Computer vision

Local descriptors

Doc. Ing. Vanda Benešová, PhD.

Basic concept of local description (repet.)

Local desriptors

More robust

Occlusions of objects

Changes of camera view position

Rotation, scale invariance

Intra category variations

http://www.cs.utexas.edu/~grauman/courses/fall2009/papers/local_features_synthesis_draft.pdf

http://vgg.fiit.stuba.sk/kniha/

Local description (detectors + desriptors)

What we need:

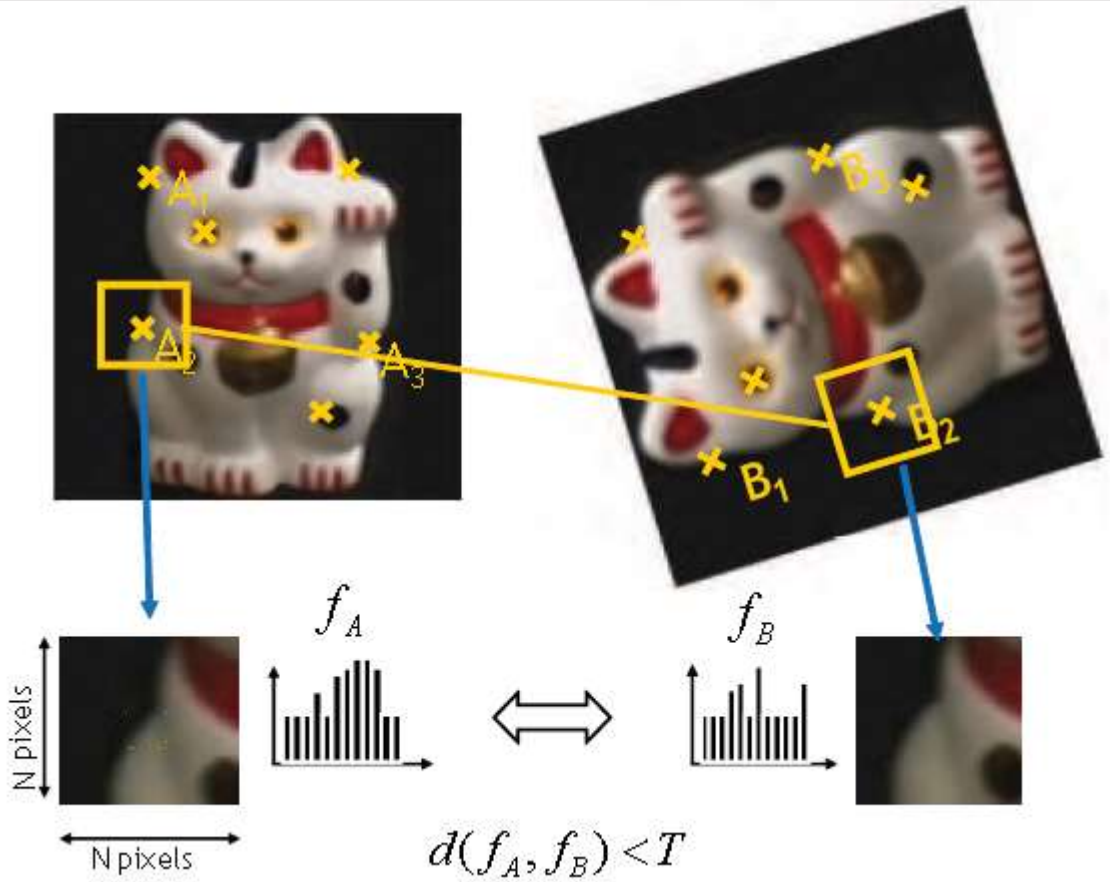
- Local description (detectors + desriptors) includes:

  Local detectors - Detection of interesting points (IP) –key-points

  Local descriptors - Description of surrounding area of the key-points in the form of a descriptor.

- Matching algorithm

An illustration of the recognition procedure with local features

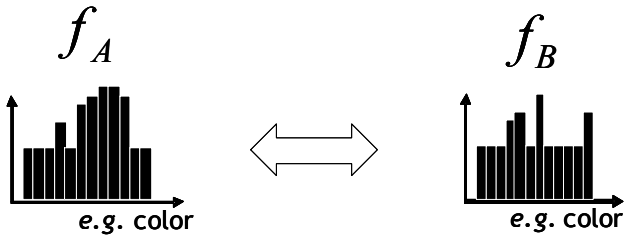


Scale- and rotation-invariant!

FIGURE FROM Krystian Mikolajczyk

Object detection using local descriptors

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



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

Methods of local description examples of detectors and desriptors

Local detectors :

- Harris corner detector
- Shi-Thomasi „Good features to track“
- FAST detector

Local descriptors

- Float : SIFT, SURF, DAISY, FREAK
- Binary: ORB, BRIEF

Example – Logo detection

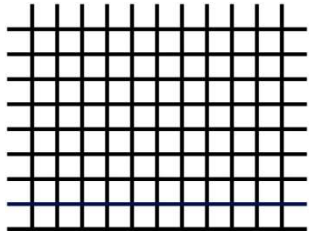
Paper2010_Oliver_Michal\Deskriptory_Videa

Local detectors

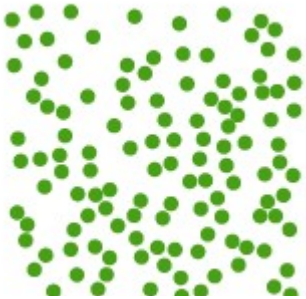
Harris corner detector
Shi-Tomasi detector
Hessian point detector
Fast key-point detector

DETECTION OF INTEREST POINTS AND REGIONS - KEYPOINT LOCALIZATION

Regular



Random



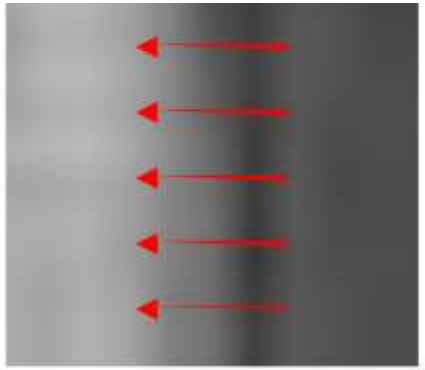
Interest point detector

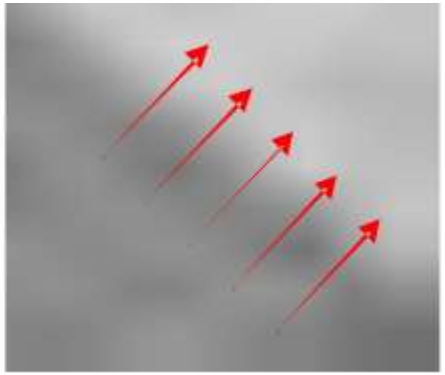
Harris corner detector

Intuition:

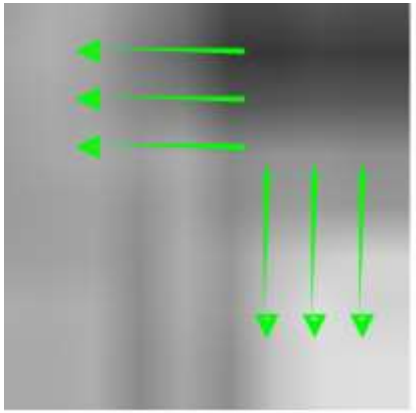
Search for local neighborhoods where the image content has two main directions.

undistinguished patches:





distinguished patch:



HARRIS, Chris; STEPHENS, Mike. A combined corner and edge detector. In: *Alvey vision conference*. 1988. p. 10.5244.

Harris corner detector

The detector uses a derivation of grayscale images.

Harris detector M (x, y) is defined by the array of derivatives (differences) Ix, Iy of the image intensity I:

$$M(\mathbf{x}) = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

where I is the image I (x, y),

w (x, y) is a function of the window which is determined as the weighted sum (Gaussian core),

Ix, Iy are the derivatives along the axis x and y.

Corner detector

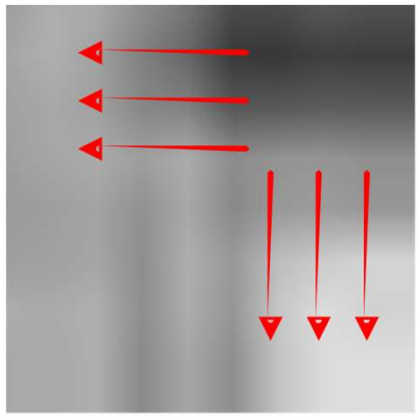
For detecting of corners are important eigenvalues of matrix M, namely ($\lambda 1$, $\lambda 2$).

($\lambda 1$, $\lambda 2$) reflect the variance in two main ways.

If the variance in both directions is large (eigenvalues $\lambda 1$, $\lambda 2$ are large enough), the point is located at the corner (corner point), and if it is large only one value, lies on the edge (one direction).

Corner detection using eigenvalues of matrix M

(λ1,λ2) = eigenvalues(M)



"Corner"

λ_1 and λ_2 are large,

$\lambda_1 \sim \lambda_2$;

E increases in all directions

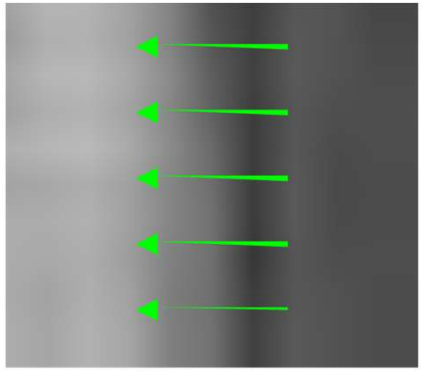
"Flat" region

λ_1 and λ_2 are small;

E is almost constant in all directions

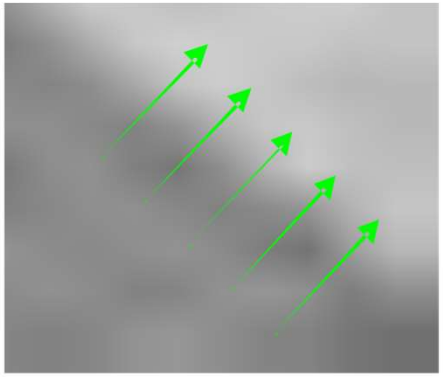
"Edge"

$\lambda_1 >> \lambda_2$



"Edge"

$\lambda_2 >> \lambda_1$



Metrics of corner detectors (Harris + Shi-Tomasi detector)

Originaly used metric by [Harris (1988)]

$R(x) = \det(M(\mathbf{x})) - k\,\mathrm{trace}(M(\mathbf{x}))2$

This metric has been simplified and streamlined in Article [J. Shi a C. Tomasi 94]:

~~The corner detector searches for image neighbourhoods where the second-moment matrix M has :~~

~~two eigenvalues > Th~~

~~-> corresponding to two dominant orientations~~

HARRIS, Chris; STEPHENS, Mike. A combined corner and edge detector. In: *Alvey vision conference*. 1988. p. 10.5244.

J. Shi a C. Tomasi. „Good features to track". *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*. Jún 1994, str. 593–600.

Metrics of corner detector
Shi-Tomasi detector („Good features to track")

$$R(x) = \min(\lambda 1, \lambda 2)$$

Decision:

$$R(x) > Th$$

J. Shi a C. Tomasi. „Good features to track". *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*. Jún 1994, str. 593–600.

Harris corner detector

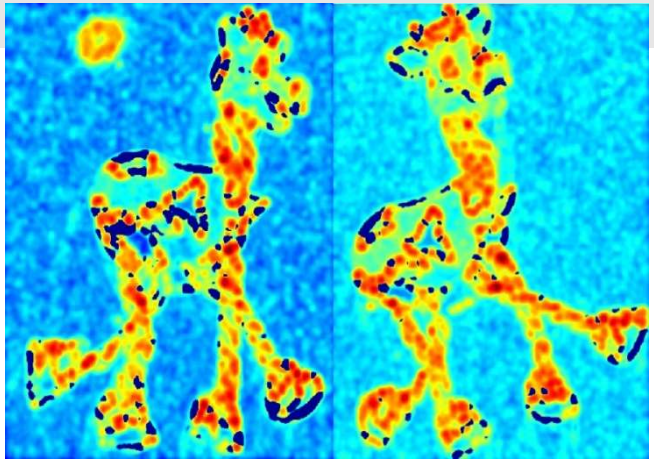
Algorithm properties:

+ invariant to 2D image shift and rotation
+ invariant to shift in illumination
+ invariant to small view point changes
+ low numerical complexity

- not invariant to larger scale changes
- not invariant to high contrast changes
- not invariant to bigger view point changes

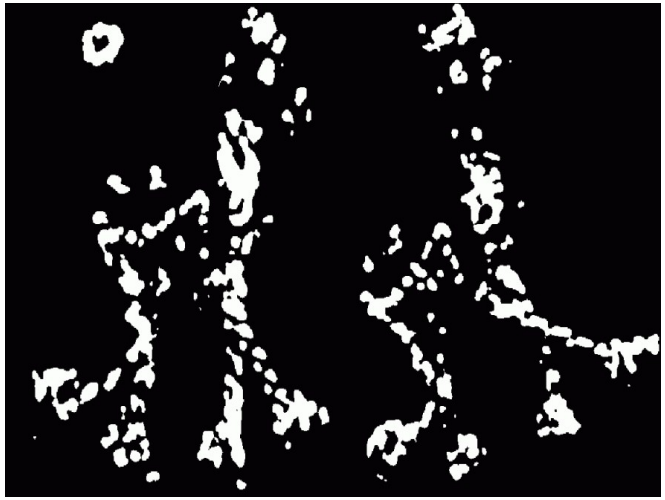
Harris corner detector example

Compute corner response *R:*

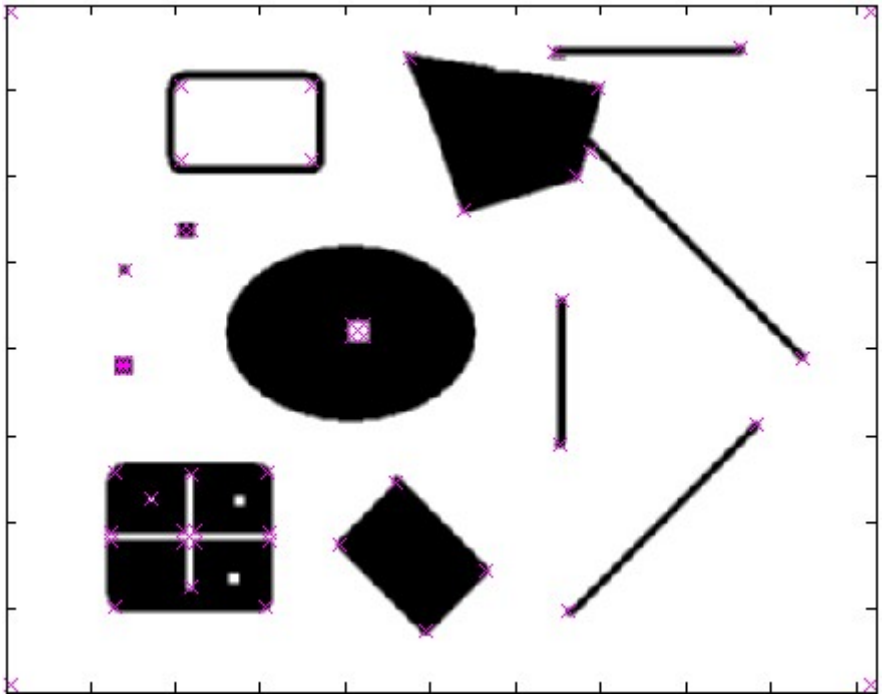


Take only the points of local maxima of R

Find points with large corner response:
R>threshold:



Harris corner detector



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

Hessian point detector

Intuition:

Search for strong derivatives in two orthogonal directions

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

I_{xx} is second partial derivative in the a direction x

I_{xy} is mixed partial second derivative in the x and y directions.

The detector computes the second derivatives Ixx, Ixy, and Iyy for each image point and then searches for points where the determinant of the Hessian becomes maximal:

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

BEAUDET, Paul R. Rotationally invariant image operators. In: International Joint Conference on Pattern Recognition. 1978. p. 583.

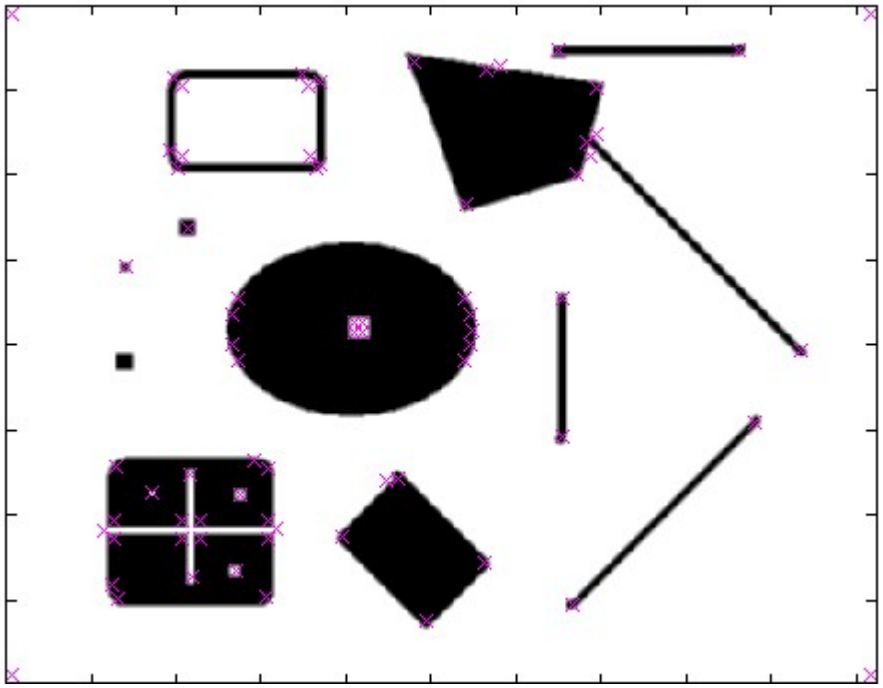
Hessian point detector

- a result image containing the Hessian determinant values

- and then applying *non-maximum suppression* using a 3 × 3 window.

- the search window is swept over the entire image, keeping only pixels whose value is larger than the values of all 8 immediate neighbours inside the window
- the detector then returns all remaining locations whose value is above a pre-defined threshold θ.

Hessian point detector

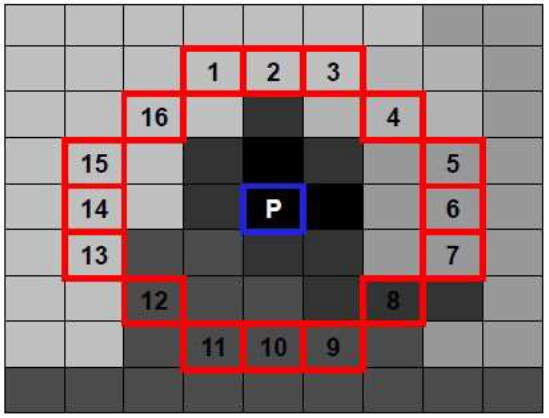


Effect: **Responses mainly on corners and strongly textured areas.**

FAST keypoint detector

The algorithm works on each point of the P picture on the Bresenhamovho (rasterized) circle of radius r.

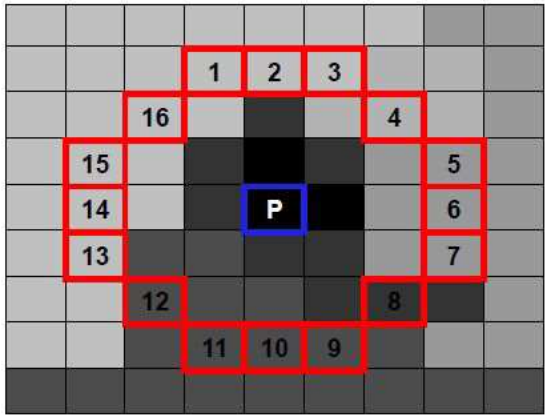
If exist
n neighboring pixels their intensity is higher or
n neighboring pixels their intensity is lower
compared with the origin pixel value P
-> then the origin pixel P is considered as a key-point.



n>threshold

FAST keypoint detector example

Example of detection of interesting points using FAST method:



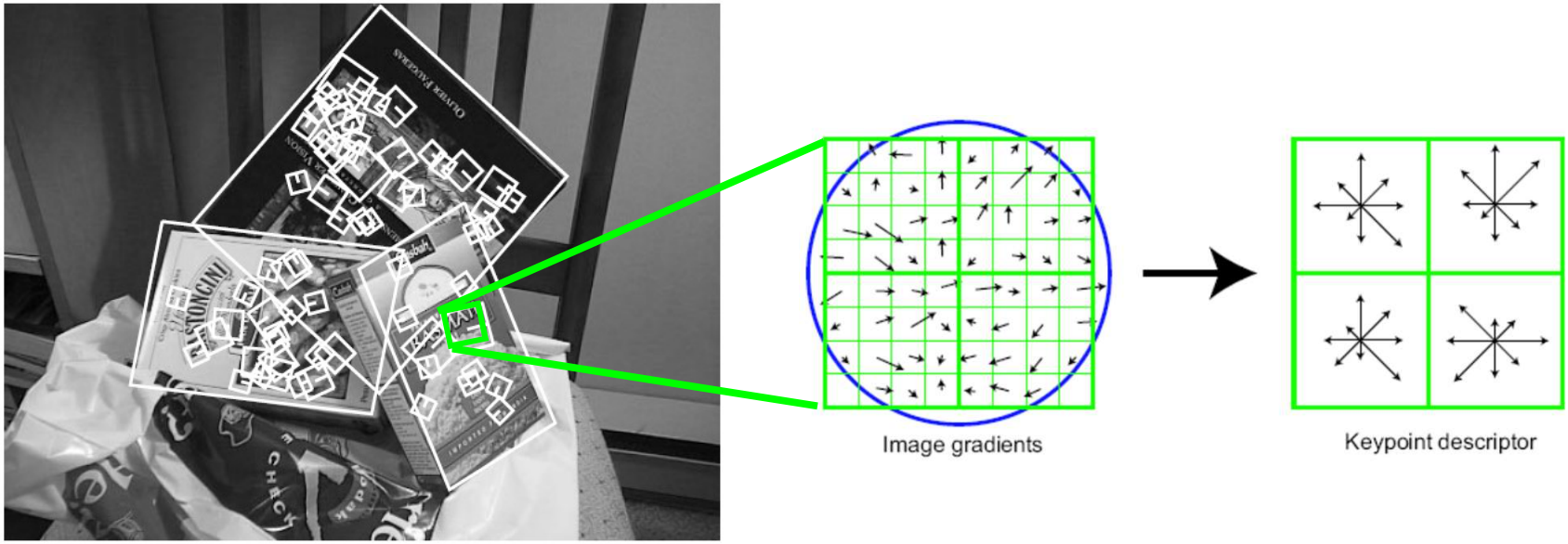
P is origin point,

in the are given by distance r = 3 exist 11 of adjacent points (1-7, 13-16) which intensity of is higher than P.

Point P is detected as an interesting point.

SIFT (detector + descriptor)

SIFT
Scale-invariant feature transform



Histogram of oriented gradients

Captures important texture information

Robust to small translations / affine deformations

LOWE, David G. Object recognition from local scale-invariant features. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Ieee, 1999. p. 1150-1157.

Scale-invariance
SIFT detector

SIFT detector

- Key-point detector: based on Hessian point detector
- Scale invariance
- Orientation invariance

SIFT detector includes Automatic scale selection

The principle of automatic scale selection :

For a given a key-point location,

we evaluate a scale-dependent signature function on the key-point neighbourhood and plot the resulting value as a function of the scale.

SIFT detector includes Automatic scale selection

corresponding neighbourhood sizes can be determined by searching for scale-space extrema of the signature function independently in both images.

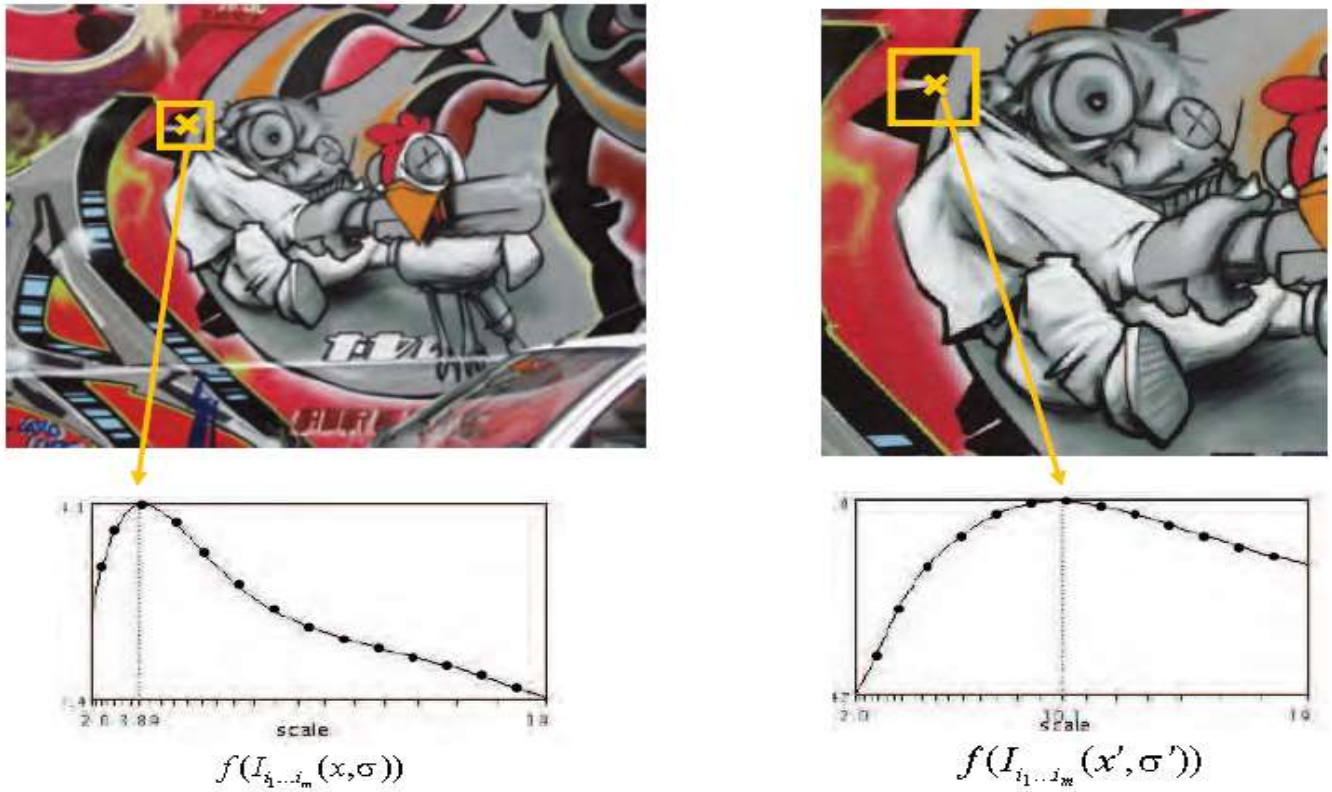
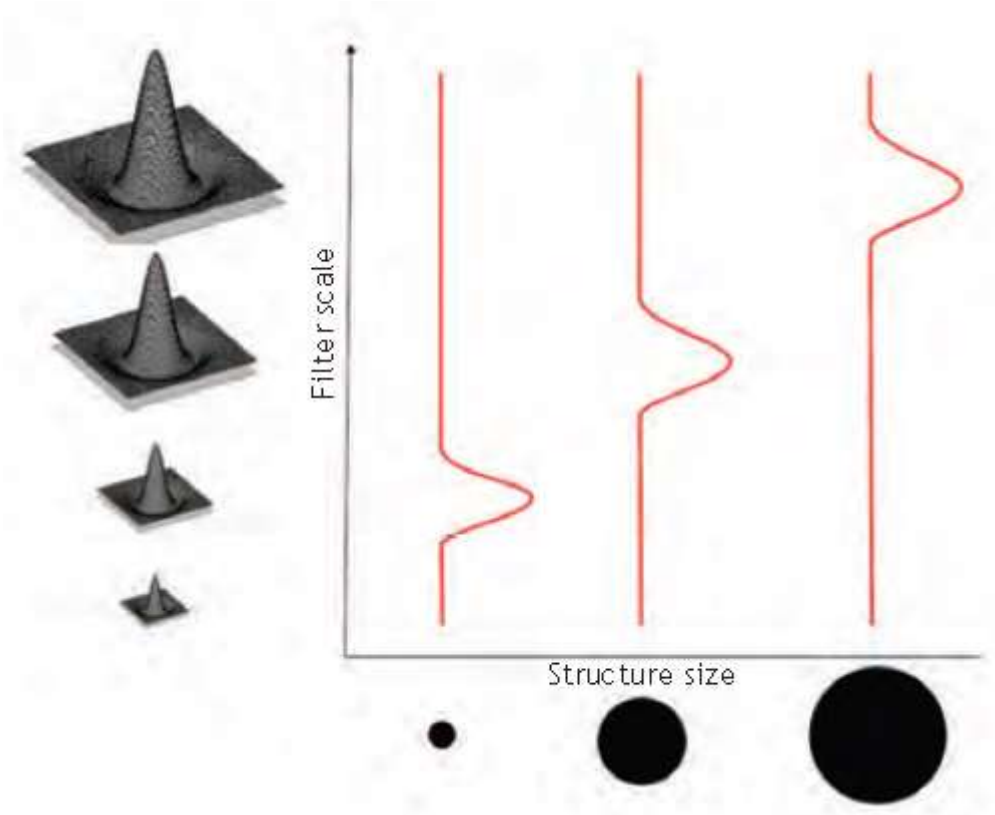


FIGURE FROM

Krystian Mikolajczyk

SIFT - Useful Signature Function

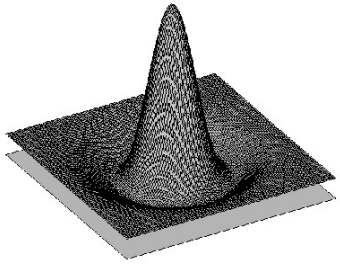
Laplacian-of-Gaussian = "blob" detector

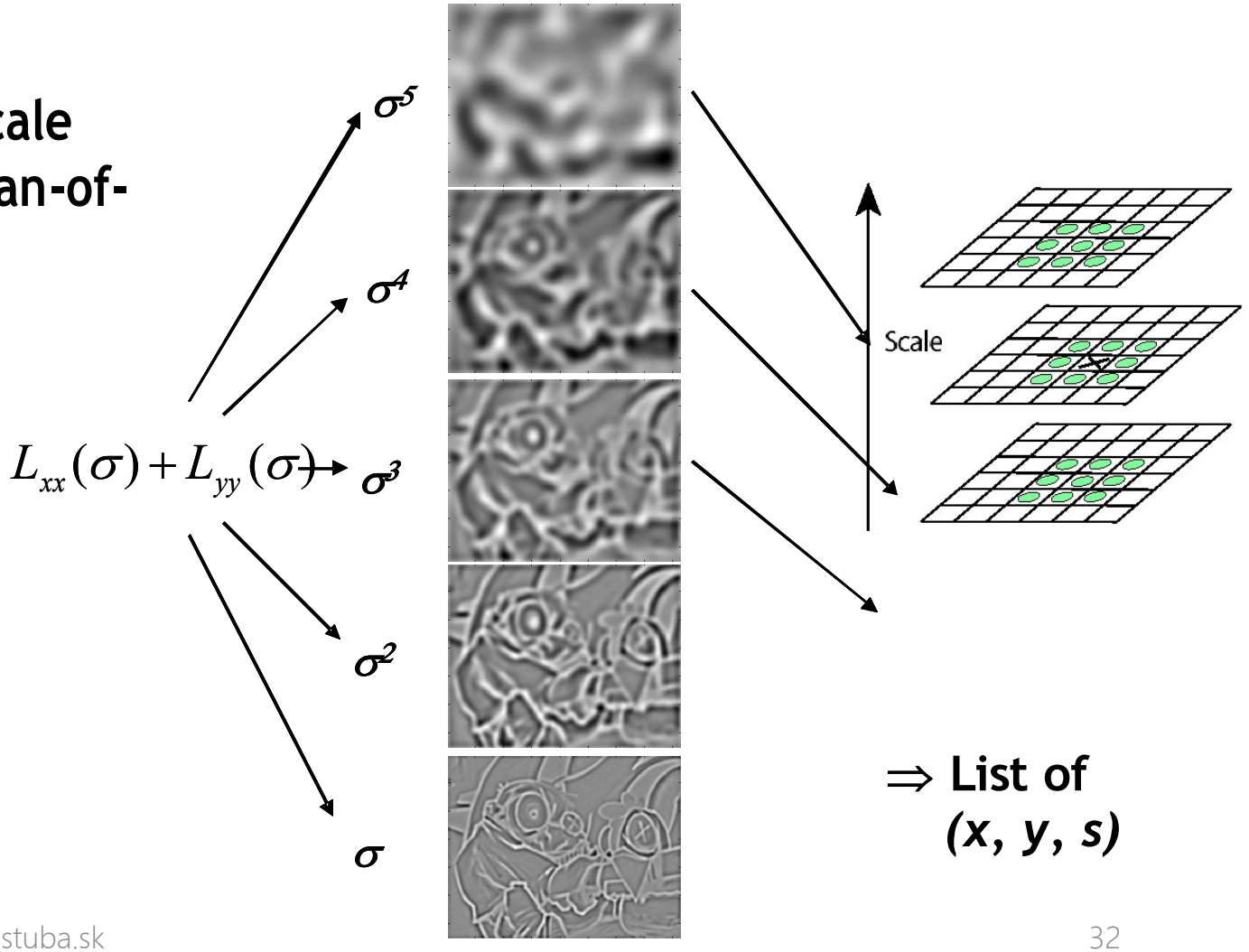


LoG filter mask corresponds to a circular center-surround structure, with positive weights in the center region and negative weights in the surrounding ring structure.

Laplacian of Gaussian (LOG)

Local maxima in scale space of Laplacian-of-Gaussian

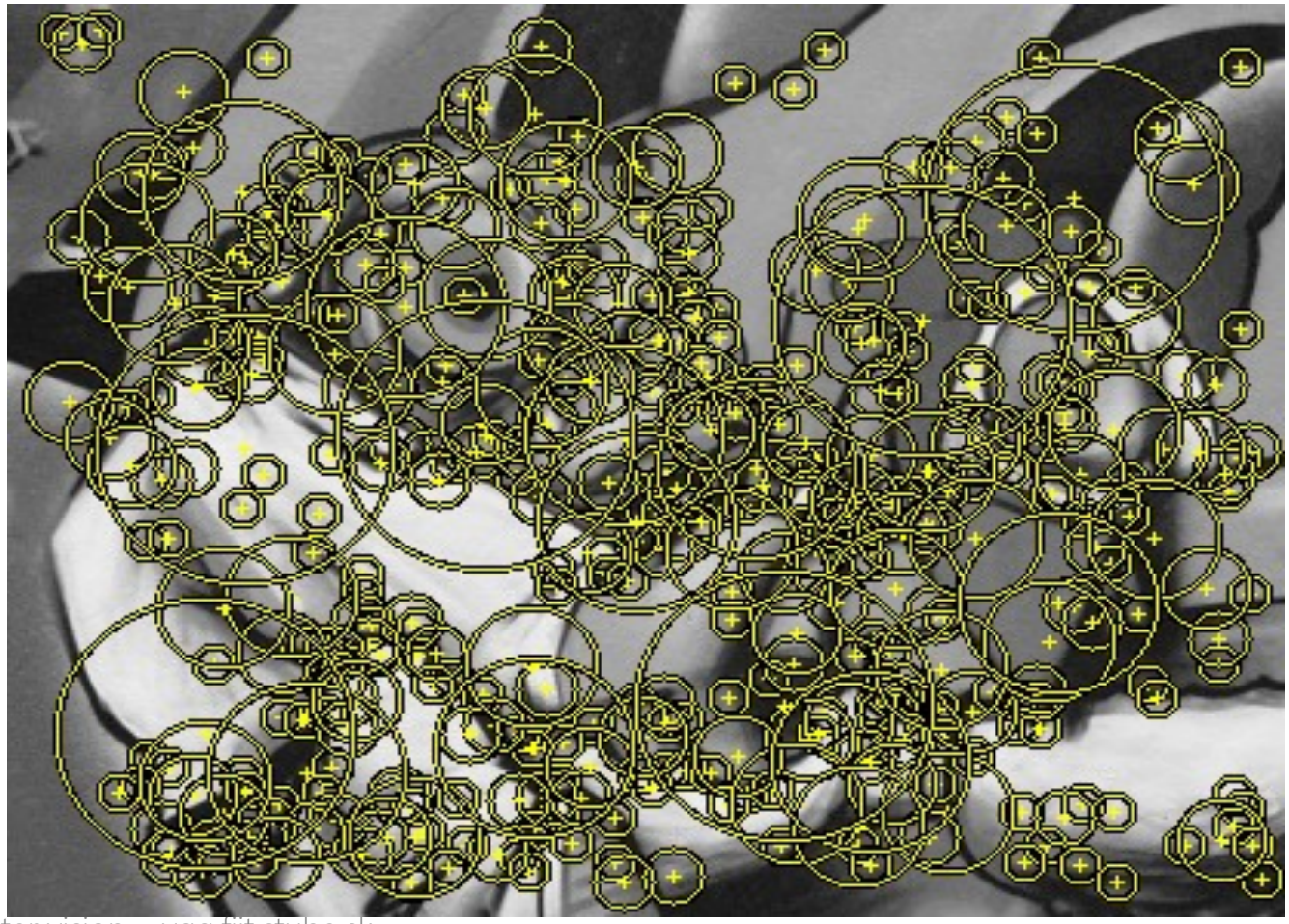




⇒ **List of** ***(x, y, s)***

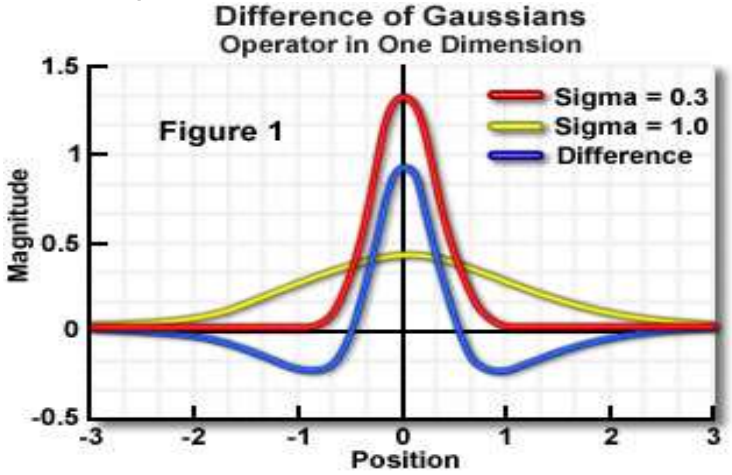
K. Grauman, B. Leibe

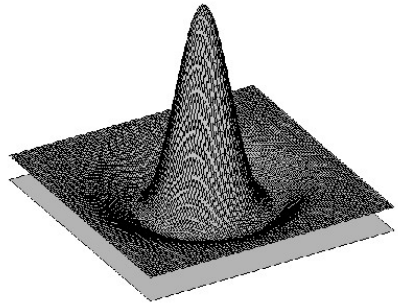
SIFT
detected size of patches used for descriptor



Difference of Gaussians (DOG) as approximation of the Laplacian-of-Gaussian

Difference of Gaussian images (DOG) are approximately equivalent to the Laplacian of Gaussian (LOG)

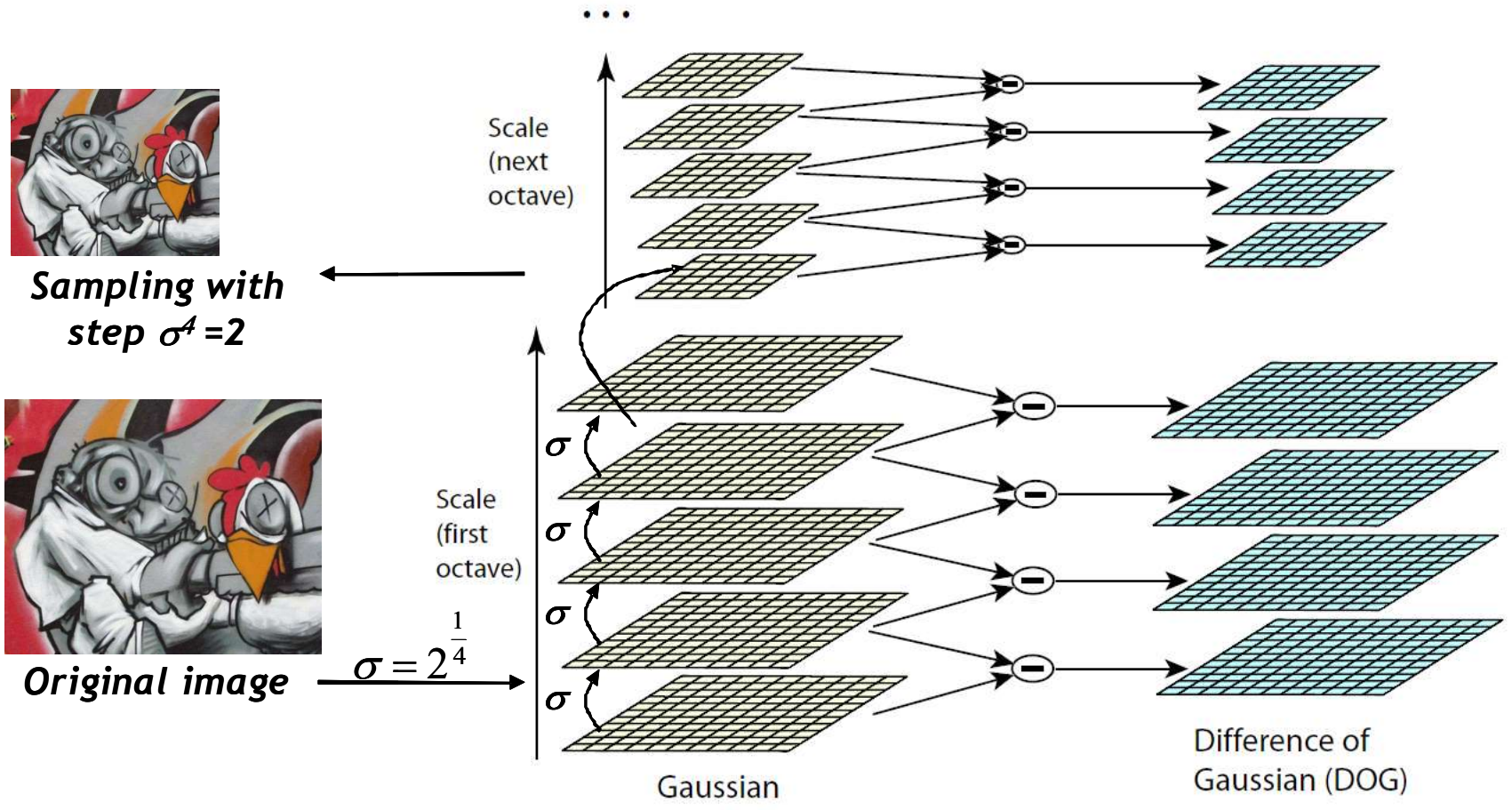




-

=

Computation of DOG

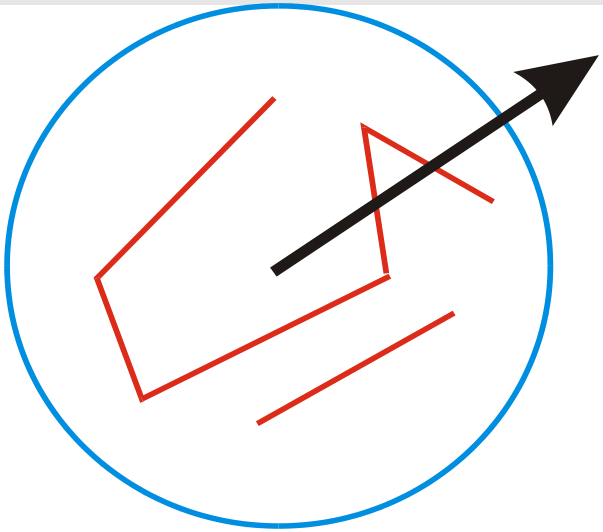


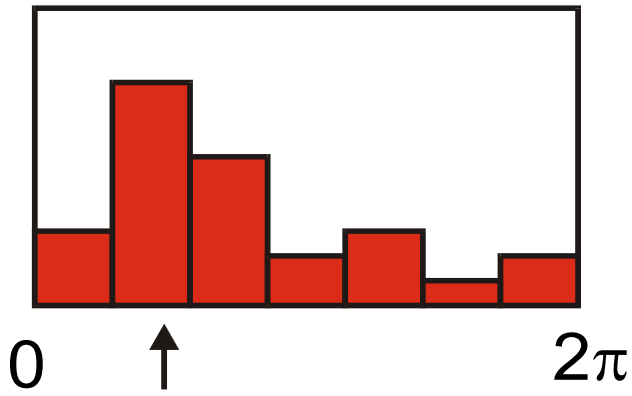
K. Grauman, B. Leibe

Orientation-invariance
SIFT detector

Finding Keypoints Orientation

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

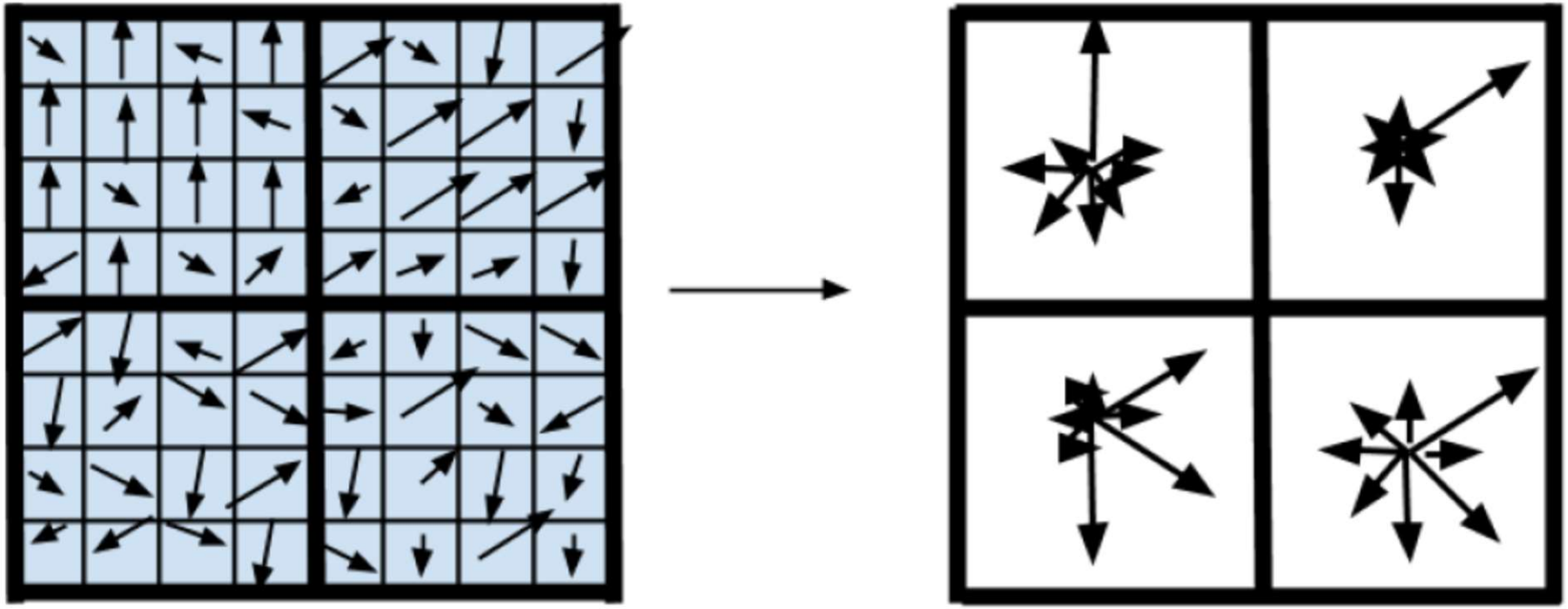




SIFT descriptor

Creating Signature

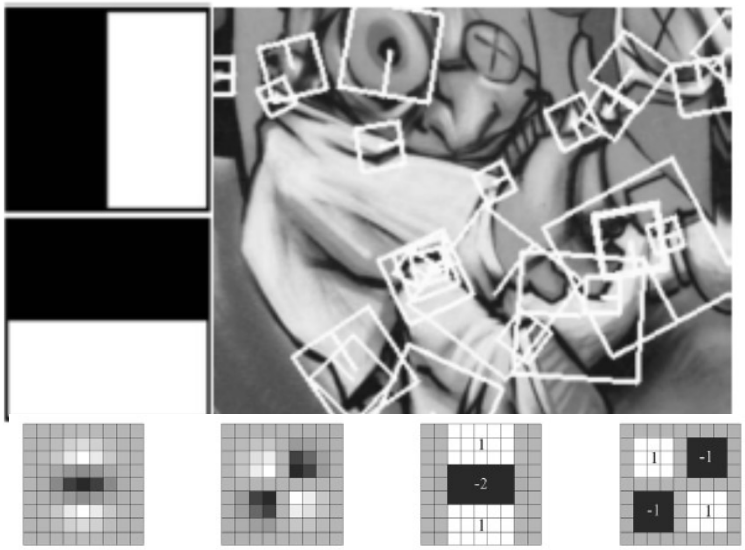
- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions



SURF

SURF

- Fast approximation of SIFT idea
- Efficient computation by 2D box filters & integral images
  6 times faster than SIFT
- Equivalent quality (?) for object identification



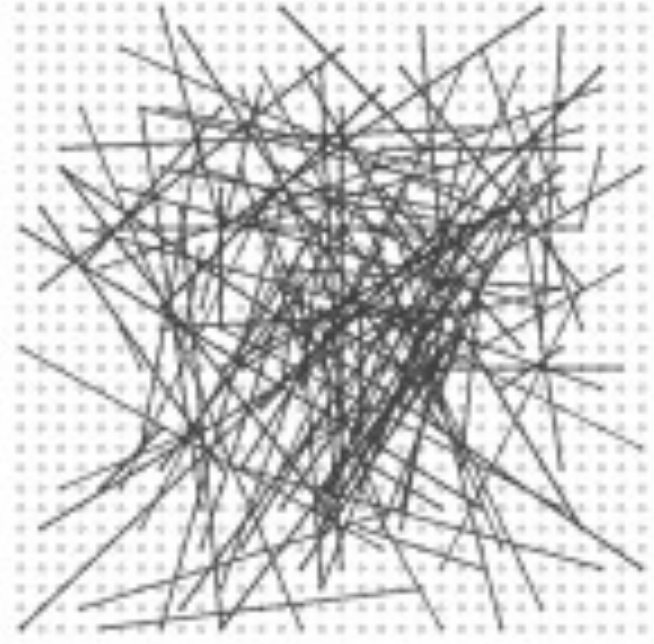
[Bay, ECCV'06], [Cornelis, CVGPU'08]

Binary local desriptors

Binary Robust Invariant Elementary Features : BRIEF

The BRIEF descriptor is constructed using a set of binary tests between pixels. around a key-point.

01000111010100010000010001010

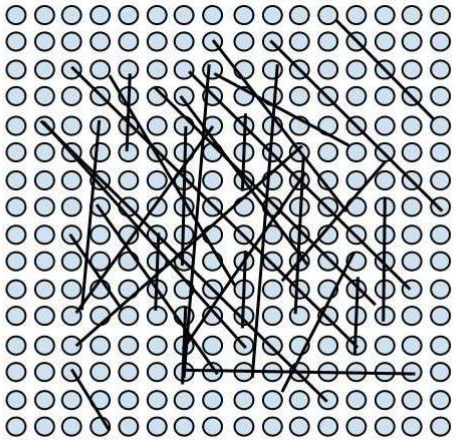


Binary descriptor - BRIEF

BRIEF a binary descriptor which describes a patch of the image, where each bit corresponds to a binary test result to the intensity difference.

We can imagine the descriptor pattern like the small sticks disseminated on the patch. Then we look at the two ends of each stick and compare their value of intensity.

The resulting descriptor length will contain
1 if:
I(start) > I(end)
0 if:
I(start) <= I(end)



Oriented BRIEF : ORB

ORB uses the descriptor modification BRIEF, -> BRIEF with rotation.

The rotation of the detected interesting point is extracted using a center of gravity of the intensity of the surrounding area.

The method assumes that the vector of the interesting point in the center of gravity (Center of gravity of intensity) induces rotation.

Binary descriptor
FREAK: Fast Retina Keypoint

- Keypoint descriptor inspired by the human visual system and more precisely the retina, coined Fast Retina Keypoint (FREAK).
- A cascade of binary strings is computed by efficiently comparing image intensities over a retinal sampling pattern.

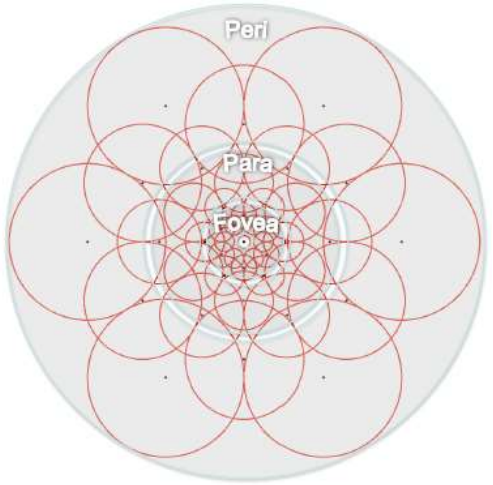


illustration of the FREAK sampling pattern similar to the retinal ganglion cells distribution with their corresponding receptive fields.

Each circle represents a receptive field where the image is smoothed with its corresponding Gaussian kernel.

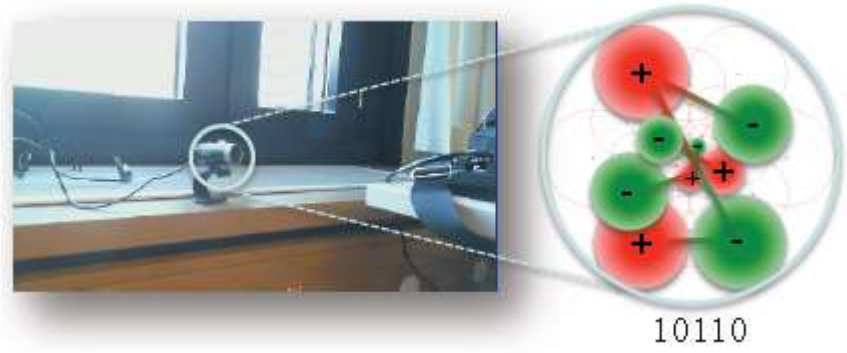
Alexandre Alahi, Raphael Ortiz, Pierre Vandergheynst

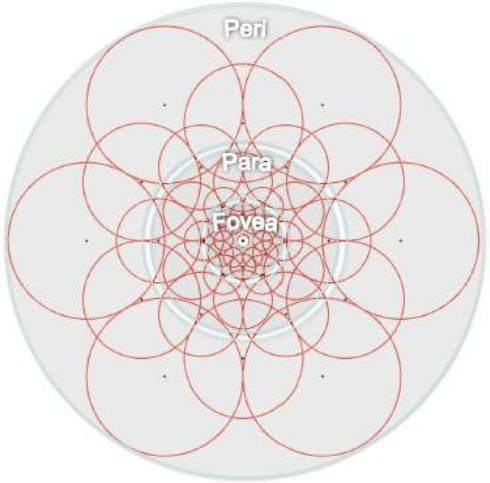
Binary descriptor
FREAK: Fast Retina Keypoint

Smoothing by its corresponding Gaussian kernel

A series of Difference of Gaussians (DoG) over a retinal pattern are 1 bit quantized

Ilustration of our FREAK descriptor.





Alexandre Alahi, Raphael Ortiz, Pierre Vandergheynst

Regions Local Detector

MSER Maximally Stable Extremal Regions

MSER regions are connected areas characterized by almost uniform intensity, surrounded by contrasting background.

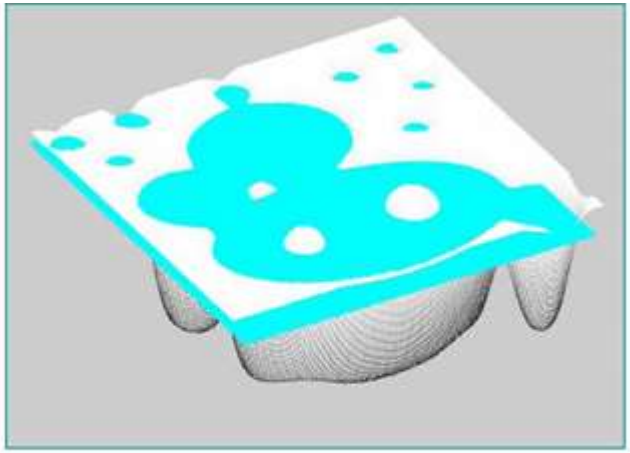
They are constructed through a process of trying multiple thresholds.

The selected regions are those that maintain unchanged shapes over a large set of thresholds.

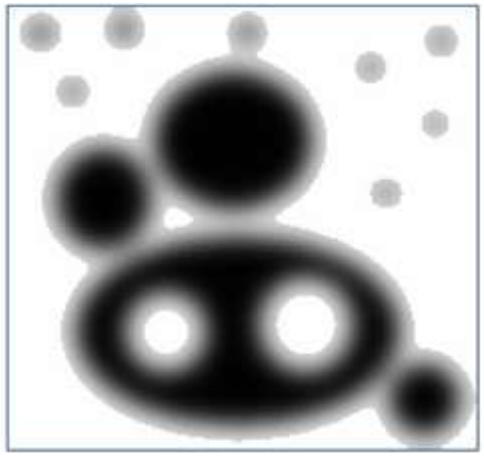
Matas, J., O. Chum, M. Urba, and T. Pajdla. "Robust wide baseline stereo from maximally stable extremal regions." *Proceedings of British Machine Vision Conference*, pages 384-396, 2002.

Jiri Matas – key speaker of IITSRC 2014!

MSER Construction



Threshold simulation



Extremal Regions (represented by their original lumiance values)

For each region, and for each threshold value, the region area is saved.

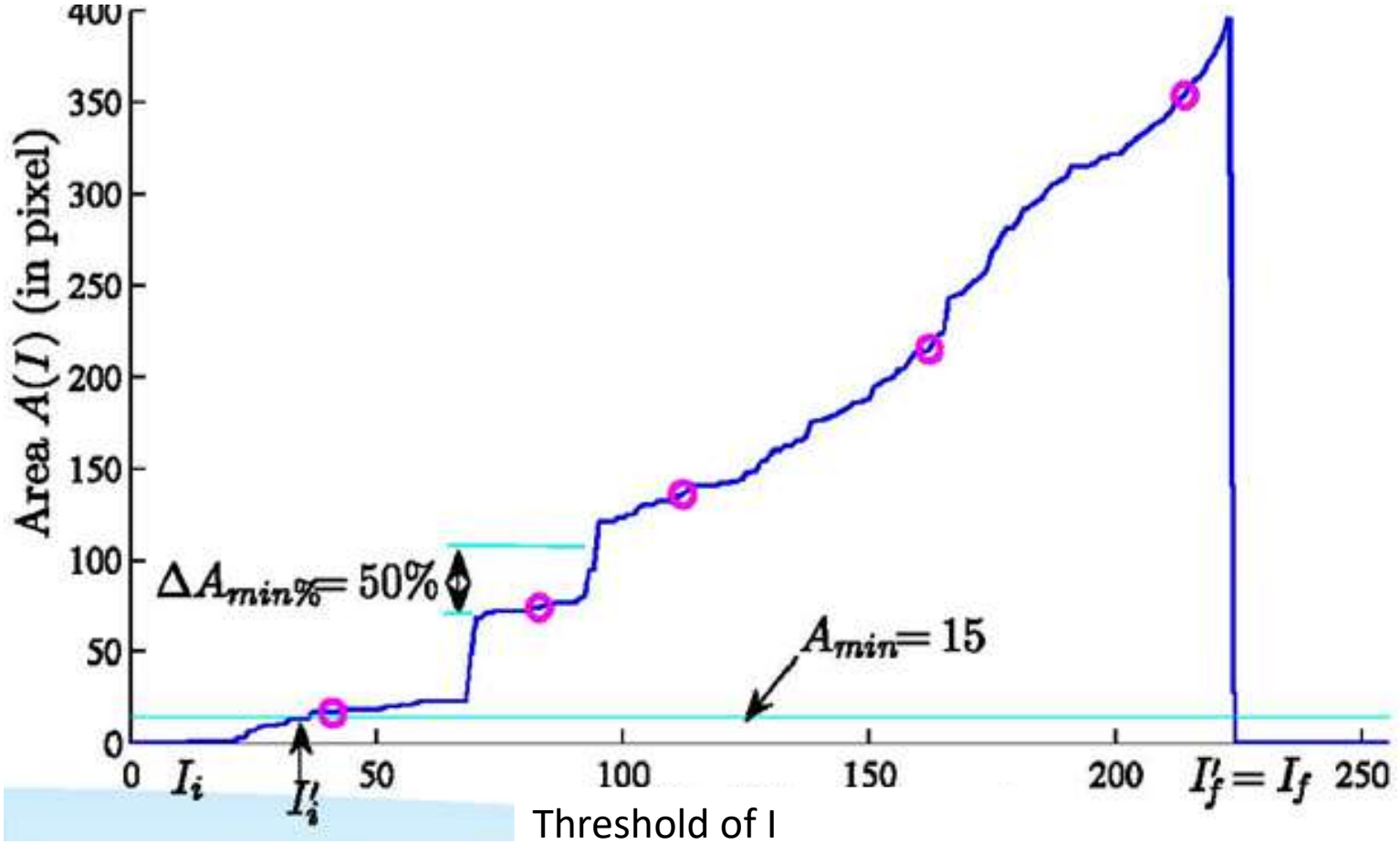
MSER Computation

For each threshold, compute the connected binary regions.

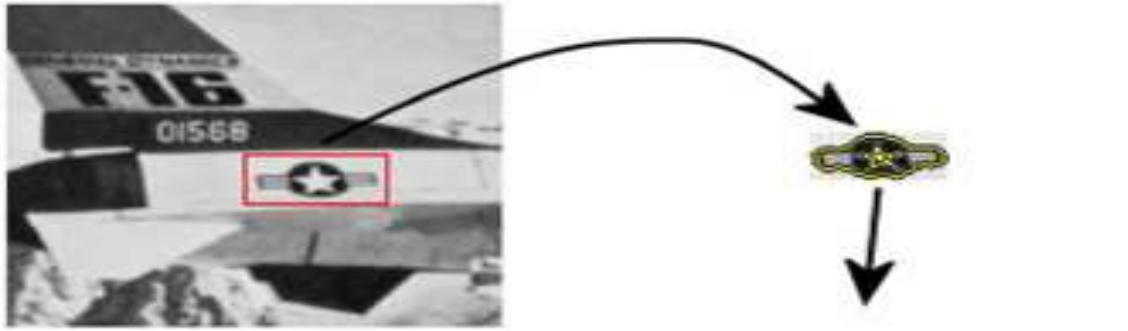
Compute a function, such as area A(i), at each threshold value i.

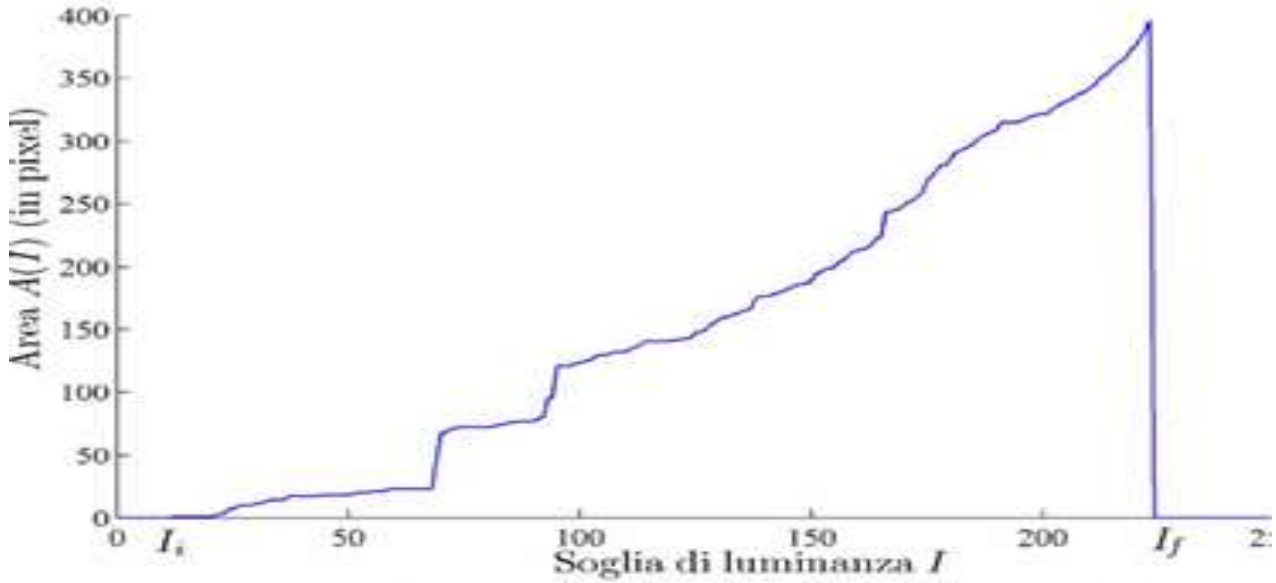
Analyze this function for each potential region to determine those that persist with similar function value over multiple thresholds.

Analysis of Area Function

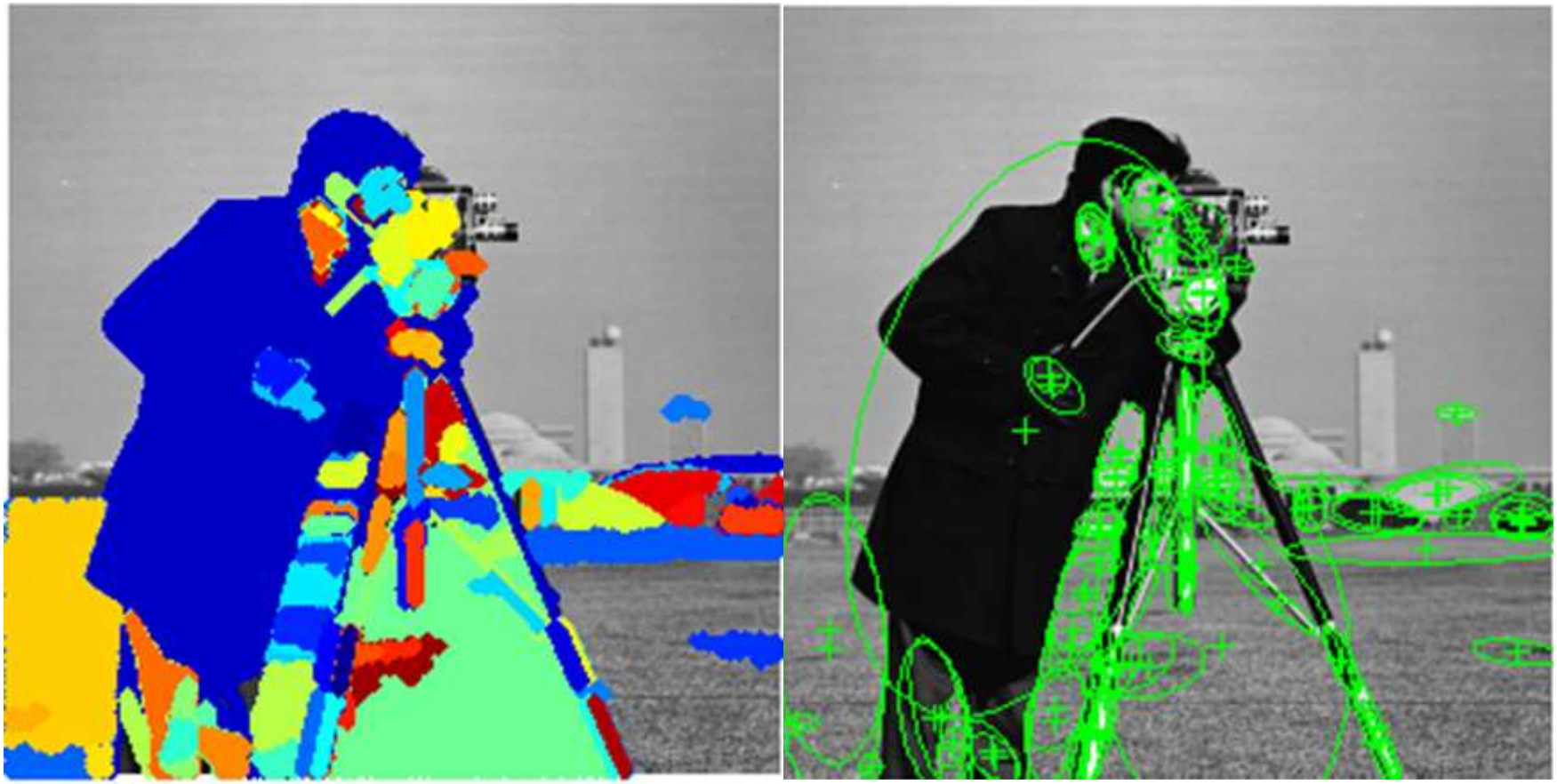


Regions detected at different thresholds have different areas





MSER - example

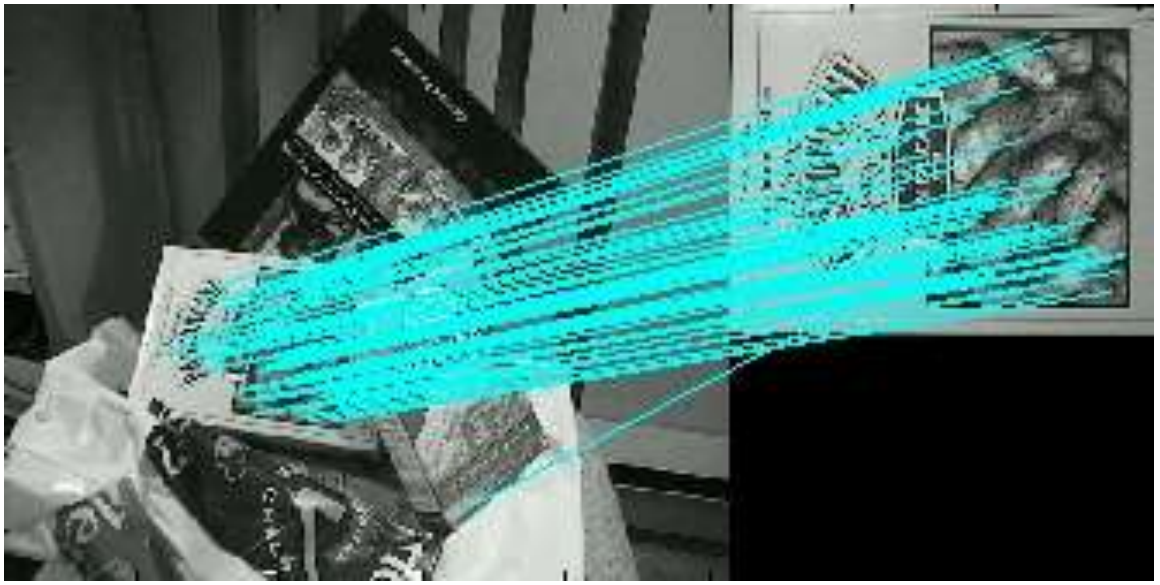


Example - Matlab

Matching of local features

Matching Local Features

Euclidean , Hamming… distance to the nearest neighbour
similarity search



Direct Method (brute force)

Direct Method (brute force)

Need to define image distance function:

- Search over all feature pairs

calculate Dist(F1,F2)

Select minimum F1, F2 with minimum distance

-> corresponding features

Distance function

Float features: Euclideans distance

Binary featurec: Hamming distance

The Hamming Distance is a number used to denote the difference between two binary strings. (XOR)

RANSAC approach
Filtering inliers/outliers

1. Randomly select enough matches points to homography
2. Compute homography
3. Using that homography, measure error on inliers/outliers (min 50%inliers)
4. Output best one found

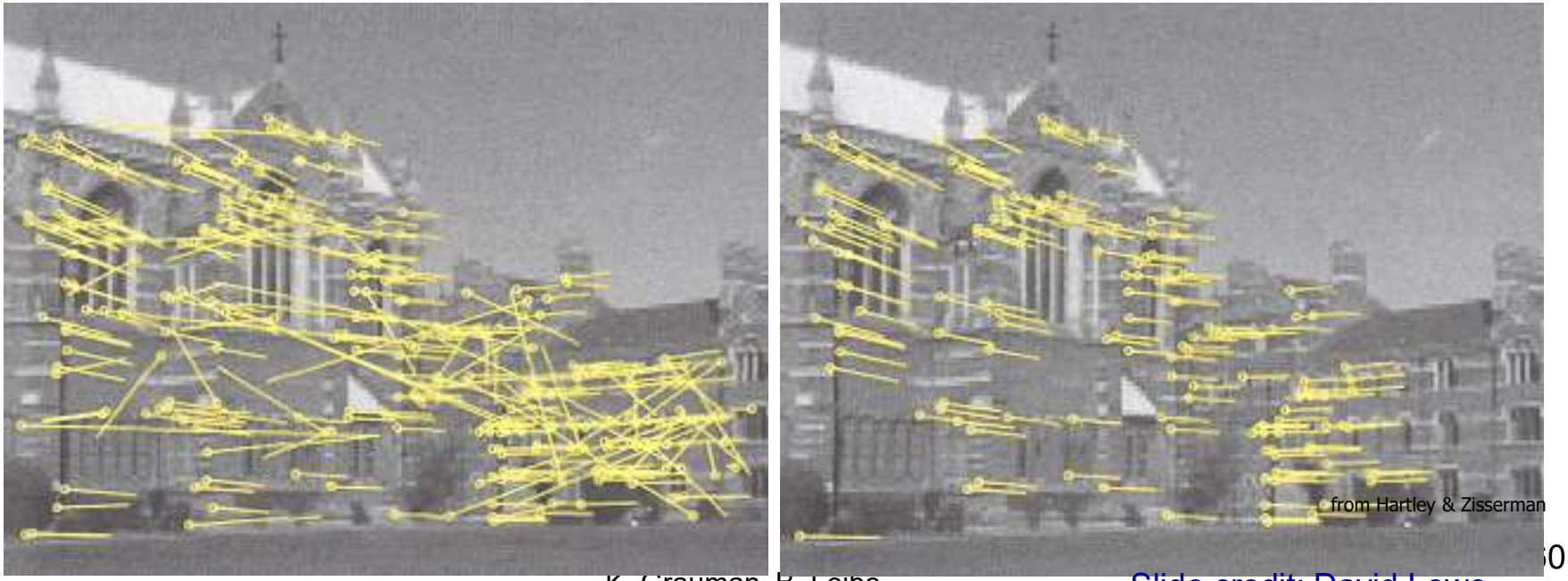
–RANSAC [Obdrzalek02, Chum05, Nister06]

Example: Finding Feature Matches

Find best stereo match within a square search window (here 300 pixels2)

Global transformation model: epipolar geometry

before RANSAC **after RANSAC**



from Hartley & Zisserman

Slide credit: David Lowe

RANSAC (RANdom Sample Consensus)

Iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers

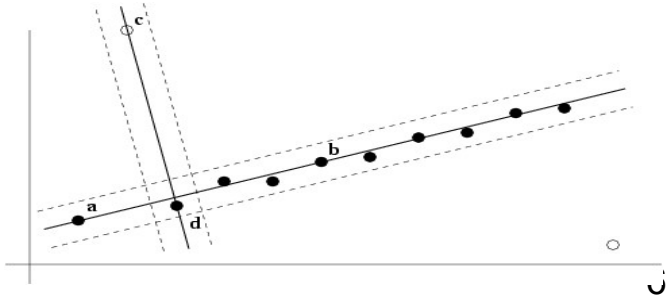
Randomly choose a minimal subset of data points necessary to fit a model (a sample)

Points within some distance threshold t of model are a consensus set. Size of consensus set is model's support.

Repeat for N samples; model with biggest support is most robust fit

Points within distance t of best model are inliers

Fit final model to all inliers

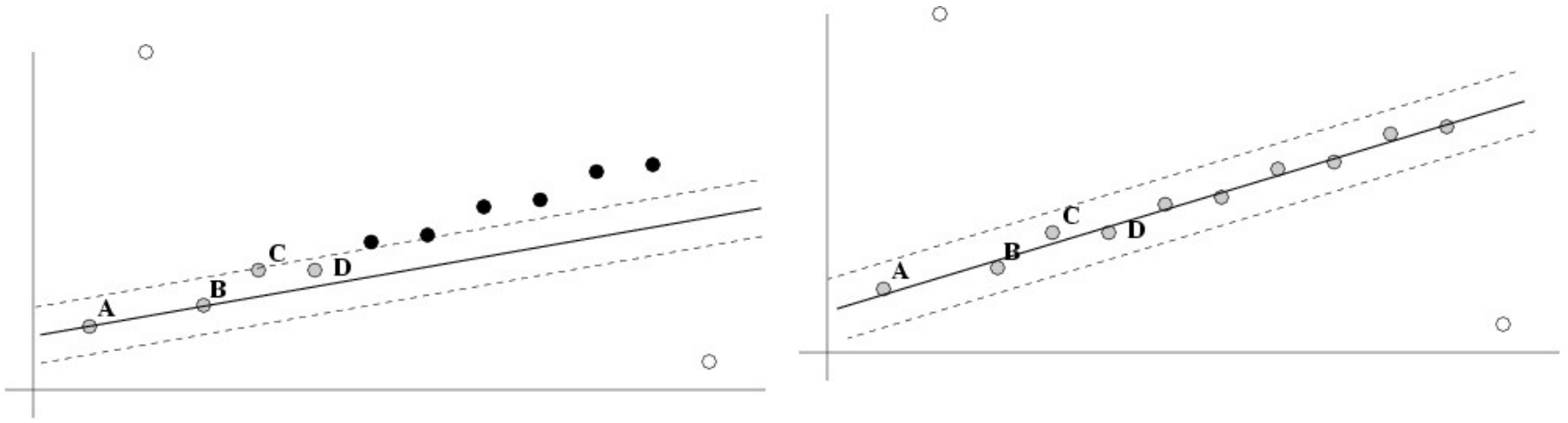


RANSAC (RANdom Sample Consensus)

RANSAC divides data into inliers and outliers and yields estimate computed from minimal set of inliers

Improve this initial estimate with estimation over all inliers (e.g. with standard least-squares minimization)

But this may change inliers, so alternate fitting with re-classification as inlier/outlier



K. Grauman, B. Leibe

Slide credit: David Lowe

Homography

- Consider a point x = (u,v,1) in one image and x′=(u′,v′,1) in another image
- A homography is a 3 by 3 matrix M

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

- The homography relates the pixel co-ordinates in the two images if x′ = M x
- When applied to every pixel the new image is a warped version of the original image

RANSAC loop (iterative method)

RANSAC loop:

- Select four feature pairs (at random)
- Compute homography H (exact)
- Compute *inliers*
- Keep largest set of inliers
- Re-compute least-squares H estimate on all of the inliers

Visual words
Bag of words - BoF

Indexing local features - Visual words

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

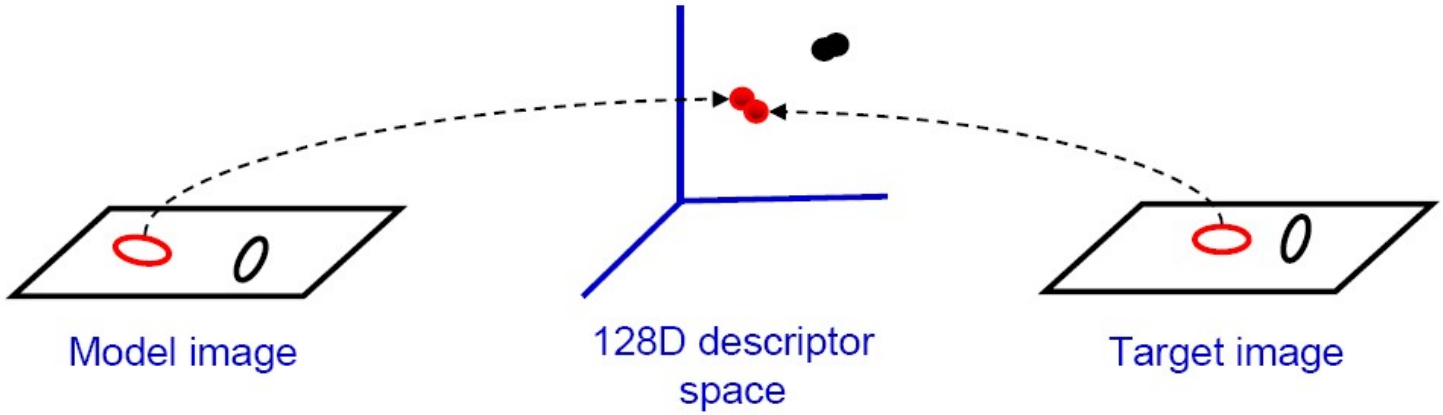
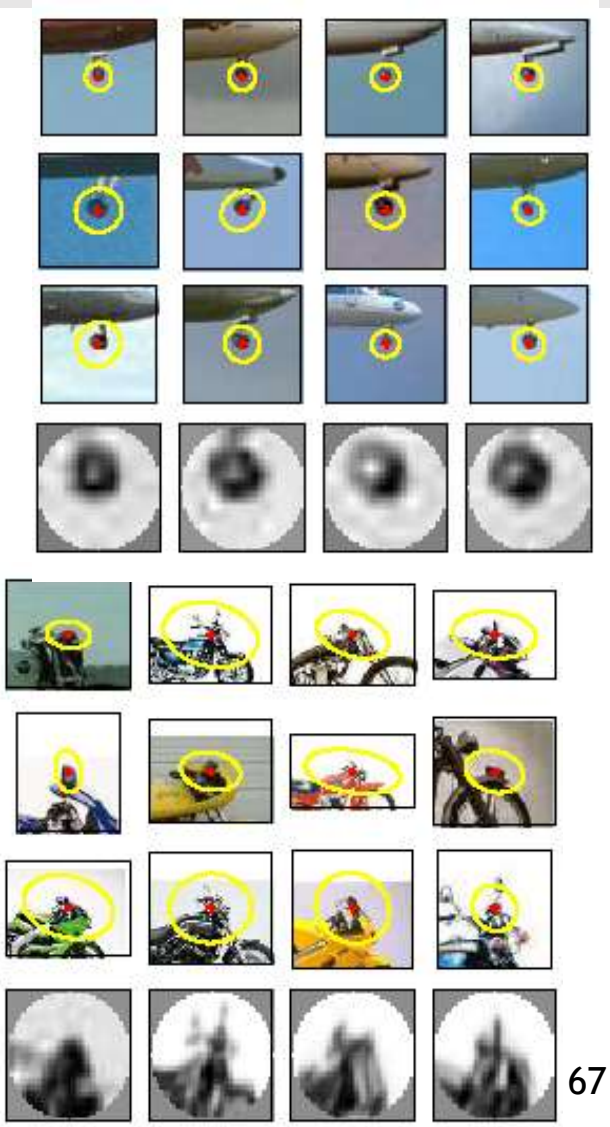


Figure credit: A. Zisserman

K. Grauman, B. Leibe

Visual words - main idea

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

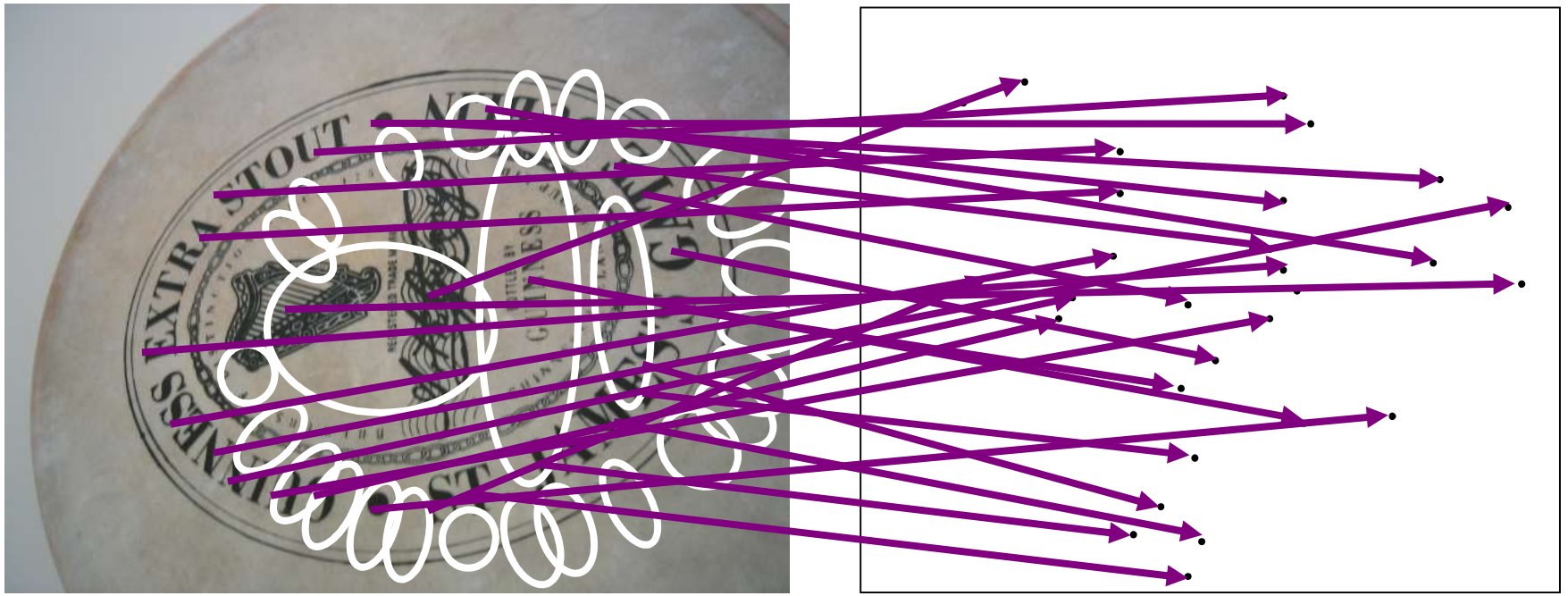


Sivic & Zisserman 2003; Csurka, Bray, Dance, & Fan 2004; many others.

K. Grauman, B. Leibe

Visual words: main idea

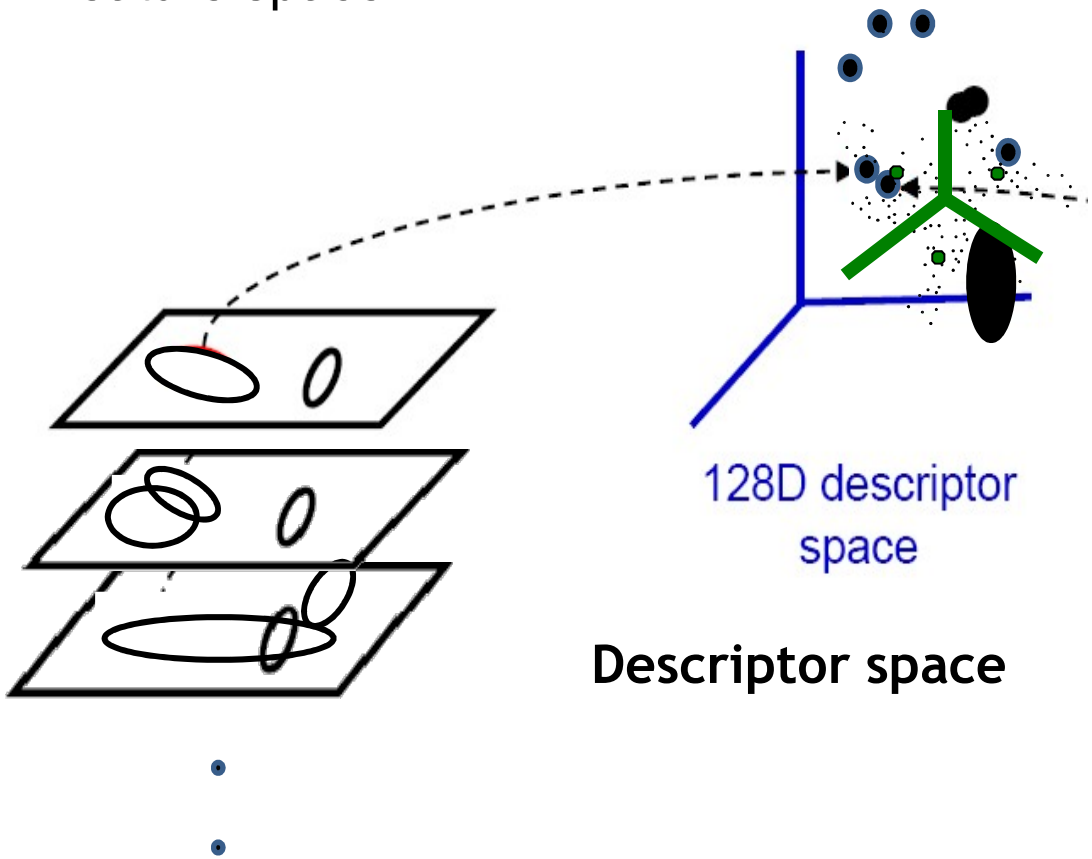
Extract local features from a number of images …



e.g., SIFT descriptor space: each point is 128-dimensional

Visual words: quantization of feature space

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



Descriptor space

- Quantize via clustering, let cluster centers be the prototype “words”
- Determine which word to assign to each new image region by finding the closest cluster center.

Visual words - Example

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

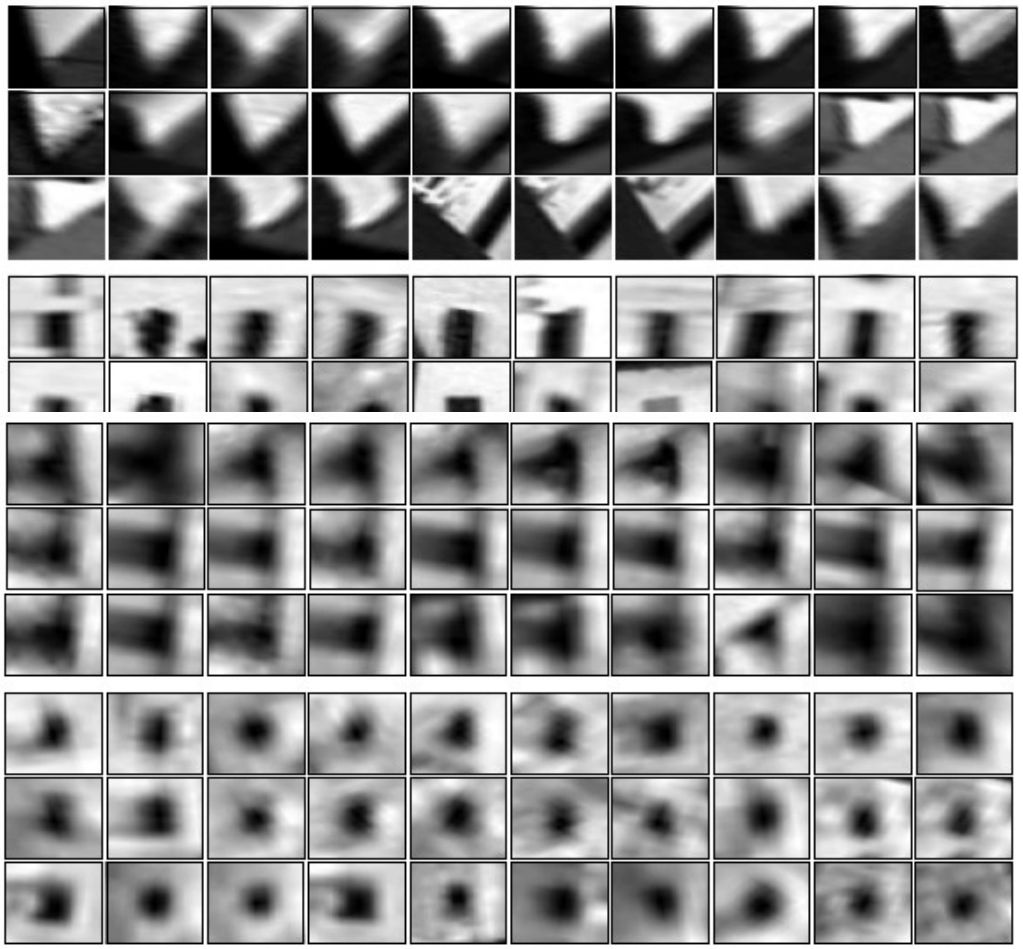


Figure from Sivic & Zisserman, ICCV 2003

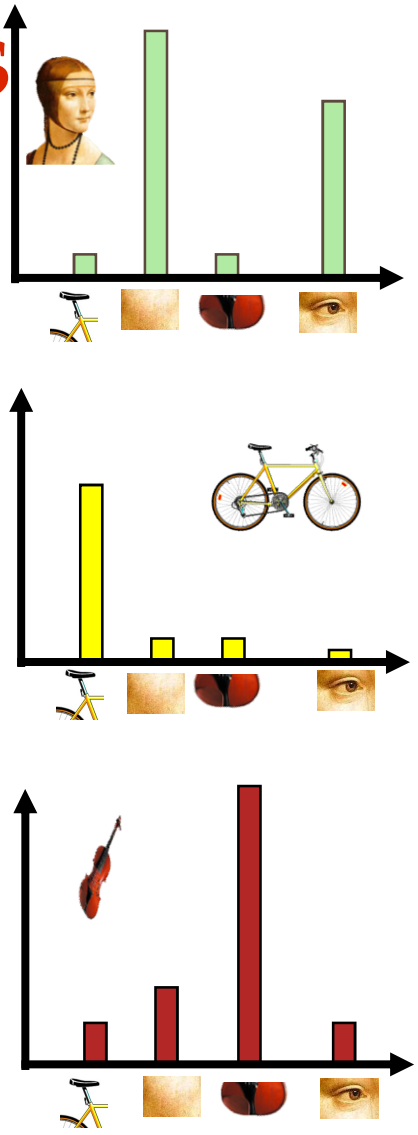
Visual Vocabularies and Bags of Words

Histogram representation of the image

- inspiration from the text retrieval community:

text words are discrete „tokens“, whereas local image descriptors are high-dimensional, real-valued feature points.

Bags of visual words



Summarize entire image based on its distribution (histogram) of word occurrences.

Analogous to bag of words representation commonly used for documents.

Visual vocabulary formation

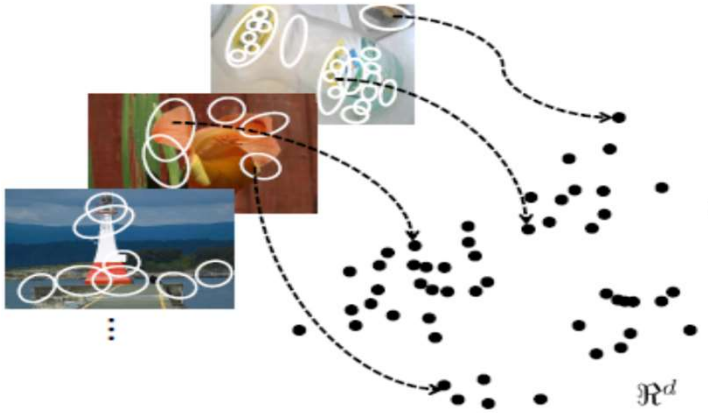
Clustering / quantization algorithm

Unsupervised vs. supervised

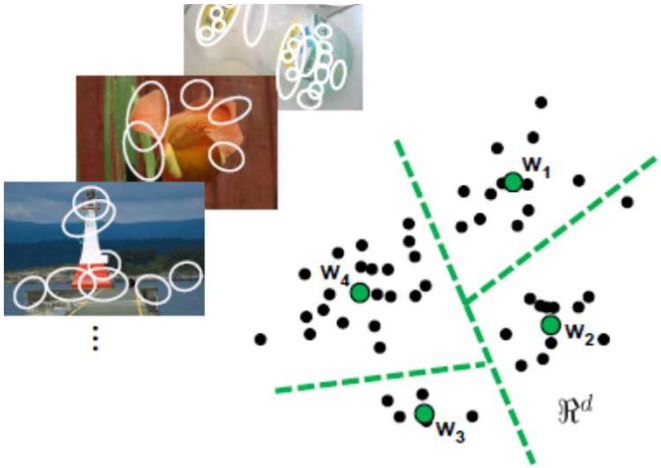
What corpus provides features (universal vocabulary?)

Vocabulary size, number of words

Visual Vocabularies and Bags of Words



(a)



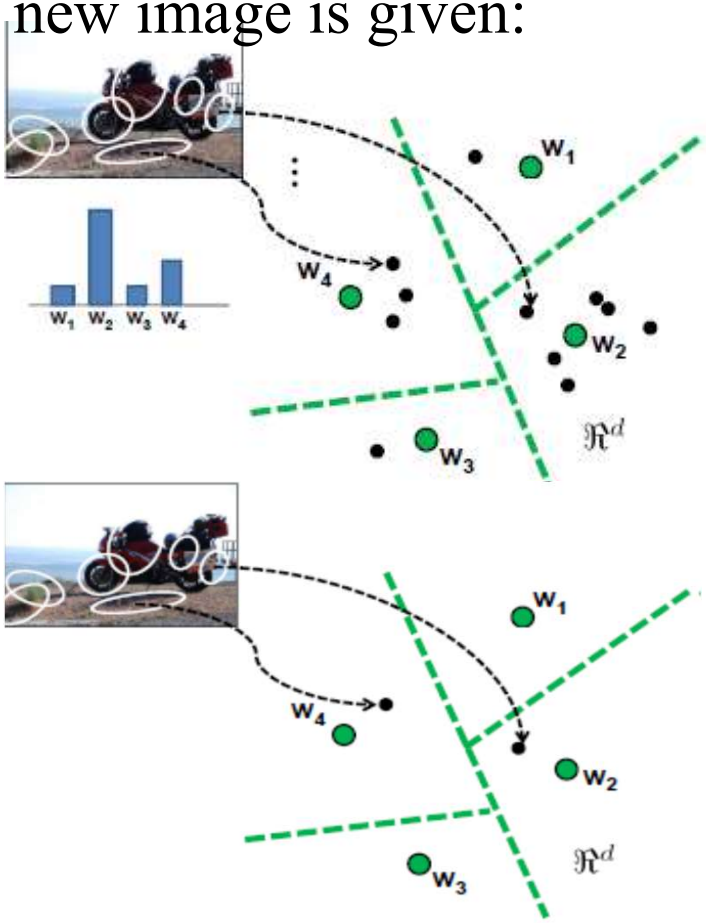
(b)

A large corpus of representative images are used to populate the feature space with descriptor instances. The white ellipses denote local feature regions, and the black dots denote points in some feature space, *e.g.*, SIFT.

The sampled features are clustered (k-means) in order to quantize the space into a discrete number of visual words. The visual words are the cluster centers, denoted with the large green circles.

Visual Vocabularies and Bags of Words

A new image is given:



(c)

The nearest visual word is identified for each of its features. This maps the image from a set of high-dimensional descriptors to a list of word numbers.

A bag-of-visual-words histogram can be used to summarize the entire image.

It counts how many times each of the visual words occurs in the image.

+- of Bags of Words

view point invariance and scale invariance (?)

disadvantages of BoW is that it ignores the spatial relationships

Data:

Object Recognition Data from Oxford VGG:

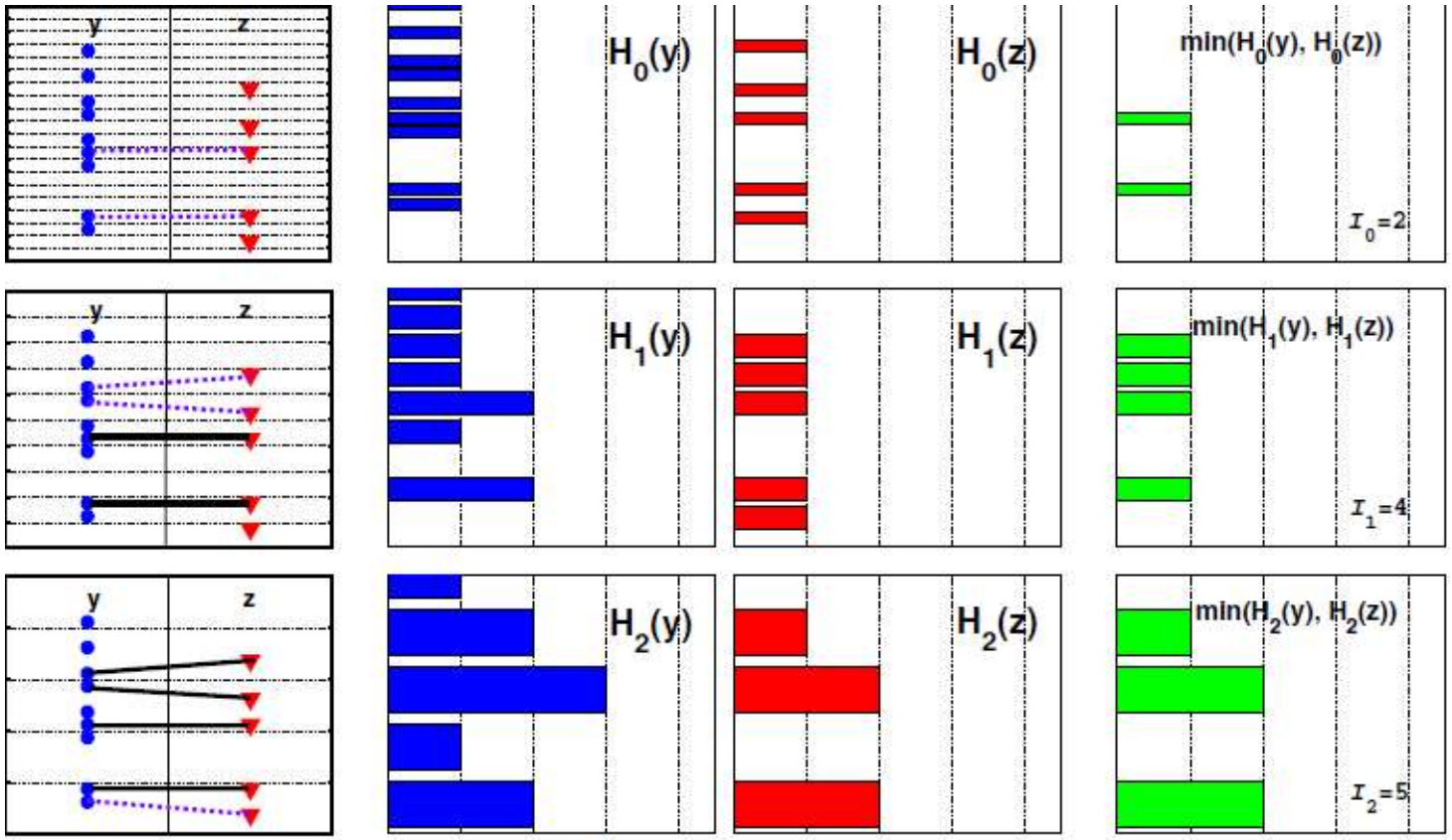
http://www.robots.ox.ac.uk/~vgg/data/

Pascal2: http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/

Pyramid match kernel -Measure of similarity

This „pyramid match" maps unordered feature sets to multi-resolution histograms and computes a weighted histogram intersection in this space.

Pyramid match kernel -Measure of similarity



(a) Point sets (b) Histogram pyramids (c) Intersections

Indexing local features
Inverted file indexing schemes

Indexing local features - main idea

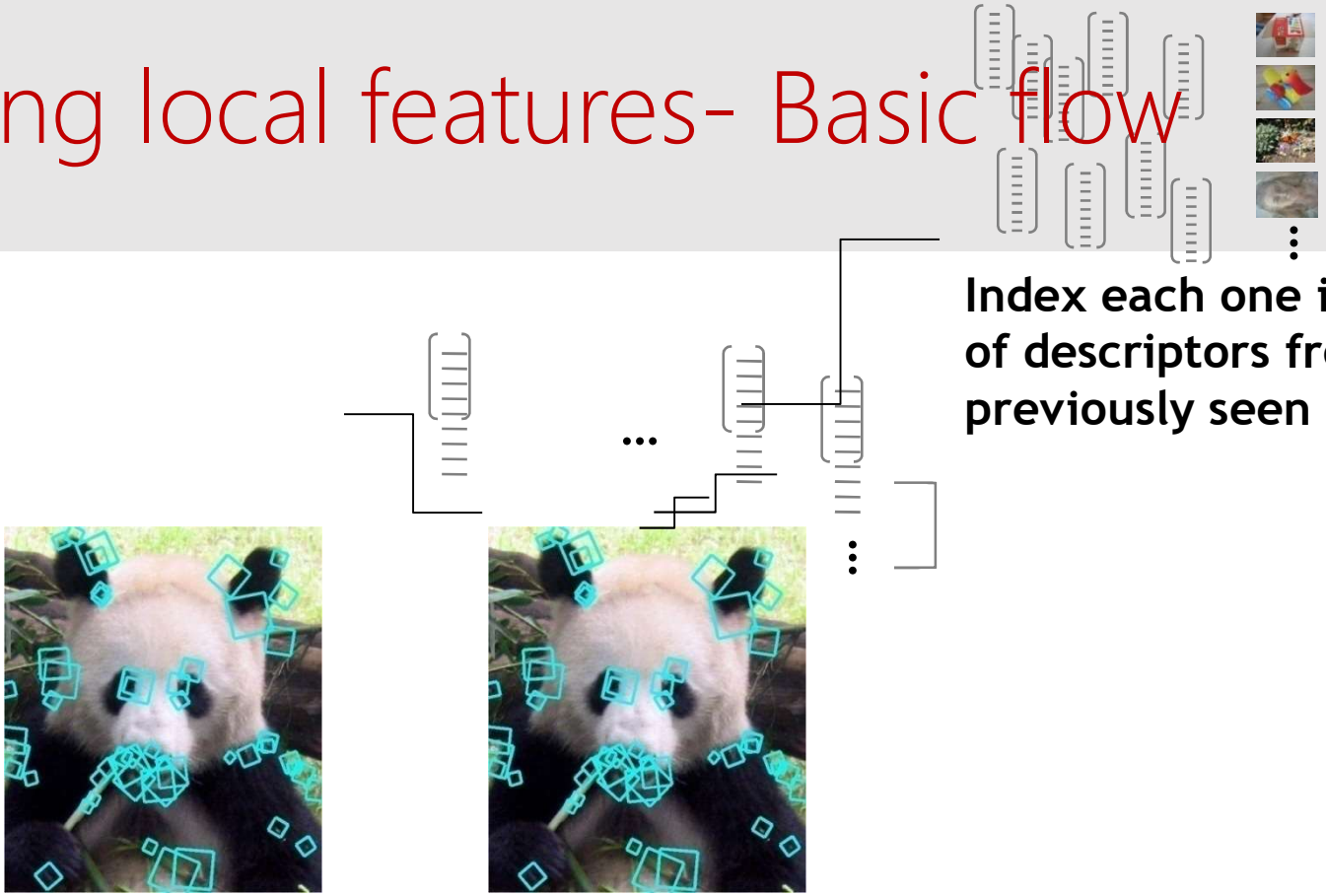
With potentially thousands of features per image, and hundreds to millions of images to search, how to efficiently find those that are relevant to a new image?

Low-dimensional descriptors : can use standard efficient data structures for nearest neighbour search

High-dimensional descriptors: approximate nearest neighbour search methods more practical

Inverted file indexing schemes

Indexing local features- Basic flow



K. Grauman, B. Leibe

Indexing local features: inverted file index

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".

Inverted file index for images comprised of visual words

frame #5

frame #10

List of image numbers

Word number

Posting list

1 → 5,10, …

2 → 10,…

… …

Image credit: A. Zisserman K. Grauman, B. Leibe

Applications of local features

Applications of local features

object recognition

panoramas

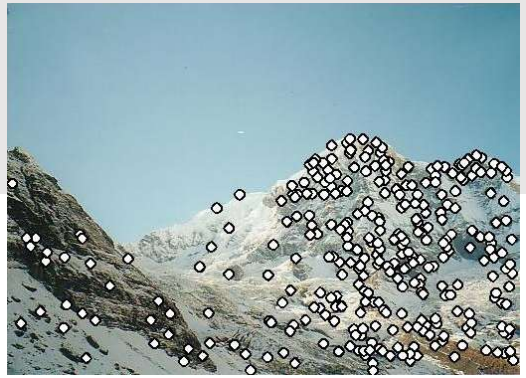
stereo reconstruction

custom camera calibration

object motion tracking

image retrieval

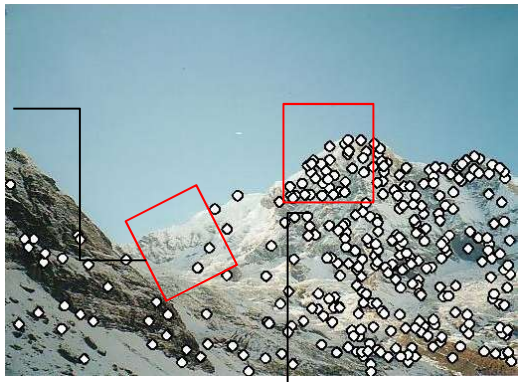
Local features - panorama



Detection: Identify the interest points

$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$

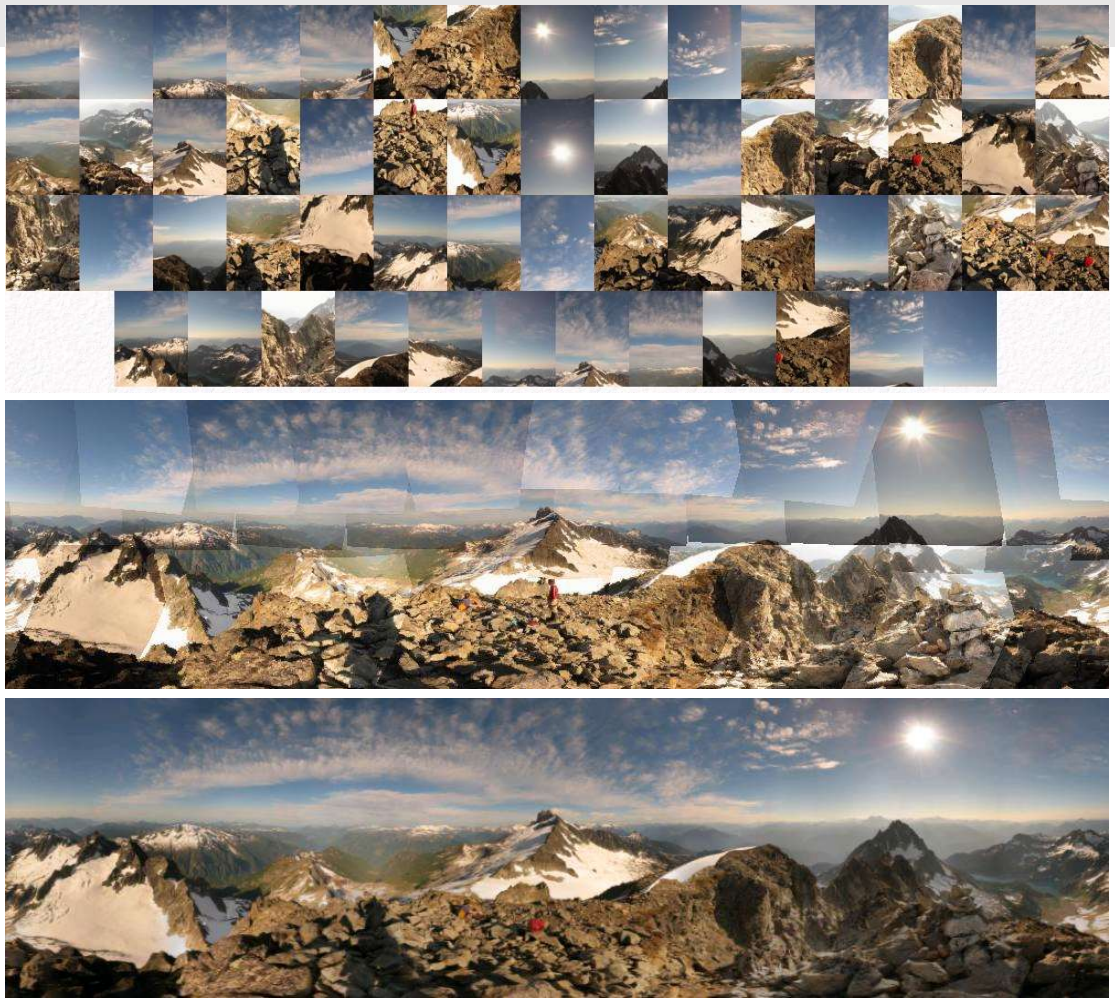
Description:Extract vector feature descriptor surrounding each interest point.



$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

Matching: Determine correspondence between descriptors in two views

Automatic mosaicing



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

Wide baseline stereo

[Image from T. Tuytelaars ECCV 2006 tutorial]

Recognition of specific objects, scenes

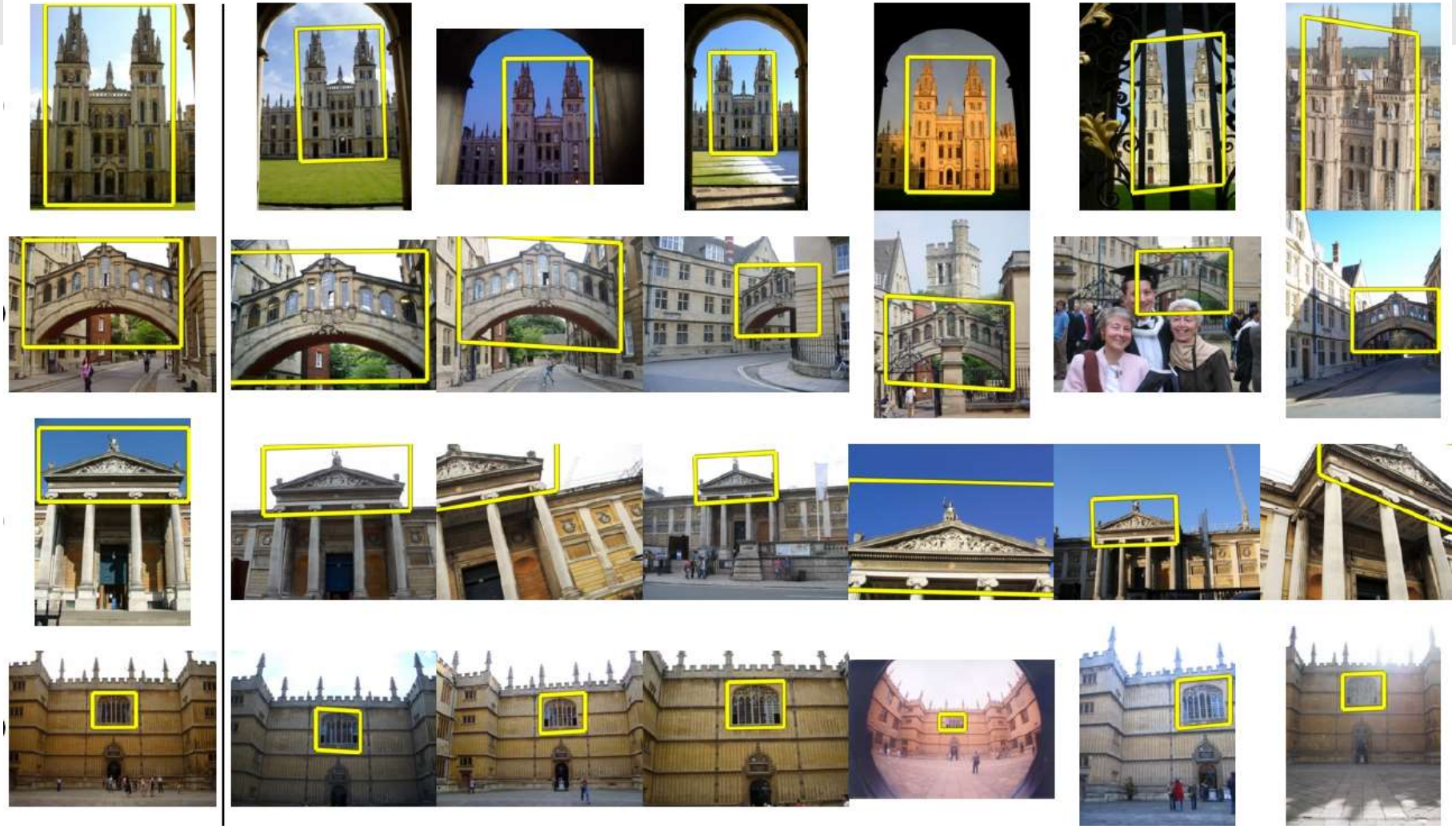
Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

Lowe 2002

Application: Large-Scale image retrieval



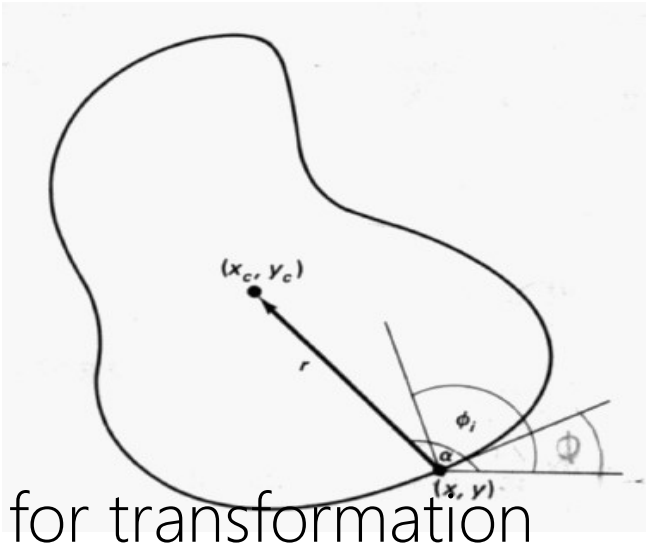
Query Results from 5k Flickr images (demo available for 100k set)

[Philbin CVPR'07]

Generalized Hough Transform

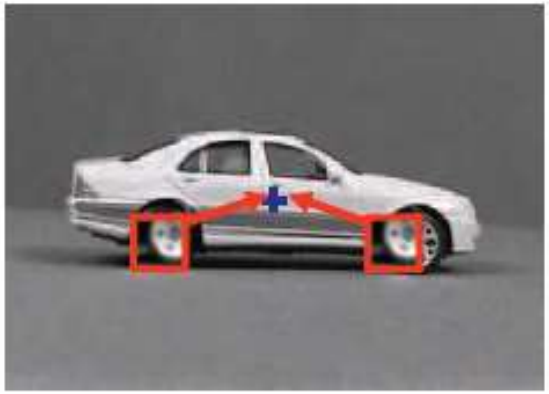
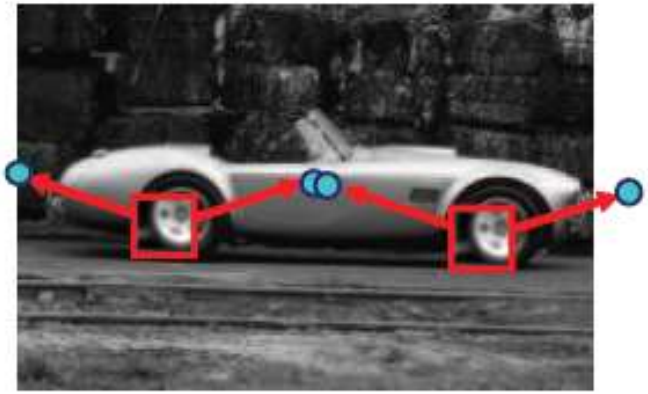
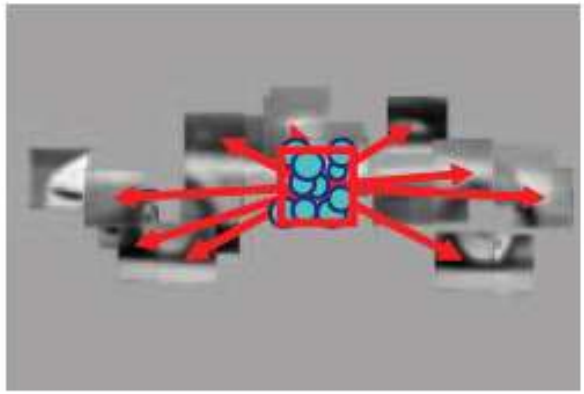
Generalized Hough Transform

- Generalization for an arbitrary contour or shape
- Choose reference point for the contour (e.g. center)
- For each point on the contour remember where it is located w.r.t. to the reference point
- Remember radius r and angle relative to the contour tangent
- Recognition: whenever you find a contour point, calculate the tangent angle and 'vote' for all possible reference points



- Instead of reference point, can also vote for transformation
- The same idea can be used with local features!

Generalized Hough Transform

Visualization of IMPLICIT SHAPE MODEL. (left)

During training, we learn the spatial occurrence distribution of each visual word relative to the object center. (middle)

For recognition, we use those learned occurrence distributions in order to cast probabilistic votes for the object center in an extension of the Generalized Hough Transform. (right)

Once a maximum in the voting space has been found, we can backproject the contributing votes in order to get the hypothesis's support in the image.