Local Feature Extraction and Description for

Wide-Baseline Matching, Image Retrieval, Stitching and more ...

Jiří Matas,

Center for Machine Perception, Czech Technical University Prague

Includes slides by:

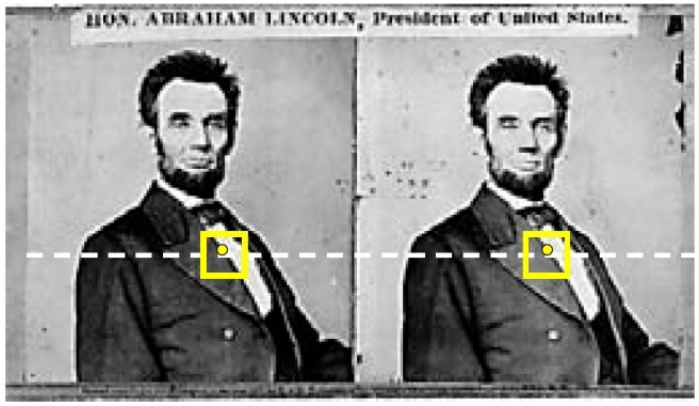
- **Ondra Chum, CMP Prague,**
- **Krystian Mikolajczyk, University of Surrey**,
- Darya Frolova, Denis Simakov,The Weizmann Institute of Science
- Martin Urban , Stepan Obdrzalek, Ondra Chum Center for Machine Perception Prague
- Matthew Brown,David Lowe, University of British Columbia

Outline

1. Local features: introduction, terminology
2. History: generalisation of local stereo to wide-baseline stereo
3. Examples and Applications:
   retrieval, panorama, augmented reality
4. Local invariant features:
   1. to rotation: Harris, Hessian, Laplacian of Gaussian
   2. to scale: as in 1. with scale selection in a pyramid
   3. to affine tr.: MSERs, as in 2. with Baumberg iteration,
   4. FAST, BRIEF-multi-scale FAST with orientation, ORB
5. Comparison of properties
6. Descriptors
7. The Two-view matching pipeline

History: Generalization of Local Stereo to WBS

1. Local Feature (Region) = a rectangular “window”
   - robust to occlusion, translation invariant
   - matched by correlation, assuming small displacement , …
   - successful in stereo matching of *rectified images* (Lincoln)
2. Local Feature (Region) = a circle around an “interest point”
   - robust to occlusion, translation and rotation invariant
   - 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

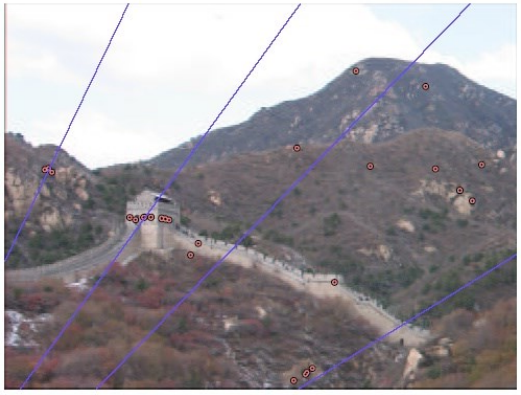


Hard Impossible for a Local feature based method

Motivation: Generalization of Local Stereo to WBS

CENTER FOR MACHINE PERCEPTION

3. Widening of baseline or zooming in/out
   - local deformation is well modelled by affine or similarity transformations
   - how can the “local feature” concept be generalised? *The set of ellipses is closed under affine tr., but it’s too big to be tested*
   - window scanning approach becomes computationally difficult.




Local Feature Detector

CENTER FOR MACHINE PERCEPTION

Definition: a feature **detector** (extractor) is an algorithm taking an image as input and outputting a set of regions ("local features").

"Local Features" are **regions,** i.e. in principle arbitrary sets of pixels, not necessarily contiguous, which are at least :

- **distinguishable** regardless of viewpoint/illumination
- **robust to occlusion** must be **local**
- Must have a discriminative neighborhood: they are **"features"**

Terminology has not stabilised:
Local Feature = Interest "Point" = Keypoint =
= Feature "Point" = The "Patch"
= Distinguished Region = Features
= (Transformation) Covariant Region

Feature Detectors: Desiderata

CENTER FOR MACHINE PERCEPTION

1. **Covariance** to a broad class of geometric and
   Invariance photometric transforms respectively
2. **Efficiency**: many applications require real-time
3. **Quantity/Density** of features to cover objects/part of scenes
4. **Robustness** to:
   - occlusion and clutter (requires *locality)*
   - to noise, blur, discretization, compression
5. **Distinctiveness:** features recognizable in a large database
6. **Stability over time** for long-temporal-baseline matching
7. **Geometrically accuracy**: precise localization
8. **Generalization** to similar objects
9. Complementarity, number of geometric constraints, …

No detector dominates in all aspects, some properties are competing, e.g. level of invariance x speed

Feature Descriptor.

CENTER FOR MACHINE PERCEPTION

Definition: a **descriptor** is a vector-valued function computed on an image region defined by a detector. The descriptor is a representation of the intensity (colour, depth) in the region.

.

Desiderata for feature descriptors:

1. Discriminability
2. Robustness to misalignment, illumination, blur, compression, ...
3. Efficiency: real-time often required in applications
4. Compactness: small memory footprint. Very significant on mobile large-scale applications

Note: The region on which a descriptor is computed is called a **measurement region.** This may be directly the feature detector output or any other function of it (eg. convex hull, triple area region..)

Application Domains

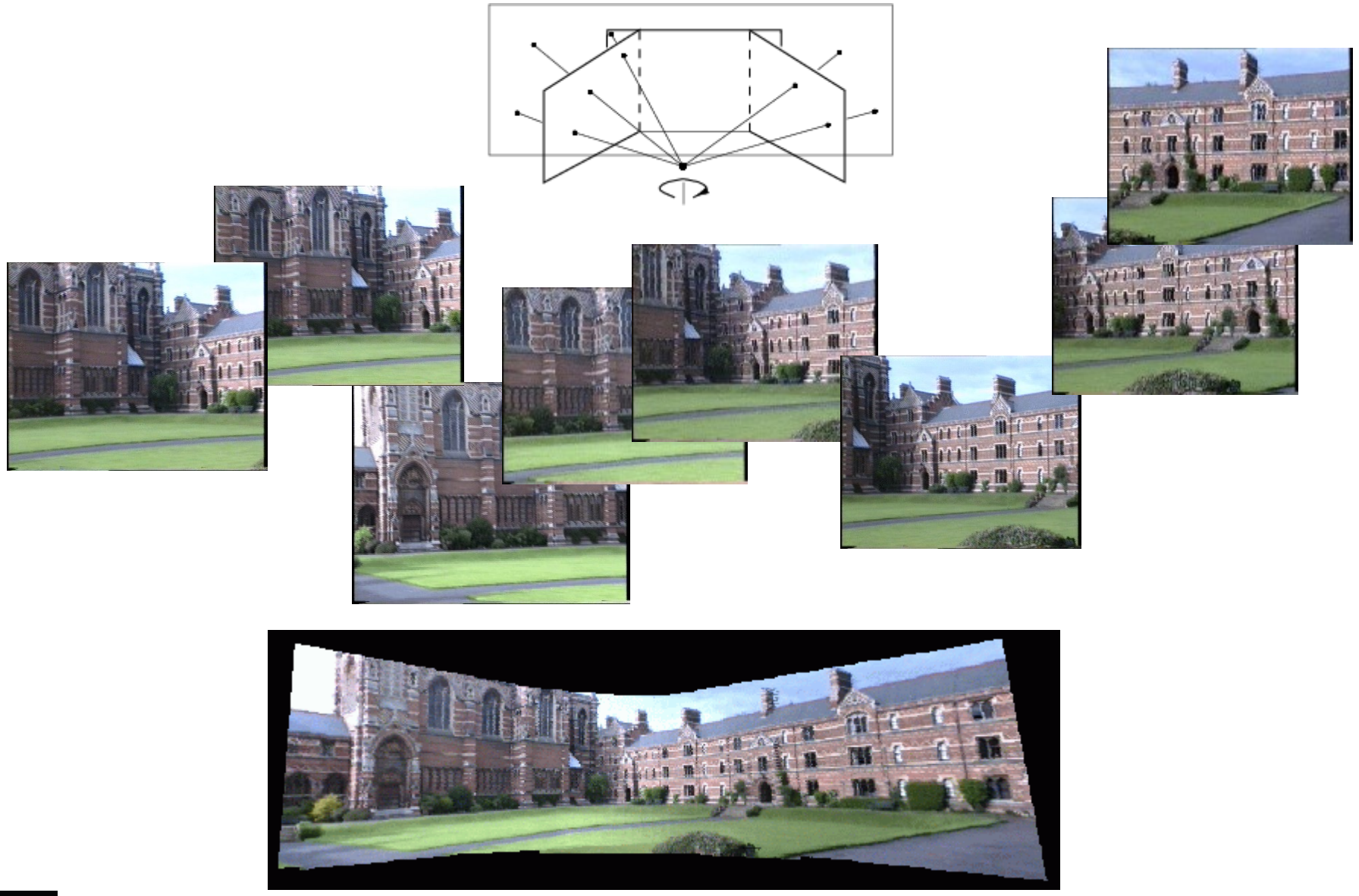
- **Methods based on "Local Features" are the state-of-the-art for number of computer vision problems,** mostly those that require local correspondences or extract geometry.
- Suited to instance matching over change in viewpoint, scale, lighting, partial occlusion, region of interest …
- Multiple views of the same scene, e.g.
  - Computing epipolar geometry or a homography
  - Photo Tourism
  - Panoramic mosaic
- Query by example search in large scale image datasets, e.g.
  - Google goggles
  - Where am I? Match to Streetview
  - Copy detection
- Re-acquisition in tracking

Applications: Wide baseline matching

- Establish correspondence between two (or more) images
- Useful in visual geometry: Camera calibration, 3D reconstruction, Structure and motion estimation, …

Local transf: scale/affine – Detector: affine-Harris Descriptor: SIFT

Applications: Panoramic mosaic



Applications: Building a Panorama

M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003

Applications

AutoStitch iPhone

Home | Usage | Gallery | FAQ | Reviews | Company | News

Automatic Image Stitching for the iPhone

AutoStitch iPhone is a fully automatic image stitcher for the iPhone. This application unleashes the power of your iPhone's camera to create wide-angle views and panoramas with any arrangment of photos.

AutoStitch uses the most advanced stitching technology available today, but it's very simple to use. To see how it works on the iPhone/iPod Touch, see our usage instructions or tutorial video.

AutoStitch iPhone brings together years of research and development experience into an amazing application that is available now on your iPhone at a very low price.

*** **Note:** If you recently upgraded to iOS4 or to version 3.0 and are having problems, please see the top of our FAQ.

Available on the iPhone App Store

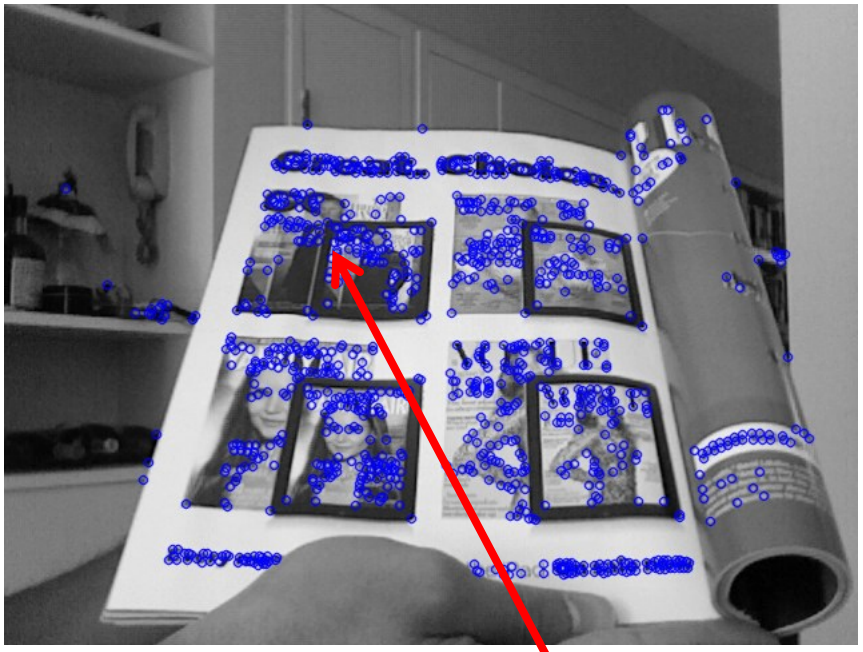
Buy AutoStitch from the App Store now for just $2.99!

Applications : Re-acquisition in tracking

CENTER FOR MACHINE PERCEPTION

Tracking target

Input image



Weight vector $\mathbf{w}_i$ per keypoint

Descriptor $\mathbf{d}_j$ per keypoint

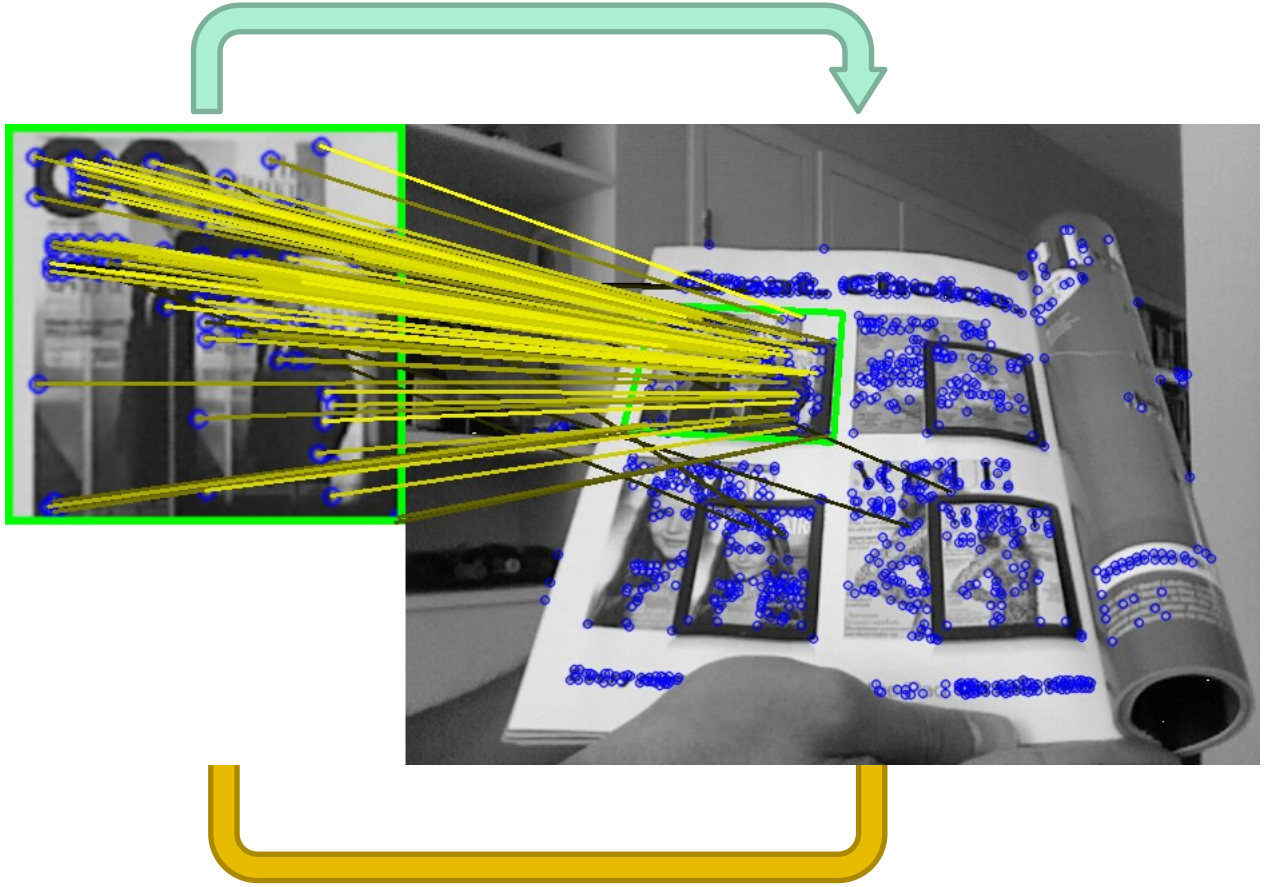
Correspondence score: $s_{ij} = \langle \mathbf{w}_i , \mathbf{d}_j \rangle$

Hare, Amri, Torr, CVPR 2012

Applications : Re-acquisition in tracking

1. Tracking Loop

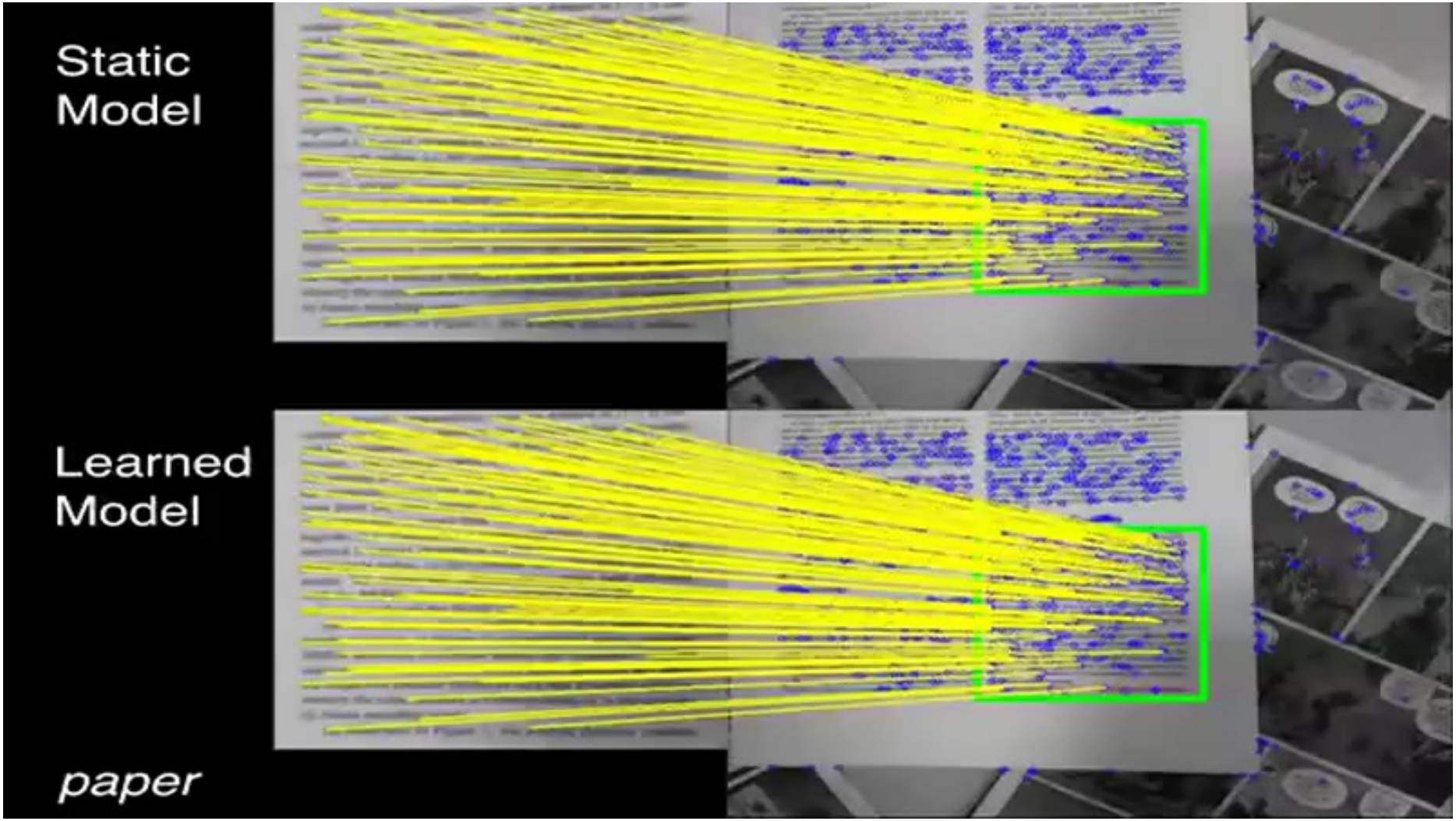
Detect — Correspondence generation + PROSAC



Hare, Amri, Torr,
CVPR 2012

Update — Structured SVM +
stochastic gradient descent

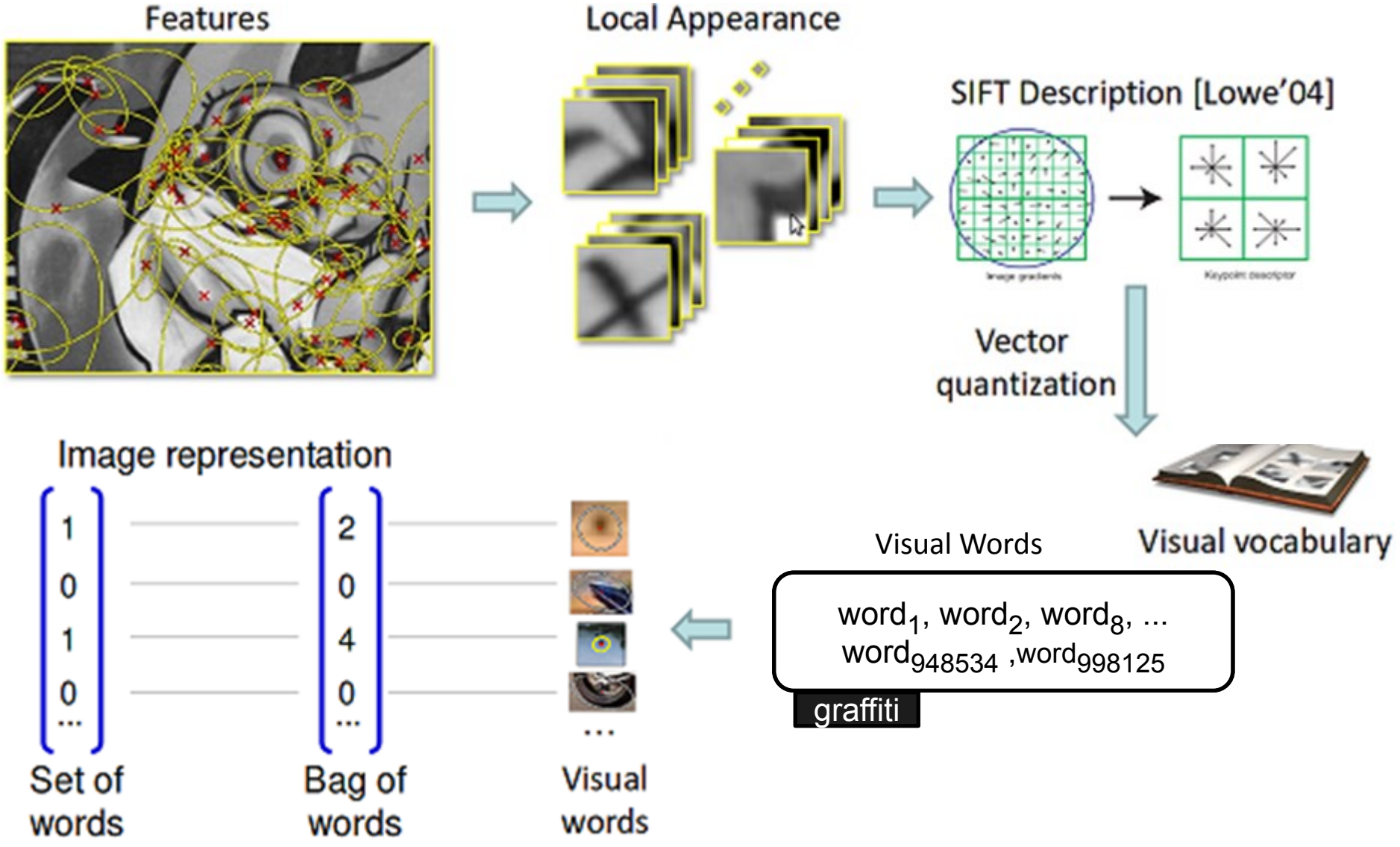
Example 8: Re-acquisition in tracking



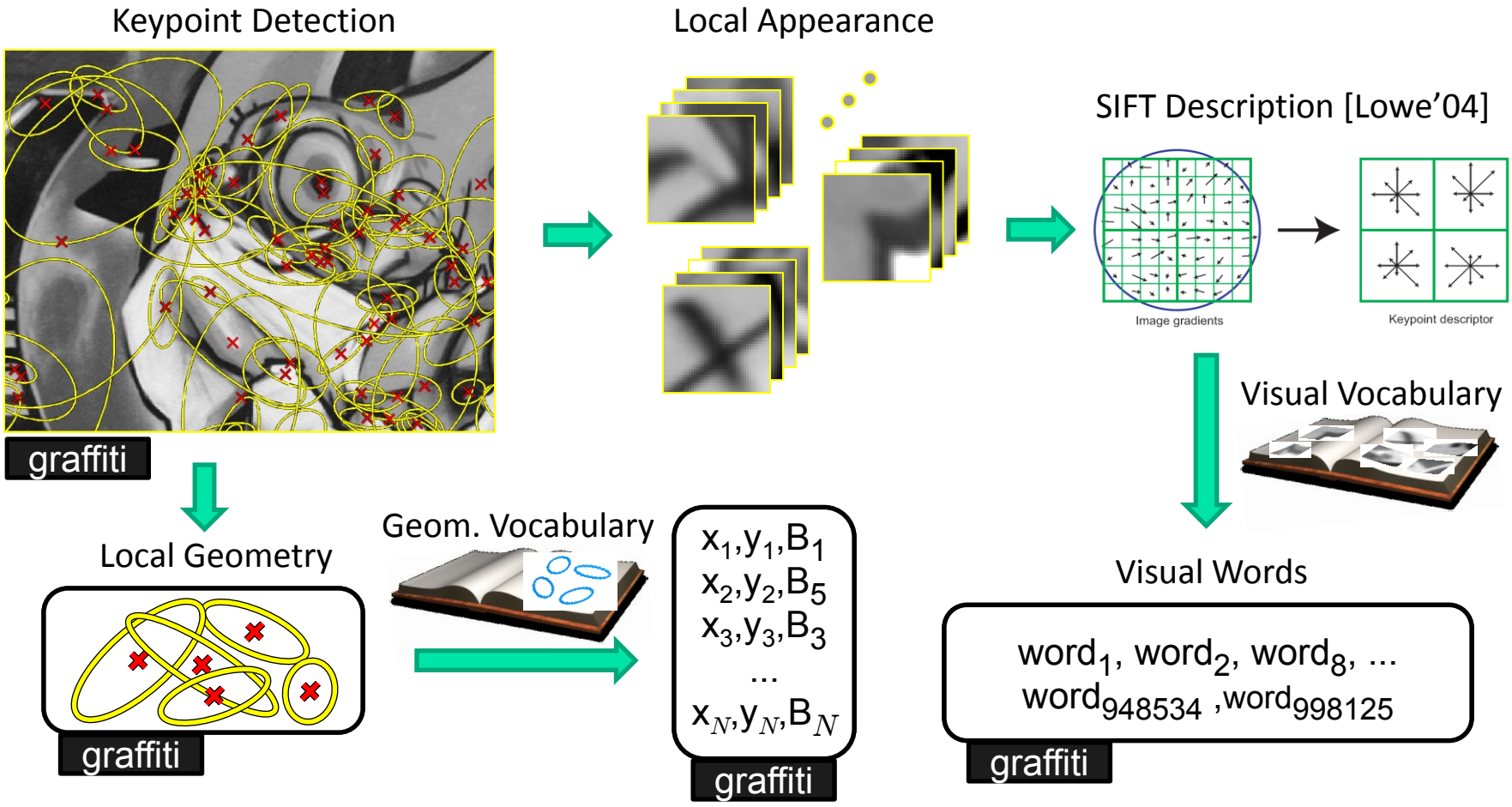
Applications : Re-acquisition in tracking

Courtesy of Graz University of Technology

Applications: BoW Image Retrieval

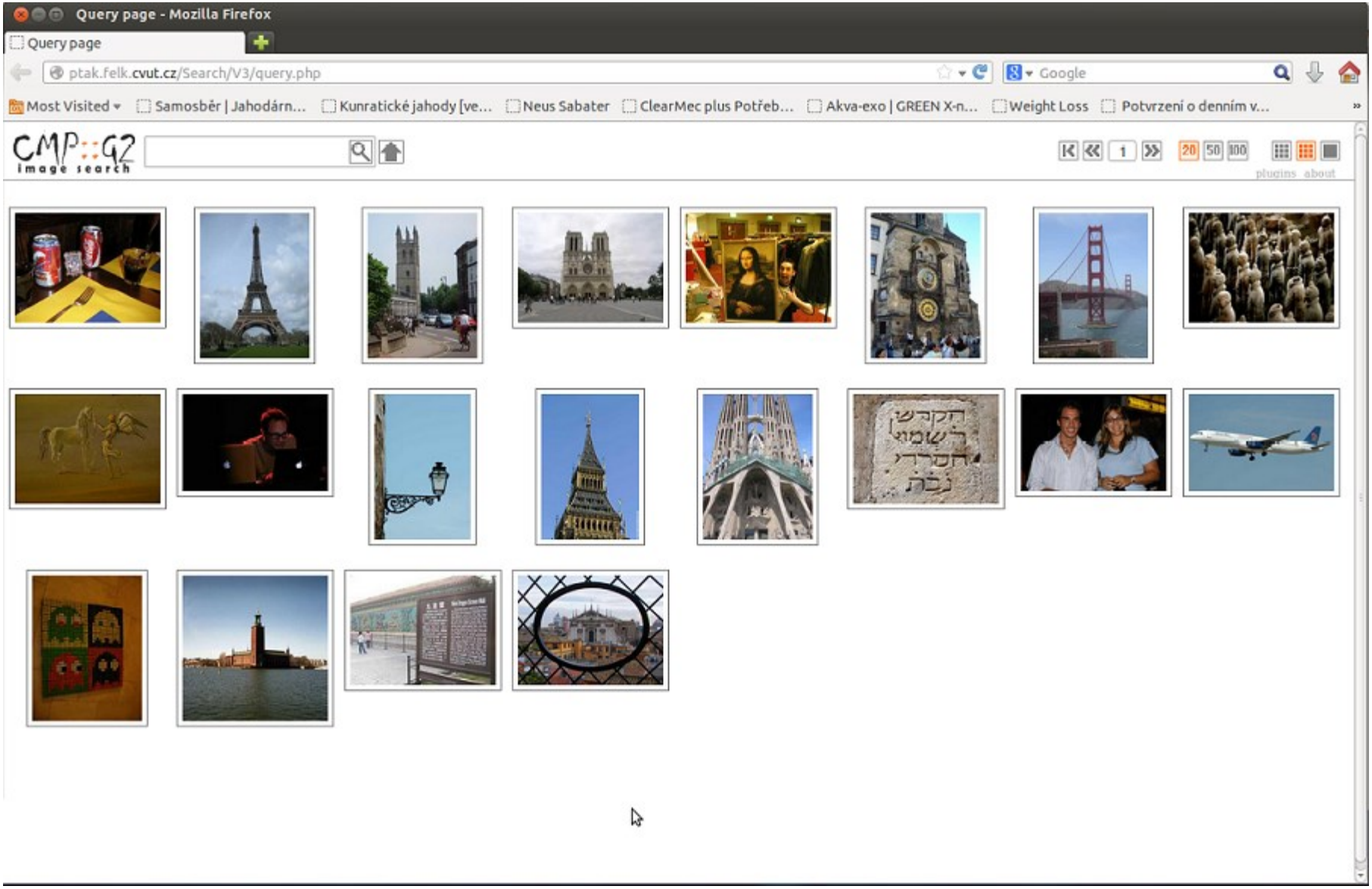


Applications: BoW Image Retrieval

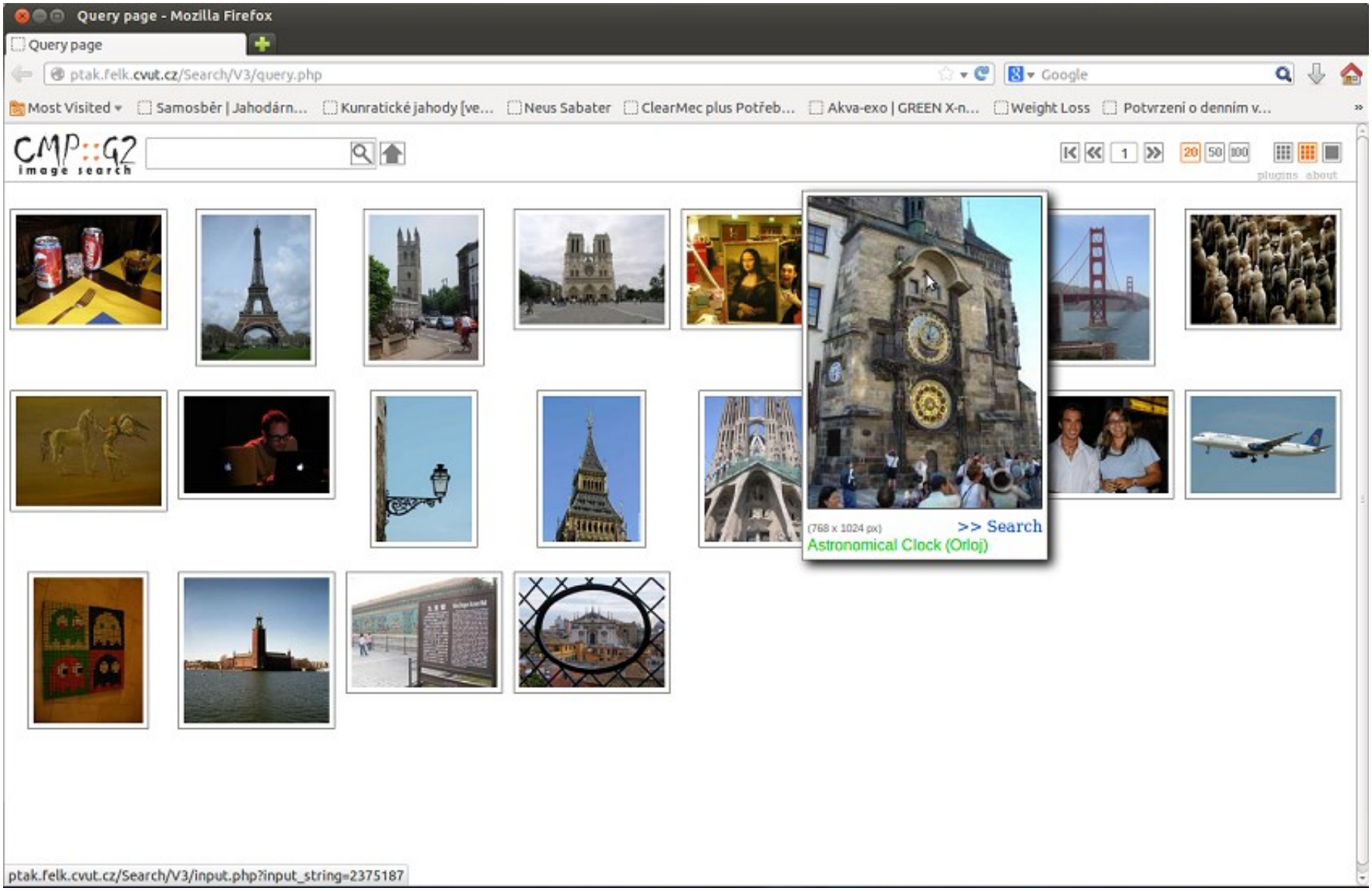


Coordinates x_i, y_i are quantized separately, and encoded using 16bits.

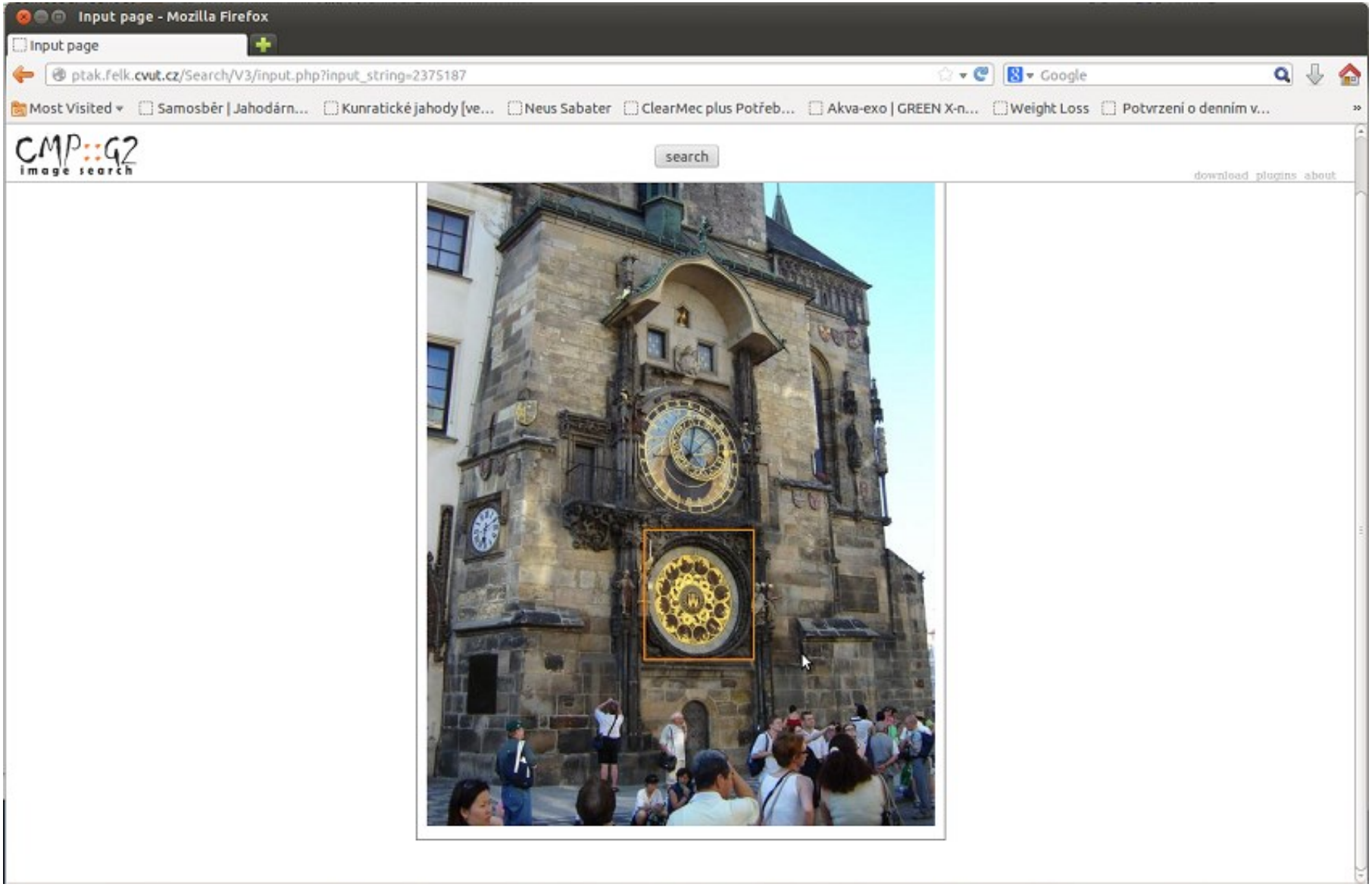
Applications: Image Retrieval



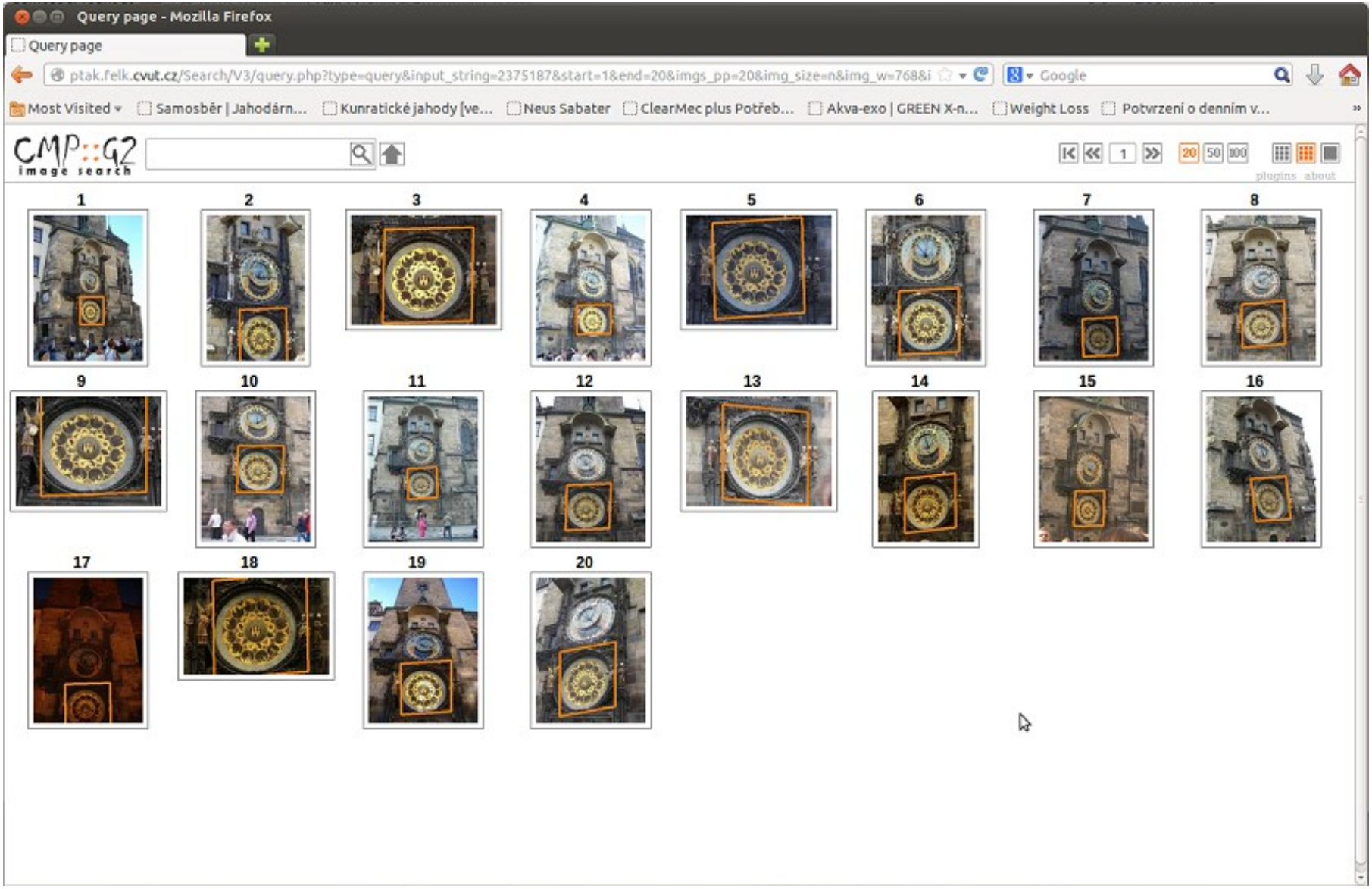
Applications: Image Retrieval



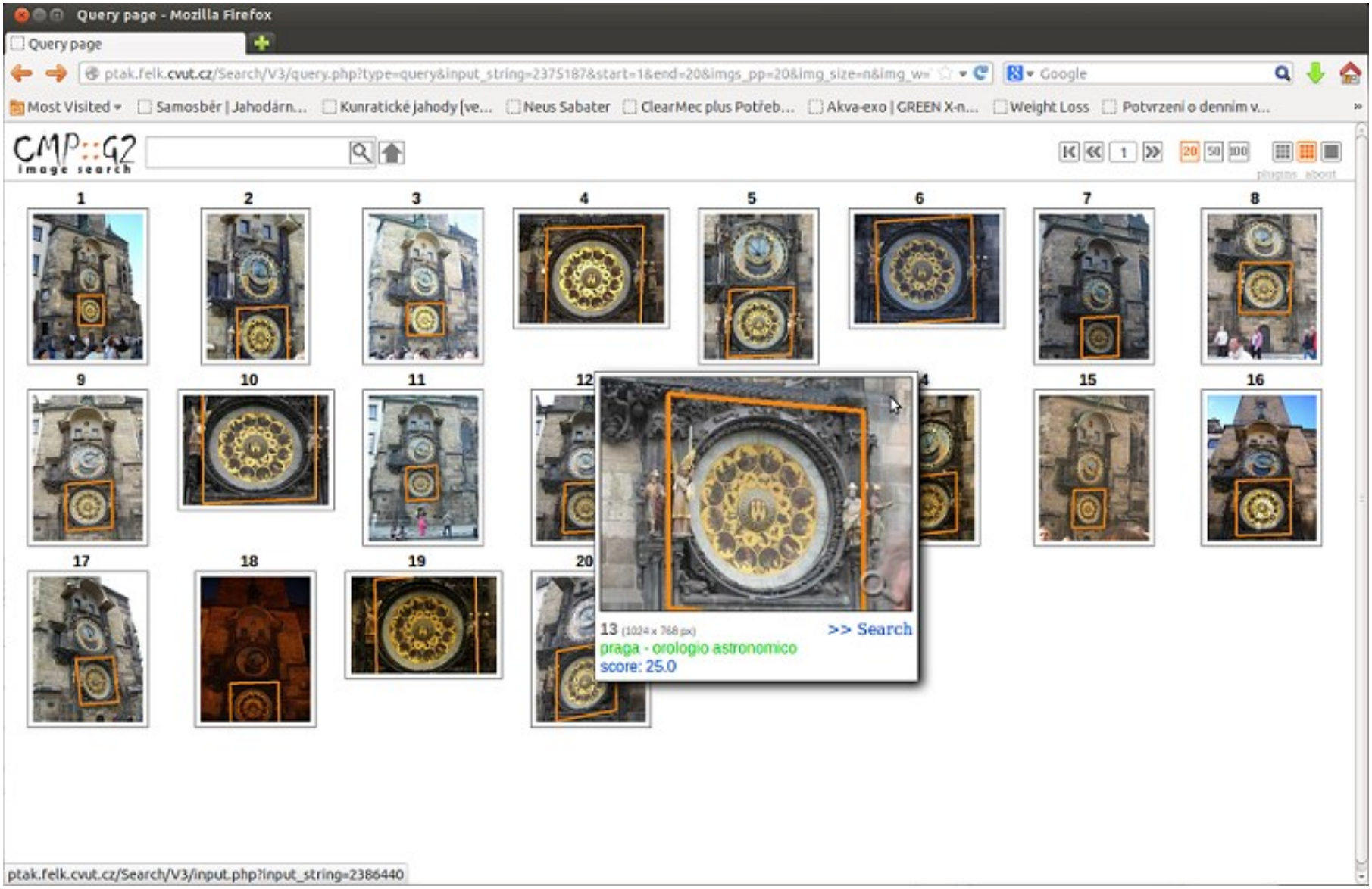
Applications: Image Retrieval



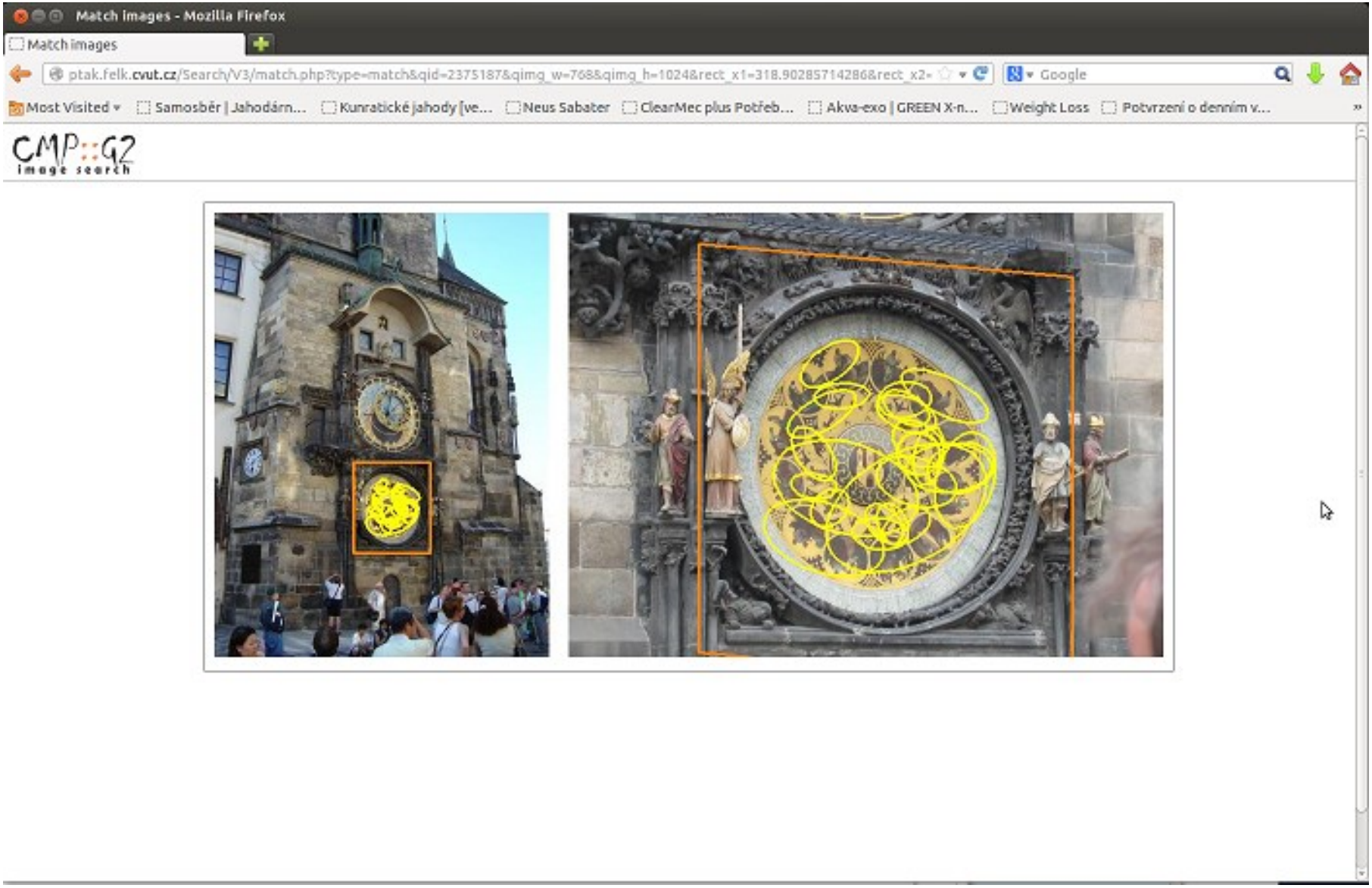
Applications: Image Retrieval



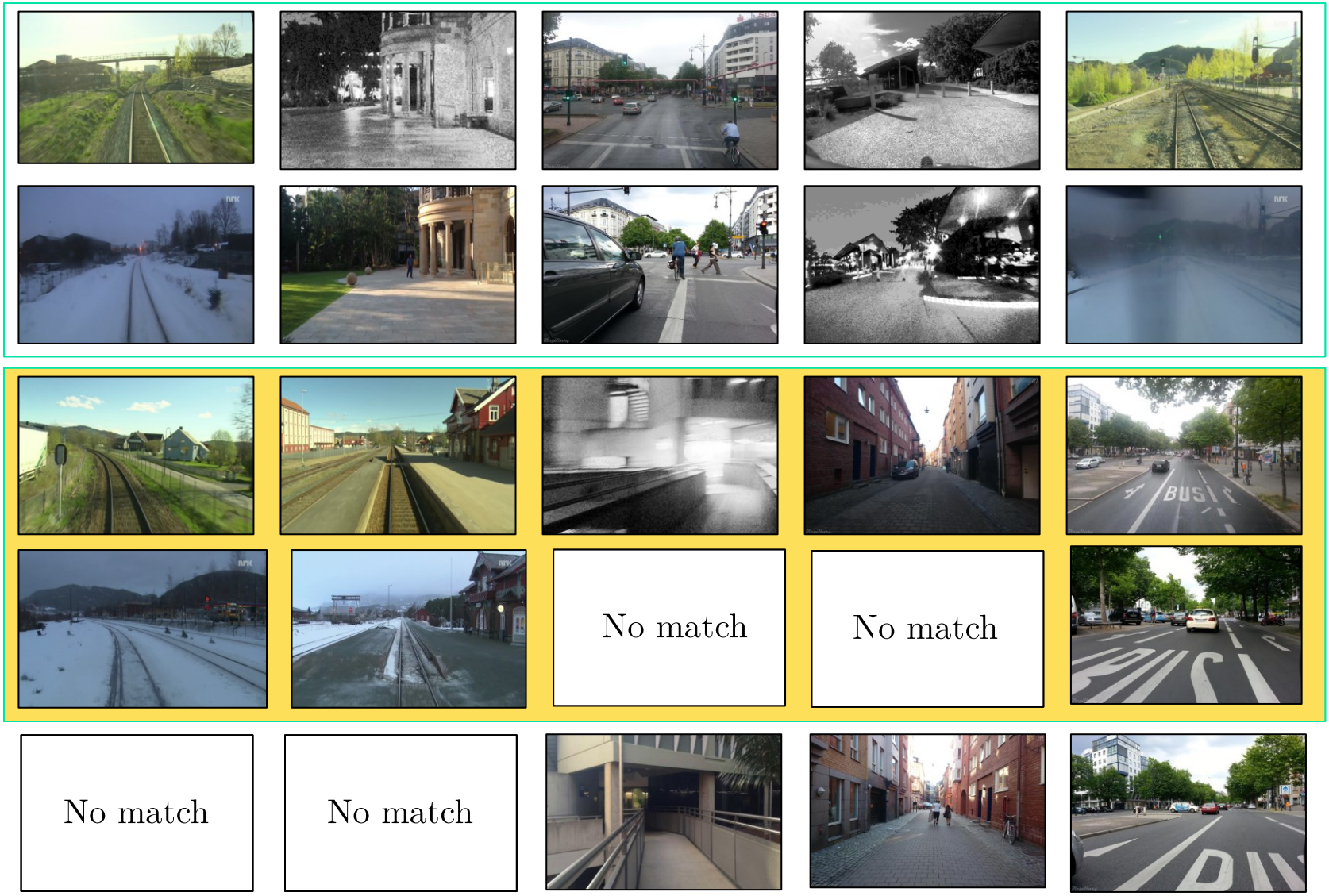
Applications: Image Retrieval



Applications: Image Retrieval



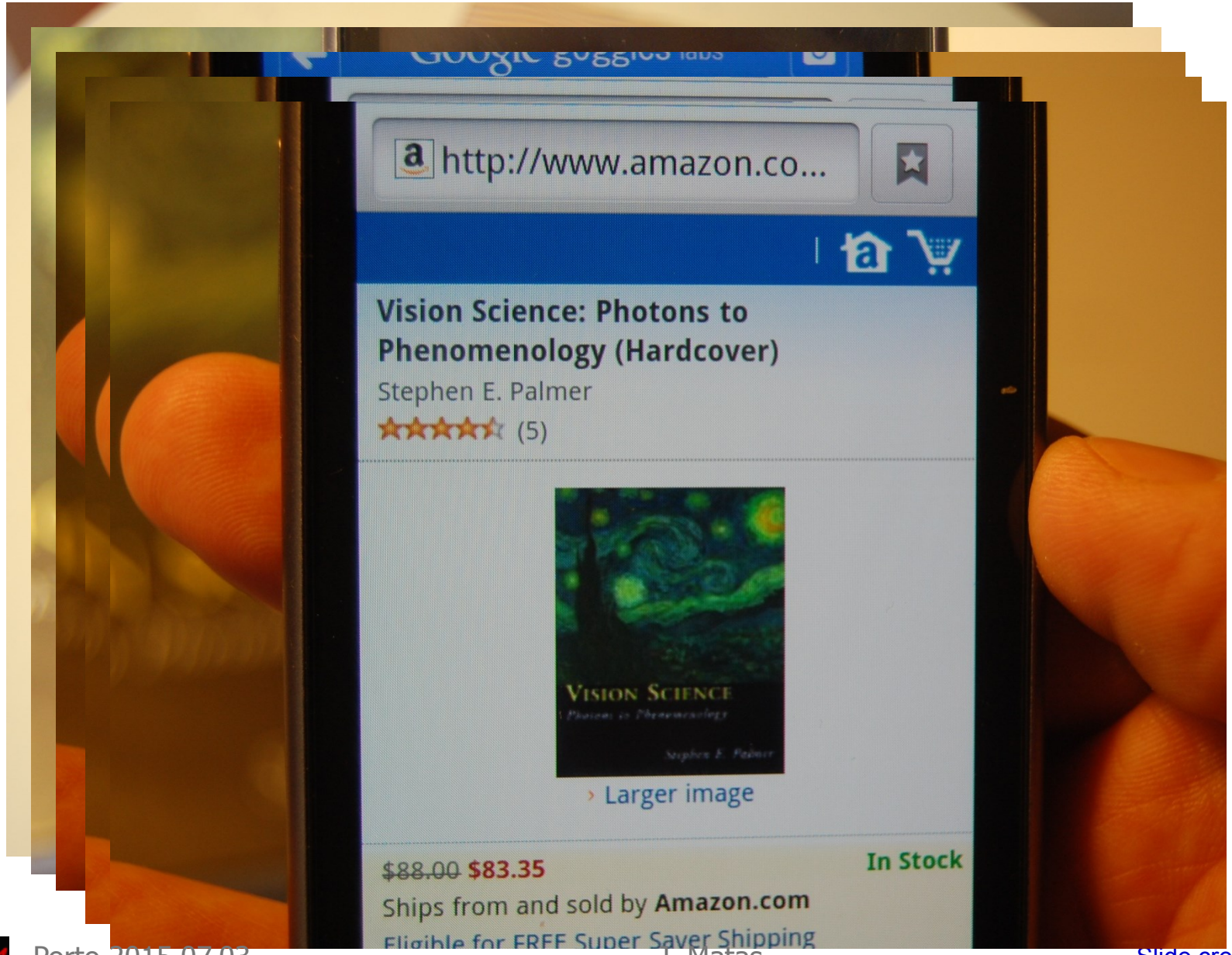
Success and failure cases



Why not use CNN?

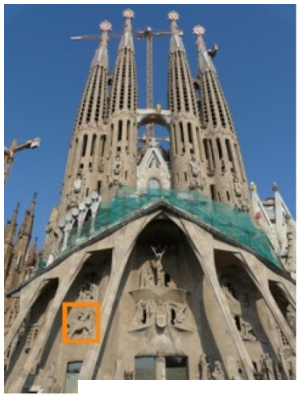
Possible, but needs learning for each dataset

Application: Google goggles



Slide credit: I. Laptev

Retrieval for Browsing: Zoom in



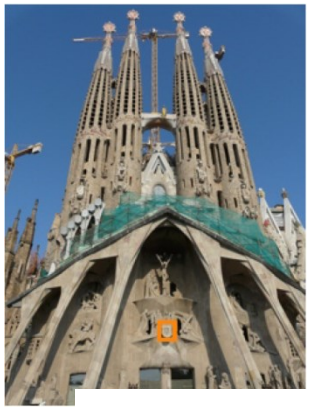
Query 1



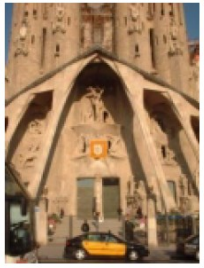






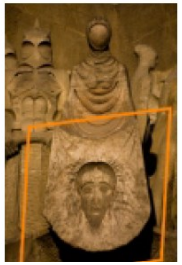
Query 2





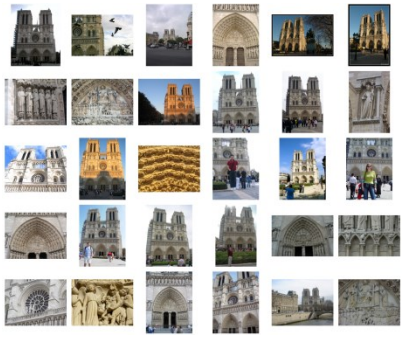




Applications : 3D reconstruction

CENTER FOR MACHINE PERCEPTION

1. Photo Tourism overview



Input photographs

Scene reconstruction

Relative camera positions and orientations

Point cloud

Sparse correspondence

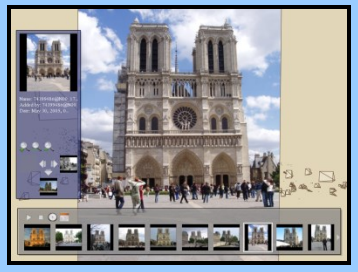


Photo Explorer

Applications : 3D reconstruction

CENTER FOR MACHINE PERCEPTION

57,845 downloaded images, 11,868 registered images. This video: 4,619 images.

The Old City of Dubrovnik

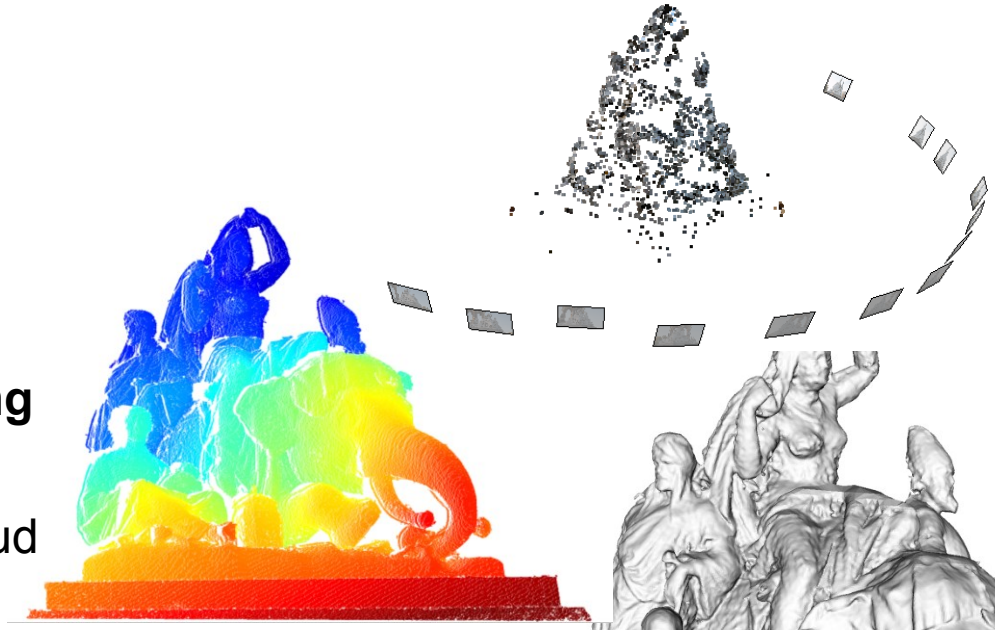
Building Rome in a Day, Agarwal, Snavely, Simon, Seitz, Szeliski, ICCV 2009
See also [Havlena, Torrii, Knop pand Pajdla, CVPR 2009].

3D reconstruction pipeline

CENTER FOR MACHINE PERCEPTION

1. **matching distinguished regions**
   - ⟹ tentative correspondences (verification)
   - ⟹ two view geometry

2. **camera calibration**
   - ⟹ camera positions
   - ⟹ sparse reconstruction

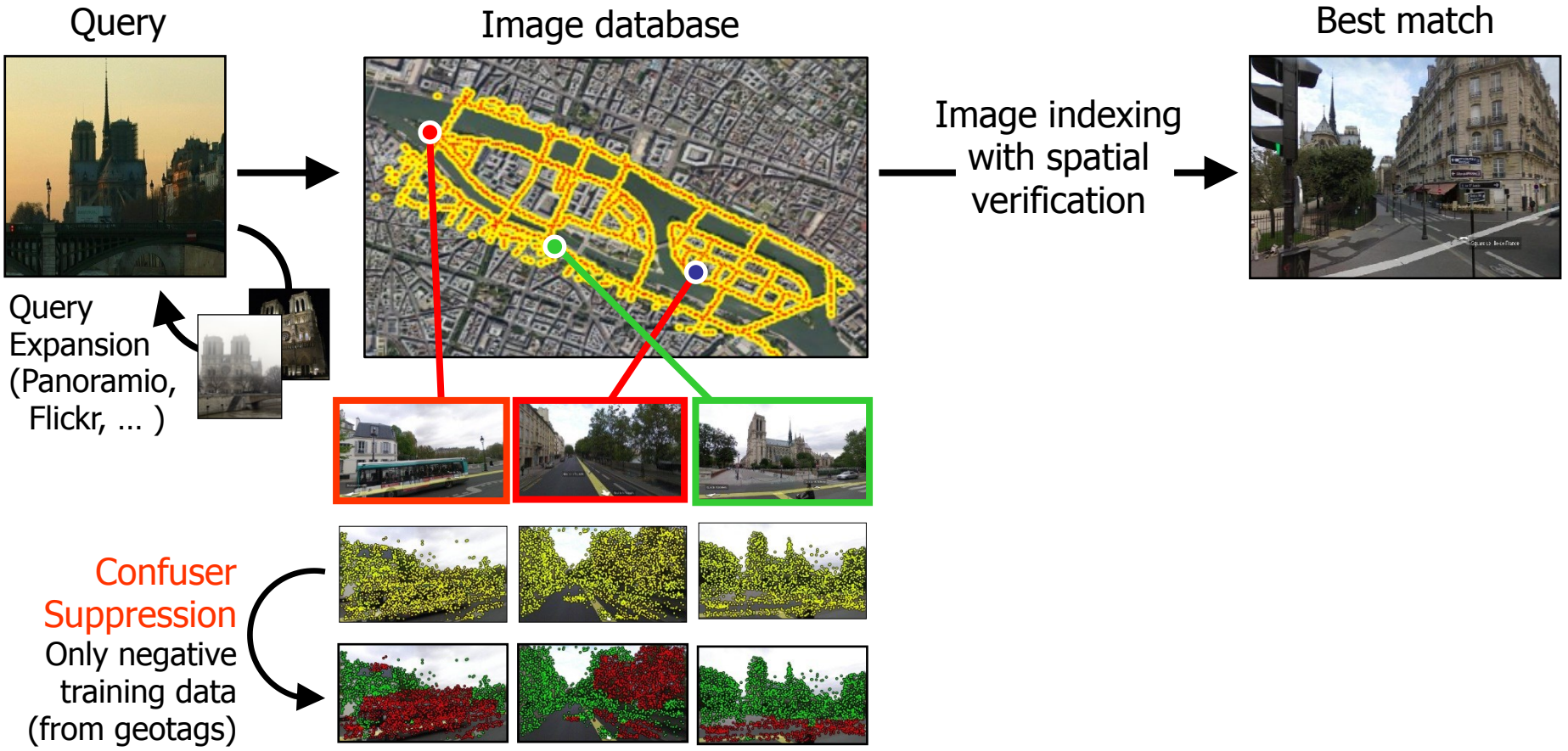


3. **dense stereoscopic matching**
   - ⟹ pixel/sub-pixel matching
   - ⟹ depth maps, 3D point cloud

4. **surface reconstruction**
   - ⟹ surface refinement
   - ⟹ triangulated 3D model

Applications: Where am I?

1. Place recognition - retrieval in a structured (on a map) database

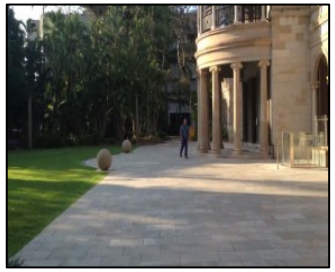


[Knopp, Sivic, Pajdla, ECCV 2010] http://www.di.ens.fr/willow/research/confusers/

CVPR 2015 Workshop and Contest on Visual Place Recognition in Changing Environments

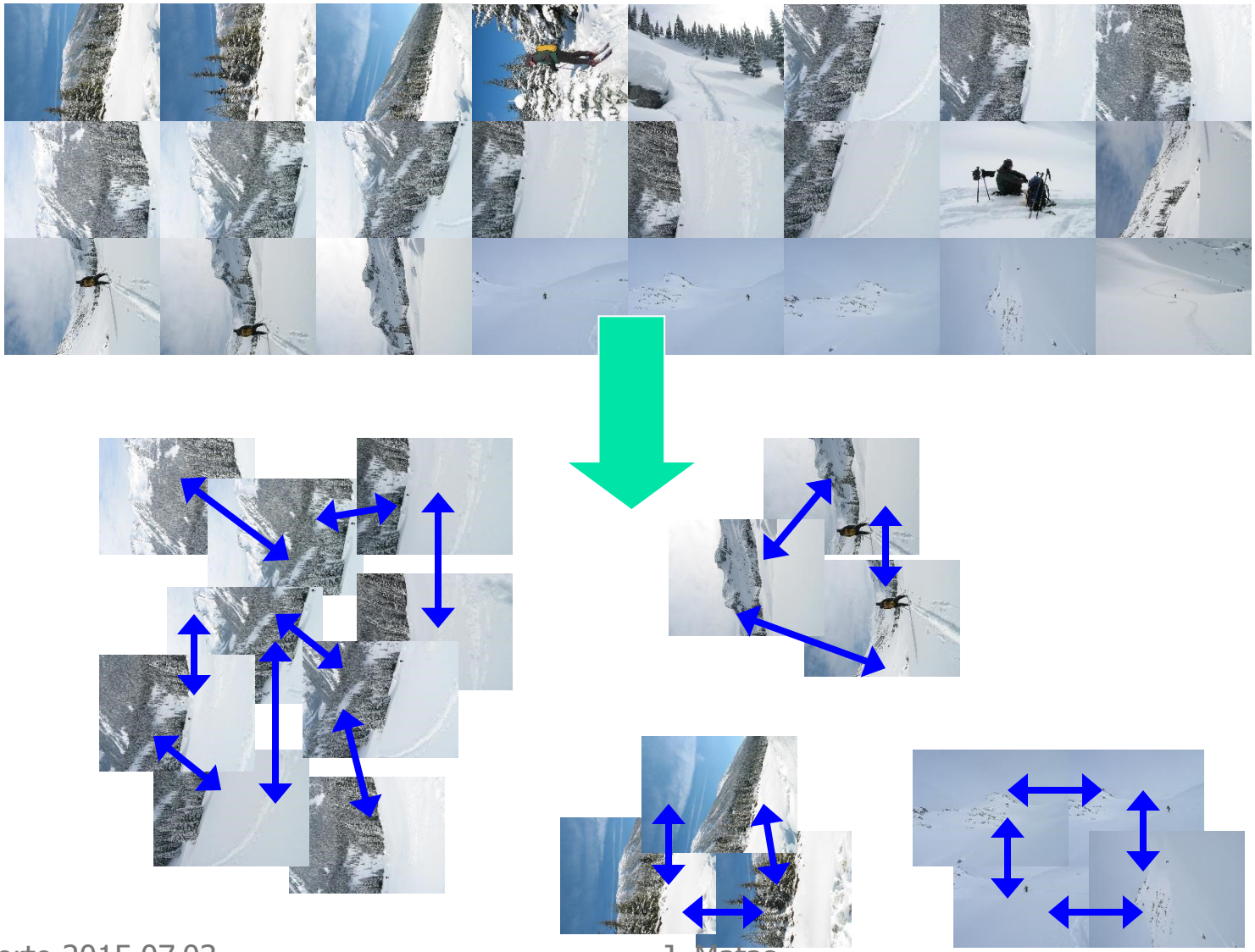
- Illumination and appearance
- Sensor
- Viewpoint
- Occlusions





Clustering: Find Connected Sets of Images



Why is Establishing Correspondence Difficult?

Finding correspondences is not easy

due to large viewpoint change (including scale)

$\Rightarrow$

the wide-baseline stereo problem

Applications:

- pose estimation
- 3D reconstruction
- location recognition

Finding correspondences is not easy

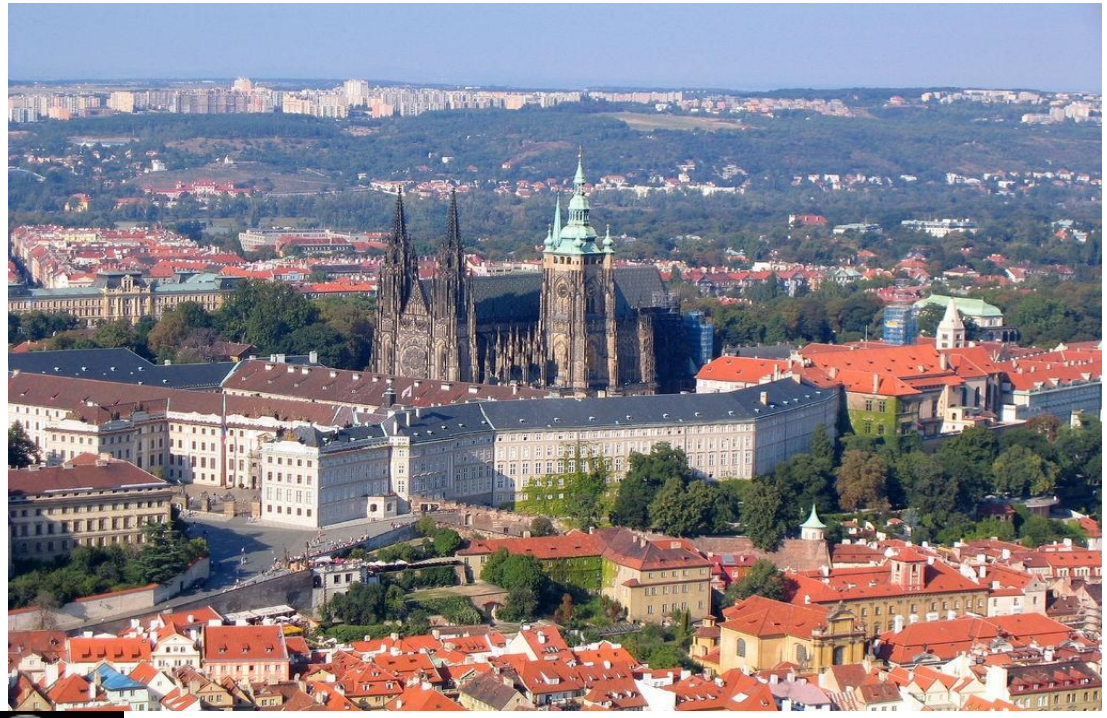
due to large viewpoint chang

(including scale)

$=>$

the wide-baseline (WBS)

stereo problem



Finding correspondences is not easy

due to large

illumination change

$=>$

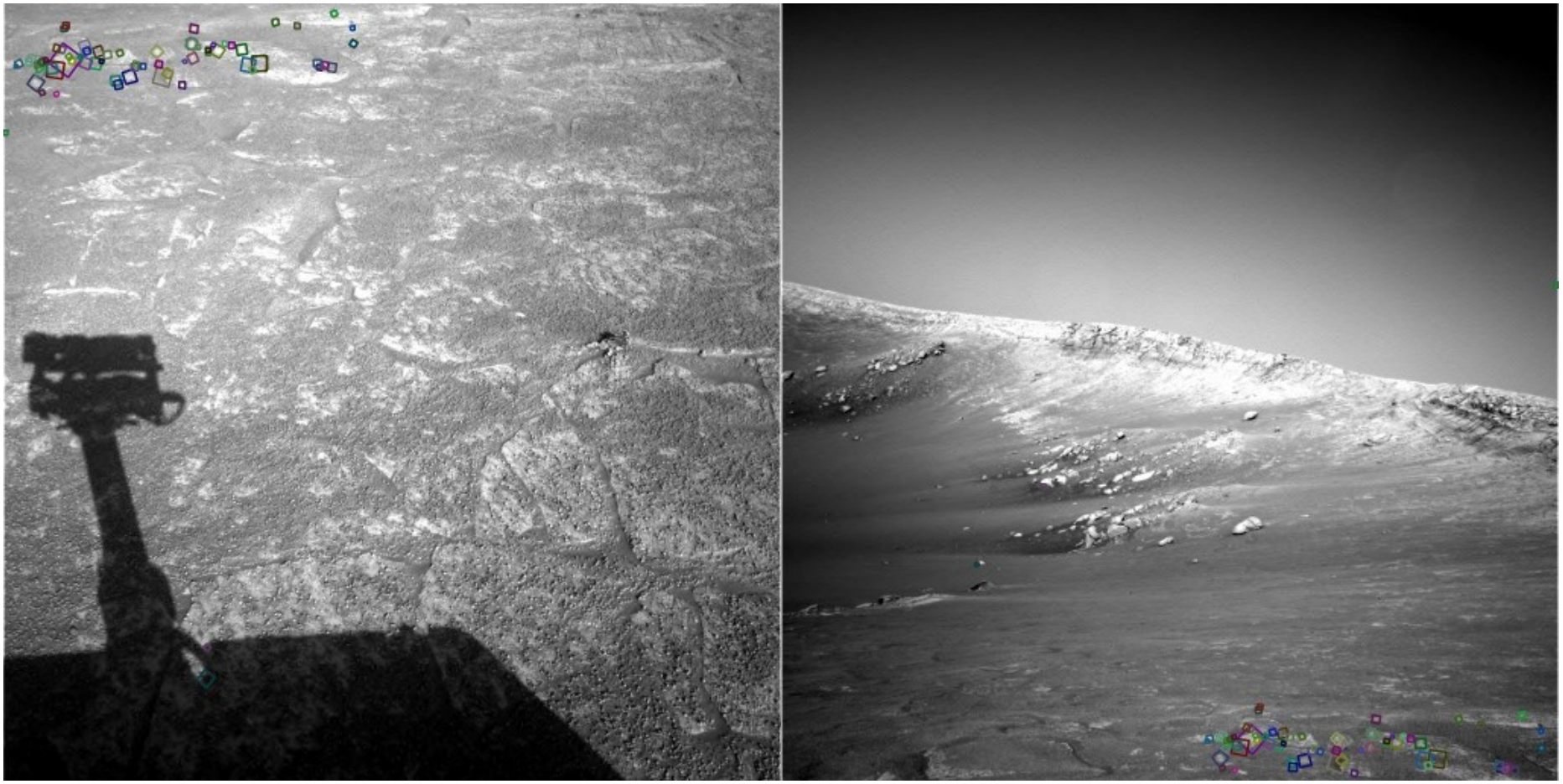
wide “illumination-baseline

stereo problem

Applications:

- location recognition
- summarization of image collections

Find the matches (look for tiny colored squares...)



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

Finding correspondences is not easy

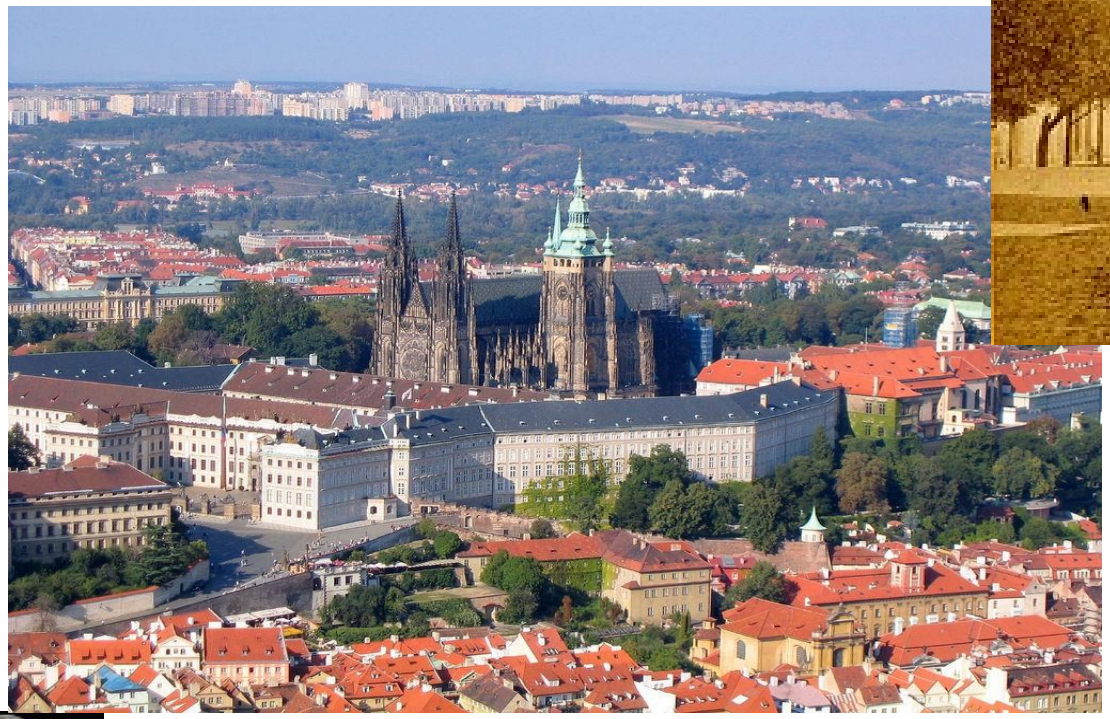
due to large

time difference

=¿

wide temporal-baseline

stereo problem



Applications:

- historical reconstruction
- location recognition
- photographer recognition
- camera type recognition

Finding Correspondences is not easy

CENTER FOR MACHINE PERCEPTION

due to occlusion

Applications:

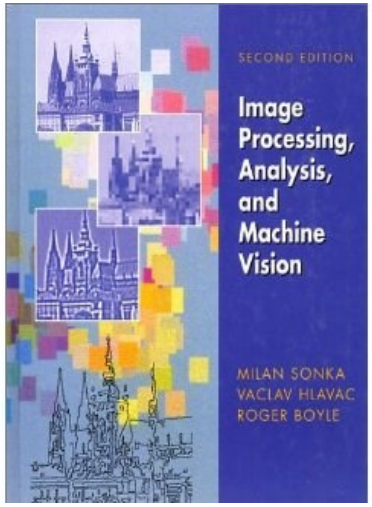
- pose estimation
- impainting

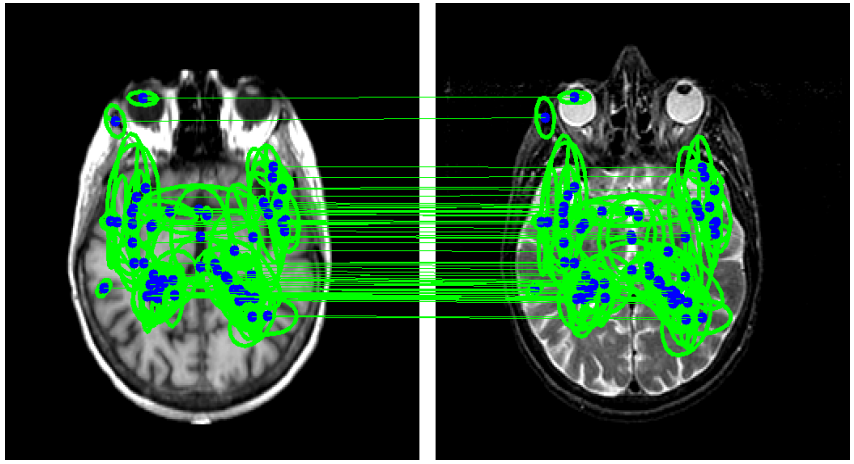
Finding Correspondences is not easy

change of modality

Applications:

- medical imaging
- remote sensing





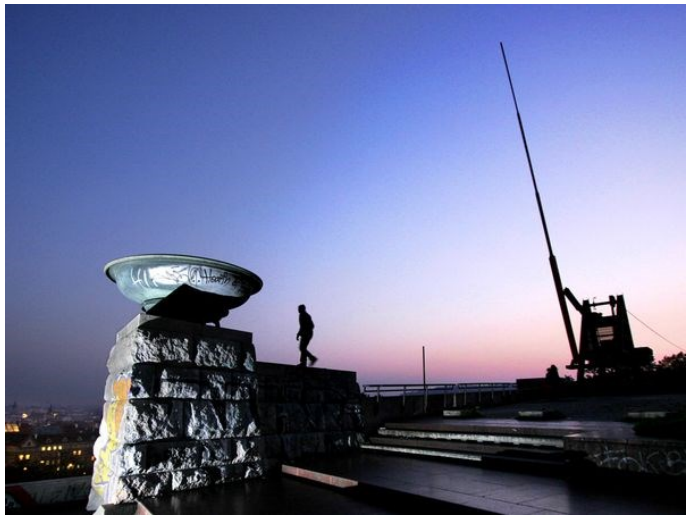
Finding Correspondences is not easy

CENTER FOR MACHINE PERCEPTION

WxBS all these effects can **happen at once**!

And more:

- non-rigid deformations, articulations
- repetitive patterns



The Basic Idea.

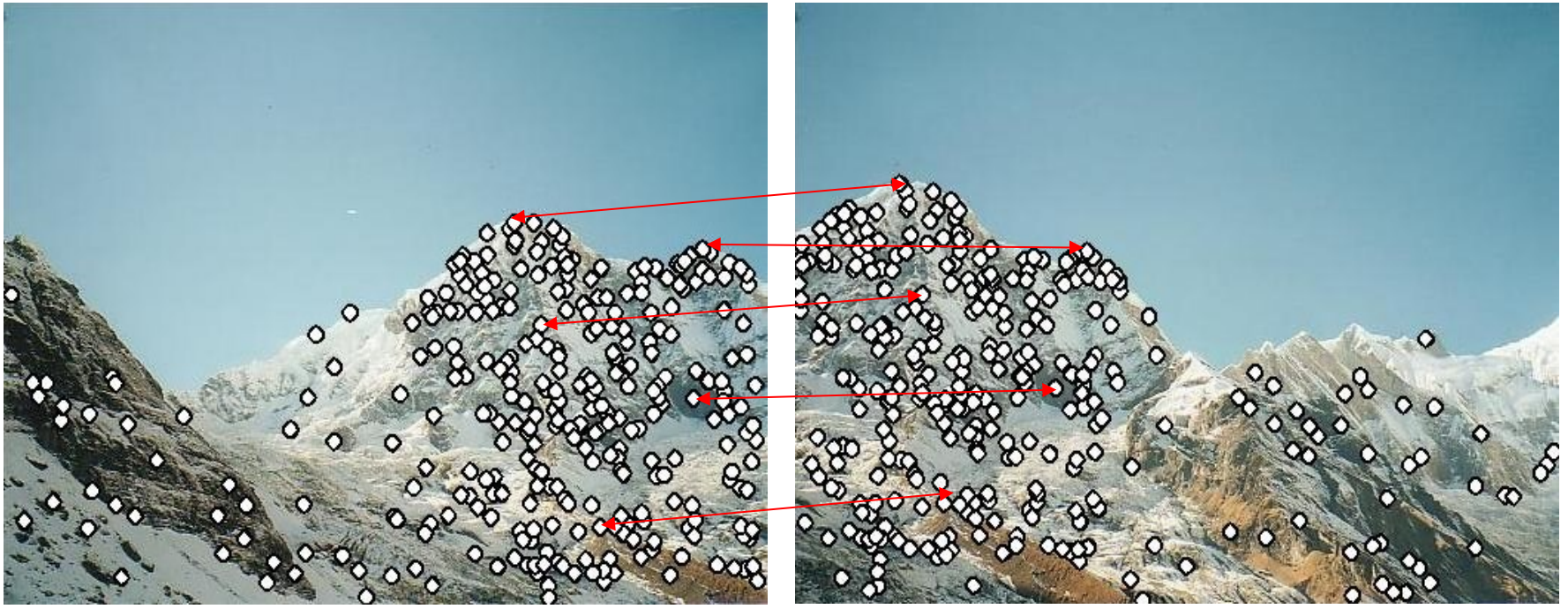
Local Features in Action (2): Building a Panorama

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

Local Features in Action (2): Building a Panorama

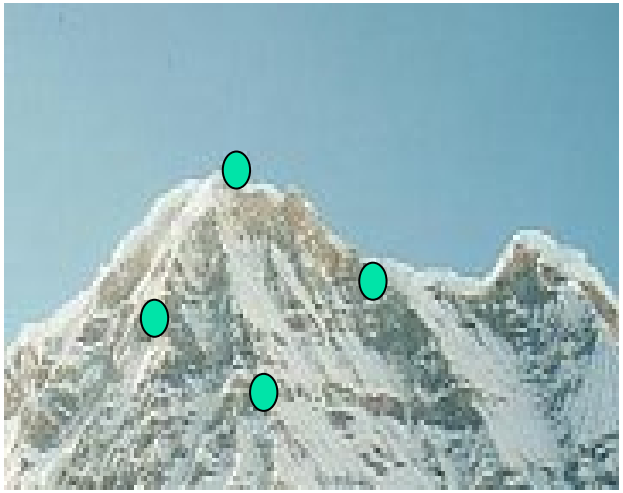
Possible approach:

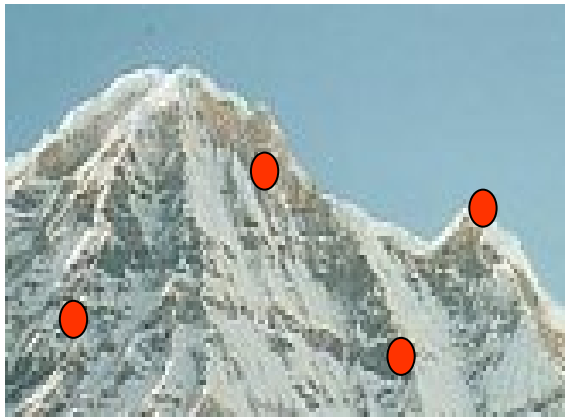
1. Detect features in both images
2. Find corresponding pairs
3. Estimate transformations (Geometry and Photometry)
4. Put all images into one frame, blend.



Local Features in Action (2): Building a Panorama

1. Problem 1:
   1. Detect the *same* feature *independently* in both images*
   2. Note that the set of "features" is rather sparse





no chance to match!

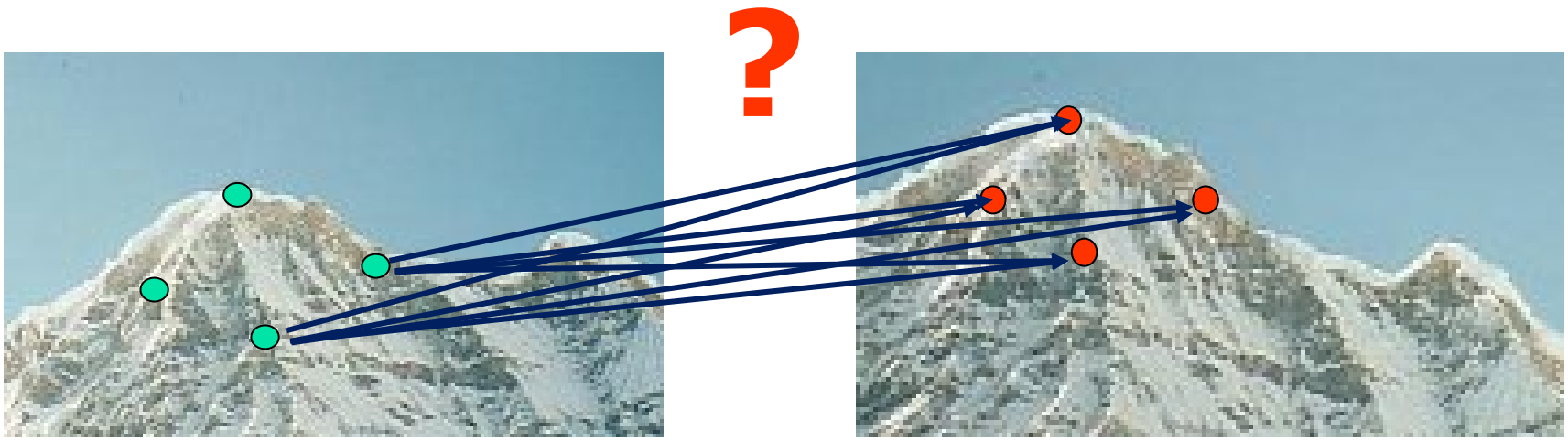
A repeatable detector needed.

* Other methods exist that do not need independency

Local Features in Action (2): Building a Panorama

CENTER FOR MACHINE PERCEPTION

1. Problem 2:
   1. how to correctly recognize the corresponding features?



Solution:

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

Local Features in Action (2): Building a Panorama

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

Any alternatives?

Local Invariant Features

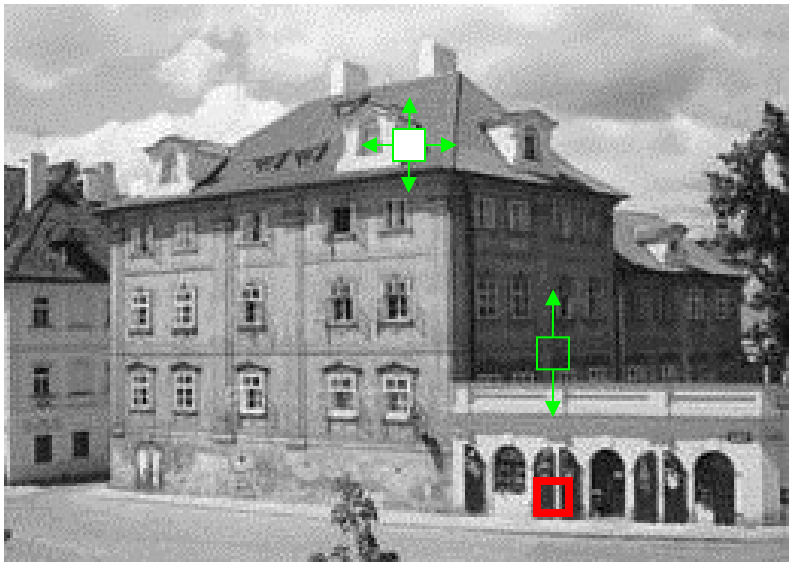
Design of Local Features

CENTER FOR MACHINE PERCEPTION

1. "Local Features" are **regions,** i.e. in principle arbitrary sets of pixels (not necessarily contiguous) with
2. High **repeatability** (invariance in theory) under
   1. Illumination changes
   2. Changes of viewpoint $\Rightarrow$ geometric transformations

   i.e. are **distinguishable** in an image regardless of viewpoint/illumination $\Rightarrow$ are **distinguished regions**
3. Are **robust to occlusion** $\Rightarrow$ must be **local**
4. Must have a discriminative neighborhood $\Rightarrow$ they are **"features"**
5. **Speed** is important in most applications. We'll cover features that run from 1 fps on 1MPxl image to 100 fps with various speed v. repeatability / invariance / robustness trade-offs.

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



undistinguished patches:

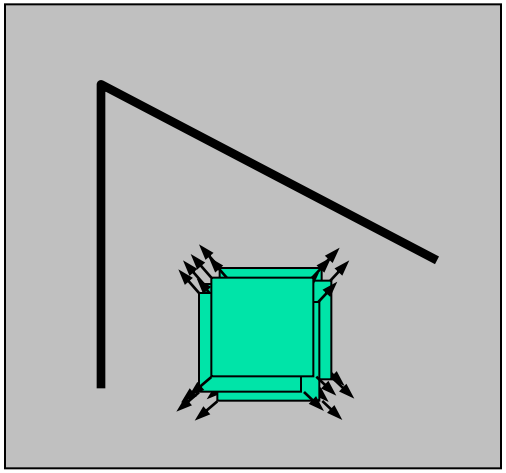
distinguished patch:

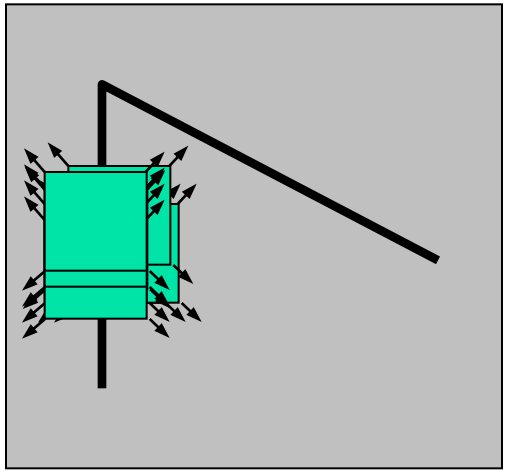
Two core ideas (in “modern terminology”):

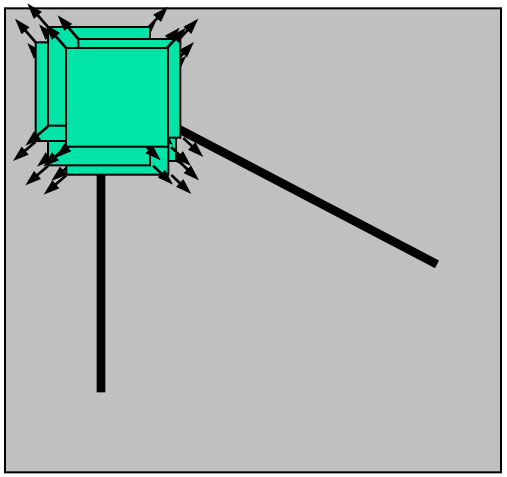
1. To be a distinguished region, a region must be *at least* distinguishable from *all* its neighbours.
2. Approximation of Property 1. can be tested very efficiently, without explicitly testing *all neighbours.*

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

Harris Detector: Basic Idea







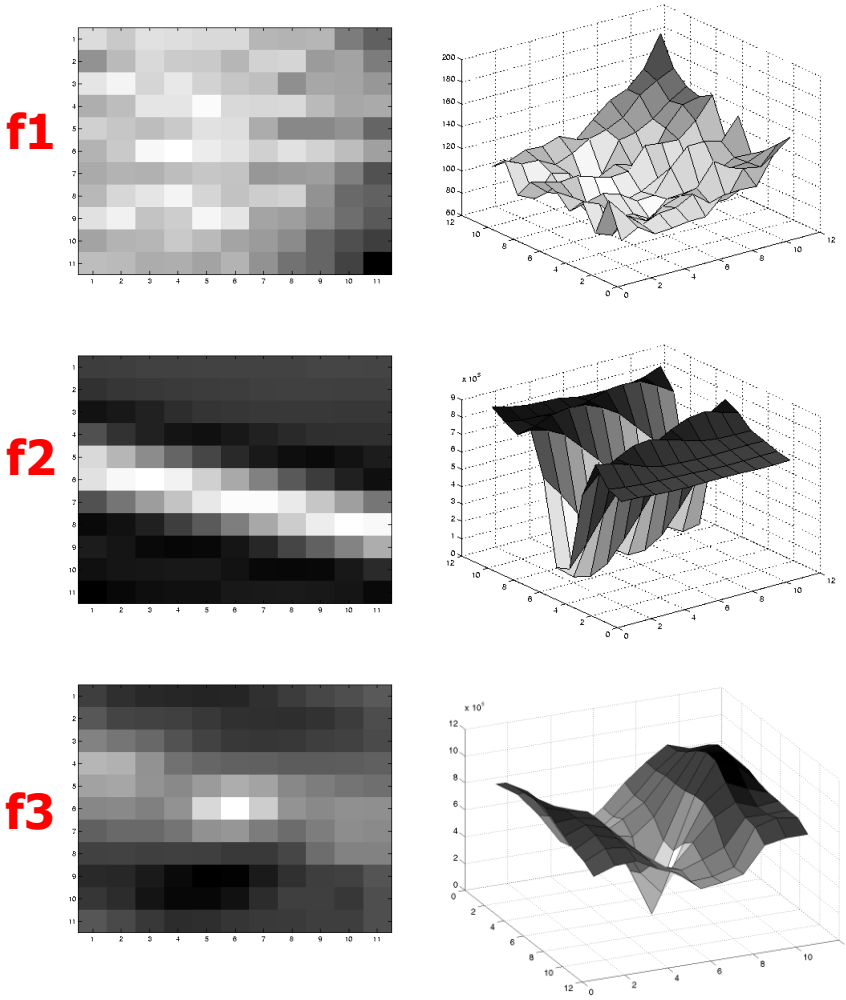
"flat" region: no change in all directions

"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: Basic Idea

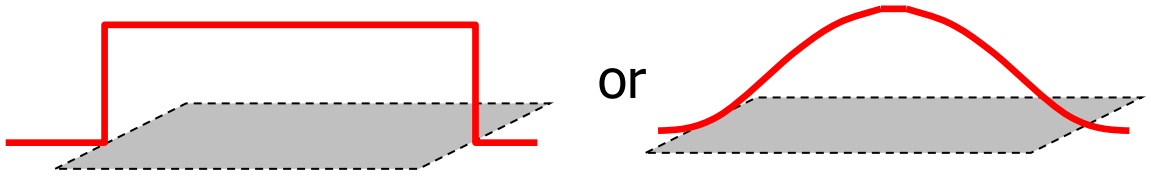


Harris Detector: Mathematics

Tests how similar the image function $I(x_0, y_0)$ is in a window centered at point (x_0, y_0) to a window shifted by small (u, v) using a quadratic loss function:

$$\mathrm{E}(x_0, y_0; u, v) = \sum_{(x,y) \in W(x_0, y_0)} w(x, y)(I(x, y) - I(x + u, y + v))^2$$

- the quality of the window around $I(x_0, y_0)$ is given by $\min_{u,v} \mathrm{E}(x_0, y_0; u, v)$
- $w(x_0, y_0)$ is a window centered at point (x_0, y_0)
- $w(x, y)$ can be constant or a (preferably) a trimmed Gaussian



1 in window, 0 outside Gaussian

Harris&Stevens (Foerstner, Shi&Tomasi)

- made the following observation:

The weighted *sum of squared differences* (SSD) between these two patches, denoted S, is given by:

$$S(x,y) = \sum_u \sum_v w(u,v)\ (I(u+x, v+y) - I(u,v))^2$$

$I(u + x, v + y)$ can be approximated by a Taylor expansion. Let I_x and I_y be the partial derivatives of I, such that

$$I(u+x, v+y) \approx I(u,v) + I_x(u,v)x + I_y(u,v)y$$

This produces the approximation

$$S(x,y) \approx \sum_u \sum_v w(u,v)\ (I_x(u,v)x + I_y(u,v)y)^2,$$

which can be written in matrix form:

$$S(x,y) \approx \begin{pmatrix} x & y \end{pmatrix} A \begin{pmatrix} x \\ y \end{pmatrix},$$

where A is the structure tensor,

$$A = \sum_u \sum_v w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}$$

Harris Detector: Mathematics

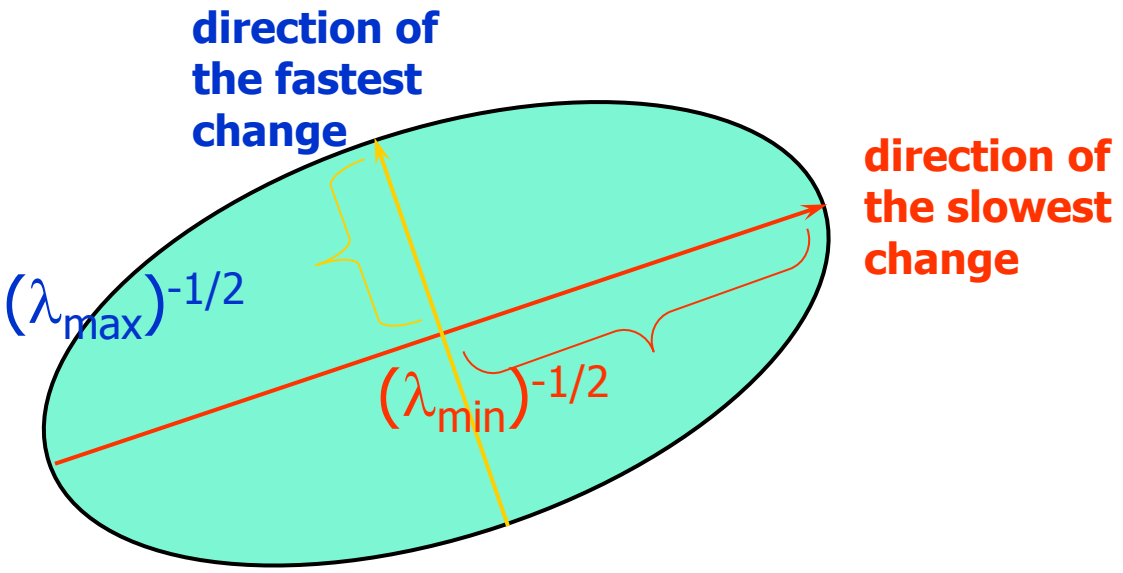
$$\mathrm{E}(x_0, y_0; u, v) \approx [u, v] A(x_0, y_0) \begin{bmatrix} u \\ v \end{bmatrix}$$

Intensity change in shifting window: eigenvalue analysis of A

- λ_1, λ_2 – eigenvalues of A
- A symmetric, positive definite

Ellipse:

$$E(x_0, y_0; u, v) = \text{const}$$



Harris Detector: Mathematics

Measure of corner response ("cornerness"):

$$R = \det A - k(\text{trace } A)$$

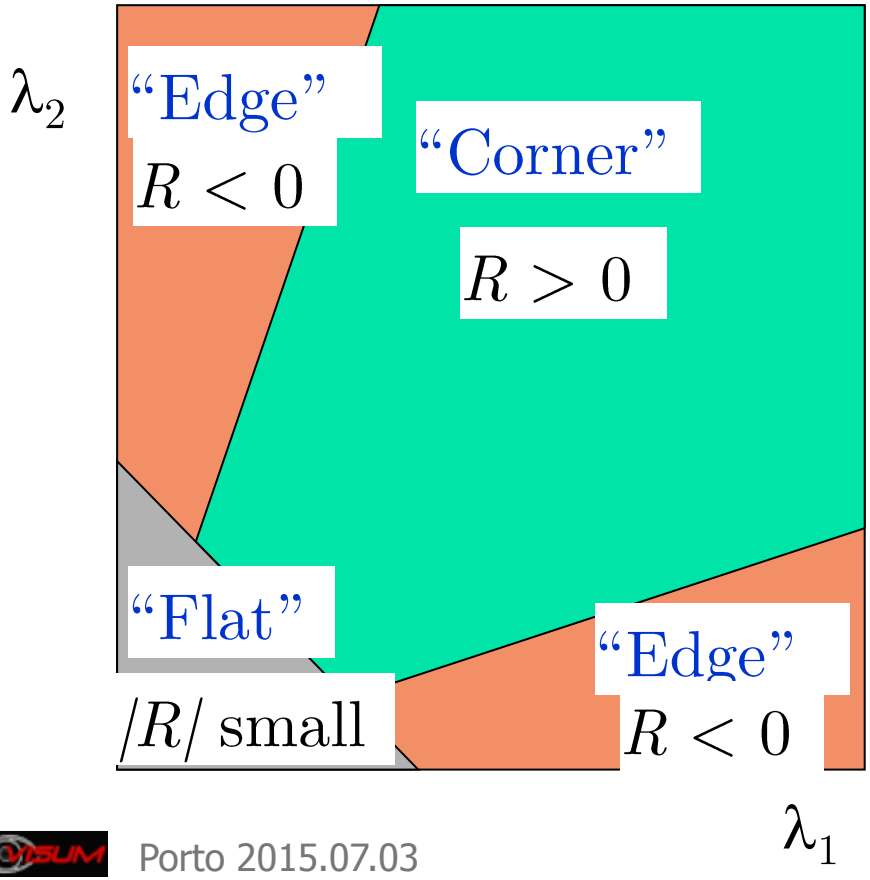
- $A = \begin{bmatrix} a & b \\ d & c \end{bmatrix}$
- $\det M = \lambda_1 \lambda_2 = ac - b^2$
- $\text{trace } M = \lambda_1 + \lambda_2 = a + c$
- k ... empirical constant, $k \in (0.04, 0.06)$

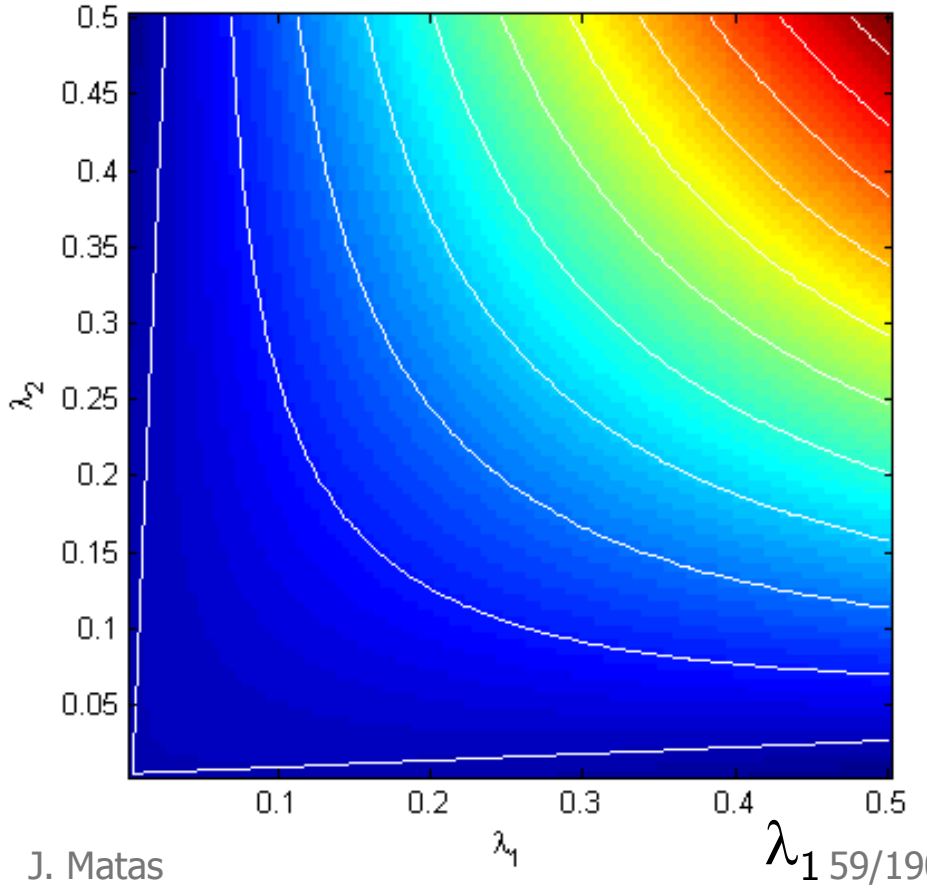
Find corner points as **local maxima** of corner response R:

- points greater than its neighbours in given neighbourhood (3×3, or 5×5)

Harris Detector: Mathematics

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





Harris Detector

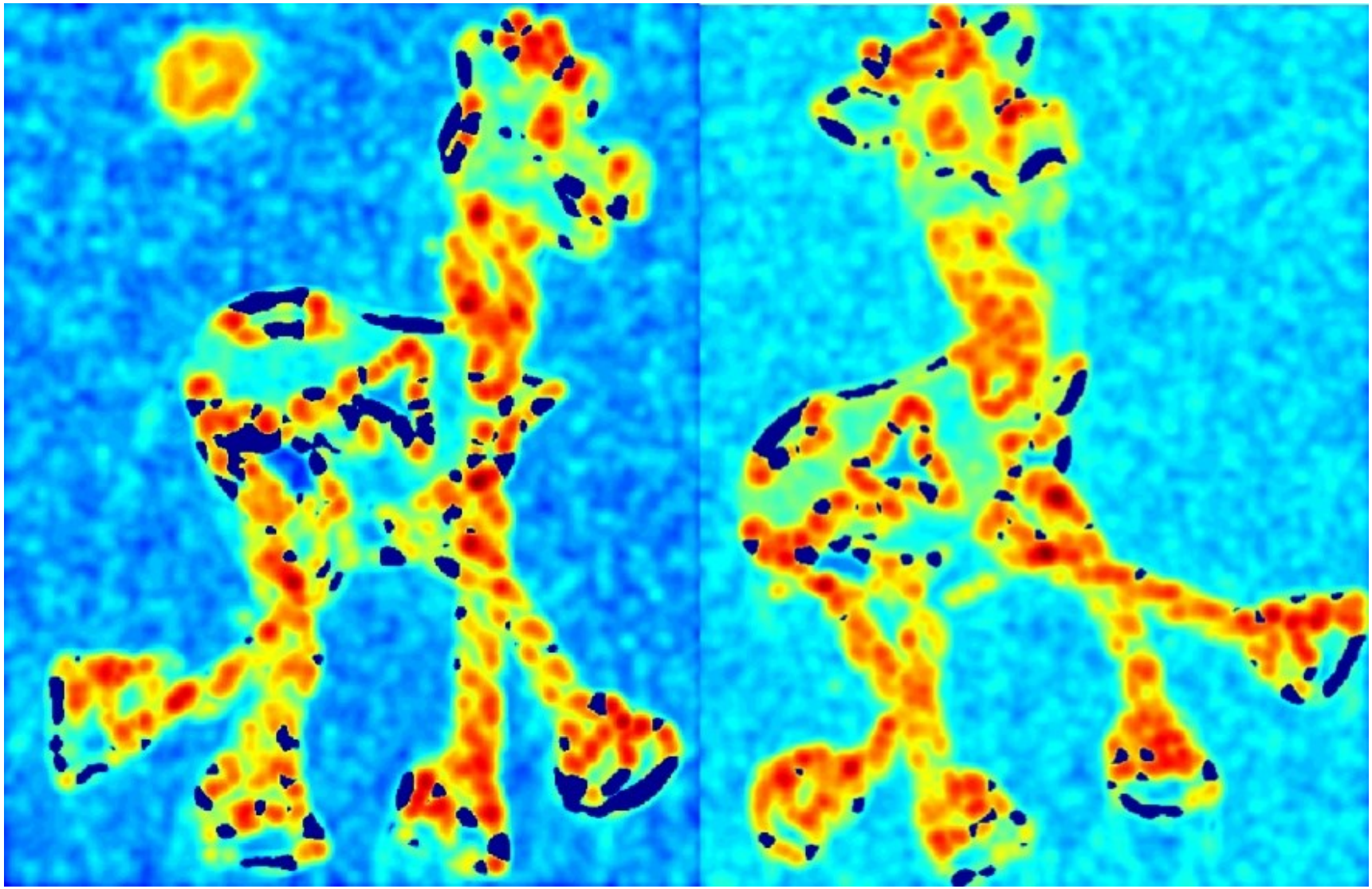
1. The Algorithm:
   1. Compute partial derivatives I_x, I_y
   2. Compute: $a = \sum_W {I_x}^2,\ b = \sum_W I_x I_y,\ \mathrm{c} = \sum_W {I_y}^2$
   3. Compute corner response R
   4. Find local maxima in R

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

Harris Detector: Workflow

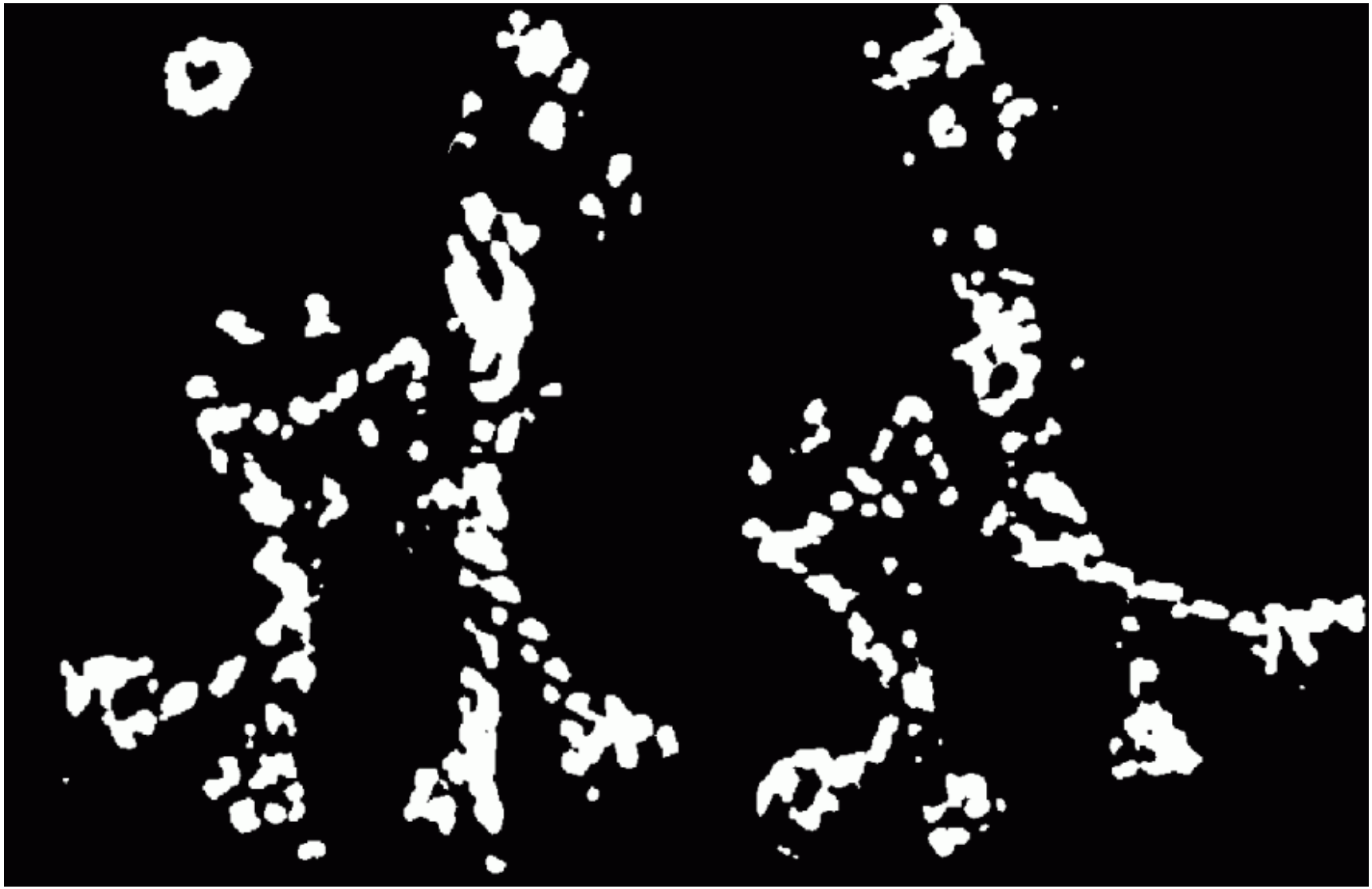
Harris Detector: Workflow

Compute corner response R



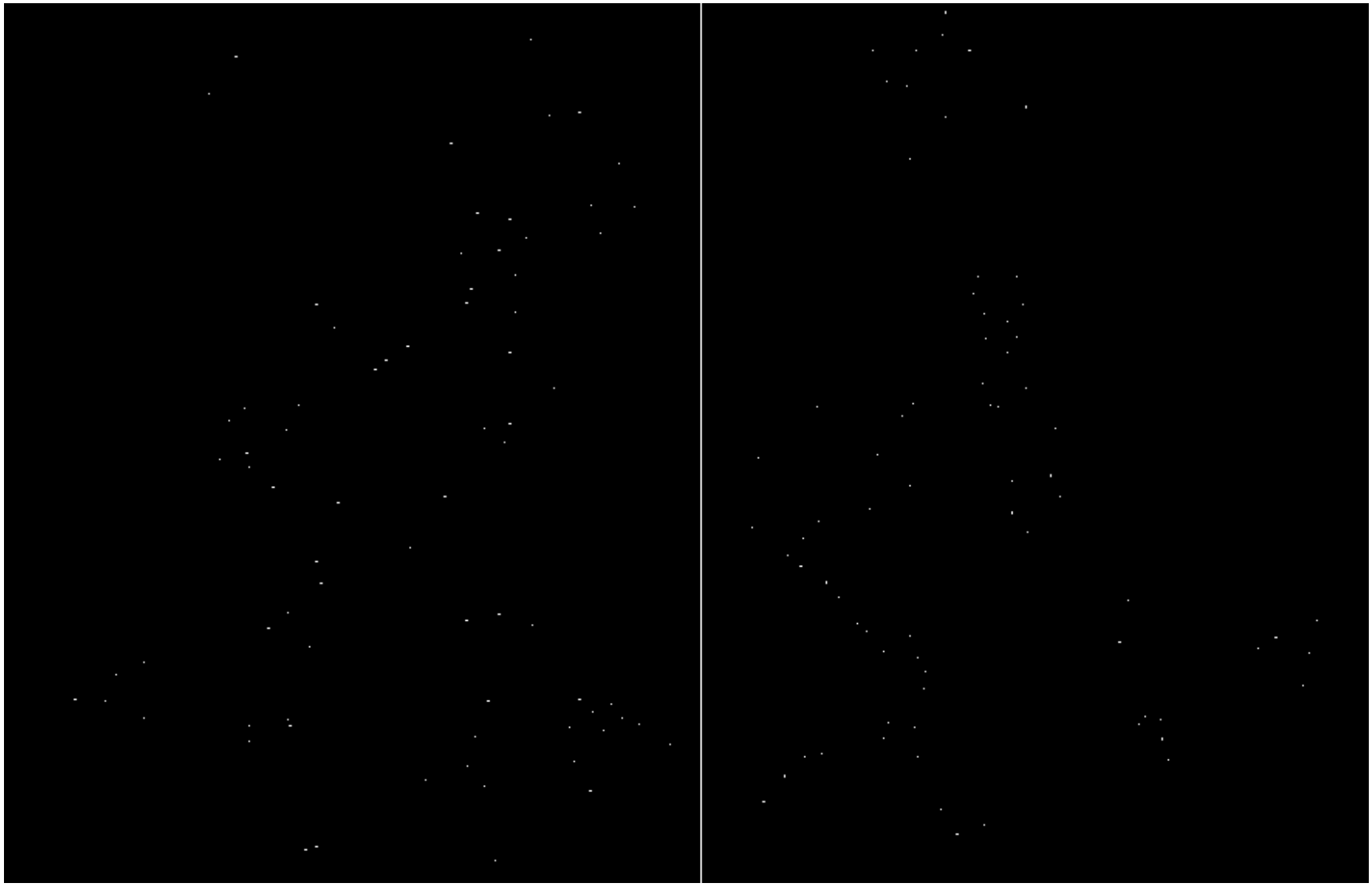
Harris Detector: Workflow

Find points with large corner response: R>threshold



Harris Detector: Workflow

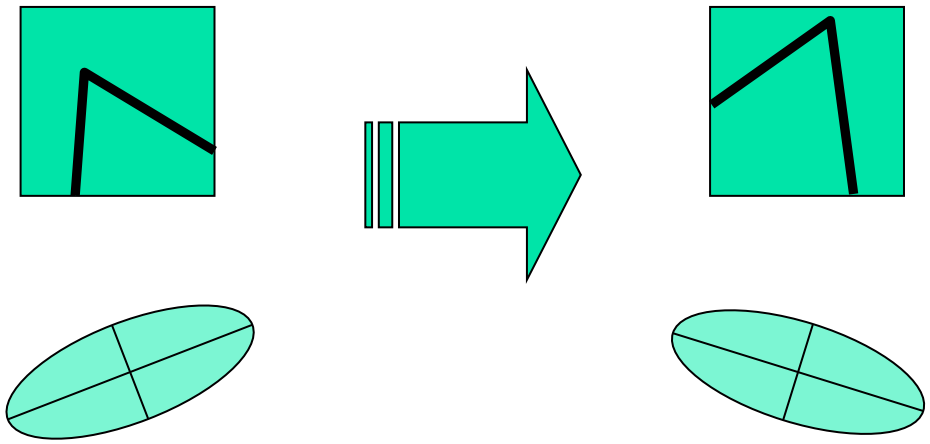
Take only the points of local maxima of R



Harris Detector: Workflow

Harris Detector: Properties

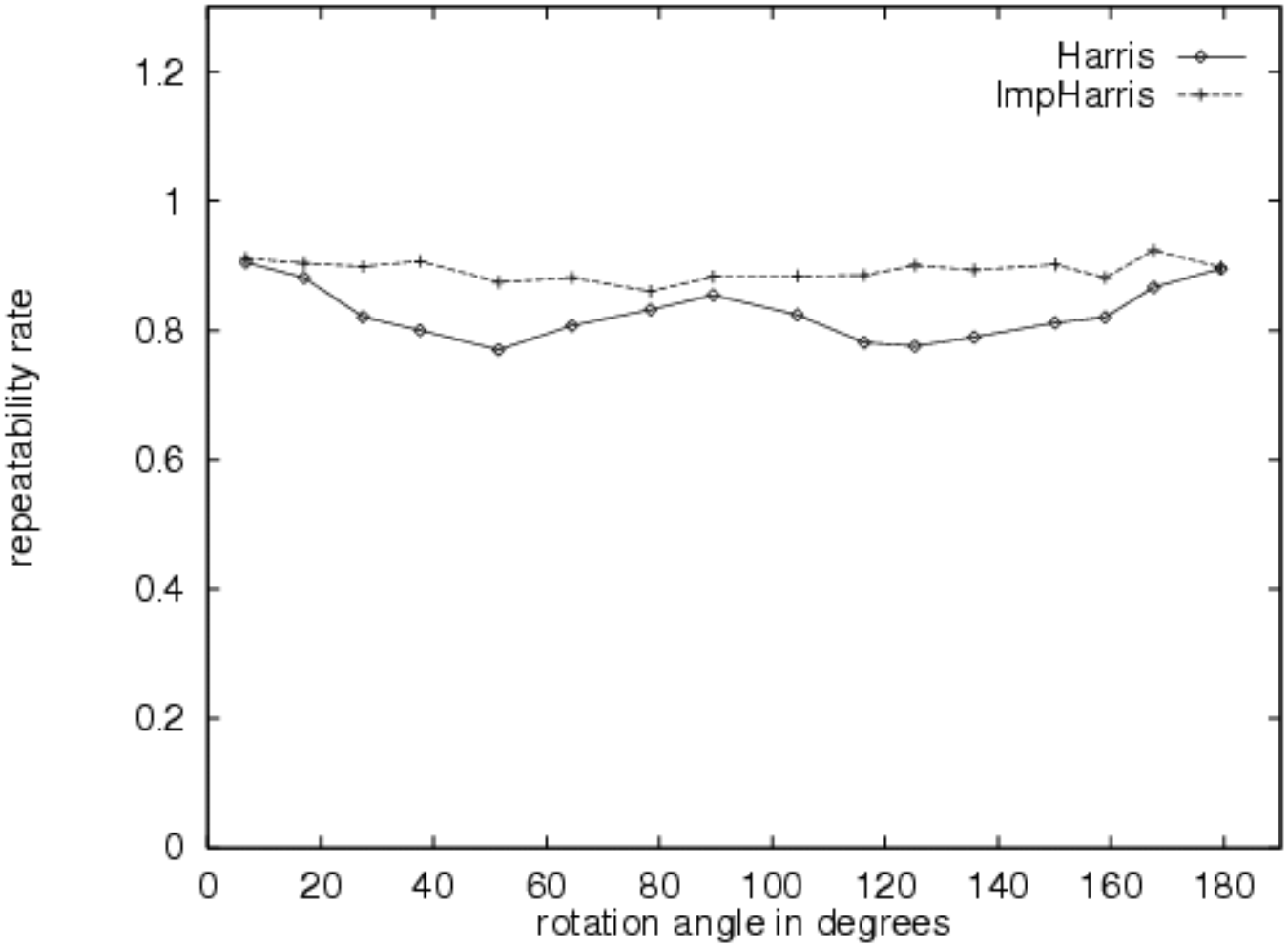
1. Rotation invariance



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

Corner response R is invariant to image rotation

Rotation Invariance of Harris Detector



C.Schmid et.al. "Evaluation of Interest Point Detectors". IJCV 2000

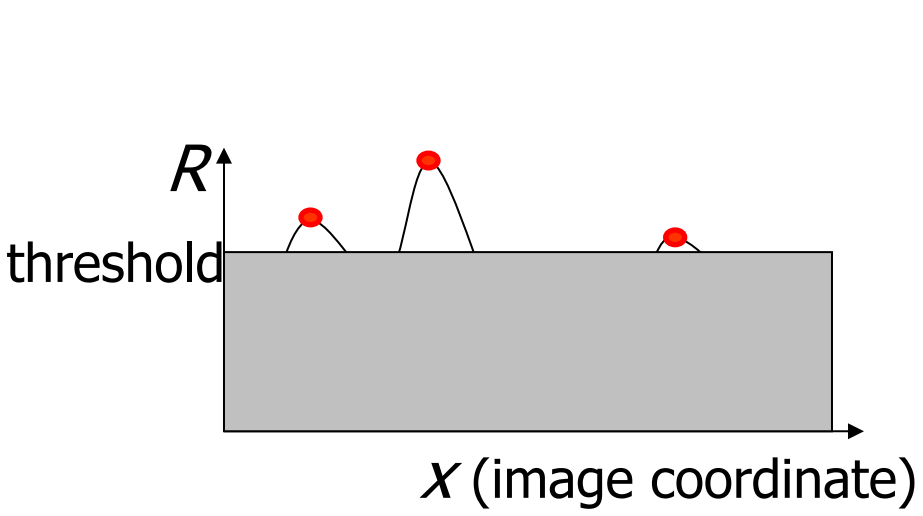
Harris Detector: Intensity change

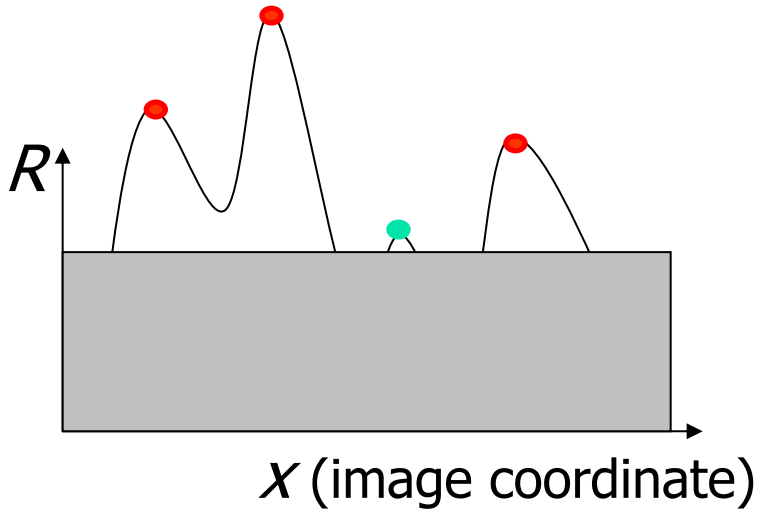
1. Partial invariance to additive and multiplicative intensity changes

   ✓ Only derivatives are used $\Rightarrow$

   invariance to intensity shift $I \rightarrow I + b$

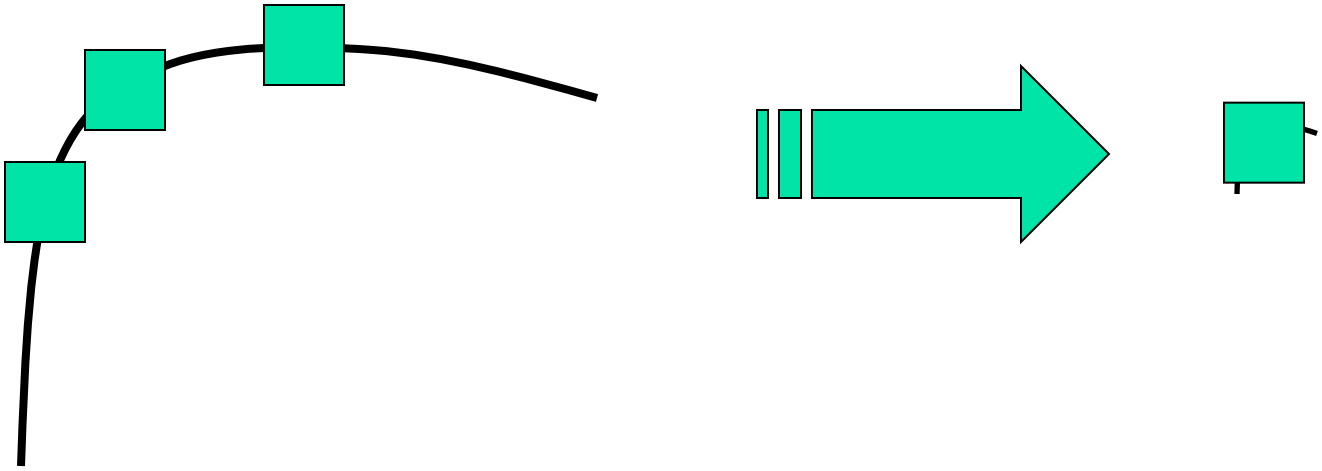
   ? Intensity scale: $I \rightarrow a\, I$





Harris Detector: Scale Change

1. Not invariant to *image scale*!



All points will be classified as edges

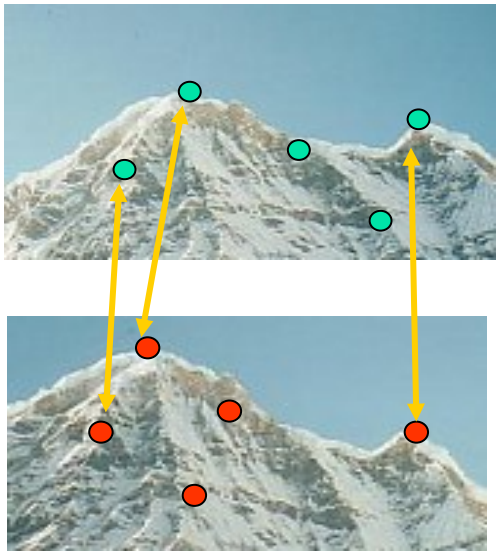
Corner !

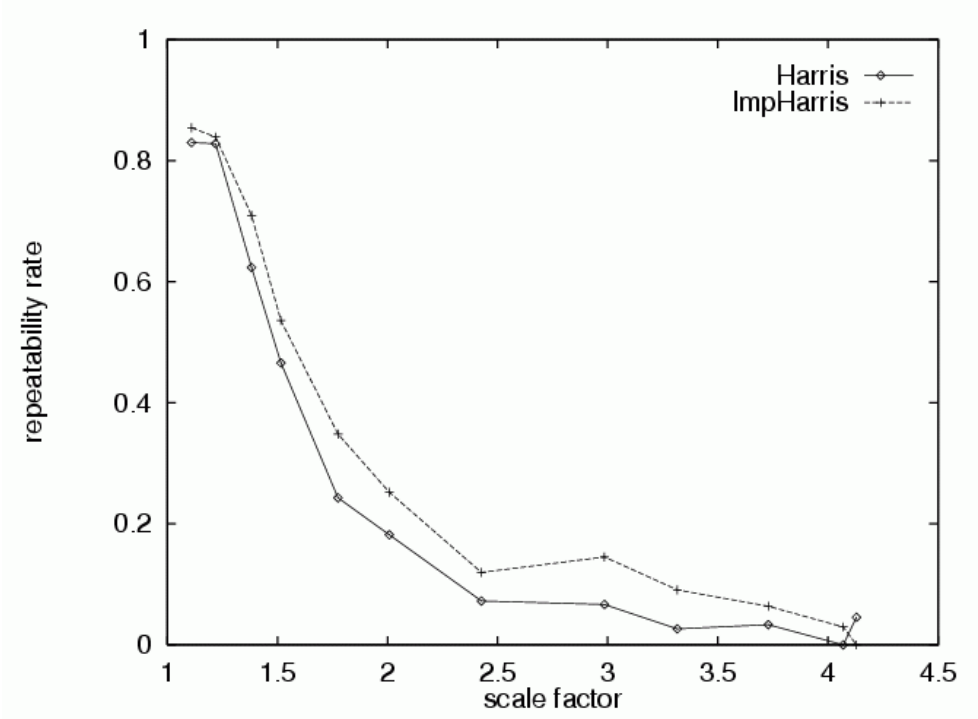
Harris Detector: Scale Change

1. Quality of Harris detector for different scale changes

Repeatability rate:

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





C.Schmid et.al. "Evaluation of Interest Point Detectors". IJCV 2000

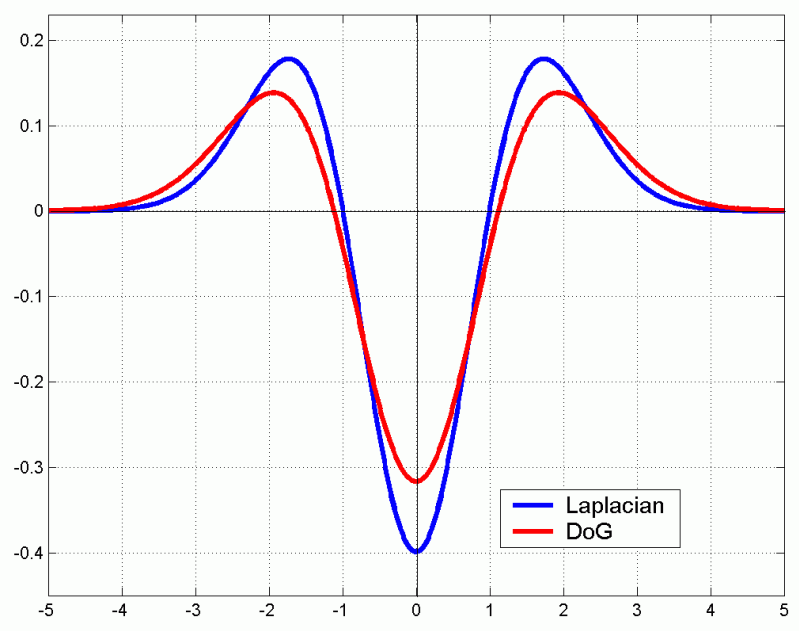
Laplacian of Gaussian detector

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

(Laplacian)

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

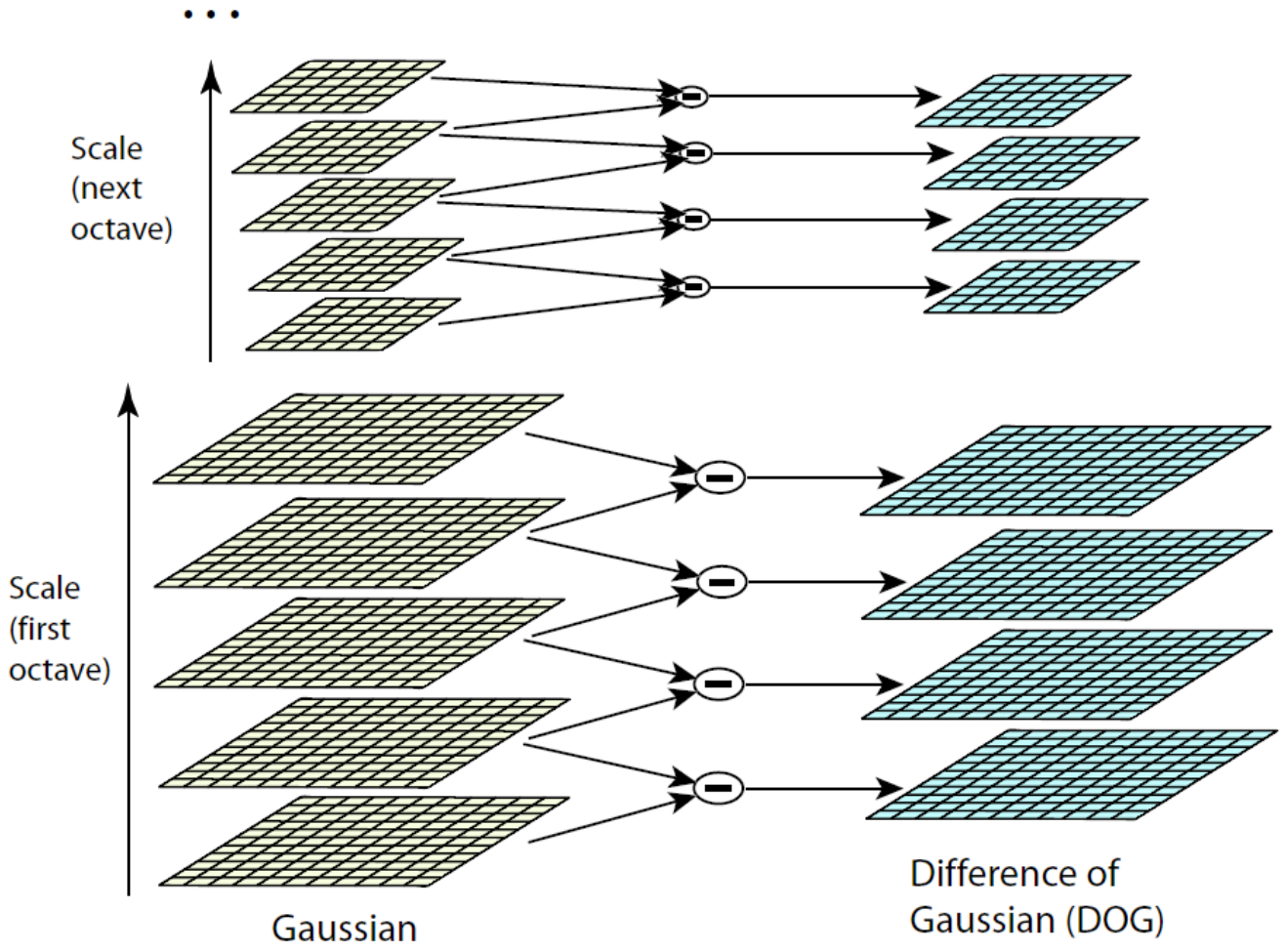
(Difference of Gaussians)



where Gaussian

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

Lowe's : DoG


Scale
(next
octave)
Scale
(first
octave)
Gaussian
Difference of
Gaussian (DOG)

Hessian

finds local maxima of the determinant of the image Hessian:

$$\mathbf{H} = \begin{bmatrix} D_{xx}(x, y; \sigma) & D_{xy}(x, y; \sigma) \\ D_{xy}(x, y; \sigma) & D_{yy}(x, y; \sigma) \end{bmatrix}$$

Hessian (compared to Harris):

- is faster
- has less paramters
- typically has more keypoint

but

- Harris is less influenced by noise, it uses only 1^{st} dervitives
- Harris is more accurate.

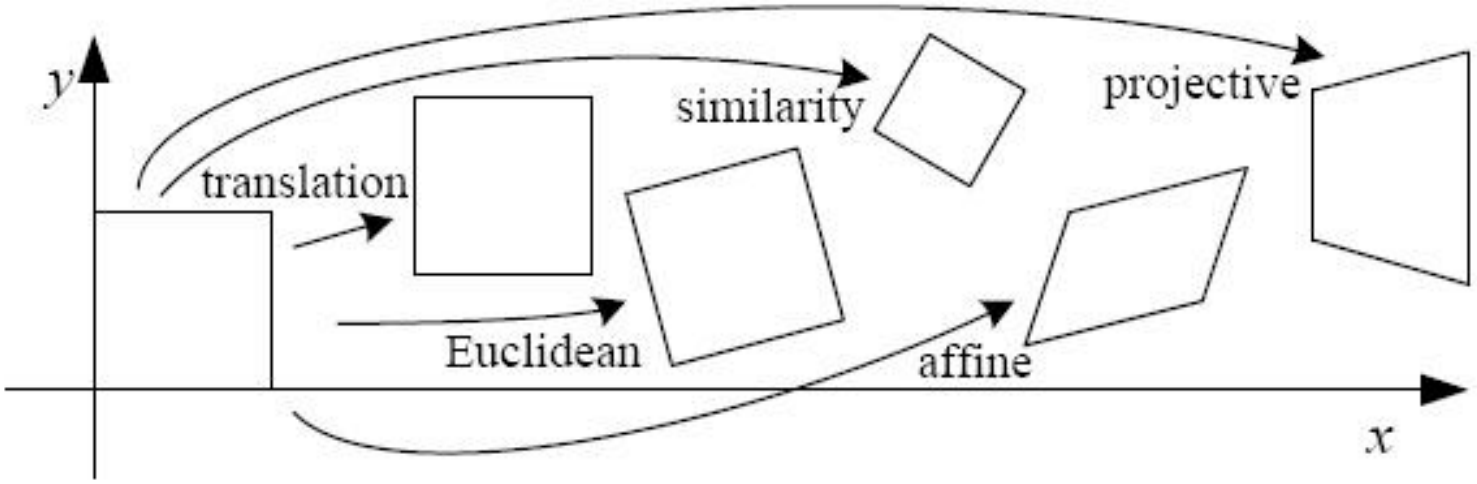
Hessian (compared to Laplace = DoG)

- superior in all respects, only slightly slower

Geometric transformations

1. Translation
2. Euclidean (translation + rotation)
3. Similarity (transl. + rotation + scale)
4. Affine transformations
5. Projective transformations

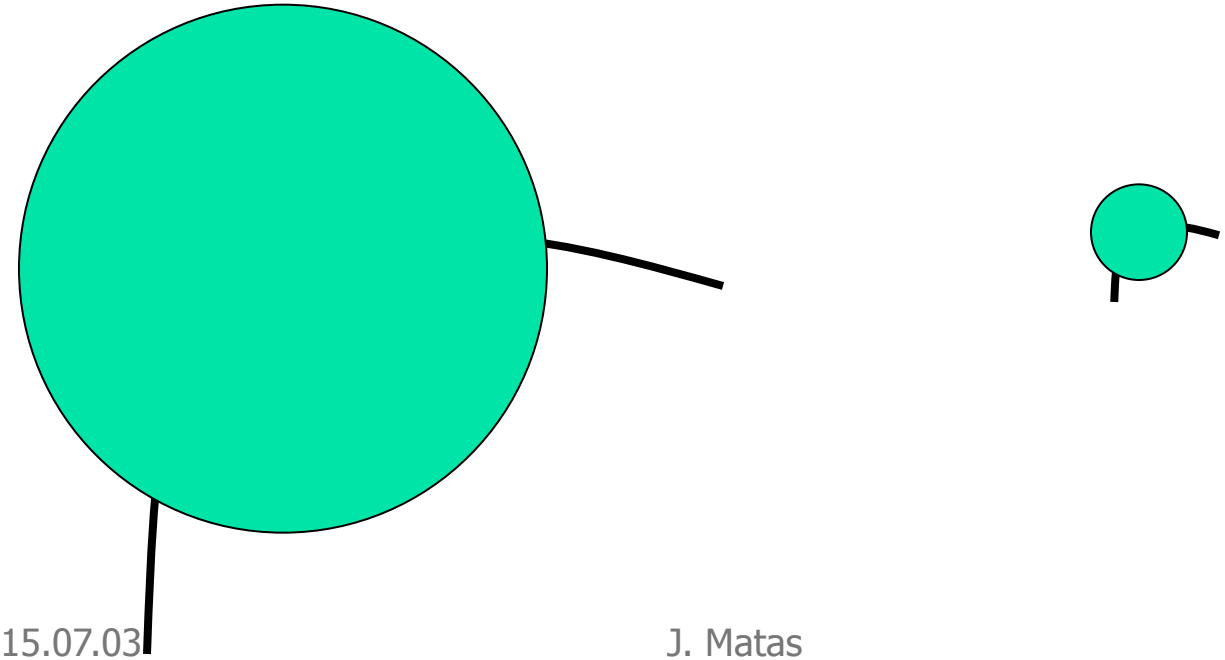
Only holds for planar patches!



Scale Covariant Detectors

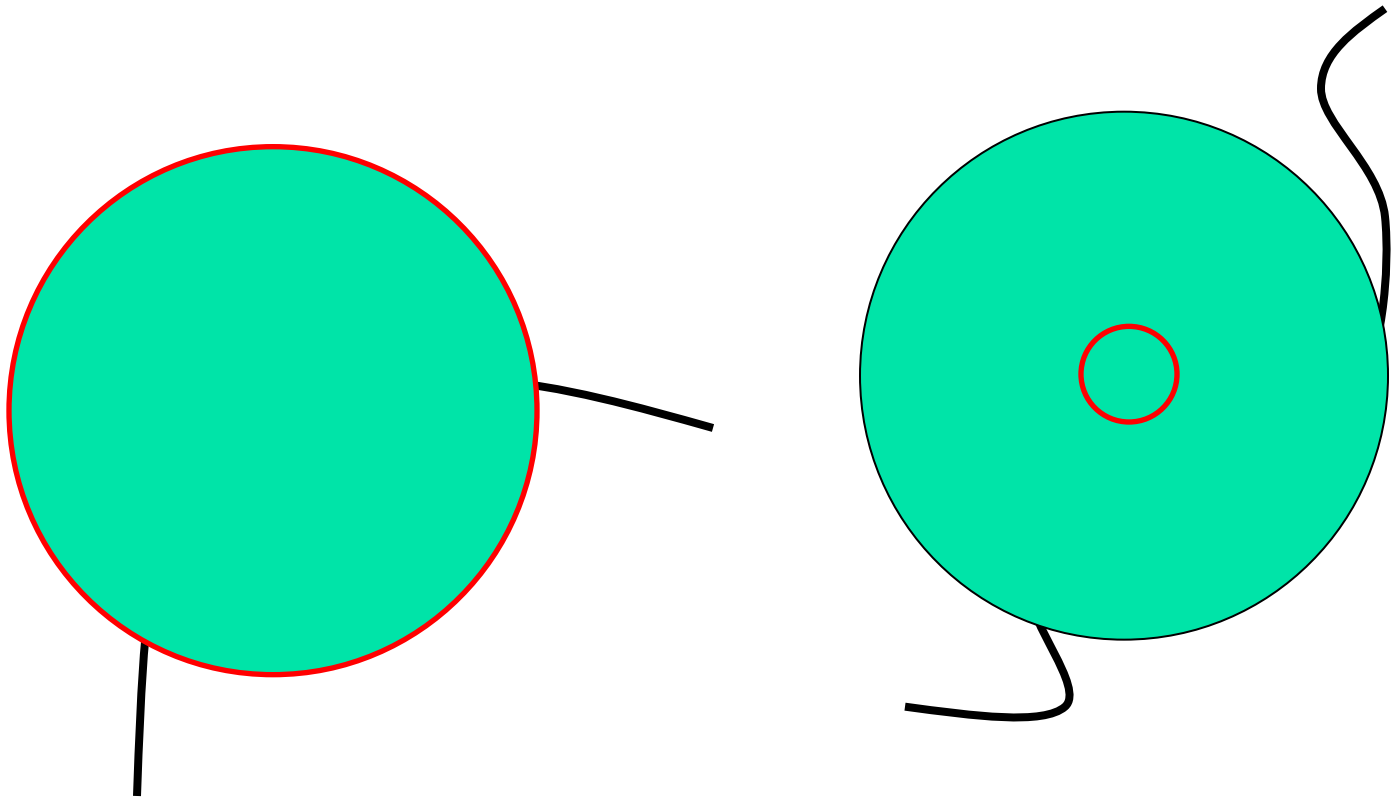
A possible approach (close to brute force search):

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



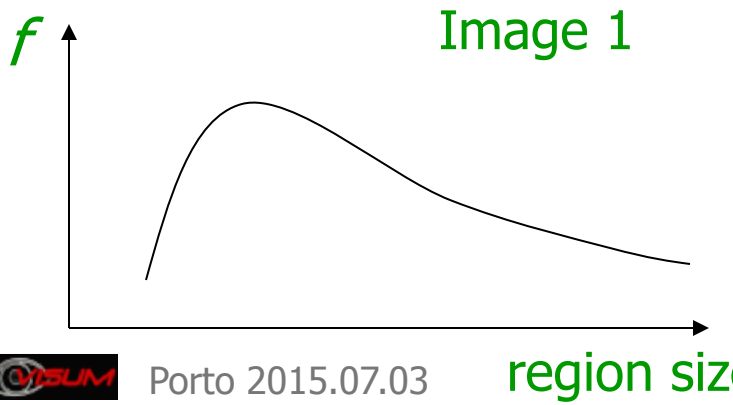
Scale Invariant Detection

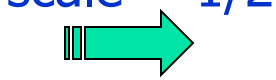
1. The problem: how do we choose corresponding circles *independently* in each image?

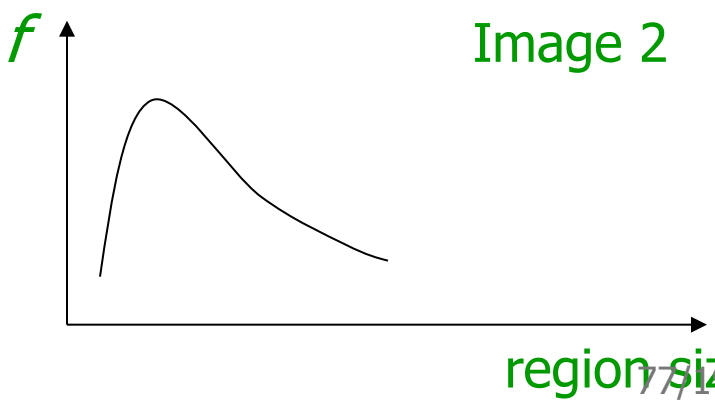


Scale Covariant Detectors

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

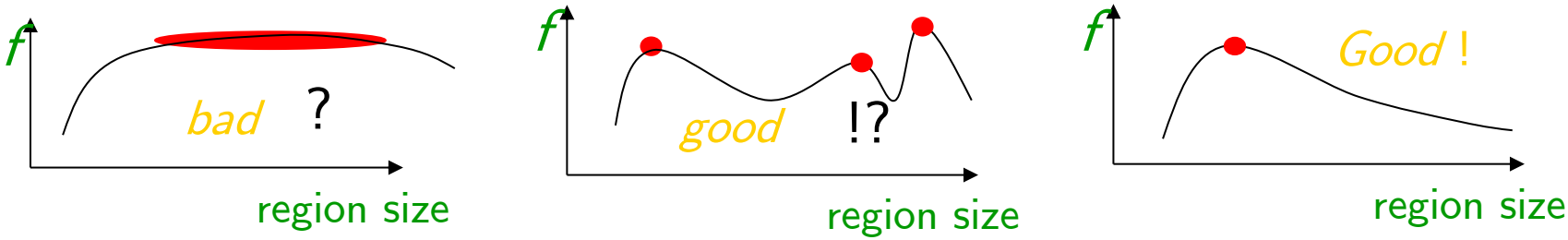






Scale Invariant Detection

1. A “good” function for scale detection:
   has (one ?) stable sharp peak(s)



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

Scale Invariant Detection

1. Observation: functions based on normalized second order Gaussian derivatives are suitable scale selectors
   1. Determinant of the Hessian matrix

$$H(x, y, \sigma) = N_{xx}(x, y, \sigma)N_{yy}(x, y, \sigma) - N_{xy}(x, y, \sigma)N_{yx}(x, y, \sigma)$$

   2. Laplacian - trace of the Hessian matrix

$$L(x, y, \sigma) = N_{xx}(x, y, \sigma) + N_{yy}(x, y, \sigma)$$

   have interesting properties when computed with increasing value of σ

1. Normalized derivatives compensate the attenuation of the signal

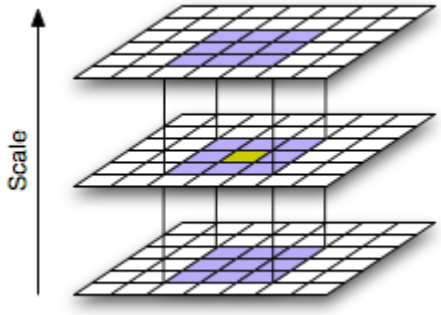
$$N_{ab}(x, y, \sigma) = \sigma^2 G_{ab}(x, y, \sigma)$$

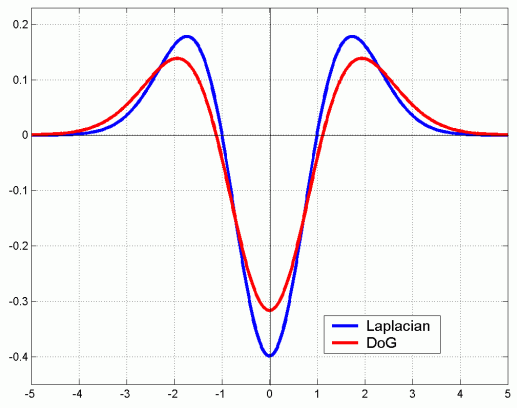
   where

$$G_{ab}(x, y, \sigma) = \frac{\partial^2 f}{\partial a \partial b} G(x, y, \sigma) \quad G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Scale Invariant Detectors

1. Scale invariant detectors, find local extrema (both in space and scale) of Laplacian and determinant of Hessian response in gaussian scalespace.



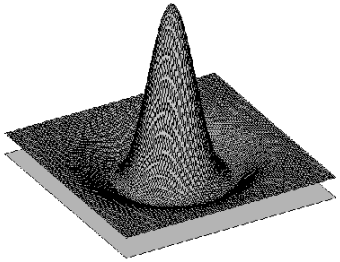


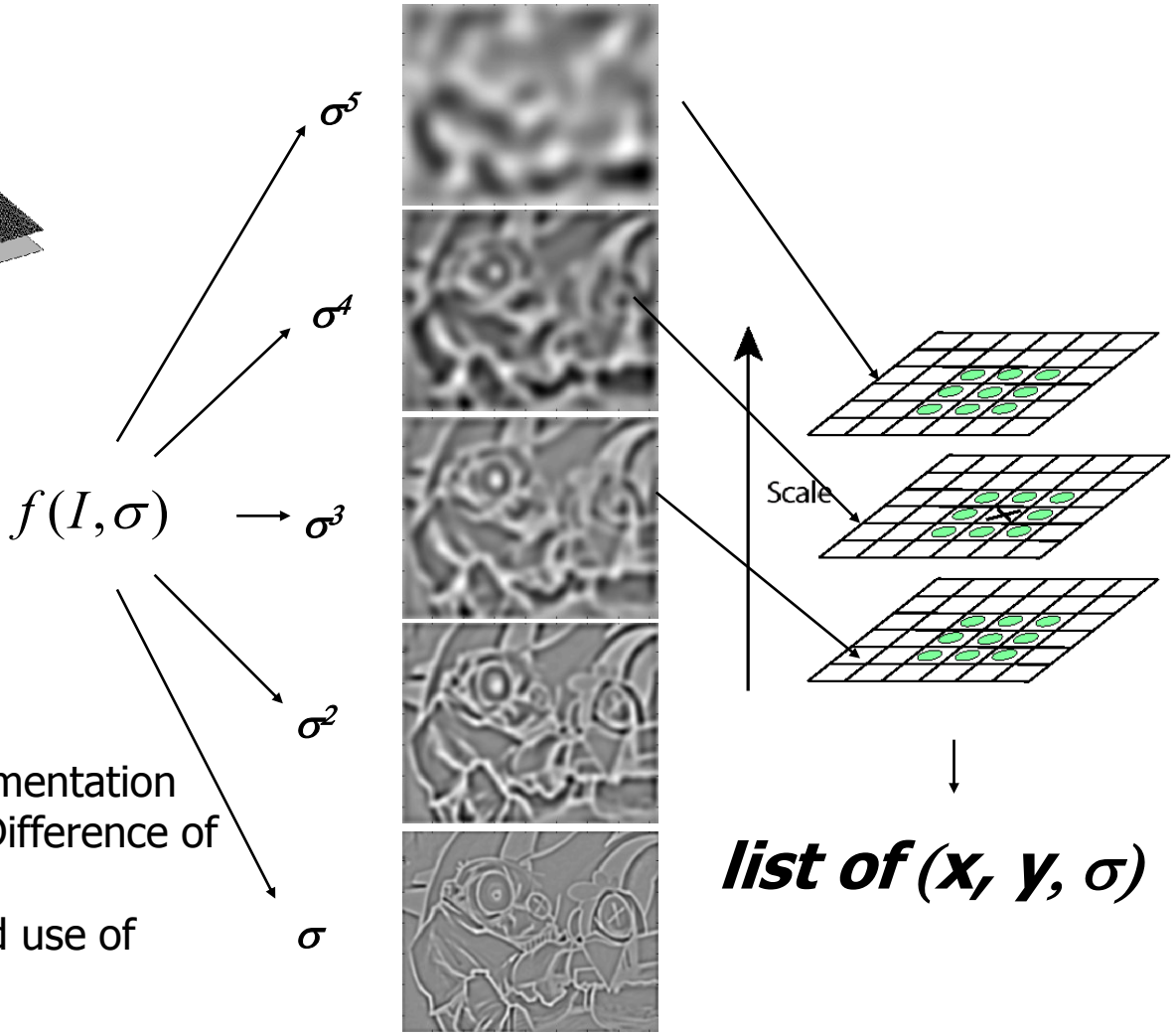
1. Faster detection?
   1. Scalespace pyramid[1], GPU implementation
   2. Approximation of Laplacian by Difference of Gaussians[1]
   3. Approximation by box filters and use of integral images

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

Scale invariant detectors

- Scale invariant detectors, find local extrema (both in space and scale) of Laplacian and determinant of Hessian response in gaussian scalespace.



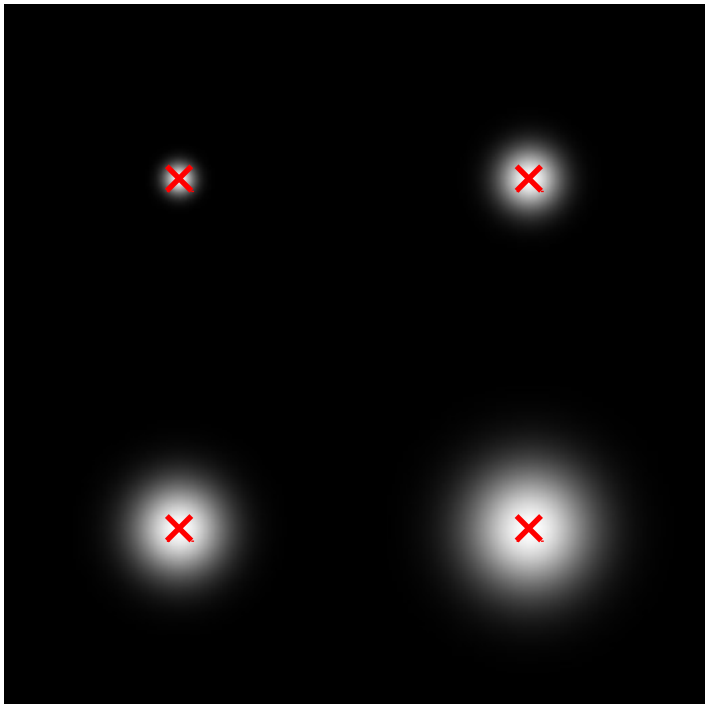


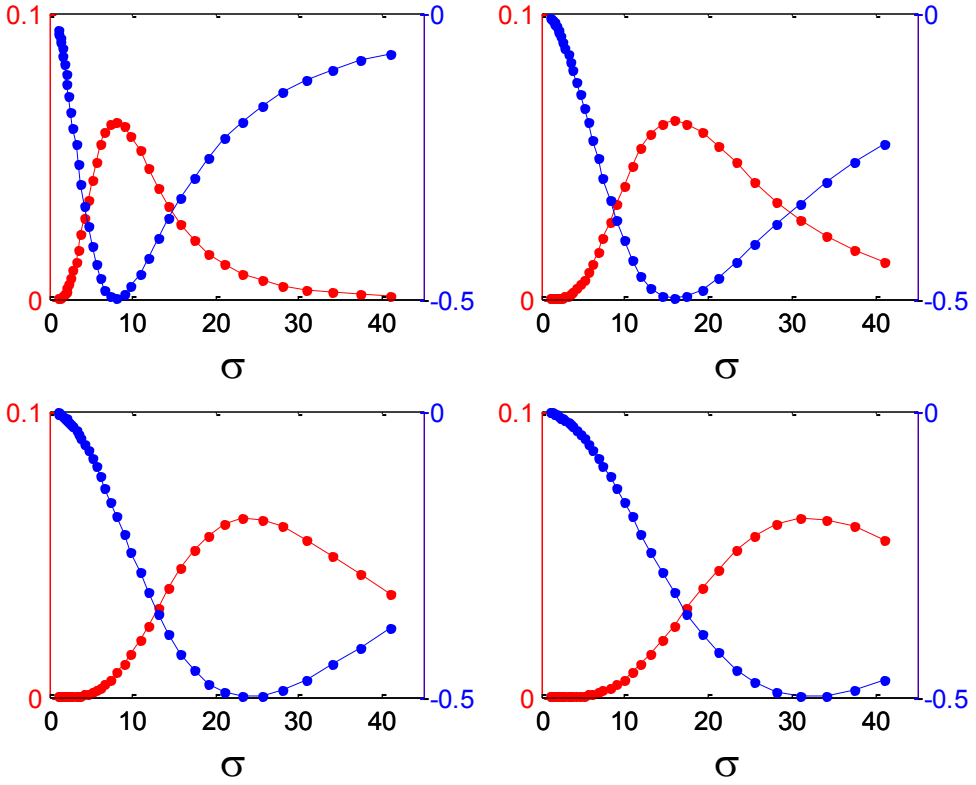
Faster detection?

- Scalespace pyramid, GPU implementation
- Approximation of Laplacian by Difference of Gaussians
- Approximation by box filters and use of integral images

Automatic Scale Selection

1. Gaussian scalespace, “stack of gradually smoothed versions” of original image
2. Response of Laplacian and the determinant of the Hessian on Gaussian blobs with standard deviations 8,16,24 and 32 in red x points of Gaussian scalespace

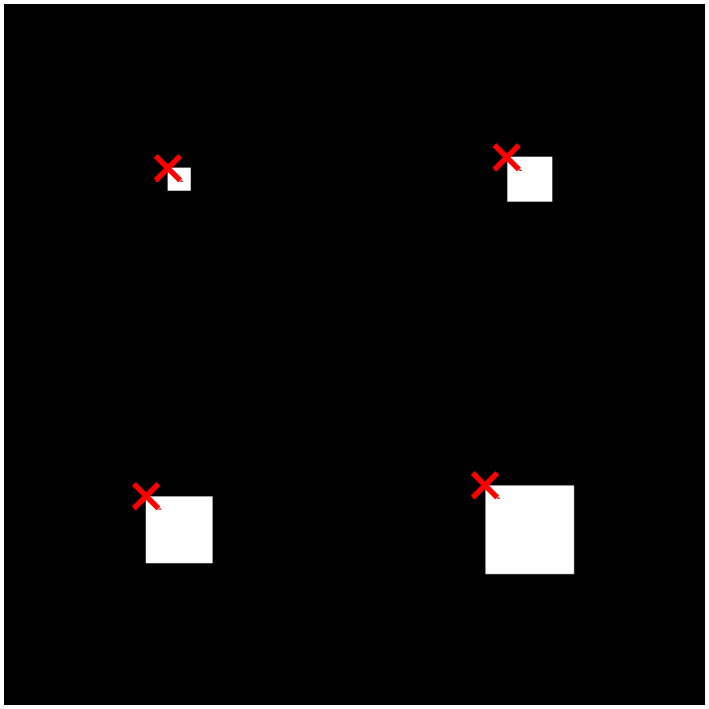


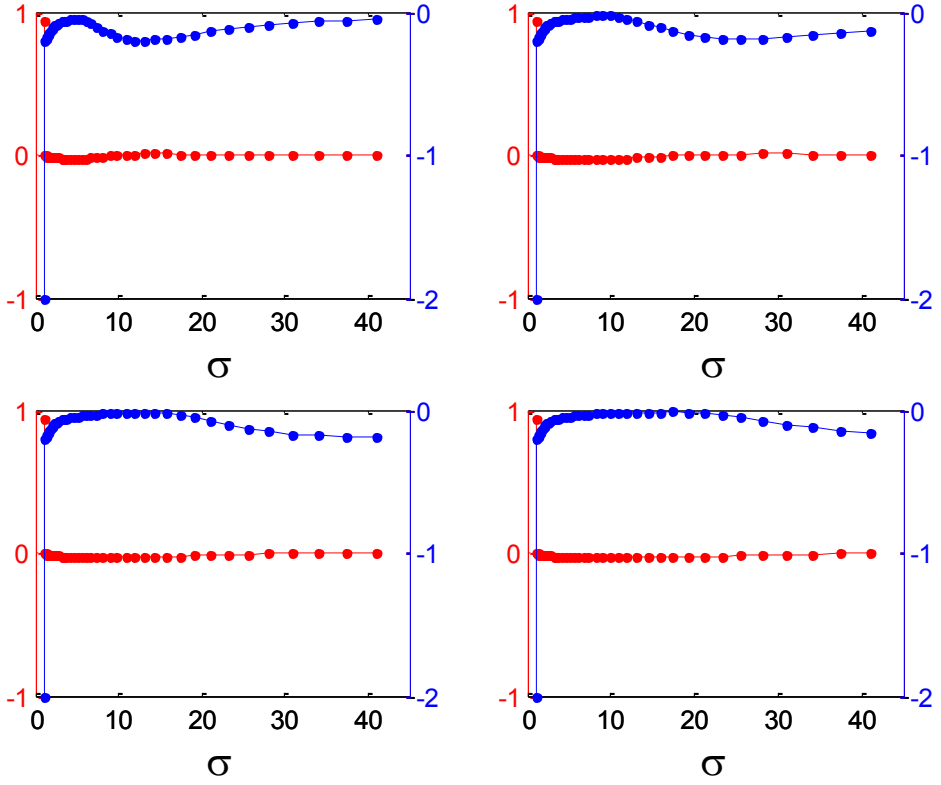


Automatic Scale Selection

CENTER FOR MACHINE PERCEPTION

1. Automatic scale selection breaks down on points “without” an intrinsic scale defined

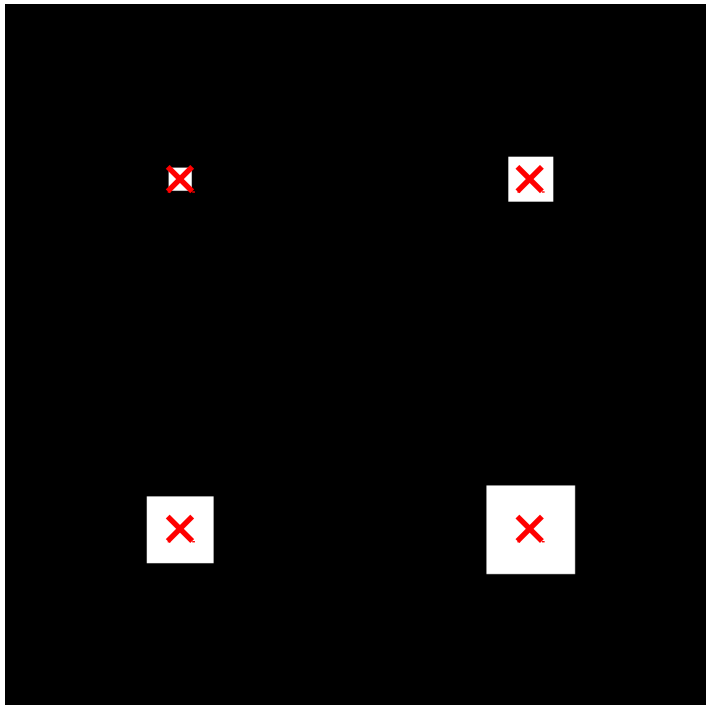


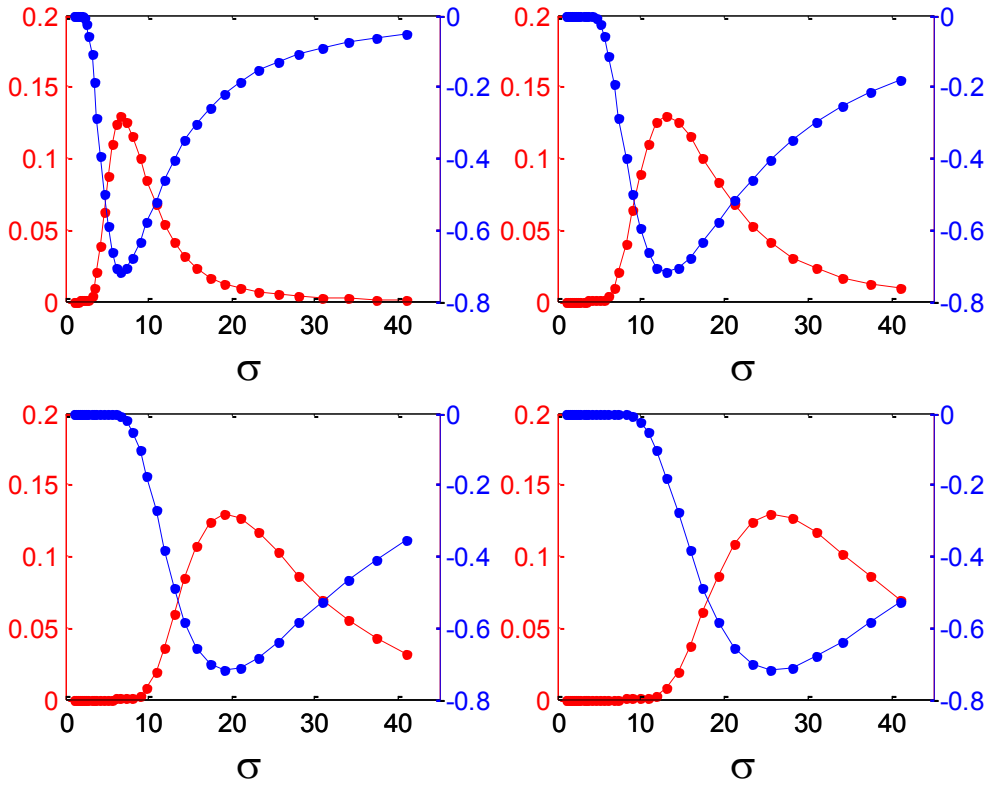


Automatic Scale Selection

CENTER FOR MACHINE PERCEPTION

1. Works for other structures as well
2. Response of Laplacian and determinant of Hessian on squares with sizes 17, 33, 49 an 65 in red x points of gaussian scalespace



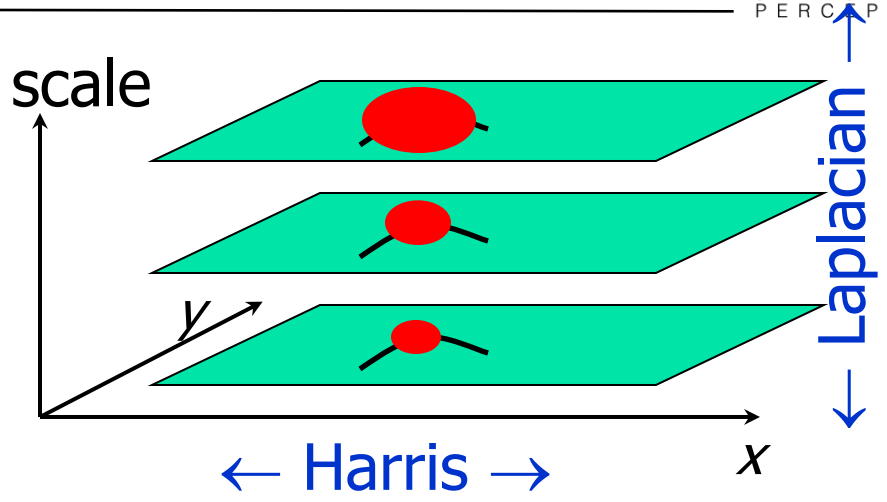


Scale Invariant Detectors

Harris-Laplacian[1]

Find local maximum of:

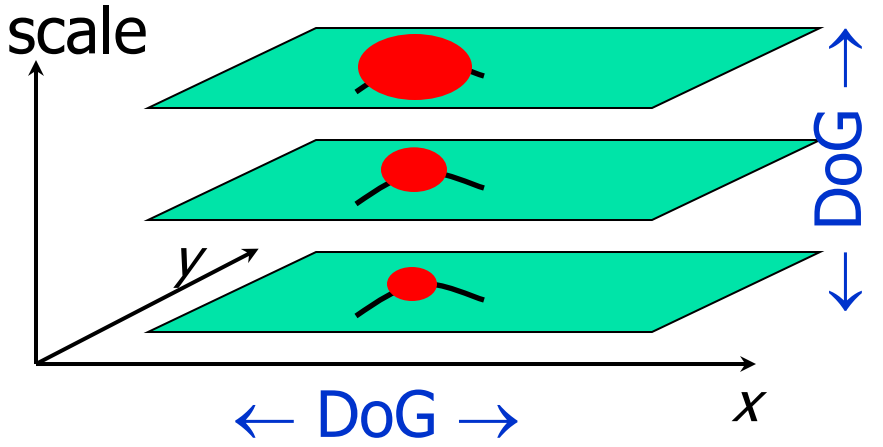
1. Harris corner detector in space (image coordinates)
2. Laplacian in scale



Laplacian-Laplacian = "SIFT" (Lowe)[2]

Find local maximum of:

- Difference of Gaussians in space and scale



Other options: Hessian, ...

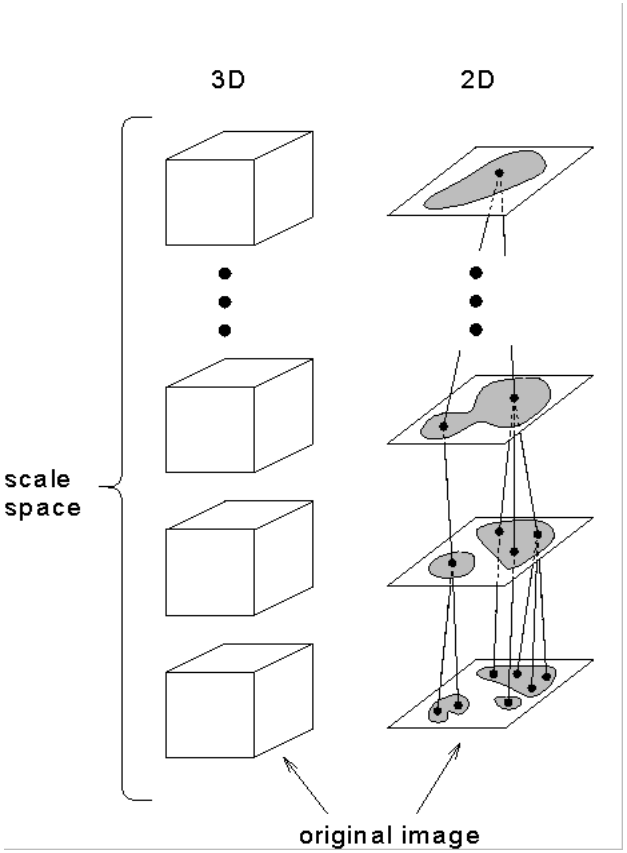
Harris does not work well for scale selection

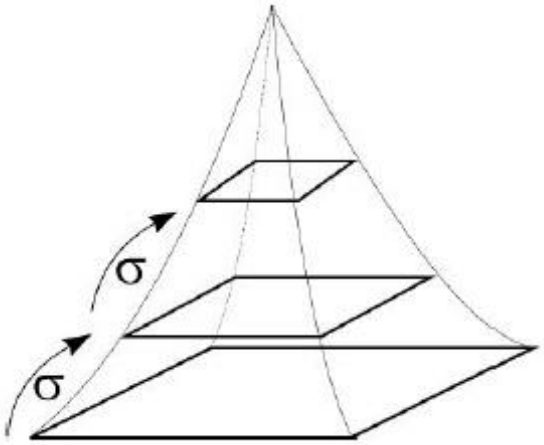
[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

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

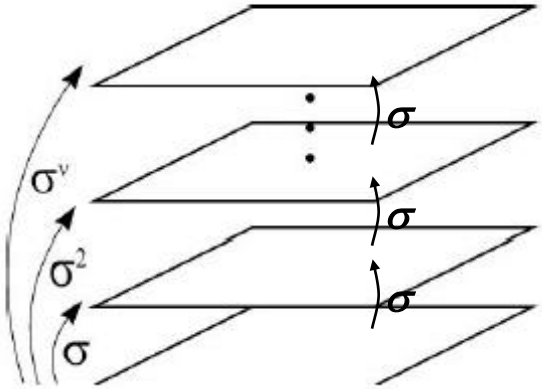
Scale-space representation

- Exponential increase of kernel size – uniform information change
- Kernel size and sampling interval are directly related.





Convolution with kernel of constant size and sampling



Convolution with kernel of increasing size

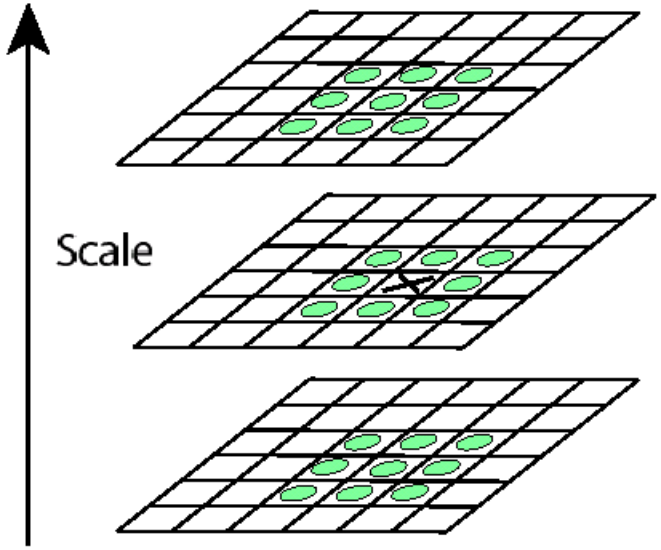
Sub-pixel/ Sub-level Keypoint Localization

- 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:

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

te

differences for derivatives):

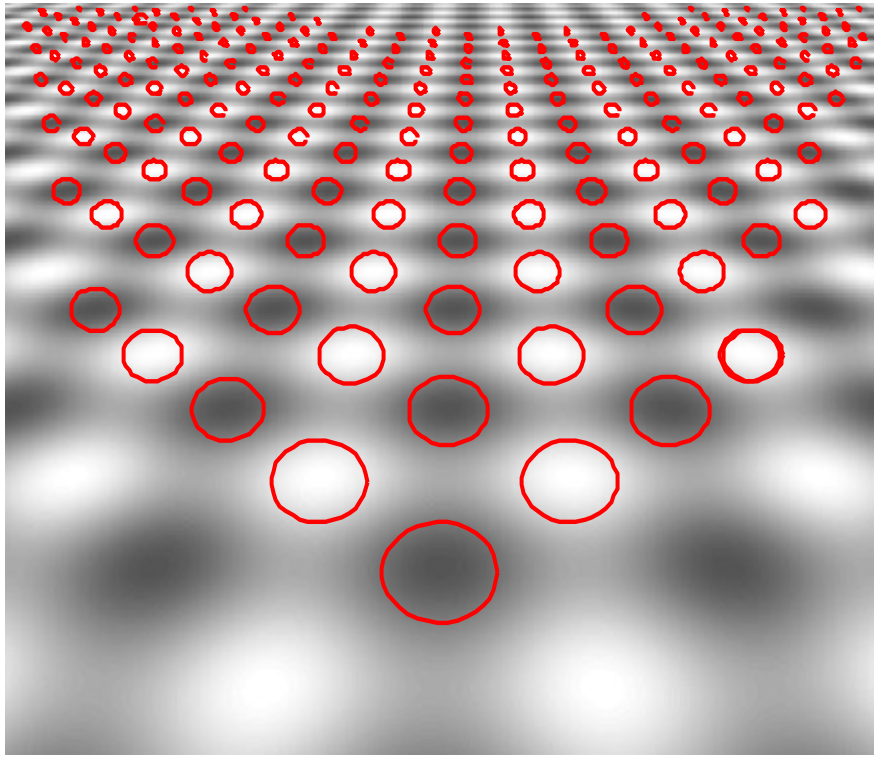


$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}}$$

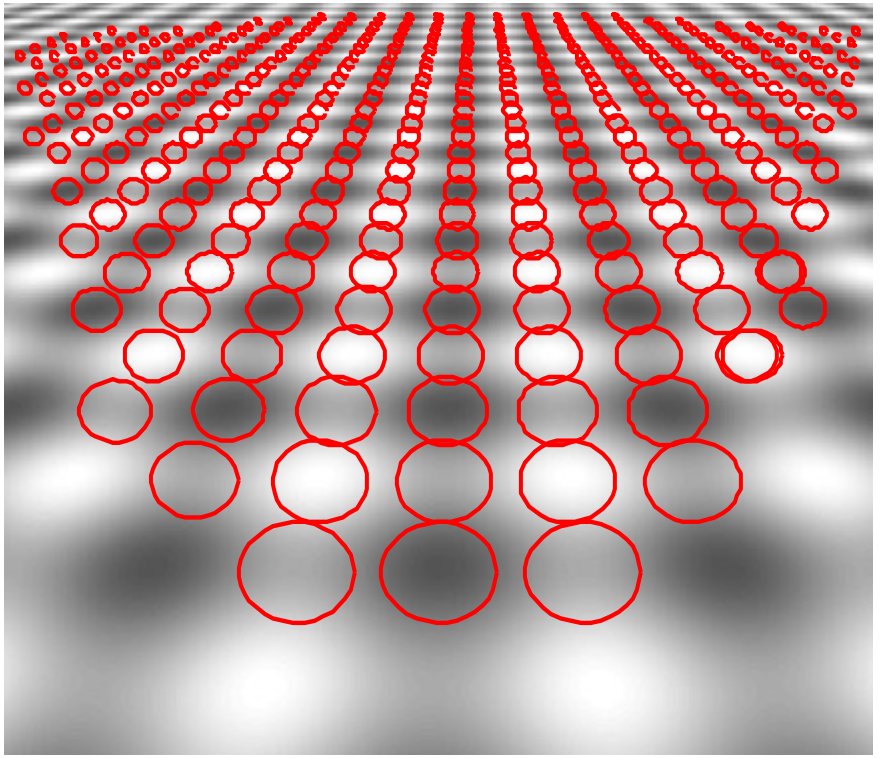
Scale Invariant Detection - Examples

CENTER FOR MACHINE PERCEPTION

Multiscale maxima and minima of Laplacian



Multiscale maxima and minima determinant of Hessian

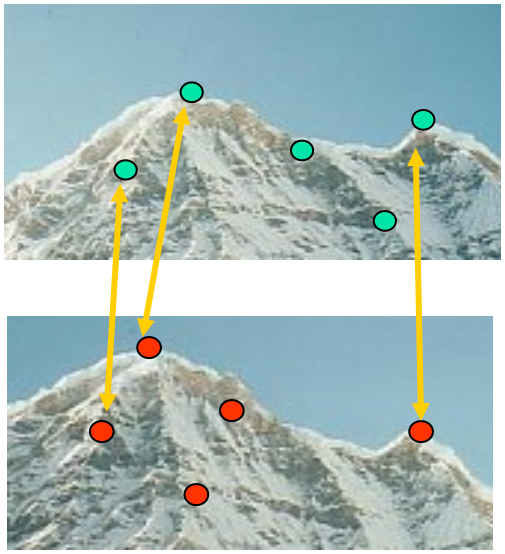


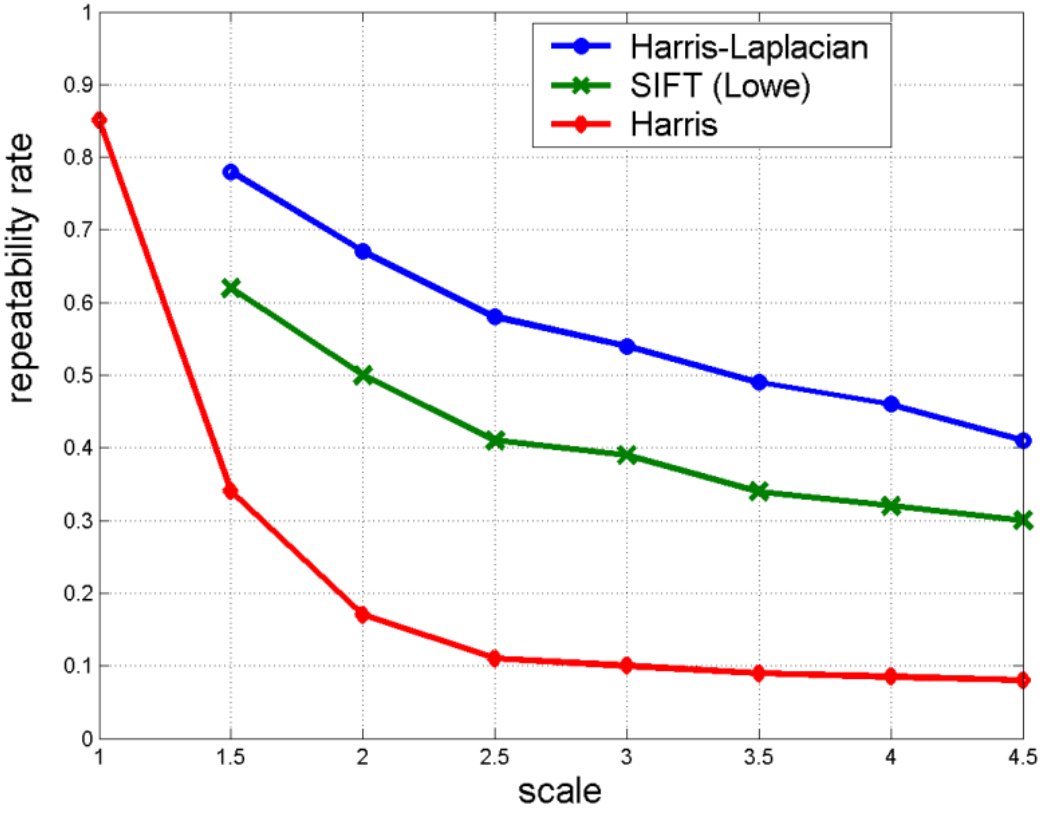
Scale Invariant Detectors

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

Repeatability rate:

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

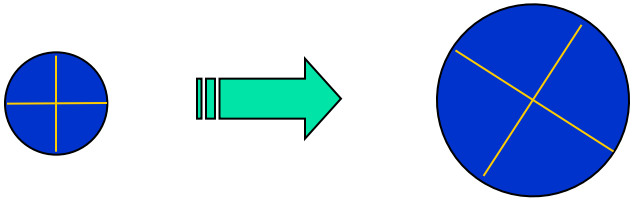




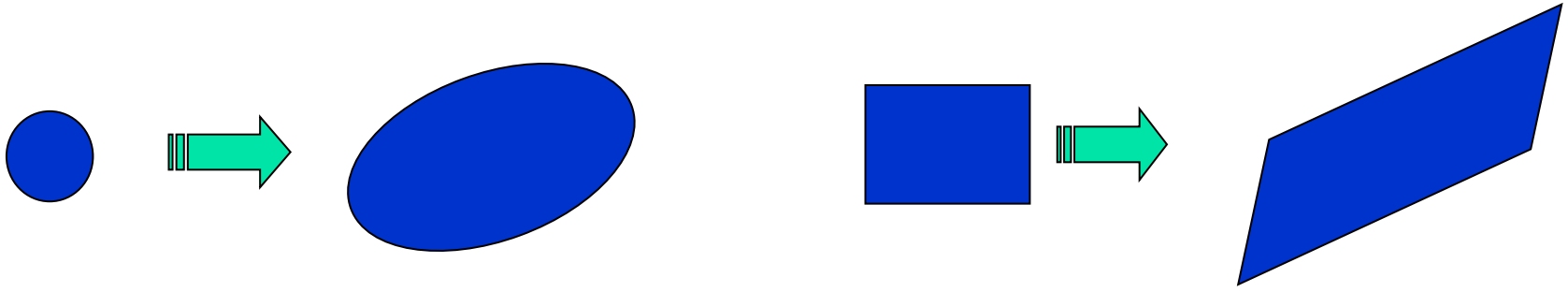
K. Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

Affine Invariant Detection

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

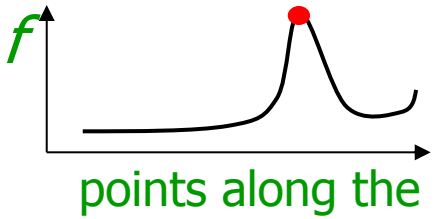


- Now we go on to:
  Affine transform (rotation + non-uniform scale)



Affine Invariant Detection

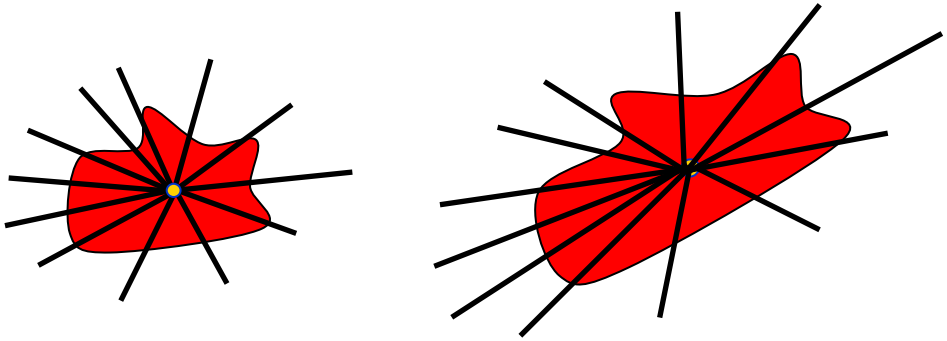
1. Take a local intensity extremum as initial point
2. Go along every ray starting from this point and stop when extremum of function f is reached



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

- We will obtain approximately corresponding regions

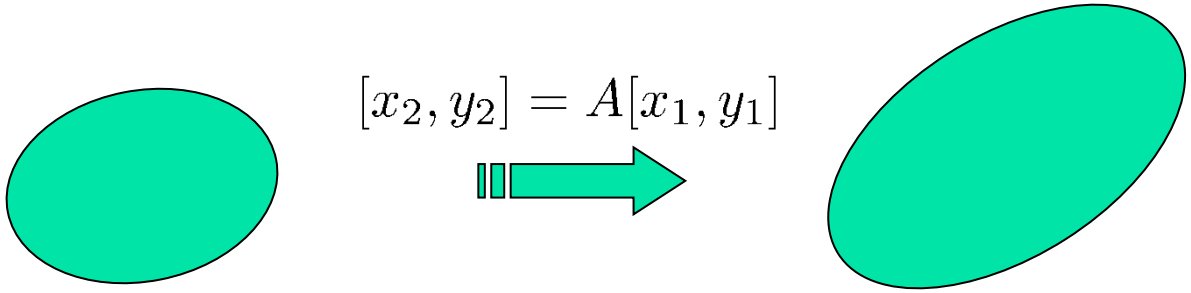
Remark: we search for scale in every direction



T.Tuytelaars, L.V.Gool. "Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions". BMVC 2000.

Affine Invariant Detection

- Covariance matrix of region points defines an ellipse:



$$[x_1, y_1]^T \sum_1^{-1} [x_1, y_1] = 1$$

$$\sum_1 = \langle [x_1, y_1][x_1, y_1]^T \rangle_{\text{region}_1}$$

$$[x_2, y_2]^T \sum_2^{-1} [x_2, y_2] = 1$$

$$\sum_2 = \langle [x_2, y_2][x_2, y_2]^T \rangle_{\text{region}_2}$$

$$\sum_2 = A \sum_1 A^T$$

Ellipses, computed for corresponding regions, also correspond!

Affine Covariant Detection

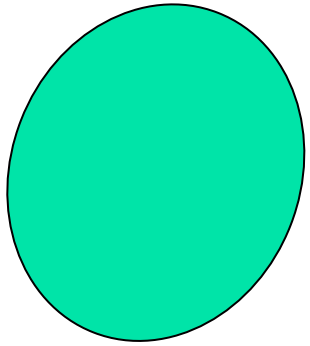
1. The regions found may not exactly correspond, so we approximate them with ellipses

- Geometric Moments:

$$m_{pq} = \int x^p y^q f(x,y) dxdy$$

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

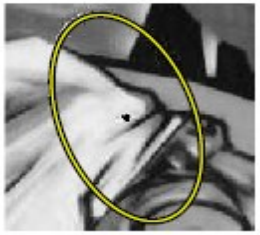
Taking f to be the characteristic function of a region (1 inside, 0 outside), moments of orders up to 2 allow to approximate the region by an ellipse



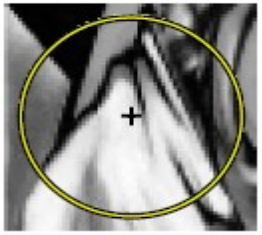
This ellipse will have the same moments of orders up to 2 as the original region

Harris/Hessian Affine Detector

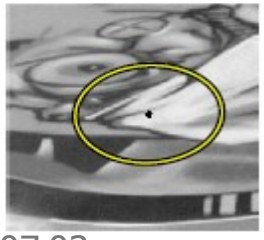
1. Detect initial region with Harris or Hessian detector and select the scale
2. Estimate the shape with the second moment matrix
3. Normalize the affine region to the circular one
4. Go to step 2 if the eigenvalues of the second moment matrix for the new point are not equal



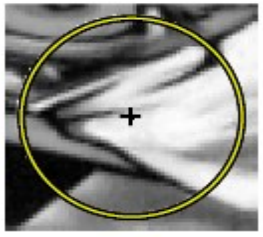
$$[x_1, y_1] \rightarrow M_1^{-1/2}[x'_1, y'_1]$$



$$[x'_1, y'_1] \rightarrow R[x'_2, y'_2]$$

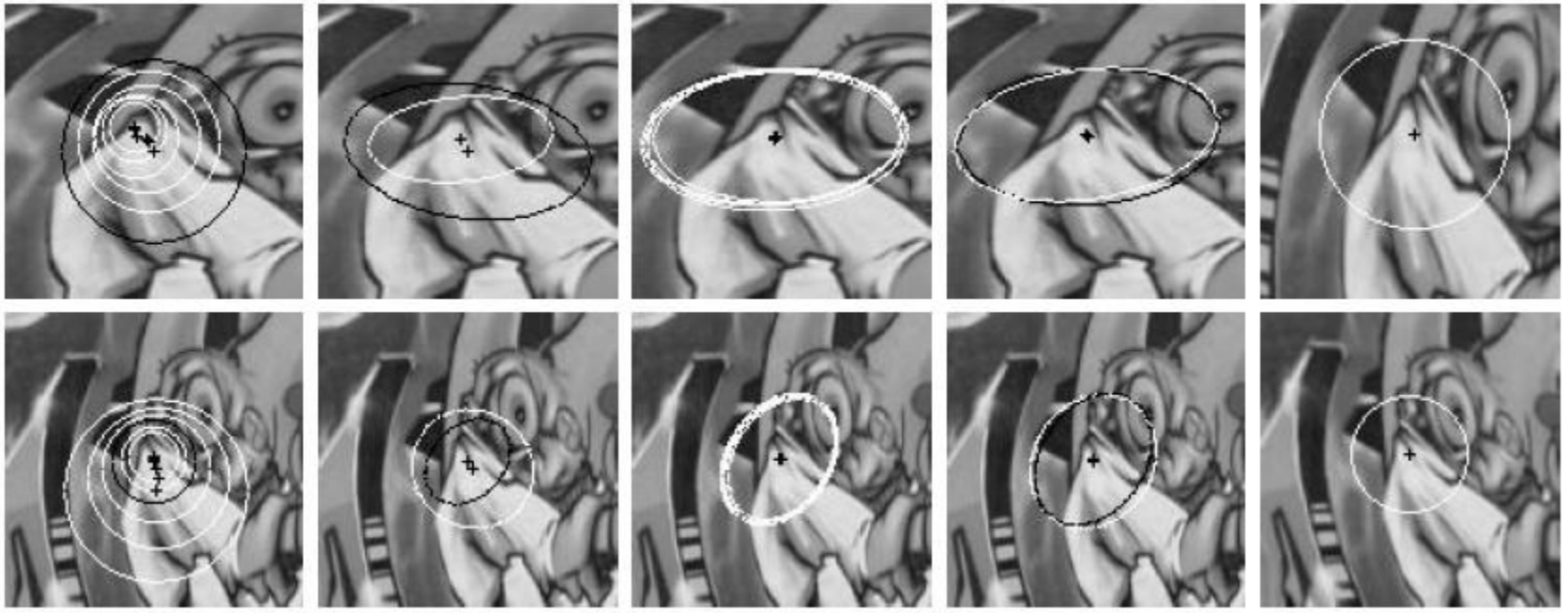


$$[x_2, y_2] \rightarrow M_2^{-1/2}[x'_2, y'_2]$$



Harris / Hessian Affine

1. Detect multi-scale Harris / Hessian points
2. Automatically select the scales
3. Adapt affine shape based on second order moment matrix
4. Refine point location

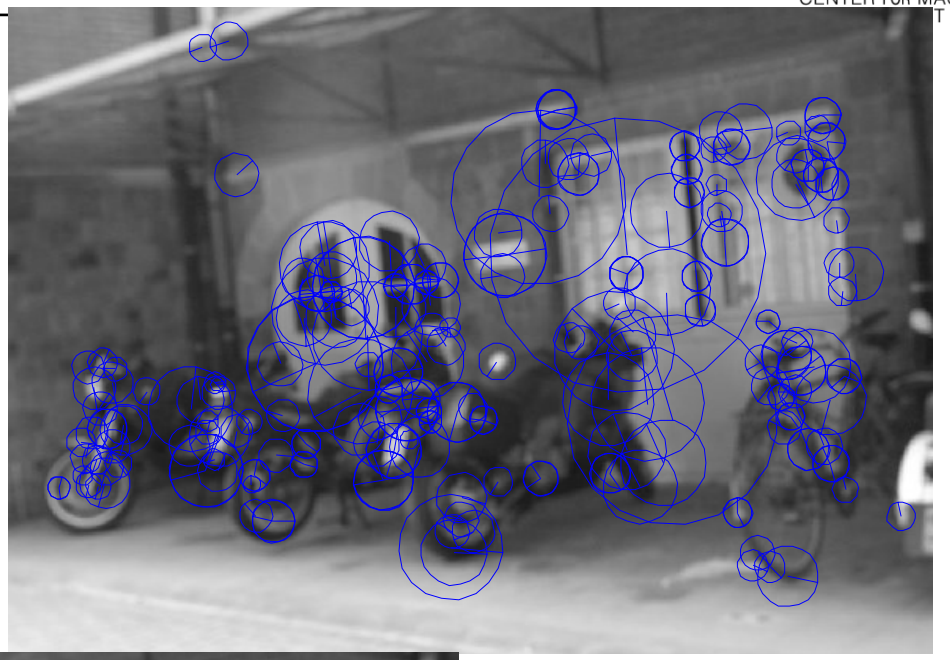


Harris Affine

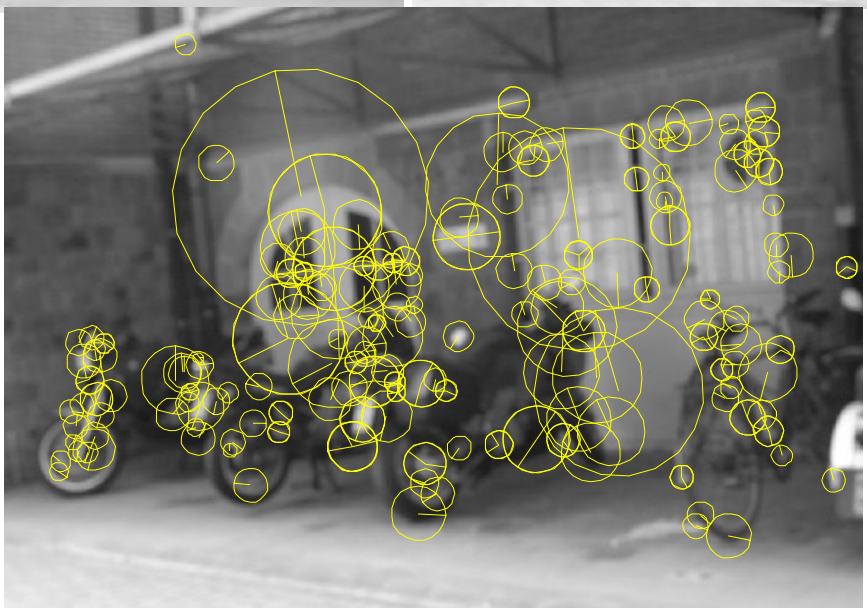
Hessian Affine

Strongest 200 points of det Hessian, Laplace and DoG

det Hessian



Laplacian



DoG

The Maximally Stable Extremal Regions

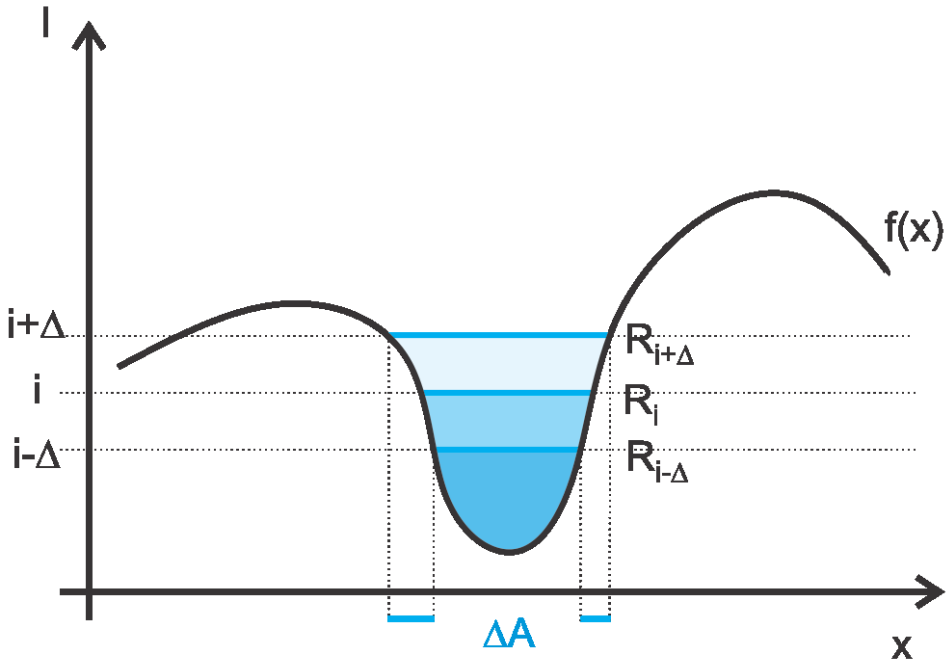
CENTER FOR MACHINE PERCEPTION

1. Region - a contiguous subset of image – connected component
2. Extremal Region R_i – all pixels are of equal or lower/higher intensity than i
3. Maximally Stable Extremal Region – a region R_i for which the relative change of area

$$q(i) = |R_{i+\Delta} \setminus R_{i-\Delta}| / |R_i|$$

is local minimum for a range of intensities.

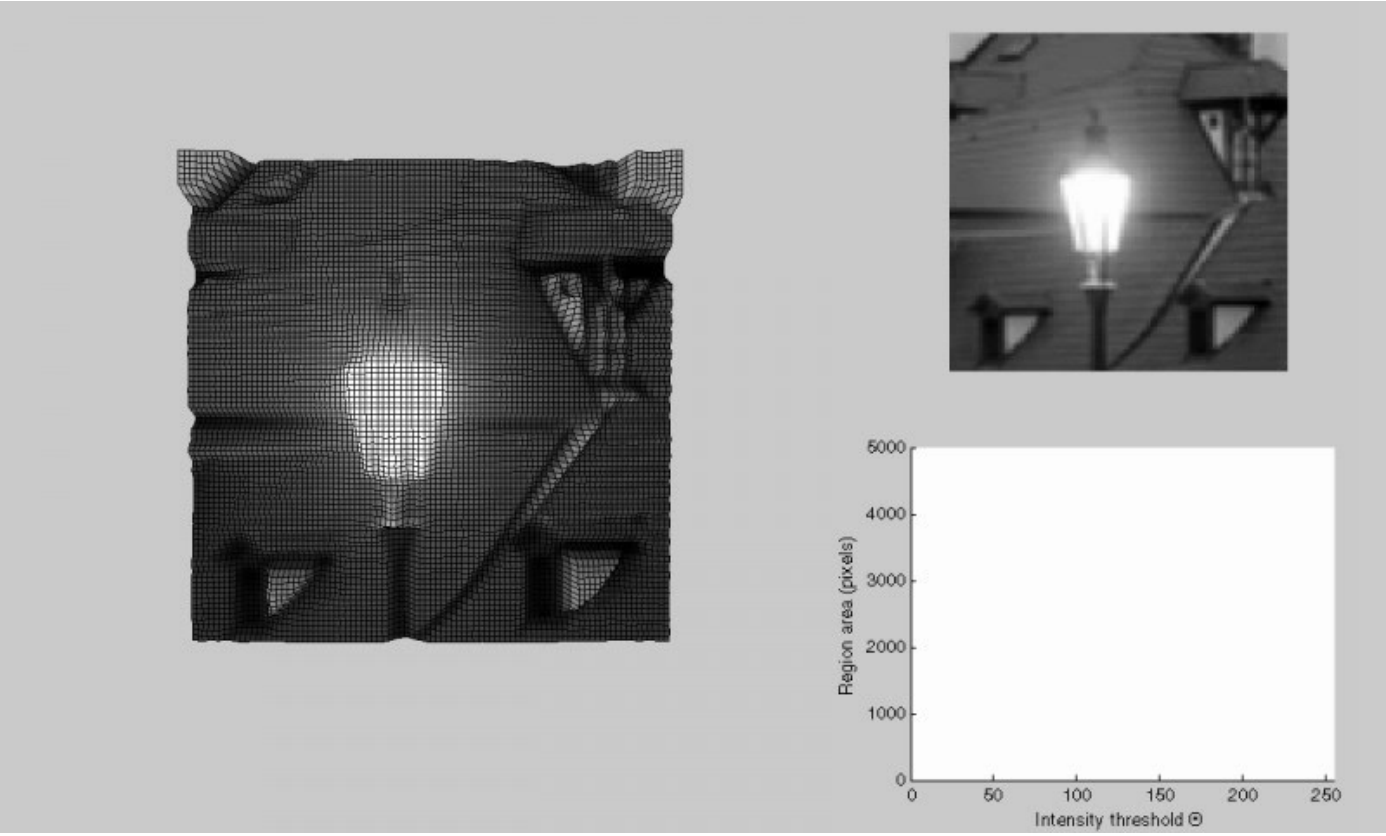
citations
1300 (2010)
1620 (2012)
3000 (2015)



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



et al. "Distinguished Regions for Wide-baseline Stereo". Research Report of CMP, 2001

The Maximally Stable Extremal Regions

CENTER FOR MACHINE PERCEPTION

video

MSER Stability

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

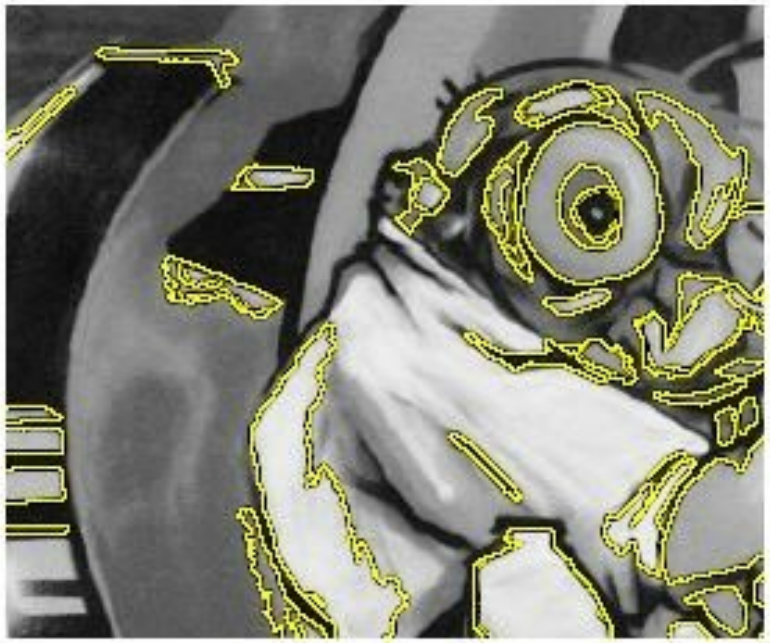
MSER Properties

- Covariant with continuous deformations of images
- Invariant to affine transformation of pixel intensities
- Enumerated in O(n log log n), close to real-time computation
- O(n) for domains with a discrete number of levels smaller than n (Stewenius and Nister)
- Generalisations possible: sort by gradient, hue, compactness, character-ness,

MSER regions (in green). The regions 'follow' the object (video1, video2).

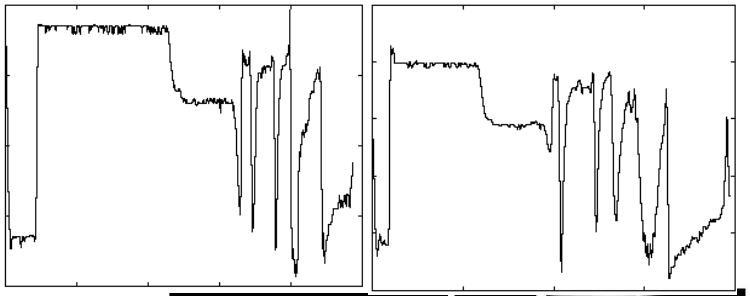
Matas, Chum, Urban, Pajdla: "Robust wide baseline stereo from maximally stable extremal regions". BMVC2002

Maximally Stable Extremal Regions



1D Affine invariance [Tell & Carlsson 00], [Matas & Burianek 00]

- a 2D affine transform maps a line to a line
- on each line, it induces a 1D affine transform

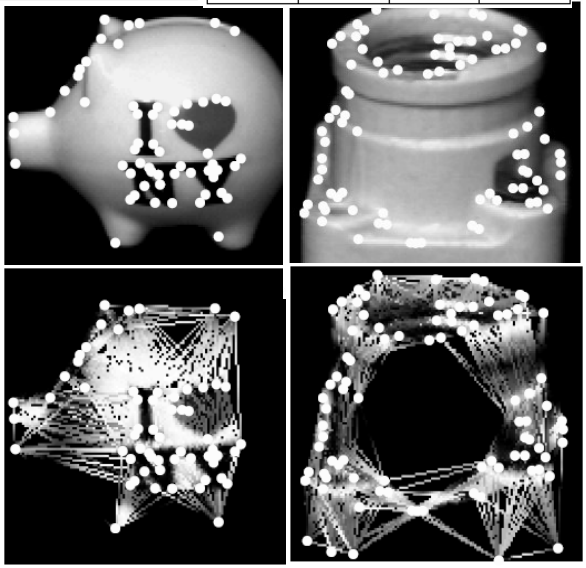


Idea:

- take pairs of points, match intensity profiles along the line segments

Problems:

- which pairs of points to choose?
- the pre-images of the line segments must be planar
- line segment pre-image is often on discontinuities

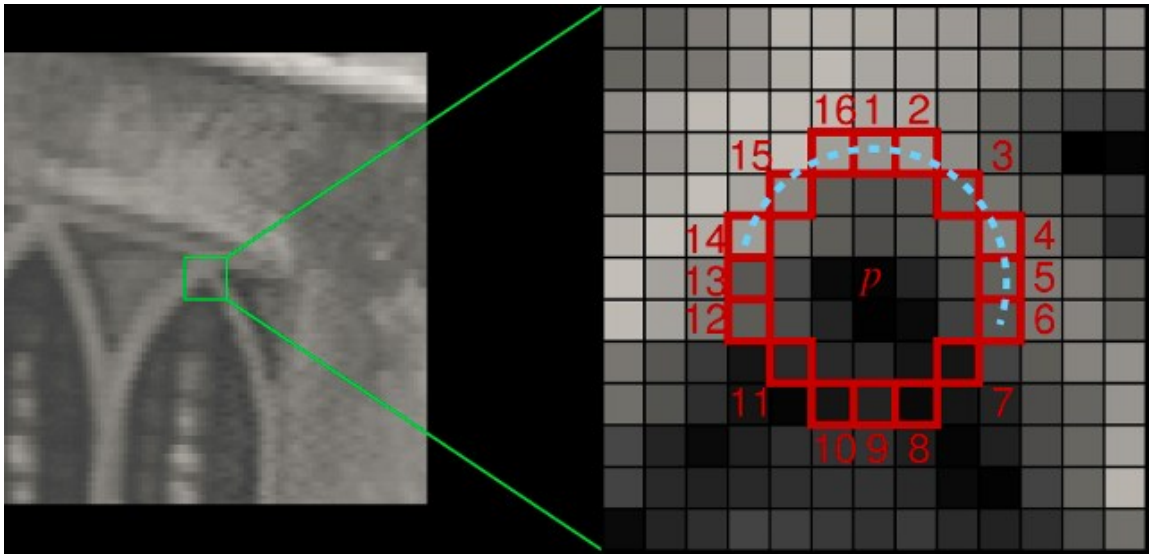


Tell, Carlsson: "Wide Baseline Point Matching Using Affine Invariants Computed from Intensity Profiles",ECCV00
Matas, Burianek: "Object Recognition using the Invariant Pixel-Set Signature", BMVC2000
Hundelshausen, Sukthankar, D-Nets: Beyond Patch-Based Image Descriptors, CVPR12

Talking about Speed

Fast-9 and Fast-ER (E. Rosten)

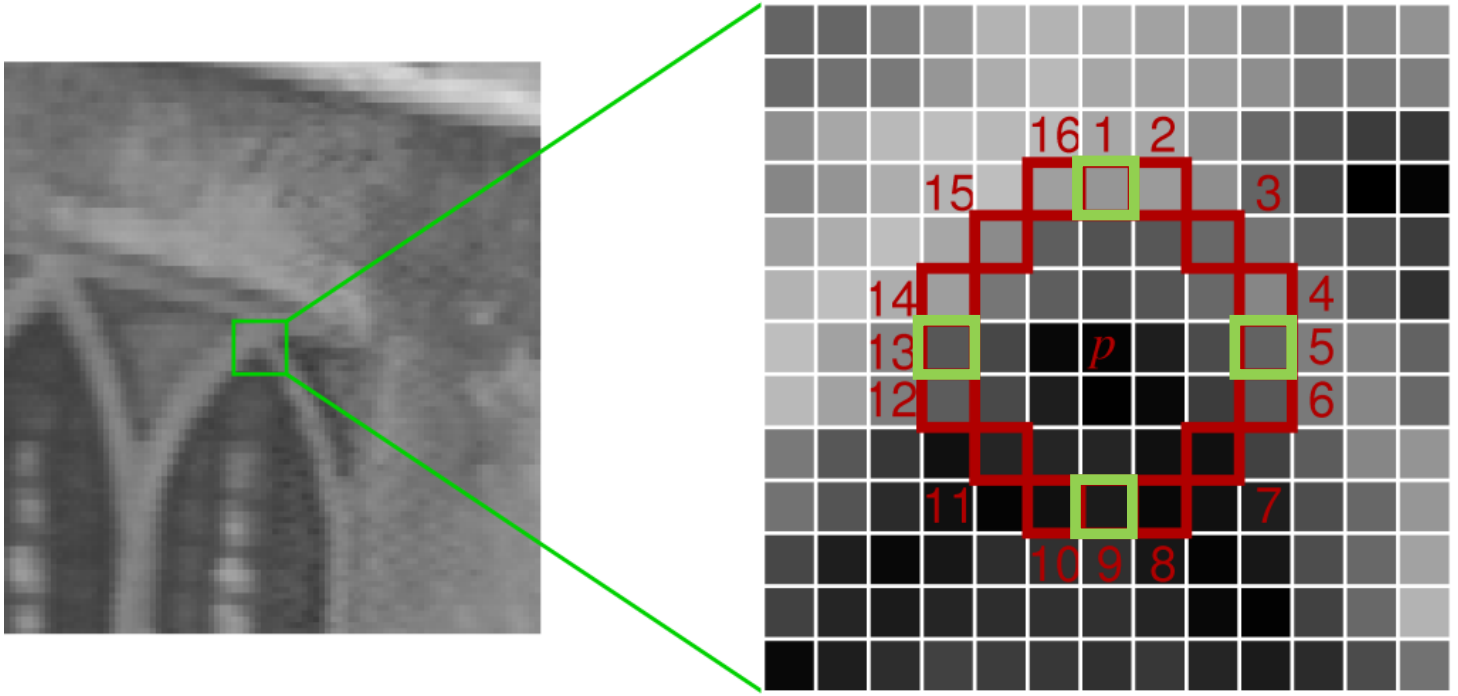
1. in some situations (controlled lighting, tracking), invariance/robustness is less important than speed
2. simple detector based on intensity comparisons could be very fast, and yet "repeatable enough"



citations
730 (2012)
≈ 2000 (2015)

3. detection: 12 contiguous pixels are darker/brighter than the central pixel by at least t.
4. http://www.edwardrosten.com/work/fast.html

FAST Feature Detector



- Considers a circle of 16 pixels around the corner candidate p
- $\geq$ 12 contiguous pixels brighter/darker than $I_p \pm t, t...$ threshold
- Rapid rejection by testing 1,9,5 then 13
  - Only if at least 3 of those are brighter/darker than $I_p \pm t$, the full segment test is applied

Slide credit: E. Rosten

FAST: Weaknesses

1. Corners are clustered together:
   1. Use non-maximal suppression:

$$V = \max\left(\sum_{q\in S_b} |I_q - I_p| - t, \sum_{q\in S_d} |I_p - I_q| - t\right)$$

   where

$$S_b = \{q|I_q \geq I_p + t\}, S_d = \{q|I_q \leq I_p - t\}$$

2. The high speed test does not generalize well for $n < 12$
3. Multiple features are detected adjacent to one another

Fixes:

1. ORB, ..., ...

SURF: Speeded Up Robust Features

CENTER FOR MACHINE PERCEPTION

Idea:

1. Approximate DoG + SIFT calculation with a computationally efficient algorithm.

citations
2300 (2010)
4000 (2012)

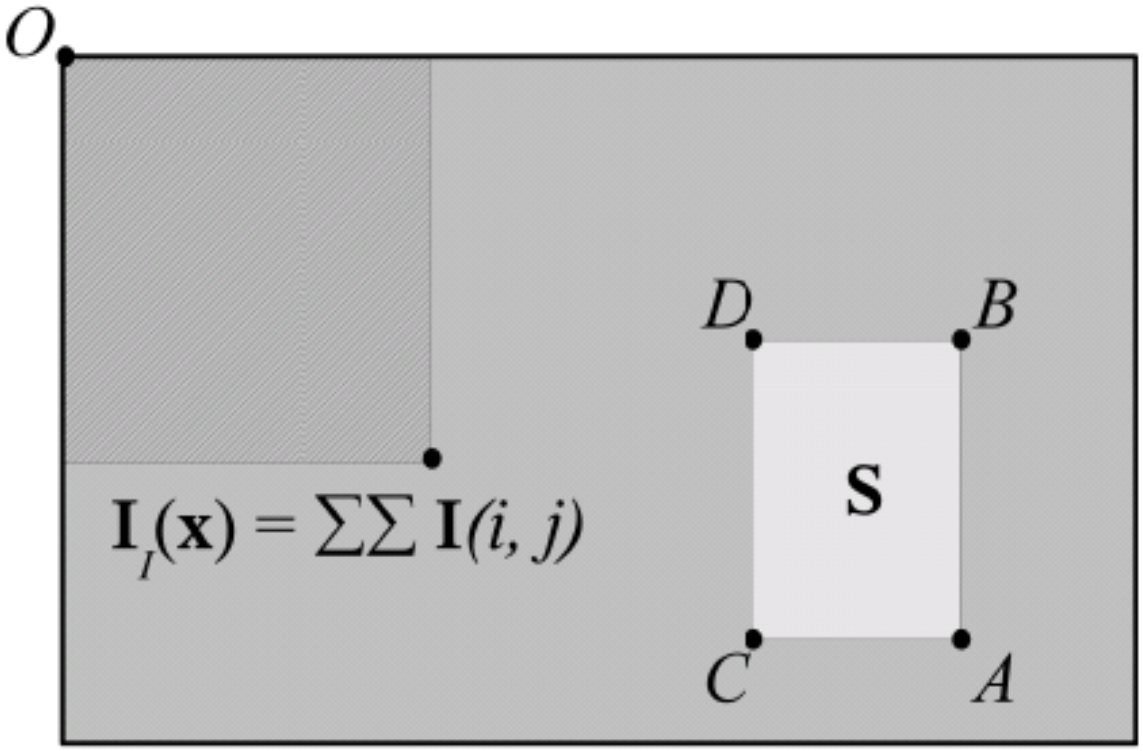
Properties:

- exploit the integral image
- the SURF detector is an approximation to the Hessian
- reuse the calculations needed for detection in descriptor computation
- maintain robustness to rotation, scale illumination change
- approximately 2x faster than DoG

10x faster Hessian-Laplace detector

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool , SURF: Speeded Up Robust Features, ECCV 2006.

The Integral image (Sum Table)



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

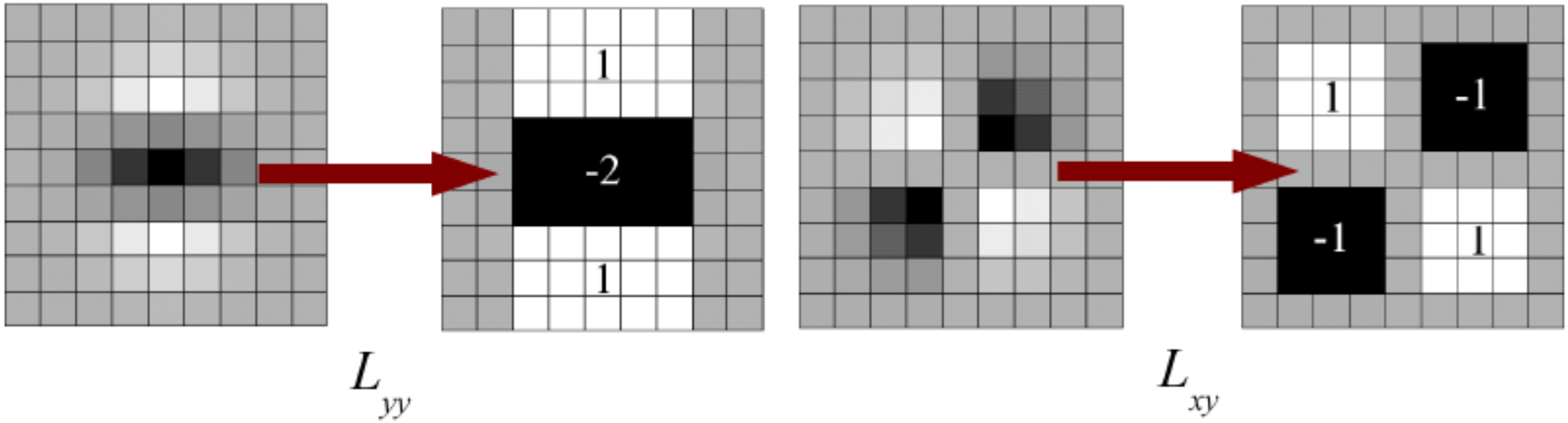
To calculate the sum in the DBCA rectangle, only 3 additions are needed

SURF Detection

- Hessian-based interest point localization:
- $L_{xx}(x,y,\sigma)$ is the convolution of the *Gaussian* second order derivative with the image

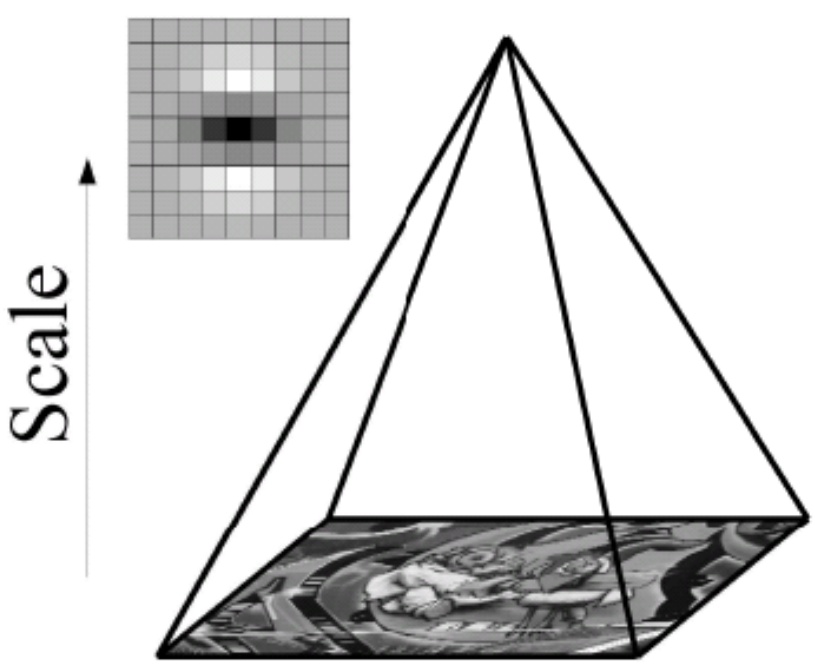
$$H = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix}$$

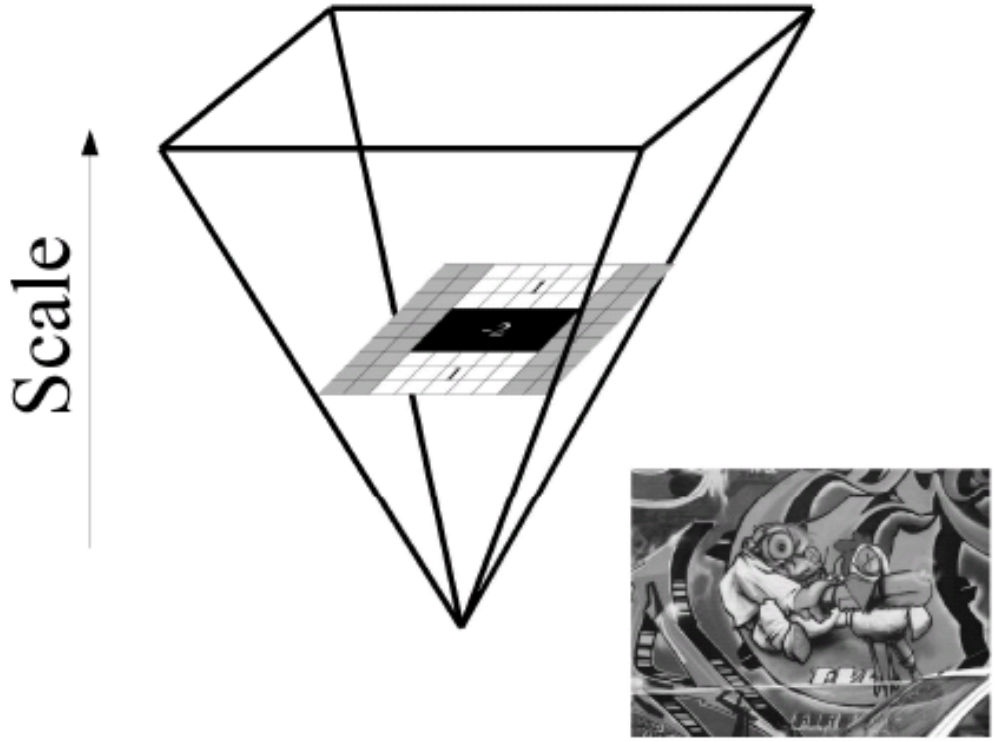
1. Approximate second order derivatives with box filters filters



SURF Detection

1. Scale analysis easily handled with the integral image





9 x 9, 15 x 15, 21 x 21, 27 x 27 → 39 x 39, 51 x 51 …

1st octave *2nd octave*

FAST: running times

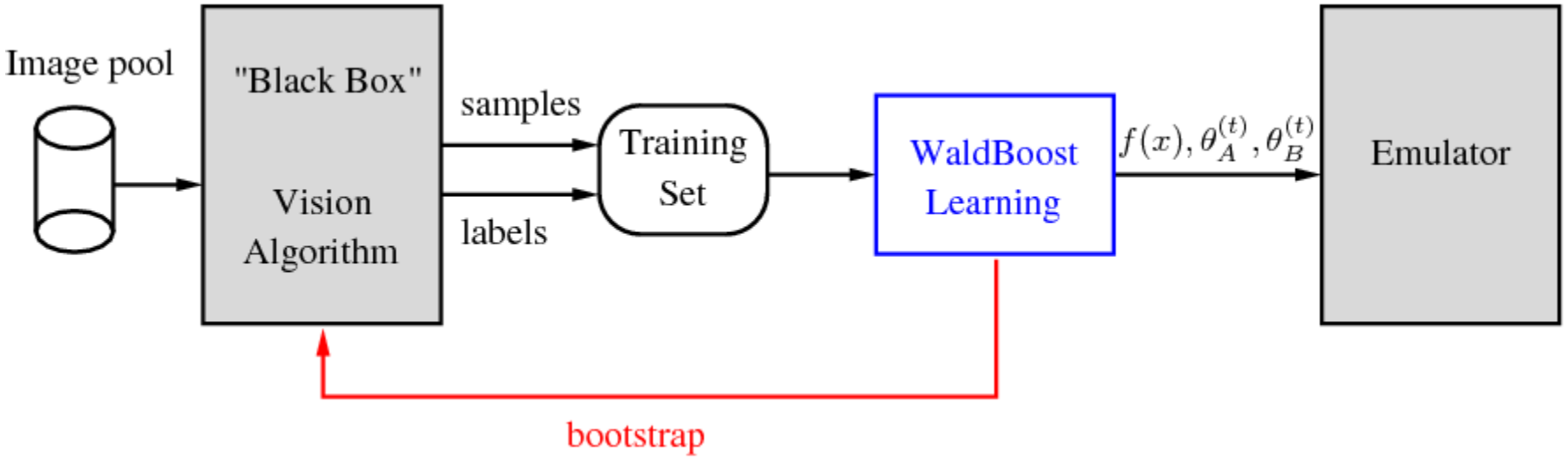
Detector	Opteron 2.6GHz		Pentium III 850MHz	
	ms	%	ms	%
Fast $n = 9$ (non-max suppression)	1.33	6.65	5.29	26.5
Fast $n = 9$ (raw)	1.08	5.40	4.34	21.7
Fast $n = 12$ (non-max suppression)	1.34	6.70	4.60	23.0
Fast $n = 12$ (raw)	1.17	5.85	4.31	21.5
Original FAST $n = 12$ (non-max suppression)	1.59	7.95	9.60	48.0
Original FAST $n = 12$ (raw)	1.49	7.45	9.25	48.5
Harris	24.0	120	166	830
DoG	60.1	301	345	1280
SUSAN	7.58	37.9	27.5	137.5

Table 1. Timing results for a selection of feature detectors run on fields (768×288) of a PAL video sequence in milliseconds, and as a percentage of the processing budget per frame. Note that since PAL and NTSC, DV and 30Hz VGA (common for web-cams) have approximately the same pixel rate, the percentages are widely applicable. Approximately 500 features per field are detected.

multiply 5x to get processing time for 1MPx image @ 2.6 GHz
But the winner is GPUSIFT @ about 5ms per HD image.

Slide credit: E. Rosten

Black-box Generated Training Set



- The Emulator approximates the behavior of the black-box algorithm
- The black-box algorithm can potentially provide almost unlimited number of training samples
  - Efficiency of training is important
  - Suitable for incremental or online methods

Speeding up Detectors using Machine Learning

CENTER FOR MACHINE PERCEPTION

- Use the WaldBoost Sequential Classifier as a fast emulator of a detector
- WaldBoost Combines AdaBoost training (which selects measurements) with Wald's sequential decision making theory (for time quasioptimal decisions)

- AdaBoost learning converges asymptotically to

$$\lim_{T\to\infty} H_T(x) = \tilde{H}(x) = -\frac{1}{2}\log R(x) + \frac{1}{2}\log\frac{P(+1)}{P(-1)}.$$

- Sequential WaldBoost classifier

$$H_t(x) = \begin{cases} +1, & f_t(x) \geq \theta_B^{(t)} \\ -1, & f_t(x) \leq \theta_A^{(t)} \\ \sharp, & \theta_A^{(t)} < f_t(x) < \theta_B^{(t)} \end{cases}$$

$$f_t(x) = \sum_{q=1}^{t} h^{(q)}(x)$$

- Set of weak classifiers (features) a
  are found during training

$$\theta_A^{(t)}, \theta_B^{(t)}, t = 1, \ldots, T$$

Process

CENTER FOR MACHINE PERCEPTION

The Idea

- Given a black box algorithm **A** performing a useful binary (decision) task
- Train a sequential classifier **S** to (approximately) emulate output of algorithm **A** while minimizing time-to-decision
- Allow user to control quality of the approximation

Advantages

- Instead of spending man-months on code optimization, choose relevant feature class and train sequential classifier **S**
- A (slow) Matlab code can be speeded up this way!

[1] J. Sochman and J. Matas.
Waldboost – Learning For Time Constrained Sequential Detection. CVPR 2005

[2] J. Sochman and J. Matas.
Learning fast emulators of binary decision processes. IJCV 83(1):149 – 163, 2009.

Descriptors of Local Invariant Features

Descriptors Invariant to Rotation

CENTER FOR MACHINE PERCEPTION

1. 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 their magnitude

Rotation invariant descriptor consists of magnitudes of moments: $|m_{kl}|$

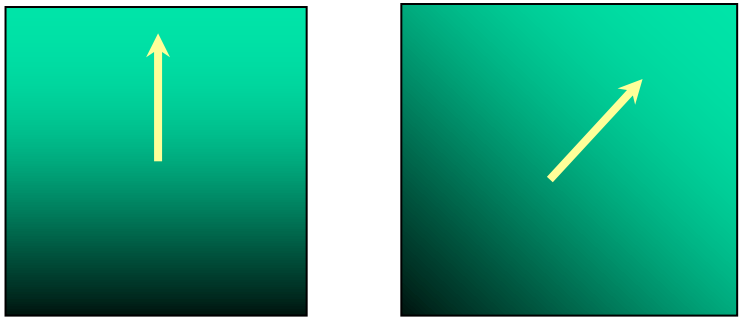
Matching is done by comparing vectors $[|m_{kl}|]_{k,l}$

J.Matas et.al. "Rotational Invariants for Wide-baseline Stereo". Res. Report of CMP, 2003

Descriptors Invariant to Rotation

- 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

Descriptors Invariant to Scale

CENTER FOR MACHINE PERCEPTION

1. 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: sI_x

Affine Invariant Descriptors

CENTER FOR MACHINE PERCEPTION

1. Affine invariant color moments

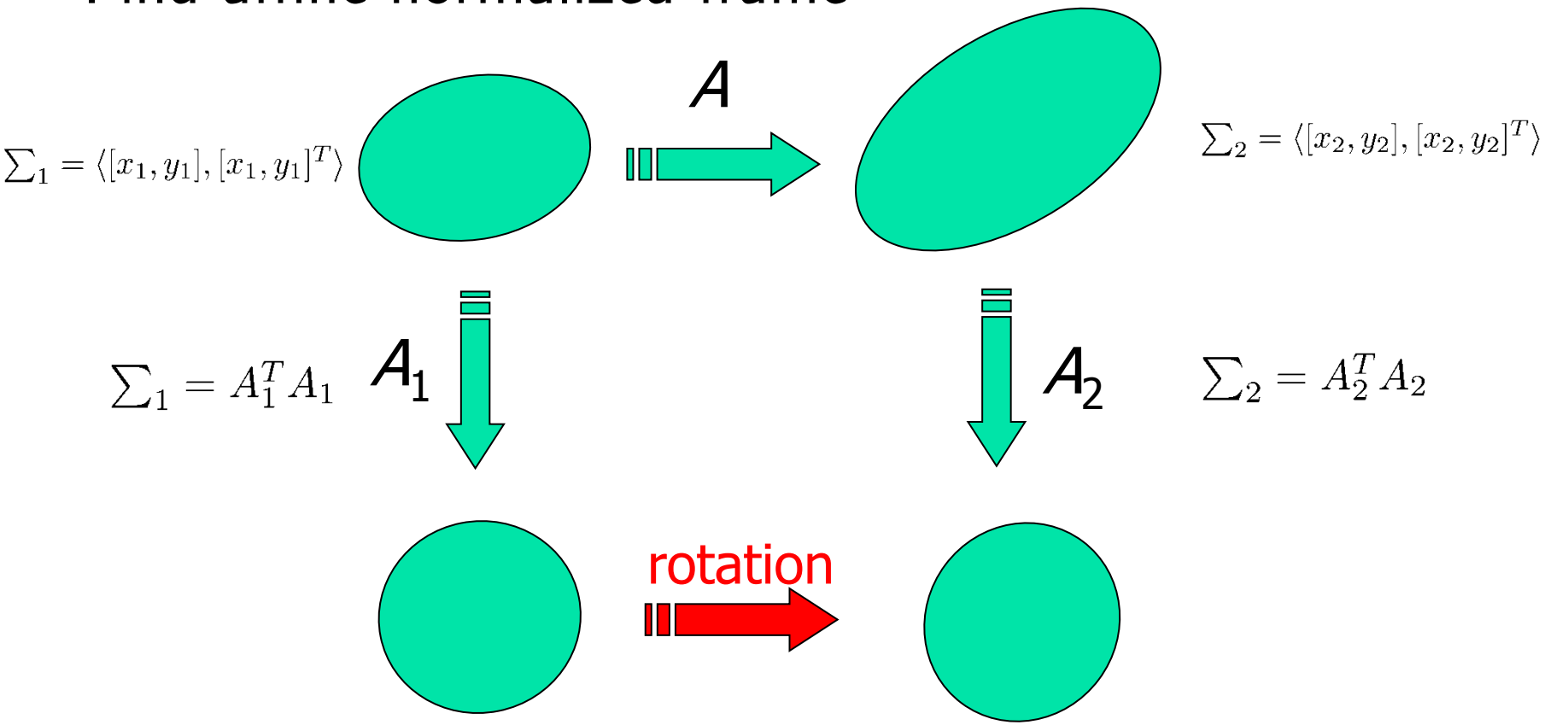
$$m_{pq}^{abc} = \int_{region} x^p y^q R^a(x,y) G^b(x,y) B^c(x,y) dxdy$$

- Different combinations of these moments are fully affine invariant
- Also invariant to affine transformation of intensity $I \rightarrow a\, I + b$

F.Mindru et.al. "Recognizing Color Patterns Irrespective of Viewpoint and Illumination". CVPR99

Affine Invariant Descriptors

- Find affine normalized frame

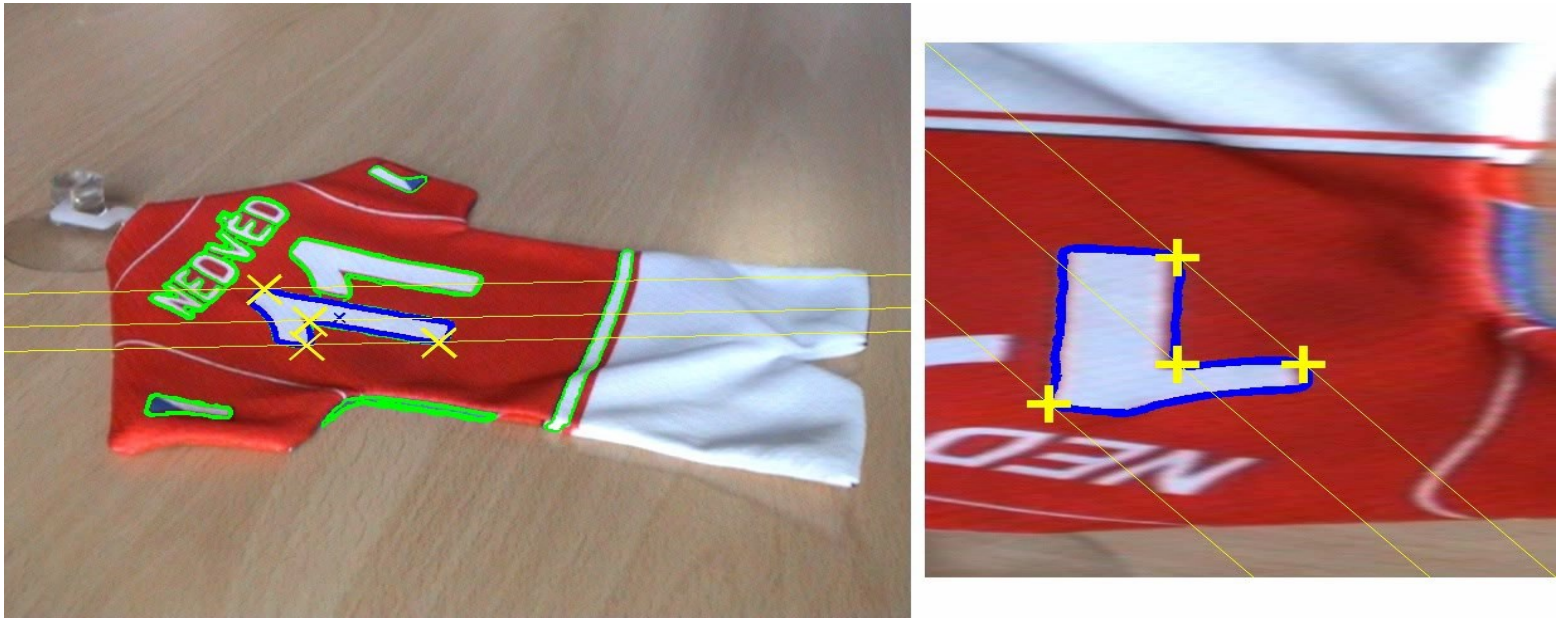


- Compute rotational invariant descriptor in this normalized frame

J.Matas et.al. "Rotational Invariants for Wide-baseline Stereo". Res. Report of CMP, 2003

Local Affine Frames

Step 1: Find MSERs (maximaly stable extremal regions)
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

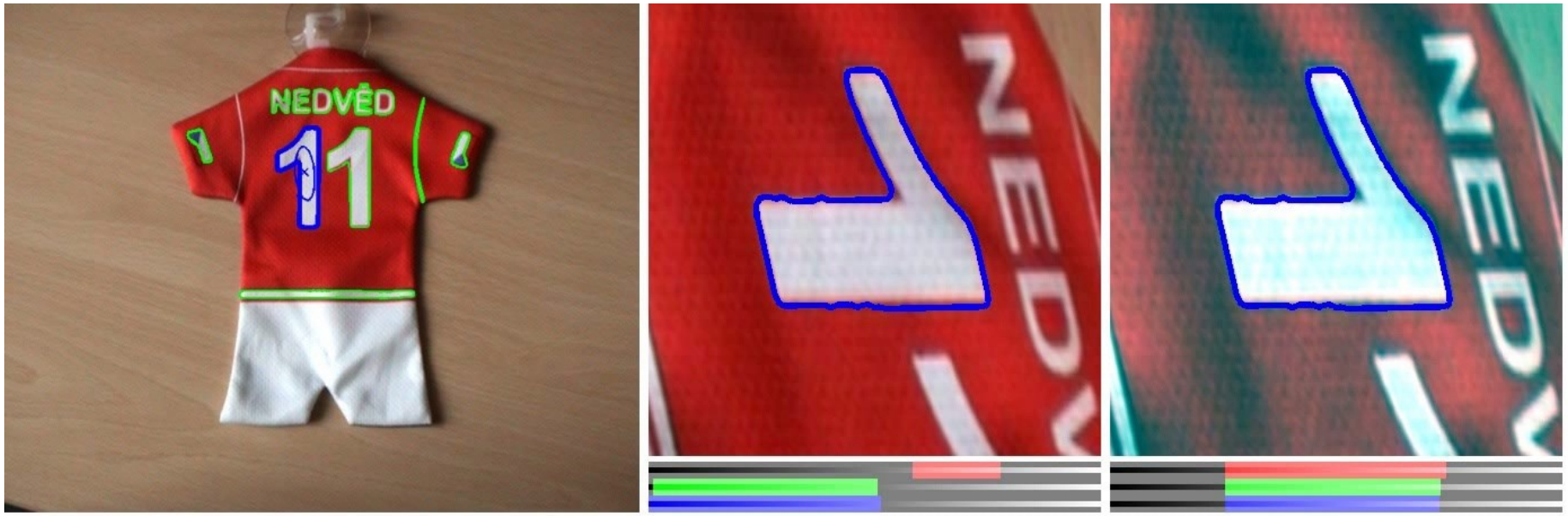
Obdržálek and Matas: "Object recognition using local affine frames on distinguished regions". BMVC02
Obdržálek and Matas: "Sub-linear Indexing for Large Scale Object Recognition", BMVC 2005

MSER-LAF-Tree, Obdrzalek and Matas, 2005

CENTER FOR MACHINE PERCEPTION

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

[video-1, video-2]



Affine-Covariant Constructions: Taxonomy

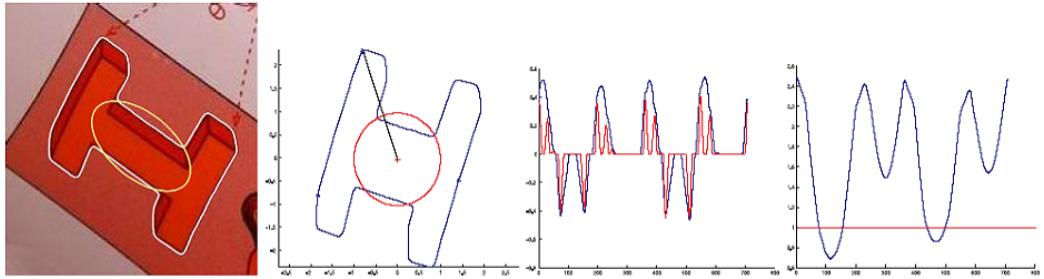
1. Derived from ***region outer boundary***
   1. Region area (1 constraint)
   2. Center of gravity (2 constraints)
   3. Matrix of second moments (symmetric 2x2 matrix: 3 constraints)

$$|\Omega| = \int_{\Omega} \mathbf{1} d\Omega$$

$$\mu = \frac{1}{|\Omega|} \int_{\Omega} \mathbf{x} d\Omega$$

$$\Sigma = \frac{1}{|\Omega|} \int_{\Omega} (\mathbf{x} - \mu)(\mathbf{x} - \mu)^T d\Omega$$

   1. Points of extremal distance to the center of gravity
   2. Points of extremal curvature (2 constraints)

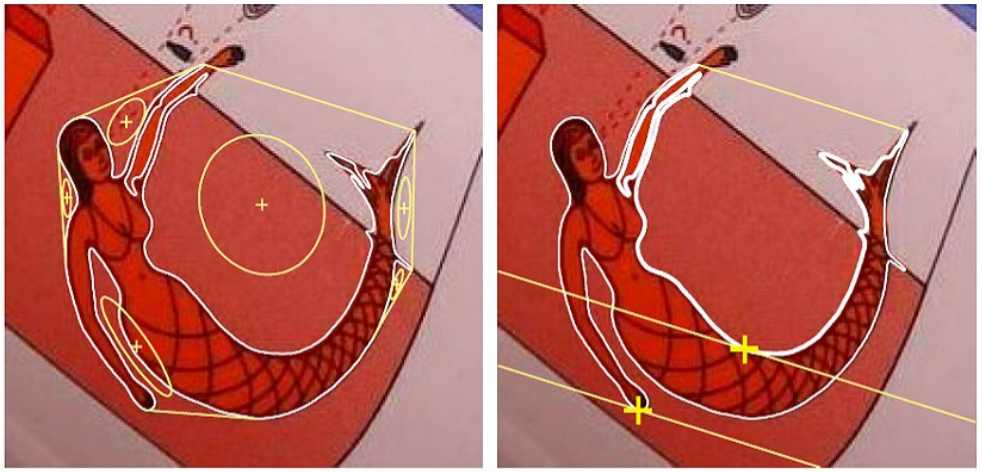


(a) (b) (c) (d)

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.

Affine-Covariant Constructions: Taxonomy

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



(a) (b)

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.

Affine-Covariant Constructions: Taxonomy

CENTER FOR MACHINE PERCEPTION

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

   4. many other

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

Affine-Covariant Constructions: Taxonomy

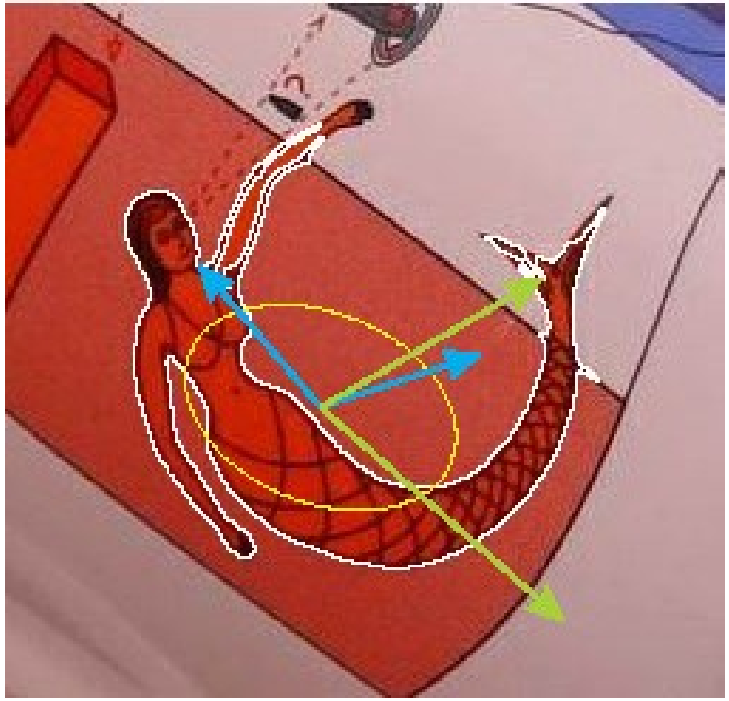
1. Derived from ***topology*** of regions
   1. mutual configuration of regions (combined constraints)
      1. nested regions
      2. incident regions
      3. neighbouring regions

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

3. Convex hull of a region without loosing affine invariance

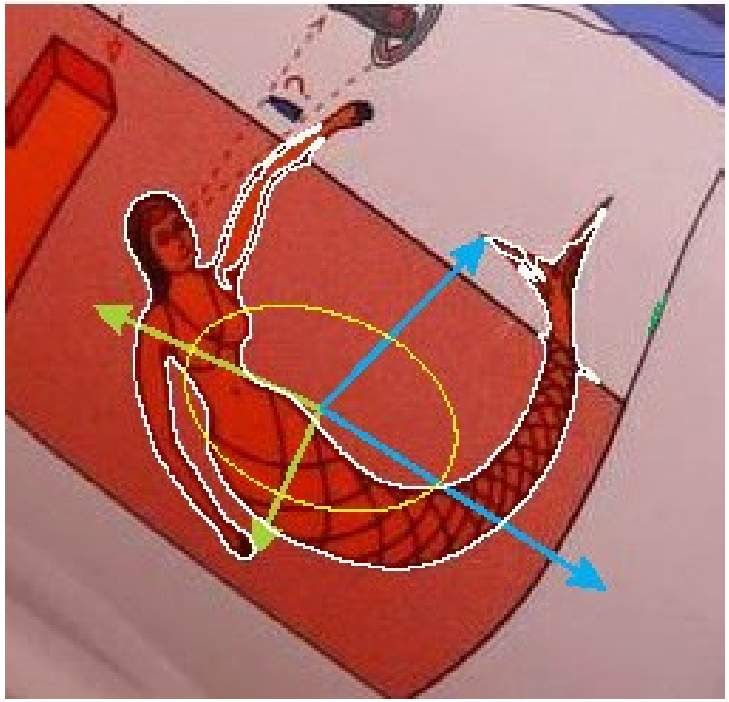
Constructions of Local Affine Frames

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



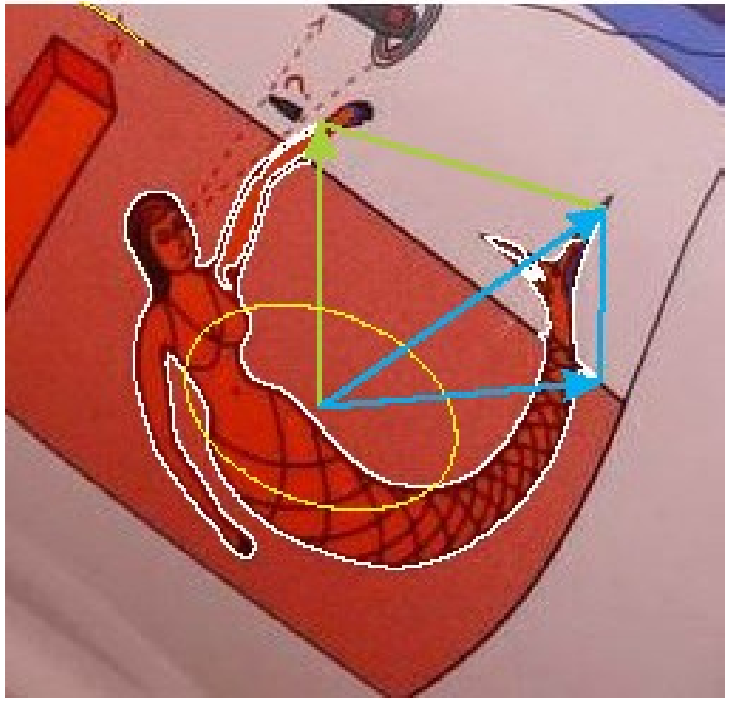
Constructions of Local Affine Frames

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



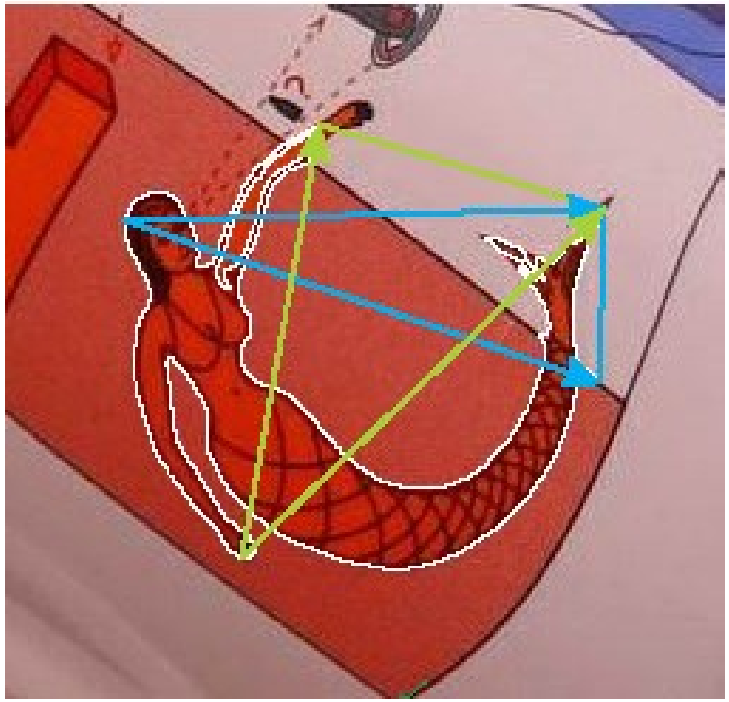
Constructions of Local Affine Frames

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



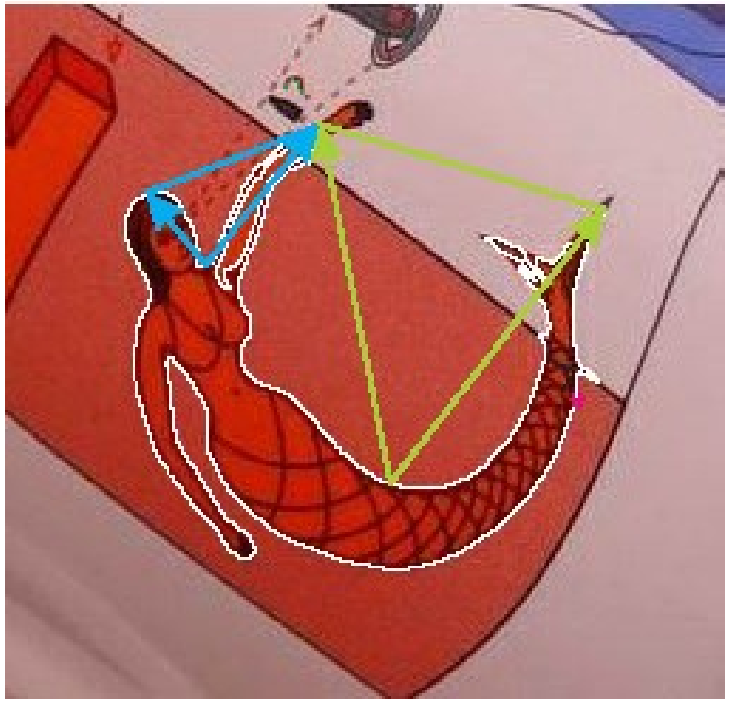
Constructions of Local Affine Frames

1. Combinations of constructions used to form the local affine frames
   1. tangent points + farthest point of the region



Constructions of Local Affine Frames

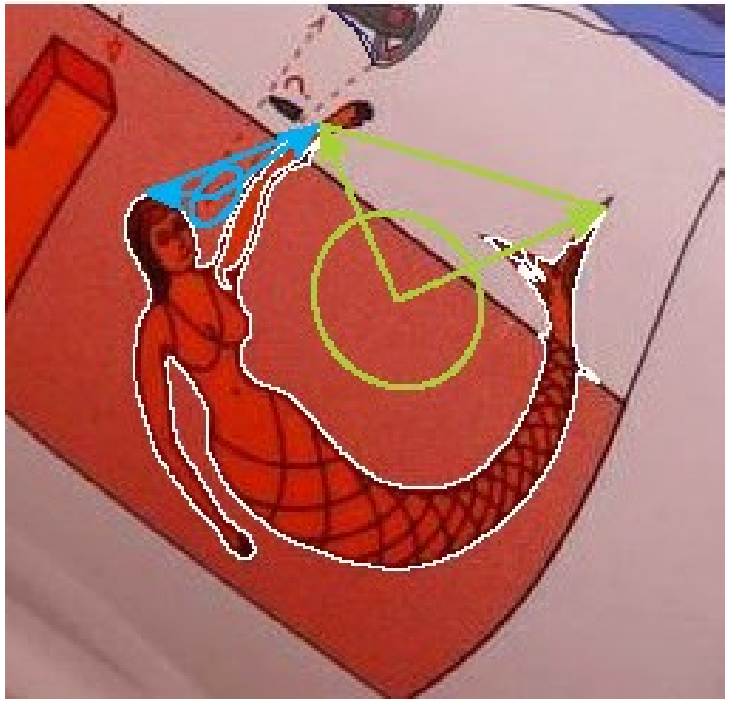
1. Combinations of constructions used to form the local affine frames
   1. tangent points + farthest point of the concavity



Constructions of Local Affine Frames

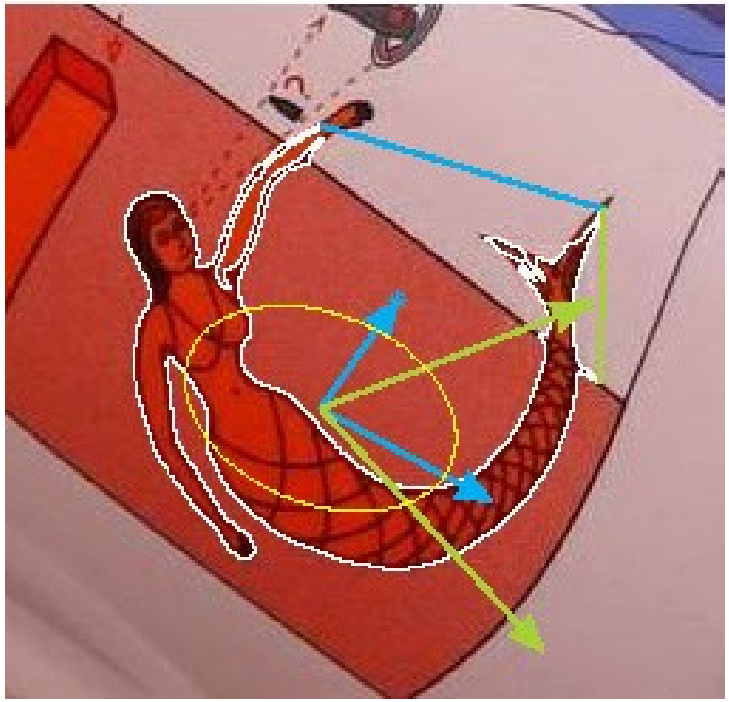
CENTER FOR MACHINE PERCEPTION

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



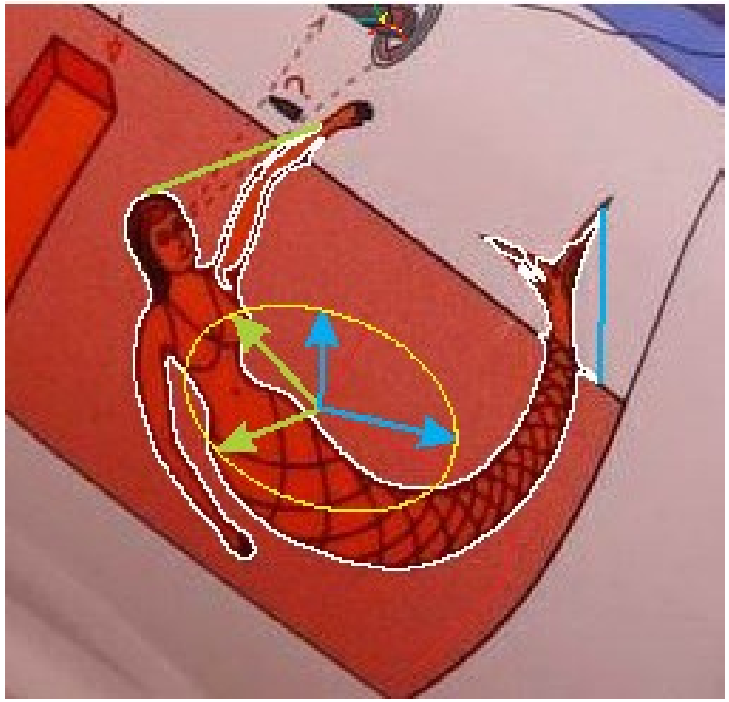
Constructions of Local Affine Frames

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



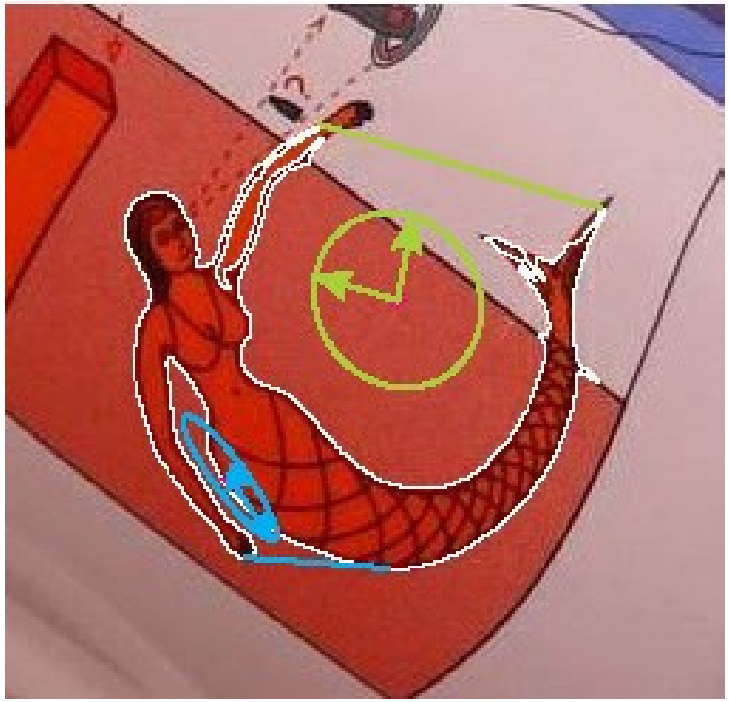
Constructions of Local Affine Frames

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



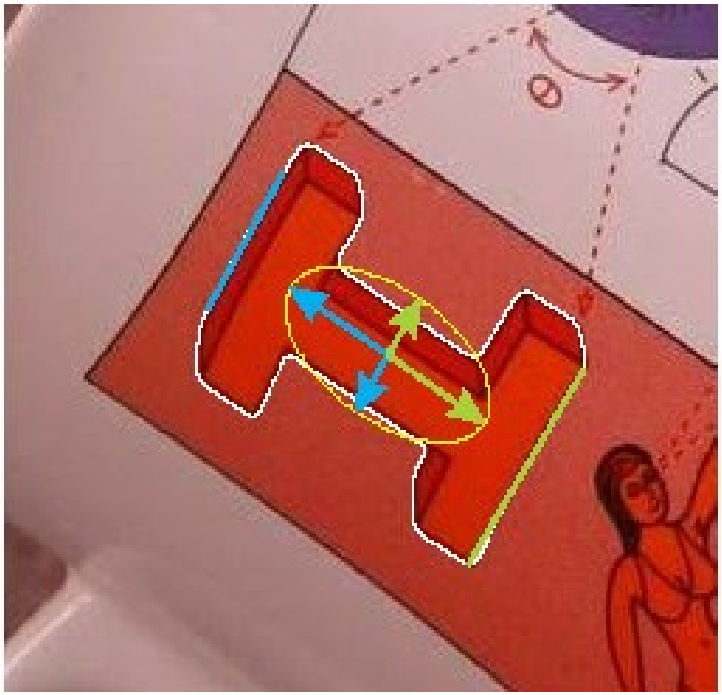
Constructions of Local Affine Frames

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



Constructions of Local Affine Frames

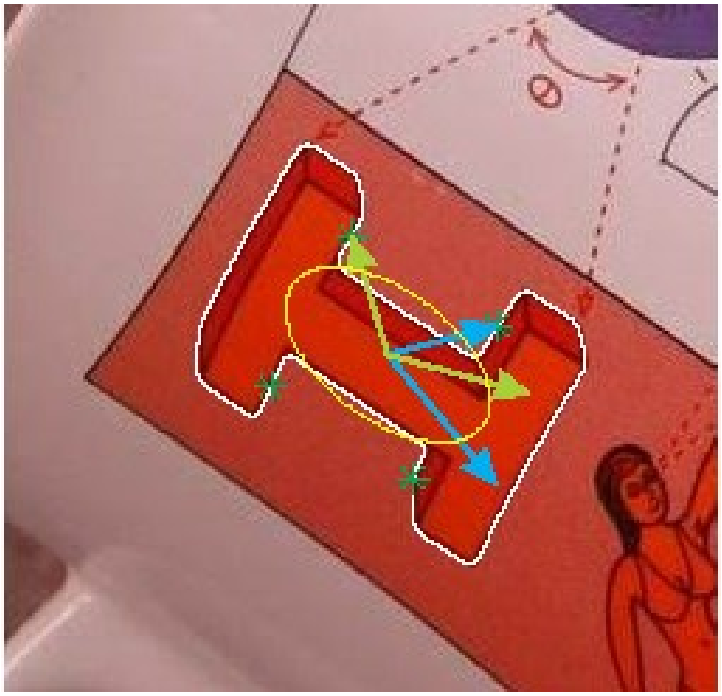
1. Combinations of constructions used to form the local affine frames
   1. center of gravity + covariance matrix + the direction of a linear segment of the contour



Constructions of Local Affine Frames

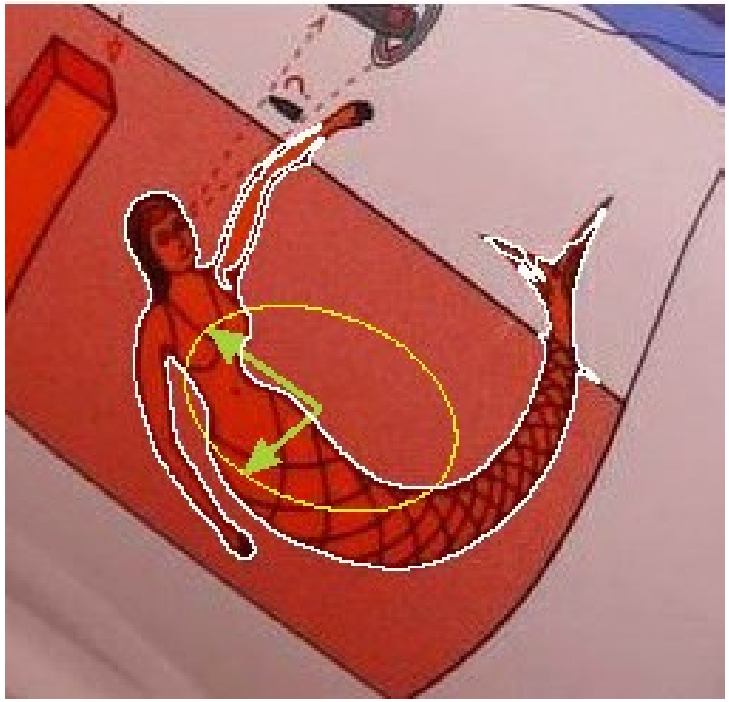
CENTER FOR MACHINE PERCEPTION

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



Constructions of Local Affine Frames

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



Affine-Covariant Constructions: Taxonomy

1. Derived from ***region outer boundary*** (continued)
   1. Points of curvature inflection (2 constraints)
      1. curvature changes from convex to concave or vice-versa
   2. Straight line segments (1 stable constraint for direction, or 4 for the end-points)
   3. Higher than 2nd order moments

      a complex number formed from 3rd order moments

      whose phase angle

      changes covariantly wi[illegible] (1 constraint)

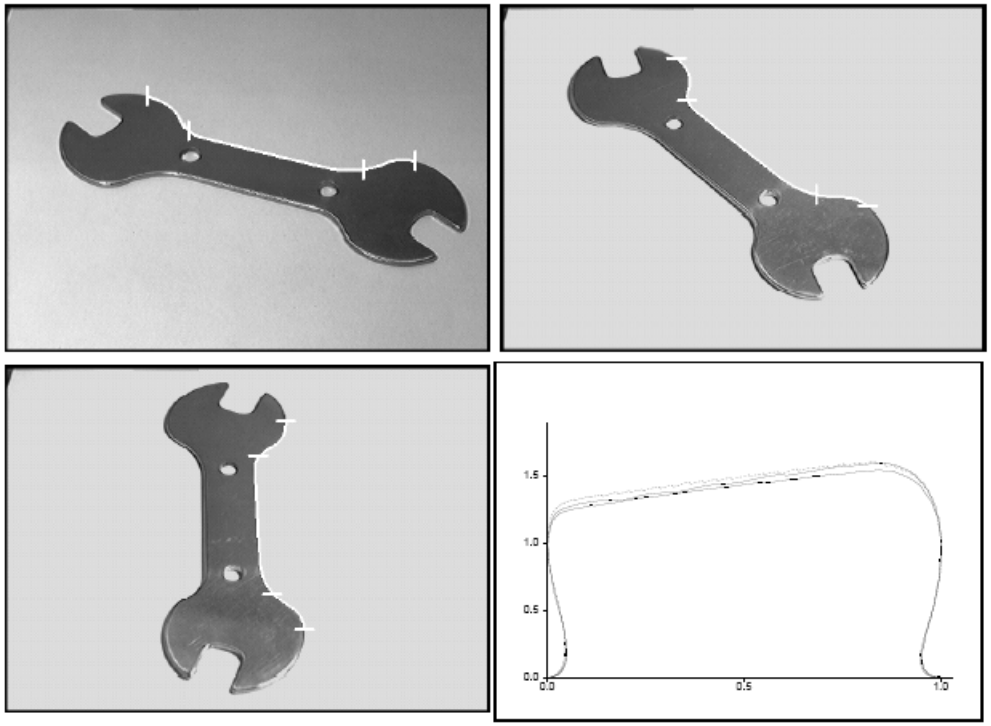
$$c = \mu_{x^3} + \mu_{xy^2} + i(\mu_{x^2y} + \mu_{y^3})$$

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

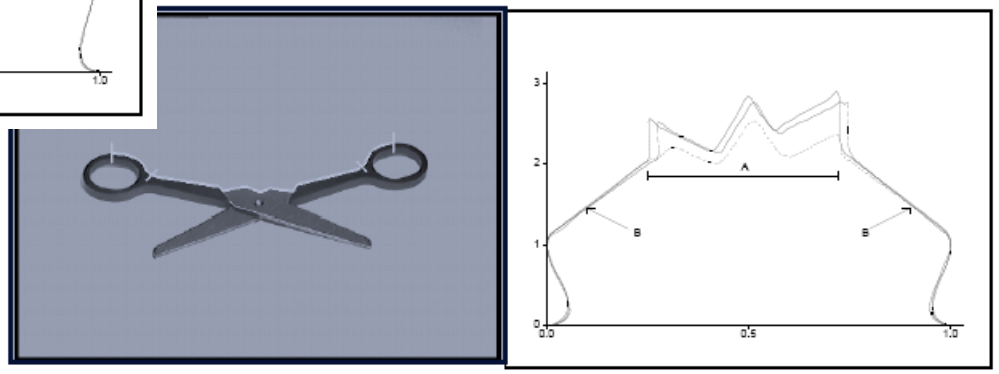
[Hei04] Janne Heikkilä. Pattern matching with affine moment descriptors. *Pattern Recognition*, 37(9):1825–1834, 2004.

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

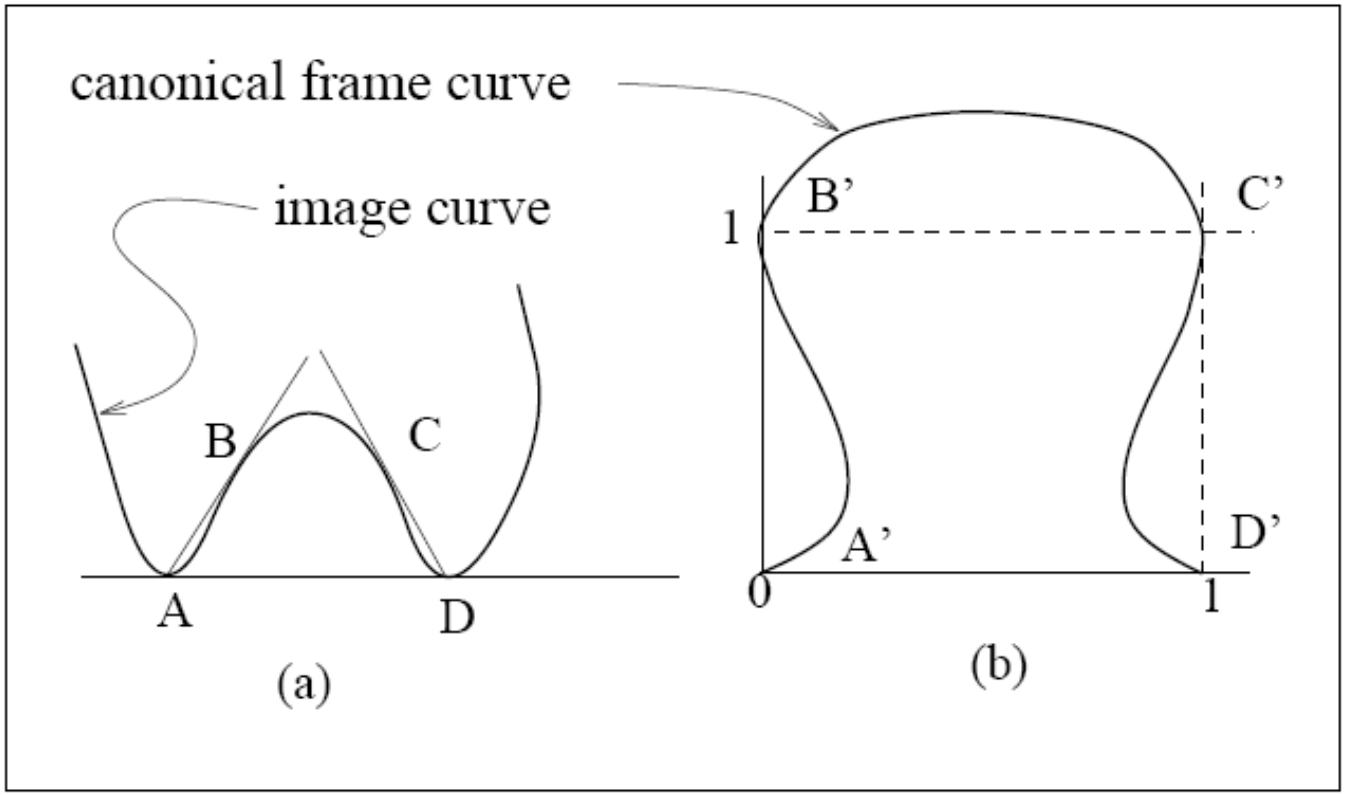


Fig. 1. (a) Construction of the four points necessary to define the canonical frame for a concavity. The first two points $(A\ D)$ are points of bitangency that mark the entrance to the concavity. Two further distinguished points, $(B\ C)$, are obtained from rays cast from the bitangent contact points 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.

Impressing the Reader: Robustness to occlusion, clutter, multiple objects

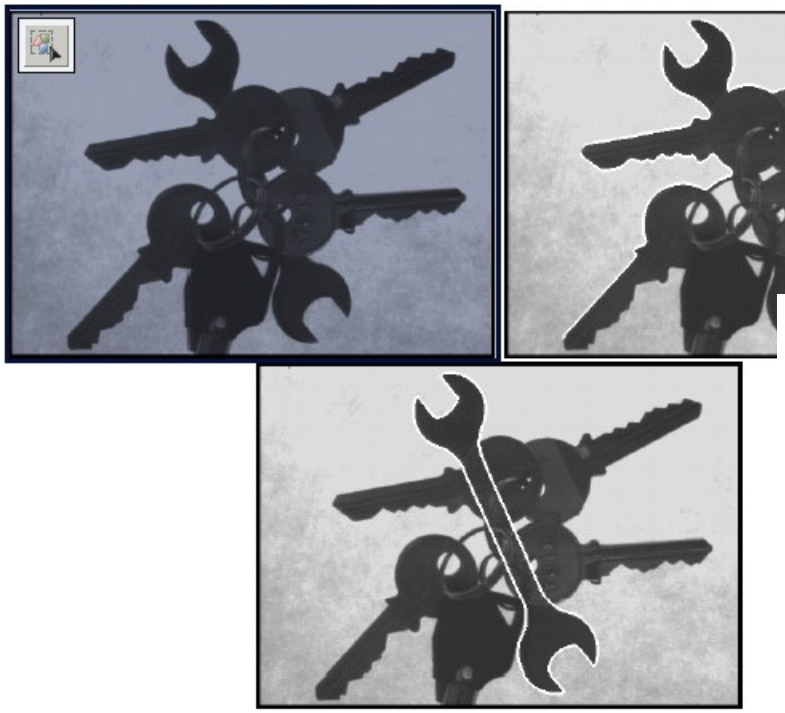


Fig. 6. (a) Spanner almost entirely occluded by keys. The keys are not the li
in this scene. (b) Detected concavities, highlighted in white, which are usec
(c) The spanner which is the only model in the scene contained in the libra
the end slot concavity. The projected outline used for verification is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

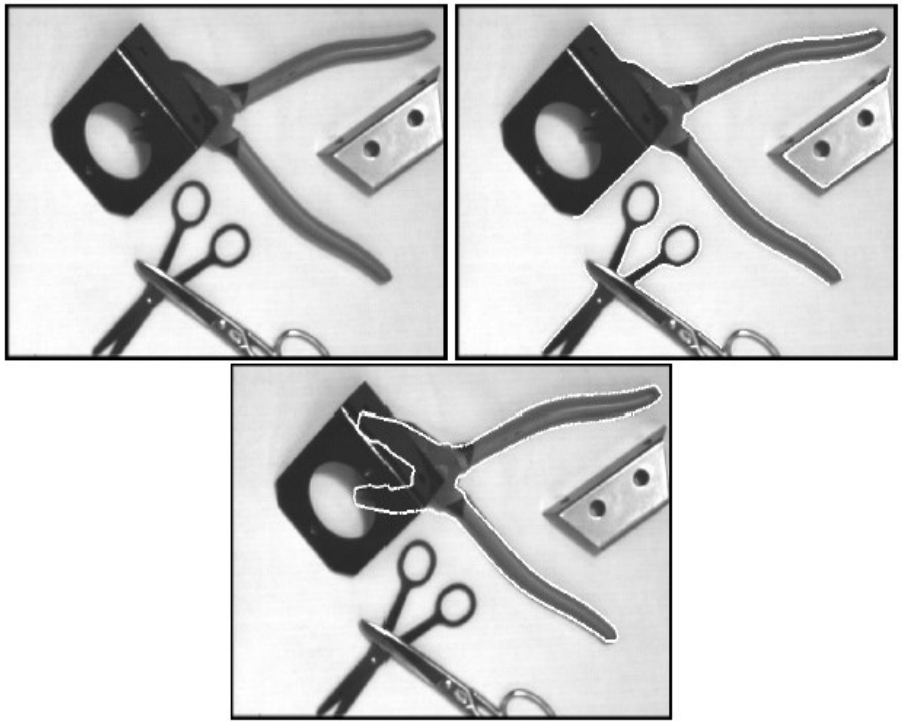


Fig. 7. (a) Image of various planar objects. (b) Concavities, highlighted in white, which are used to compute indexes (c) The pliers which are the only model in the scene contained in the library, is recognised and verified by projecting the edgels from an acquisition image, and checking overlap with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

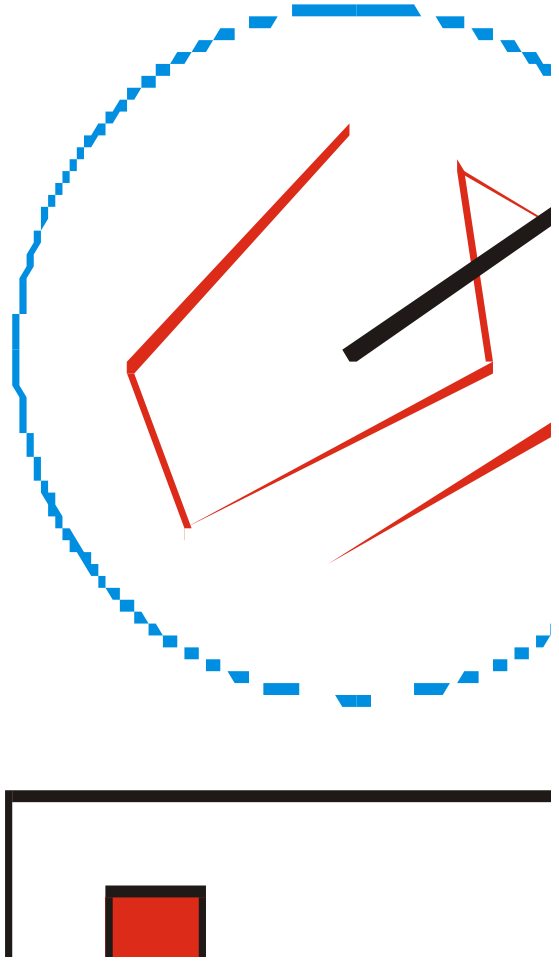
Descriptors

Building a Similarity Frame (s) (my terminology)

Select canonical orientation (s)

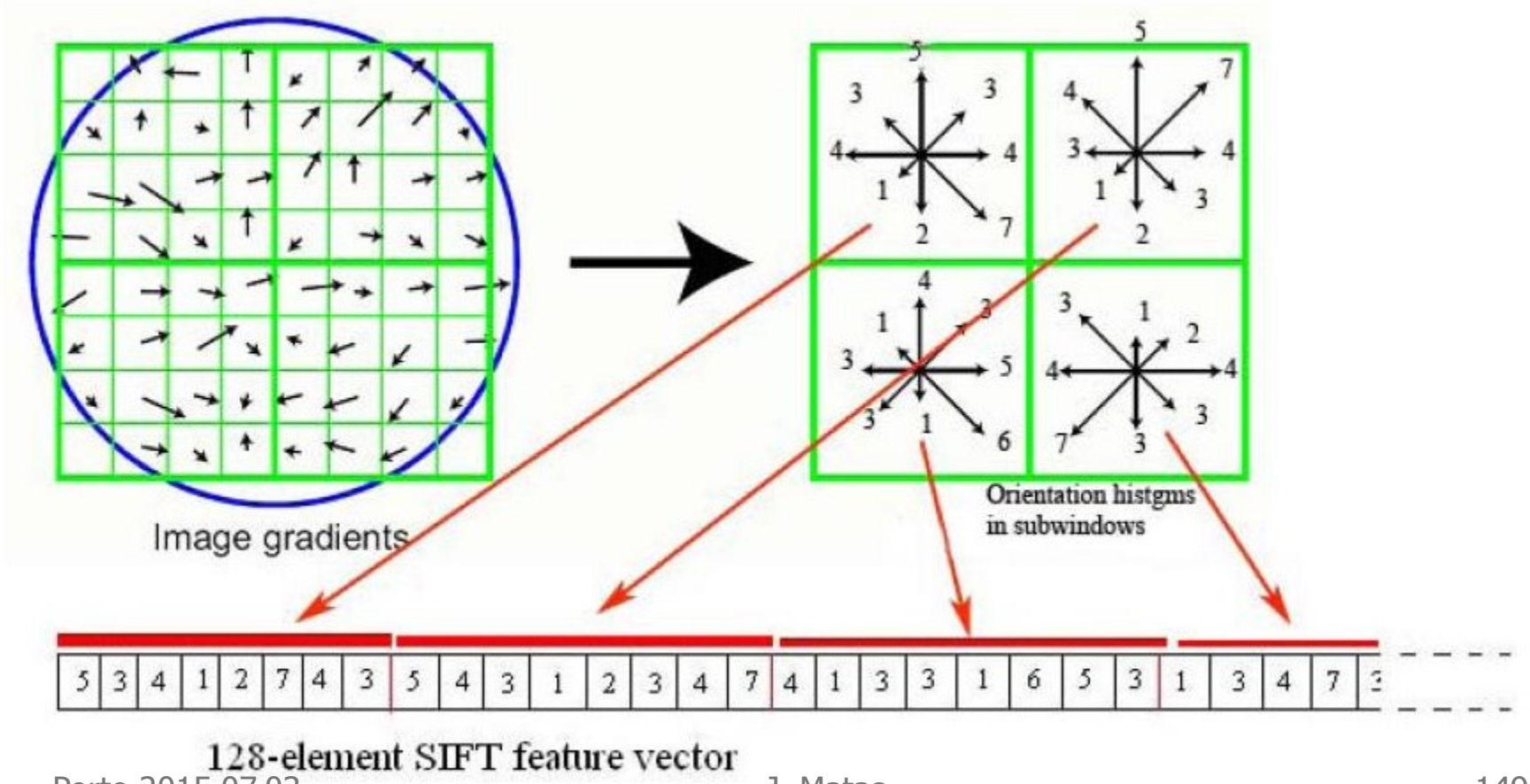
1. Compute a histogram of local gradient directions computed at the selected scale
2. Assign canonical orientation(s) at peak(s) of smoothed histogram
3. (x, y, scale) + orientation defines a local *similarity frame;* equivalent to detecting 2 distinguished points

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



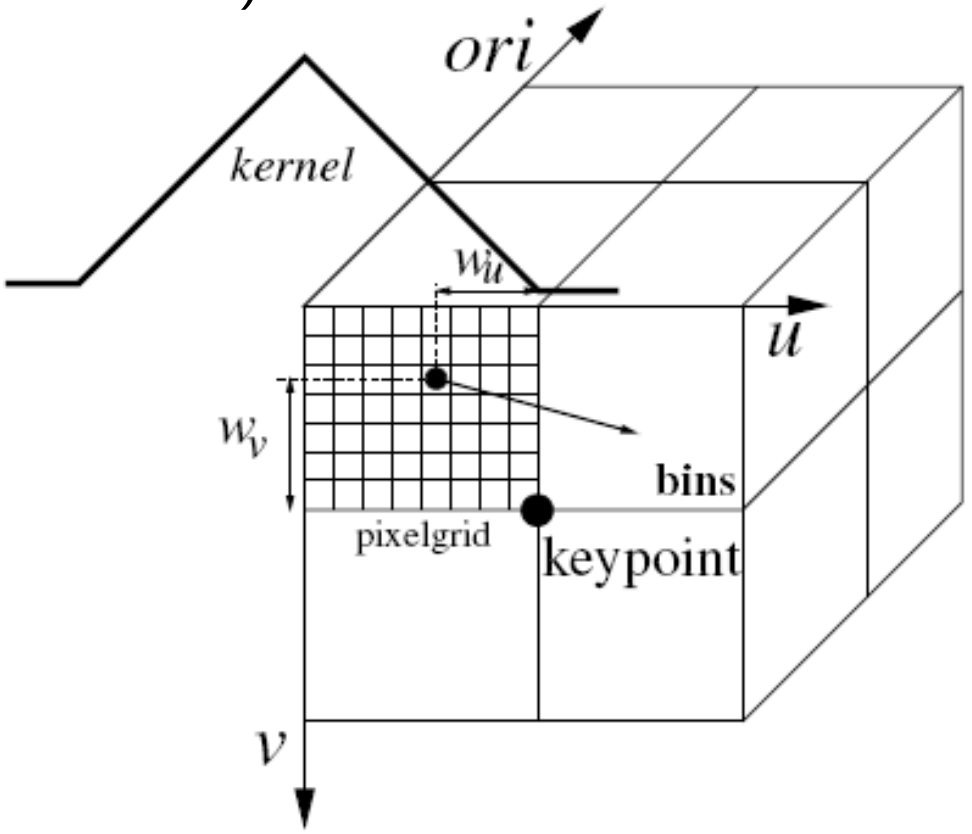
SIFT Descriptor

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



SIFT Descriptor

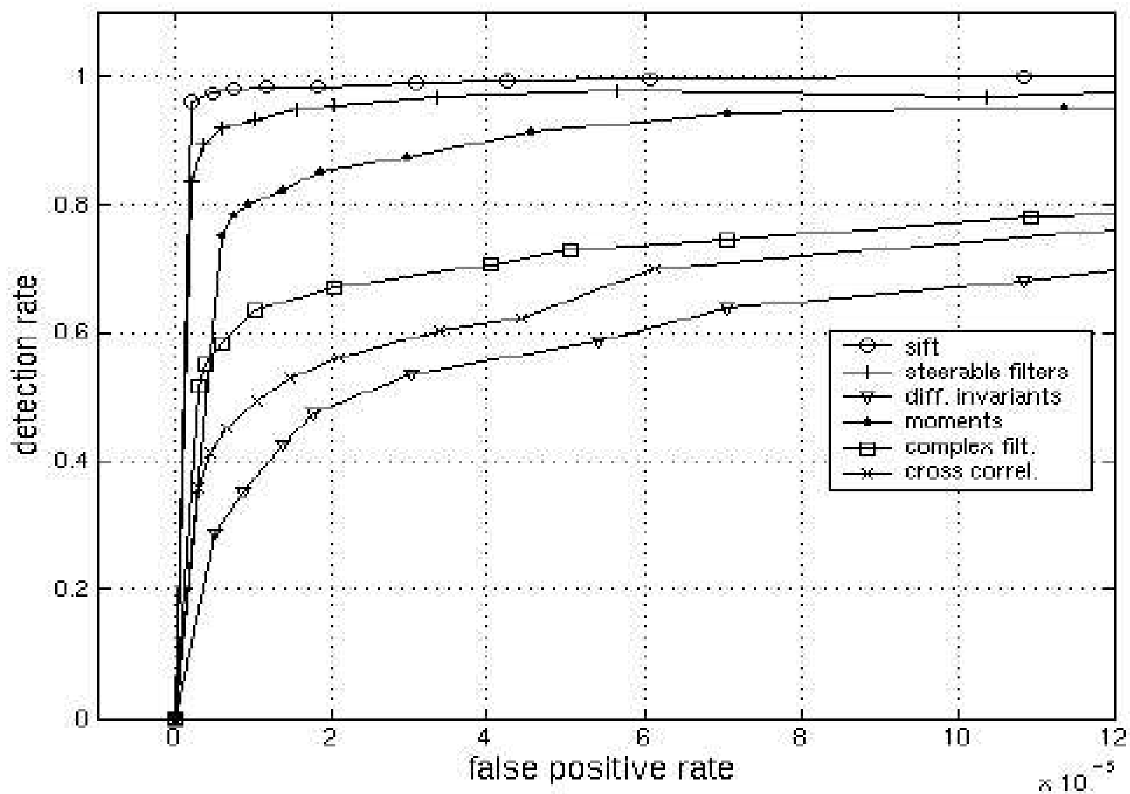
1. SIFT descriptor can be viewed as a 3–D histogram in which two dimensions correspond to image spatial dimensions and the additional dimension to the image gradient direction (normally discretised into 8 bins)



SIFT – Scale Invariant Feature Transform[1]

1. Empirically found[2] to show very good performance, invariant to *image rotation*, *scale*, *intensity change*, and to moderate *affine* transformations

Scale = 2.5
Rotation = 45^0



[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

SIFT invariances

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

SIFT Descriptor

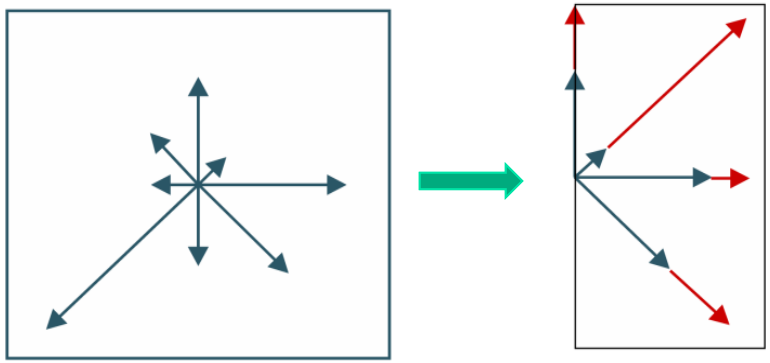
1. By far the most commonly used distinguished region descriptor:
   1. fast
   2. compact
   3. **works for a broad class of scenes**
   4. source code available
2. large number of ad hoc parameters $\Rightarrow$ Enormous follow up literature on both "improvements" and improvements [HoG, Daisy, Cogain]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

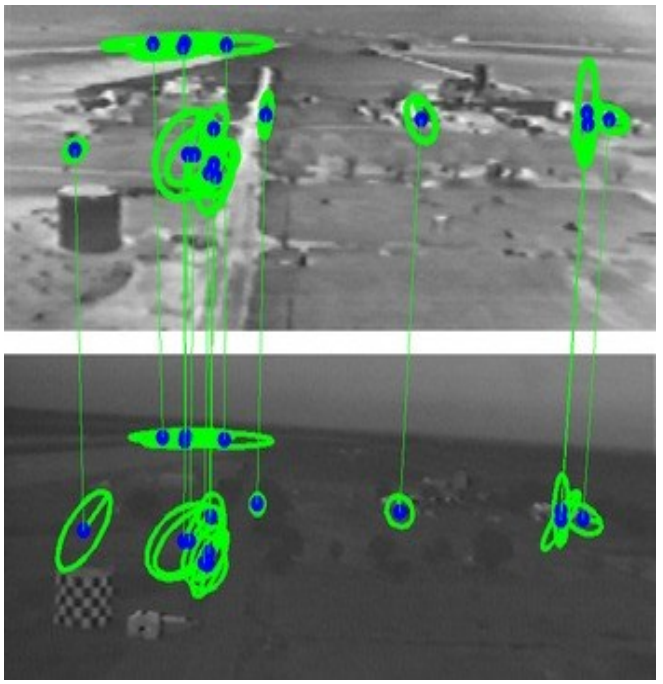
Illumination and sensor

WxBS features:

1. **Adaptive thresholding** for detect point in low-contrast regions: if #MSERs $< \theta_{MSER}$, or #HesAffs $< \theta_{HA}$, lower a detection threshold
2. Use HalfRootSIFT for description.

SIFT bins **HalfSIFT** bins

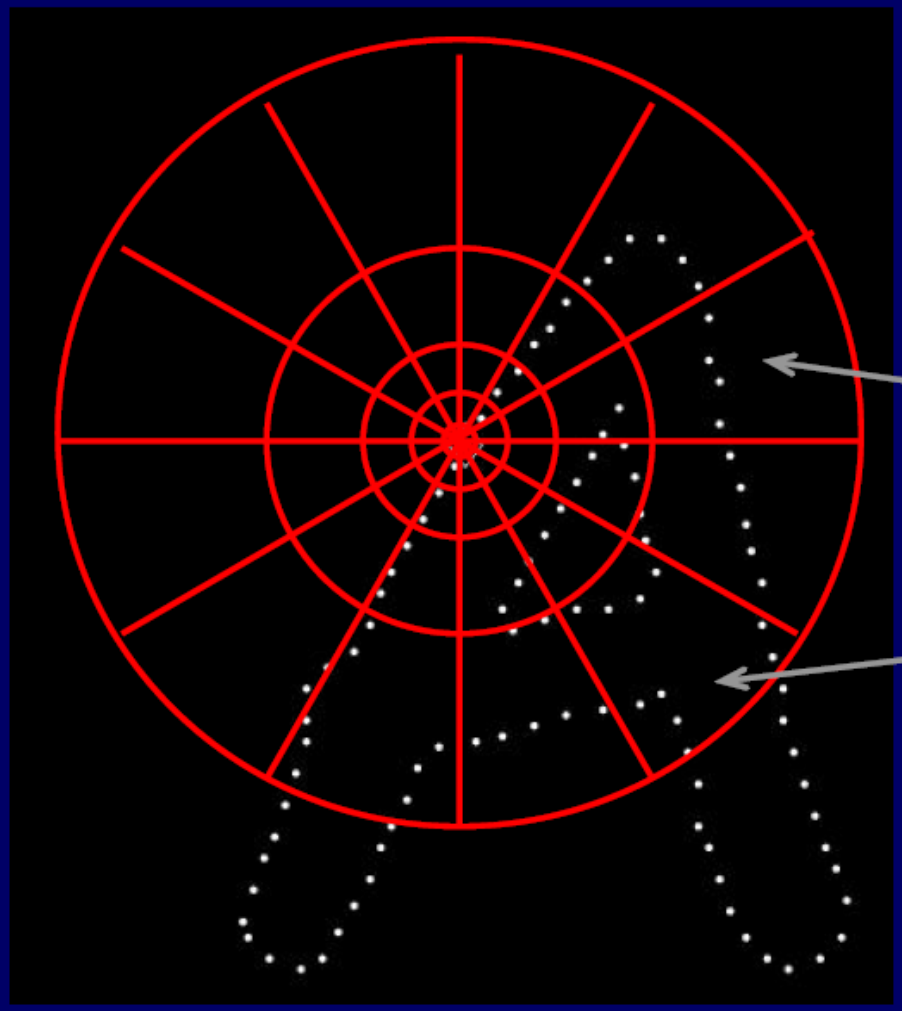




J. Chen, et a;/. **A partial intensity invariant feature descriptor for multimodal retinal image registration**. IEEE Transactions on Biomedical Engineering, 2010

D. Mishkin, M. Perdoch, J. Matas and K. Lenc. **WxBS: Wide Baseline Stereo Generalizations**, arXiv 2015,

Shape Context



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

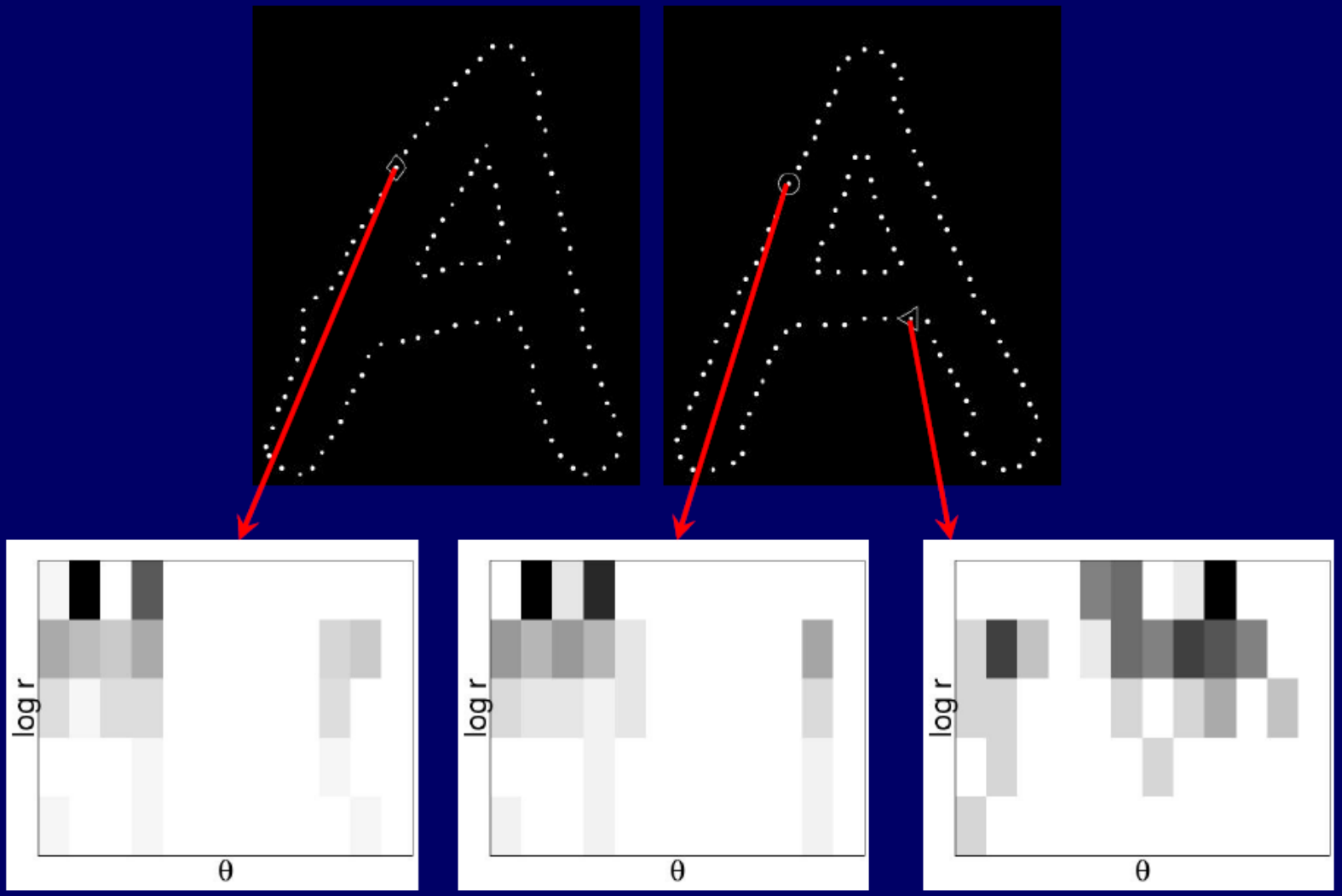
Count = 4

⋮

Count = 10

☞ Compact representation of distribution of points relative to each point

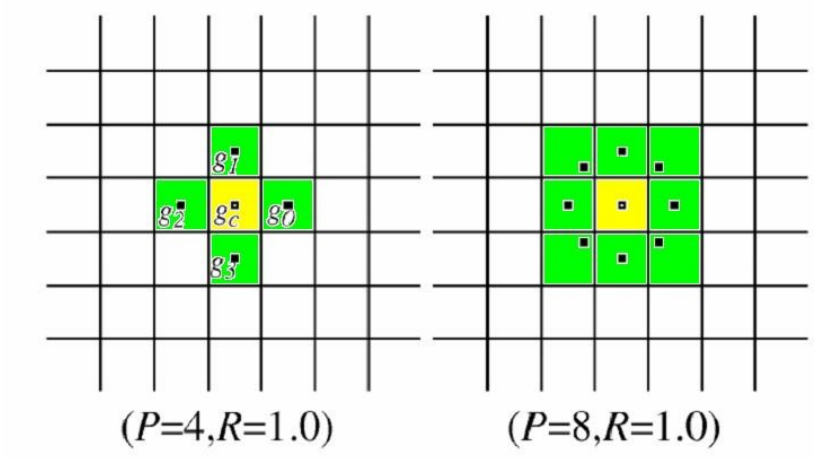
Shape Context



Local Binary Pattern (LBP) Descriptor

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

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



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

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

LBP values in a 3 x 3 block

The LBP descriptor is invariant to any monotonic transformation of image

Rotation Invariant LBP …

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

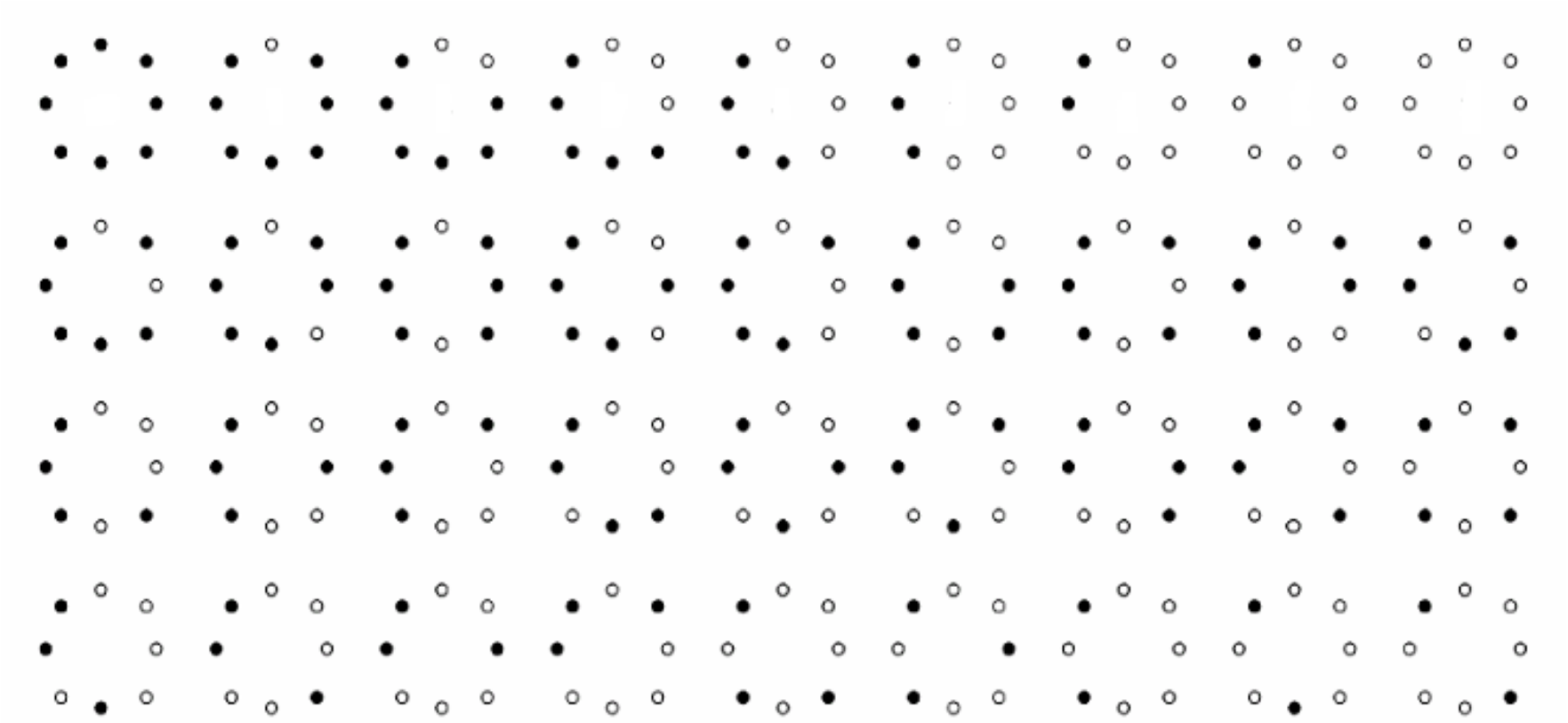
$$LBP_{P,R}^{ri} = \min\left\{ ROR(LBP_{P,R}, i) \;\middle|\; i = 0,1,\ldots,P-1 \right\}$$

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 $LBP_{8,1}$.

Rotation Invariant LBP …

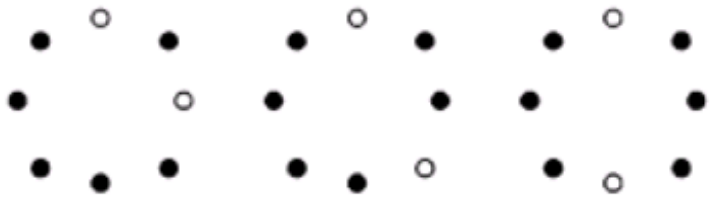
- This figure shows 36 unique rotation invariant binary patterns.



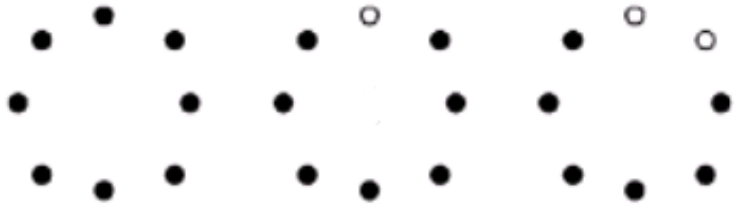
slide credit: Sara Arasteh et al.

Rotation Invariant LBP …

1. Rotation Invariant LBP patterns include:
    1. Uniform patterns
        1. At most two transitions from 0 to 1
    2. Non-uniform patterns
        1. More than two transitions from 0 to 1



Samples of non-uniform patterns



Samples of uniform patterns

slide credit: Sara Arasteh et al.

Uniform LBP (ULBP)

1. 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.

1. They function as templates for microstructures such as :
   1. Bright spot (0)
   2. Flat area or dark spot (8)
   3. Edges of varying positive and negative curvature (1-7)

Uniform Local Binary Patterns

LBPs are popular, numerous modifications exist

slide credit: Sara Arasteh et al.

BREIF (Calonder et al. ECCV2010)

Binary Robust Independent Elementary Features:

1. Defines binary test on patch $\mathbf{p}$ of size $S \times S$:

$$\tau(\mathbf{p};\mathbf{x},\mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases}$$

   where $\mathbf{p(x)}$ is the smoothed pixel intensity at $\mathbf{x} = (u;v)$

1. Choose a set of n_d $(\mathbf{x},\mathbf{y})$ - location pairs
   1. Uniquely defines a set of binary tests

2. BRIEF descriptor (n_d - dimensional bitstring):

$$f_{n_d}(\mathbf{p}) = \sum_{1 \le i \le n_d} 2^{i-1} \tau(\mathbf{p};\mathbf{x_i},\mathbf{y_i})$$

3. Compare by Hamming distance

BRIEF: parameters

1. Smoothing kernels
   1. Gaussian kernel; the best: variance: 2, window size: 9x9
2. Spatial arrangement of the binary tests (the best G II)

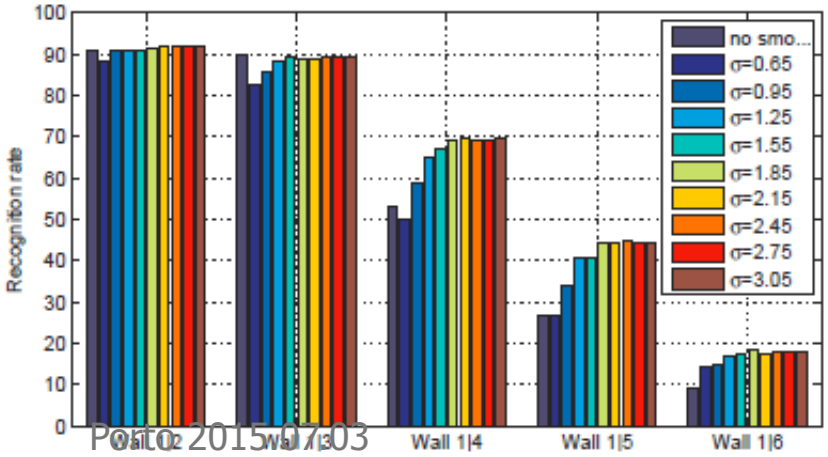
I. (X,Y)~i.i.d. Uniform $(-\frac{S}{2}, \frac{S}{2})$

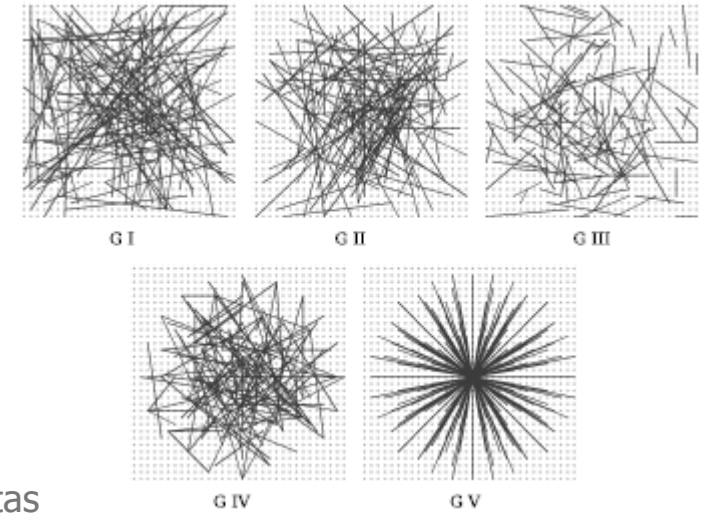
II. (X,Y)~i.i.d. Gaussian $(0, \frac{1}{25}S^2)$

III. X~i.i.d. Gaussian $(0, \frac{1}{25}S^2)$, Y~i.i.d. Gaussian $(\mathbf{x}_i, \frac{1}{100}S^2)$

IV. Randomly sampled from discrete locations of a coarse polar grid introducing a spatial quantization.

V. $\forall i : \mathbf{x}_i = (0,0)^T$ and $\mathbf{y}_i$ takes all possible values on a coarse polar grid containing points





BRIEF: pros & cons

1. Pros:
   1. Very fast (35- to 41-fold speed-up over SURF)
   2. Higher recognition rates (when rotations and scale changes are not required)

2. Cons:
   1. Fails when large plane rotations are present
   2. Bad/no scale invariance
   3. Smoothing part the most time consuming

ORB (Rublee et al. ICCV2011)

1. ORB = Oriented FAST and Rotated BRIEF
2. oFAST: FAST Keypoint Orientation
   1. Order FAST keypoints based on the Harris measure and pick the top *N* points
   2. To achieve multiscale: an image scale pyramid is employed
   3. Orientation by **Intensity Centroid**:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad \text{where} \quad m_{pq} = \sum_{x,y} x^p y^q I(x,y)$$

      1. Orientation of the patch:

$$\theta = \text{atan2}(m_{01}, m_{10})$$ (atan2 ... quadrant-aware version of arctan)

      2. Moments computed in a circular region of size *r*, which is set to be the patch size

ORB: rBRIEF

CENTER FOR MACHINE PERCEPTION

1. rBRIEF: Rotation-Aware BRIEF
   1. Smoothing by integral image, where each test point is a 5 x 5 subwindow of a 31 x 31 pixel patch
   2. Define the 2 x *n* matrix for any feature set of binary tests:

$$S = \begin{pmatrix} \mathbf{x}_1, \ldots, \mathbf{x}_n \\ \mathbf{y}_1, \ldots, \mathbf{y}_n \end{pmatrix}$$

   3. Define the new operator:

- ... original BRIEF operator

where $g_n(\mathbf{p}, \theta) = f_n(\mathbf{p} | (\mathbf{x_i}, \mathbf{y_i}) \in \mathbf{S}_\theta$ and f_n is the rotation matrix corresponding to patch oriantation θ

$\mathbf{S}_\theta = \mathbf{R}_\theta \mathbf{S}$ $\mathbf{R}_\theta$

1. Learning good binary features:
   1. All possible binary tests are searched to find those that both have high variance as well as being uncorrelated

ORB: rBRIEF Algorithm

CENTER FOR MACHINE PERCEPTION

1. Algorithm = greedy search for a set of uncorrelated tests with means near 0.5:
   1. Run each test against all training patches
   2. Order the tests by their distance from a mean of 0.5, forming the vector T
   3. Greedy search:
      a. Put the first test into the results vector R and remove it from T
      b. Take the next test from T and compare it against all test in R. If its absolute correlation is grater than a threshold, discard it; otherwise add it to R
      c. Repeat the previous step until there are 256 tests in R. It there are fewer than 256, raise the threshold and try again

ORB: properties

CENTER FOR MACHINE PERCEPTION

1. Invariant to rotations, relatively immune to Gaussian noise

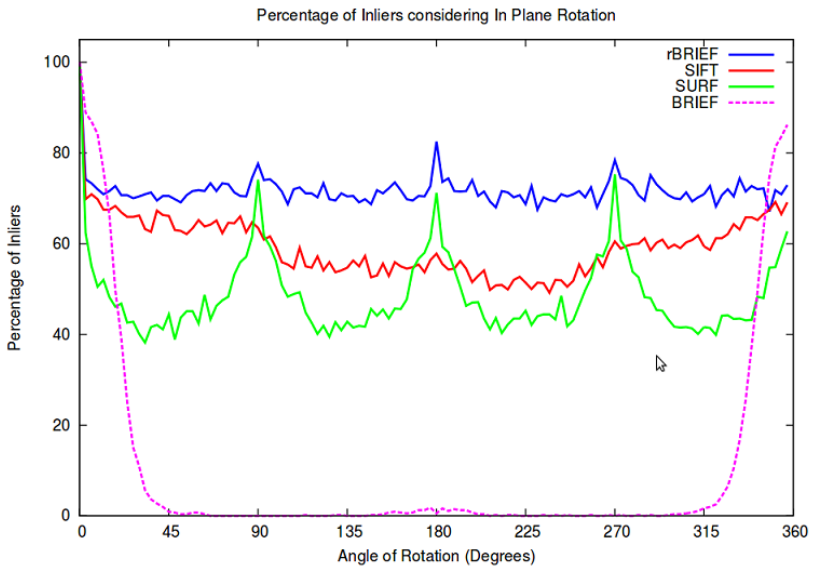


Figure 7. Matching performance of SIFT, SURF, BRIEF with FAST, and ORB (oFAST +rBRIEF) under synthetic rotations with Gaussian noise of 10.

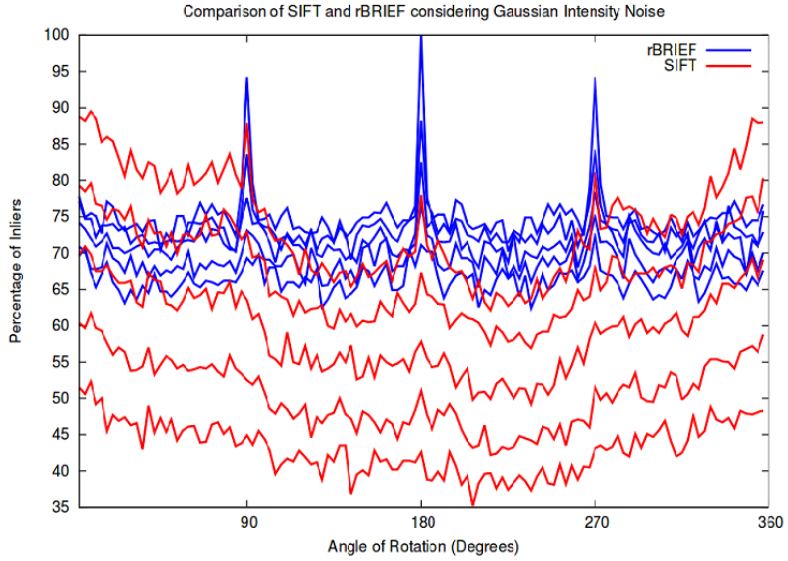


Figure 8. Matching behavior under noise for SIFT and rBRIEF. The noise levels are 0, 5, 10, 15, 20, and 25. SIFT performance degrades rapidly, while rBRIEF is relatively unaffected.

- Very fast comparing to SURF and SIFT

ORB:	Pyramid	oFAST	rBRIEF
Time(ms)	4.43	8.68	2.12

Detector	ORB	SURF	SIFT
Time/frame(ms):	15.3	217.3	5228.7

Two-view Matching

Common Structure of "Local Feature" Algorithms

CENTER FOR MACHINE PERCEPTION

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**

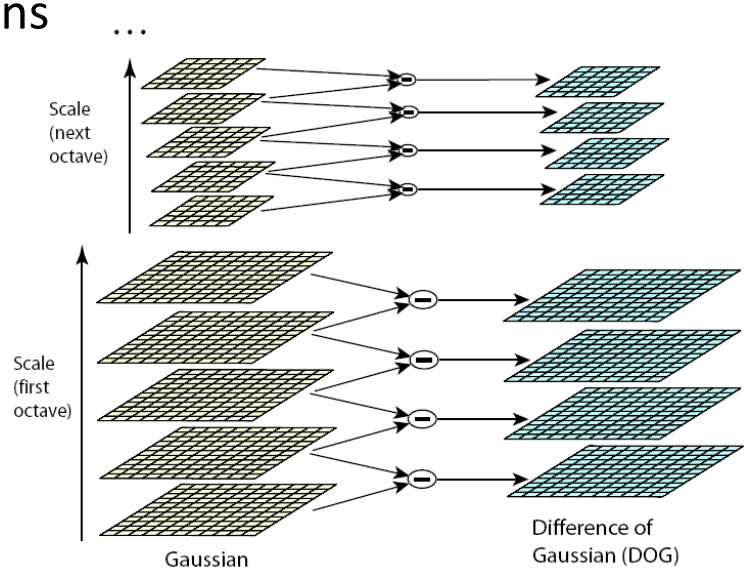
5. **Verification of global geometric arrangement**
   Confirms or rejects a candidate match

D. Lowe, *Object recognition from local scale-invariant features*, ICCV, 1999 2000 citations

CENTER FOR MACHINE PERCEPTION

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

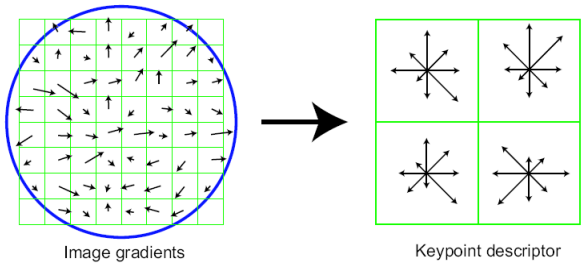
Matching:

- 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.

Nearest-neighbor matching

CENTER FOR MACHINE PERCEPTION

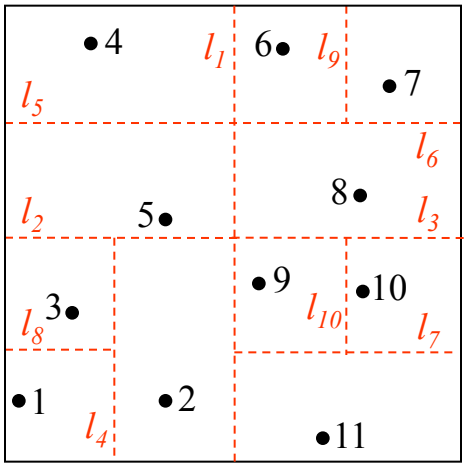
1. Solve following problem for all feature vectors, $\mathbf{x}$:

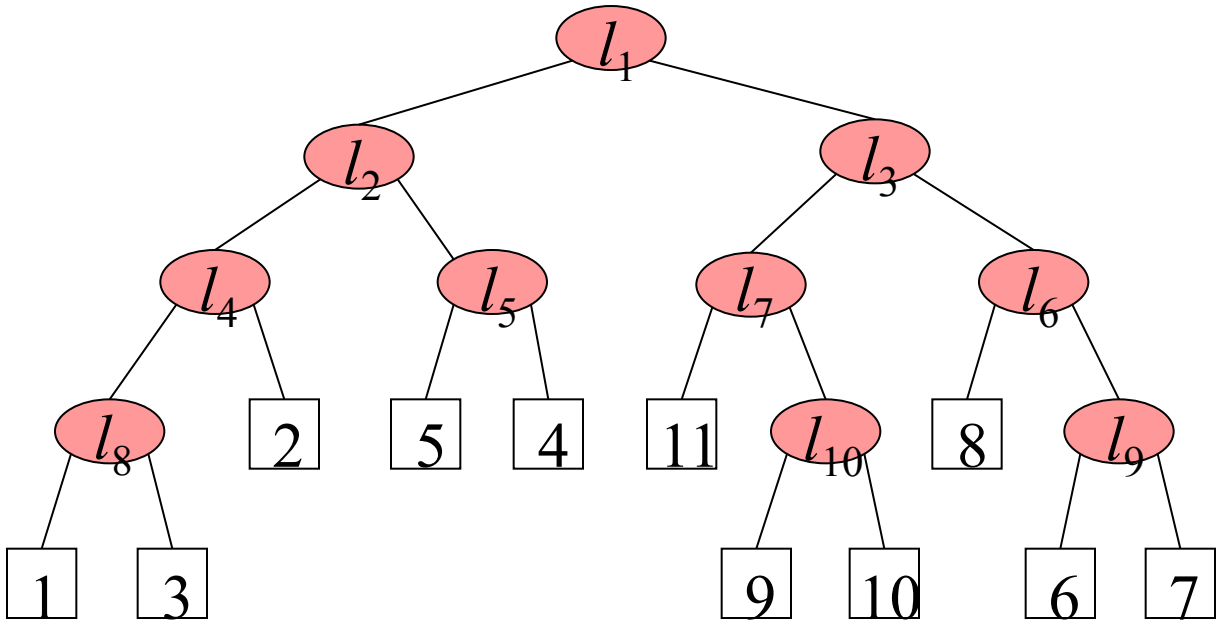
$$\forall j \; NN(j) = \arg\min_i ||\mathbf{x}_i - \mathbf{x}_j||, \; i \neq j$$

2. Nearest-neighbor matching is the major computational bottleneck
   1. Linear search performs dn^2 operations for n features and d dimensions
   2. No exact methods are faster than linear search for d>10 (?)
   3. 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

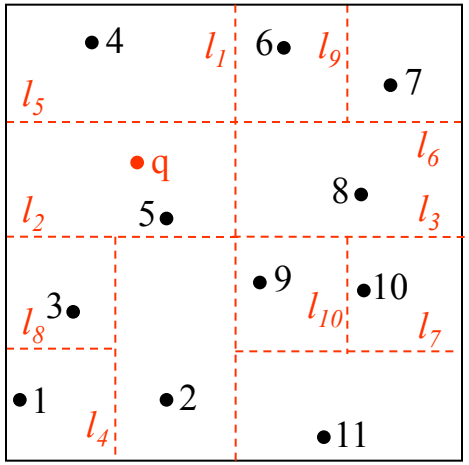
Simple 2D example

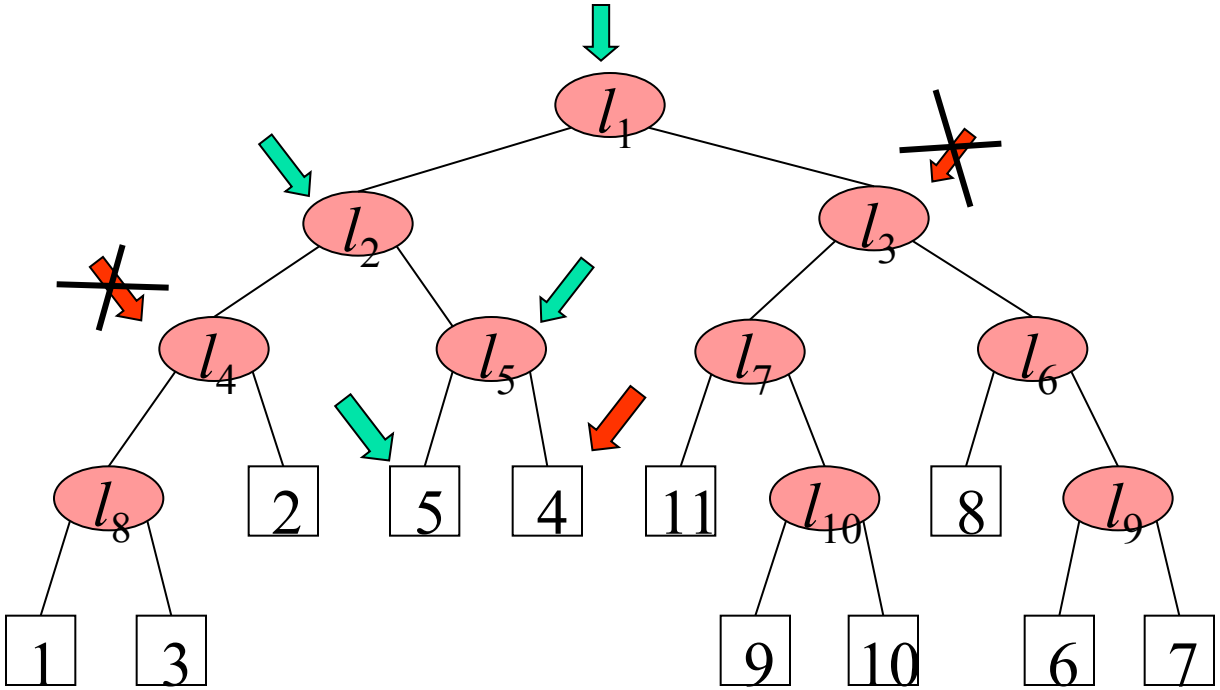




Slide credit: Anna Atramentov

K-d tree query



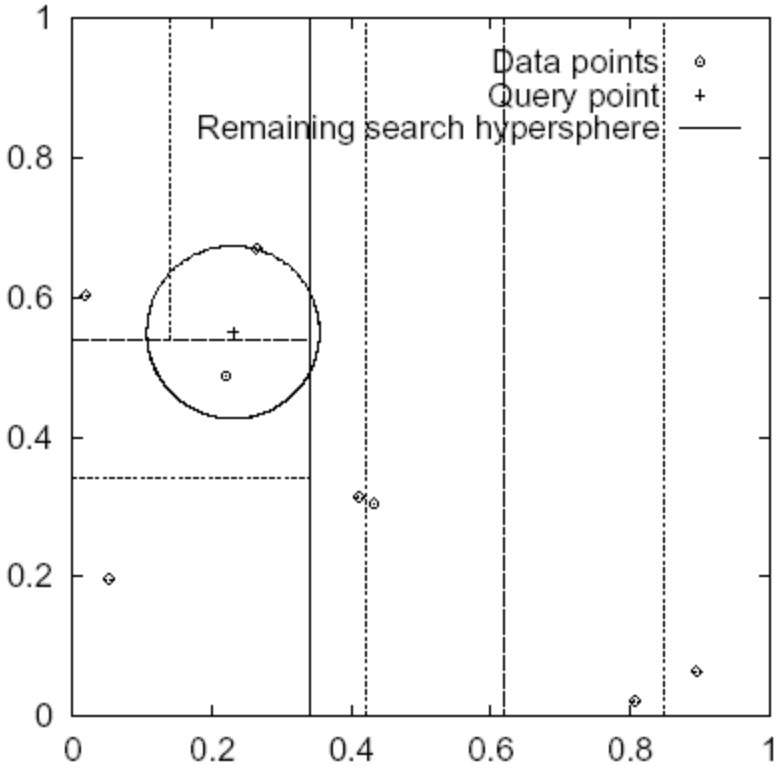


Slide credit: Anna Atramentov

Approximate k-d tree matching

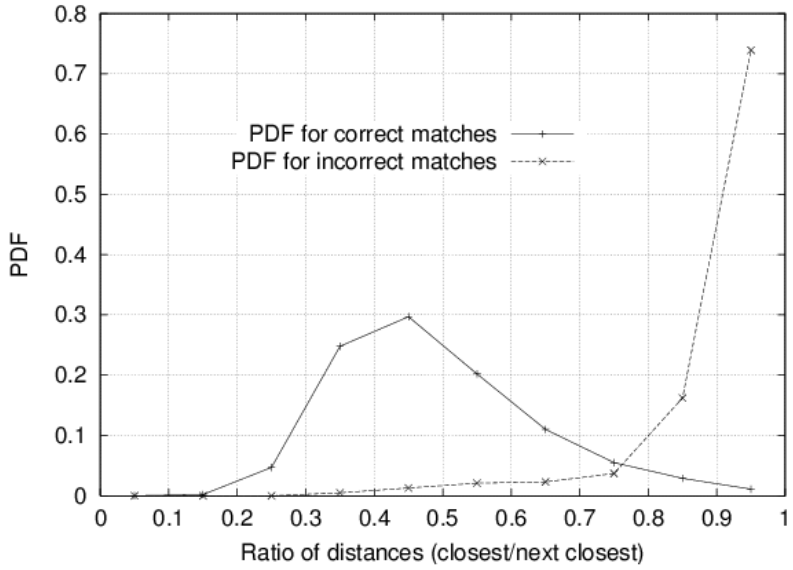
Key idea:

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



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
  1. Ratio will be high for features that are not distinctive
  2. Threshold of 0.8 provides good separation



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

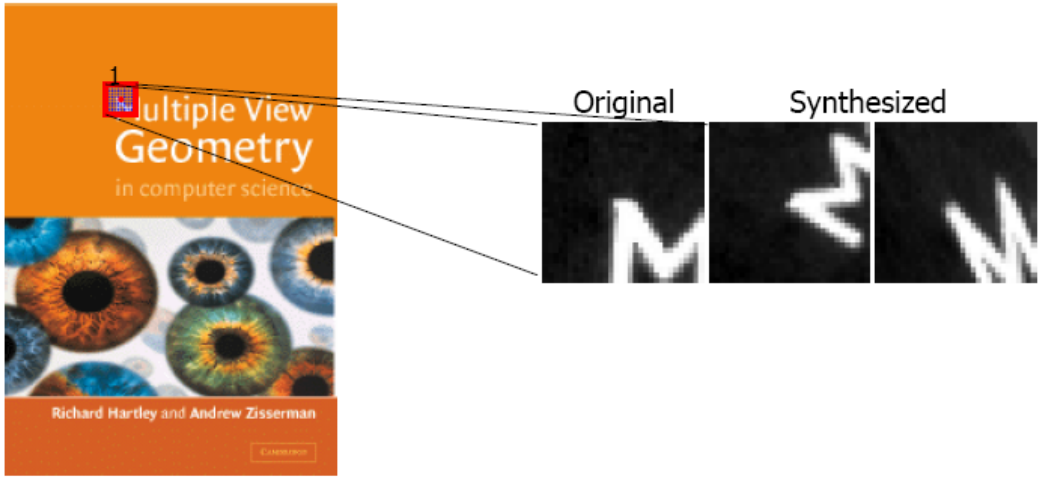
Randomized Forests

1. Feature matching as a classification problem

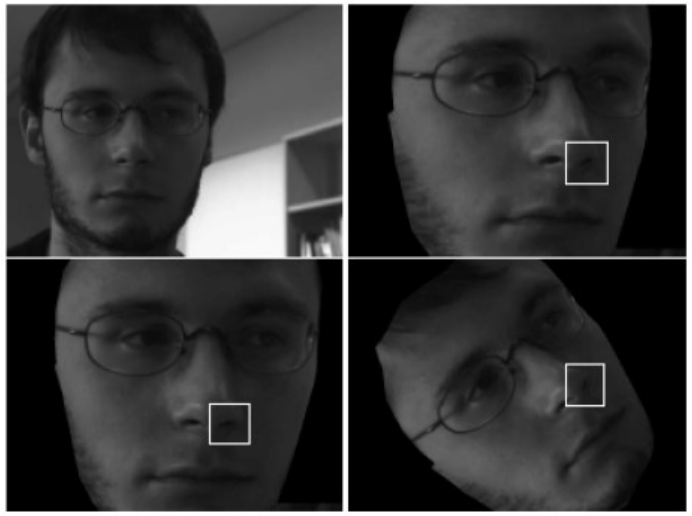
Lepetit, Lagger and Fua. Randomized Trees for Real-Time Keypoint Matching. CVPR 2005

Synthesize training examples

1. Planar object



3-D object



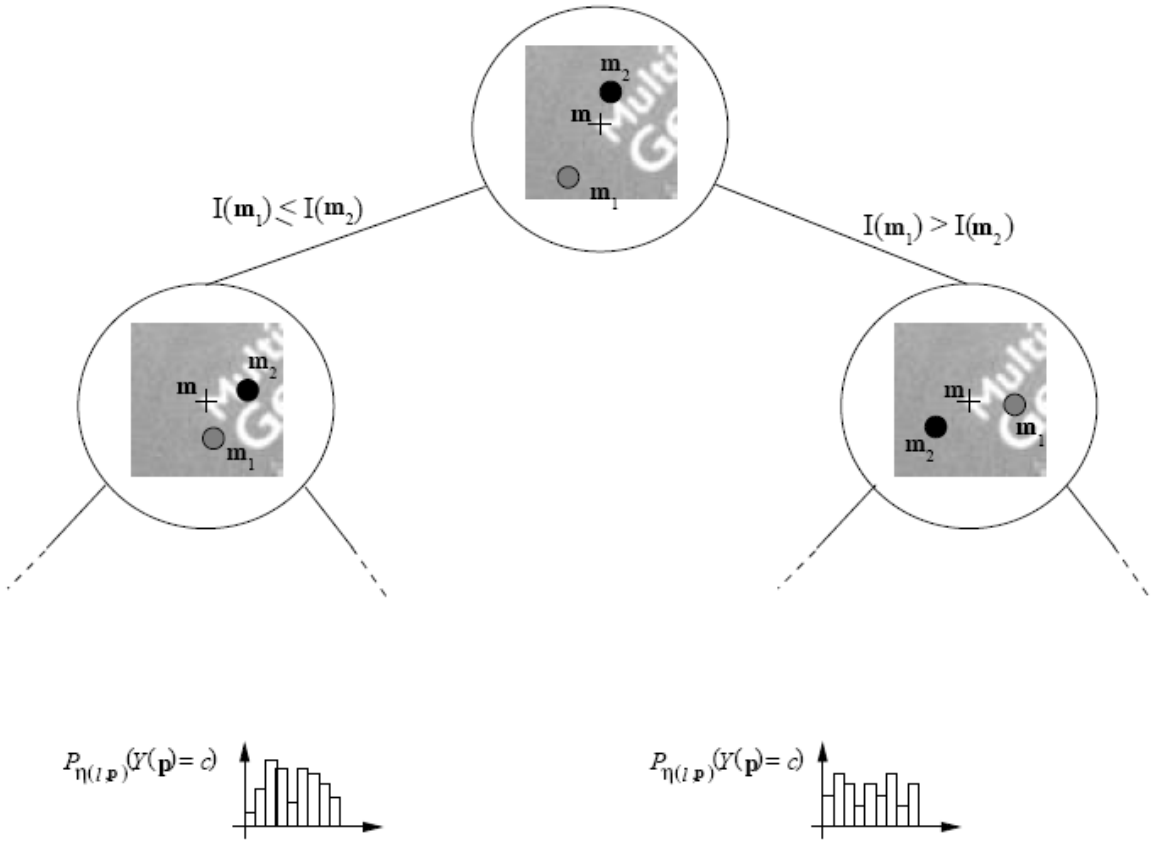
Lepetit, Lagger and Fua. Randomized Trees for Real-Time Keypoint Matching. CVPR 2005

Randomized Decision Tree

1. Compare intensity of pairs of pixels
2. In construction, pick pairs randomly

• Insert all training examples into tree

• Distribution at leaves is descriptor for the particular feature



Randomized Forests

CENTER FOR MACHINE PERCEPTION

1. Use multiple trees (i.e. forest) to improve performance

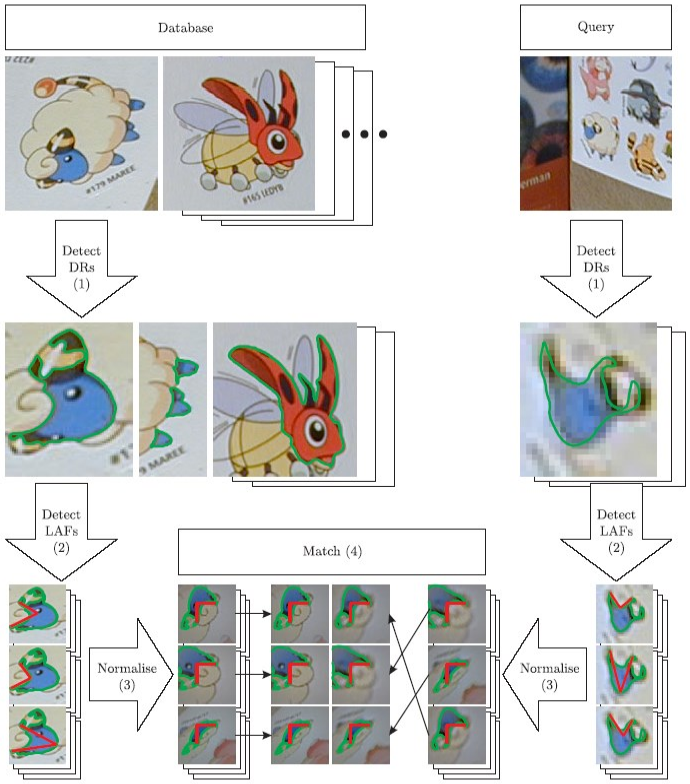
2. Very quick to compute in testing
   1. Just comparison of pairs of pixels
   2. Real-time performance

3. ~10x faster than SIFT, but slightly inferior performance

MSER-LAF-Tree, Obdrzalek and Matas, 2005 180 citations

CENTER FOR MACHINE PERCEPTION

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.)

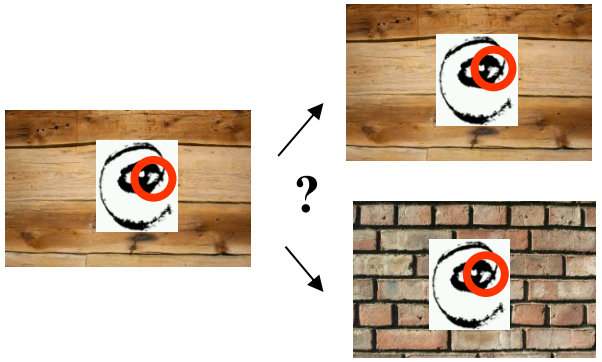


Matas, Chum, Urban, Pajdla: "Robust wide baseline stereo from maximally stable extremal regions". BMVC2002
Obdržálek and Matas: "Object recognition using local affine frames on distinguished regions". BMVC02
Obdržálek and Matas: "Sub-linear Indexing for Large Scale Object Recognition", BMVC 2005

Correspondence Verification

1. Difficult matching problems:
   1. Rich 3D structure with many occlusions
   2. Small overlap
   3. Image quality and noise
   4. (Repetitive patterns)

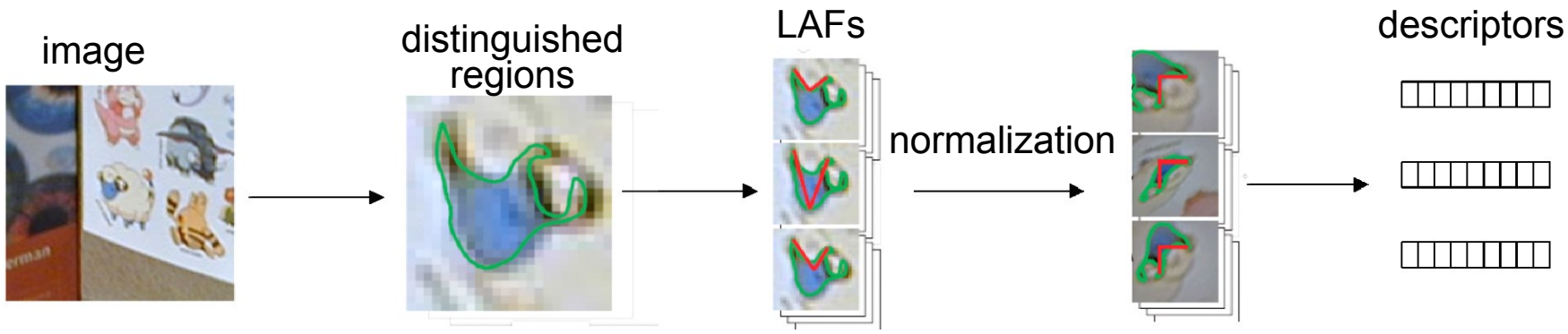
measurement region too large



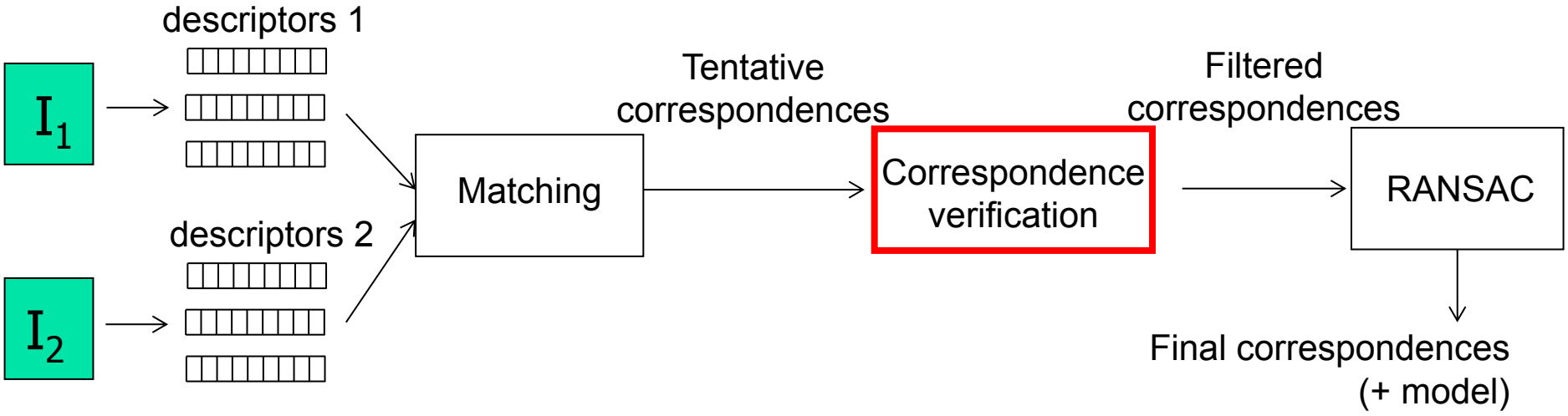
measurement region too small

Correspondence Verification

1. From image to local invariant descriptors



- Correspondence between two images



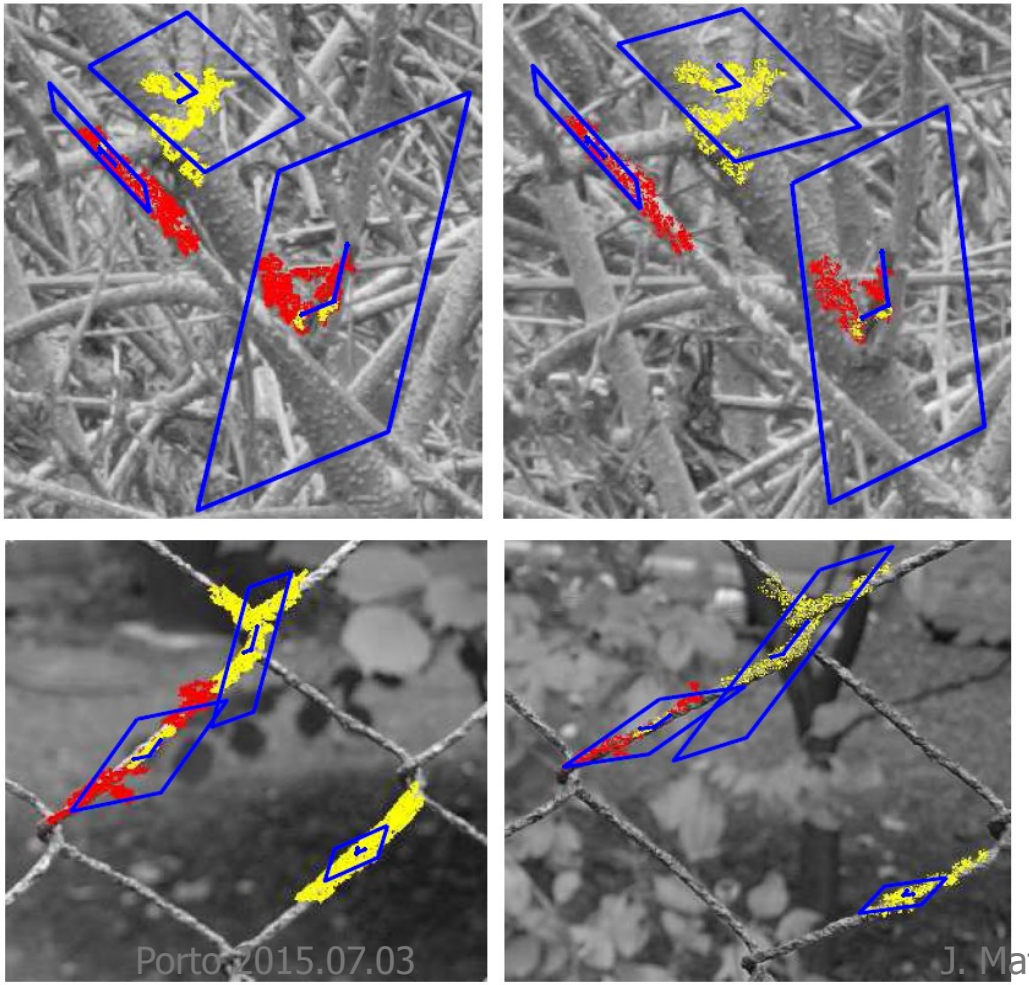
Correspondence Verification

CENTER FOR MACHINE PERCEPTION

1. Idea: “Look at both images simultaneously”

   => *Sequential Correspondence Verification by Cosegmentaion*

   [Čech J, Matas J, Perďoch M. IEEE TPAMI, 2010]



- **Input**: fixed number of tentative correspondences
- **Output**: Statistical Correspondence quality

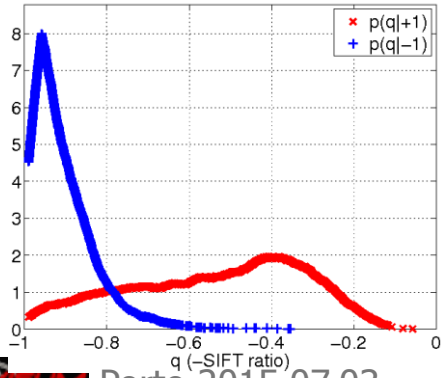
- A cosegmentaion process starts from LAF-correspondences to grow corresponding regions
- Various statistics are collected
- (Learned) Classifier to decide corresponding/non-correspond.

Correspondence Verification

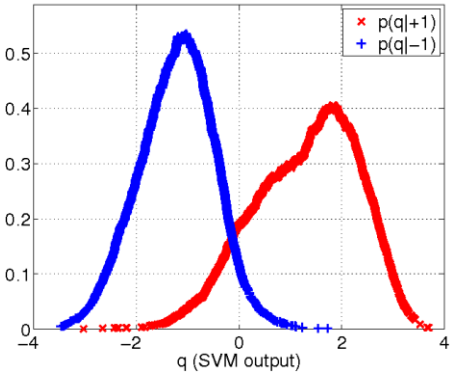
CENTER FOR MACHINE PERCEPTION

1. Learning a (sequential) classifier
   1. Training set from WBS images
   2. 16k LAF correspondences
      (40 % correct)

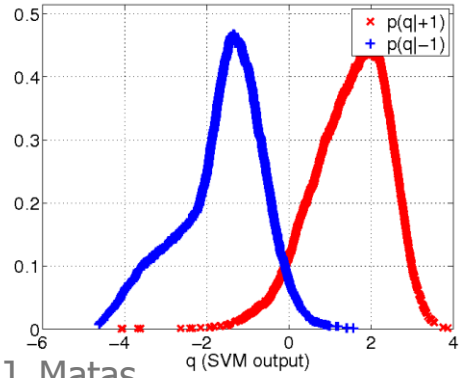
SIFT-ratio (only)



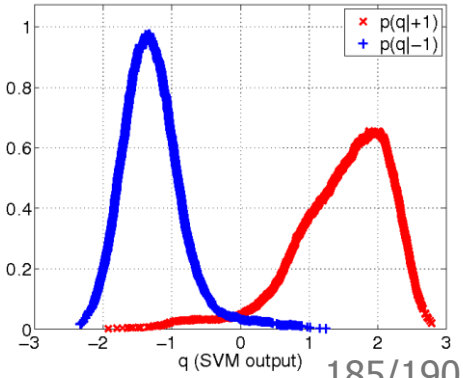
10 growing steps



100 growing steps

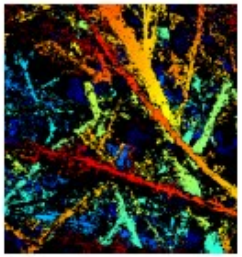


1000 growing steps

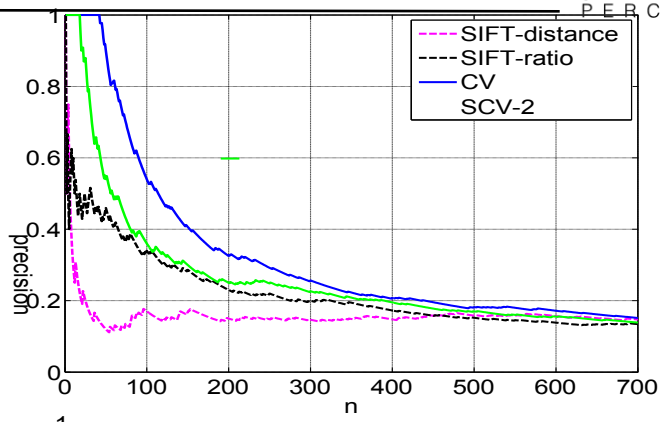


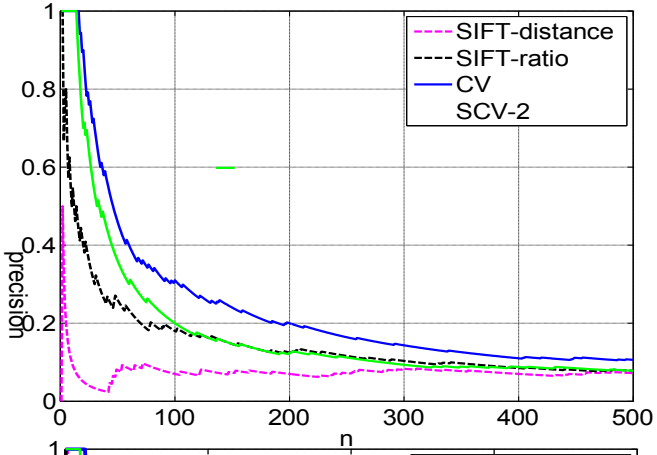
Experiments

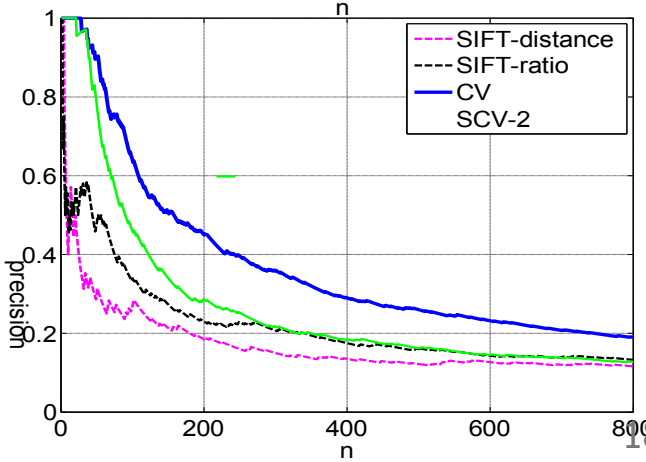




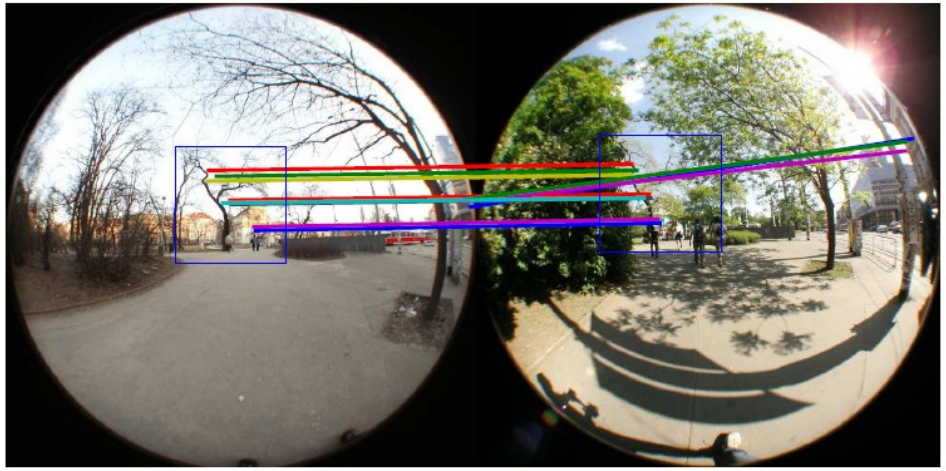


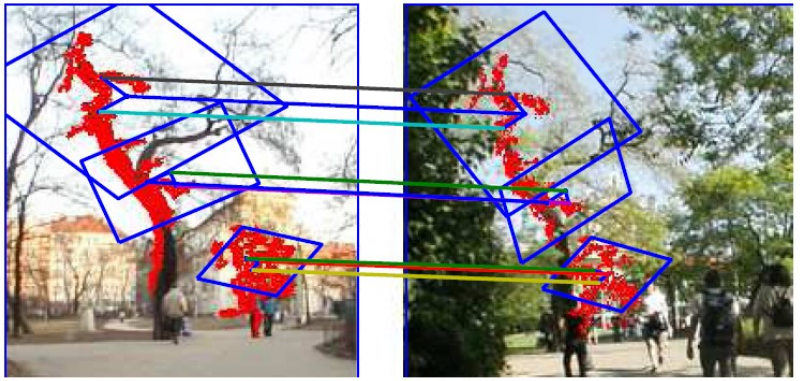






Correspondence Verification: Summary





1. high discriminability
   1. significantly outperforms a standard selection process based SIFT-ratio
2. very fast (0.5 sec / 1000 correspondences)
3. always applicable before RANSAC
4. the process generating tentative correspondences can be much more permissive
   1. 99% of outliers not a problem, correct correspondences recovered
   2. higher number of correct correspondences

Local Feature Methods: Analysis

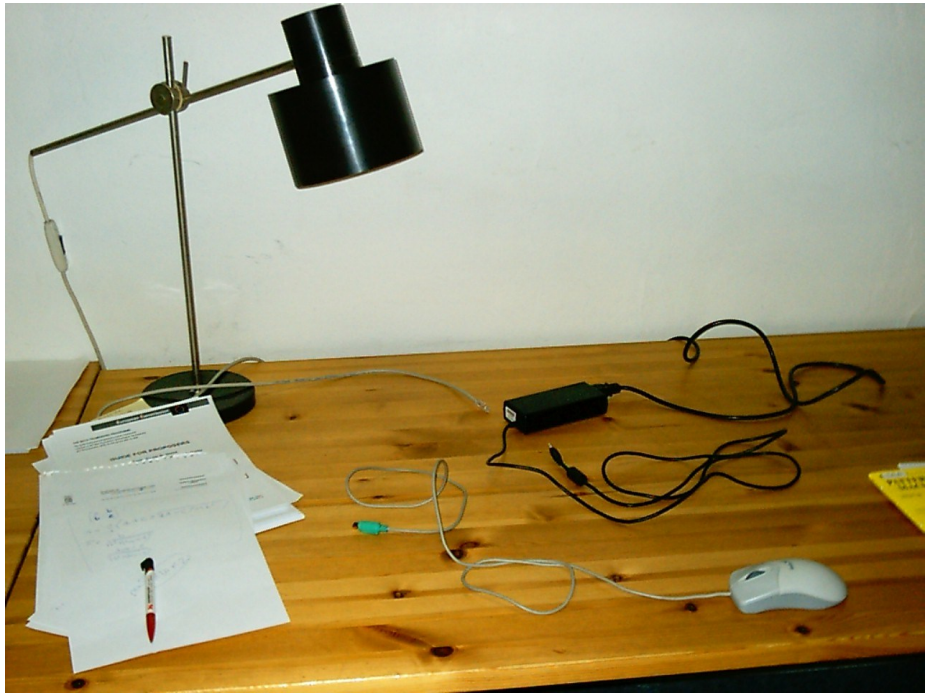
CENTER FOR MACHINE PERCEPTION

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 objects where it does not work is HUGE (examples are all around us).**

Where Local Features Fail:



Challenge: Elongated, Wiry and Flexible Objects

In this case: “no recognition without segmentation”?

macros.tex
sfmath.sty
cmpitemize.tex

Thank you for your attention.