



Local Feature Extraction and Description for

Wide-Baseline Matching, Object Recognition and
Image Retrieval Methods, Stitching and more ...

Jiří Matas and Ondra Chum

Center for Machine Perception, Czech Technical University Prague

Includes slides by:

- Darya Frolova, Denis Simakov, The Weizmann Institute of Science
- Martin Urban, Stepan Obdrzalek, Ondra Chum Center for Machine Perception Prague
- Matthew Brown, David Lowe, University of British Columbia

The Correspondence Problem

Establishing correspondence is the key issue in many computer vision problems:

- Object recognition and Image retrieval
- Wide baseline matching
- Detection and localisation
- 3D Reconstruction
- Image Stitching
- Tracking



- Methods based on “Local Features” are the state-of-the-art for number of computer vision problems (often those. that require local correspondences).
- E.g.: Wide-baseline stereo, object recognition and image retrieval.
- Terminology is a mess:
Local Feature = Interest “Point” = The “Patch” =
= Feature “Point”
= Distinguished Region
= (Transformation) Covariant Region

Image Stitching: Building a Panorama:

Example of a Method Based on Local Features



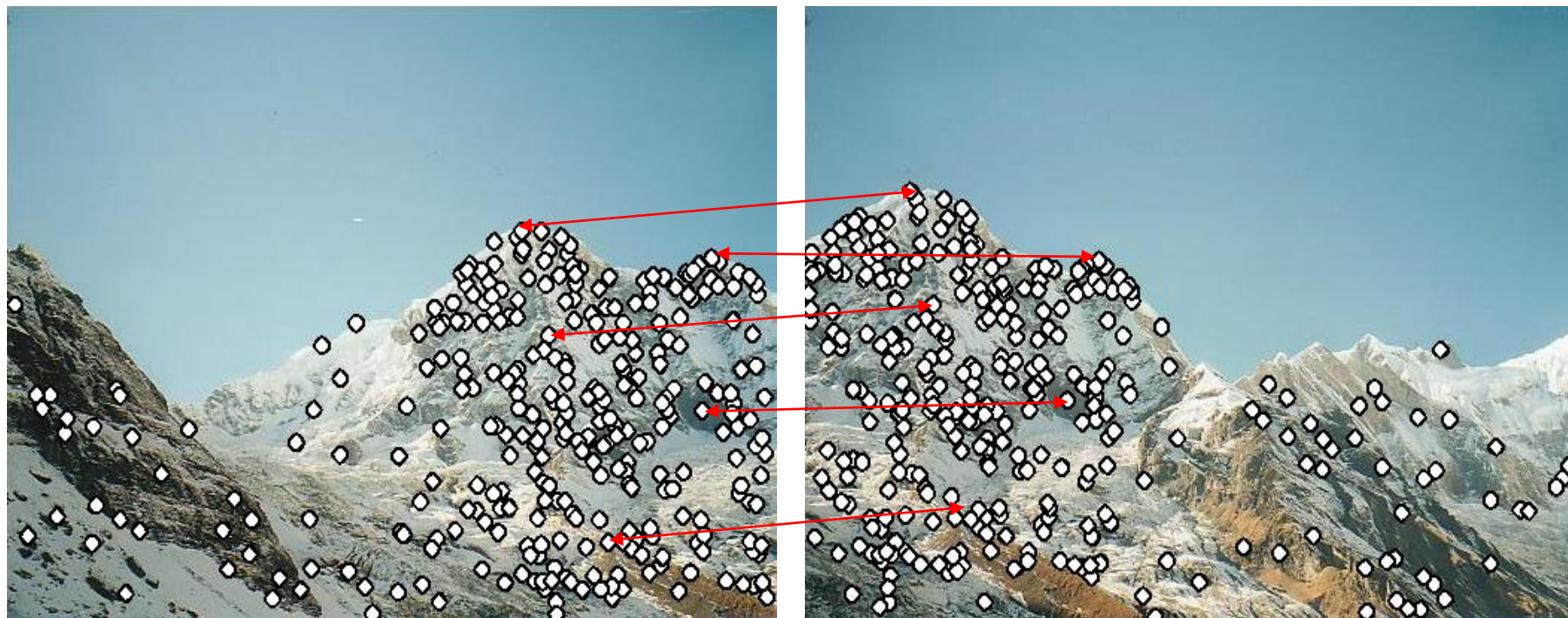
How do we build a panorama?

- We need to match (align) images = find (dense) correspondence
- (technically, this can be done only if both images taken from the same viewpoint)



Possible Approach: Matching Features

1. Detect feature points in both images
2. Find corresponding pairs
3. Estimate transformations (Geometry and Photometry)
4. Put all images into one frame, blend.



Matching with Features

■ Problem 1:

- Detect the *same* point *independently* in both images*
- Note that the set of “points” is rather sparse



no chance to match!

A repeatable detector needed.

* does it have to be independent

Matching with Features

■ Problem 2:

- how to correctly recognize the corresponding points?



Solution:

1. Find a discriminative and stable descriptor
2. Solve the matching problem

Matching with Features

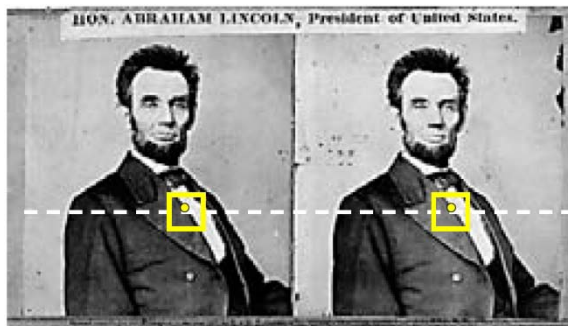
- Detect feature points in both images
- Find corresponding pairs
- Use these pairs to align images

Any alternatives?



Perhaps “Feature Points” not needed? Classical Stereo.

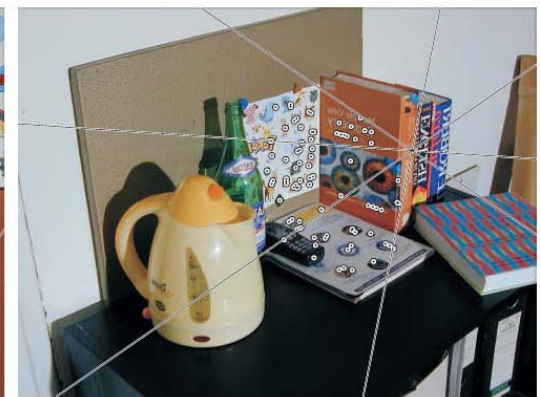
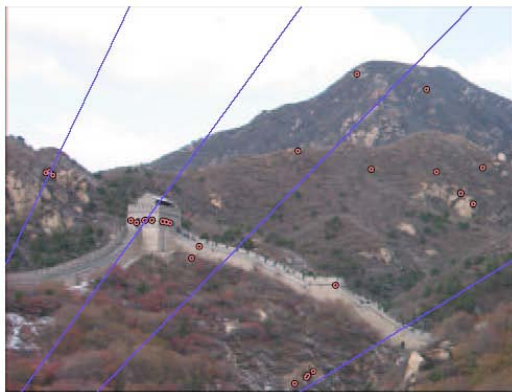
1. Local Feature (Region) = a rectangular “window”
 - robust to occlusion, translation invariant
 - windows matched by correlation, assuming small displacement
2. Local Feature (Region) = a circle around an “interest point”
 - translation and rotation invariant, robust to occlusion
 - matching based on correlation or rotation invariants (*note that the set of circles of a fixed radius is closed under translation and rotation*).
 - successful in tracking and stereo matching



Hard Impossible for a Local feature based method?

3. Widening of baseline or zooming in/out

- local deformation is well modelled by affine or similarity transformations
- How can the “interest point” concept be generalised? *The set of ellipses is closed under affine tr., but its too big to be tested ..*
- Window scanning approach becomes computationally difficult.



(Specific Object) Recognition:



Pose space search v.
Correspondence (matching)
problem?

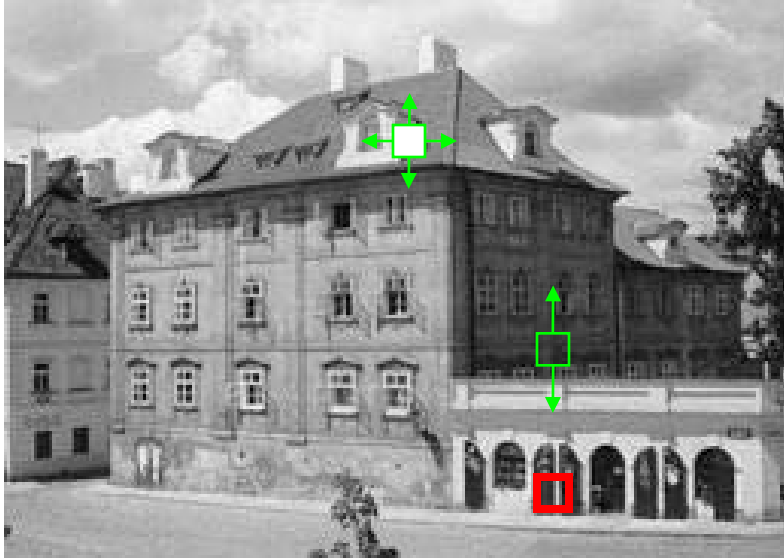
The pose space is high dimensional, but translation, scale (in a pyramid), rotation (a discrete set of angles) can be handled already ...in combination with sequential techniques.



Local Invariant Features

- “Local Features” are **regions**, i.e. in principle arbitrary sets of pixels (not necessarily contiguous) with
- High **repeatability**, (invariance in theory) under
 - Illumination changes
 - Changes of viewpoint \Rightarrow geometric transformationsi.e. are **distinguishable** in an image regardless of viewpoint/illumination \Rightarrow are **distinguished regions**
- Are **robust to occlusion** \Rightarrow must be **local**
- Must have discriminative neighborhood \Rightarrow they are “**features**”

Methods based on local features/distinguished regions (DRs) formulate computer vision problems as matching of some representation derived from DR (as opposed to matching of images)



undistinguished patches:



distinguished patches:

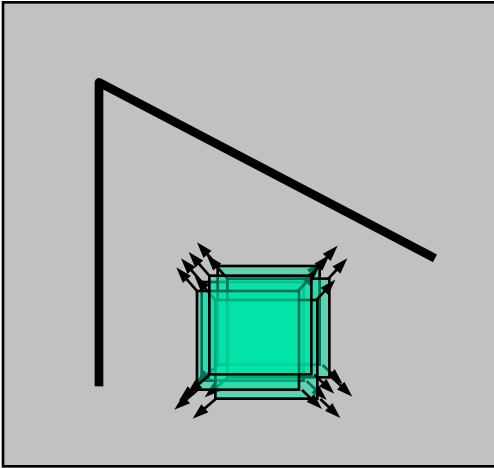


Two core ideas (in “modern terminology”):

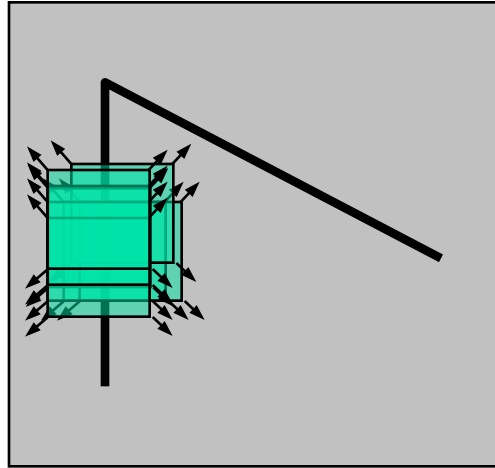
1. To be a distinguished region, a region must be *at least* distinguishable from *all* its neighbours.
2. Approximation of Property 1. can be tested very efficiently, without explicitly testing.

Note: both properties were proposed before Harris paper, (1) by Moravec, (1)+(2) by Foerstner.

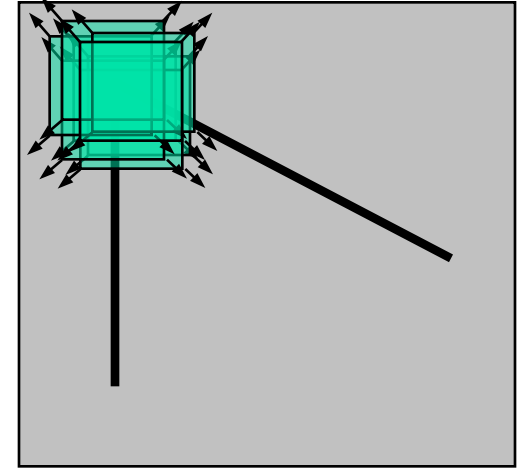
Harris Detector: Basic Idea



“flat” region:
no change in
all directions



“edge”:
no change along
the edge
direction



“corner”:
significant
change in all
directions

- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change*

Harris Detector: Mathematics

Window-averaged change of intensity for the shift $[u, v]$:

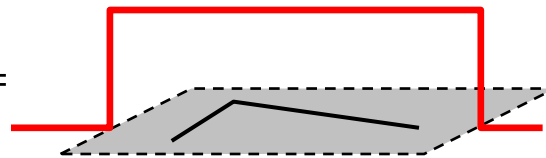
$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

Window
function

Shifted
intensity

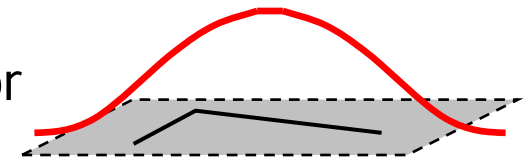
Intensity

Window function $w(x, y) =$



1 in window, 0 outside

or



Gaussian

Harris Detector: Mathematics

Expanding $E(u,v)$ in a 2nd order Taylor series expansion, we have, for small shifts $[u, v]$, a *bilinear* approximation:

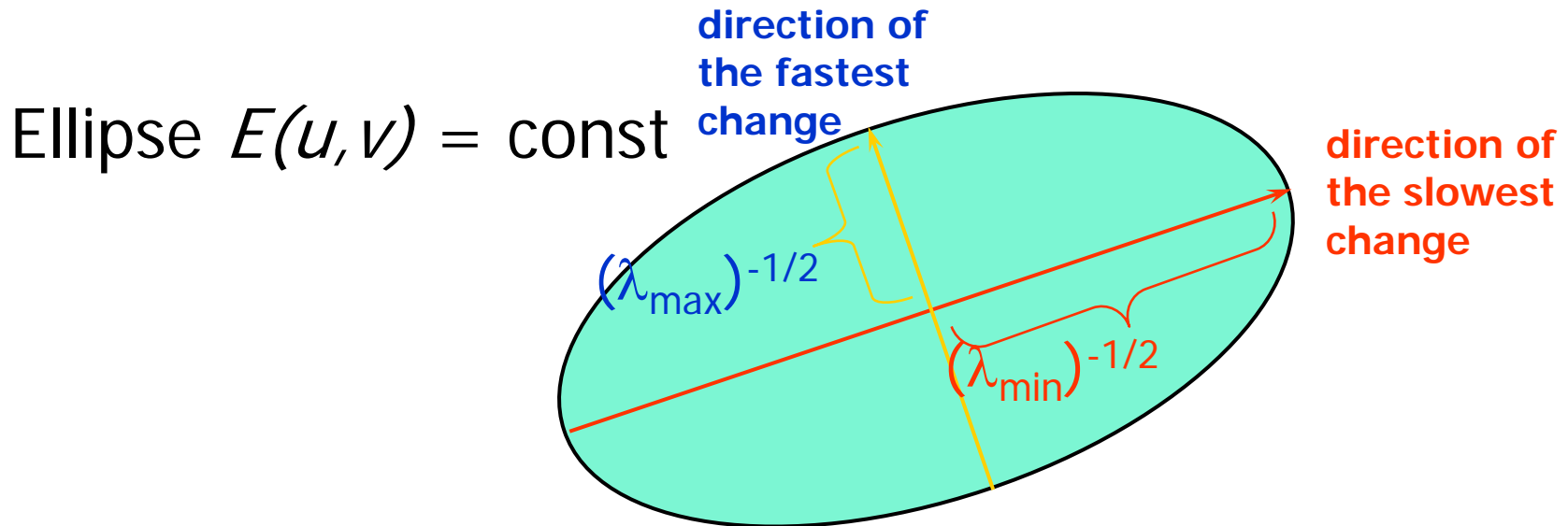
$$E(u, v) \cong [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

where M is a 2×2 matrix computed from image derivatives:

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

Intensity change in shifting window: eigenvalue analysis

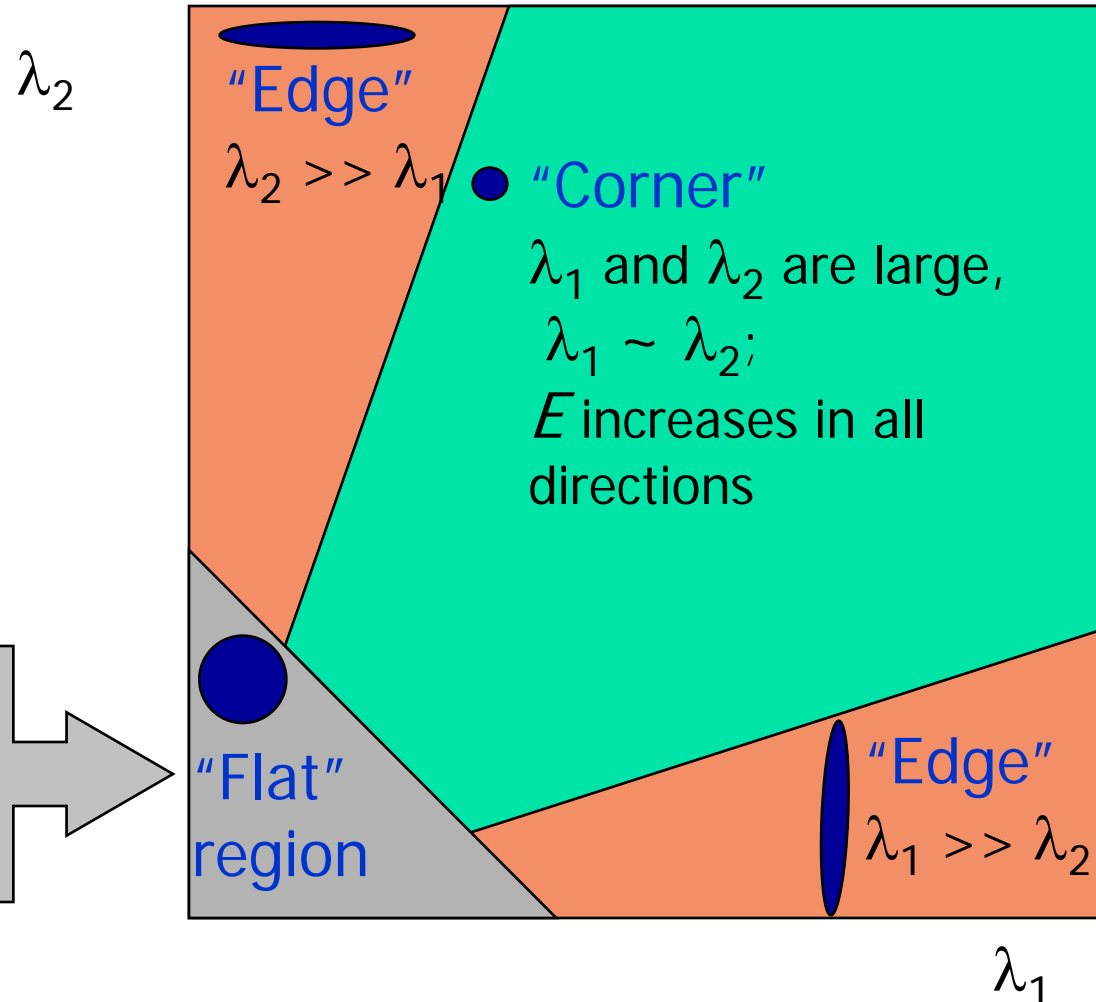
$$E(u, v) \cong [u, v] M \begin{bmatrix} u \\ v \end{bmatrix} \quad \lambda_1, \lambda_2 - \text{eigenvalues of } M$$



Harris Detector: Mathematics

Classification of
image points
using eigenvalues
of M :

λ_1 and λ_2 are
small;
 E is almost
constant in all
directions



Measure of corner
response:

$$R = \det M - k (\text{trace } M)^2$$

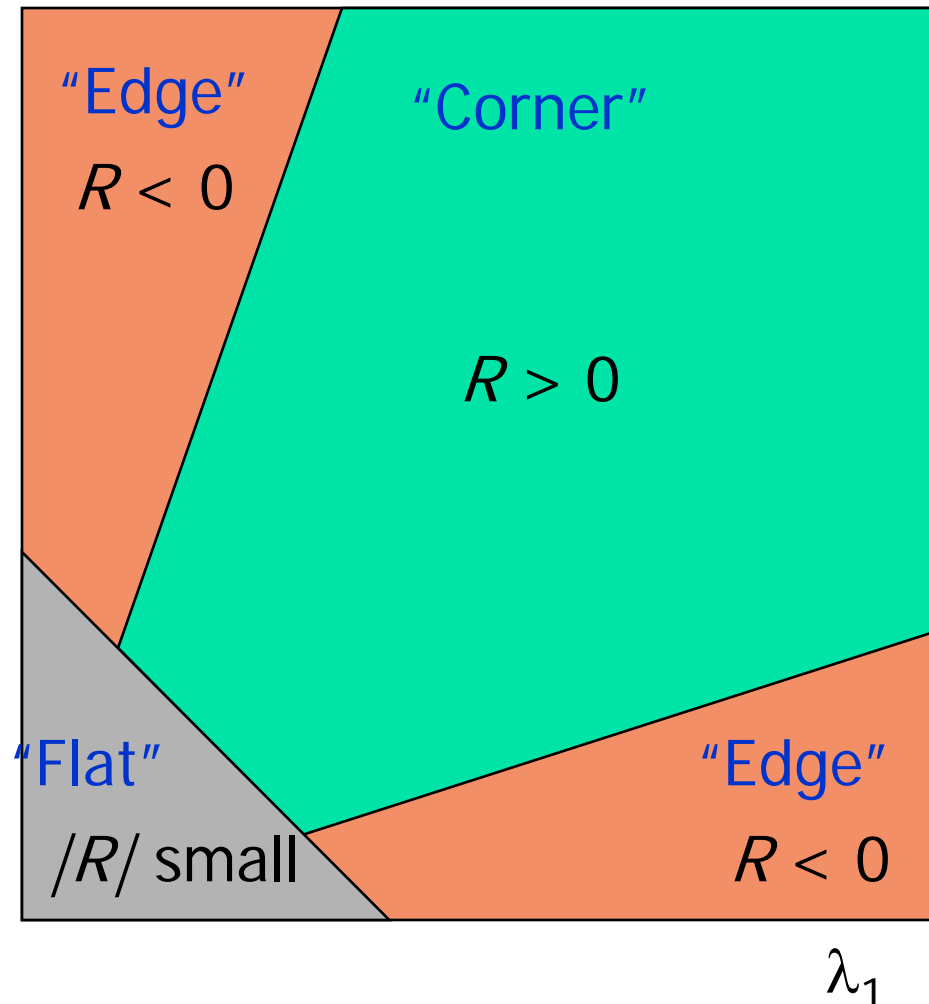
$$\det M = \lambda_1 \lambda_2$$

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

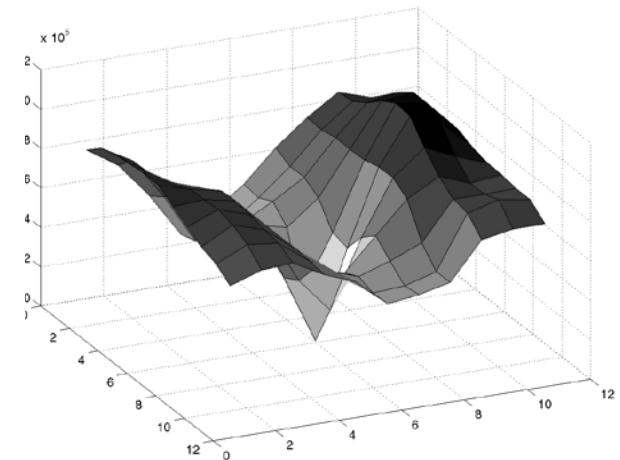
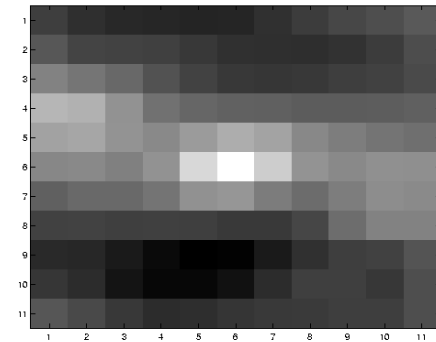
(k – empirical constant, $k = 0.04$ - 0.06)

Harris Detector: Mathematics

- R depends only on eigenvalues of M
- R is large for a **corner**
- R is negative with large magnitude for an **edge**
- $|R|$ is small for a **flat** region

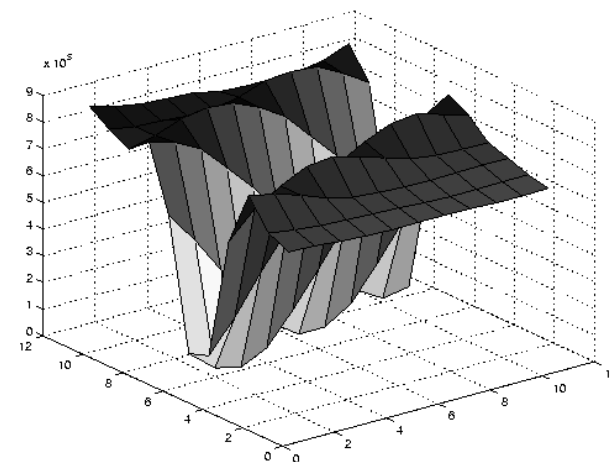
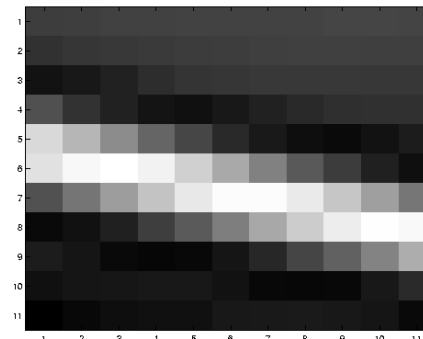


Selecting Good Features



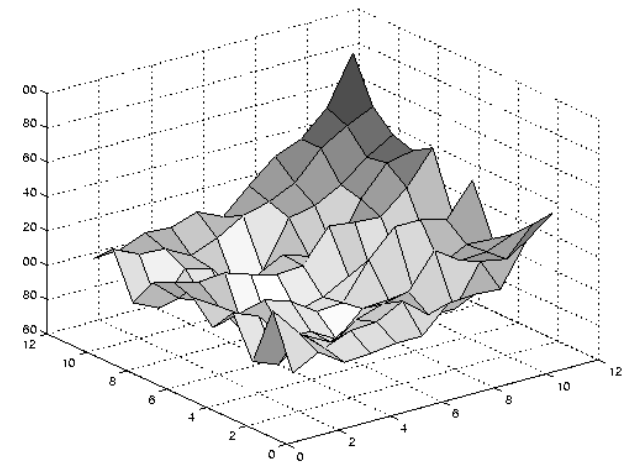
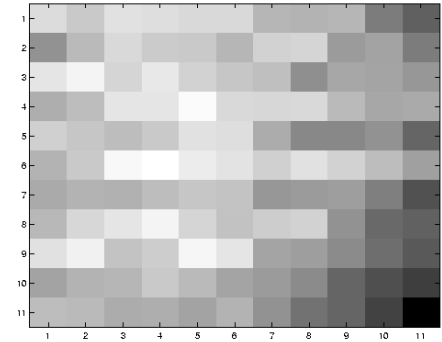
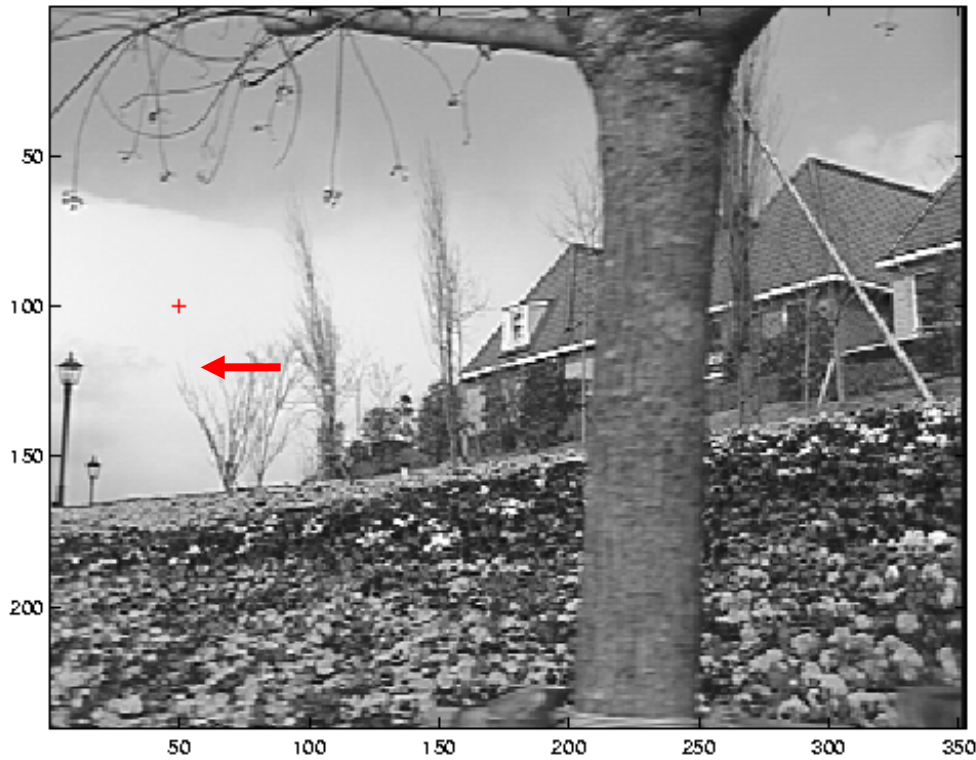
λ_1 and λ_2 are large

Selecting Good Features



large λ_1 , small λ_2

Selecting Good Features



small λ_1 , small λ_2

■ The Algorithm:

- Find points with large corner response function R ($R >$ threshold)
- Take the points of local maxima of R

■ Parameters:

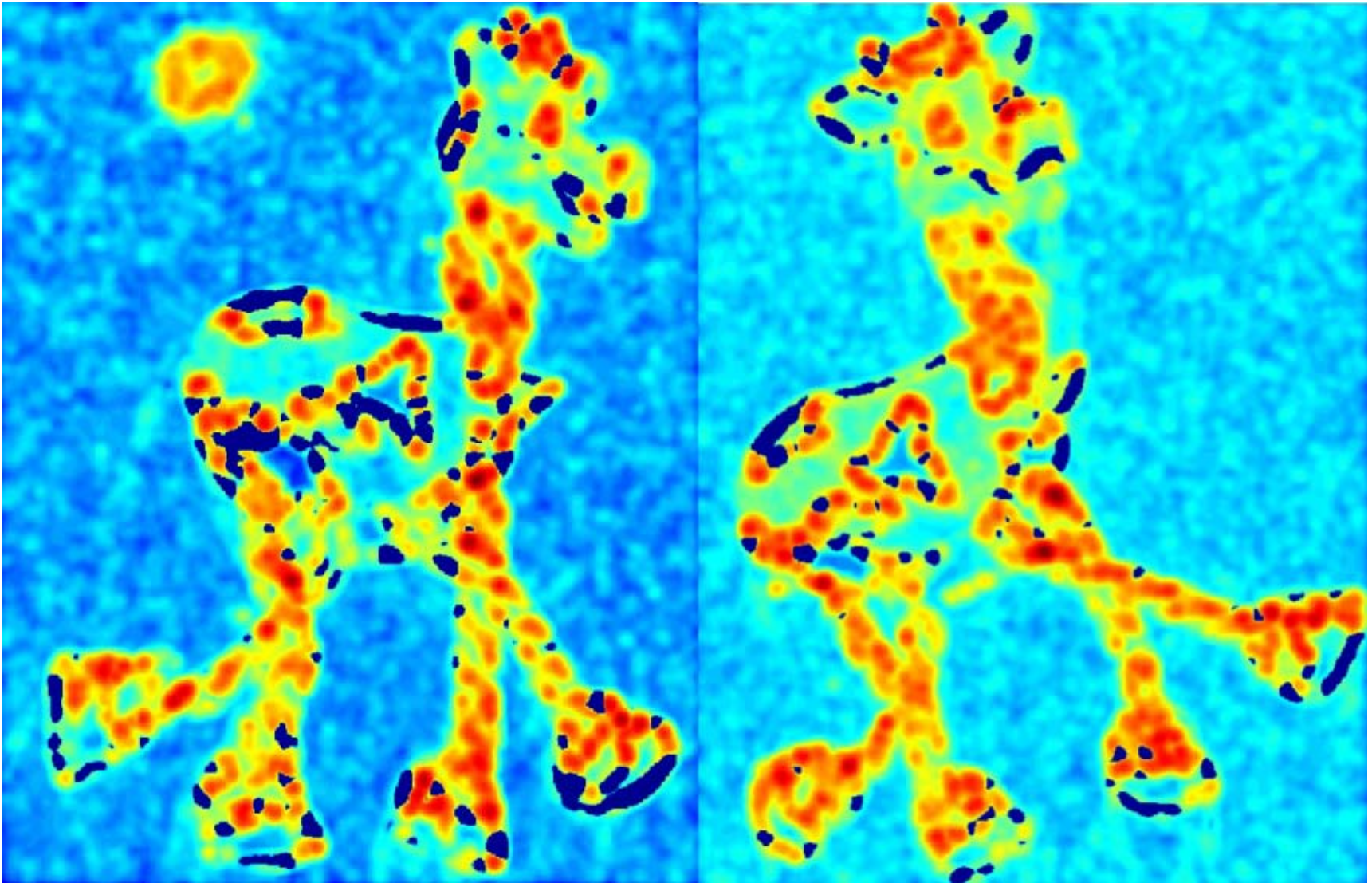
- Threshold on R
- Scale of the derivative operator (standard setting: very small, just enough to filter anisotropy of the image grid)
- Size of window W (“integration scale”)
- Non-maximum suppression algorithm

Harris Detector: Workflow



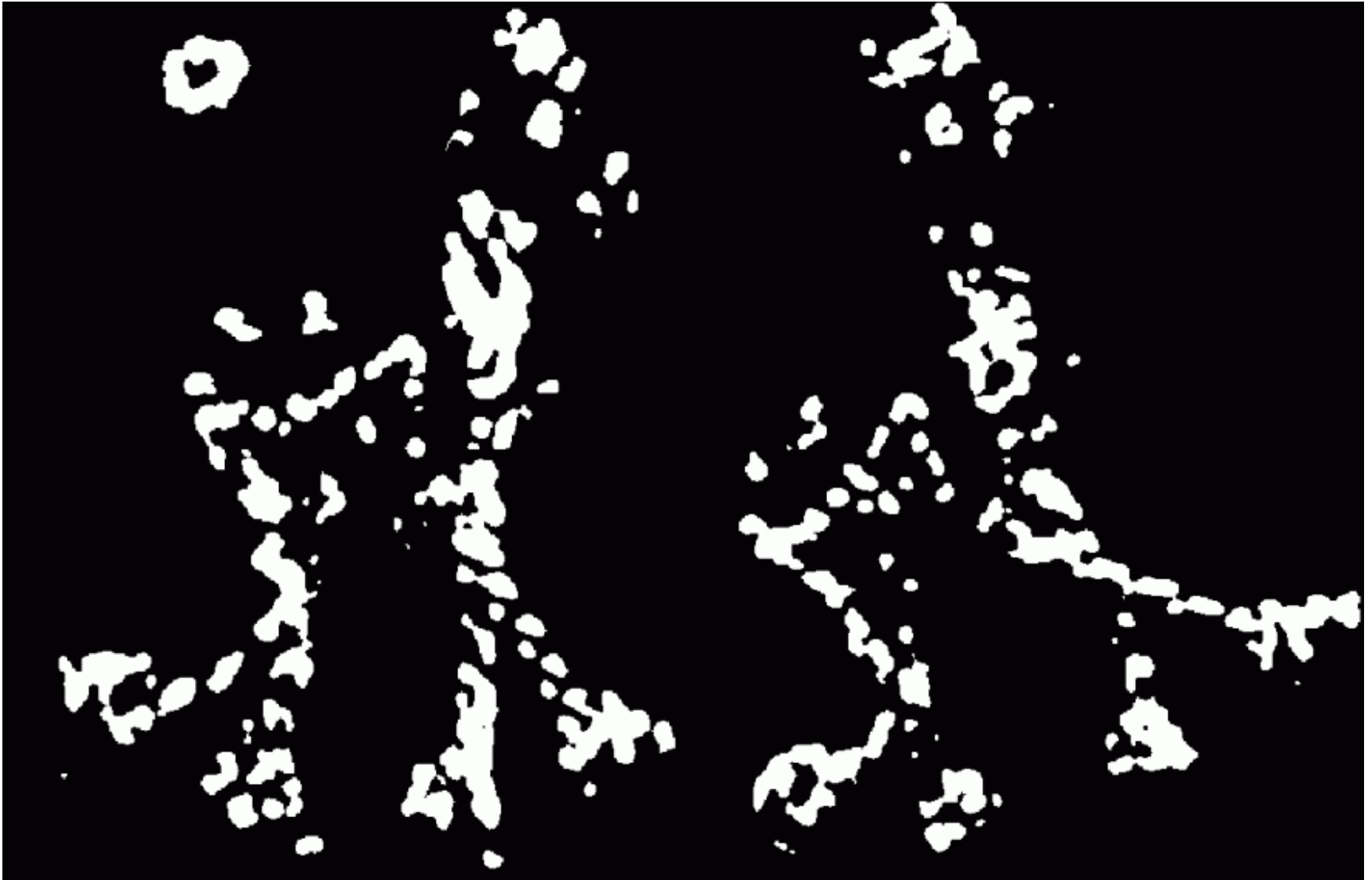
Harris Detector: Workflow

Compute corner response R



Harris Detector: Workflow

Find points with large corner response: $R > \text{threshold}$



Harris Detector: Workflow

Take only the points of local maxima of R



Harris Detector: Workflow



Harris Detector: Summary

- Average intensity change in direction $[u, v]$ can be expressed as a bilinear form:

$$E(u, v) \cong [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

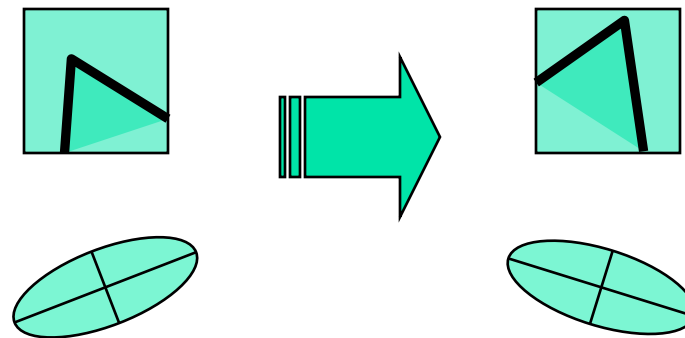
- Describe a point in terms of eigenvalues of M :
measure of corner response

$$R = \lambda_1 \lambda_2 - k (\lambda_1 + \lambda_2)^2$$

- A good (corner) point should have a *large intensity change in all directions*, i.e. R should be large positive

Harris Detector: Properties

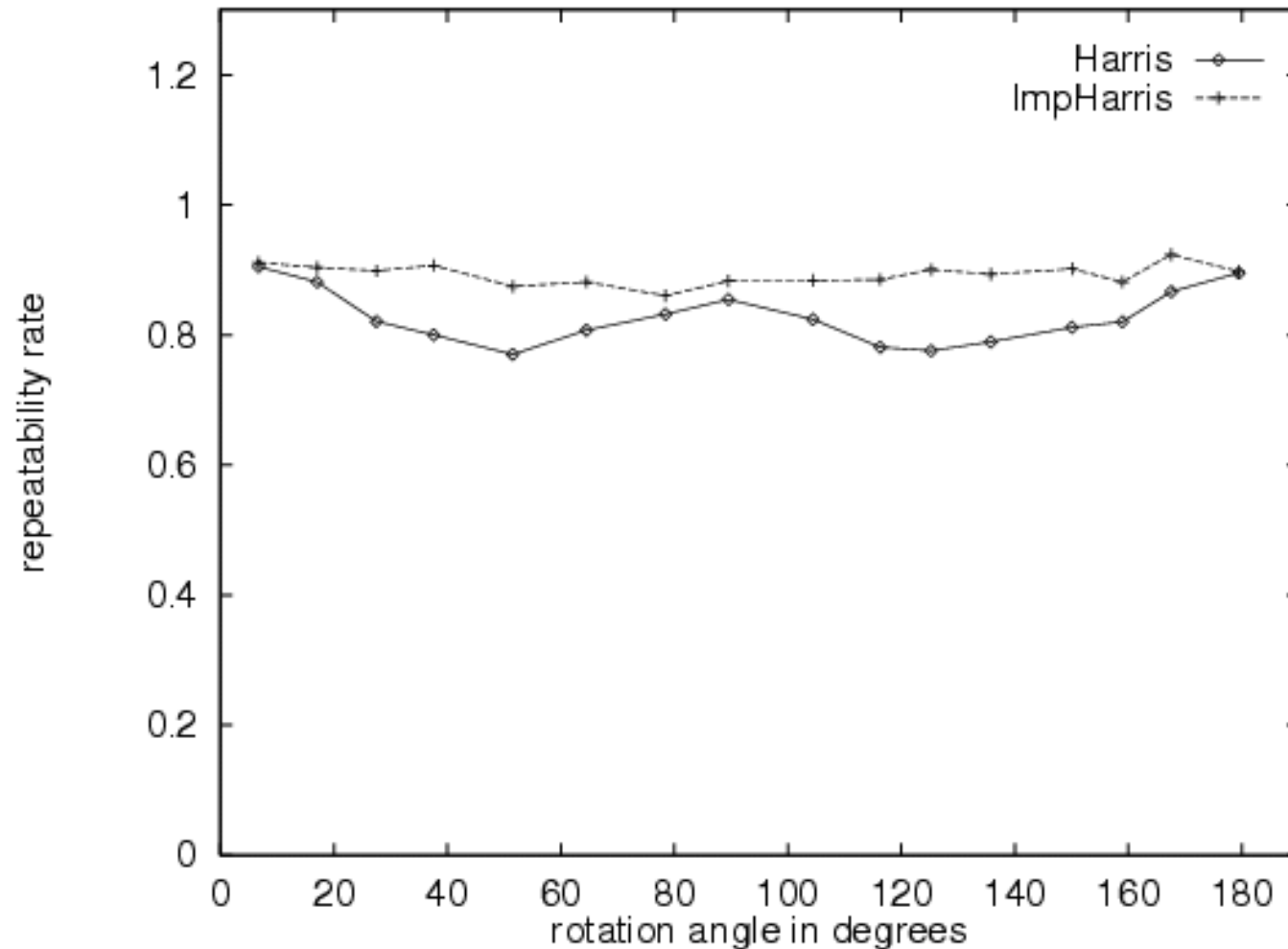
- Rotation invariance



Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner response R is invariant to image rotation

Rotation Invariance of Harris Detector

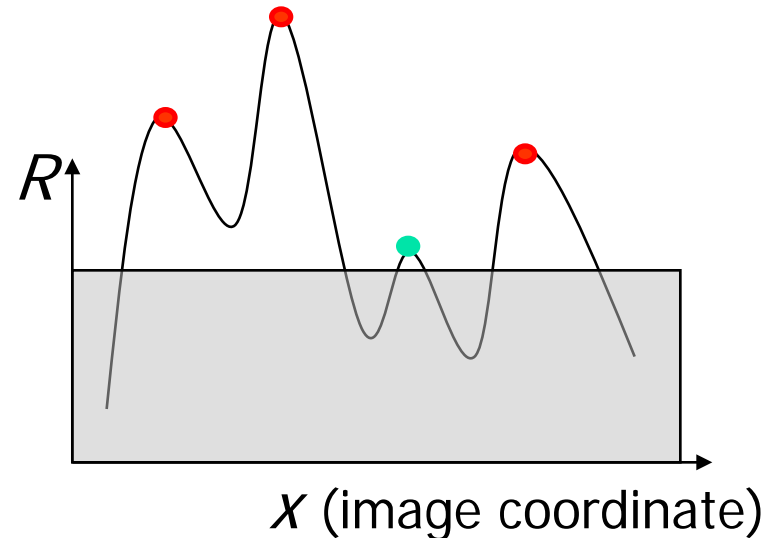
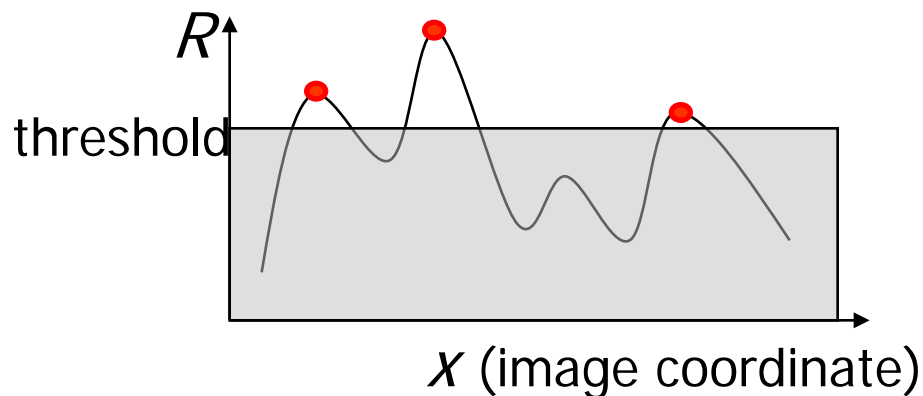


Harris Detector: Intensity change

- Partial invariance to additive and multiplicative intensity changes

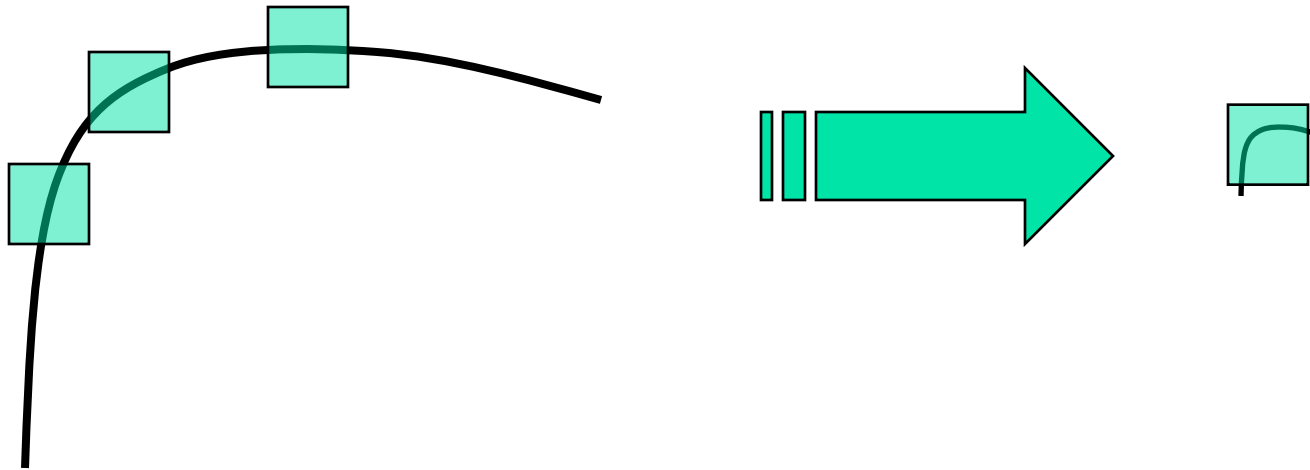
✓ Only derivatives are used \Rightarrow
invariance to intensity shift $I \rightarrow I + b$

? Intensity scale: $I \rightarrow a I$



Harris Detector: Scale Change

- Not invariant to *image scale*!



All points will be
classified as **edges**

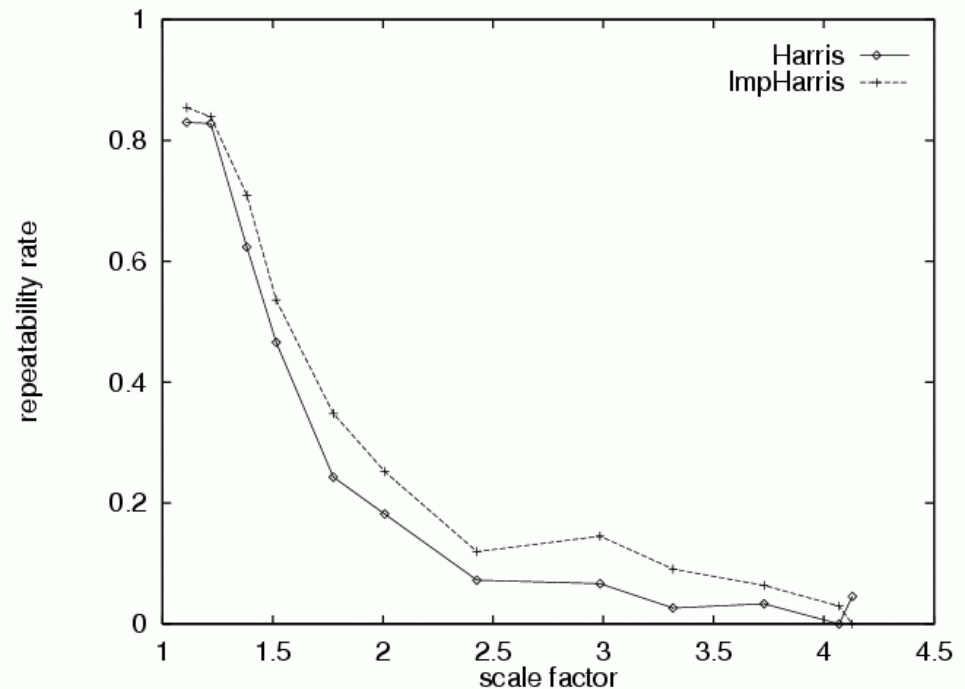
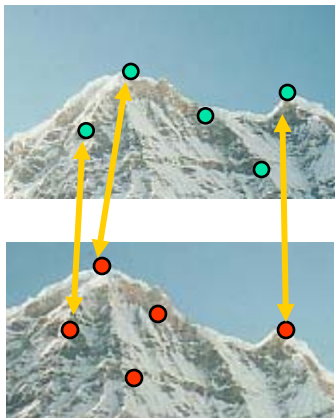
Corner !

Harris Detector: Scale Change

- Quality of Harris detector for different scale changes

Repeatability rate:

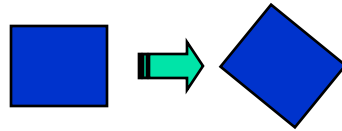
$\frac{\# \text{ correspondences}}{\# \text{ possible correspondences}}$



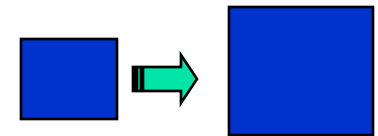
Models of Image Change

■ Geometry

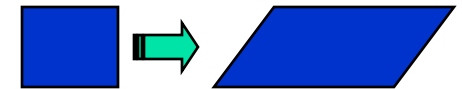
- Rotation



- Similarity (rotation + uniform scale)



- Affine (scale dependent on direction)
valid for: orthographic camera, locally planar object



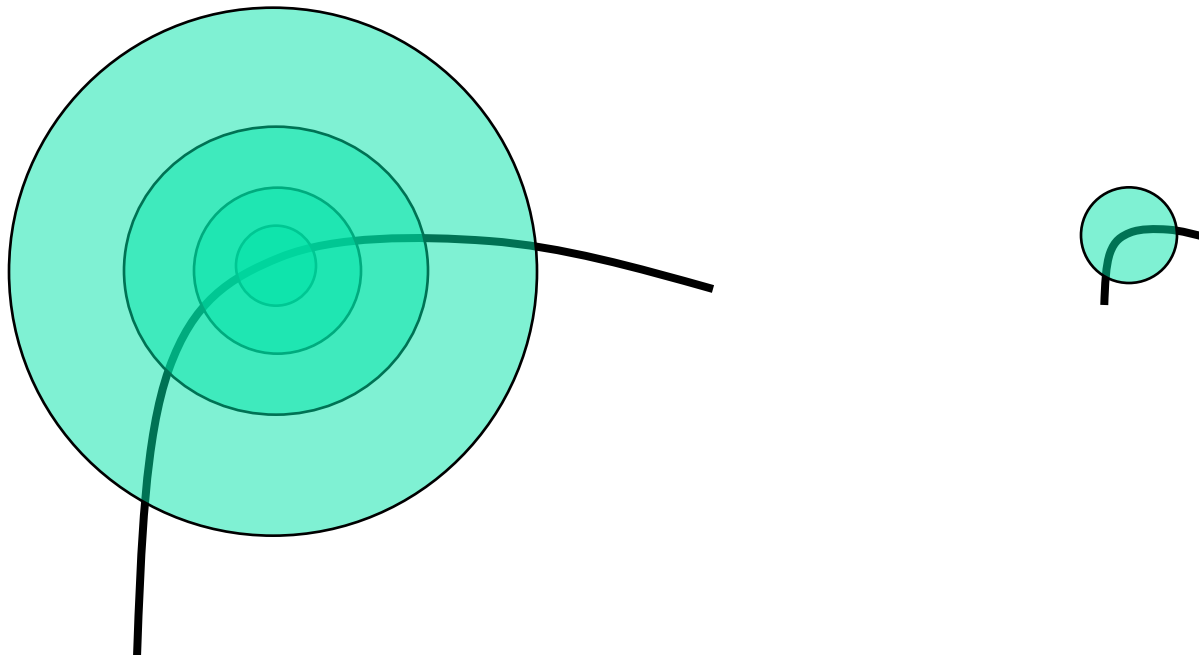
■ Photometry

- Affine intensity change ($I \rightarrow a I + b$)



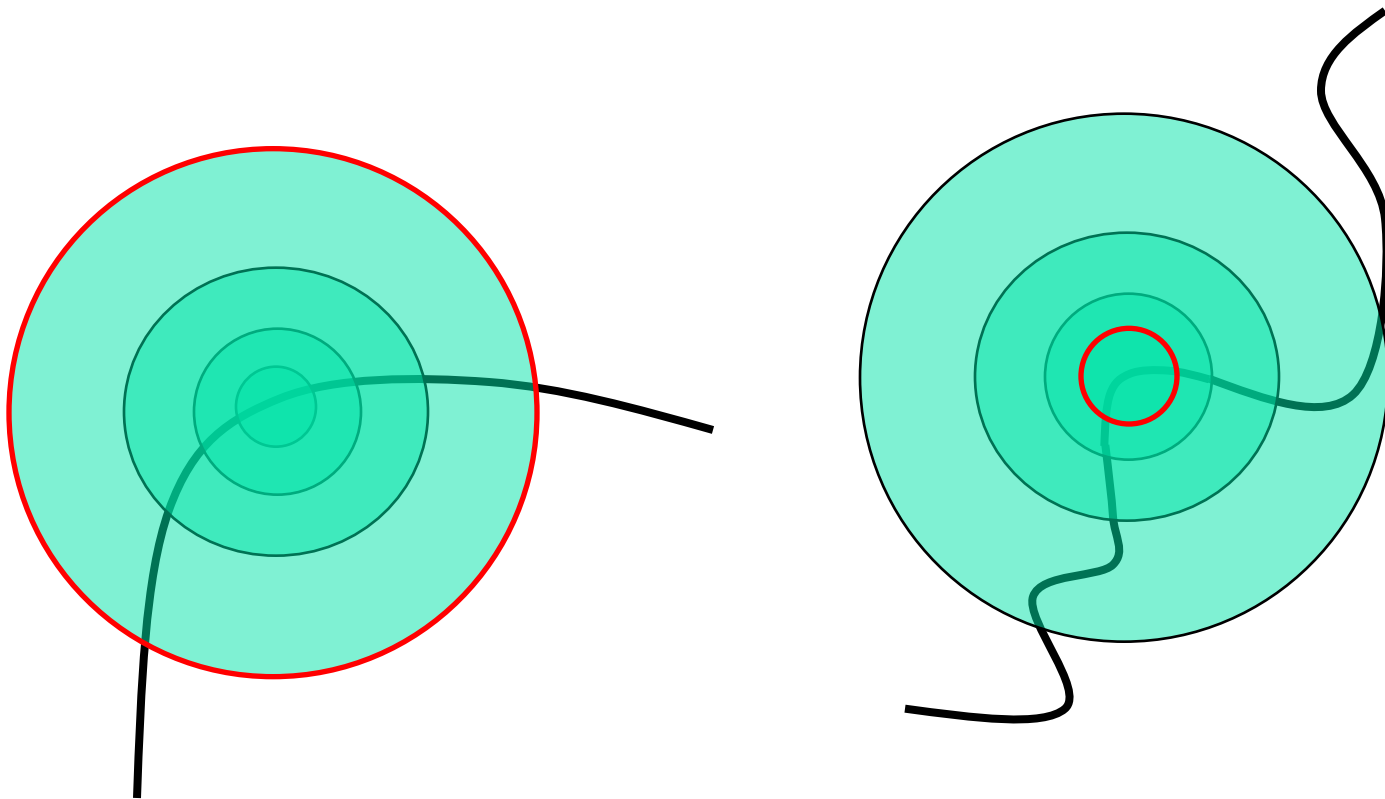
Scale Invariant Detection

- Consider regions (e.g. circles) of different sizes around a point
- Regions of corresponding sizes will look the same in both images



Scale Invariant Detection

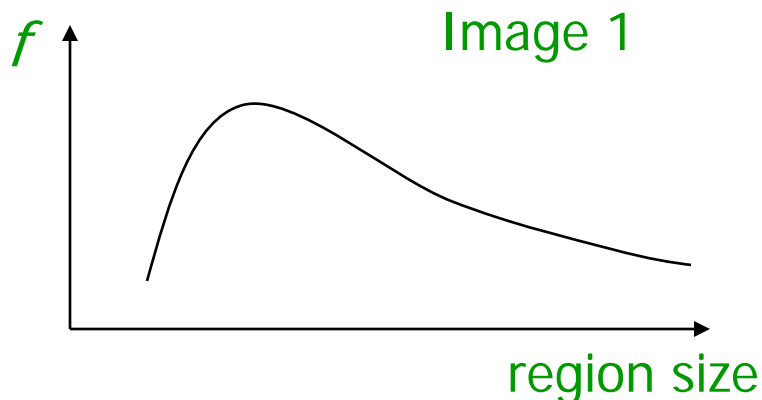
- The problem: how do we choose corresponding circles *independently* in each image?



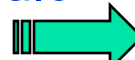
Scale Invariant Detection

■ Solution:

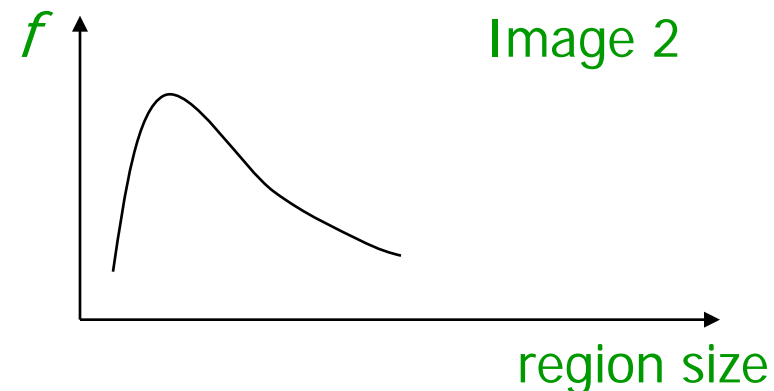
- Design a function on the region (circle), which is “scale covariant” (the same for corresponding regions, even if they are at different scales)
- For a point in one image, we can consider it as a function of region size (circle radius)



scale = 1/2



A green arrow points from the Image 1 graph to the Image 2 graph, indicating a scaling operation.



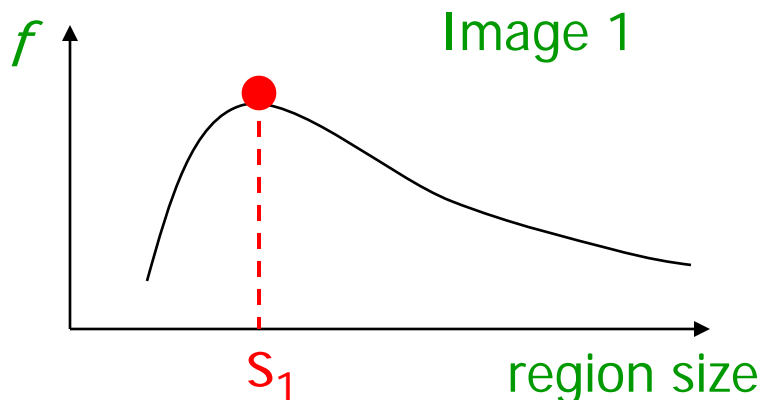
Scale Invariant Detection

■ Common approach:

Take a local maximum of some function

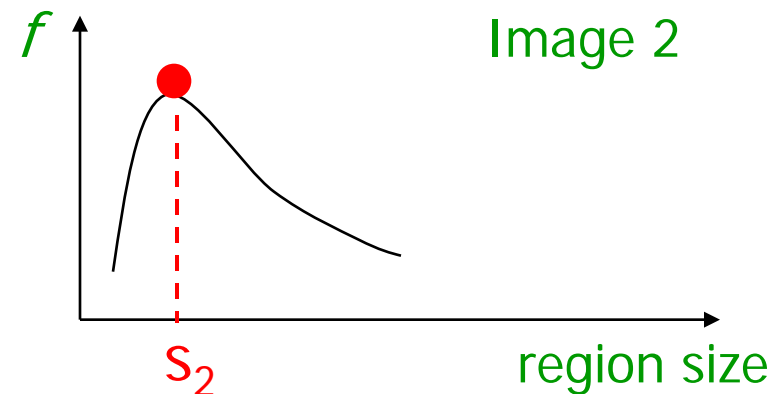
Observation: region size, for which the maximum is achieved, should be *invariant* to image scale.

Important: this scale invariant region size is found in each image **independently!**
(*but think about verification*)



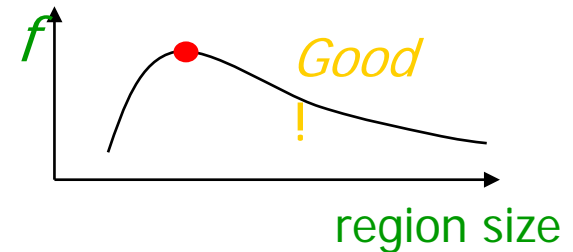
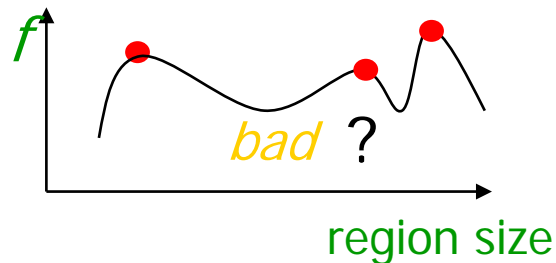
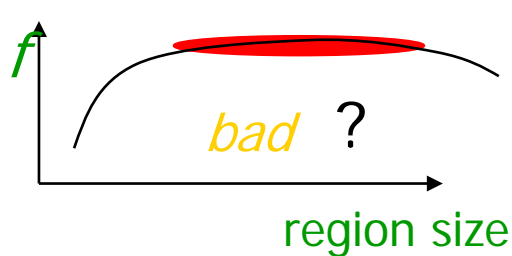
scale = 1/2

Detailed description: A green arrow pointing from the 'Image 1' graph to the 'Image 2' graph. Above the arrow, the text 'scale = 1/2' is written in blue.



Scale Invariant Detection

- A “good” function for scale detection:
has one stable sharp peak



- For usual images: a good function would be a one which responds to contrast (sharp local intensity change)

Scale Invariant Detection

- Functions for determining scale $f = \text{Kernel} * \text{Image}$

Kernels:

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

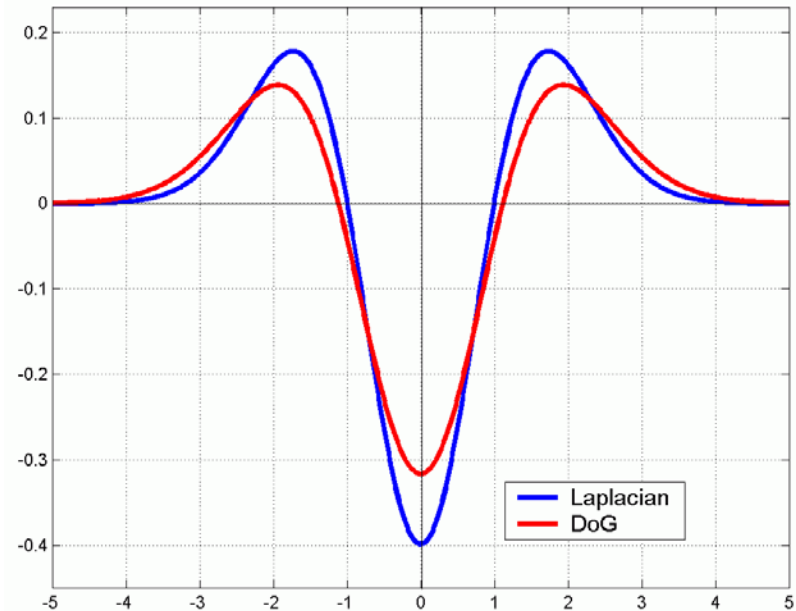
(Laplacian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)

where Gaussian

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



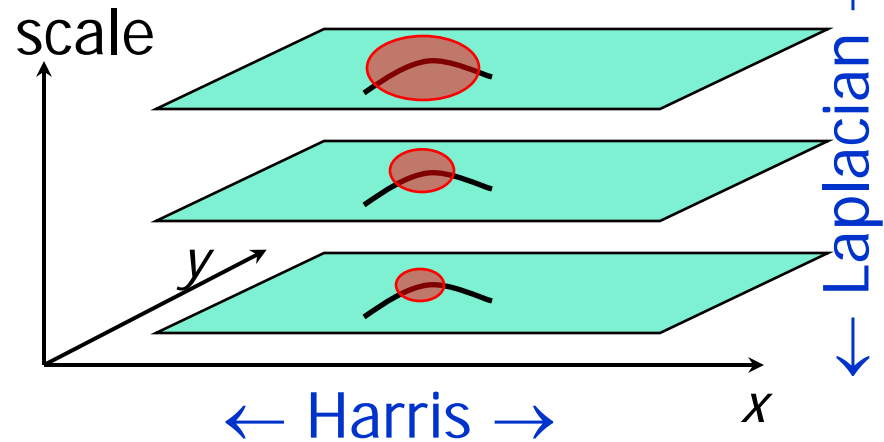
Note: both kernels are invariant to *scale* and *rotation*

Scale Invariant Detectors

Harris-Laplacian¹

Find local maximum of:

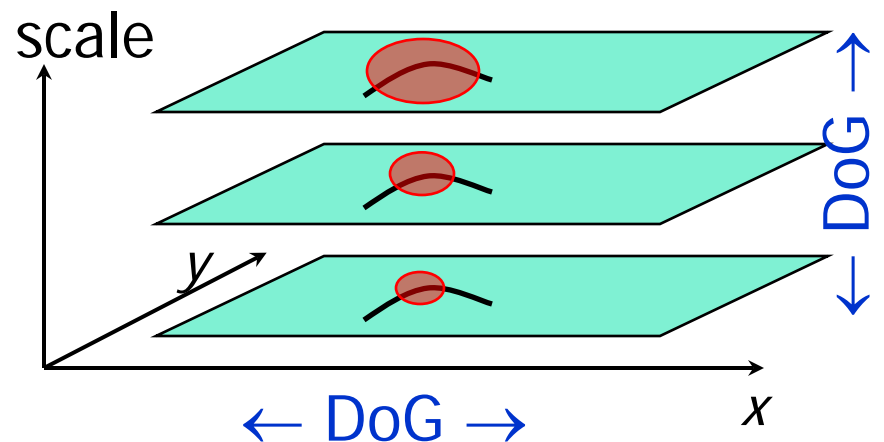
- Harris corner detector in space (image coordinates)
- Laplacian in scale



Laplacian-Laplacian = “SIFT” (Lowe)²

Find local maximum of:

- Difference of Gaussians in space and scale



Other options: Hessian, ...

Harris does not work well for scale selection

¹ K.Mikolajczyk, C.Schmid. “Indexing Based on Scale Invariant Interest Points”. ICCV 2001

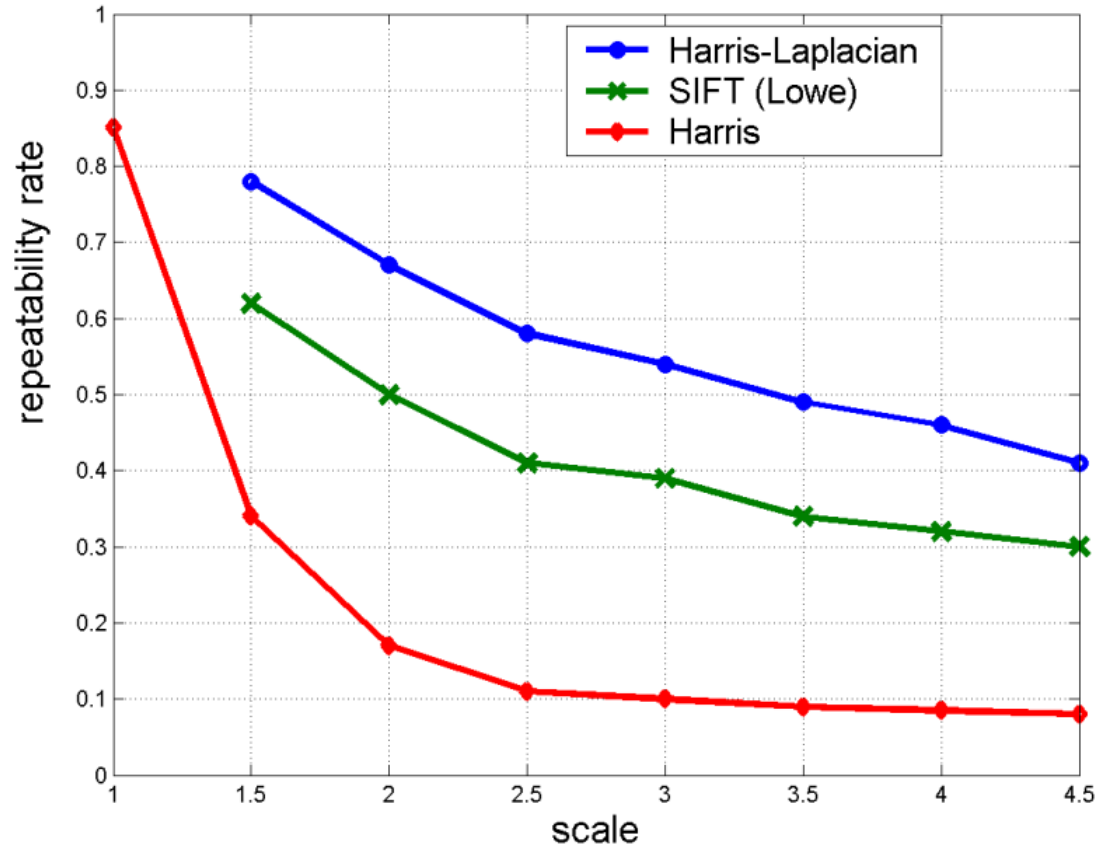
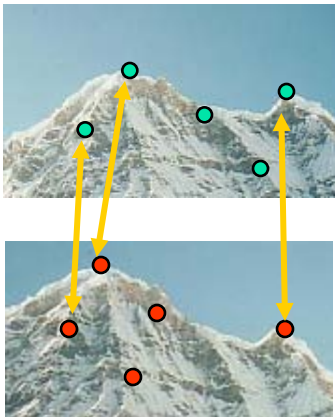
² D.Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. IJCV 2004

Scale Invariant Detectors

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

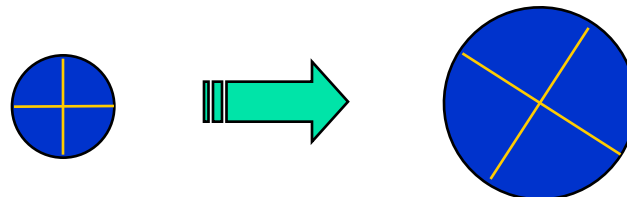
Repeatability rate:

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

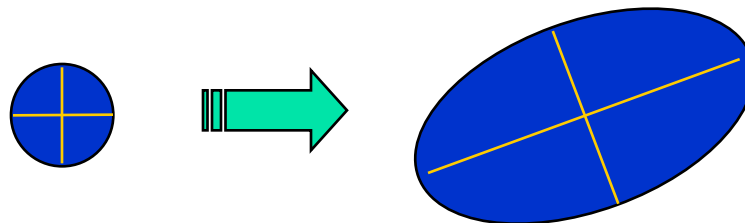


Affine Invariant Detection

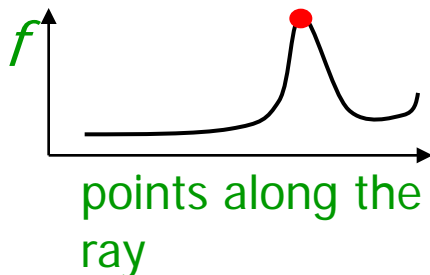
- Above we considered:
Similarity transform (rotation + uniform scale)



- Now we go on to:
Affine transform (rotation + non-uniform scale)



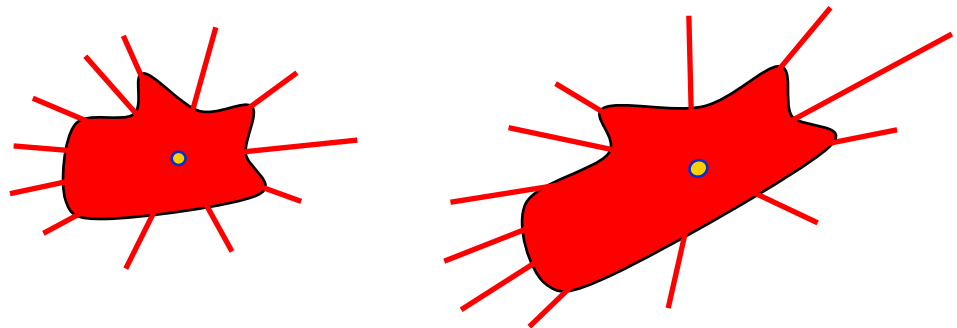
- Take a local intensity extremum as initial point
- Go along every ray starting from this point and stop when extremum of function f is reached



$$f(t) = \frac{|I(t) - I_0|}{\frac{1}{t} \int_0^t |I(t) - I_0| dt}$$

- We will obtain approximately corresponding regions

Remark: we search for scale in every direction



Affine Invariant Detection

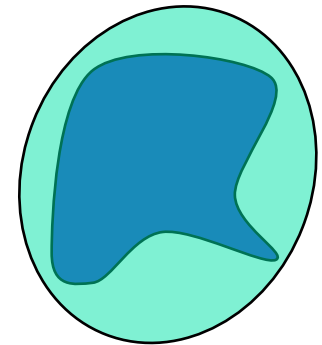
- The regions found may not exactly correspond, so we approximate them with **ellipses**
- Geometric Moments:

$$m_{pq} = \int_{\mathbb{R}^2} x^p y^q f(x, y) dx dy$$

Fact: moments m_{pq}
uniquely determine the
function f

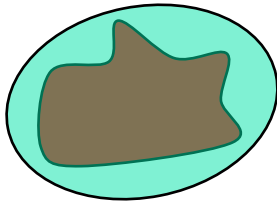
Taking f to be the characteristic function of a region (1 inside, 0 outside), moments of orders up to 2 allow to approximate the region by an ellipse

This ellipse will have the same moments of orders up to 2 as the original region



Affine Invariant Detection

- Covariance matrix of region points defines an ellipse:

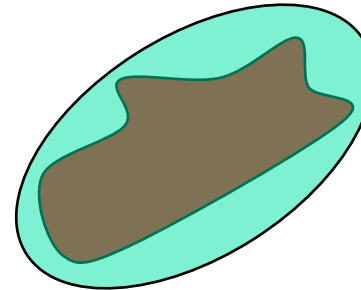
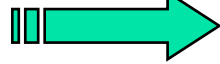


$$p^T \Sigma_1^{-1} p = 1$$

$$\Sigma_1 = \left\langle pp^T \right\rangle_{\text{region 1}}$$

($p = [x, y]^T$ is
relative to the center
of mass)

$$q = Ap$$



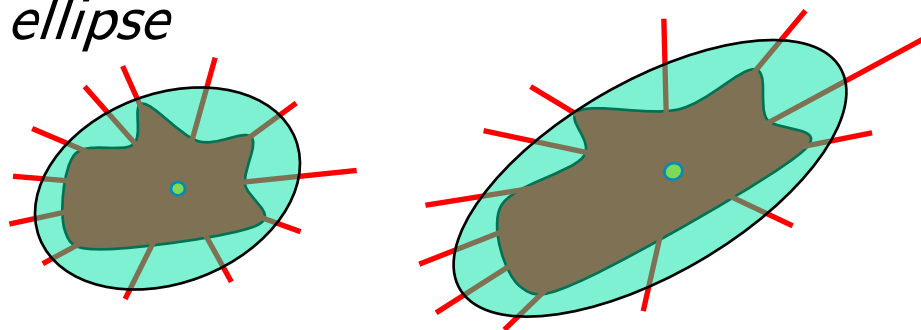
$$q^T \Sigma_2^{-1} q = 1$$

$$\Sigma_2 = \left\langle qq^T \right\rangle_{\text{region 2}}$$

$$\Sigma_2 = A \Sigma_1 A^T$$

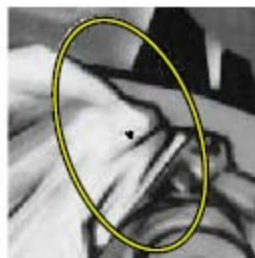
Ellipses, computed for
corresponding regions, also
correspond!

- Algorithm summary (detection of affine invariant region):
 - Start from a *local intensity extremum* point
 - Go in *every direction* until the point of extremum of some function f
 - Curve connecting the points is the region boundary
 - Compute *geometric moments* of orders up to 2 for this region
 - Replace the region with *ellipse*

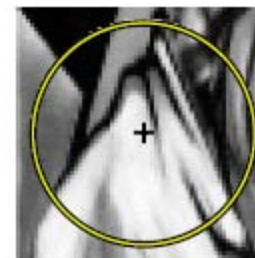


Harris/Hessian Affine Detector

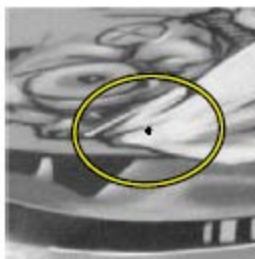
1. Detect initial region with Harris or Hessian detector and select the scale.
2. Estimate the shape with the second moment matrix
3. Normalize the affine region to the circular one
4. Go to step 2 if the eigenvalues of the second moment matrix for new point are not equal.



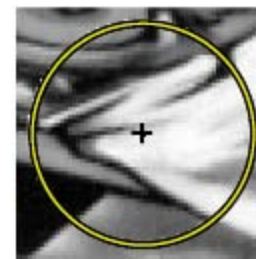
$$\mathbf{x}_L \longrightarrow M_L^{-1/2} \mathbf{x}'_L$$



$$\begin{array}{c} \downarrow \\ \mathbf{x}'_L \longrightarrow R \mathbf{x}'_R \\ \downarrow \end{array}$$



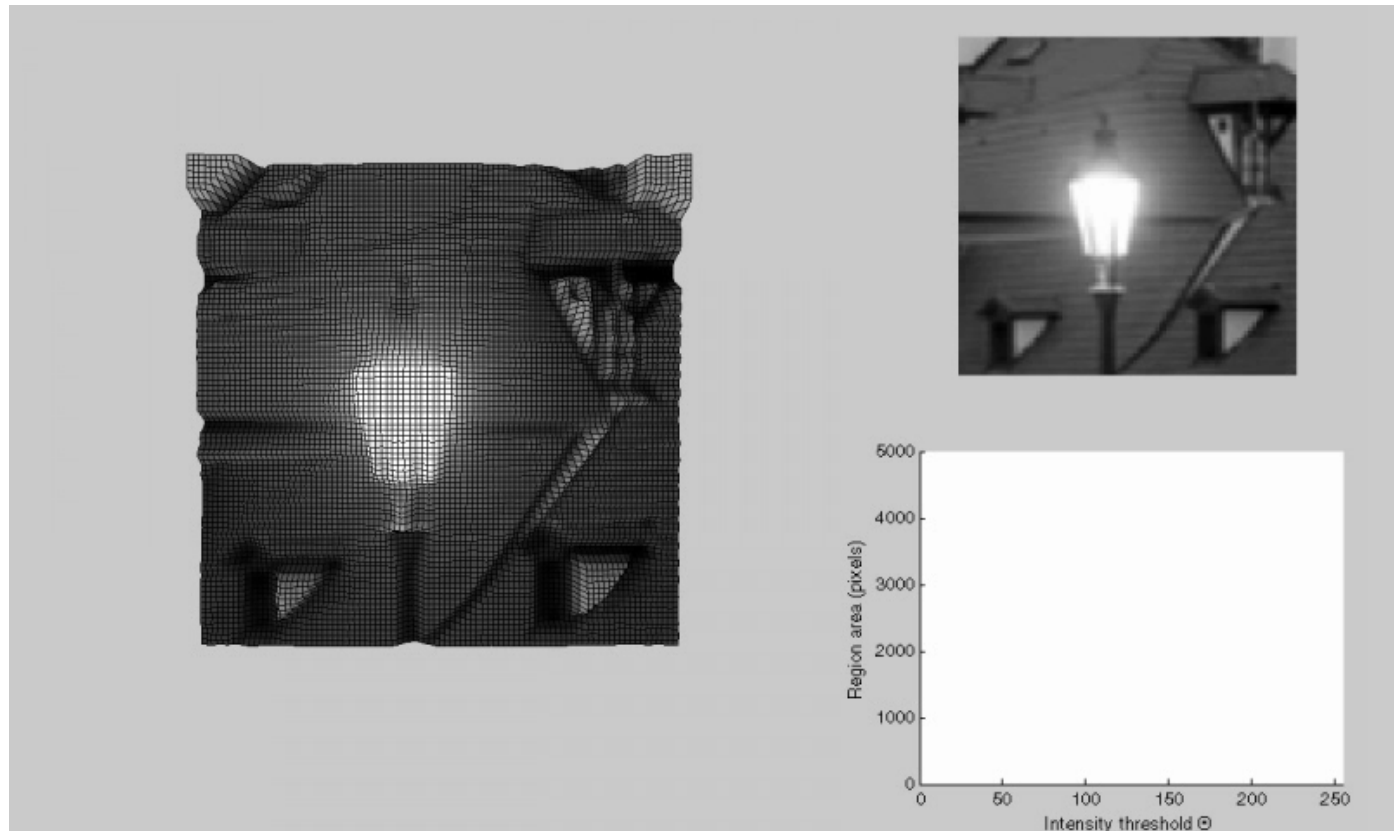
$$\mathbf{x}_R \longrightarrow M_R^{-1/2} \mathbf{x}'_R$$



The Maximally Stable Extremal Regions

- Consecutive image thresholding by all thresholds
- Maintain list of Connected Components
- Regions = Connected Components with stable area (or some other property) over multiple thresholds selected

[video](#)



The Maximally Stable Extremal Regions

video



Step 1: Detect MSERs

Properties:

- Covariant with continuous deformations of images
- Invariant to affine transformation of pixel intensities
- Enumerated in $O(n \log \log n)$, real-time computation



MSER regions (in green). The regions ‘follow’ the object ([video1](#), [video2](#)).

Matas, Chum, Urban, Pajdla: “Robust wide baseline stereo from maximally stable extremal regions”. BMVC2002

Descriptors of Local Invariant Features

■ Image moments in polar coordinates

$$m_{kl} = \iint r^k e^{-i\theta l} I(r, \theta) dr d\theta$$

Rotation in polar coordinates is translation of the angle:

$$\theta \rightarrow \theta + \theta_0$$

This transformation changes only the phase of the moments, but not its magnitude

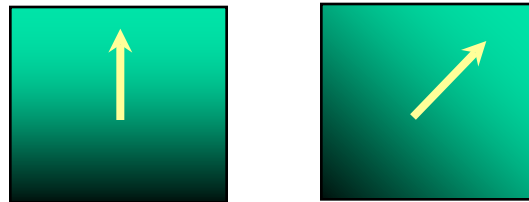
Rotation invariant descriptor
consists of magnitudes of
moments:

$$|m_{kl}|$$

Matching is done by comparing vectors $[|m_{kl}|]_{k,l}$

- Find local orientation

Dominant direction of gradient



- Compute image derivatives relative to this orientation

¹ K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

² D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

- Use the scale determined by detector to compute descriptor in a normalized frame

For example:

- moments integrated over an adapted window
- derivatives adapted to scale: s/x

■ Affine invariant color moments

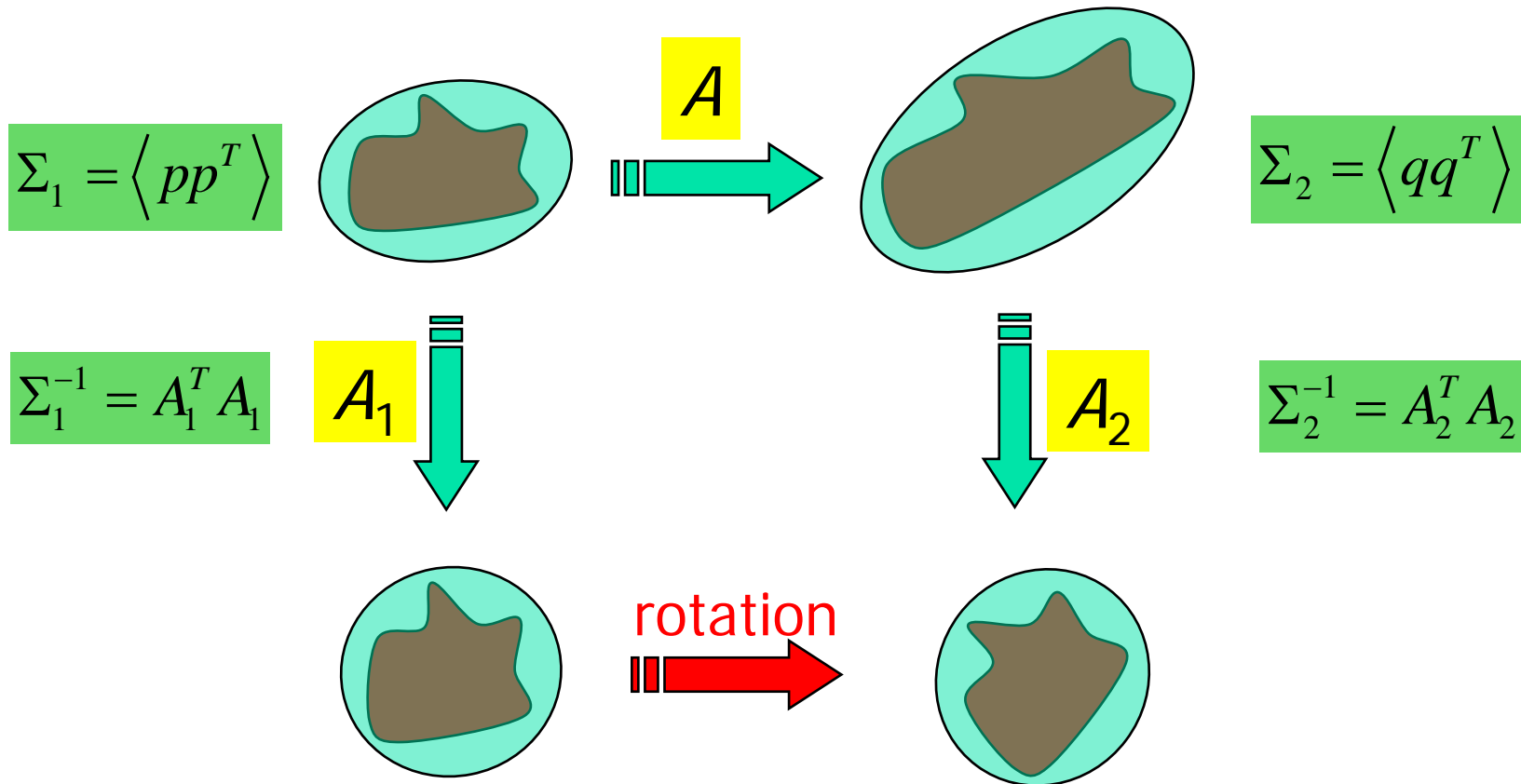
$$m_{pq}^{abc} = \int_{\text{region}} x^p y^q R^a(x, y) G^b(x, y) B^c(x, y) d\mathbf{x}$$

Different combinations of these moments are fully affine invariant

Also invariant to affine transformation of intensity $I \rightarrow aI + b$

Affine Invariant Descriptors

Find affine normalized frame

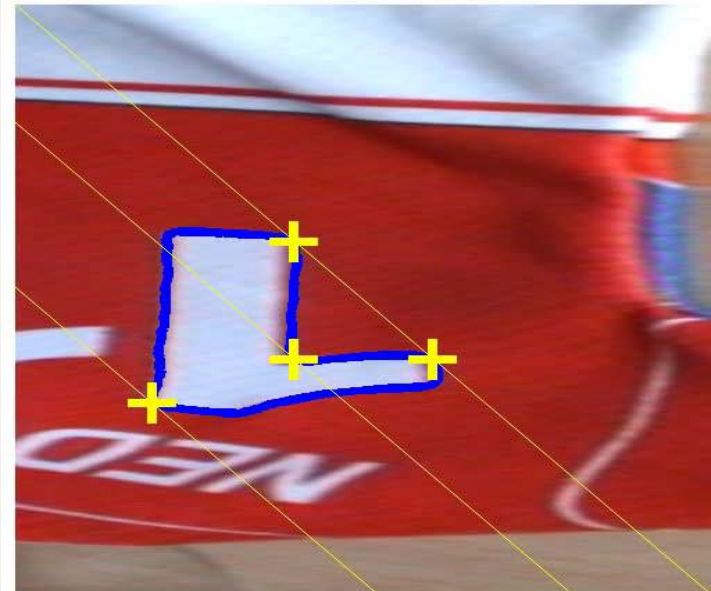


- Compute rotational invariant descriptor in this normalized frame

Step 2: Construct **Local Affine Frames (LAFs)** (local coordinate frames)

Step 3: **Geometrically normalize** some measurement region (MR)
expressed in LAF coordinates

All measurements in the nomalised frame are Invariants!



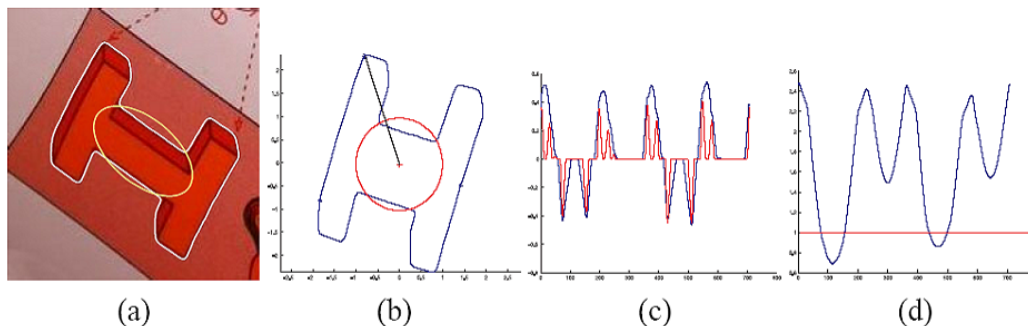
Stability of LAFs: concavity, curvature max 1, curvature max 2

Obdržálek and Matas: "Object recognition using local affine frames on distinguished regions". BMVC02

Obdržálek and Matas: "Sub-linear Indexing for Large Scale Object Recognition", BMVC 2005

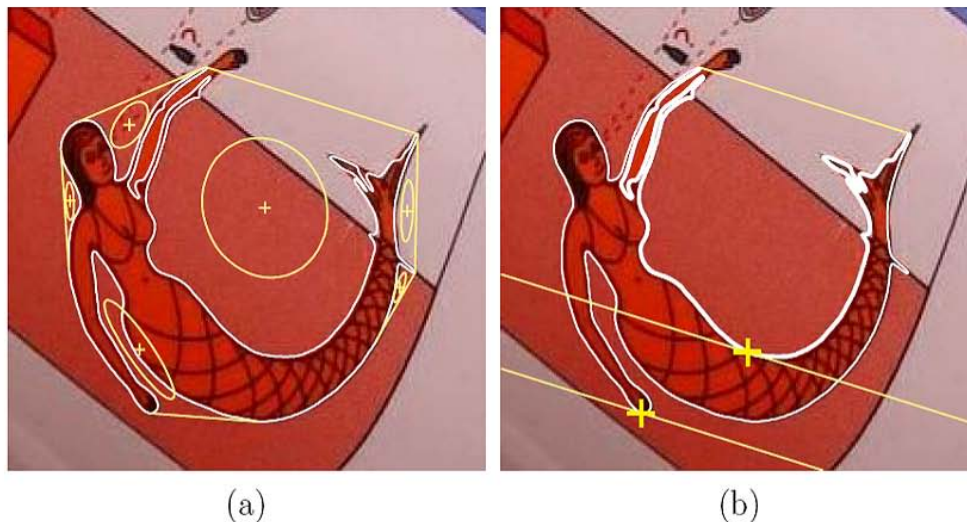
■ Derived from *region outer boundary*

- Region area (1 constraint)
 - Center of gravity (2 constraints)
 - Matrix of second moments (symmetric 2x2 matrix: 3 constraints)
- $$|\Omega| = \int_{\Omega} 1 d\Omega$$
- $$\mu = \frac{1}{|\Omega|} \int_{\Omega} \mathbf{x} d\Omega$$
- $$\Sigma = \frac{1}{|\Omega|} \int_{\Omega} (\mathbf{x} - \mu)(\mathbf{x} - \mu)^T d\Omega$$
- Points of extremal distance to the center of gravity (2 constraints)
 - Points of extremal curvature (2 constraints)



Shape normalisation by the covariance matrix. (a) a detected region, (b) the region shape-normalised to have unit covariance matrix, (c) local curvatures of the normalised shape, (d) distances to the center of gravity.

- Derived from *region outer boundary* (continued)
 - Concavities (4 constraints for 2 tangent points)
 - Farthest point on region contour/concavity (2 constraints)



Example region concavities. (a) A detected non-convex region with indicated concavities and their covariance matrices (b) One of the concavities - the bitangent line and region and concavity farthest points.

- Derived from *image intensities* in a region (or its neighbourhood)
 - From orientation of gradients
 - peaks of gradient orientation histograms [Low04] (1 constraint)
 - Direction of dominant texture periodicity (1 constraint)
 - Extrema or centers of gravity of R, G, B components, or of any scalar function of the RGB values (2 constraints)
 - many other

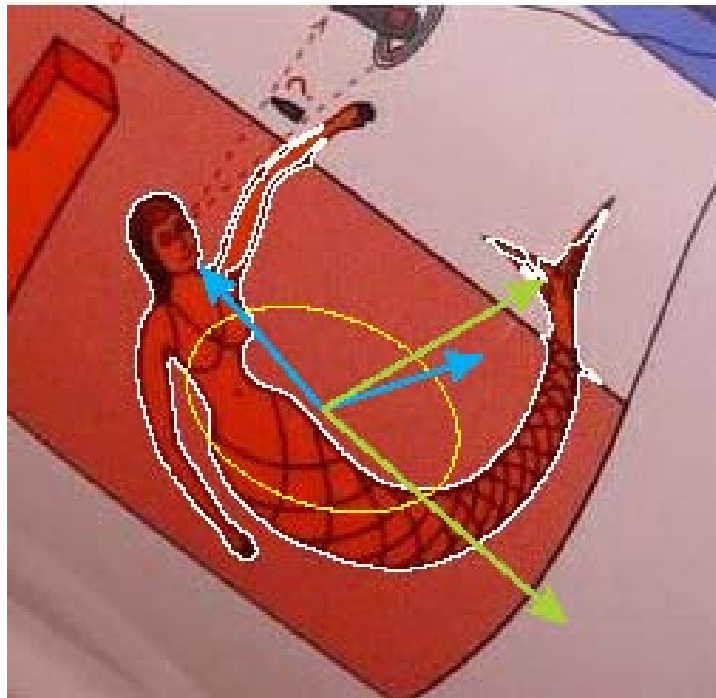
[Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision*, 2004.

- Derived from *topology* of regions
 - mutual configuration of regions (combined constraints)
 - nested regions
 - incident regions
 - neighbouring regions

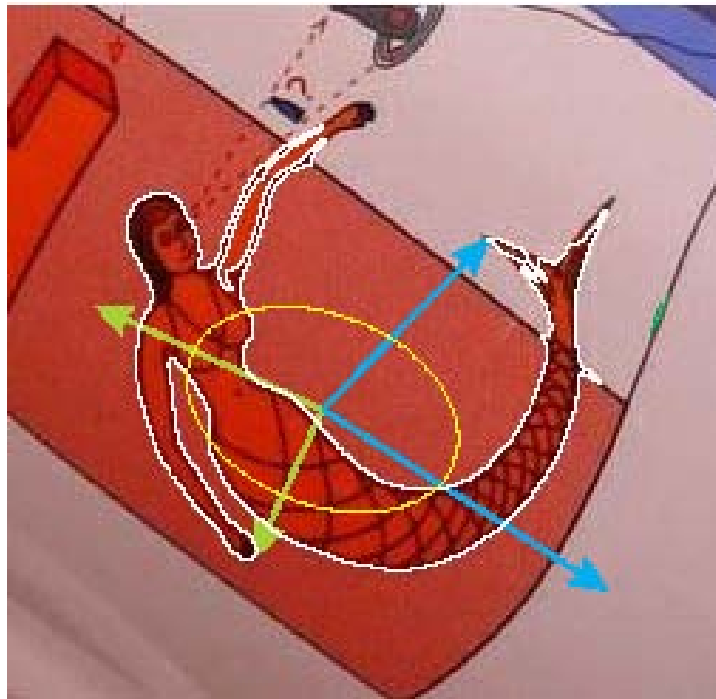
- Region holes and concavities can be considered as regions of their own
 - all aforementioned constructions recursively applicable

- Convex hull of a region without loosing affine invariance

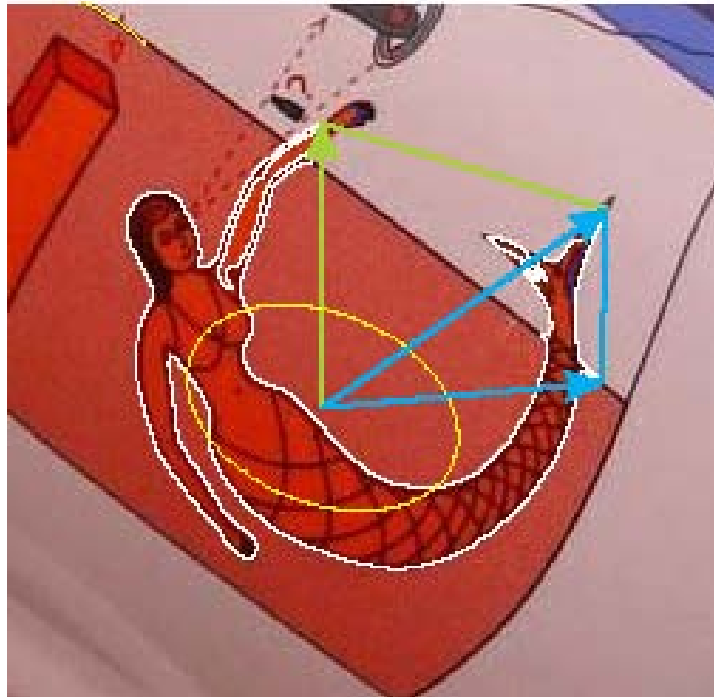
- Combinations of constructions used to form the local affine frames
 - center of gravity + covariance matrix + curvature minima



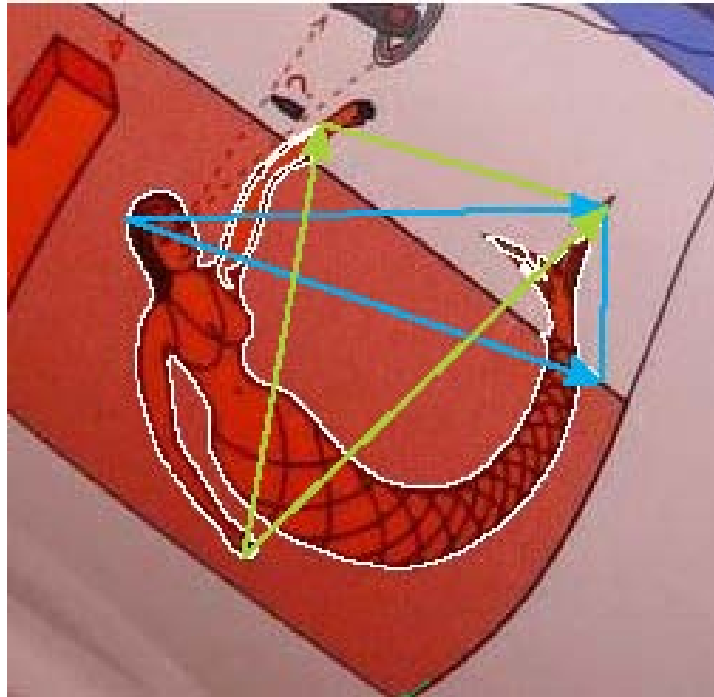
- Combinations of constructions used to form the local affine frames
 - center of gravity + covariance matrix + curvature maxima



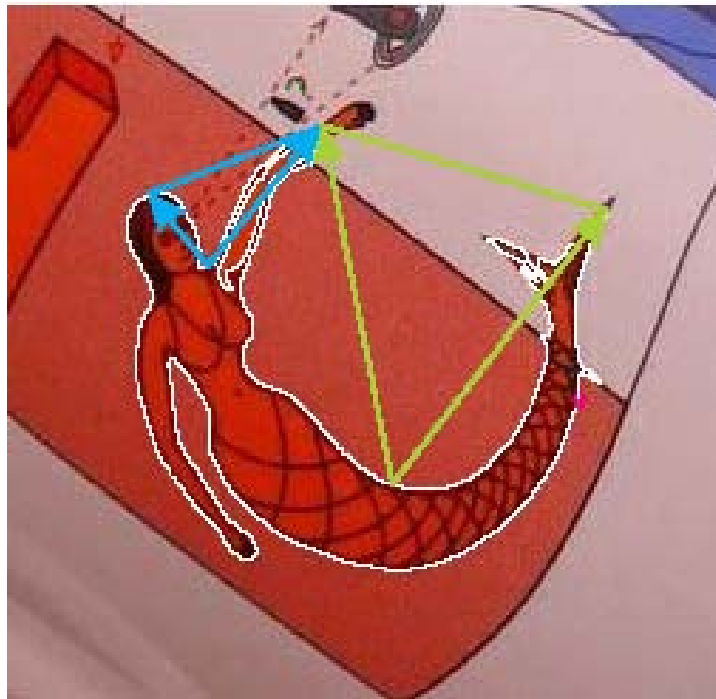
-



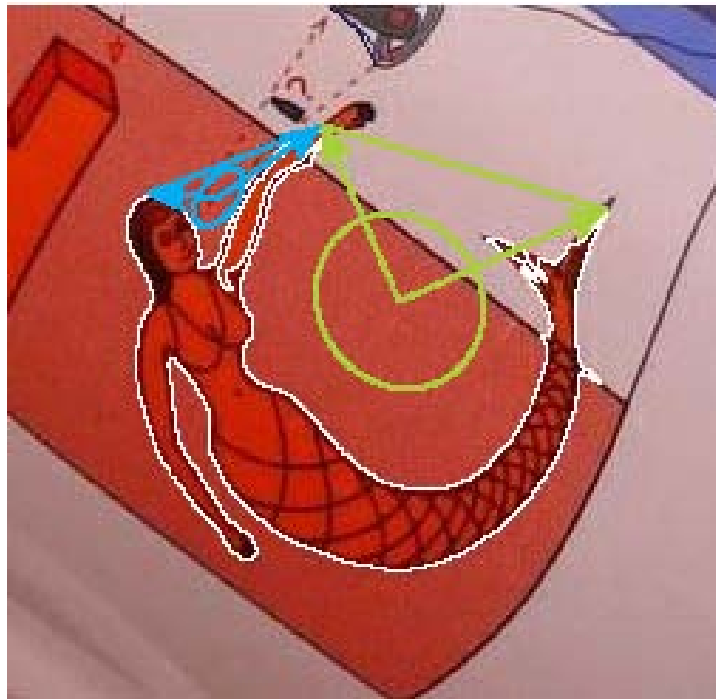
- 



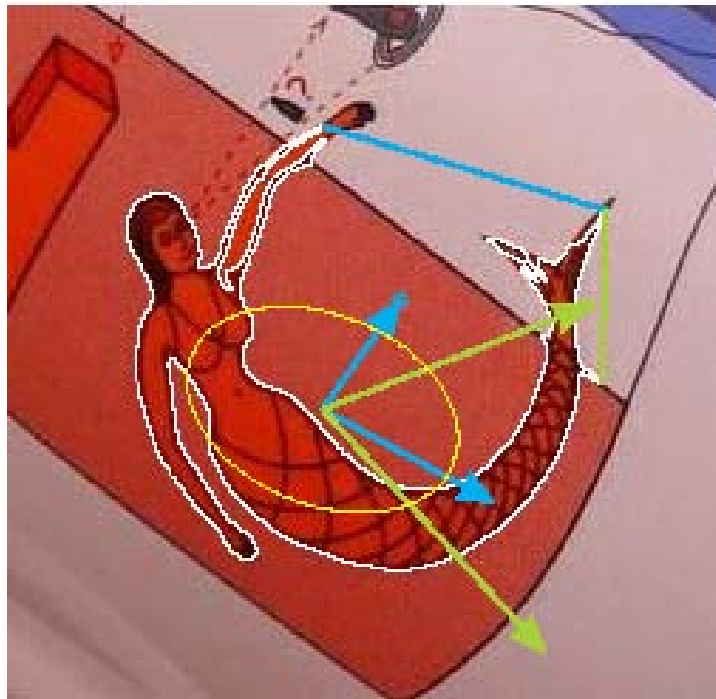
- Combinations of constructions used to form the local affine frames
 - tangent points + farthest point of the concavity



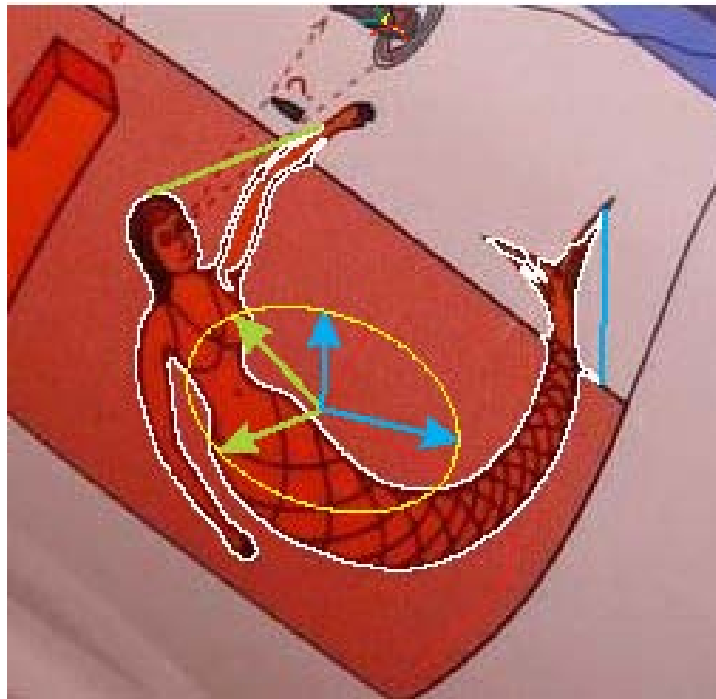
- Combinations of constructions used to form the local affine frames
 - tangent points + center of gravity of the concavity



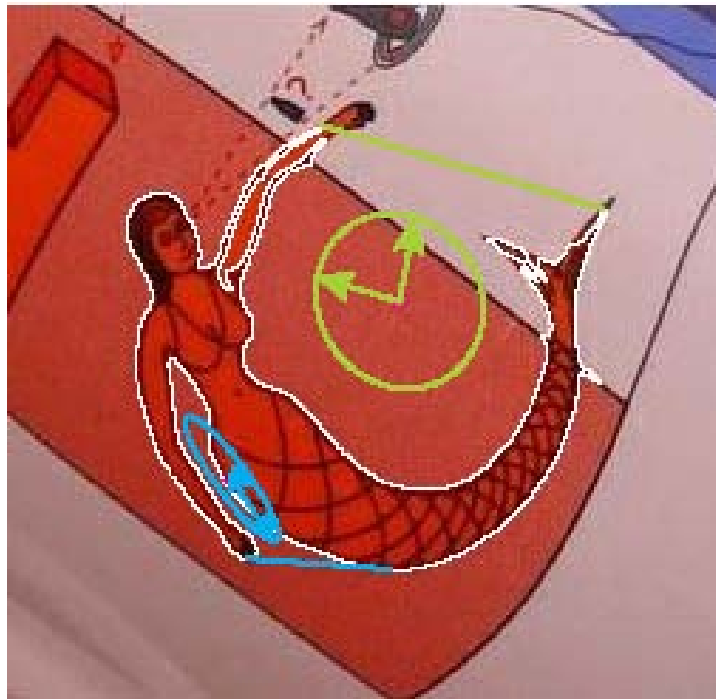
- Combinations of constructions used to form the local affine frames
 - center of gravity + covariance matrix + center of gravity of a concavity



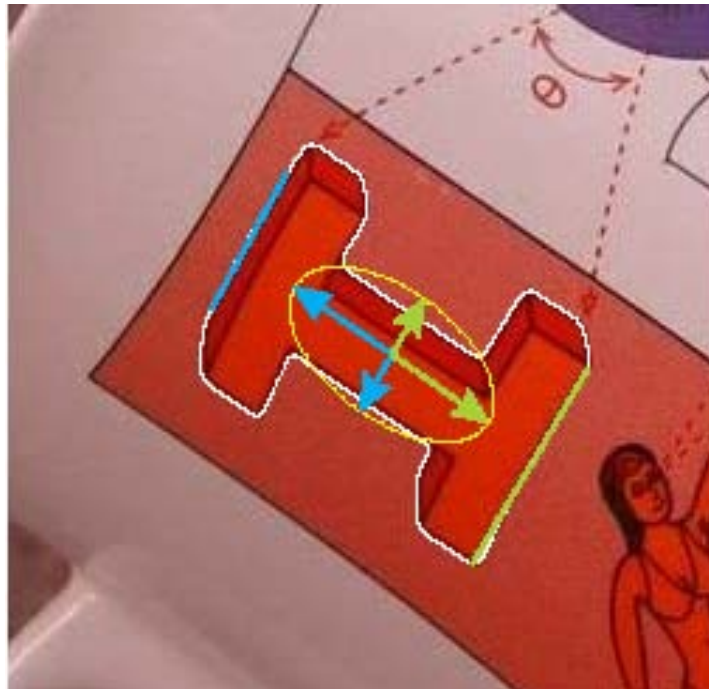
- Combinations of constructions used to form the local affine frames
 - center of gravity + covariance matrix + direction of a bitangent



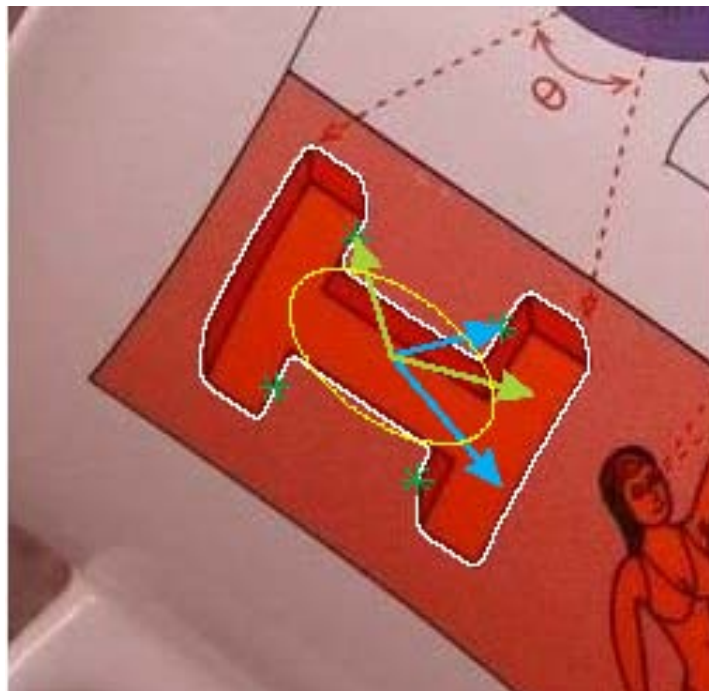
- Combinations of constructions used to form the local affine frames
 - center of gravity of a concavity + covariance matrix of the concavity + the direction of the bitangent



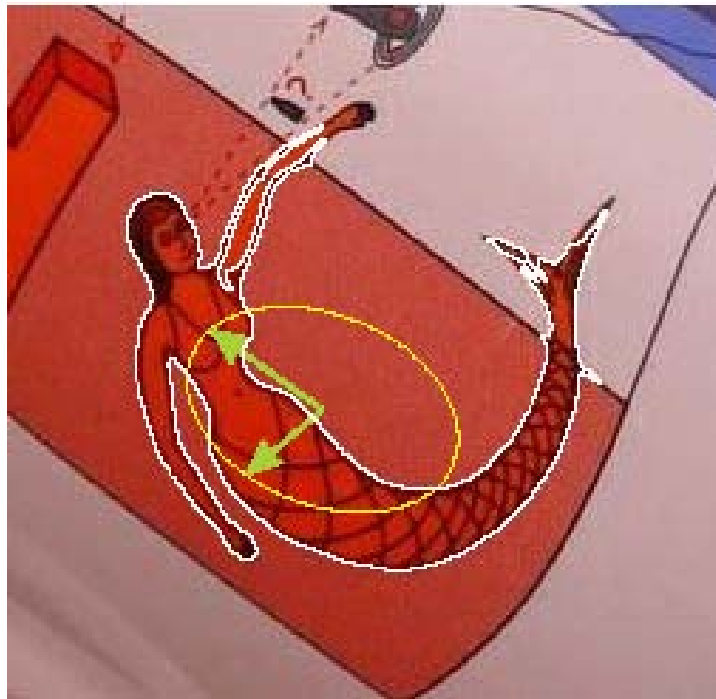
- Combinations of constructions used to form the local affine frames
 - center of gravity + covariance matrix + the direction of a linear segment of the contour



- Combinations of constructions used to form the local affine frames
 - center of gravity + covariance matrix + the direction to an inflection point



- Combinations of constructions used to form the local affine frames
 - center of gravity + covariance matrix + the direction given by the third-order moments of the region



■ Derived from *region outer boundary* (continued)

- Points of curvature inflection (2 constraints)
 - curvature changes from convex to concave or vice-versa
- Straight line segments (1 stable constraint for direction, or 4 for the end-points)
- Higher than 2nd order moments
 - a complex number formed from 3rd order moments

whose phase angle

$$c = \mu_{x^3} + \mu_{xy^2} + i(\mu_{x^2y} + \mu_{y^3})$$

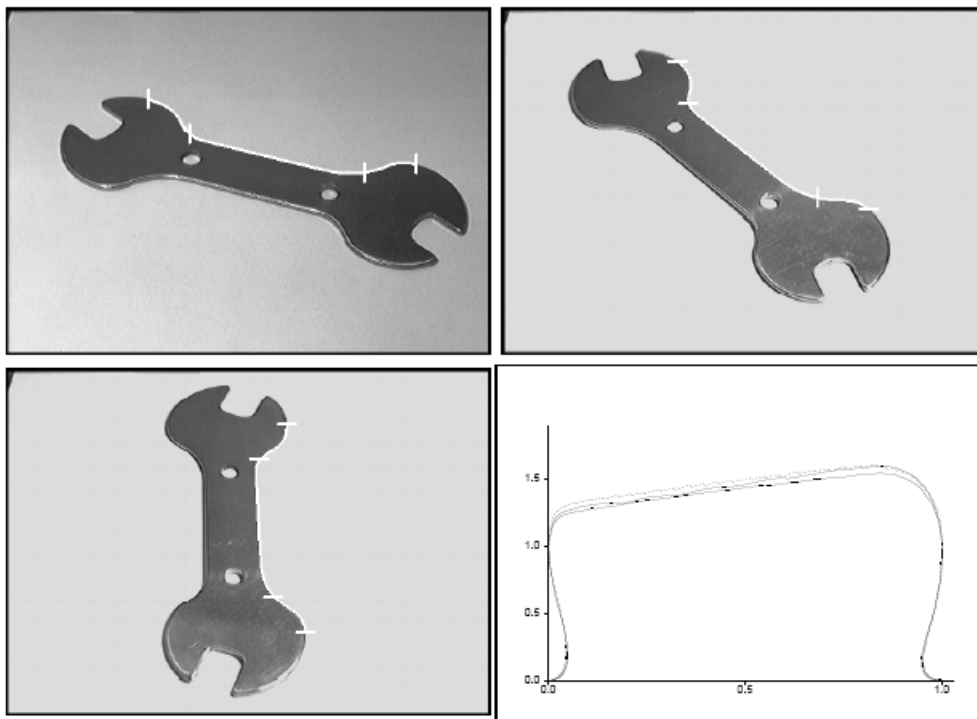
changes covariantly with the region's rotation [Hei04] (1 constraint)

$$\alpha = \tan^{-1}\left(\frac{\mu_{x^2y} + \mu_{y^3}}{\mu_{x^3} + \mu_{xy^2}}\right)$$

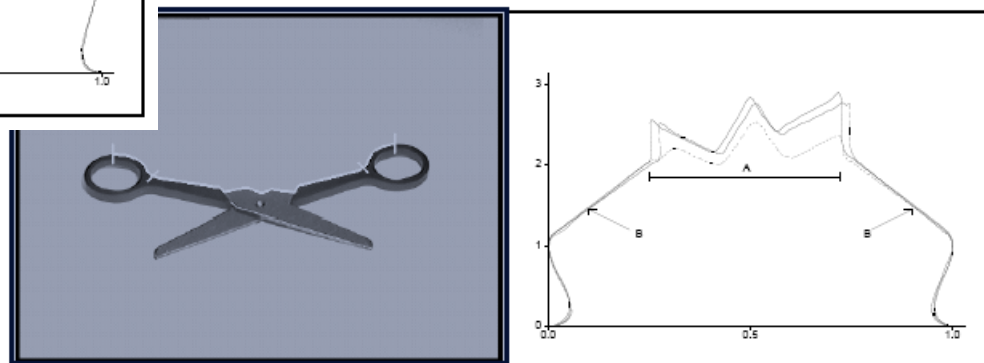
[Hei04] Janne Heikkilä. Pattern matching with affine moment descriptors. *Pattern Recognition*, 37(9):1825–1834, 2004.

Canonical Frames are an old idea ...

Rothwell, Zisserman, Forsyth, Mundy:
Canonical Frames for Planar Object Recognition, 1992



- Multiple reference frames
- Grouping of distinguished points is based on ordering on the segment



Construction of a projective frame

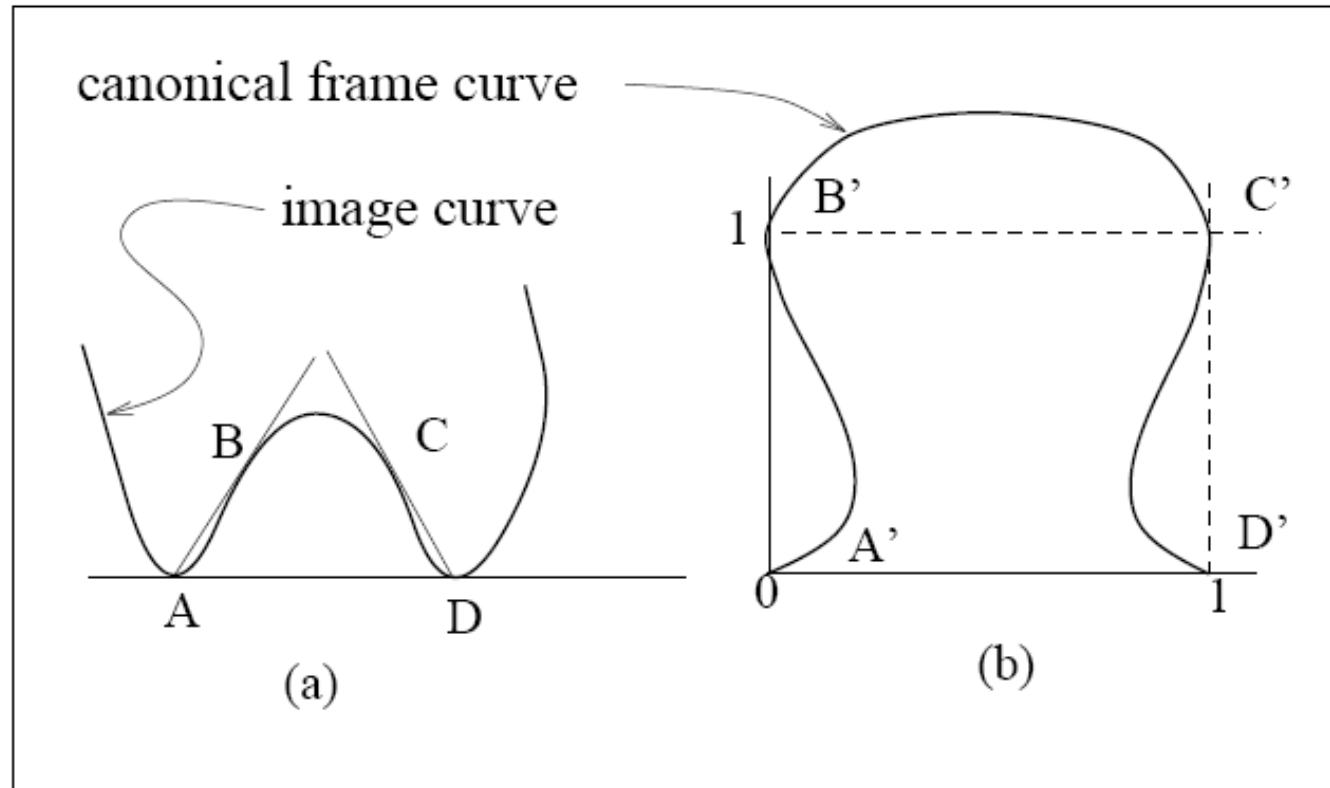


Fig. 1. (a) Construction of the four points necessary to define the canonical frame for a concavity. The first two points ($A D$) are points of bitangency that mark the entrance to the concavity. Two further distinguished points, ($B C$), are obtained from rays cast from the bitangent contact points and tangent to the curve segment within the concavity. These four points are used to map the curve to the canonical frame. (b) Curve in canonical frame. A projection is constructed that transforms the four points in (a) to the corner of the unit square. The same projection transforms the curve into this frame.

occlusion, clutter, multiple objects

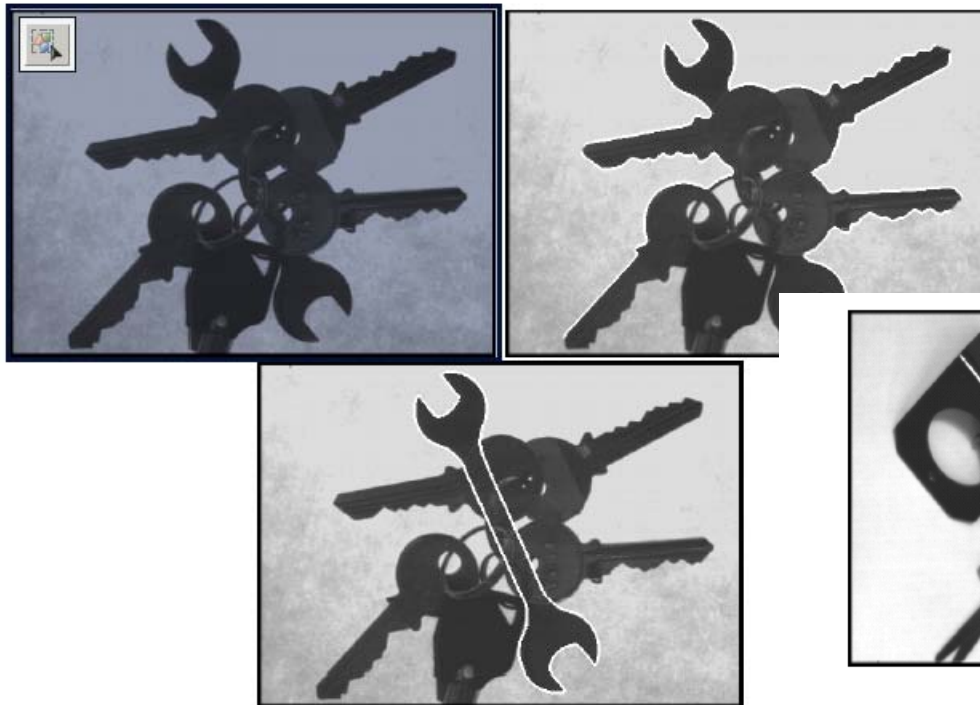


Fig. 6. (a) Spanner almost entirely occluded by keys. The keys are not the li
in this scene. (b) Detected concavities, highlighted in white, which are used
(c) The spanner which is the only model in the scene contained in the librai
the end slot concavity. The projected outline used for verification is highli

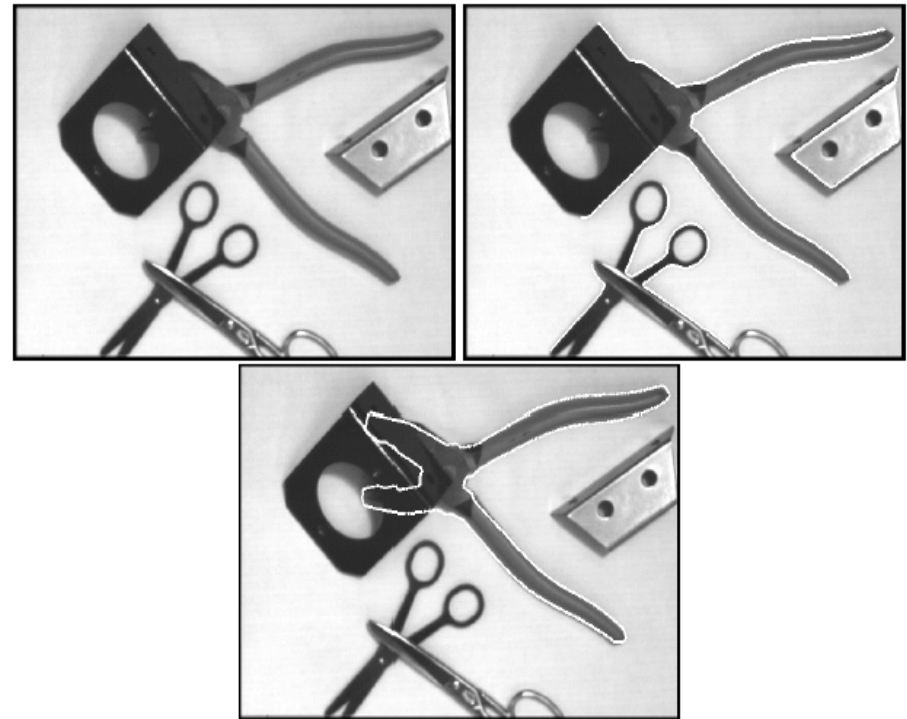
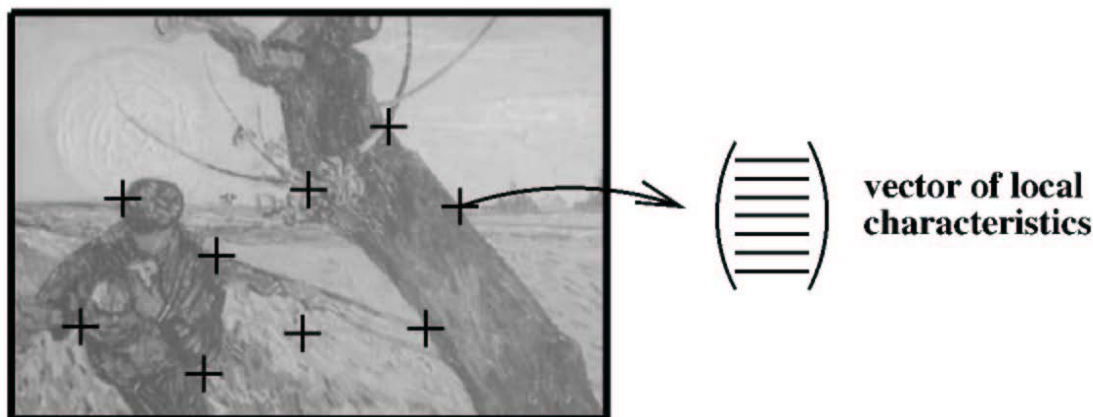


Fig. 7. (a) Image of various planar objects. (b) Concavities, highlighted in white, which are
used to compute indexes (c) The pliers which are the only model in the scene contained in
the library, is recognised and verified by projecting the edgels from an acquisition image, and
checking overlap with edgels in this image.

- 1. Detect affine- (or similarity-) covariant regions (=distinguished regions) = local features**
Yields regions (connected set of pixels) that are detectable with high repeatability over a large range of conditions.
- 2. Description: Invariants or Representation in Canonical Frames**
Representation of local appearance in a Measurement Region (MR). Size of MR has to be chosen as a compromise between discriminability vs. robustness to detector imprecision and image noise.
- 3. Indexing**
For fast (sub-linear) retrieval of potential matches
- 4. Verification of local matches**
- 5. Verification of global geometric arrangement**
Confirms or rejects a candidate match

- Multi-scale differential gray value invariants computed at Harris points
- Scale and rotation invariant
- Feature vectors compared by Mahalanobis distance
- Similarity-based geometric constraint to reject mismatches
- *Canonical Frame not used.*



C. Schmid, R. Mohr, "Local Gray-Value Invariants for Image Retrieval", IEEE Trans. PAMI, vol. 19 (5), 1997, pp. 530--535.

Detector:

- Scale-space peaks of Difference-of-Gaussians filter response (Lindeberg 1995)
- **Similarity frame** from modes of gradient histogram

SIFT Descriptor:

- Local histograms of gradient orientation
- Allows for small misalignments
=> robust to non-similarity transforms

Indexing :

- kD-tree structure

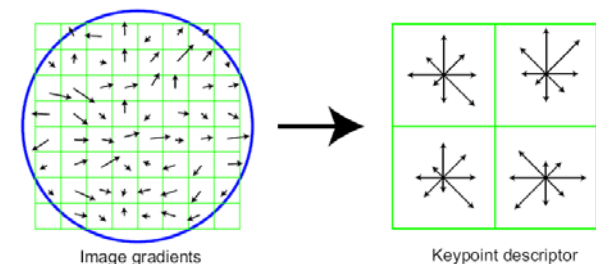
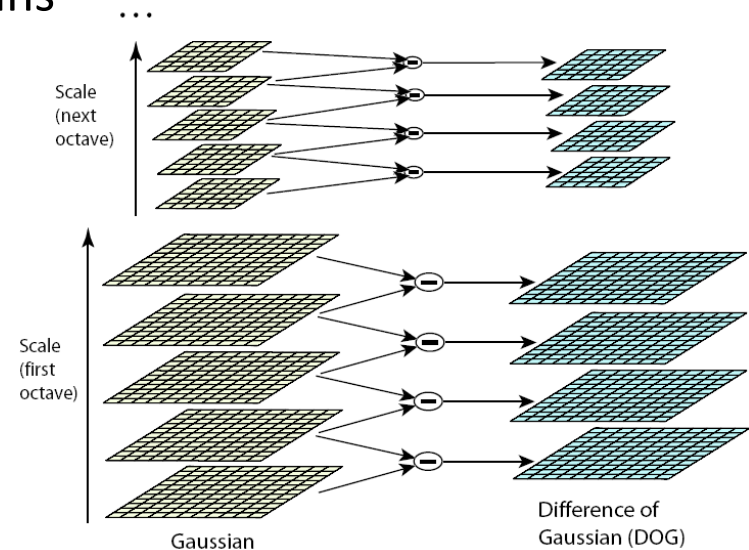
Matching:

- test on euclidean distance of 1st and 2nd match

Verification:

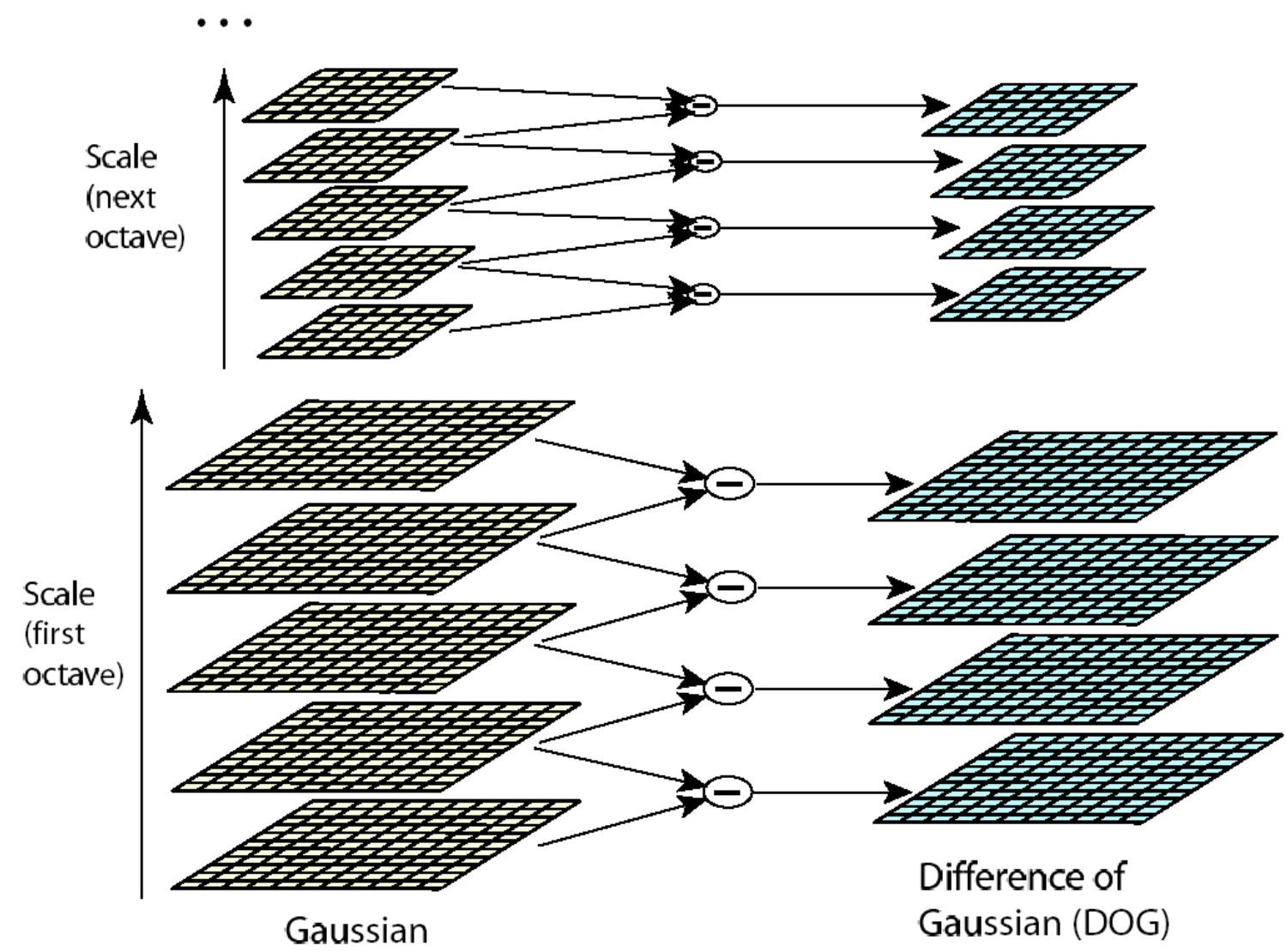
- Hough transform based clustering of correspondences with similar transformations

Fast, efficient implementation, **real-time recognition**



D. G. Lowe: “Distinctive image features from scale-invariant keypoints”. IJCV, 2004.

Scale space processed one octave at a time

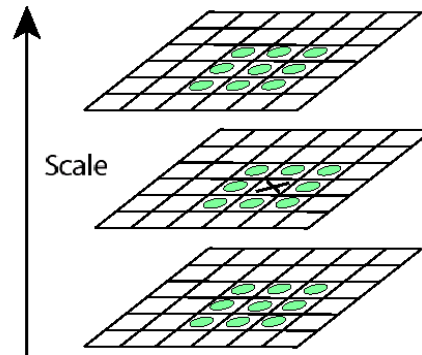


- Detect maxima and minima of difference-of-Gaussian in scale space
- Fit a quadratic to surrounding values for sub-pixel and sub-scale interpolation (Brown & Lowe, 2002)
- Taylor expansion around point:

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

- Offset of extremum (use finite differences for derivatives):

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

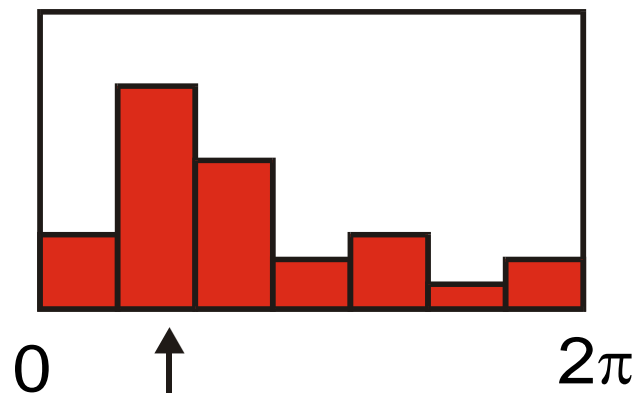
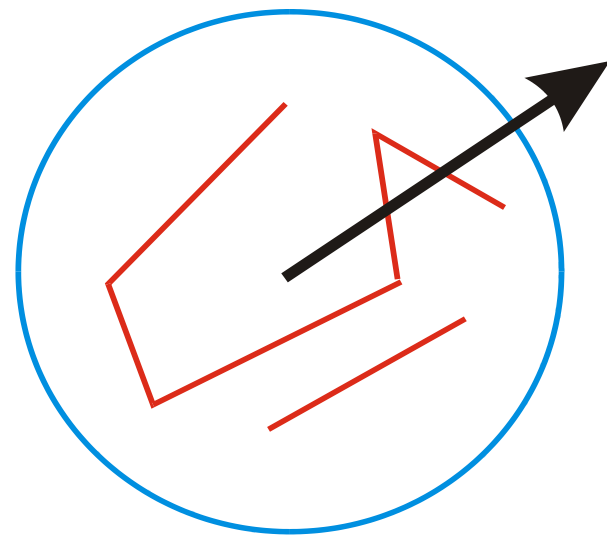


Building a Similarity Frame (s) (my terminology)

Select canonical orientation (s)

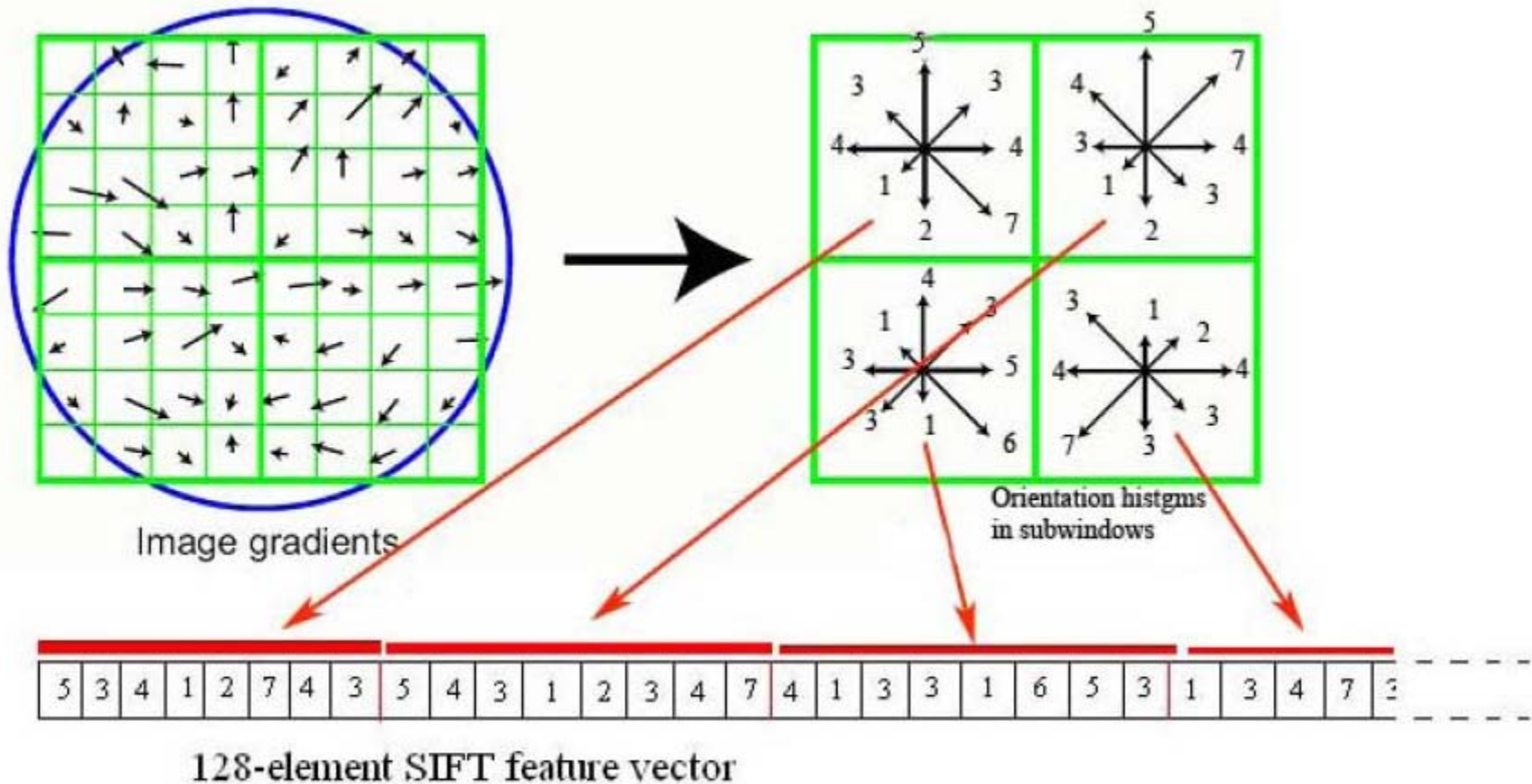
- Compute a histogram of local gradient directions computed at the selected scale
- Assign canonical orientation(s) at peak(s) of smoothed histogram
- $(x, y, \text{scale}) + \text{orientation}$ defines a local *similarity frame*; equivalent to detecting 2 distinguished points

Note: if orientation of the object (image) is known, it may replace this construction



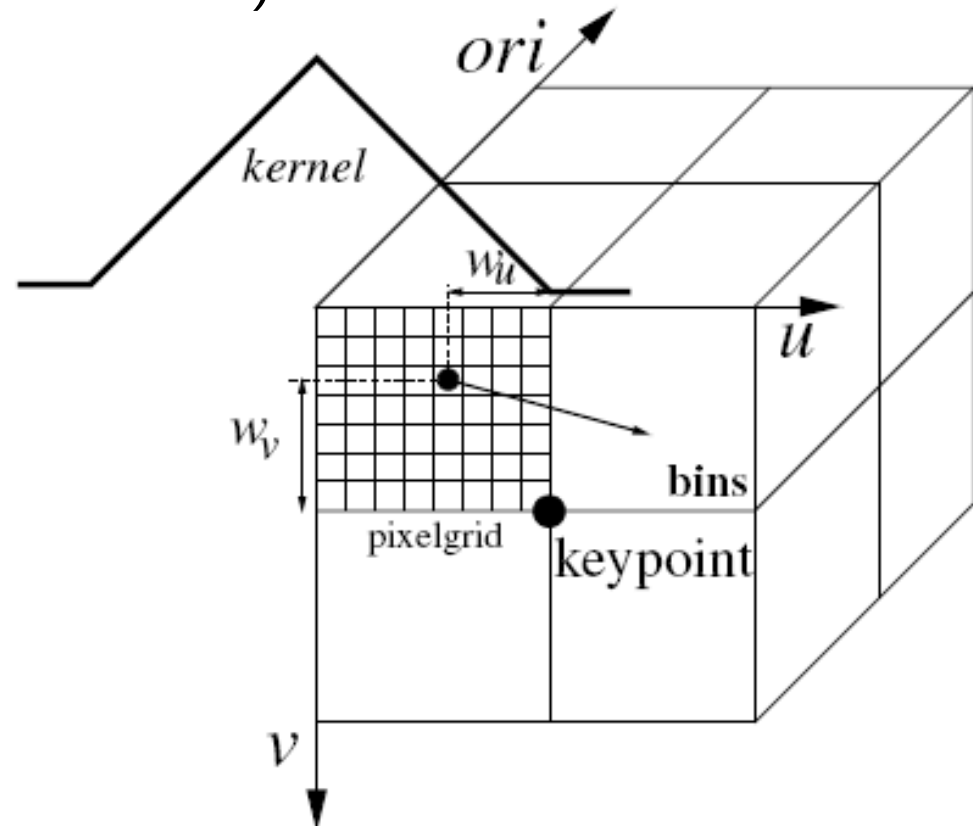
SIFT Descriptor

- A 4x4 histogram lattice of orientation histograms
- Orientations quantized (with interpolation) into 8 bins
- Each bin contains a weighted sum of the norms of the image gradients around its center, with complex normalization



SIFT Descriptor

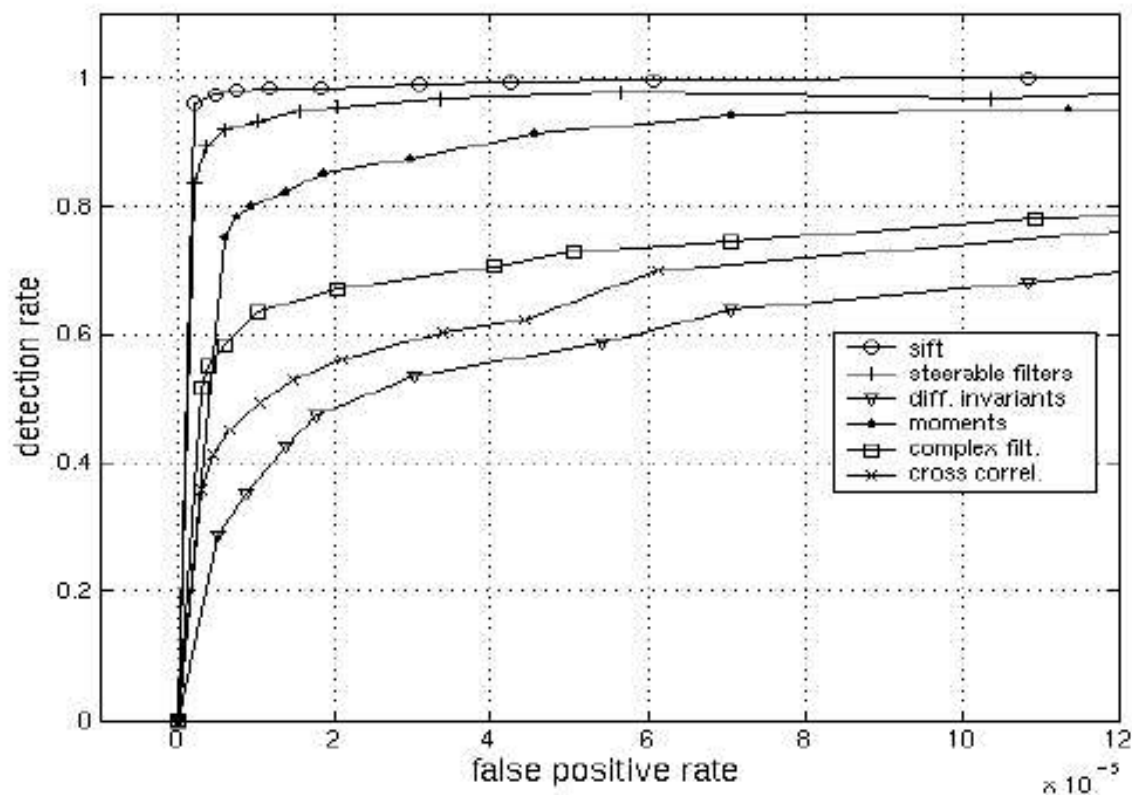
- SIFT descriptor can be viewed as a 3-D histogram in which two dimensions correspond to image spatial dimensions and the additional dimension to the image gradient direction (normally discretised into 8 bins)



SIFT – Scale Invariant Feature Transform¹

- Empirically found² to show very good performance, invariant to *image rotation*, *scale*, *intensity change*, and to moderate *affine* transformations

Scale = 2.5
Rotation = 45°



¹ D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

² K.Mikolajczyk, C.Schmid. "A Performance Evaluation of Local Descriptors". CVPR 2003

- Based on gradient orientations, which are robust to illumination changes
- Spatial binning gives tolerance to small shifts in location and scale, affine change.
- Explicit orientation normalization
- Photometric normalization by making all vectors unit norm
- Orientation histogram gives robustness to small local deformations

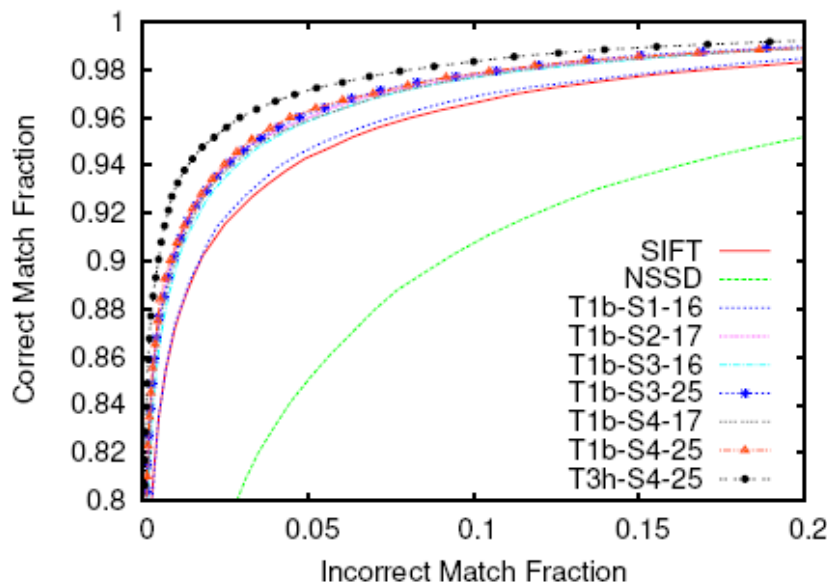
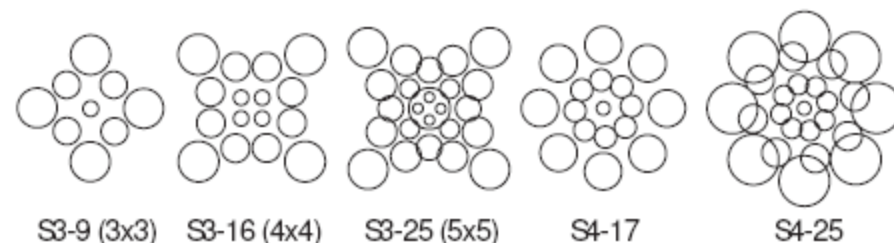
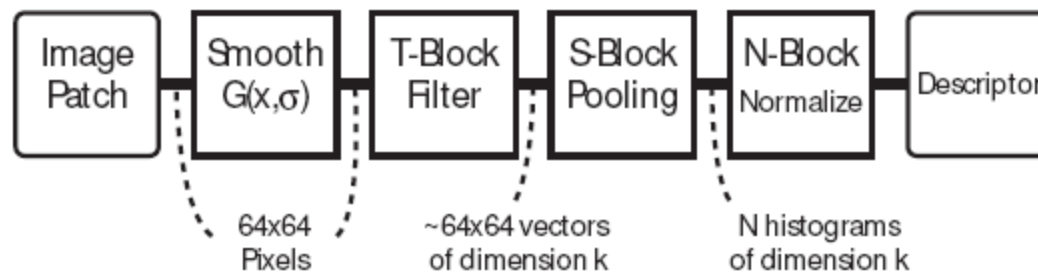
- By far the most commonly used distinguished region descriptor:
 - fast
 - compact
 - **works for a broad class of scenes**
 - source code available
- large number of ad hoc parameters \Rightarrow Enormous follow up literature on both “improvements” and improvements [HoG, Daisy, Cogain]
 - GLOH, HoG: different grid, not 4x4, not necessarily a square
 - Daisy: many parameters optimized

Learning Local Image Descriptors

Simon A. J. Winder

Matthew Brown

Microsoft Research
1 Microsoft Way, Redmond, WA 98052, USA



The best result of all was obtained by combining steerable filters with the polar plan of S4 to give T3h-S4-25. At just under a 2% error rate, this is one third of the error rate produced by SIFT at 95% correct matches. The ROC curve for this descriptor is plotted on Figure 11. However the dimensionality is quite high at 400.

DAISY local image descriptor

- I. Histograms at every pixel location are computed

$$\mathbf{h}_{\Sigma}(u, v) = [\mathbf{G}_1^{\Sigma}(u, v), \dots, \mathbf{G}_8^{\Sigma}(u, v)]^{\top},$$

$\mathbf{h}_{\Sigma}(u, v)$

\mathbf{G}_1^{Σ}

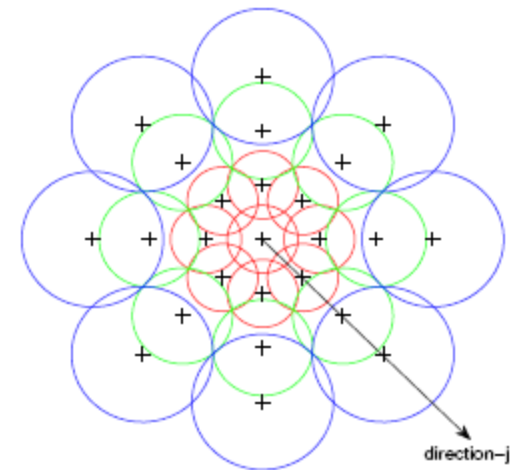
: histogram at location (u, v)

: Gaussian convolved orientation maps

- II. Histograms are normalized to unit norm

- III. as

$$\mathcal{D}(u_0, v_0) = \left[\begin{array}{l} \tilde{\mathbf{h}}_{\Sigma_1}^{\top}(u_0, v_0), \\ \tilde{\mathbf{h}}_{\Sigma_1}^{\top}(\mathbf{l}_1(u_0, v_0, R_1)), \dots, \tilde{\mathbf{h}}_{\Sigma_1}^{\top}(\mathbf{l}_N(u_0, v_0, R_1)), \\ \tilde{\mathbf{h}}_{\Sigma_2}^{\top}(\mathbf{l}_1(u_0, v_0, R_2)), \dots, \tilde{\mathbf{h}}_{\Sigma_2}^{\top}(\mathbf{l}_N(u_0, v_0, R_2)), \\ \tilde{\mathbf{h}}_{\Sigma_3}^{\top}(\mathbf{l}_1(u_0, v_0, R_3)), \dots, \tilde{\mathbf{h}}_{\Sigma_3}^{\top}(\mathbf{l}_N(u_0, v_0, R_3)) \end{array} \right]^{\top}$$



- Convolution is time-efficient for separable kernels like Gaussian
- Convolution maps with larger Gaussian kernel can be built upon convolution maps with smaller Gaussian kernel:

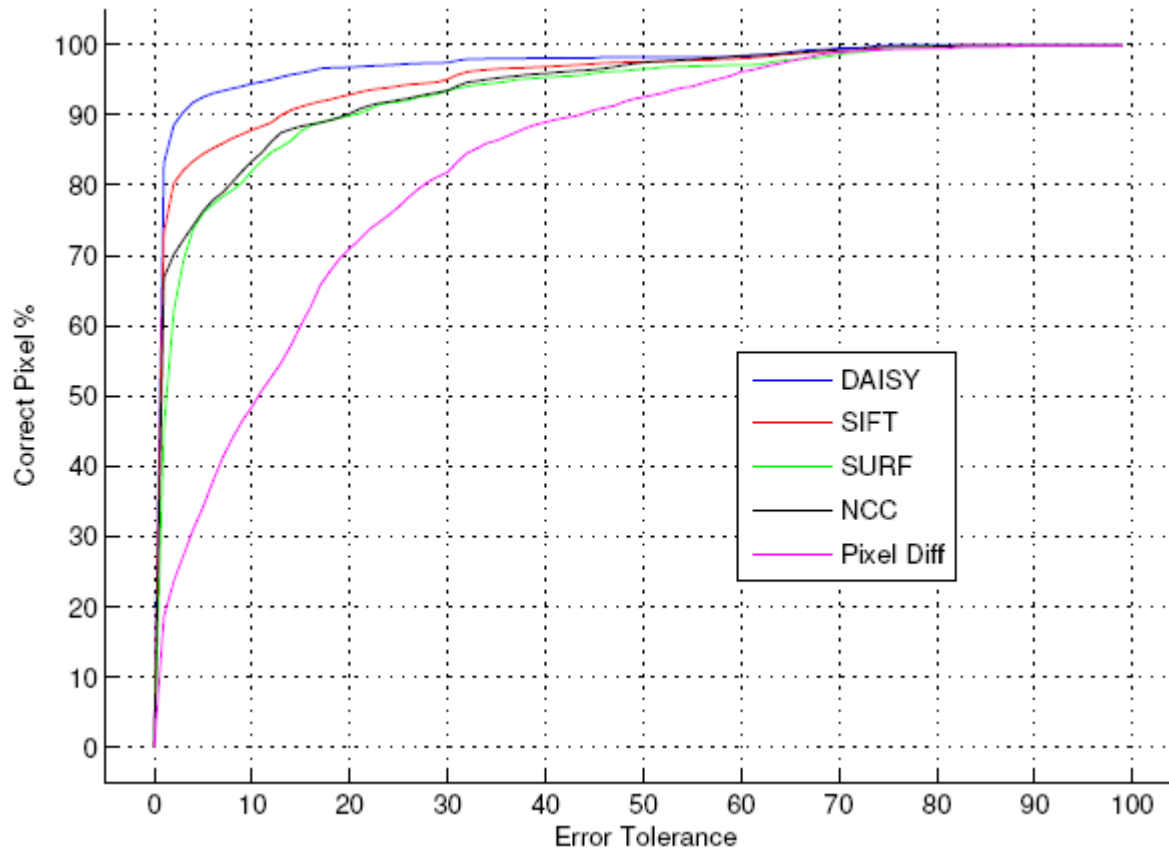
$$\mathbf{G}_o^{\Sigma_2} = G_{\Sigma_2} * \left(\frac{\partial \mathbf{I}}{\partial o} \right)^+ = G_{\Sigma} * G_{\Sigma_1} * \left(\frac{\partial \mathbf{I}}{\partial o} \right)^+ = G_{\Sigma} * \mathbf{G}_o^{\Sigma_1},$$

$$\text{with } \Sigma = \sqrt{\Sigma_2^2 - \Sigma_1^2}.$$

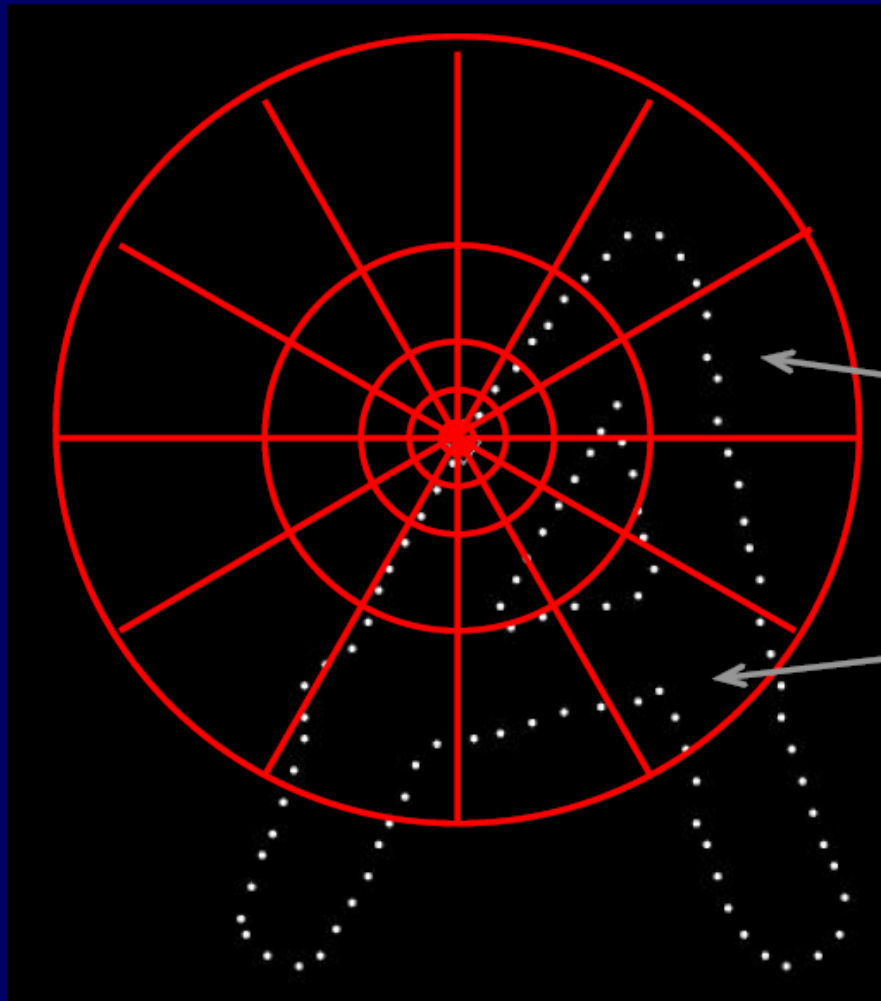
Image Size	DAISY	SIFT
800x600	5	252
1024x768	10	432
1290x960	13	651

Table 1. Computation Time Comparison (in seconds)

Results



Shape Context



Count the number of points
inside each bin, e.g.:

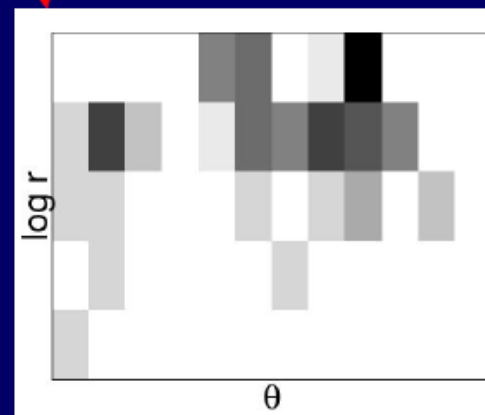
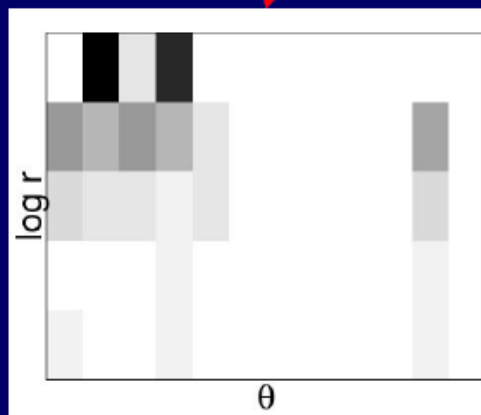
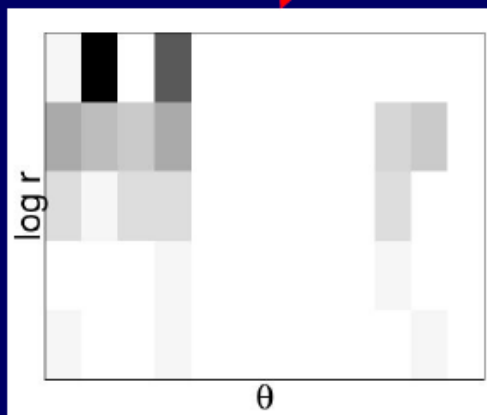
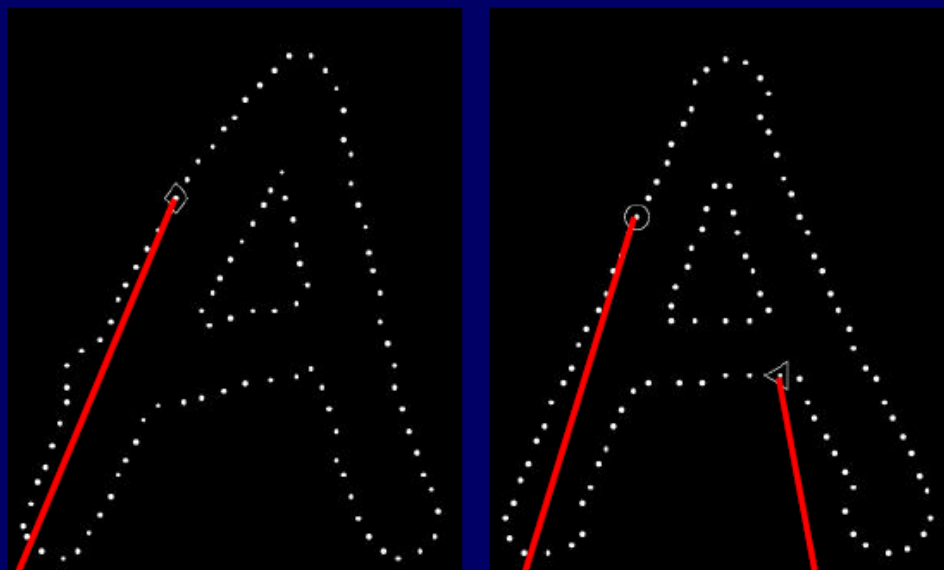
Count = 4

⋮

Count = 10

☞ Compact representation
of distribution of points
relative to each point

Shape Context



Detector:

- Scale-space peaks of Difference-of-Gaussians filter response (Lindeberg 1995)
- **Similarity frame** from modes of gradient histogram

SIFT Descriptor:

- Local histograms of gradient orientation
- Allows for small misalignments
=> robust to non-similarity transforms

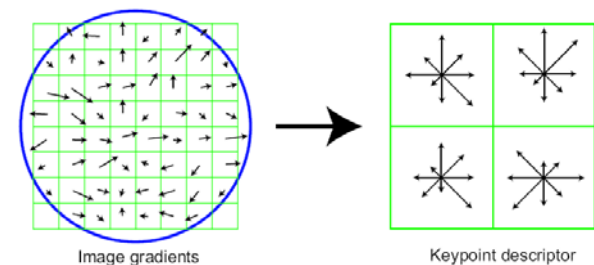
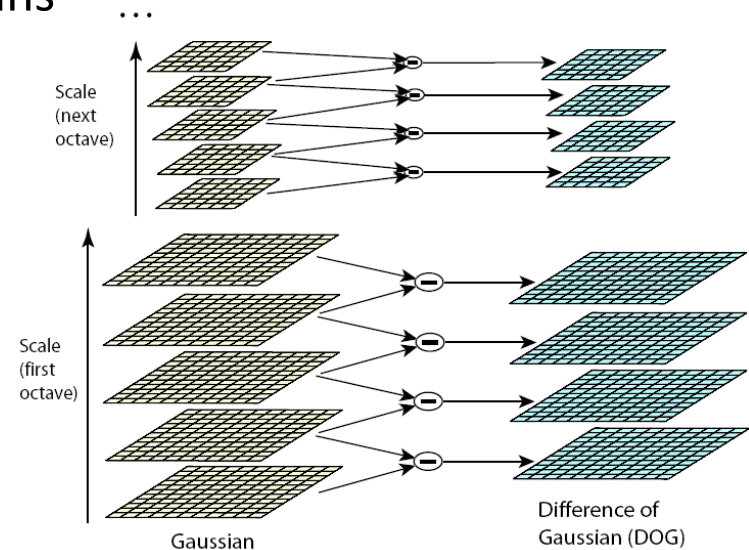
Indexing:

- Modified kD-tree structure

Verification:

- Hough transform based clustering of correspondences with similar transformations

Fast, efficient implementation, **real-time recognition**



D. G. Lowe: “Distinctive image features from scale-invariant keypoints”. IJCV, 2004.

Nearest-neighbor matching

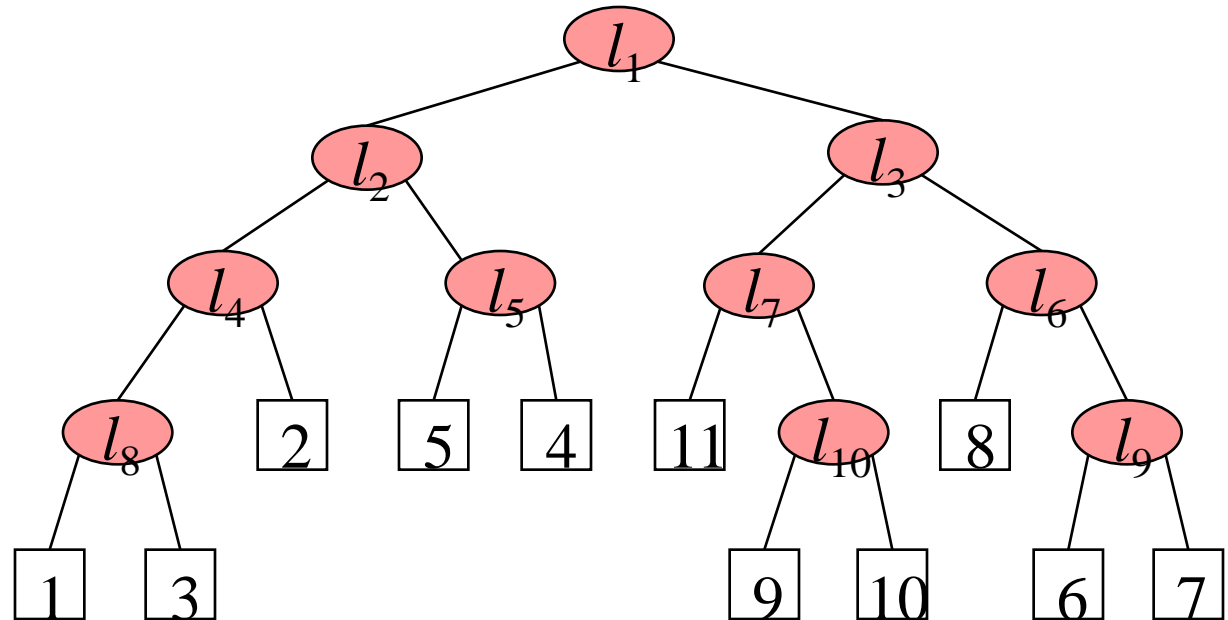
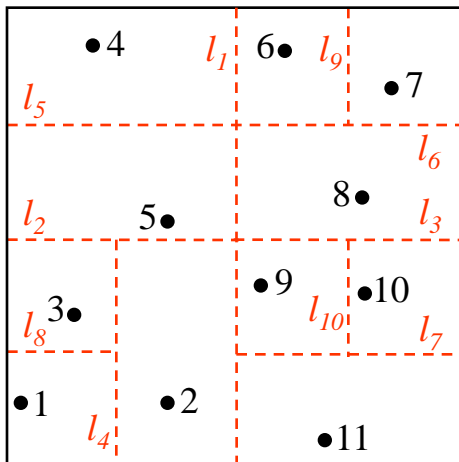
- Solve following problem for all feature vectors, \mathbf{x} :

$$\forall j \text{ } NN(j) = \arg \min_i ||\mathbf{x}_i - \mathbf{x}_j||, i \neq j$$

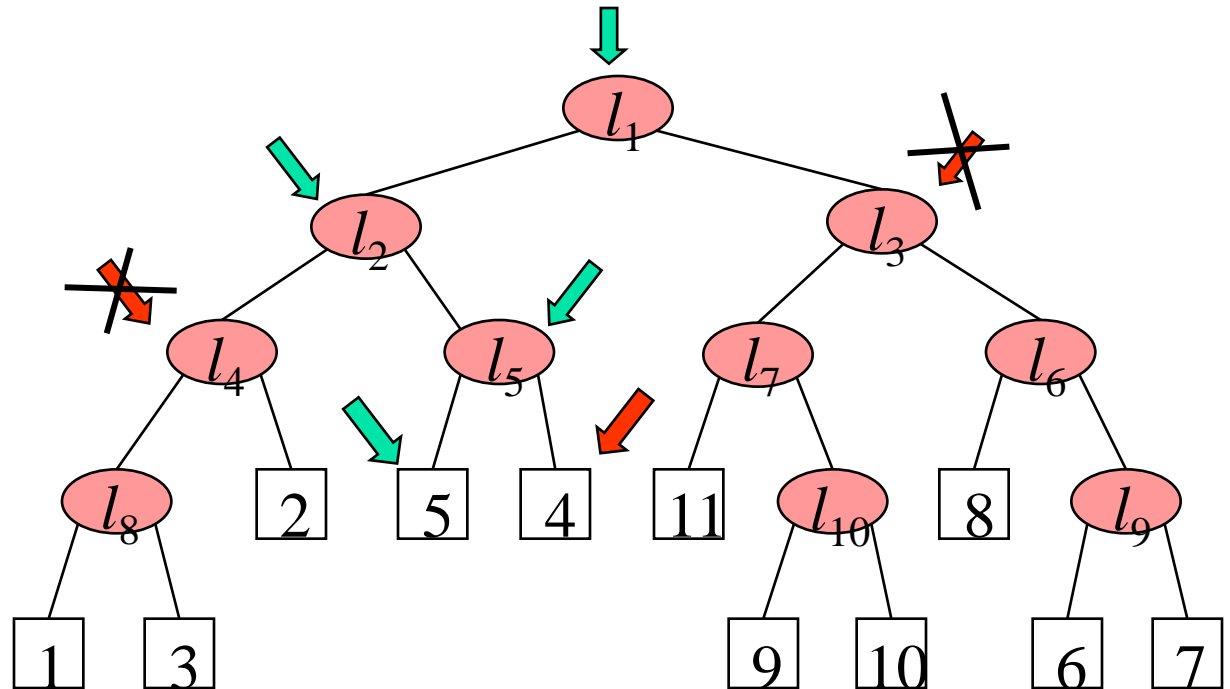
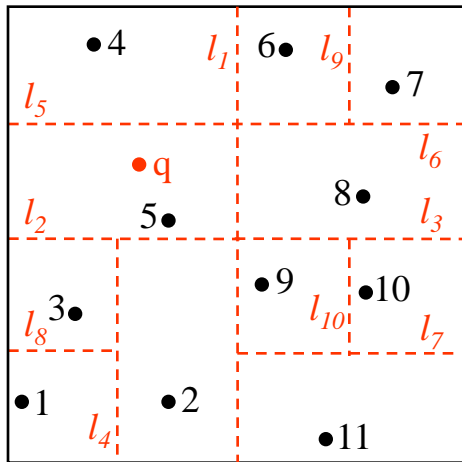
- Nearest-neighbor matching is the major computational bottleneck
 - Linear search performs dn^2 operations for n features and d dimensions
 - No exact methods are faster than linear search for $d > 10$ (?)
 - Approximate methods can be much faster, but at the cost of missing some correct matches. Failure rate gets worse for large datasets.

K-d tree construction

Simple 2D example

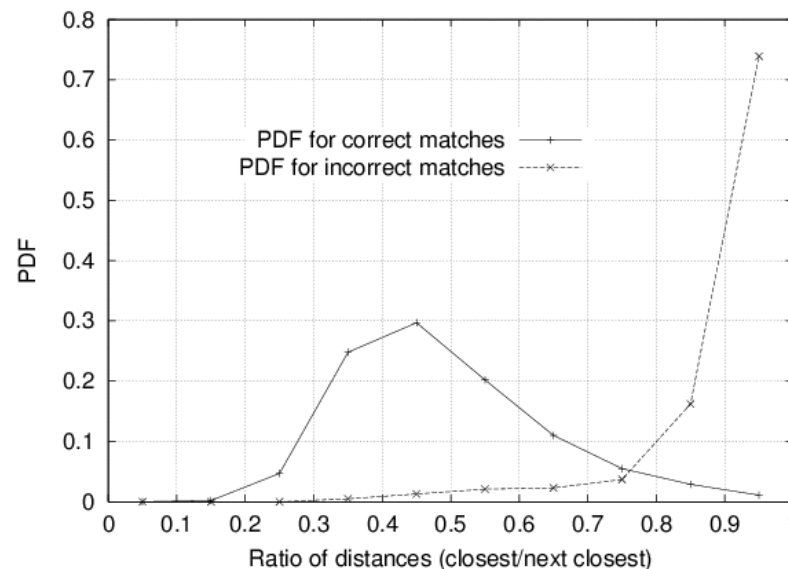


K-d tree query



Feature space outlier rejection

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

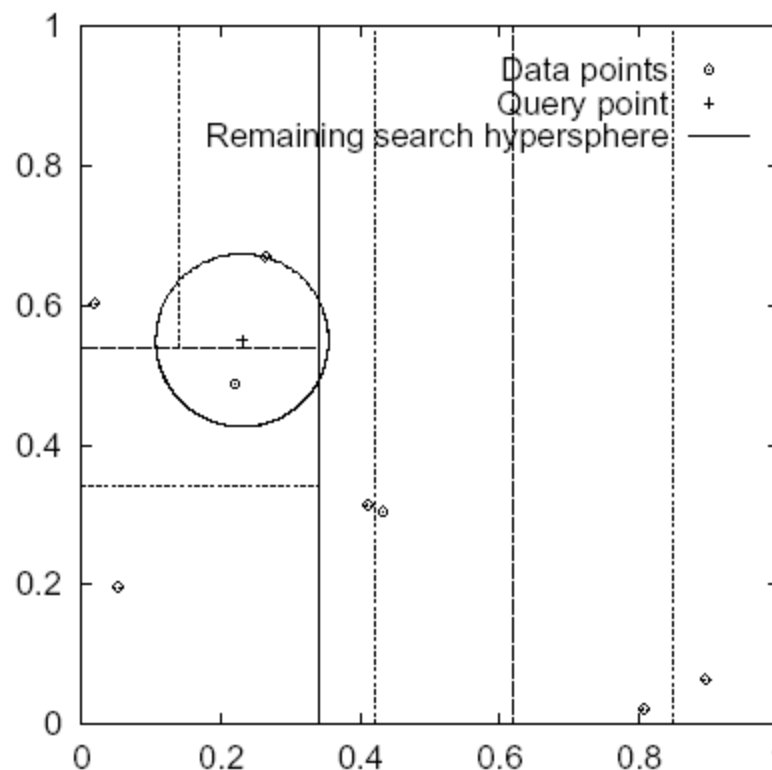


David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

Approximate k-d tree matching

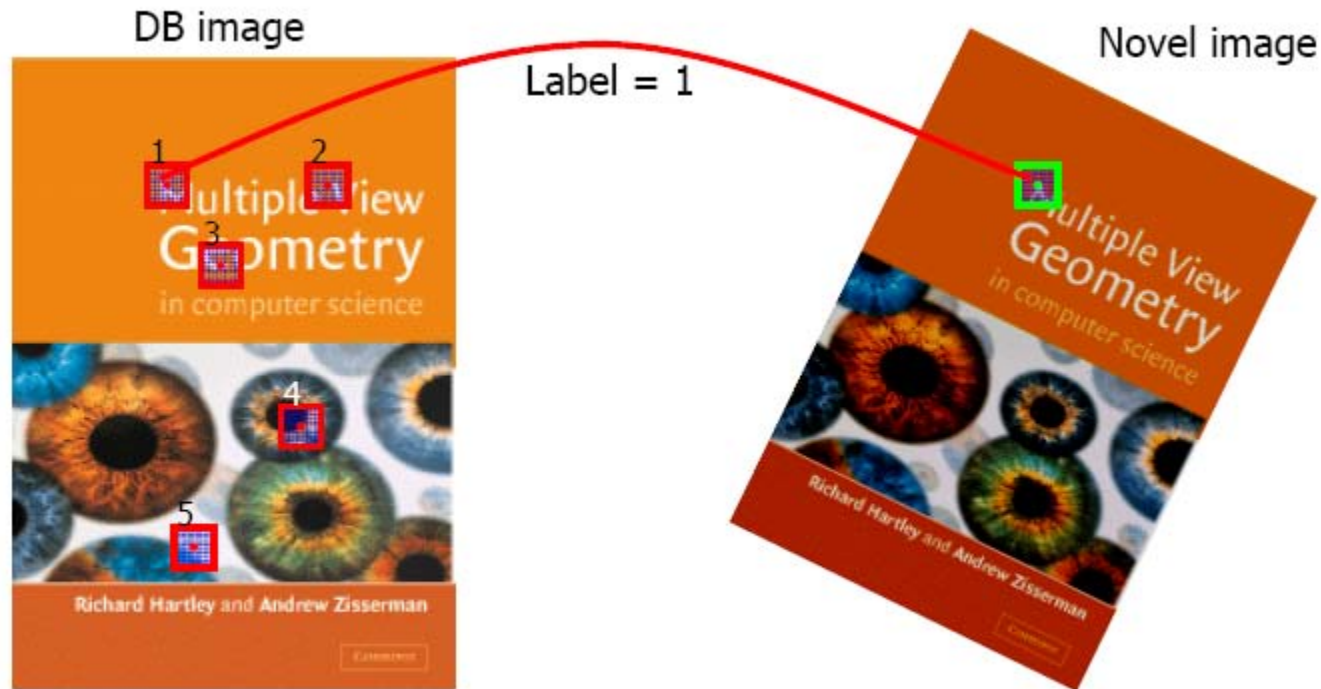
Key idea:

- n Search k-d tree bins in order of distance from query
- n Requires use of a priority queue
- n Copes better with high dimensionality
- n Many different varieties
 - n Ball tree, Spill tree etc.



Randomized Forests

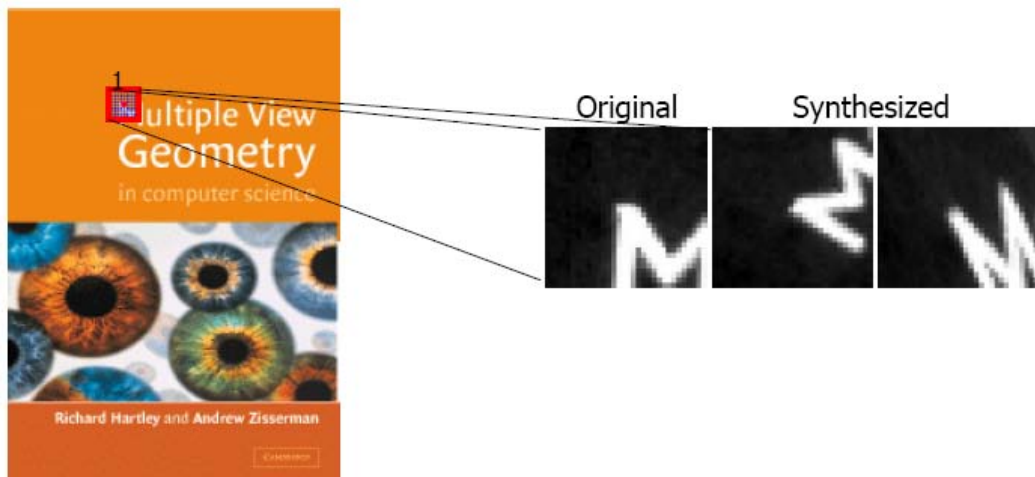
- Feature matching as a classification problem



Lepetit, Laguerre and Fua. Randomized Trees for Real-Time Keypoint Matching, CVPR 2005

Synthesize training examples

■ Planar object



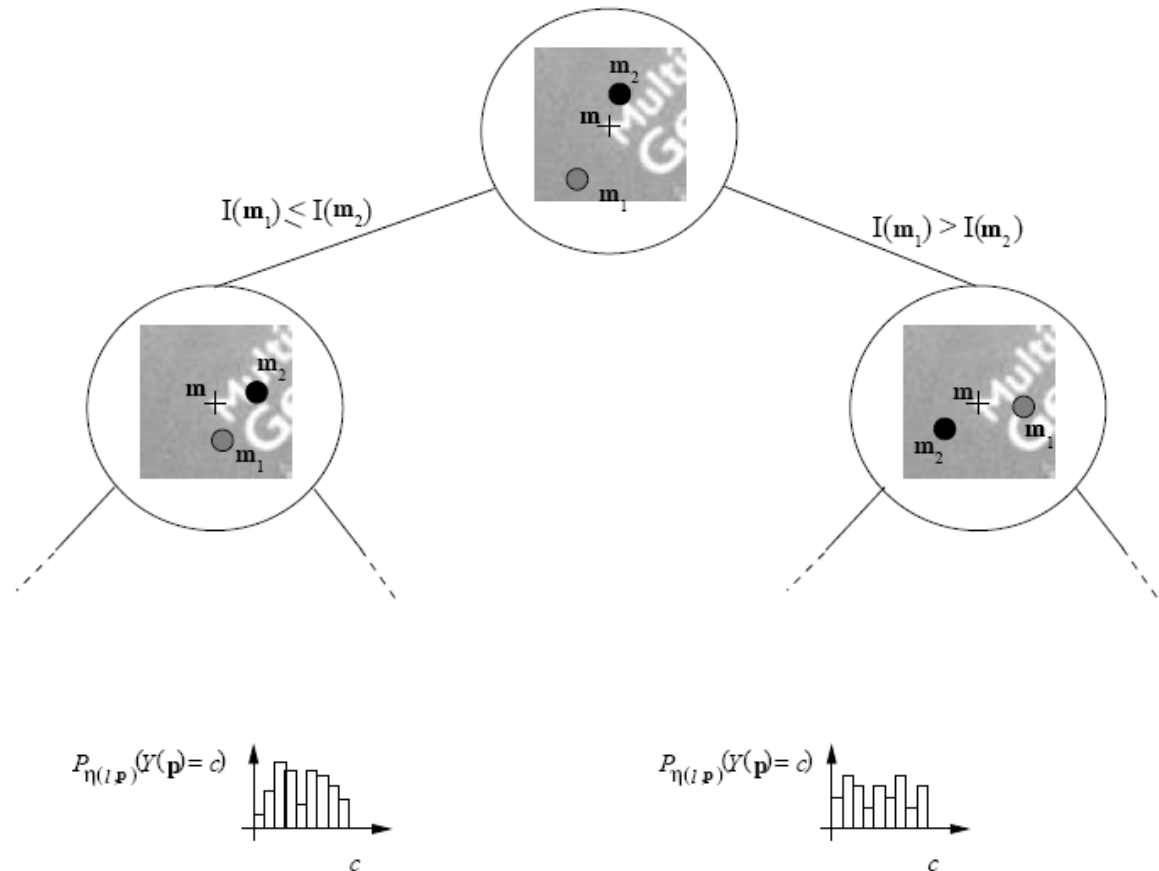
3-D object



Lepetit, Laguerre and Fua. Randomized Trees for Real-Time Keypoint Matching, CVPR 2005

Randomized Decision Tree

- Compare intensity of pairs of pixels
 - In construction, pick pairs randomly
-
- Insert all training examples into tree
 - Distribution at leaves is descriptor for the particular feature

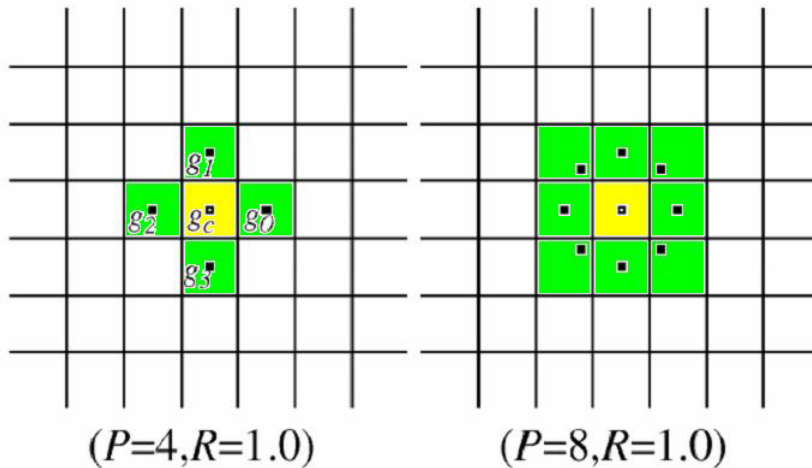


- Use multiple trees (i.e. forest) to improve performance
- Very quick to compute in testing
 - Just comparison of pairs of pixels
 - Real-time performance
- ~10x faster than SIFT, but slightly inferior performance

Local Binary Pattern (LBP) Descriptor

The primitive LBP (P,R) number that characterizes the spatial structure of the local image texture is defined as:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(x) 2^p, \quad x = g_p - g_c \quad \text{where,} \quad s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$



2^7	2^0	2^1
2^6	g_c	2^2
2^5	2^4	2^3

Circularly symmetric neighbor sets (P: angular resolution, R: spatial resolution)

LBP values in a 3 x 3 block

The LBP descriptor is invariant to any monotonic transformation of image

slide credit: Sara Arasteh et al.

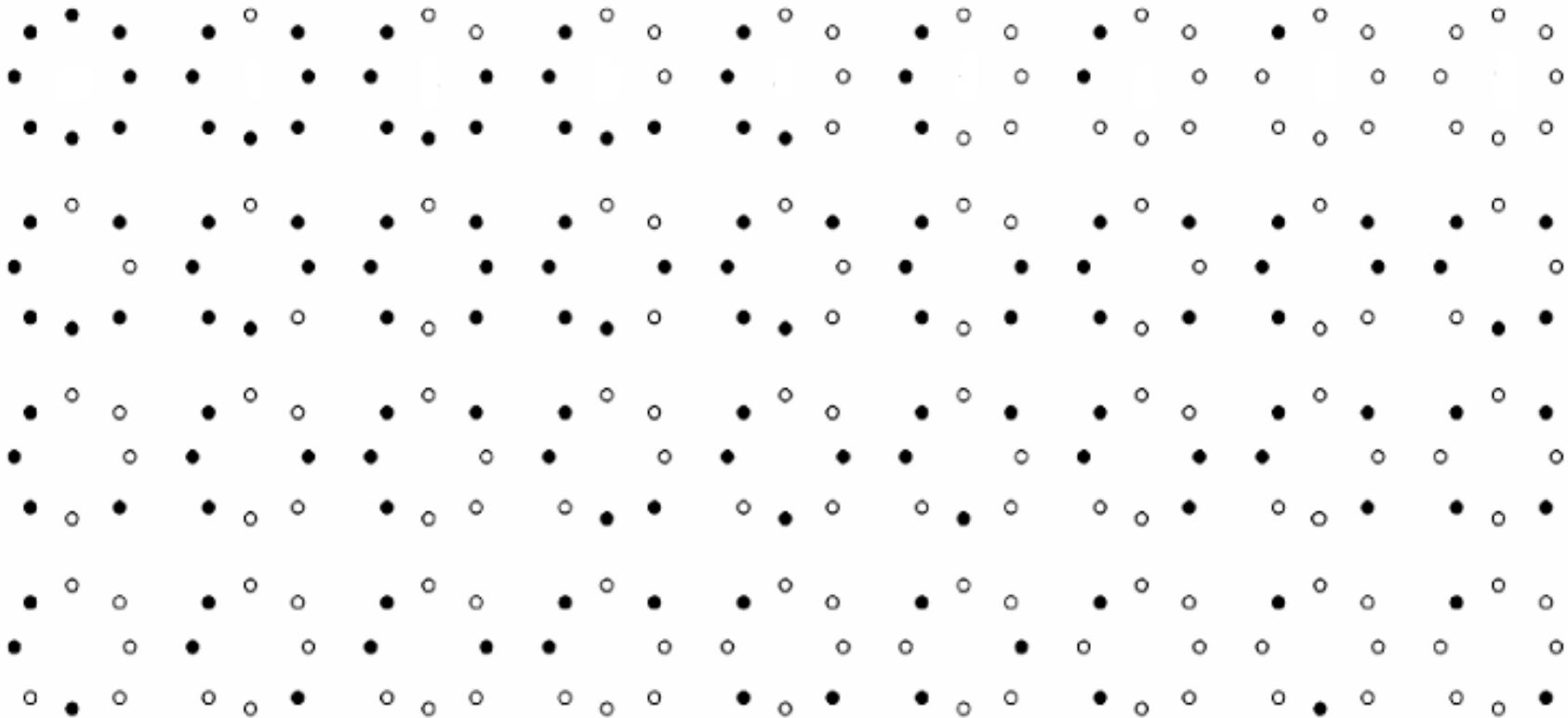
- In order to remove the effect of rotation and assign a unique identifier to each, Rotation Invariant Local Binary Pattern is defined as:

$$LBP_{P,R}^{ri} = \min \left\{ ROR(LBP_{P,R}, i) \mid i = 0, 1, \dots, P-1 \right\}$$

where $ROR(x, i)$ performs a circular bit-wise right shift on P-bit number x , i time.

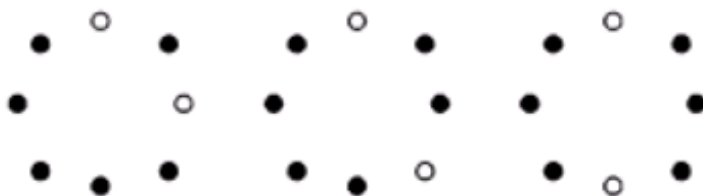
- 36 unique rotation invariant binary patterns can occur in the circularly symmetric neighbor set of $LBP_{8,1}$.

- This figure shows 36 unique rotation invariant binary patterns.

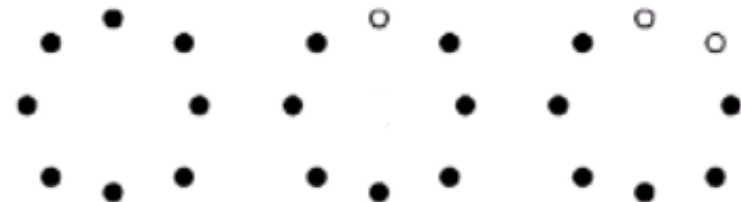


■ Rotation Invariant LBP patterns include:

- Uniform patterns
 - At most two transitions from 0 to 1
- Non-uniform patterns
 - More than two transitions from 0 to 1



Samples of non-uniform
patterns



Samples of uniform
patterns

Uniform LBP (ULBP)

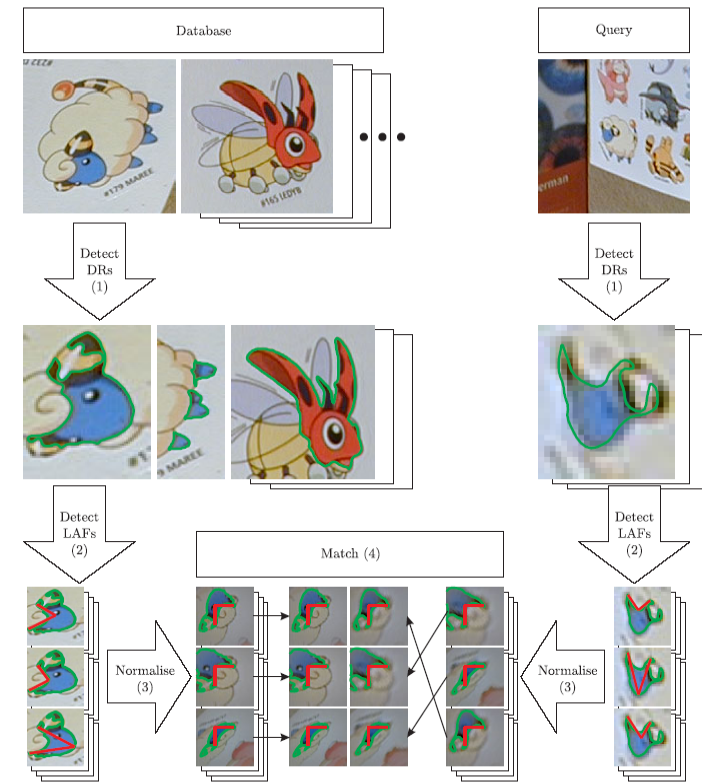
- It is observed that the uniform patterns are the majority, sometimes over 90 percent, of all 3 x 3 neighborhood pixels present in the observed textures.
- They function as templates for microstructures such as :
 - Bright spot (0)
 - Flat area or dark spot (8)
 - Edges of varying positive and negative curvature (1-7)



Uniform Local Binary Patterns

LBP's are popular, numerous modifications exist

1. **Detect Distinguished Regions** Maximally Stable Extremal Regions (MSERs)
2. Construct **Local Affine Frames (LAFs)** (local coordinate frames)
3. **Geometrically normalize** some measurement region (MR) expressed in LAF coordinates
4. **Photometrically normalize** measurements inside MR, compute some derived description
5. Establish local (tentative) correspondences by the **decision-measurement tree method**
6. Verify global geometry (e.g. by RANSAC, geometric hashing, Hough transform.)



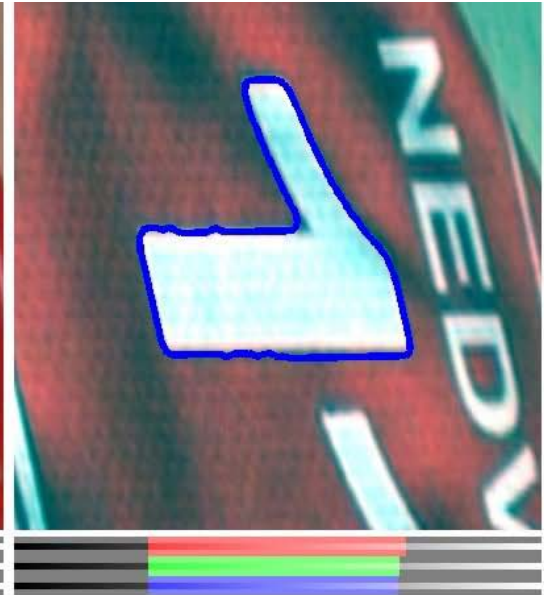
Matas, Chum, Urban, Pajdla: “Robust wide baseline stereo from maximally stable extremal regions”. BMVC2002

Obdržálek and Matas: “Object recognition using local affine frames on distinguished regions”. BMVC02

Obdržálek and Matas: “Sub-linear Indexing for Large Scale Object Recognition”, BMVC 2005

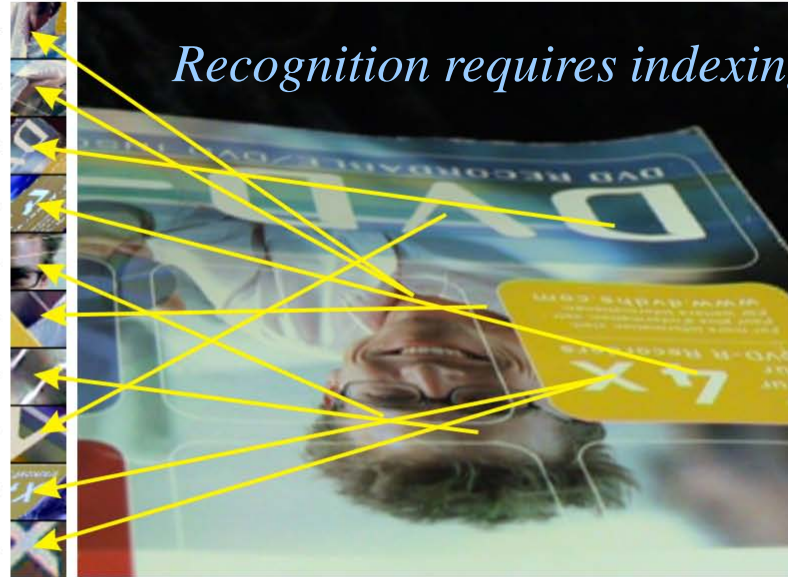
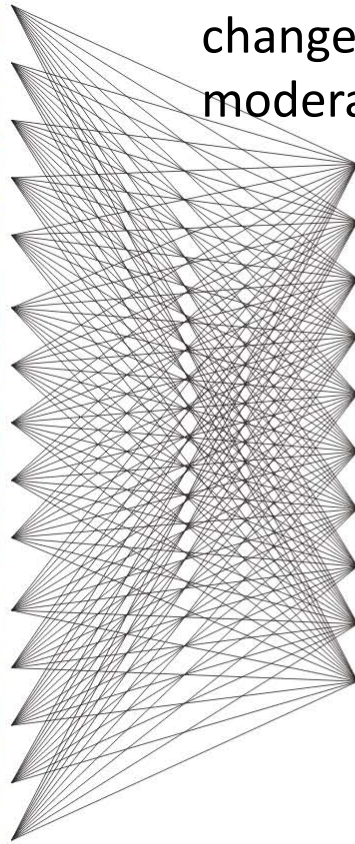
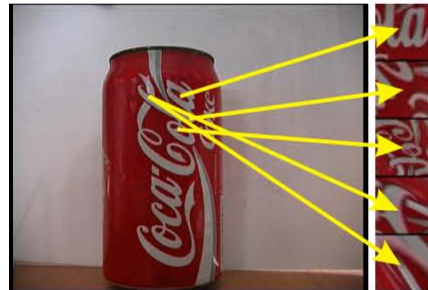
4. **Photometrically normalize** measurements inside MR,
compute some derived description

[video1](#), [video2](#)



“Recognition” as a Sequence of Wide-Baseline Matching Problems ??

Properties: robust to occlusion, clutter, handles pose change, illumination but becomes unrealistic even for moderate number of objects.

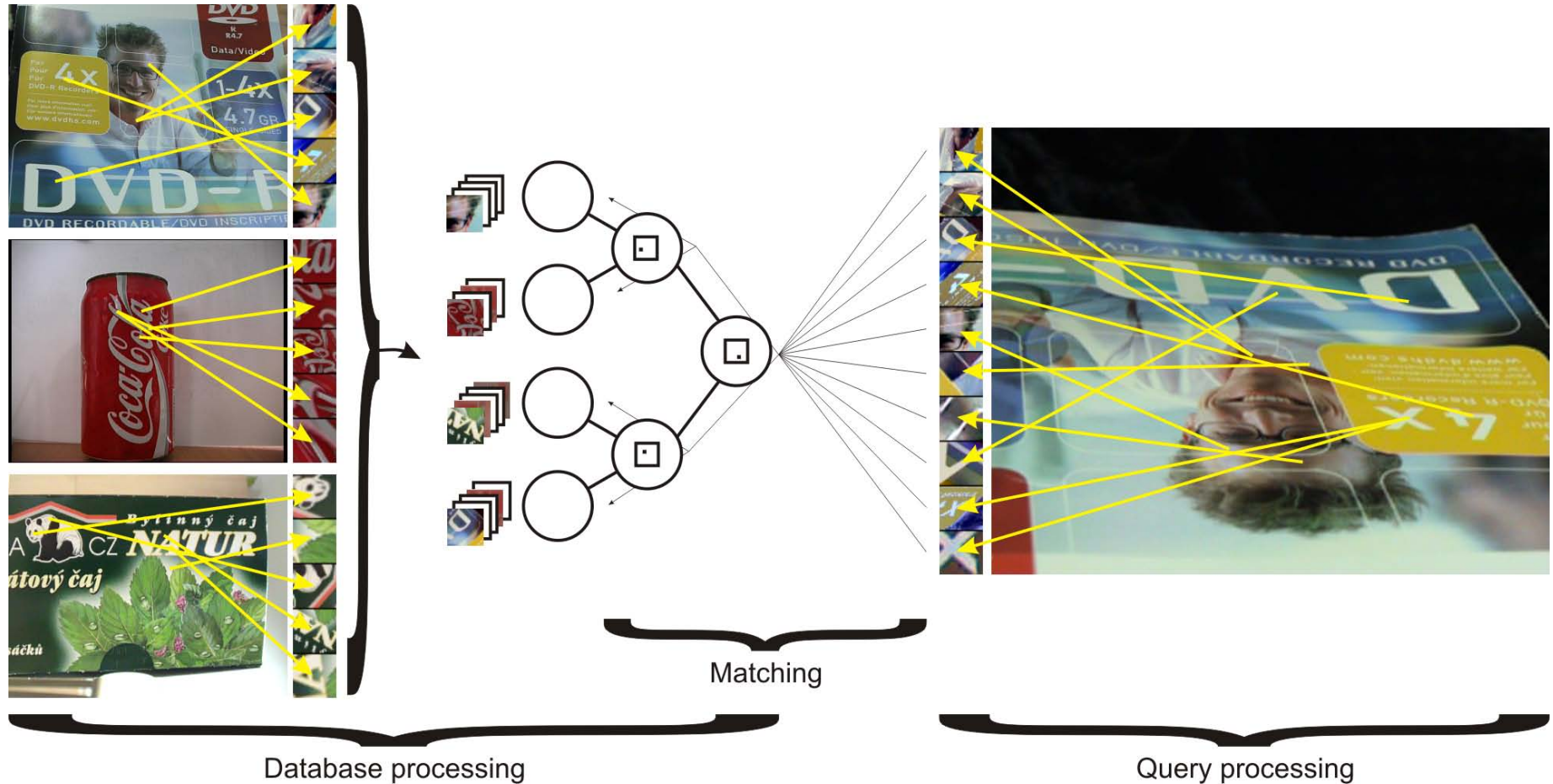


Database processing

Matching

Query processing

Simultaneous Recognition of Multiple Objects Using the Decision-Measurement Tree



Performance Evaluation 1.:Image Retrieval from ZuBuD[1]

- Publicly available dataset ZuBuD
- Database: 201 buildings, each represented by 5 images, more than 1000 images in the DB
- Queries: 115 new images
- Forced match

Recognition rates (rank 1 correct):

- Repeated LAF-MSER matching:
100% @ 27 seconds /retrieval
- Tree matching:
93% @ 0.014 seconds
99% @ 0.510 seconds

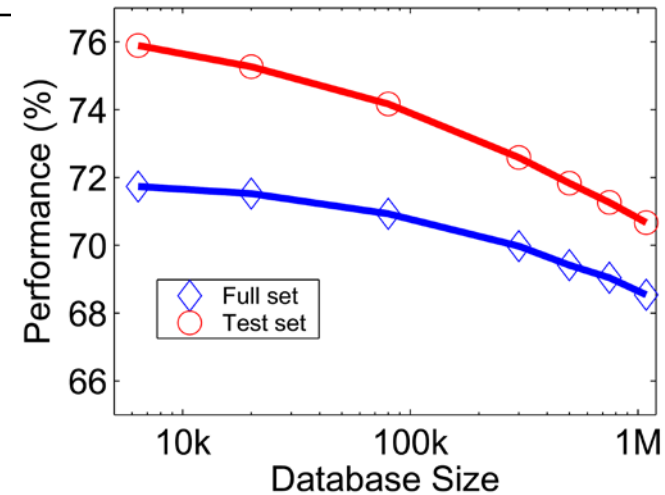


[1] Shao, Svoboda, Tuytelaars, Gool: “HPAT indexing for fast object/scene retrieval”, CIVR2004

Example 2: D. Nistér, H. Stewénus.

Scalable Recognition with a Vocabulary Tree, CVPR 2006

- MSER detector, SIFT descriptor, K-means tree
- Very carefully implemented
- Evaluated on large databases
 - Indexing with up to 1M images
- Online recognition for database of **50,000 CD covers**
 - Retrieval in ~1s



However:

- Recognition of images, not objects
- Some of the object have no chance of being recognized via MSER+SIFT on different background

ImageSearch at the VizCentre

New query: Browse... Send File

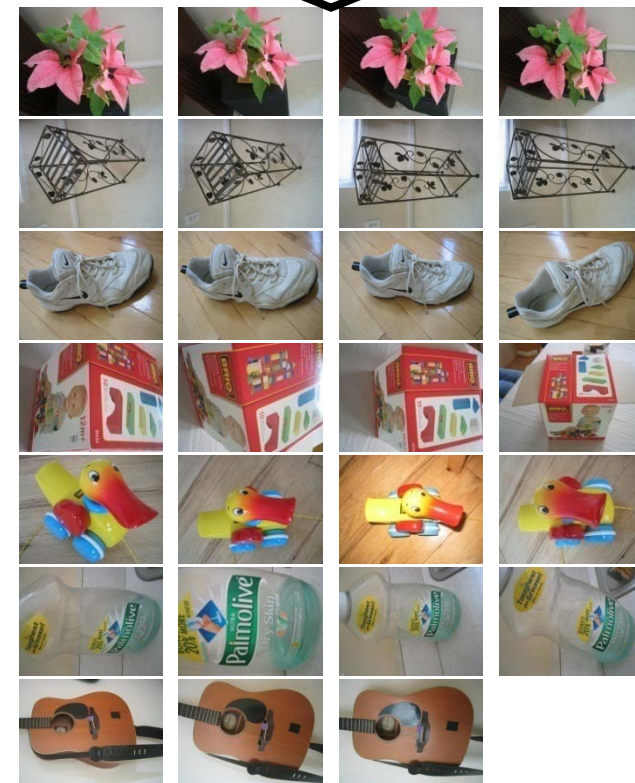
File is 500x320



Top n results of your query.



bourne/im1000043322.pgm bourne/im1000043323.pgm bourne/im1000043326.pgm bourne/im1000043327.pgm



Local Features : Application Examples

- Detection of goods in tray at supermarket checkout
- Database: 500 objects, 6 images each



- ◆ Queries: images captured from a camera at the checkout



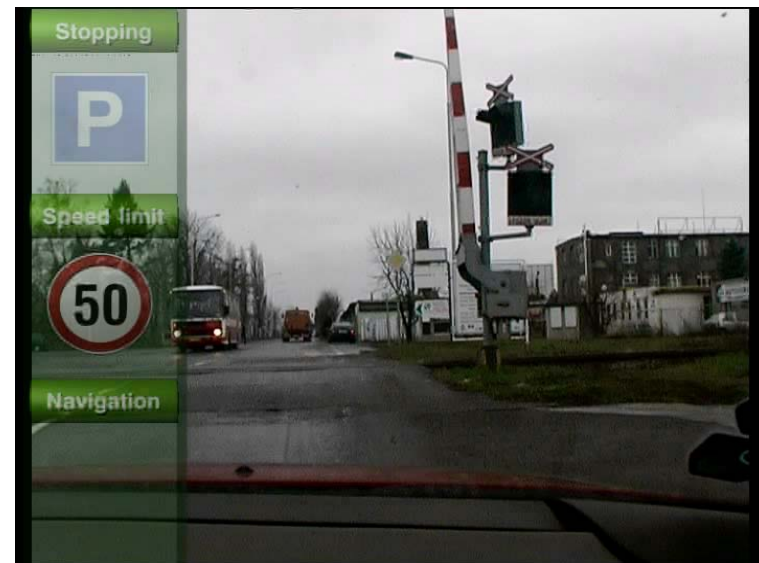
- ◆ Output: list of objects identified in the tray

Local Features : Application Examples

- Traffic sign recognition from a moving car
- Database: images of known signs



- ◆ Output: identification of signs in images taken by an in-car camera (scene-interpretation is not part of the system)



- Detection of product logos in scanned commercials



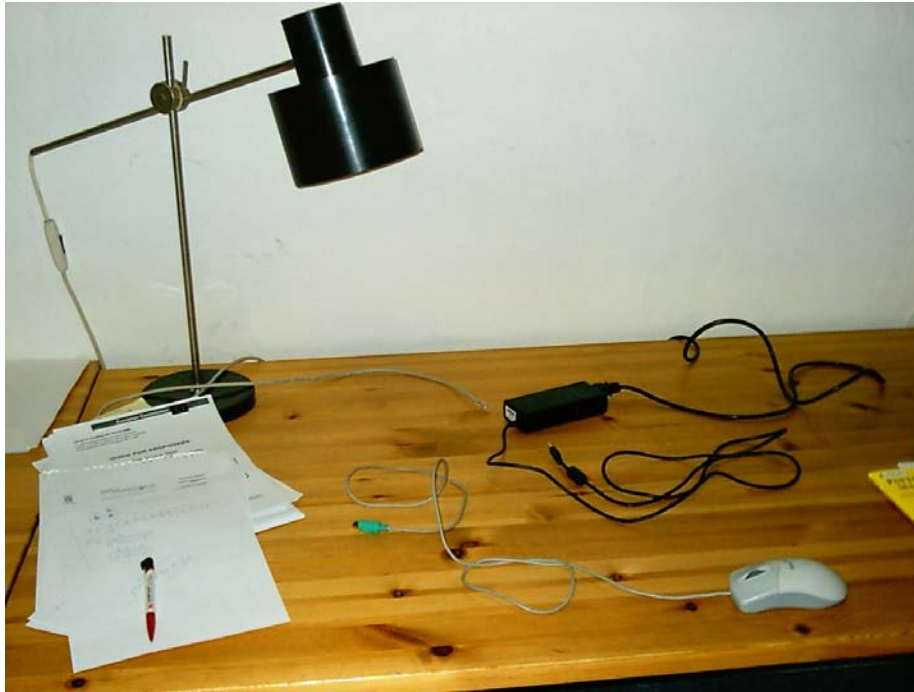
- ◆ Detection of advertising side-boards in TV coverage of sport events.
“For how long was my commercial actually broadcasted?”



- ◆ Detection of company logos in automatic fax processing

1. Methods work well for a non-negligible class of objects, that are locally approximately planar, compact and have surface markings or where 3D effects are negligible (e.g. stitching photographs taken from a similar viewpoint)
2. They are *correspondence based methods*
 - insensitive to occlusion, background clutter
 - very fast
 - handles very large dataset
 - model-building is automatic
3. **The space of problems and object where it does not work is HUGE (examples are all around us).**

Where Local Features Fail:



Challenge: Elongated, Wirey and Flexible Objects

In this case: “no recognition without segmentation”?

Where Local Features Fail:



Camouflage: No distinguished regions !
Very few animals can afford to be distinguishable



Thank you for your attention.

