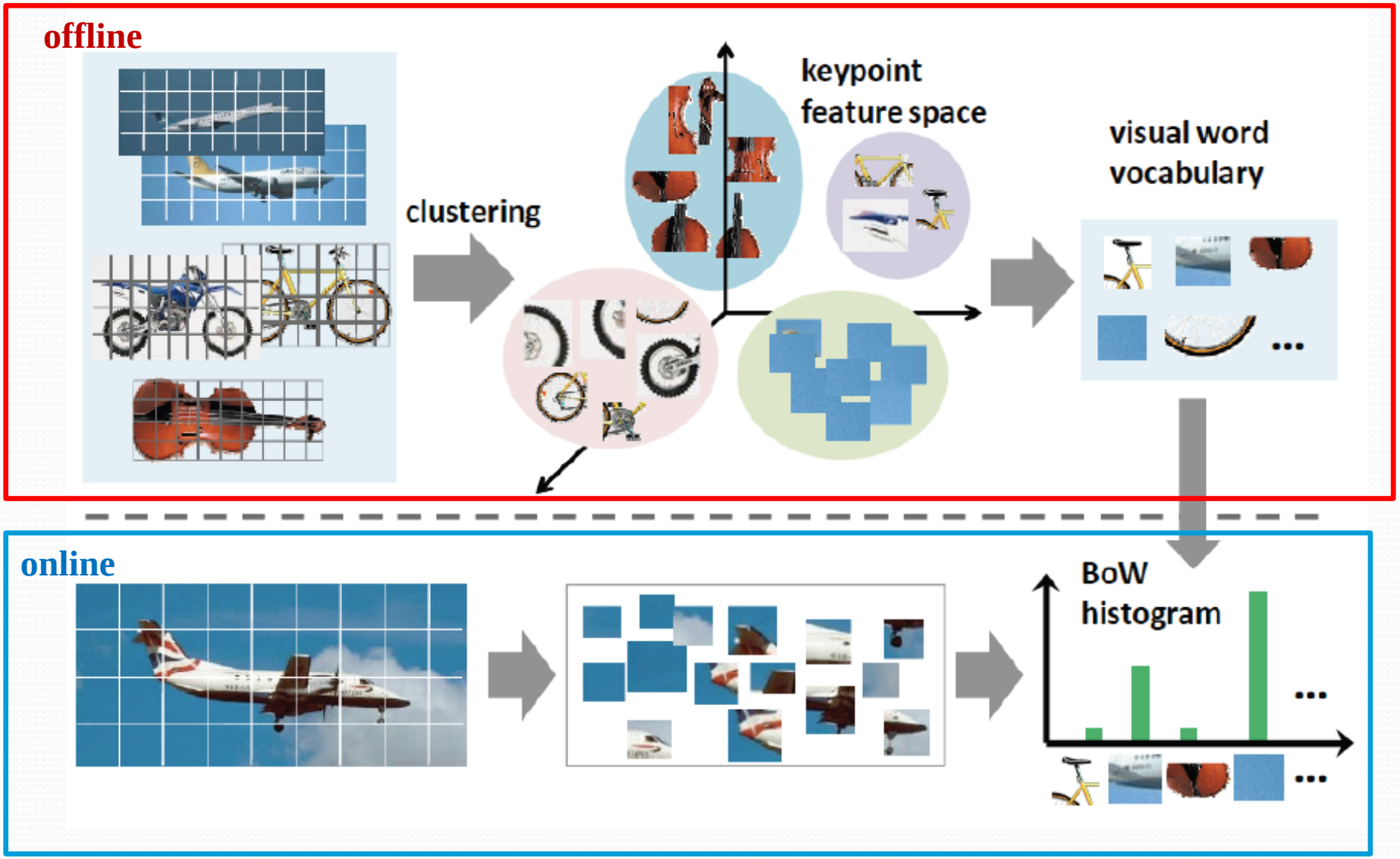
Image

Bag of Visual 'Words'



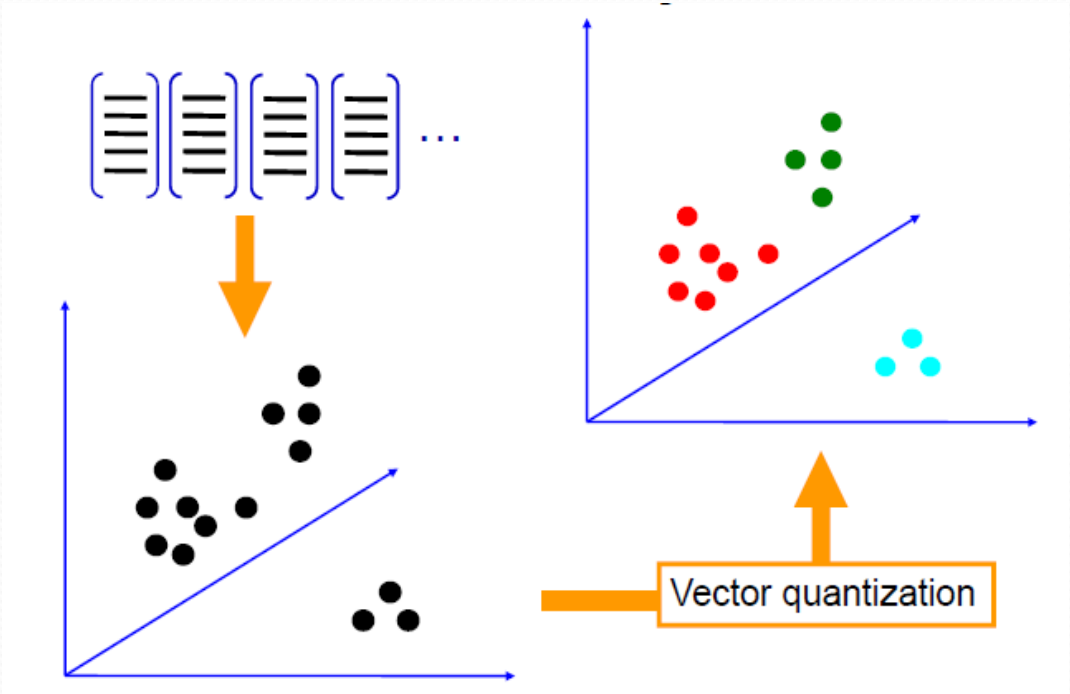
Slide credit L. Fei-Fei

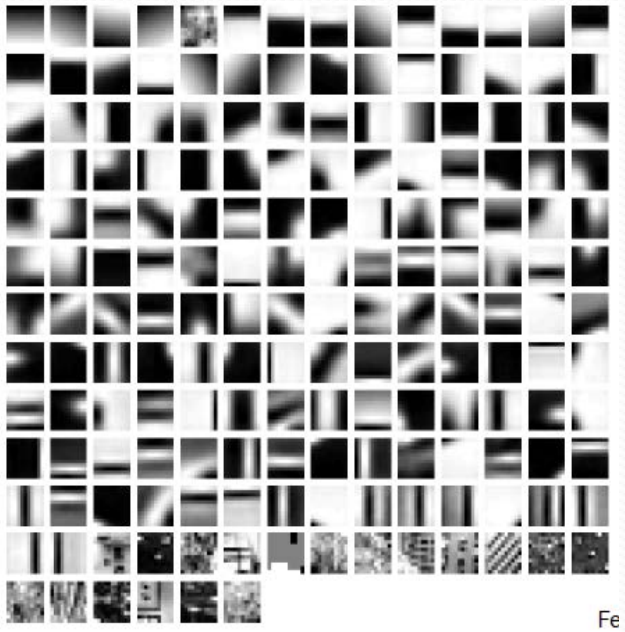
Bag of Visual Words (BoW)



Bag of Visual Words: off-line part

- Descriptor extraction, visual dictionary based approaches
  - Construction of a dictionary of visual patterns (words) adapted to the dataset, *e.g.* K-Means





Bag of Visual Words: off-line part

- Descriptor extraction, visual dictionary based approaches
  - K-Means
    - Limitations: aggregate clusters on high density areas, leave the others
    - Alternatives: Radial basis function methods
    - In practice, K-Means remains the most prominent method
    - Dictionary learning (non supervised or supervised)