Correspondence of Local Features for Wide-Baseline Matching, Image Retrieval, Stitching, 3D reconstruction and more ...

Jiří Matas, Ondra Chum, Jan Čech

Center for Machine Perception, Czech Technical University Prague

Includes slides by:

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

Outline

- Local features: introduction, terminology
- Motivation: generalisation of local stereo to wide-baseline stereo
- Example applications:
  panorama, 3D reconstruction, retrieval
- Challenges in correspondence problem
- 1. Detection of Local invariant features:
  - Harris, FAST
  - Scale invariant: SIFT, MSER, LAF
- 2. Descriptors
- 3. Matching
- (4.) Correspondence Verification
- Limitations
- 5. RANSAC (robust model fitting)

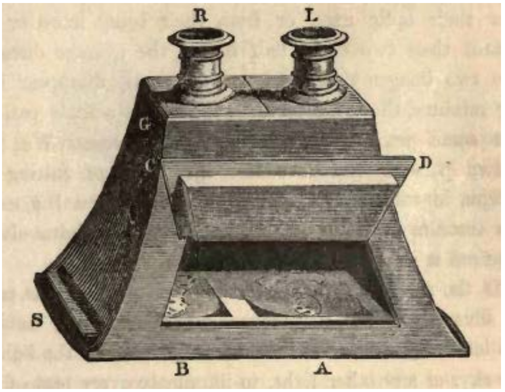
Local Features

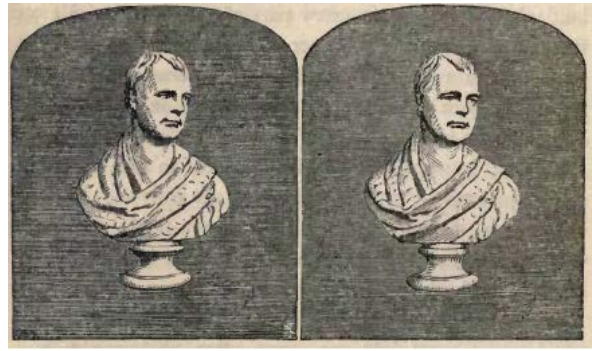
- Methods based on "Local Features" are the state-of-the-art for number of computer vision problems.

- E.g.: Wide-baseline stereo, image retrieval, 3D reconstruction

- Terminology (diverse, unfortunately) :
  Local Feature = Interest "Point" = The "Patch" =
  = Feature "Point"
  = Distinguished Region
  = (Transformation) Covariant Region

Narrow-baseline stereo

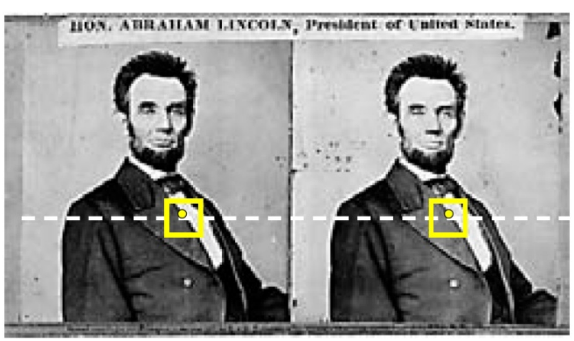




Brewster Stereoscope, 1856

A "photo" for both eyes

1. Local Feature (Region) = a rectangular "window"
   - robust to occlusion, translation invariant
   - windows matched by correlation, assuming small displacement
   - successful in Narrow-baseline stereo matching



Motivation: Generalization of Local Stereo to Wide Baseline Stereo (WBS)

2. 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.

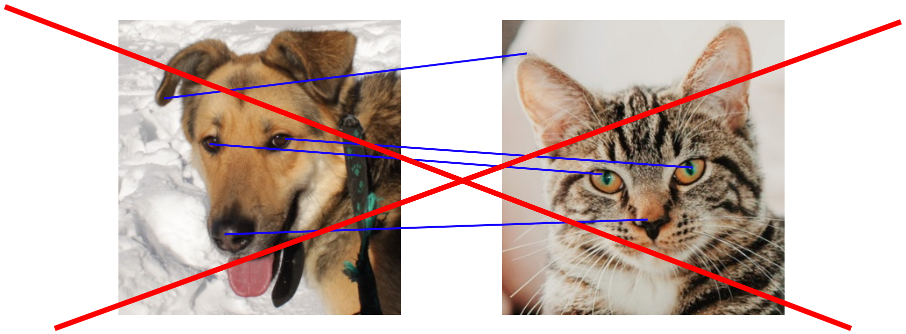
Local Features & The Correspondence Problem

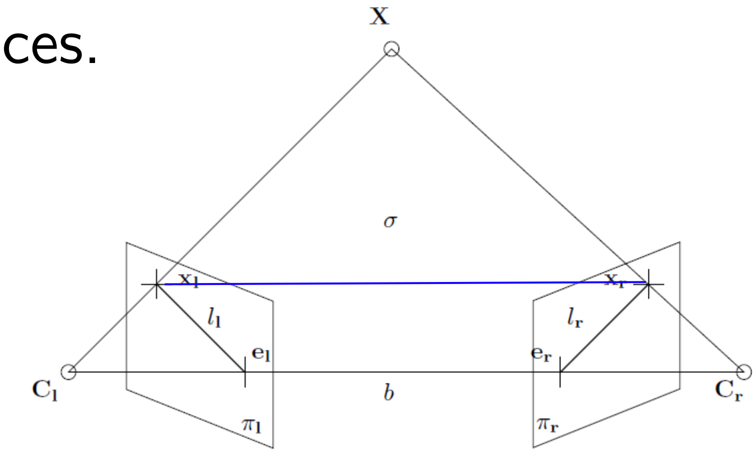
Establishing correspondence is the key issue in many computer vision problems:

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

Correspondence Problem

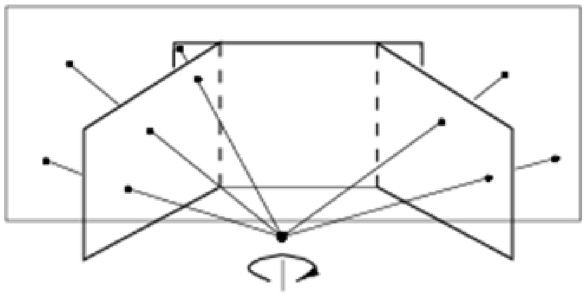
- What do we mean with a **local correspondence**?
  - Geometrical Instance correspondences.
    Not a semantic correspondence





  - Local correspondences
    - Not a global correspondence of entire related images

Local Features in Action (1): Building a Panorama



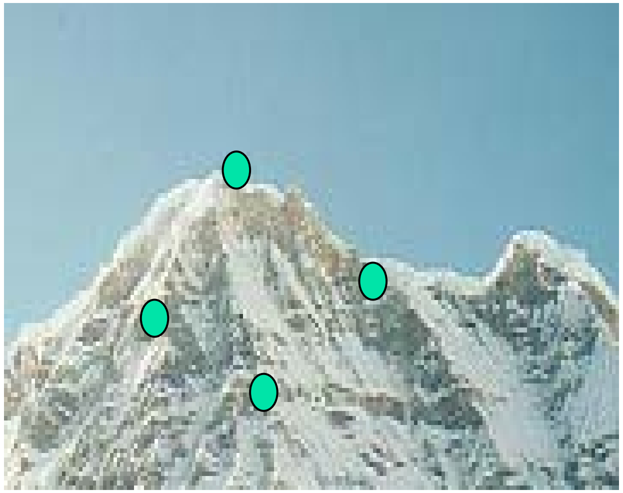
M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003

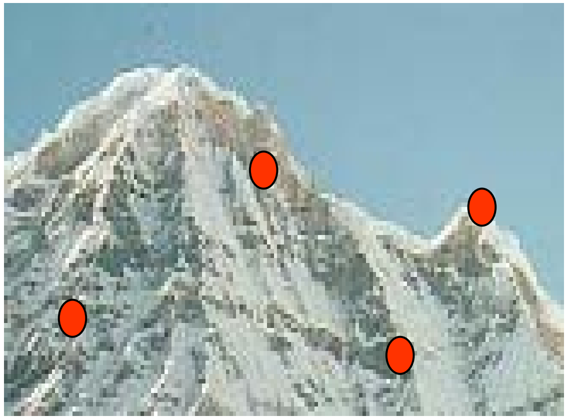
Local Features in Action (1): Building a Panorama

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

Local Features in Action (1): Building a Panorama

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





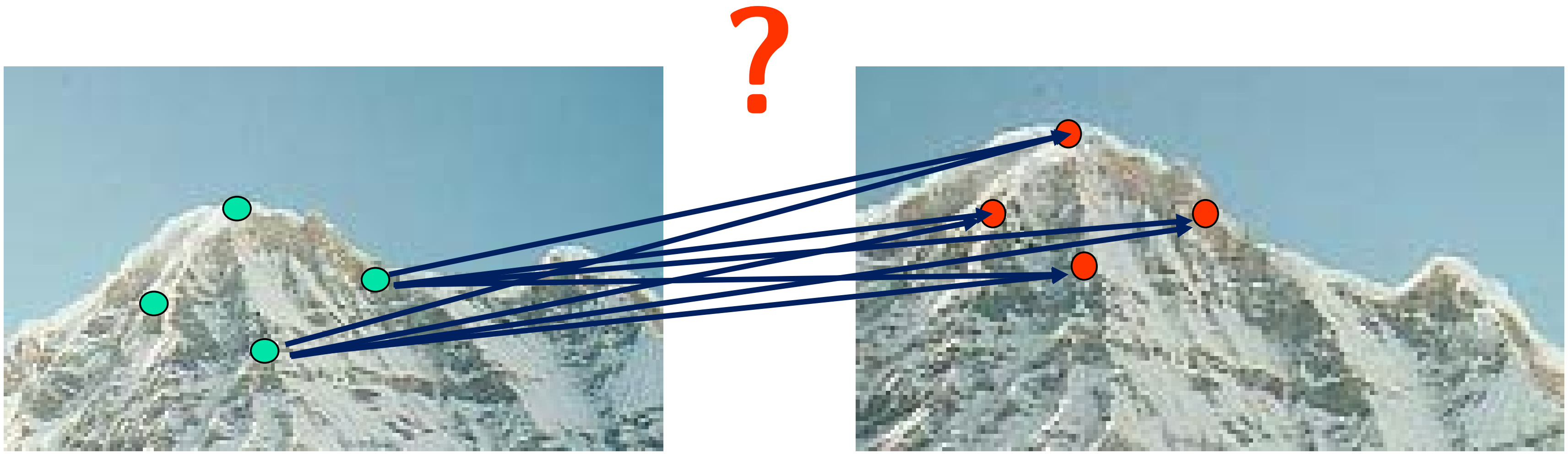
no chance to match!

A repeatable detector needed.

* Other methods exist that do not need independency

Local Features in Action (1): Building a Panorama

- Problem 2:
  - how to correctly recognize the corresponding features?



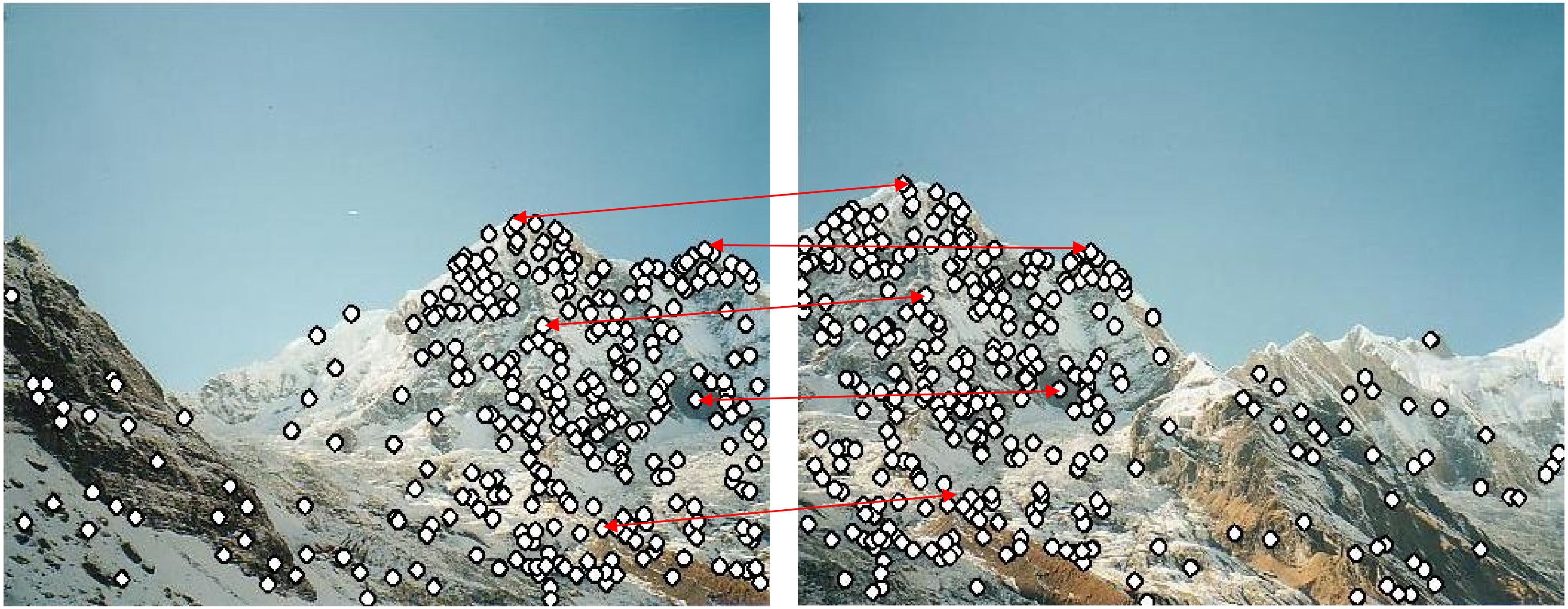
Solution:

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

Local Features in Action (1): 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 (1): Building a Panorama

Possible approach:

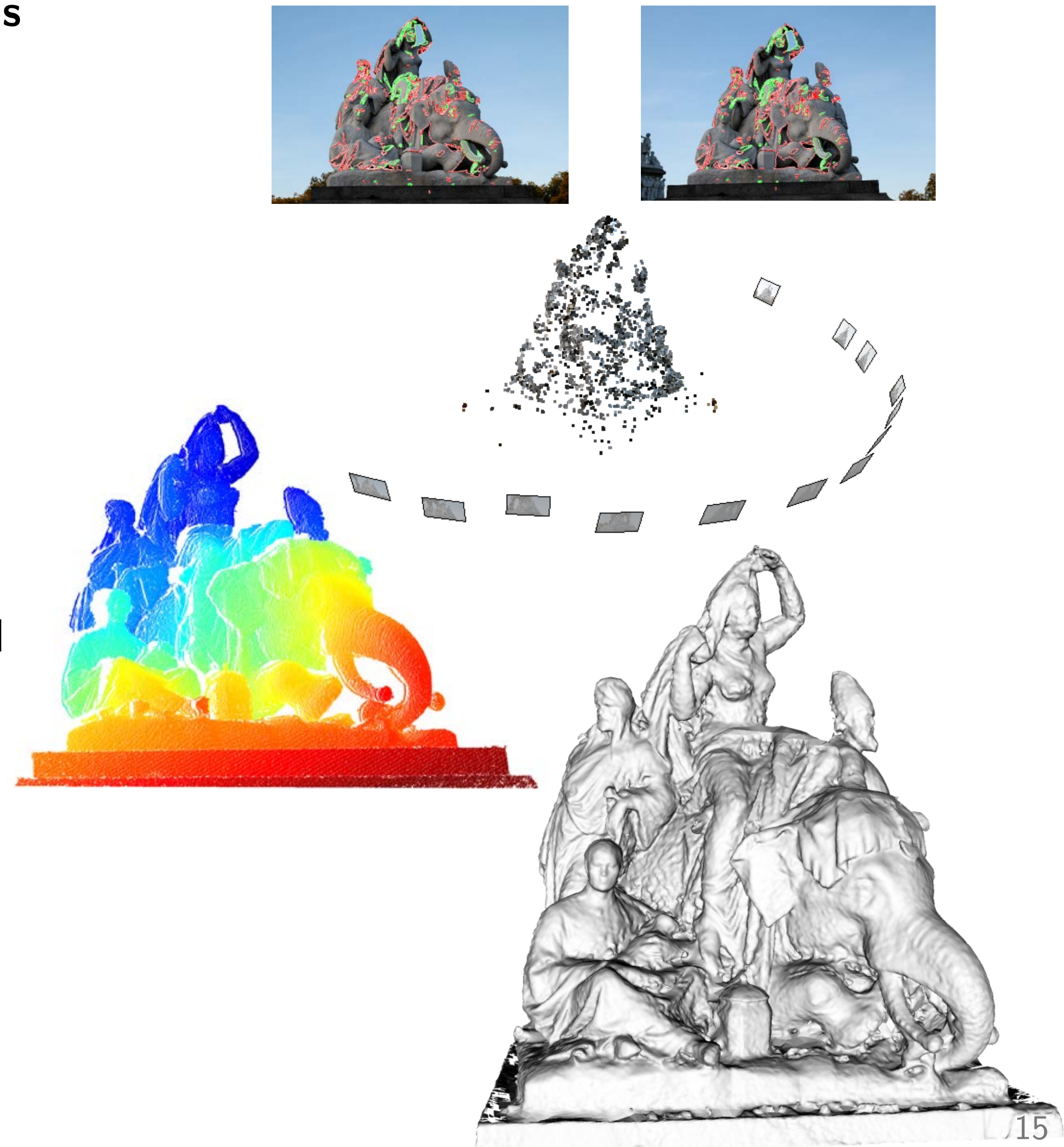
1. Detect features in both images
2. Find corresponding pairs
3. Estimate transformations (Geometry and Photometry)
4. Put all images into one frame, **blend**.

Local Features in Action (2): 3D reconstruction

- 3D reconstruction – camera pose estimation

Local Features in Action (2): 3D reconstruction

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



Matas et al., IVC 2004.
Martinec, Pajdla., CVPR 2007.
Cech, Sara, CVPR 2007.

Local Features in Action (2): 3D reconstruction

- Large scale 3D reconstruction – "Microsoft Photo Tourism"

57,845 downloaded images, 11,868 registered images.

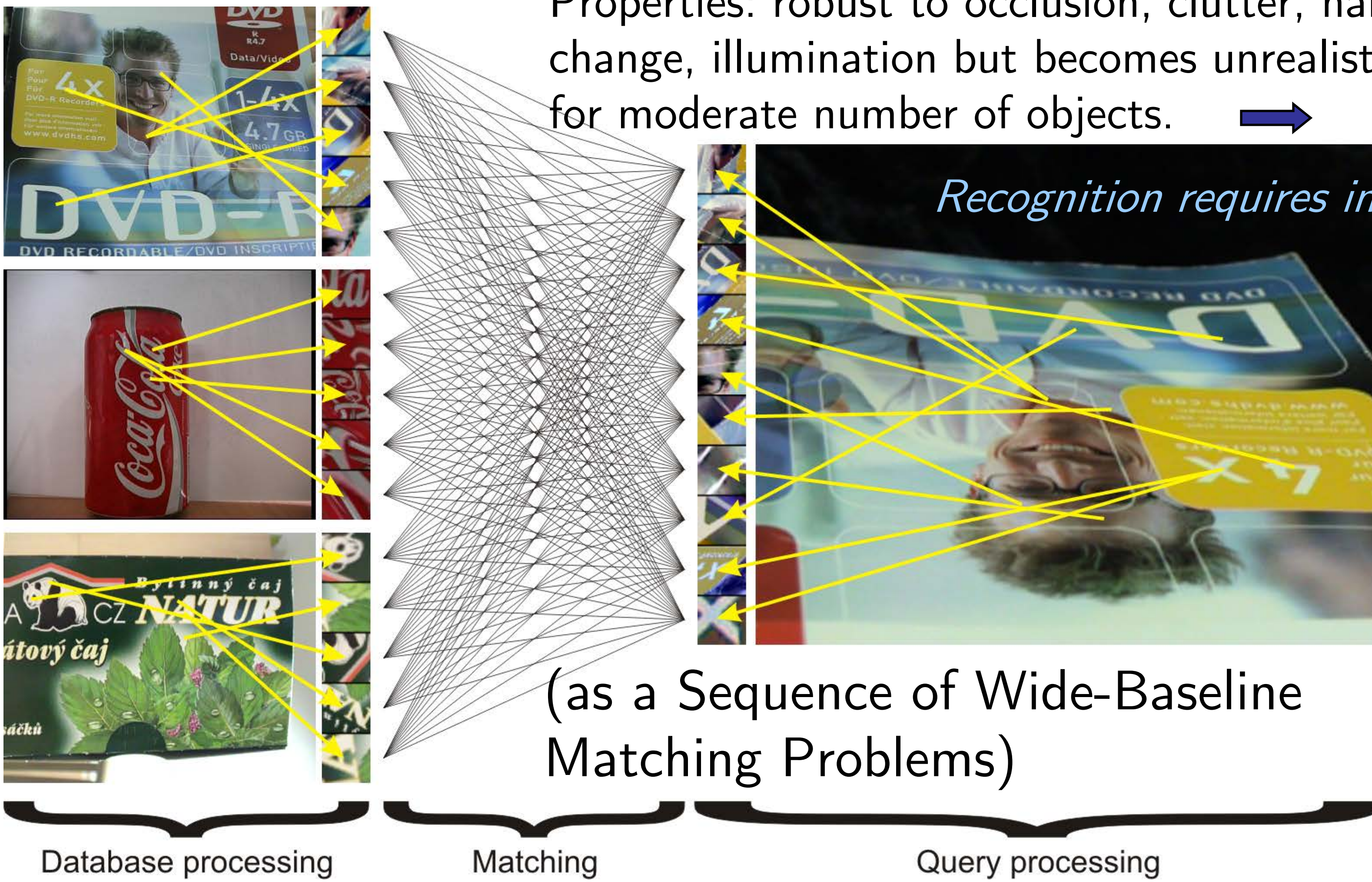
The Old City of Dubrovnik

Local Features in Action (3): "Recognition"

(as a Sequence of Wide-Baseline Matching Problems)

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

Recognition requires indexing

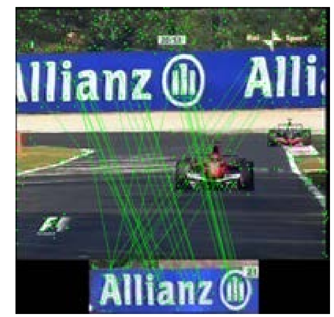


(as a Sequence of Wide-Baseline Matching Problems)

Local Features in Action (3): “Recognition”

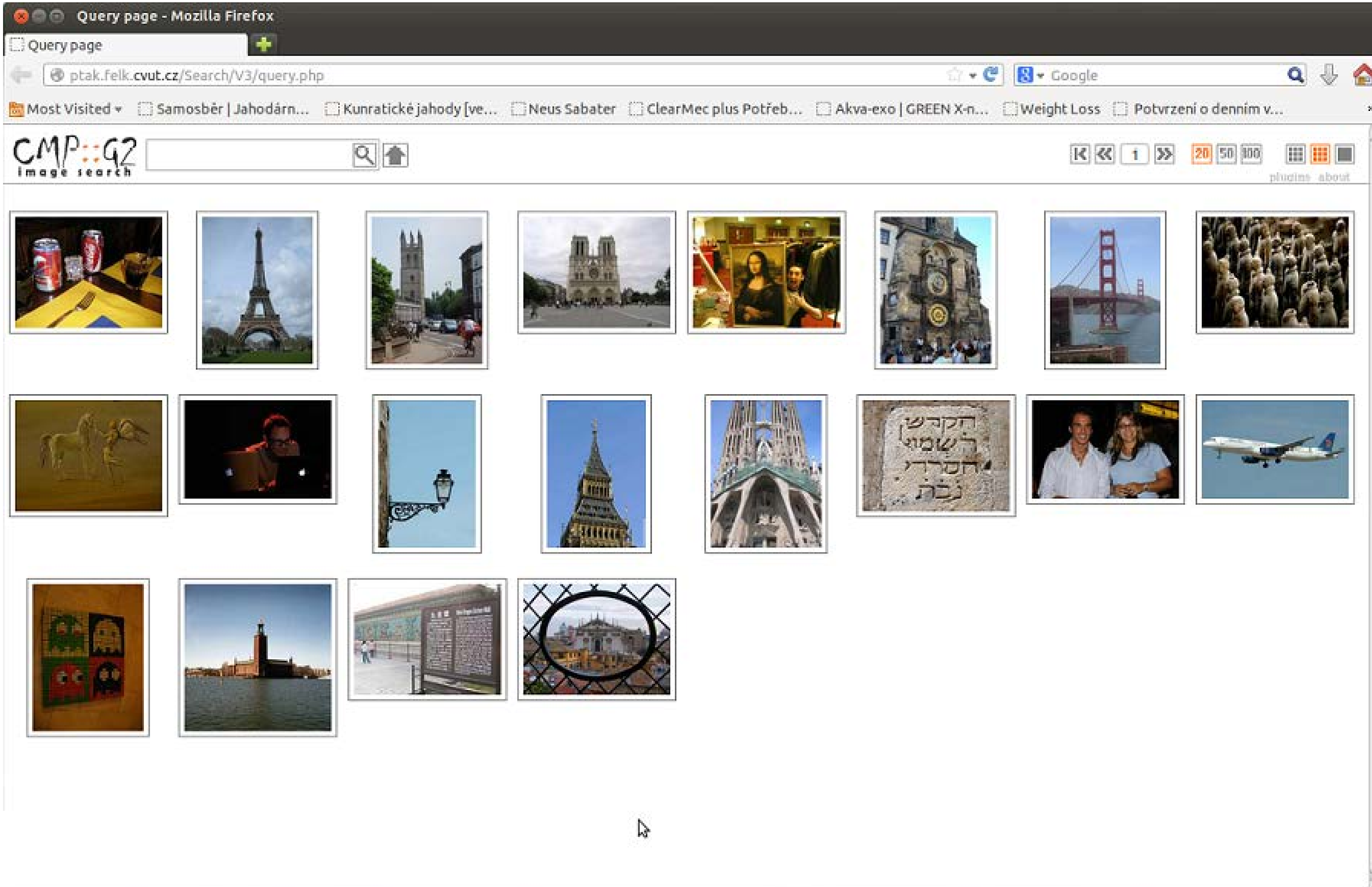
- **Applications**
  - In car traffic sign recognition

  - Product logos detection in TV/social media

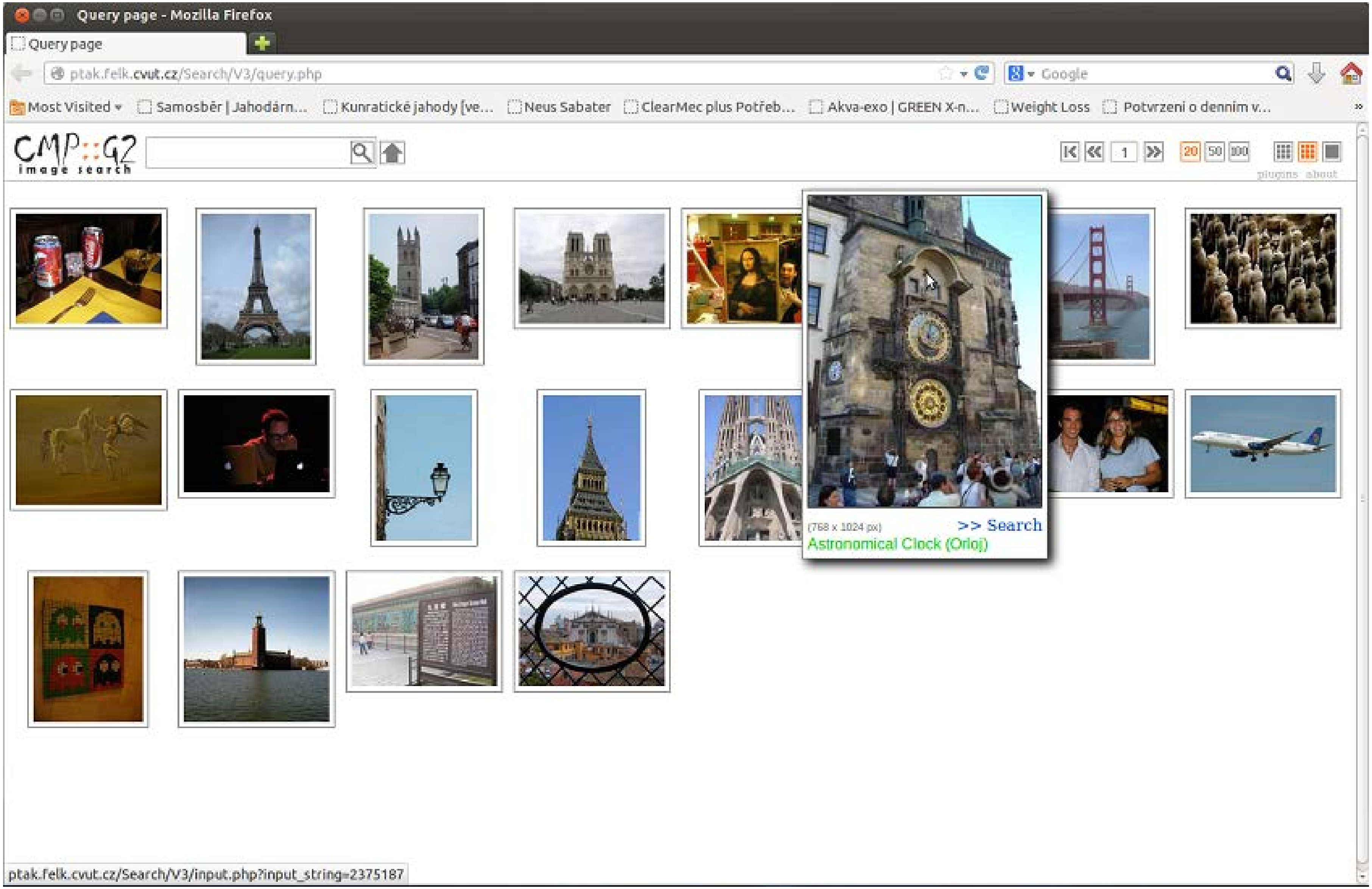


  - Detection of goods in tray at supermarket checkout

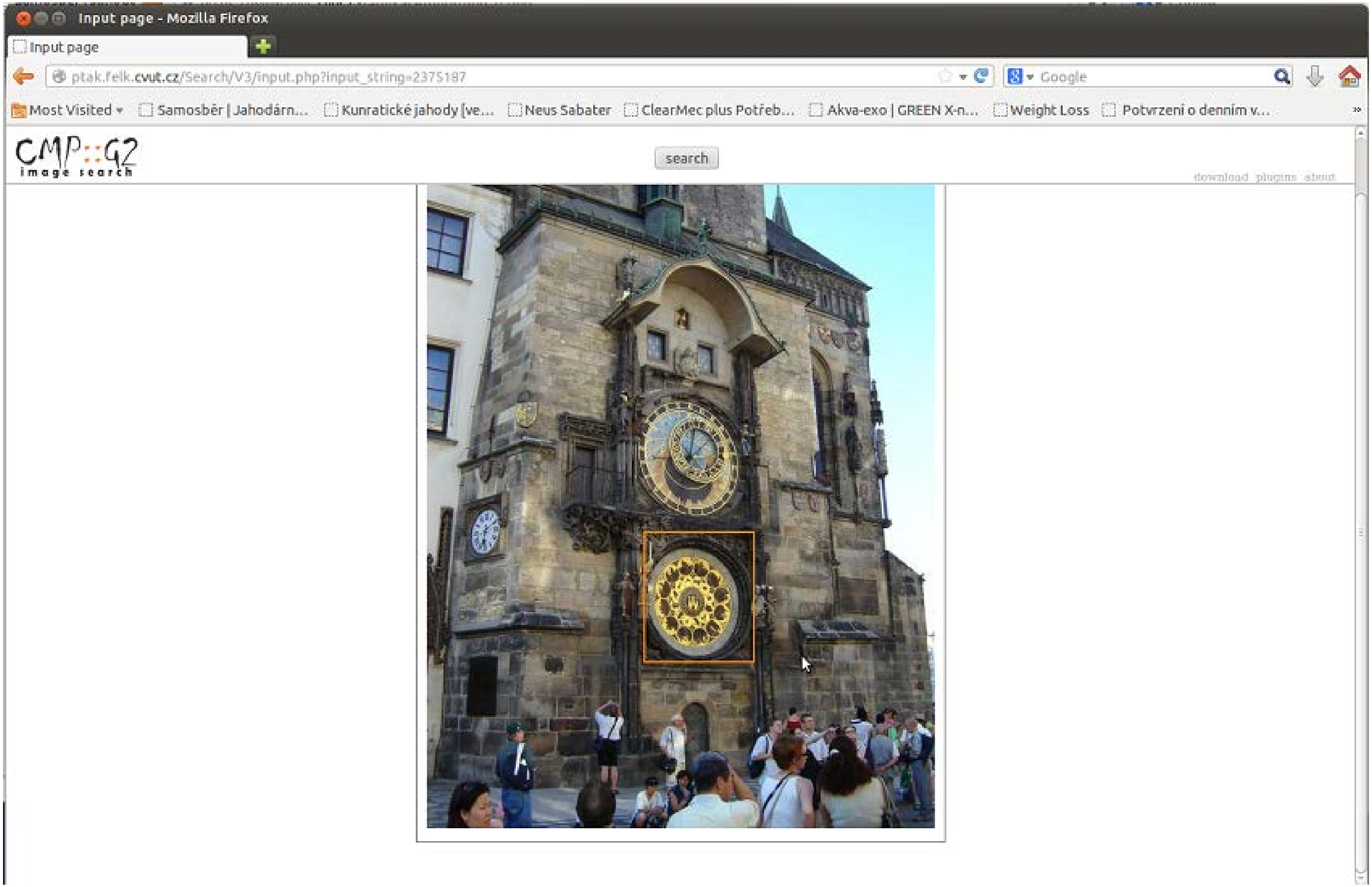
Local Features in Action (5): Image Retrieval



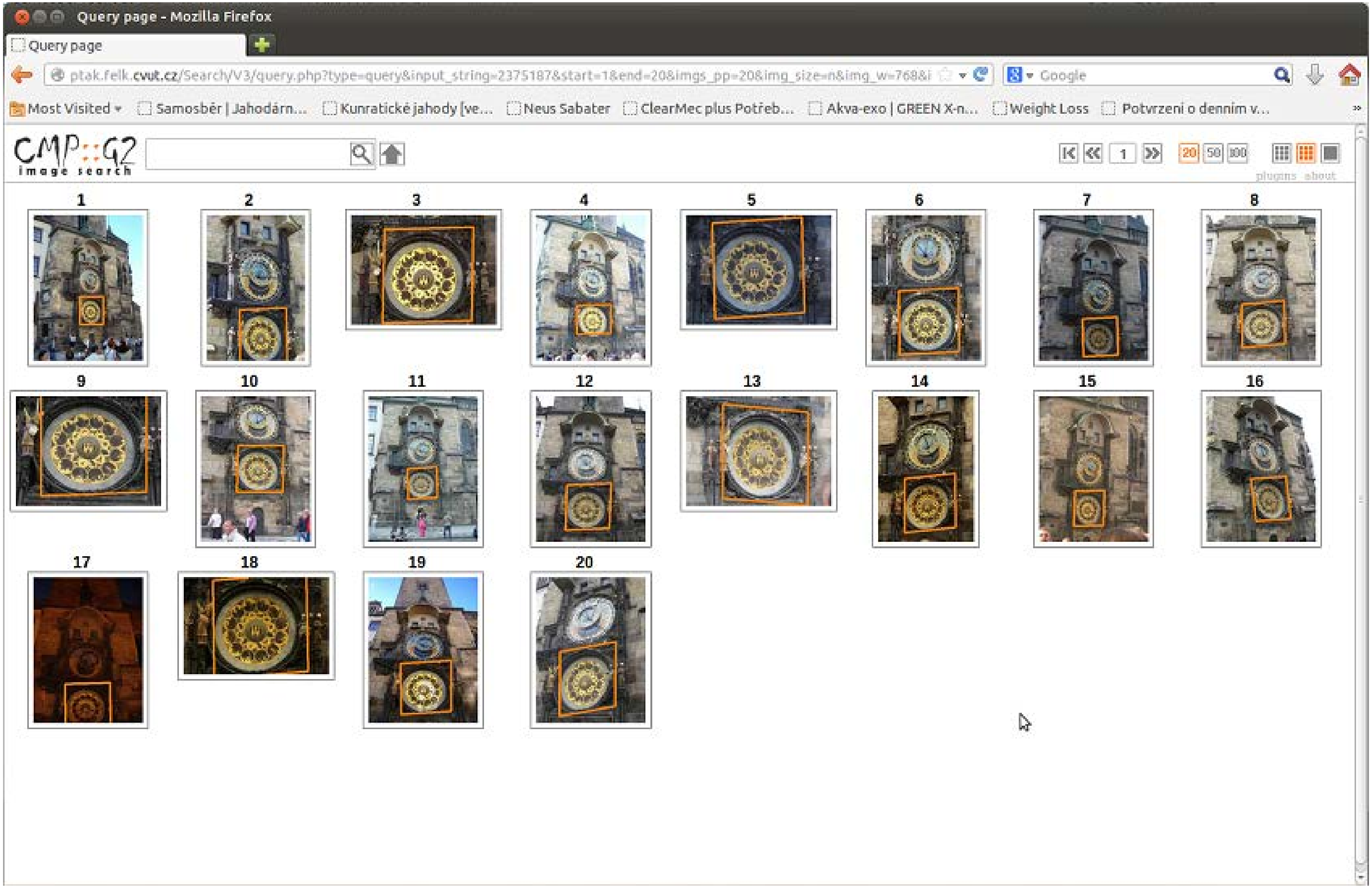
Local Features in Action (5): Image Retrieval



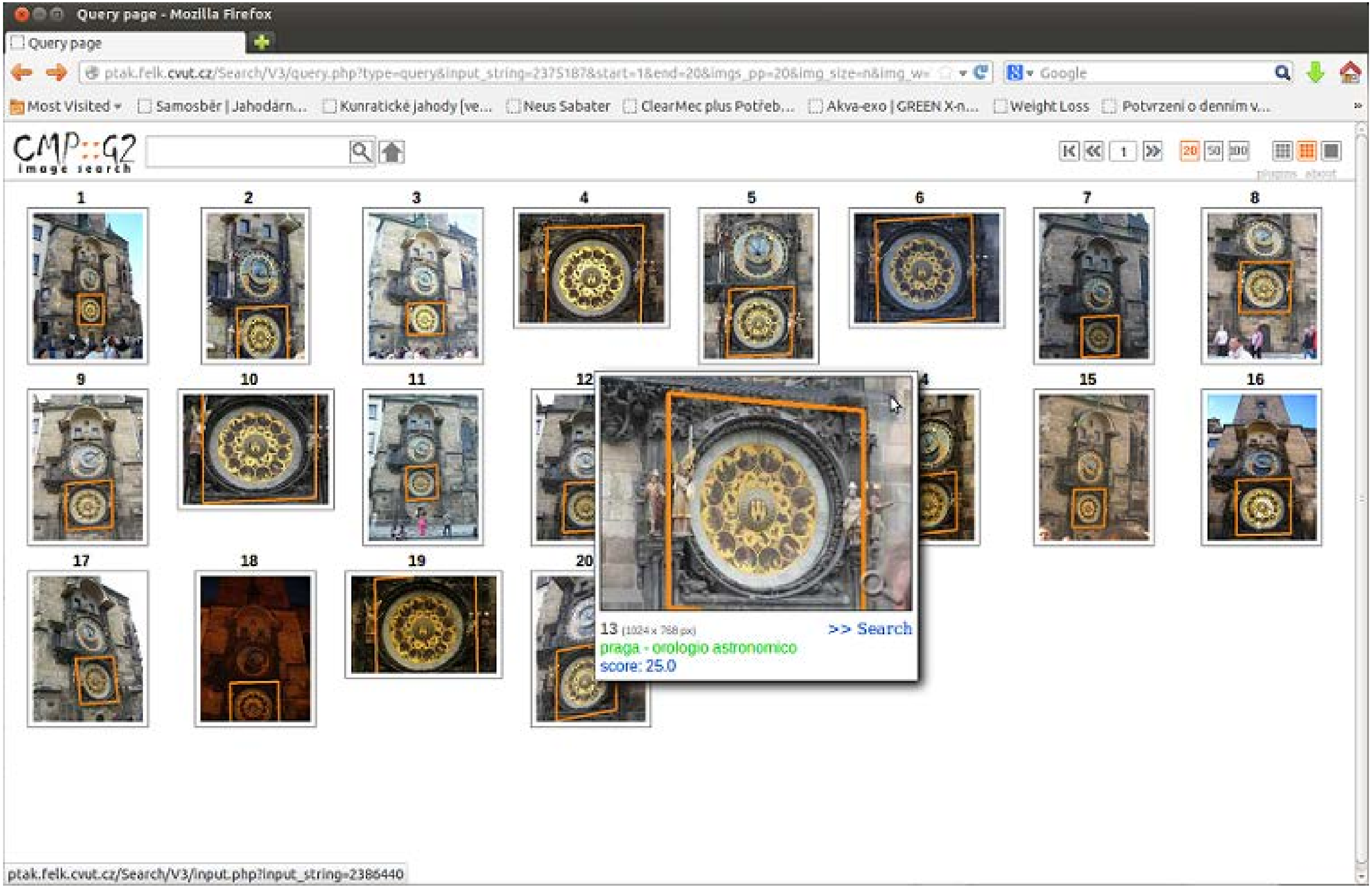
Local Features in Action (5): Image Retrieval



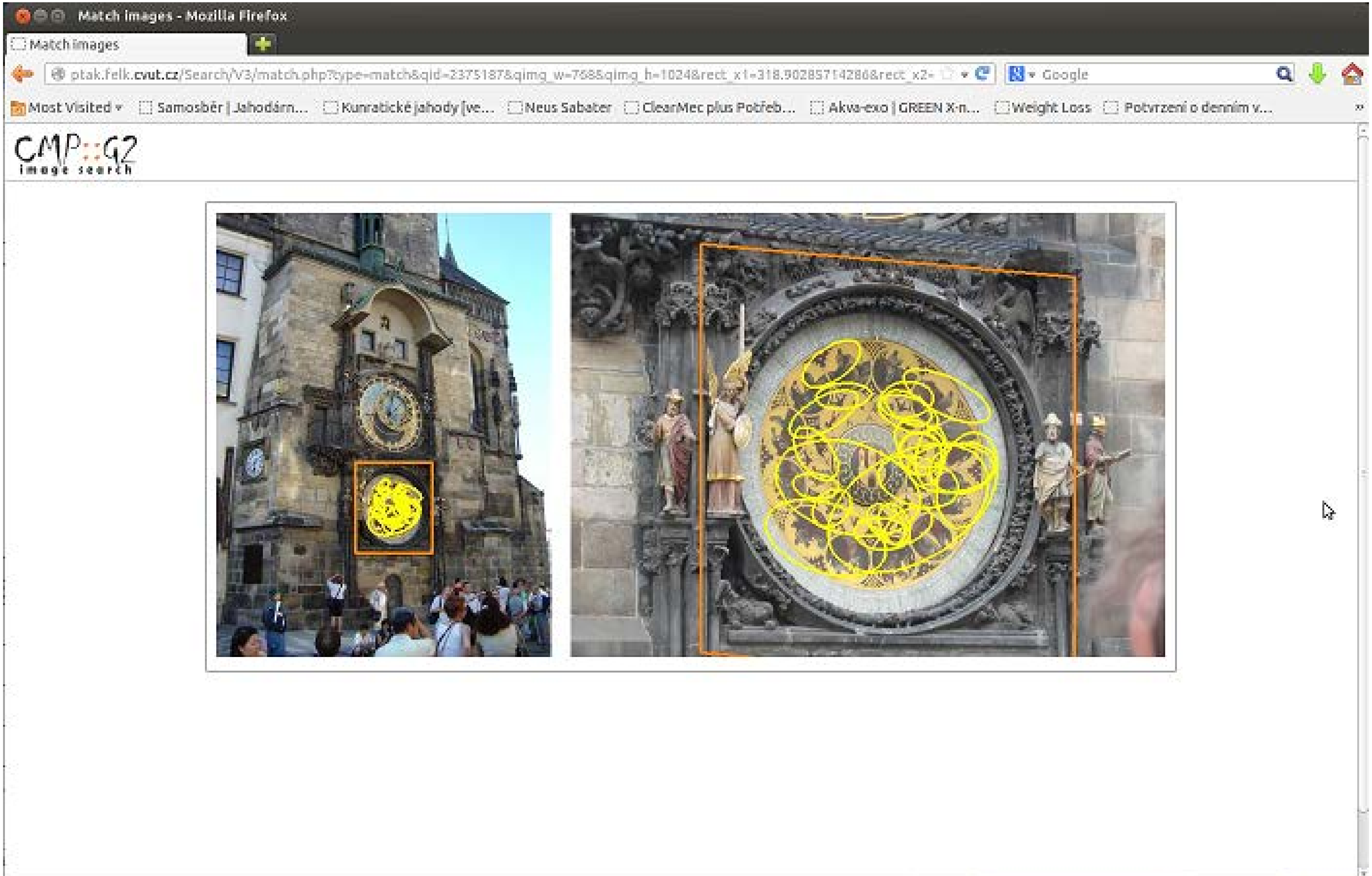
Local Features in Action (5): Image Retrieval



Local Features in Action (5): Image Retrieval



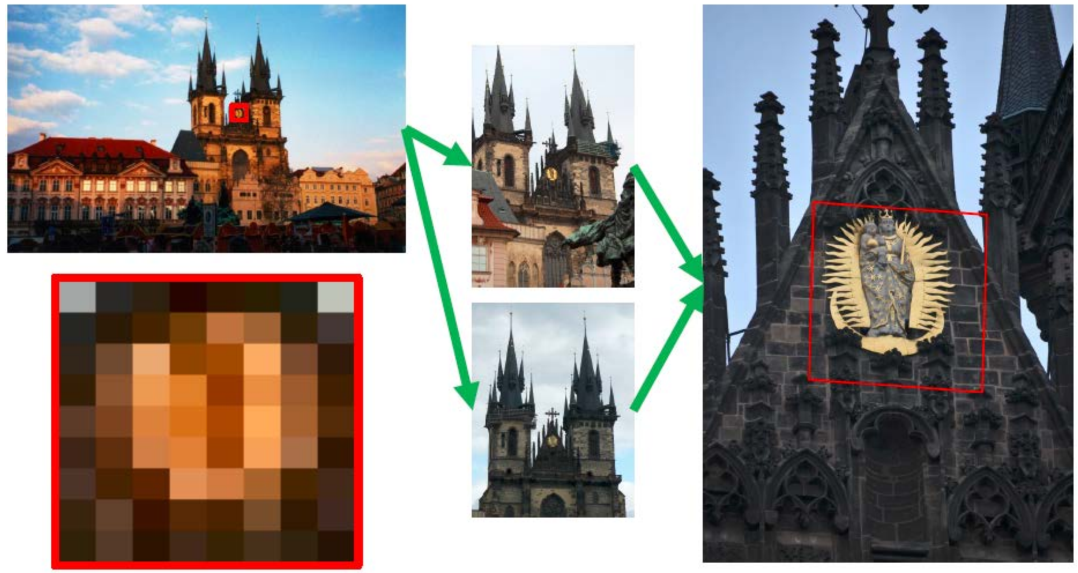
Local Features in Action (5): Image Retrieval



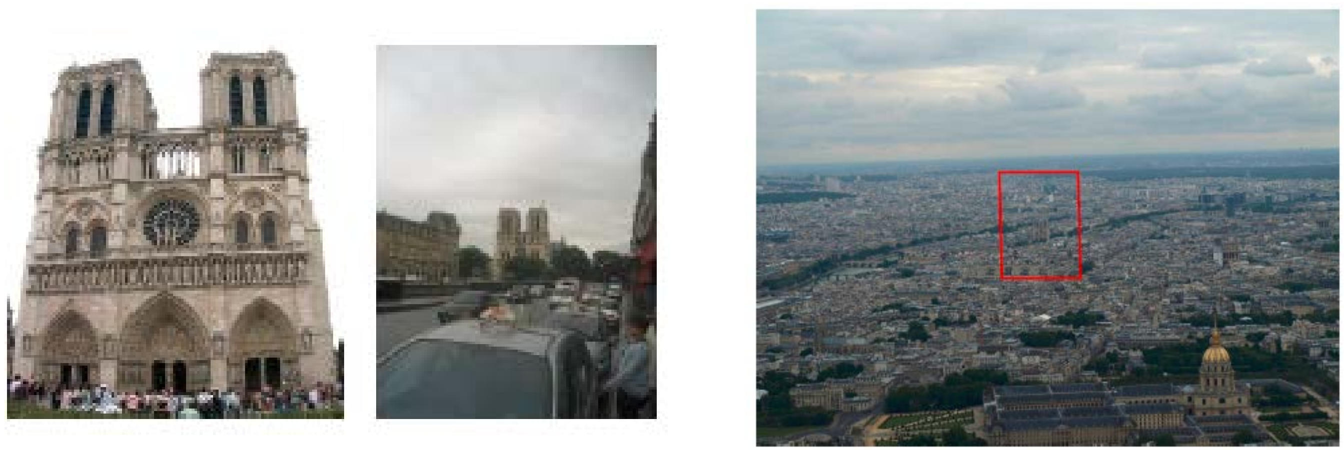
Local Features in Action (5): Image Retrieval

CENTER FOR MACHINE PERCEPTION

"Zoom in"

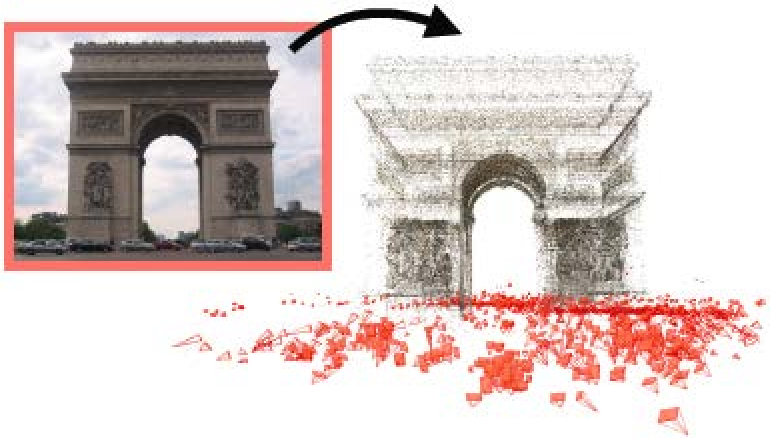


"Zoom out"

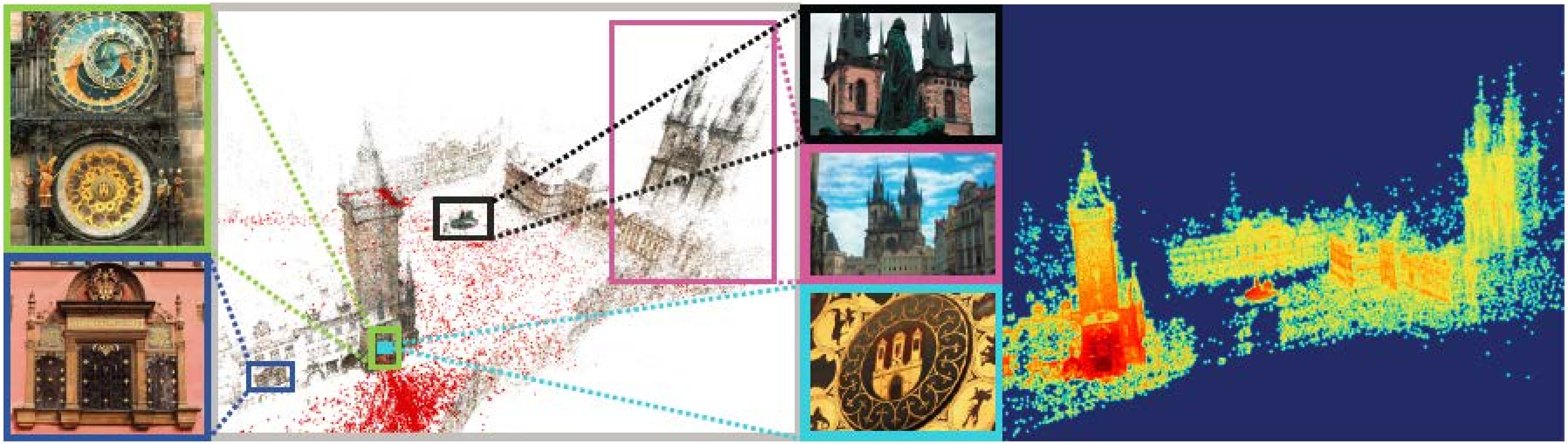


Schonberger J, Radenovic F, Chum O, Matas J. From Single Image Query to Detailed 3D Reconstruction. CVPR, 2015.

Local Features in Action (5): Image Retrieval



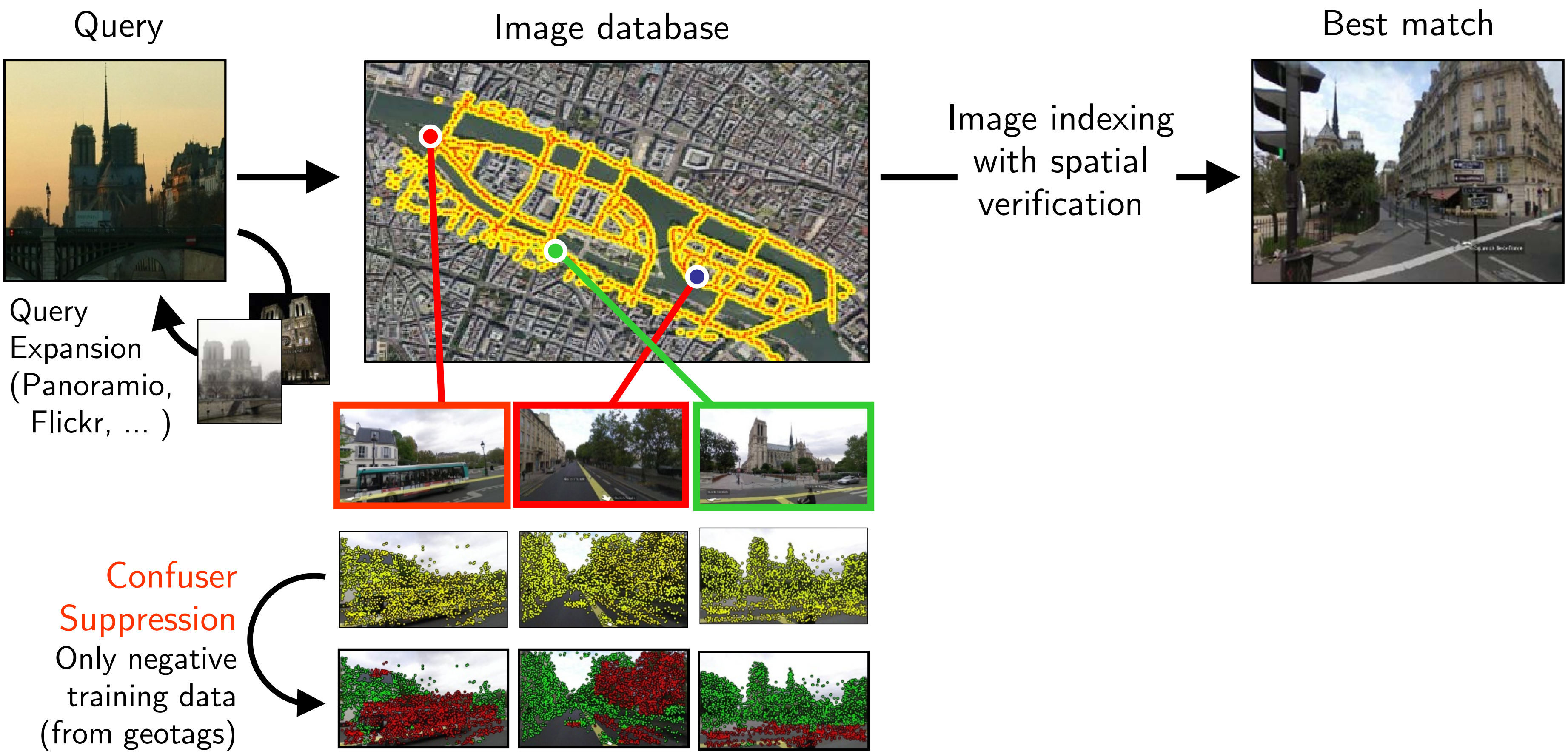
https://youtu.be/DIv1aGKqSIk



Schonberger J, Radenovic F, Chum O, Matas J. From Single Image Query to Detailed 3D Reconstruction. CVPR, 2015.

Local Features in Action (6): Localization and Mapping

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



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

Challenges in the Correspondence Problem

Why is Establishing Correspondence Difficult?

Finding correspondences is not easy

due to large viewpoint change (including scale) =>
the wide-baseline stereo problem

Applications:

- pose estimation
- 3D reconstruction
- location recognition

Finding correspondences is not easy

due to large viewpoint change (including scale)

=>

the wide-baseline (WBS) stereo problem

Finding correspondences is not easy

due to large

illumination change

=>

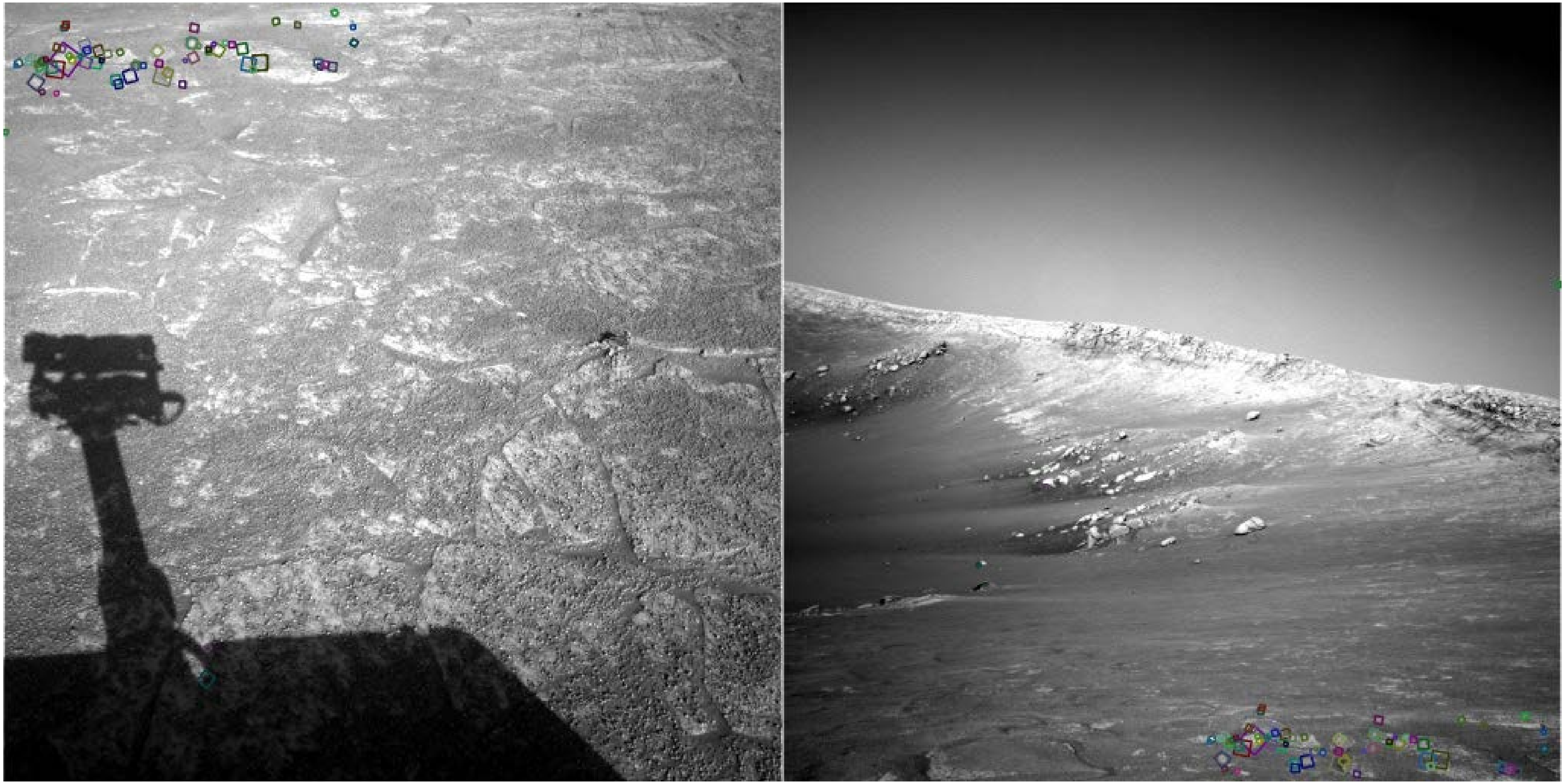
wide "illumination-baseline" stereo problem

Applications:

- location recognition
- summarization of image collections

CENTER FOR MACHINE PERCEPTION

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

due to occlusion

Applications:

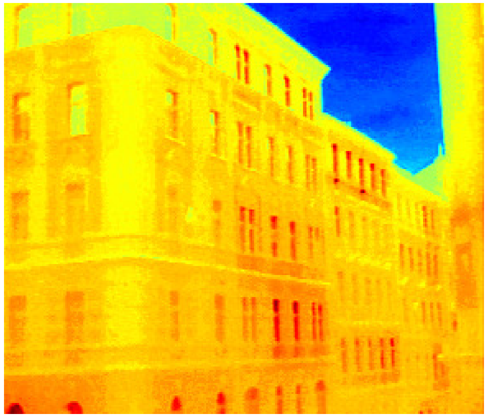
- pose estimation
- inpainting

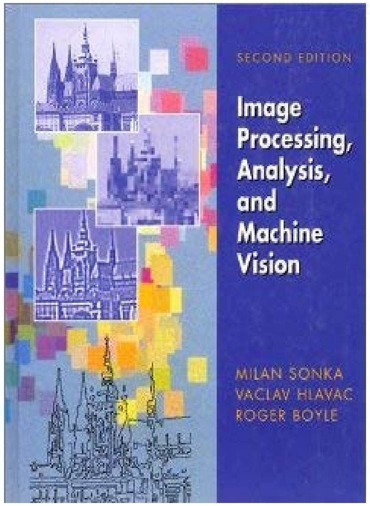
Finding Correspondences is not easy

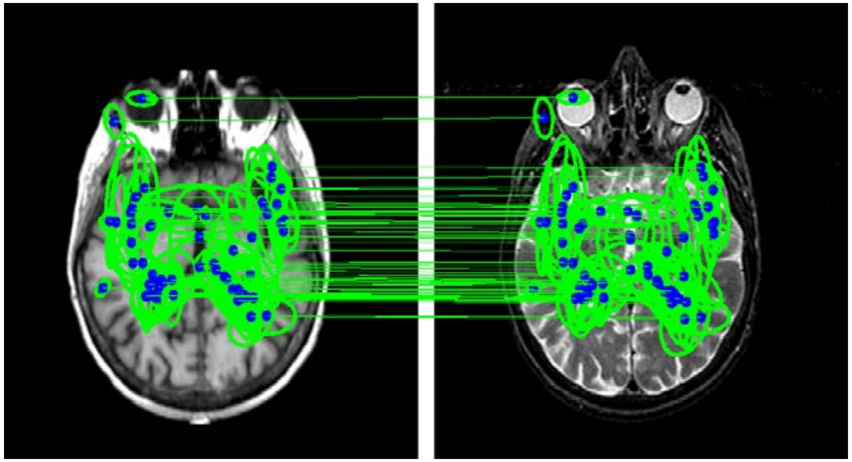
CENTER FOR MACHINE PERCEPTION

change of modality

Applications:
- medical imaging
- remote sensing







Detecting
Local Invariant Features

Design of Local Features

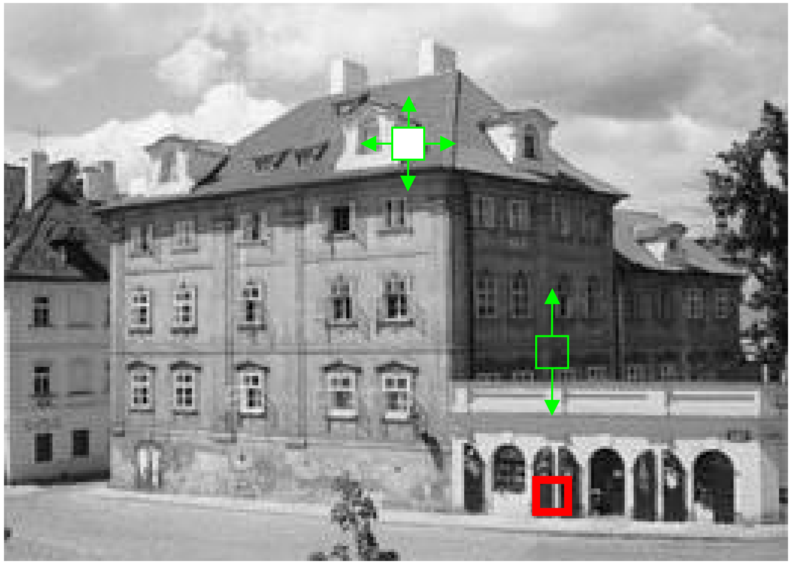
- "Local Features" are **regions,** i.e. in principle arbitrary sets of pixels (not necessarily contiguous) with
- High **repeatability,** (invariance in theory) under
  - Illumination changes
  - Changes of viewpoint => geometric transformations

  i.e. are **distinguishable** in an image regardless of viewpoint/illumination => are **distinguished regions**
- Are **robust to occlusion** => must be **local**
- Must have discriminative neighborhood => they are **"features"**

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

Harris detector (1988)

3500 citations



undistinguished patches:

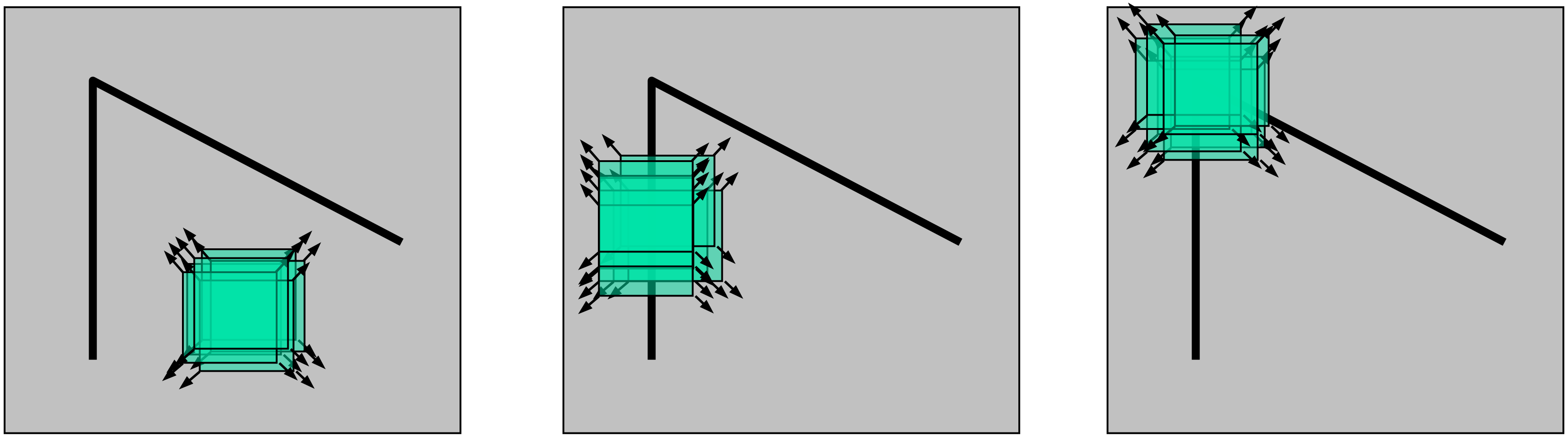
distinguished patches:

Two core ideas (in “modern terminology”):

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

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

Harris Detector: Basic Idea



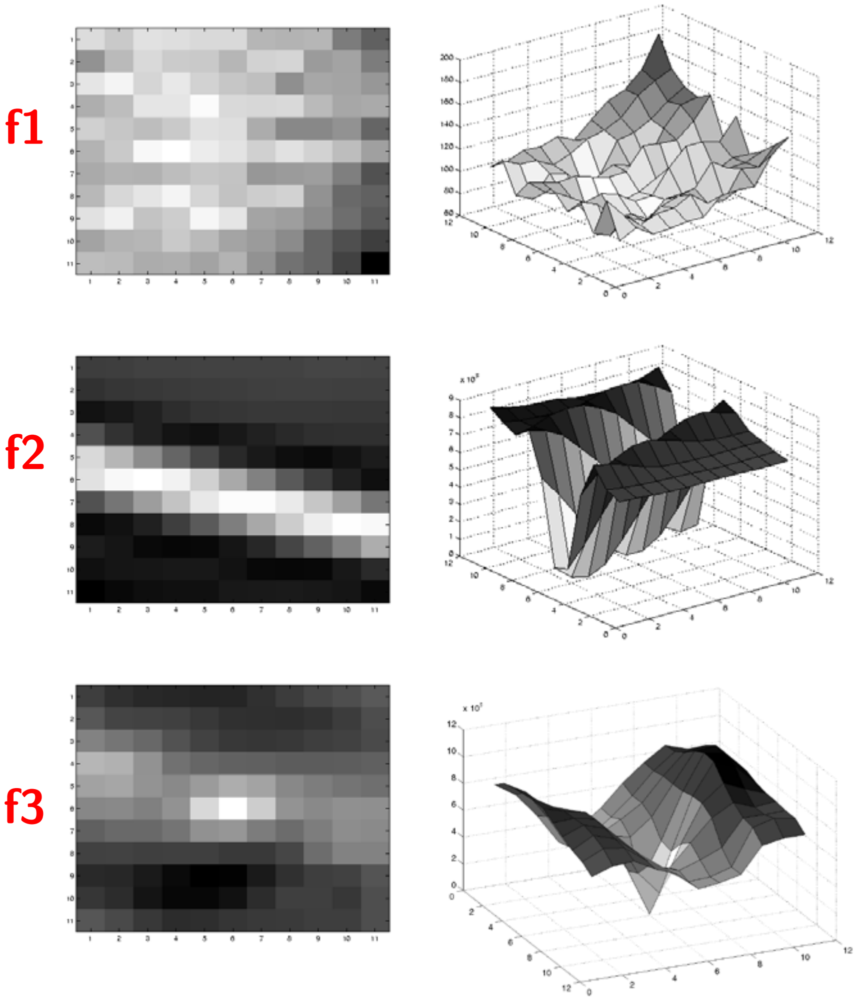
"flat" region:
no change in
all directions

"edge":
no change along
the edge
direction

"corner":
significant
change in all
directions

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

Harris Detector: Basic Idea



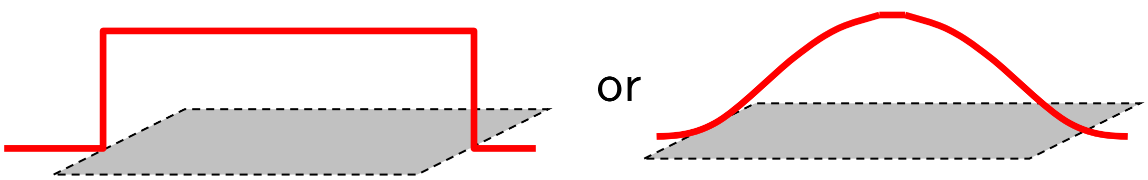
Harris Detector: Mathematics

Tests how similar is the image function $I(x_0, y_0)$ at point (x_0, y_0) to itself when shifted by (u, v):

- given by autocorrelation function

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

- $W(x_0, y_0)$ is a window centered at point (x_0, y_0)
- $w(x, y)$ can be constant or (better) Gaussian



1 in window, 0 outside

Gaussian

Harris Detector: Mathematics

Approximate intensity function in shifted position by the first-order Taylor expansion:

$$I(x+u, y+v) \approx I(x,y) + [I_x(x,y), I_y(x,y)] \begin{bmatrix} u \\ v \end{bmatrix}$$

where I_x, I_y are partial derivatives of $I(x,y)$.

$$\mathrm{E}(x_0, y_0; u, v) \approx \sum_{(x,y)\in W(x_0,y_0)} w(x,y) \left([I_x(x,y), I_y(x,y)] \begin{bmatrix} u \\ v \end{bmatrix}\right)^2$$

$$= [u, v] \sum\nolimits_W w(x,y) \begin{bmatrix} I_x(x_0,y_0)^2 & I_x(x_0,y_0) I_y(x_0,y_0) \\ I_x(x_0,y_0) I_y(x_0,y_0) & I_y(x_0,y_0)^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

Harris Detector: Mathematics

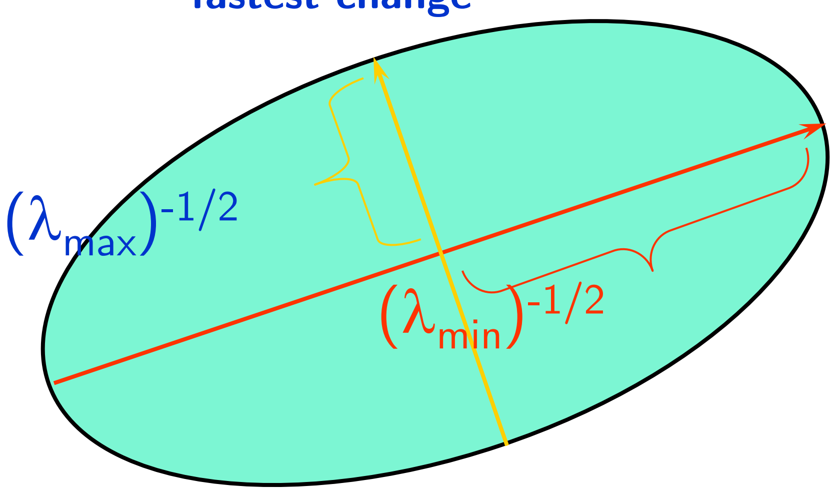
$$E(x_0, y_0; u, v) \approx [u, v] M(x_0, y_0) \begin{bmatrix} u \\ v \end{bmatrix}$$

Intensity change in shifting window: eigenvalue analysis of M

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

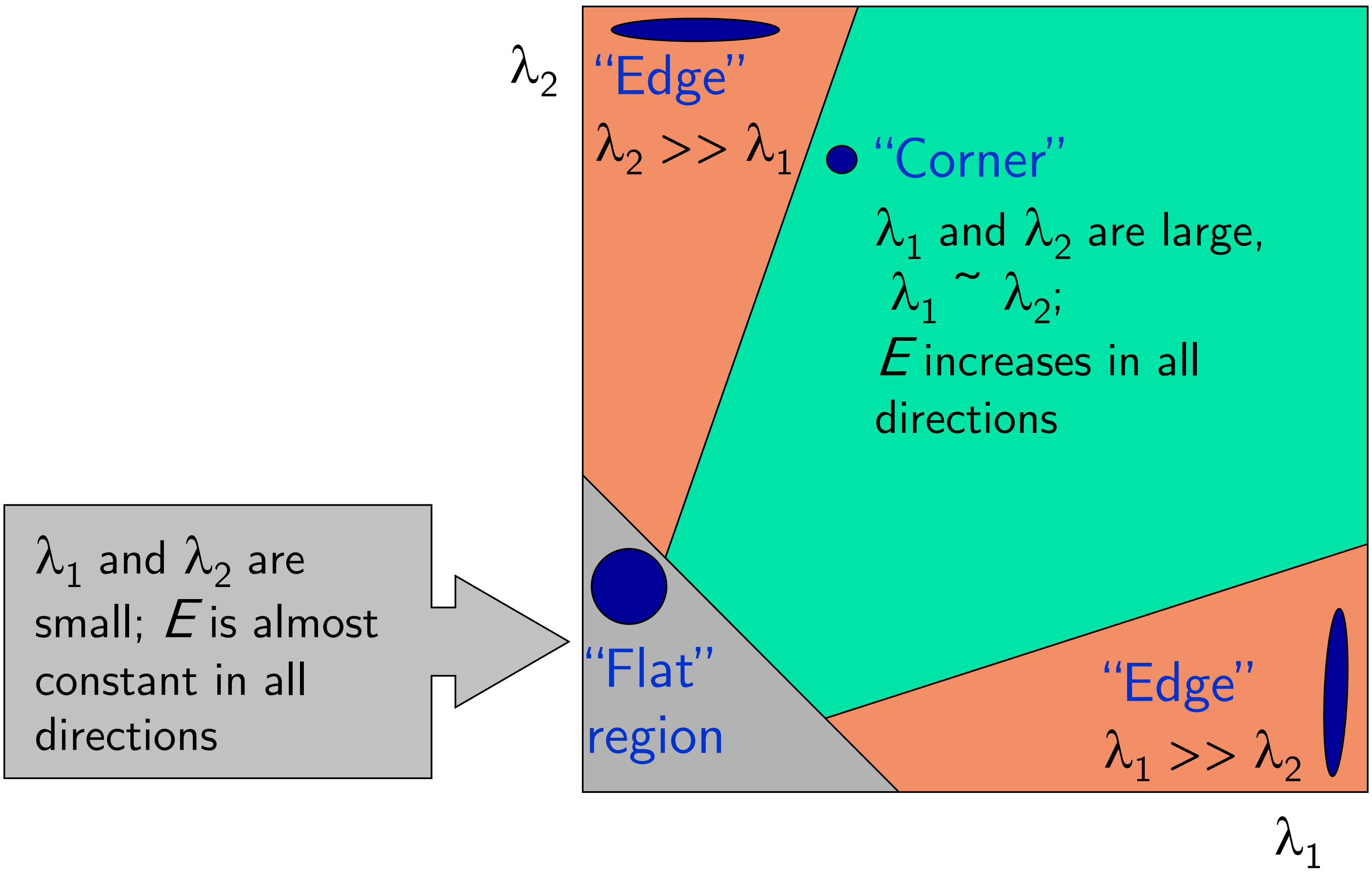
Ellipse:

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



Harris Detector: Mathematics

Classification of image points using eigenvalues of M:



Harris Detector: Mathematics

Measure of corner response ("cornerness"):

$$R = \det M - k(\operatorname{trace} M)$$

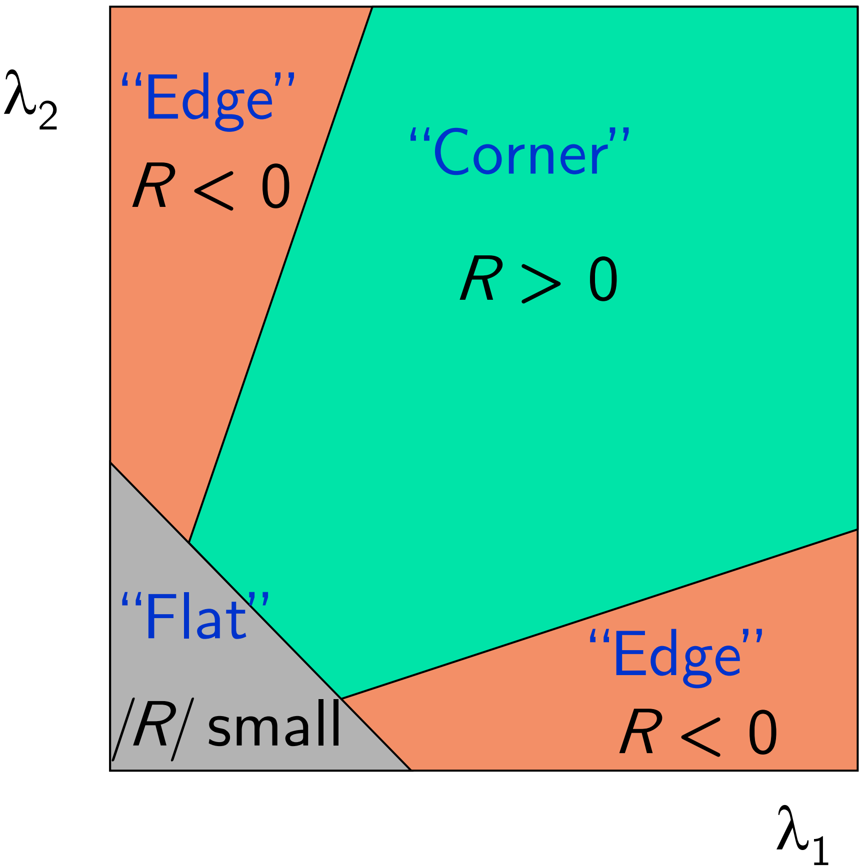
- $M = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$
- $\det M = \lambda_1 \lambda_2 = AC - B^2$
- $\operatorname{trace} M = \lambda_1 + \lambda_2 = A + C$
- k ... empirical constant, $k \in (0.04, 0.06)$

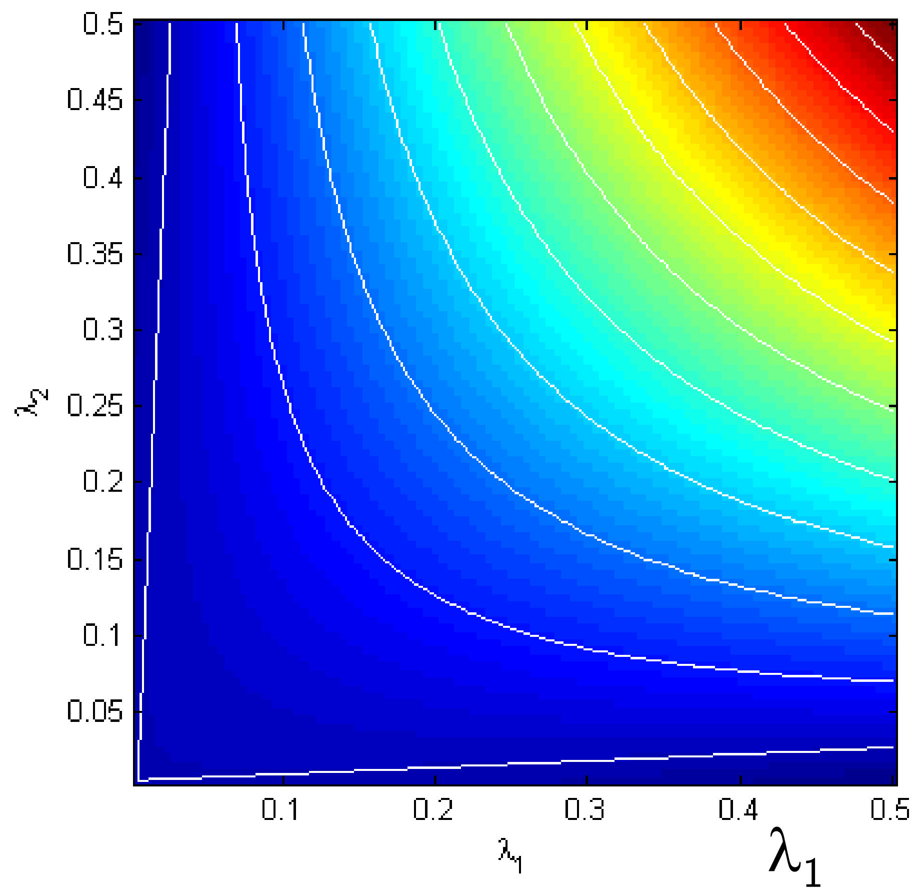
Find corner points as **local maxima** of corner response R:

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

Harris Detector: Mathematics

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





Harris Detector

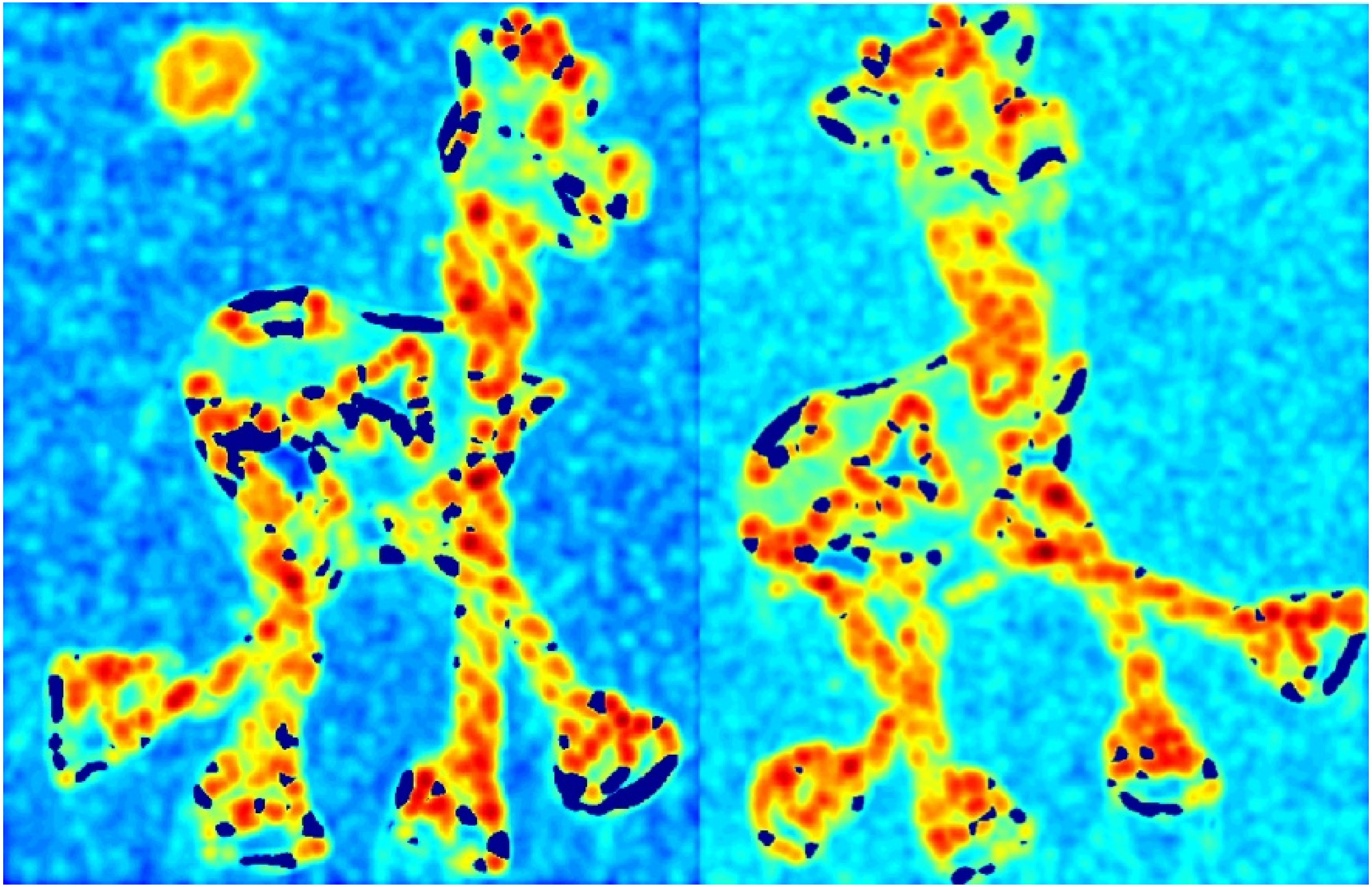
- The Algorithm:
  - Compute partial derivatives I_x, I_y
  - Compute: $A = \sum_W {I_x}^2$, $B = \sum_W I_x I_y$, $C = \sum_W {I_y}^2$
  - Compute corner response R
  - Find local maxima in R

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

Harris Detector: Workflow

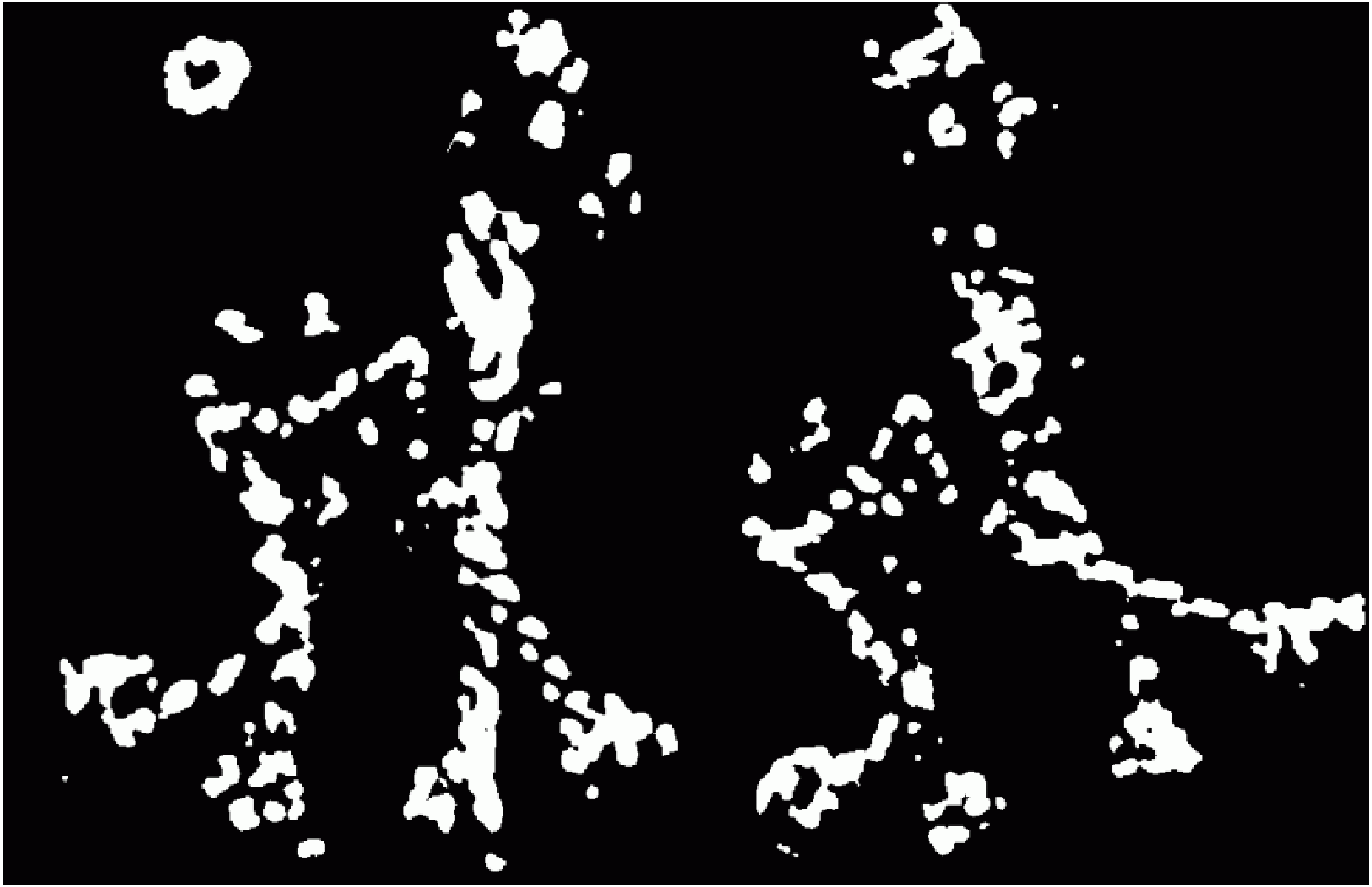
Harris Detector: Workflow

Compute corner response R



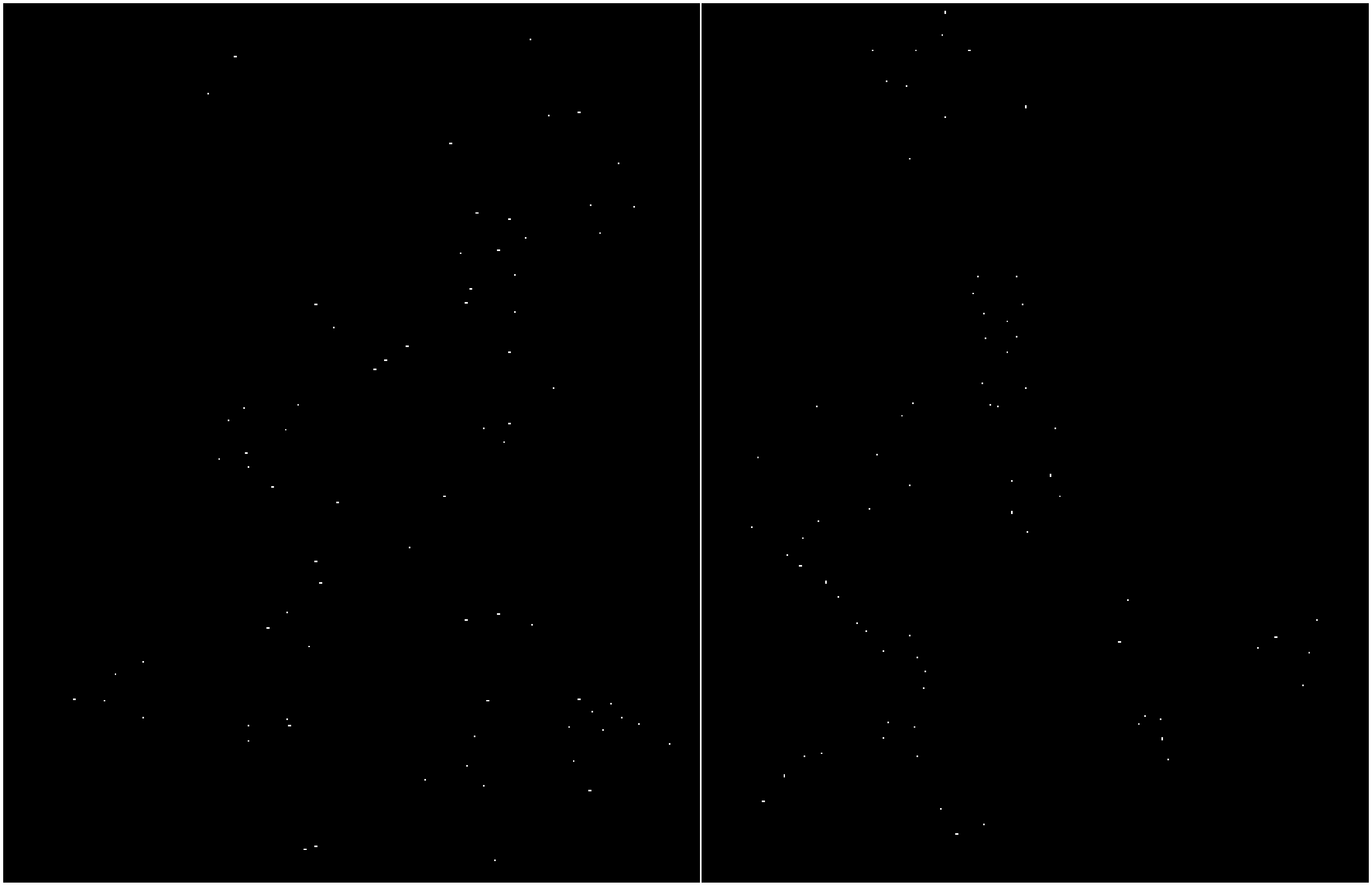
Harris Detector: Workflow

Find points with large corner response: $R>$threshold



Harris Detector: Workflow

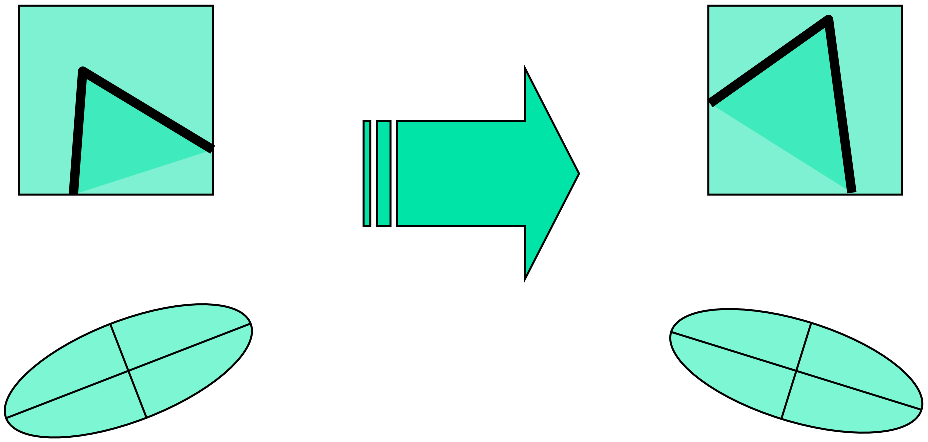
Take only the points of local maxima of R



Harris Detector: Workflow

Harris Detector: Properties

- Rotation invariance



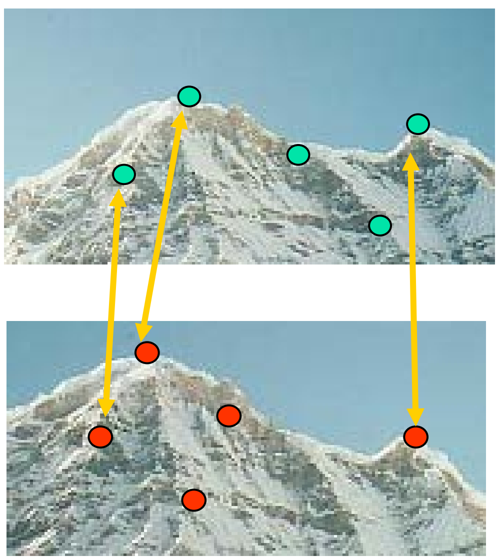
Ellipse rotates but its shape (i.e. eigenvalues) remains the same

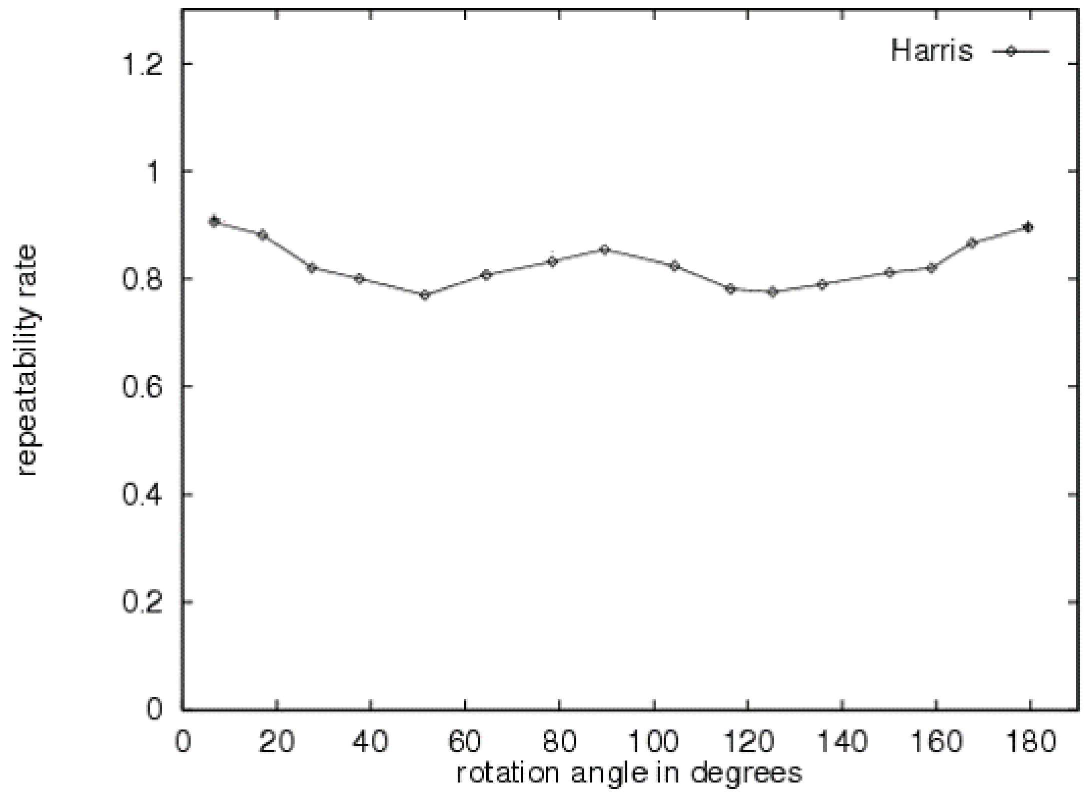
Corner response R is invariant to image rotation

Rotation Invariance of Harris Detector

Repeatable rate:

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





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

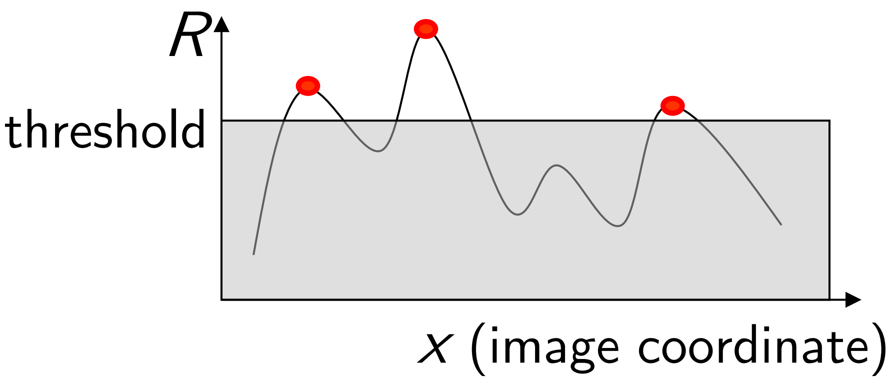
Harris Detector: Intensity change

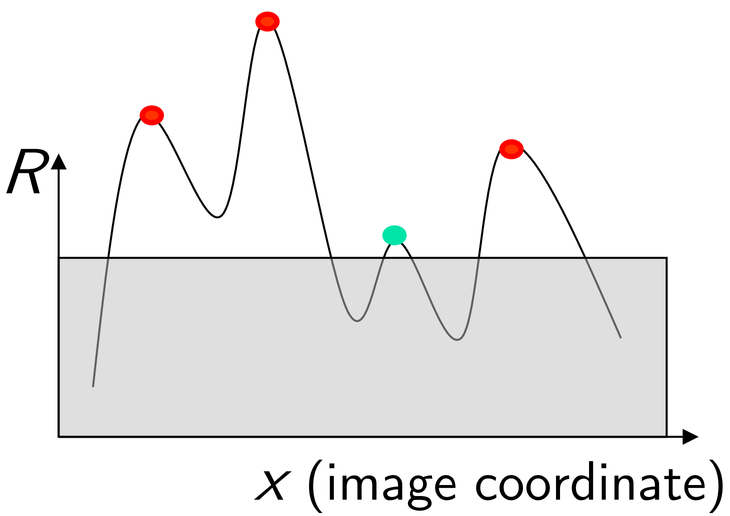
- Partial invariance to additive and multiplicative intensity changes

  ✓ Only derivatives are used =>

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

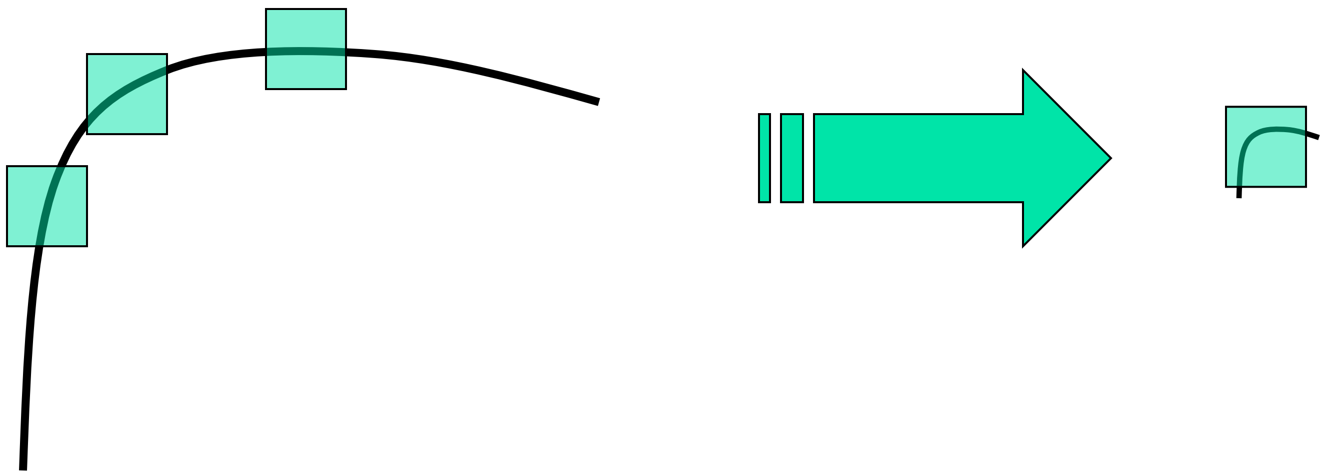
  ? Intensity scale: $I \rightarrow a\ I$





Harris Detector: Scale Change

- Not invariant to *image scale*!



All points will be classified as edges

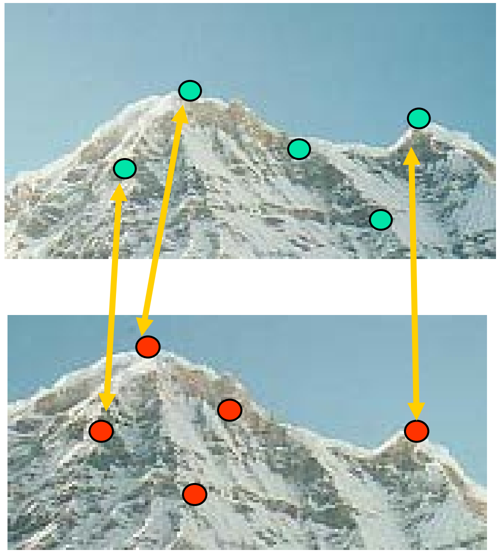
Corner !

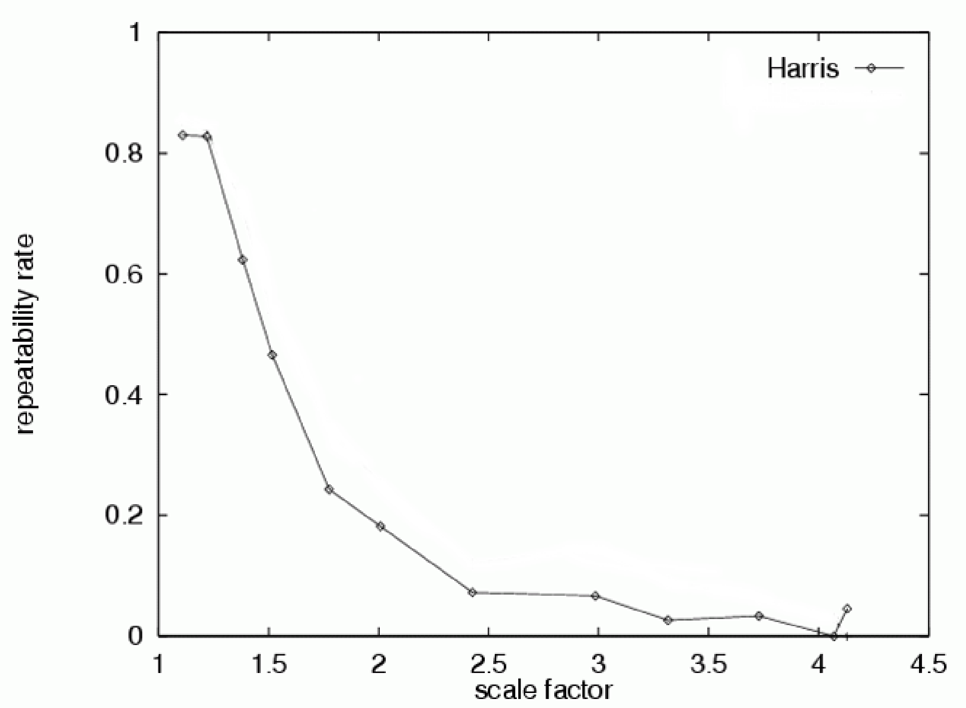
Harris Detector: Scale Change

- Quality of Harris detector for different scale changes

Repeatability rate:

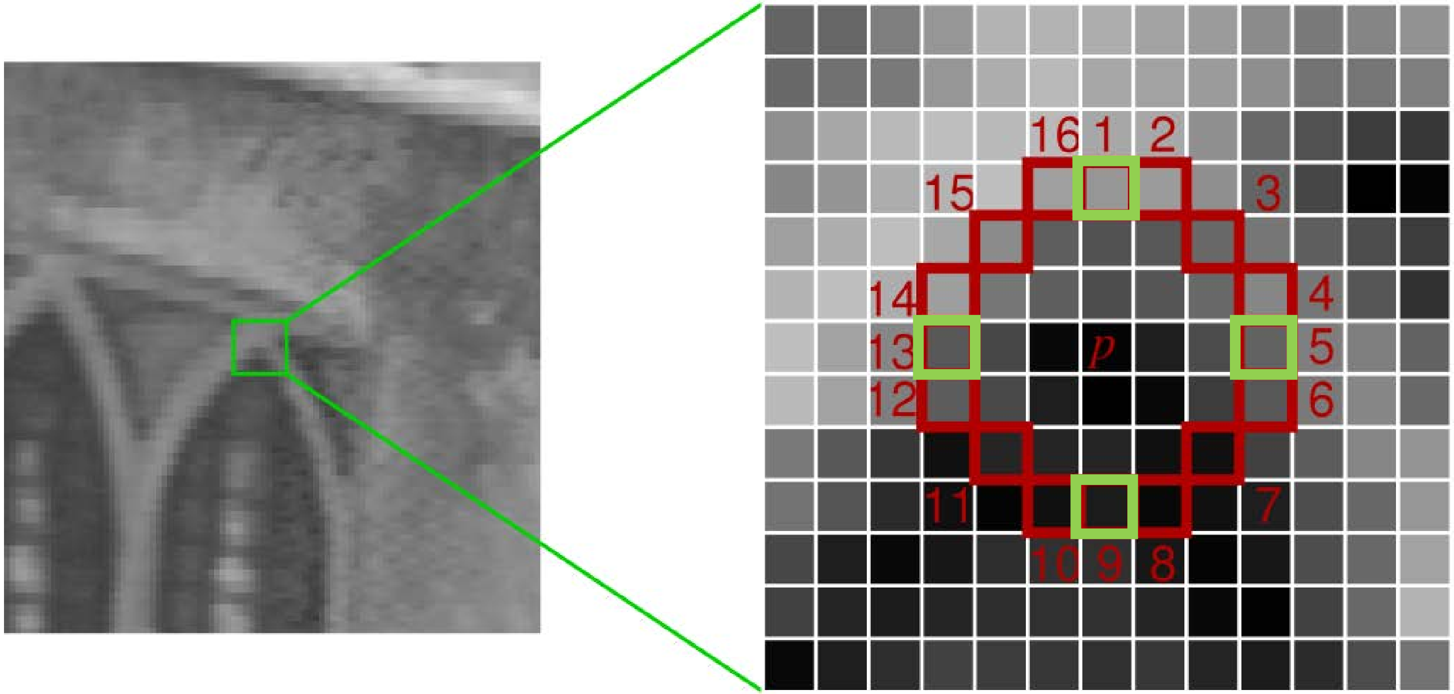
$$\frac{\#\ \text{correspondences}}{\#\ \text{possible correspondences}}$$





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

FAST Feature Detector



- Considers a circle of 16 pixels around the corner candidate *p*
- ≥ 12 contiguous pixels brighter/darker than $I_p \pm t, t \ldots$ 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, 2006

FAST: Weaknesses

- Corners are clustered together:
  - 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 $S_b = \{q | I_q \geq I_p + t\}, S_d = \{q | I_q \leq I_p - t\}$

- High speed test does not generalize well for $n < 12$
- Choice of high speed test is not optimal
- Knowledge from the first 4 tests is discarded
- Multiple features are detected adjacent to one another

Slide credit: E. Rosten, 2006

FAST: running times

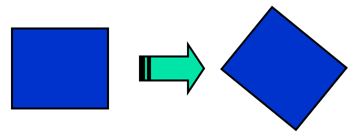
Detector	Opteron 2.6GHz		Pentium III 850MHz	
	ms	%	ms	%
Fast $n = 9$ (non-max suppression)	1.33	6.65	5.29	26.5
Fast $n = 9$ (raw)	1.08	5.40	4.34	21.7
Fast $n = 12$ (non-max suppression)	1.34	6.70	4.60	23.0
Fast $n = 12$ (raw)	1.17	5.85	4.31	21.5
Original FAST $n = 12$ (non-max suppression)	1.59	7.95	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.

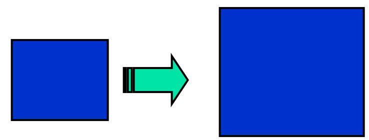
Slide credit: E. Rosten, 2006

Models of Image Change

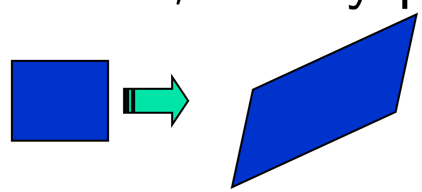
- Geometry
  - Rotation



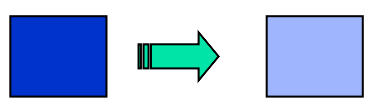
  - Similarity (rotation + uniform scale)



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

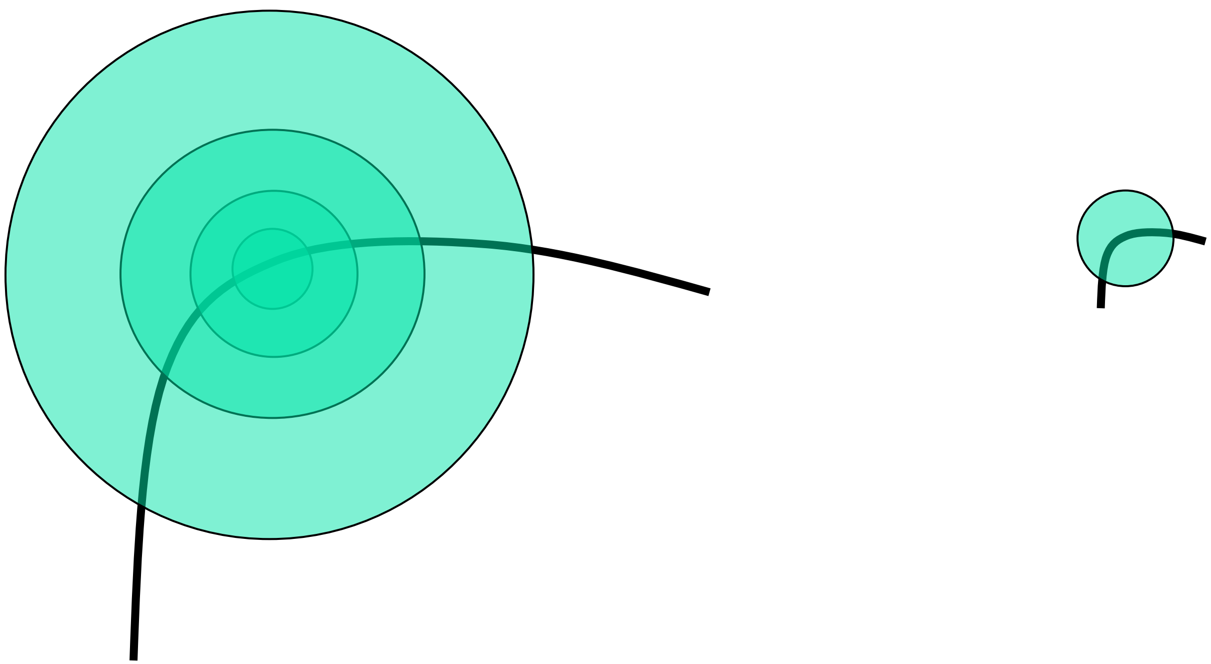


- Photometry
  - Affine intensity change ($I \rightarrow a\, I + b$)



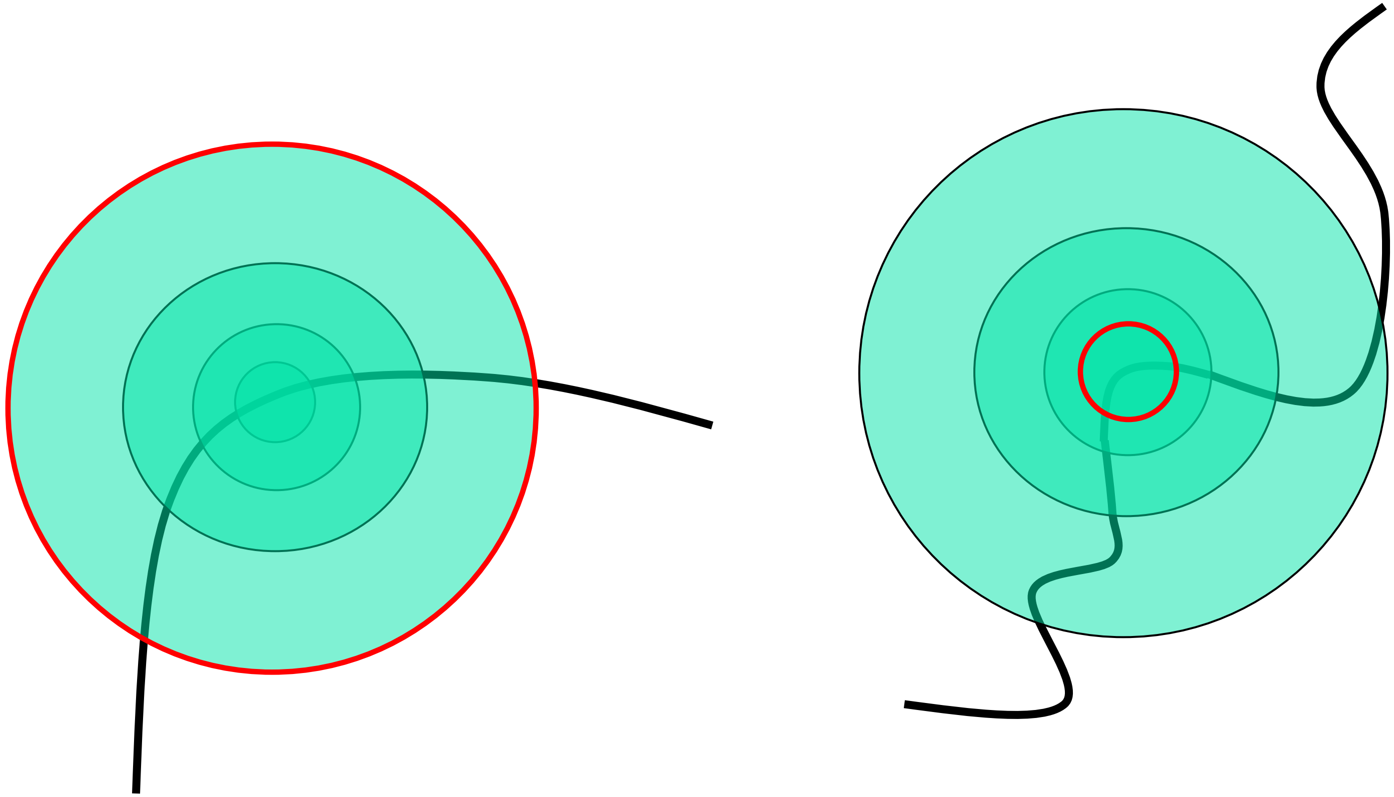
Scale Invariant Detection

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



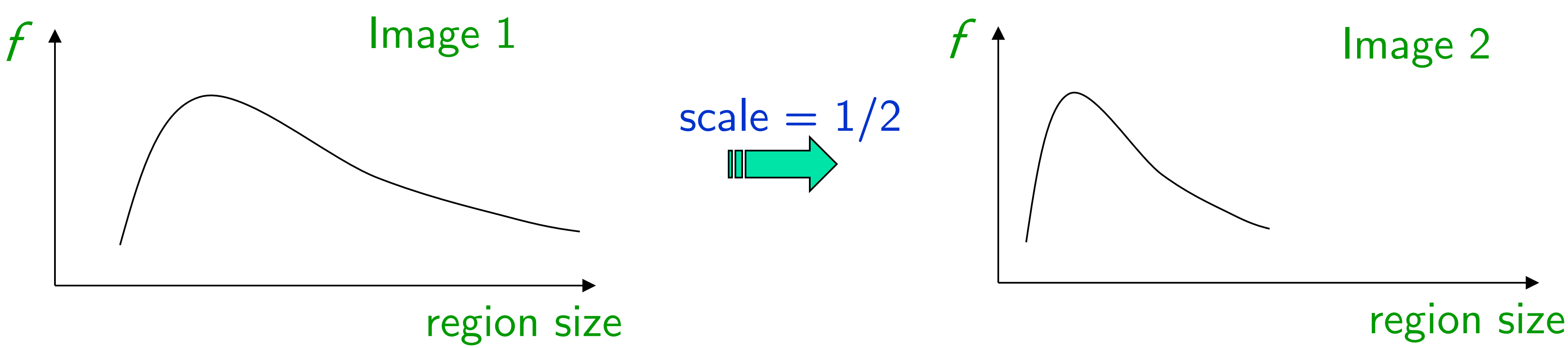
Scale Invariant Detection

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



Scale Invariant Detection

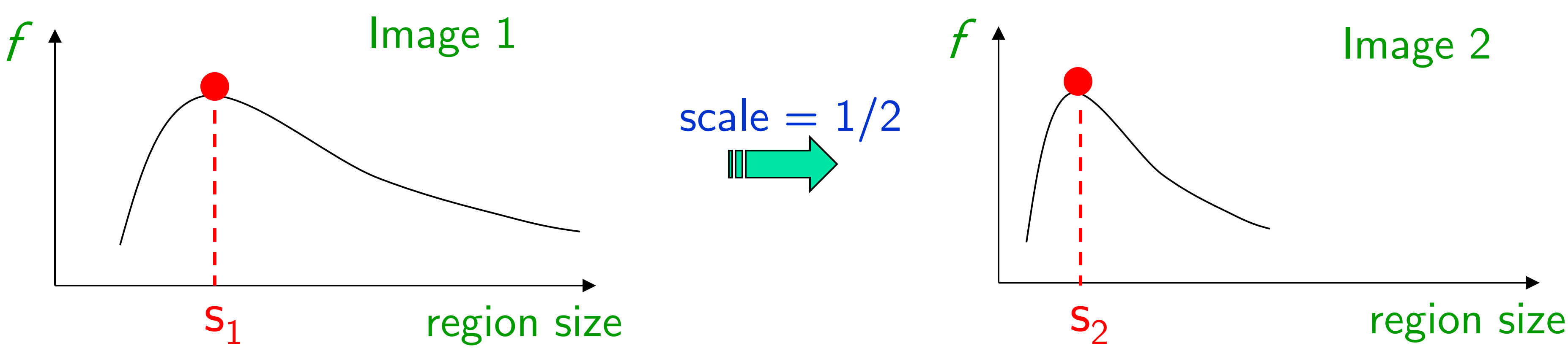
- Solution:
  - Design a function on the region (circle), which is "scale covariant" (the same for corresponding regions, even if they are at different scales)
    - For a point in one image, we can consider it as a function of region size (circle radius)



Scale Invariant Detection

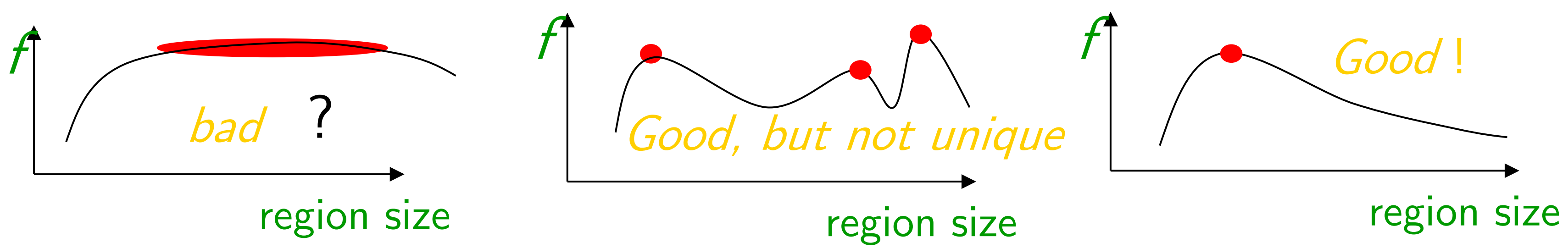
- Common approach:
  - Take a local maximum of some function
  - *Observation*: region size, for which the maximum is achieved, should be *invariant* to image scale.

Important: this scale invariant region size is found in each image independently!



Scale Invariant Detection

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



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

Scale Invariant Detection

- Functions for determining scale

$$f = \text{Kernel} * \text{Image}$$

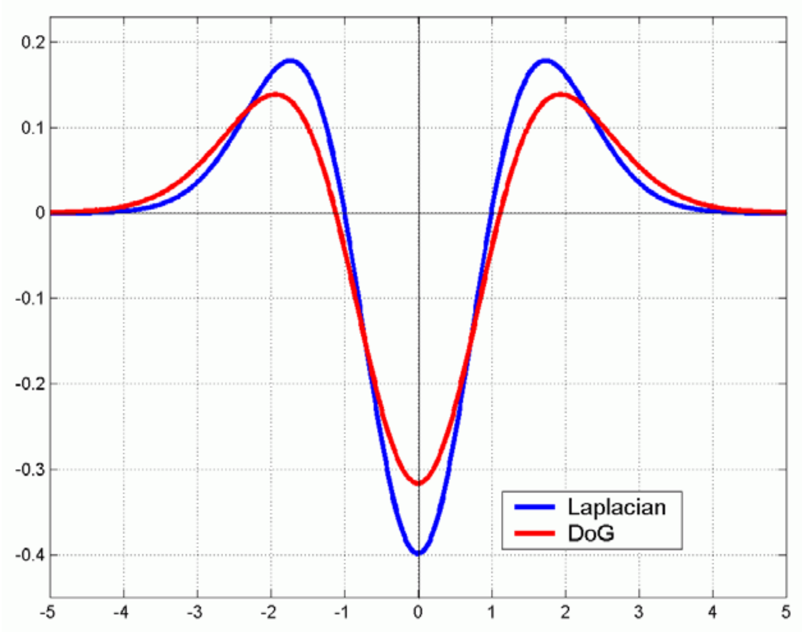
Kernels:

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

(Laplacian)

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

(Difference of Gaussians)

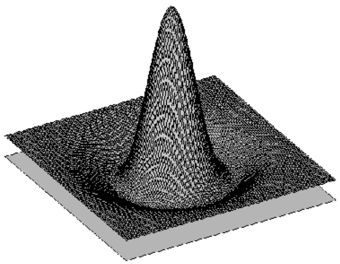


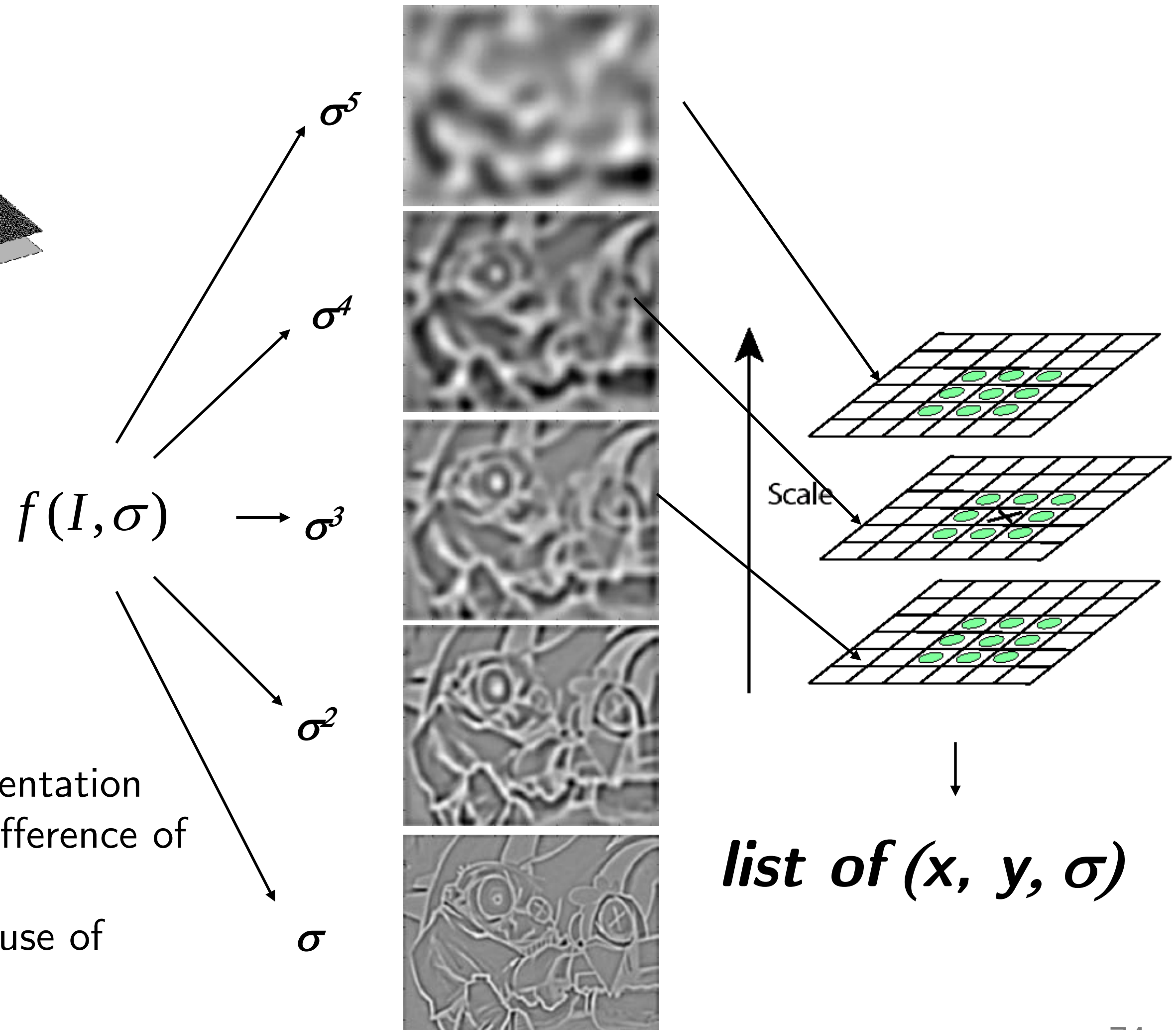
where Gaussian

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

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?

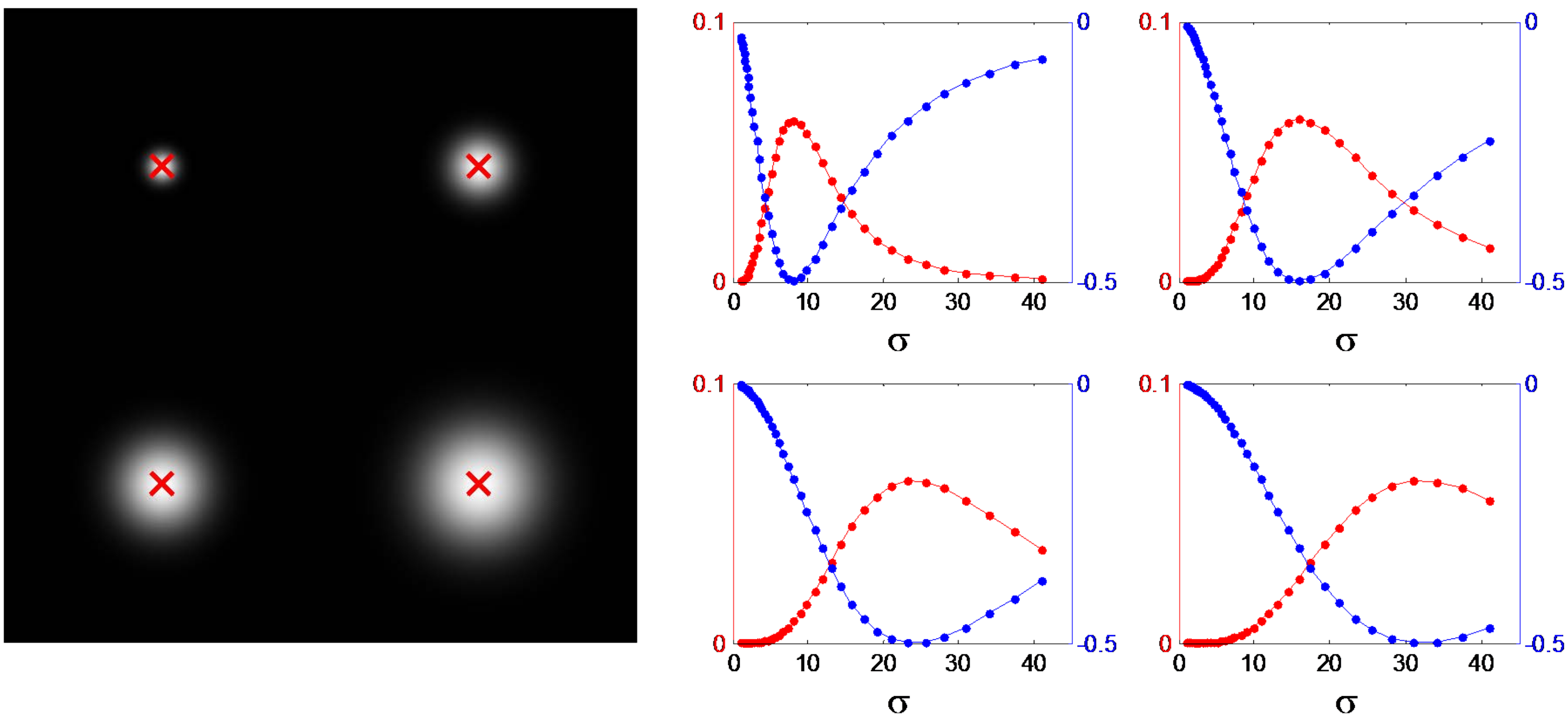
Scalespace pyramid, GPU implementation

Approximation of Laplacian by Difference of Gaussians

Approximation by box filters and use of integral images

Automatic Scale Selection

- Gaussian scalespace, “stack of gradually smoothed versions” of original image
- 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

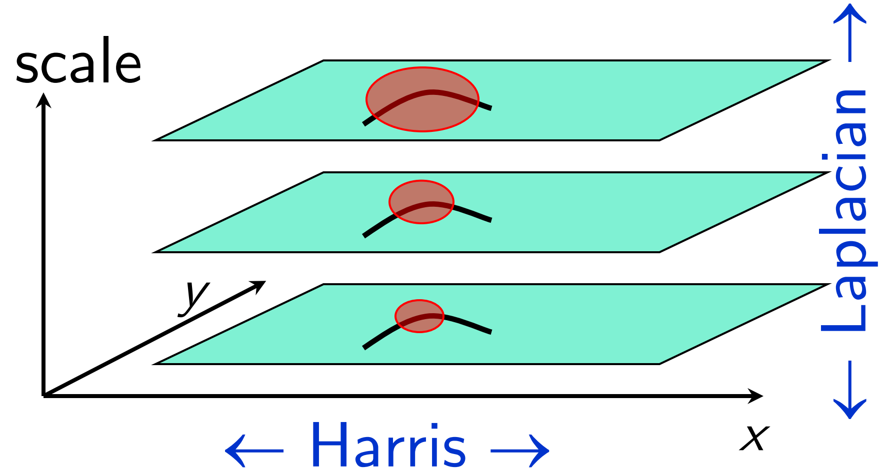


Scale Invariant Detectors

Harris-Laplacian[1]

Find local maximum of:

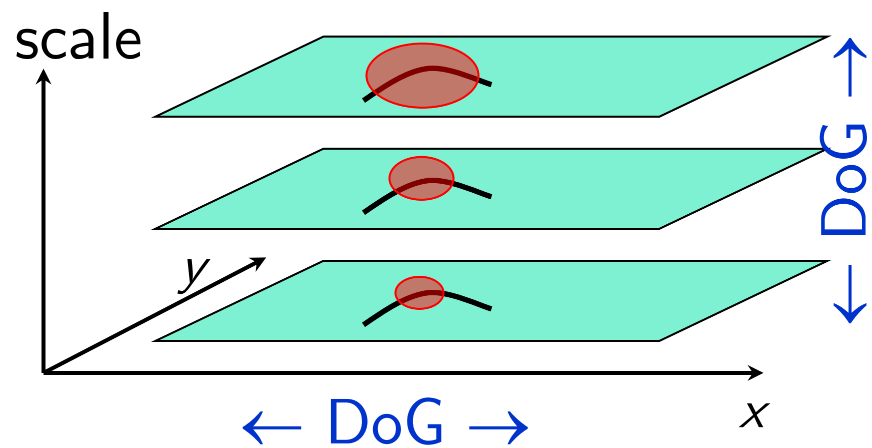
- Harris corner detector in space (image coordinates)
- Laplacian in scale



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

Find local maximum of:

- Difference of Gaussians in space and scale



Other options: Hessian, ...

Harris does not work well for scale selection

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

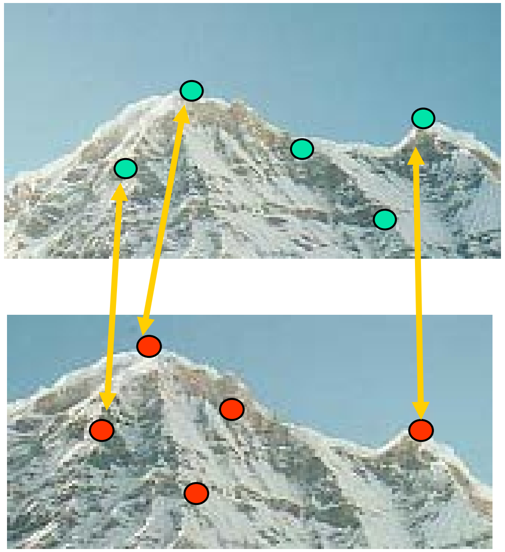
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

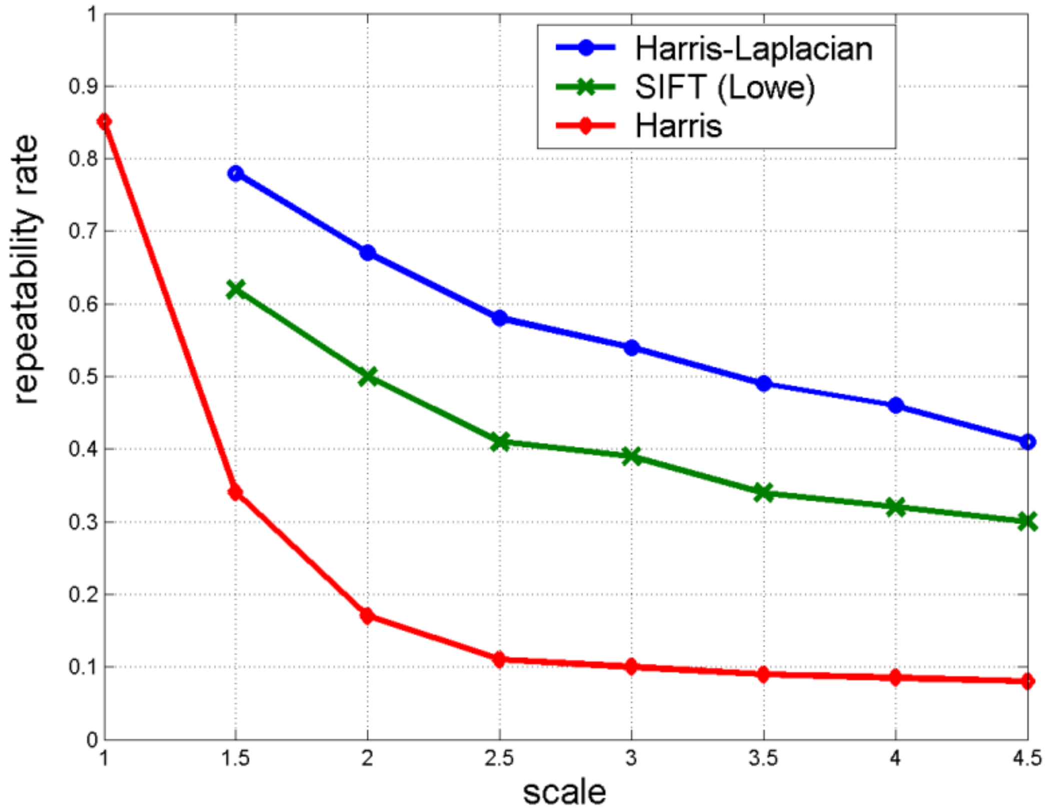
Scale Invariant Detectors

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

Repeatability rate:

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

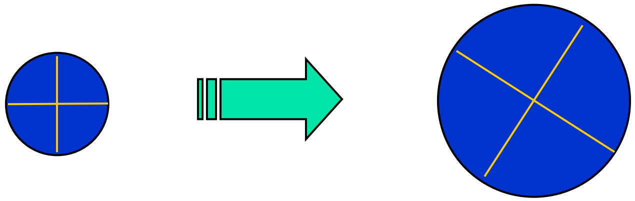




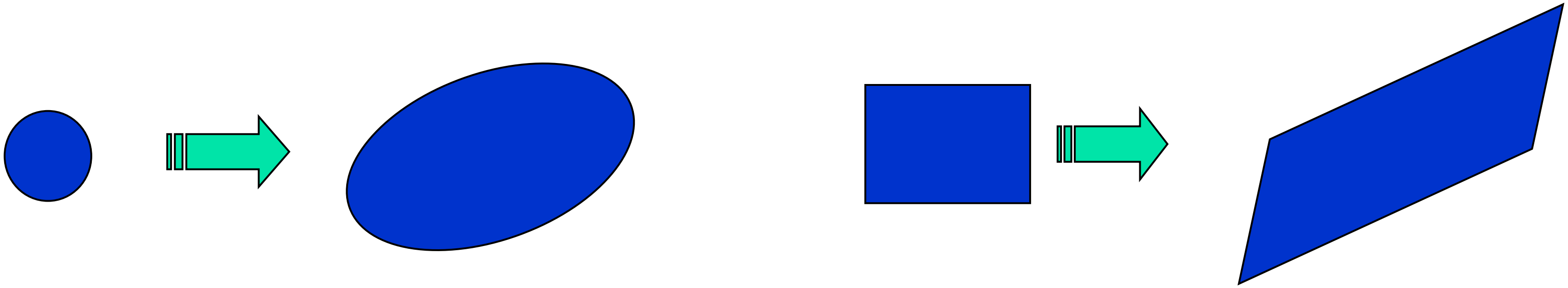
K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

Affine Invariant Detection

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



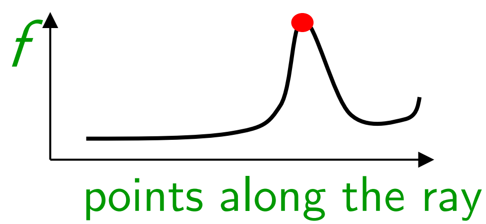
- Now we go on to:
  Affine transform (rotation + non-uniform scale)



Affine Invariant Detection

CENTER FOR MACHINE PERCEPTION

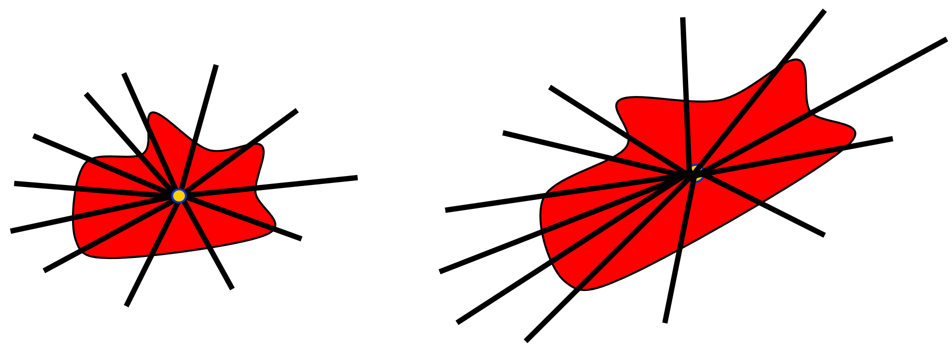
- Take a local intensity extremum as initial point
- Go along every ray starting from this point and stop when extremum of function f is reached



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

- 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

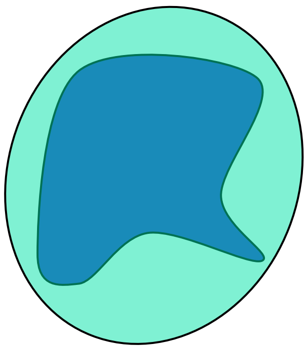
- The regions found may not exactly correspond, so we approximate them with ellipses

- Geometric Moments:

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

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

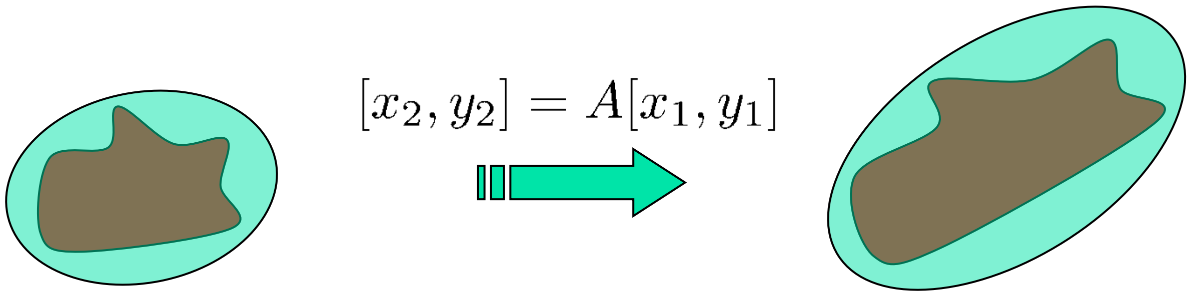
Taking f to be the characteristic function of a region (1 inside, 0 outside), moments of orders up to 2 allow to approximate the region by an ellipse



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

Affine Invariant Detection

- Covariance matrix of region points defines an ellipse:



$$[x_1, y_1]^T \Sigma_1^{-1} [x_1, y_1] = 1$$

$$\Sigma_1 = \langle [x_1, y_1][x_1, y_1]^T \rangle_{\text{region}_1}$$

$$[x_2, y_2]^T \Sigma_2^{-1} [x_2, y_2] = 1$$

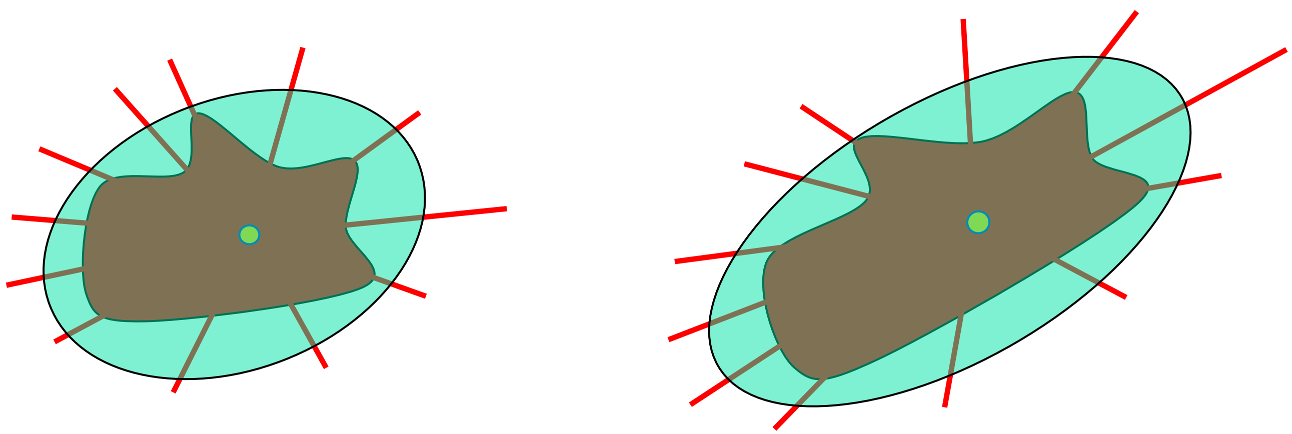
$$\Sigma_2 = \langle [x_2, y_2][x_2, y_2]^T \rangle_{\text{region}_2}$$

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

Ellipses, computed for corresponding regions, also correspond!

Affine Invariant Detection

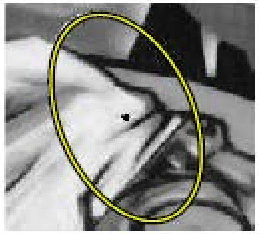
- Algorithm summary (detection of affine invariant region):
  - Start from a *local intensity extremum* point
  - Go in *every direction* until the point of extremum of some function f
  - Curve connecting the points is the region boundary
  - Compute *geometric moments* of orders up to 2 for this region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



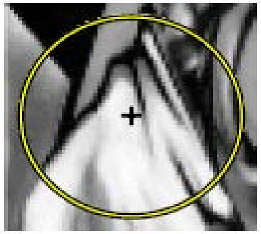
T.Tuytelaars, L.V.Gool. "Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions". BMVC 2000.

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



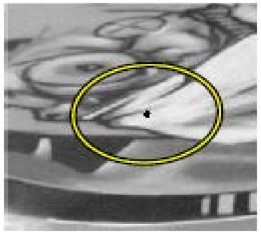
$$[x_1, y_1] \rightarrow M_1^{-1/2}[x'_1, y'_1]$$



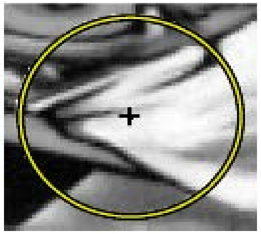
$$\downarrow$$

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

$$\downarrow$$



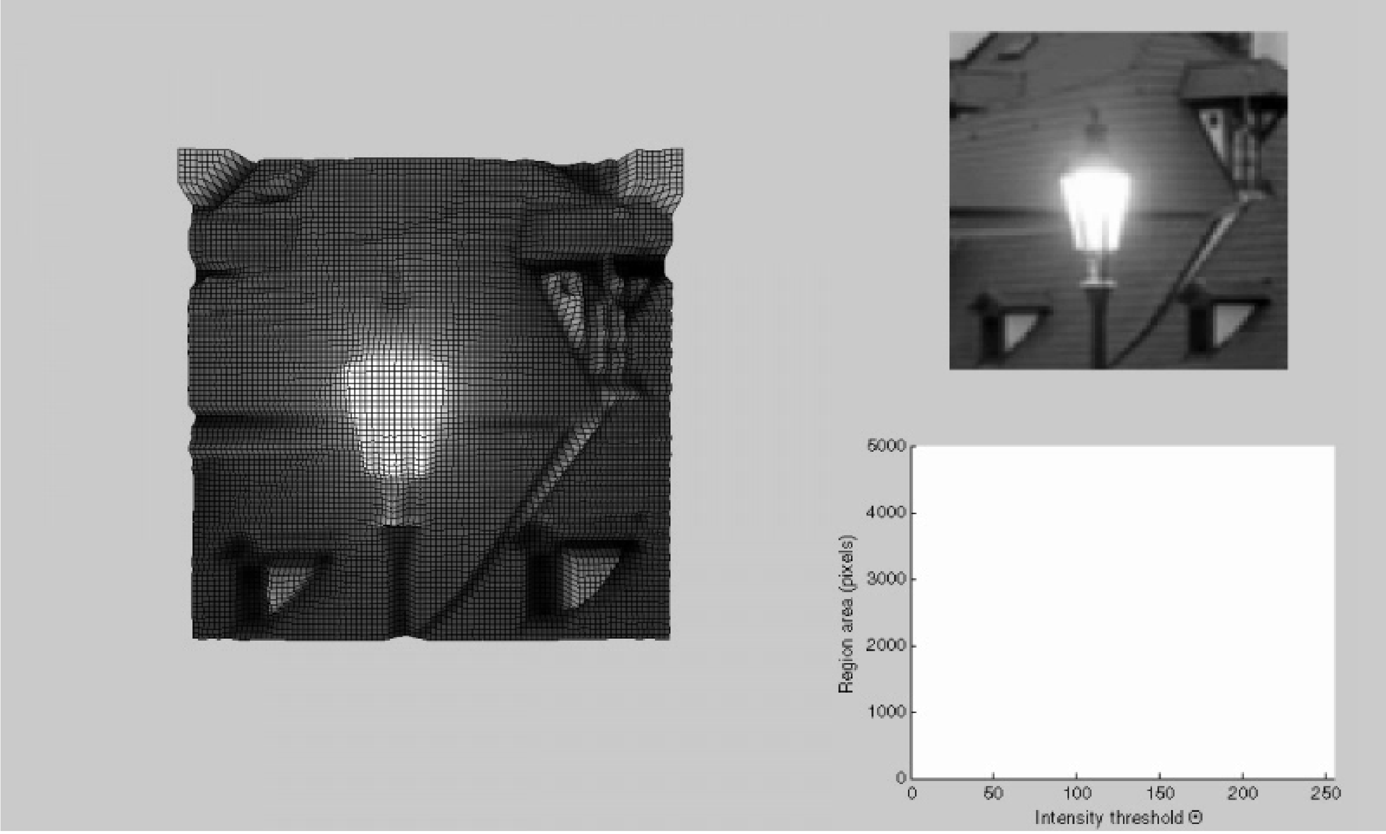
$$[x_2, y_2] \rightarrow M_2^{-1/2}[x'_2, y'_2]$$



The Maximally Stable Extremal Regions

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

video



The Maximally Stable Extremal Regions

video

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

Descriptors of Local Invariant Features

Descriptors Invariant to Rotation

- 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:

$$\theta \to \theta + \theta_0$$

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 $[|m_{kl}|]_{k,l}$

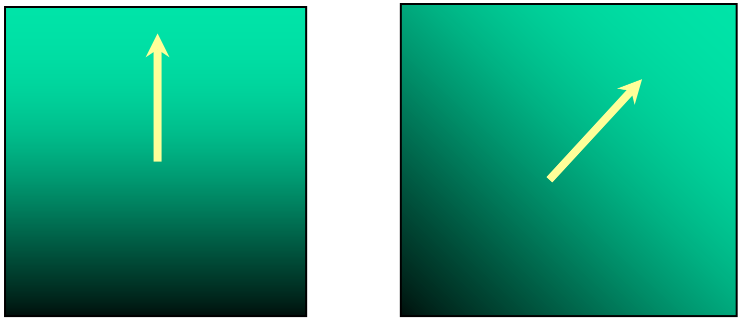
J.Matas et.al. "Rotational Invariants for Wide-baseline Stereo". Res. Report of CMP, 2003

Descriptors Invariant to Rotation

CENTER FOR MACHINE PERCEPTION

- 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

- 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

- Affine invariant color moments

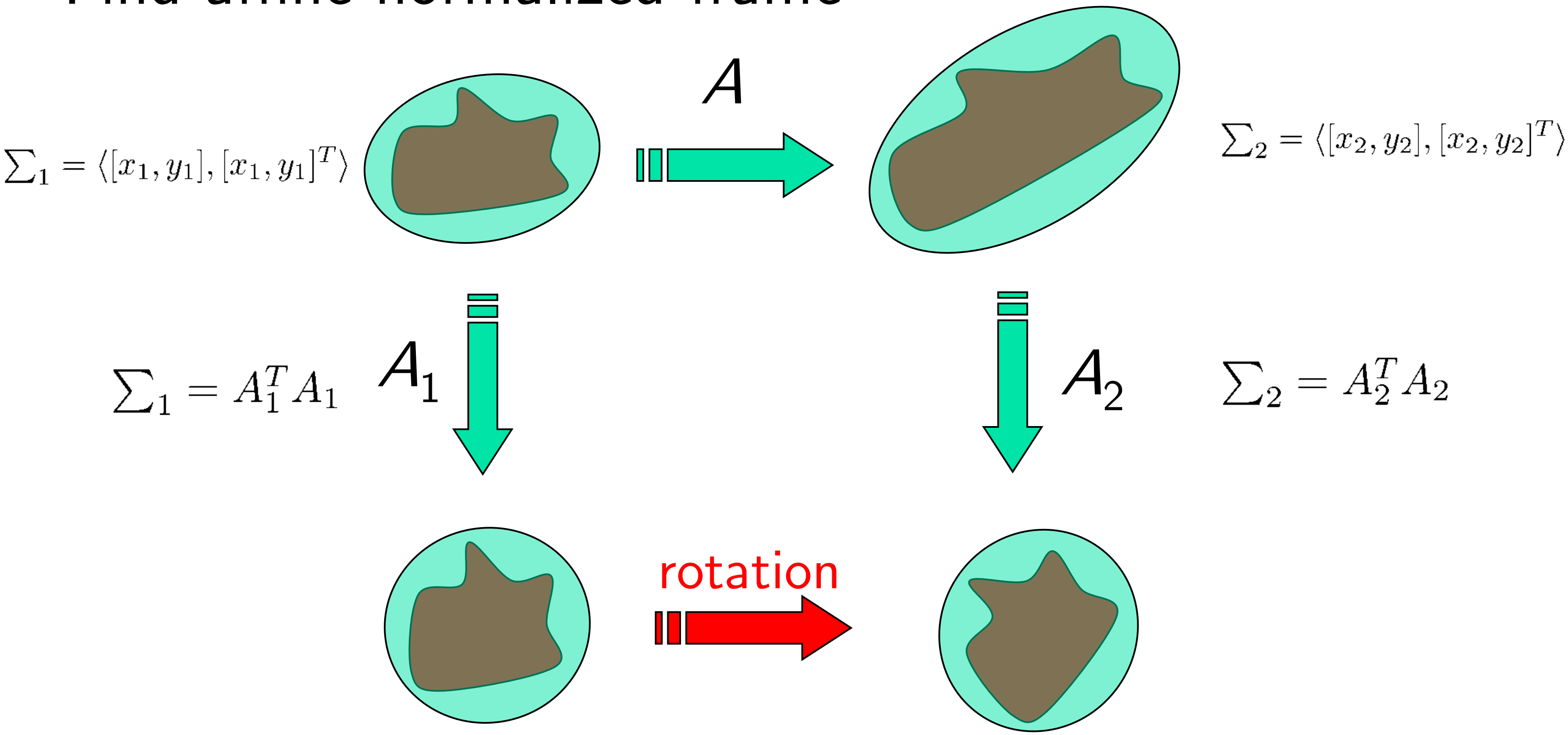
$$m_{pq}^{abc} = \int_{region} x^p y^q R^a(x,y) G^b(x,y) B^c(x,y)\,dxdy$$

- 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



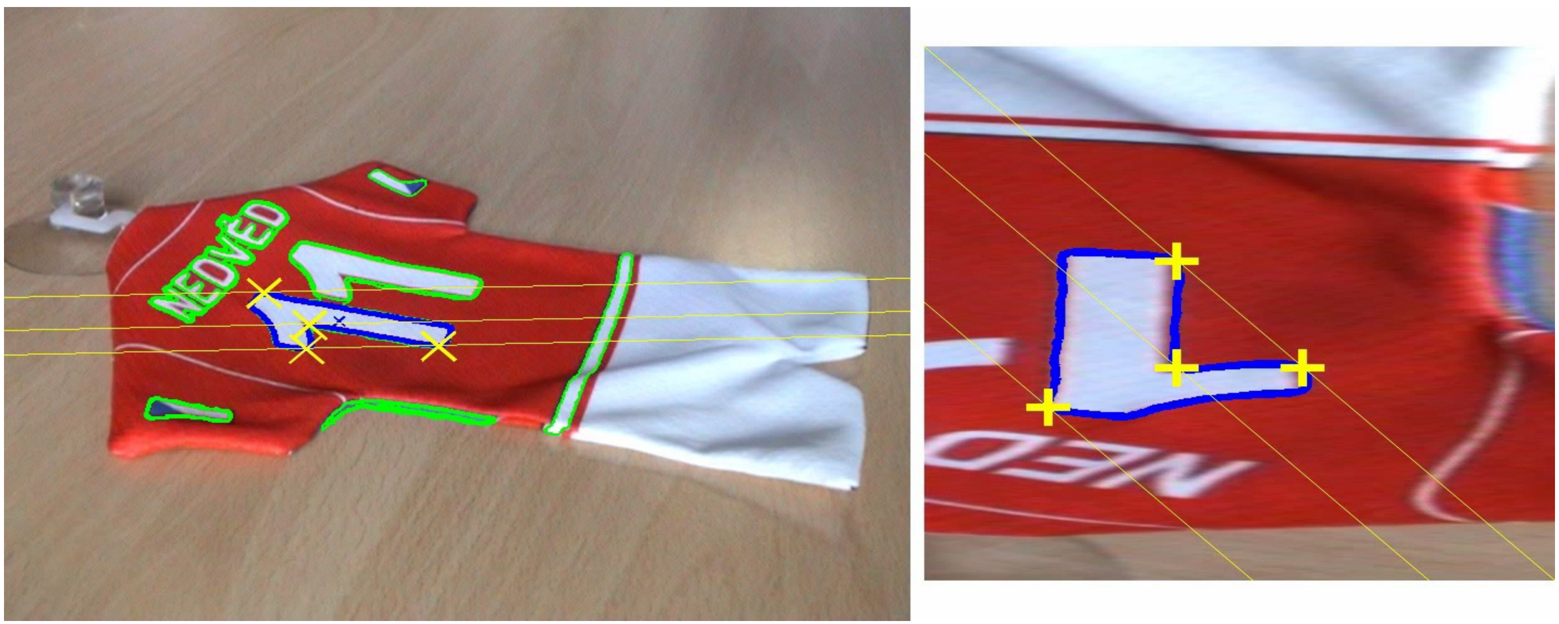
- Compute rotational invariant descriptor in this normalized frame

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

Affine-Covariant Constructions: Taxonomy

CENTER FOR MACHINE PERCEPTION

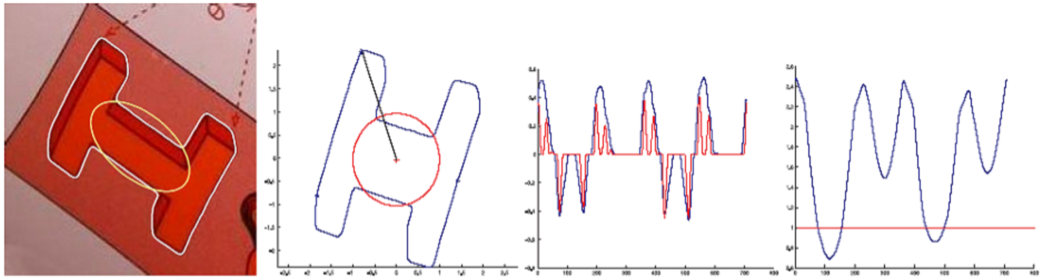
- Derived from ***region outer boundary***
  - Region area (1 constraint)
  - Center of gravity (2 constraints)
  - Matrix of second moments (symmetric 2x2 matrix: 3 constraints)

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

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

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

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

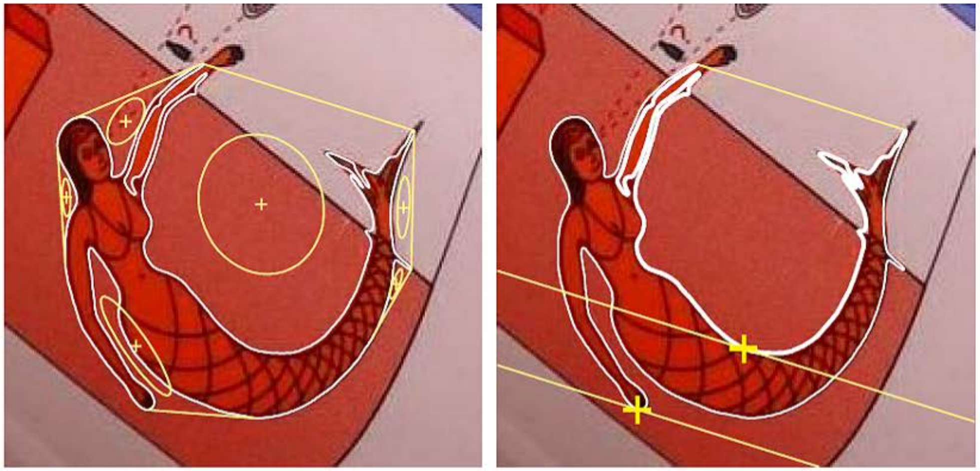


(a) (b) (c) (d)

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

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

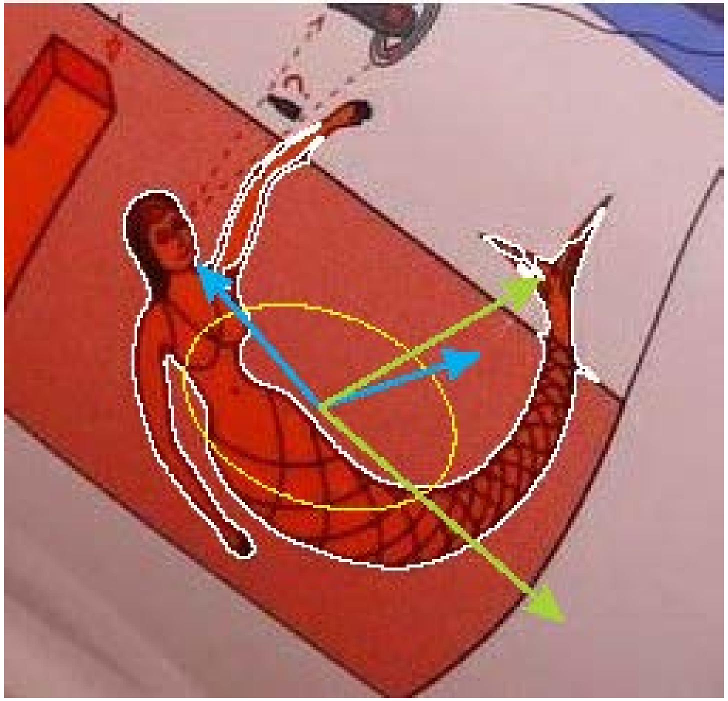


(a) (b)

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.

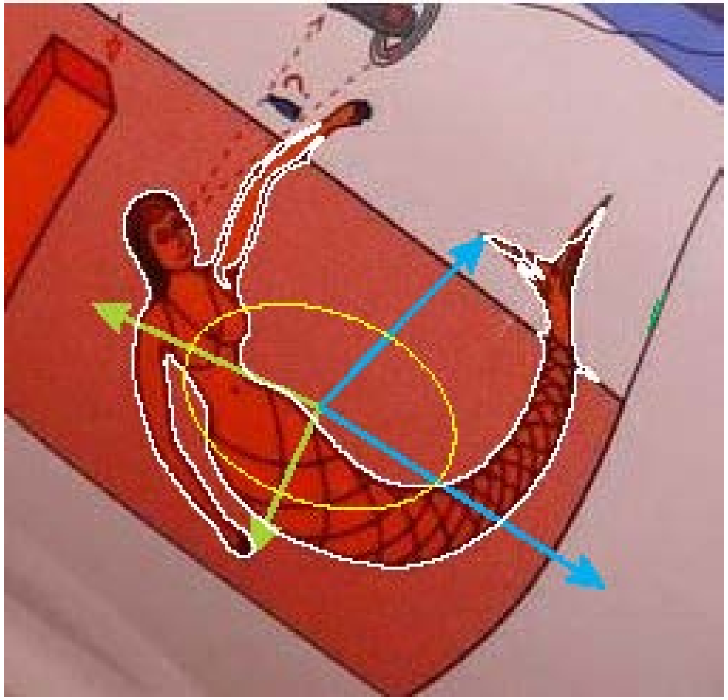
Constructions of Local Affine Frames

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



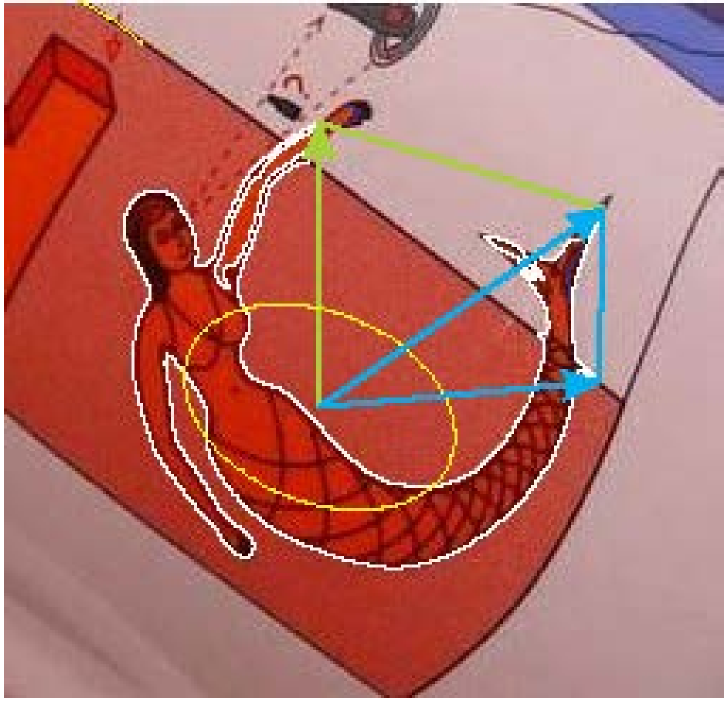
Constructions of Local Affine Frames

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



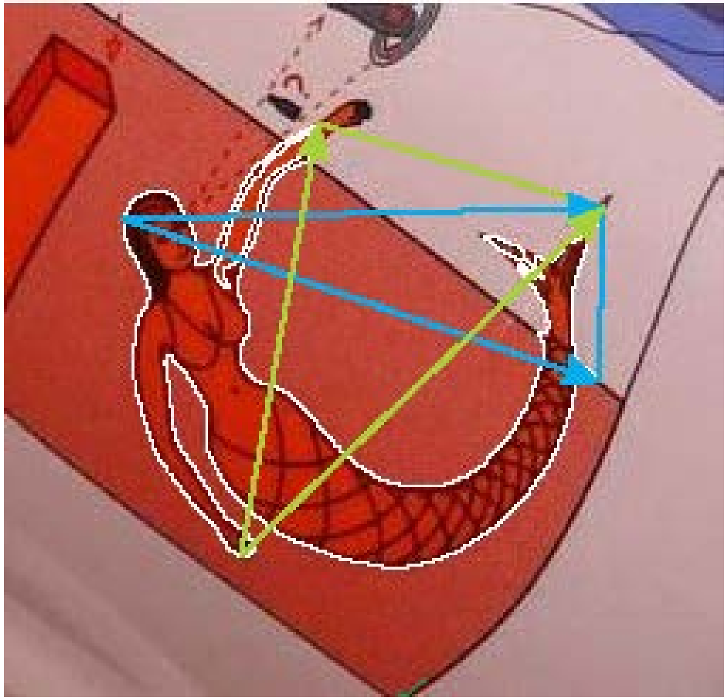
Constructions of Local Affine Frames

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



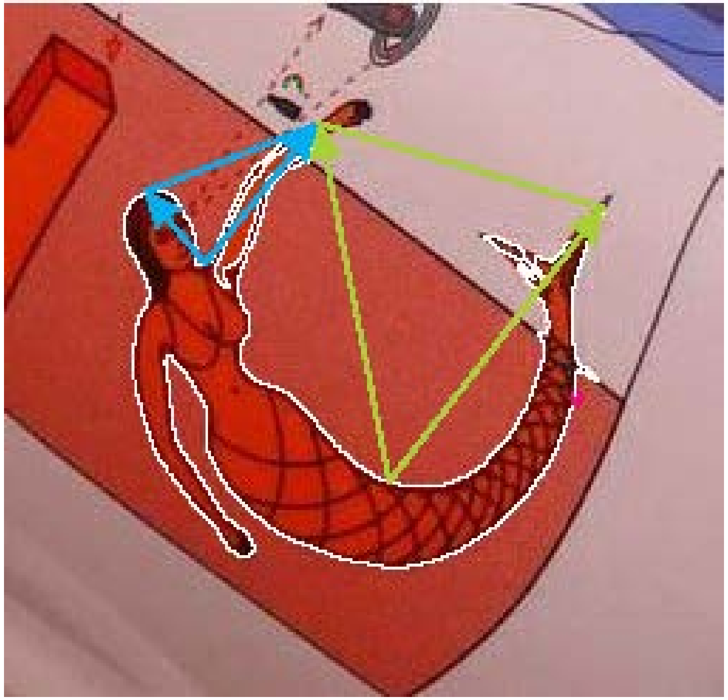
Constructions of Local Affine Frames

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



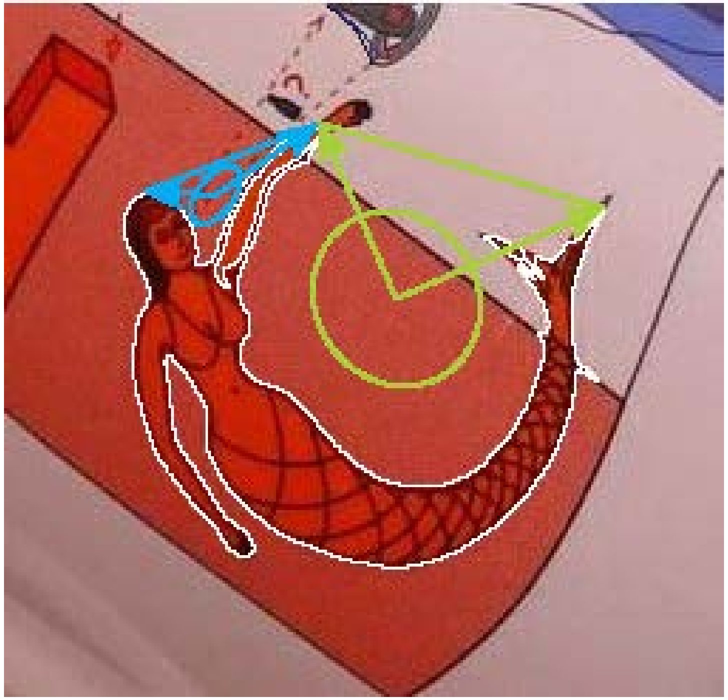
Constructions of Local Affine Frames

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



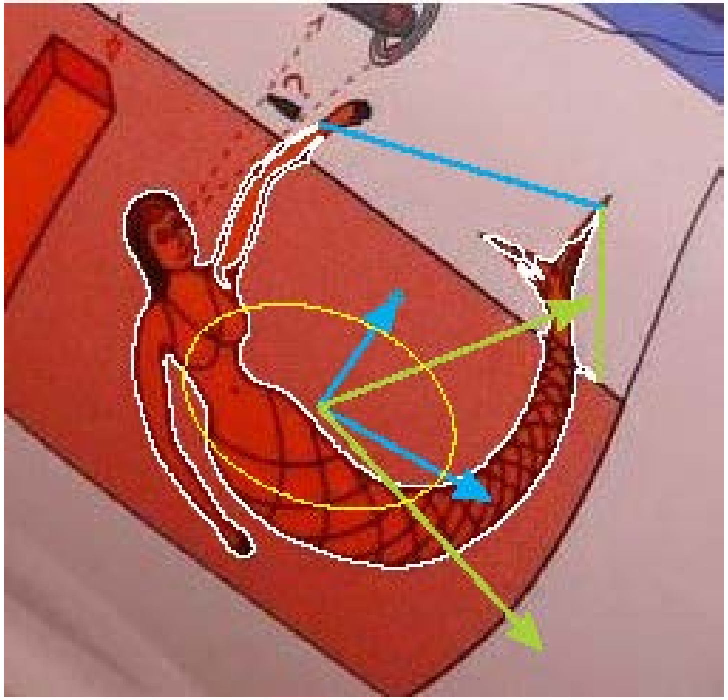
Constructions of Local Affine Frames

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



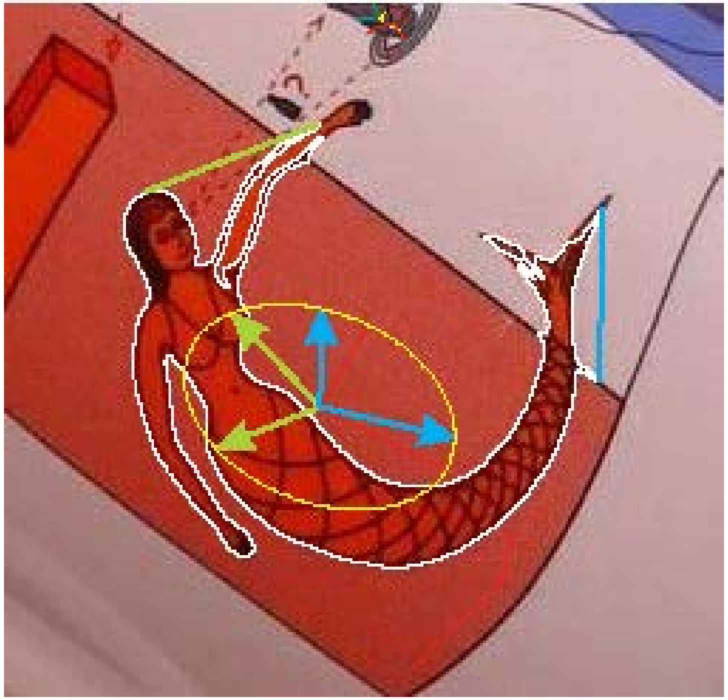
Constructions of Local Affine Frames

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



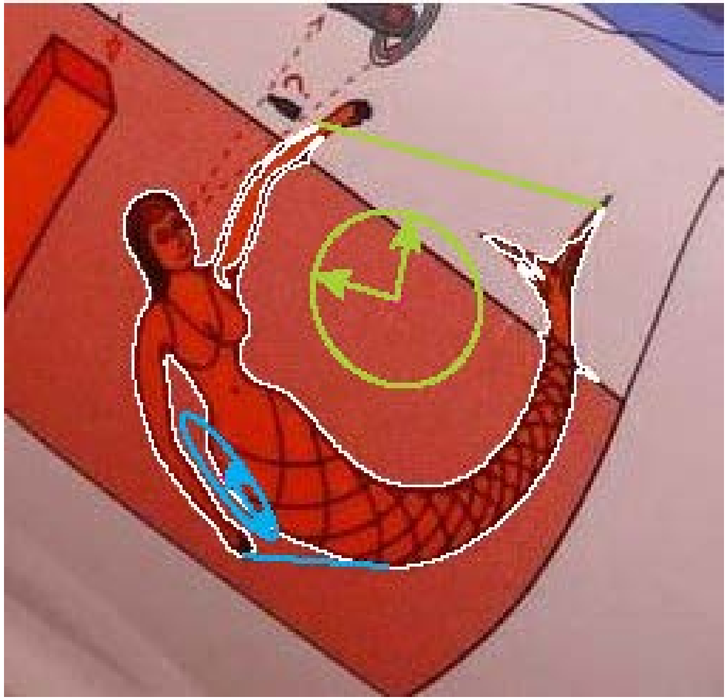
Constructions of Local Affine Frames

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



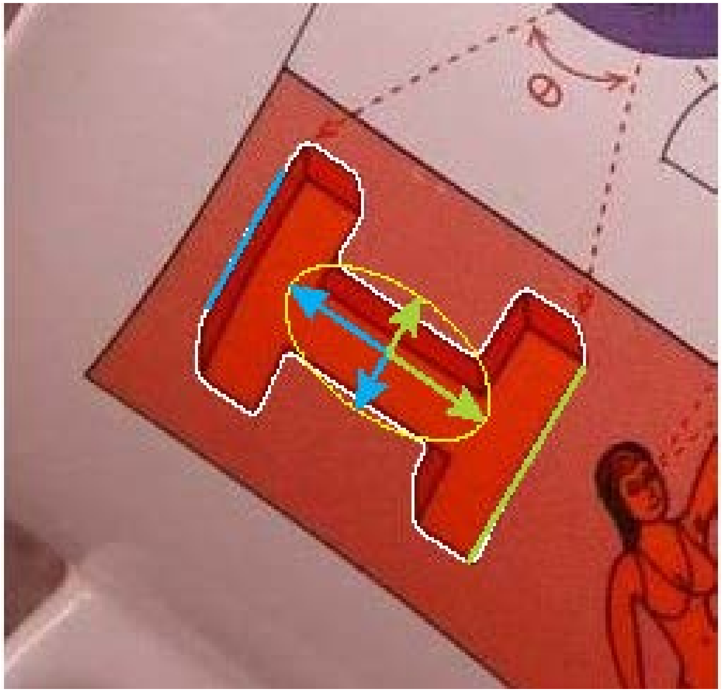
Constructions of Local Affine Frames

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



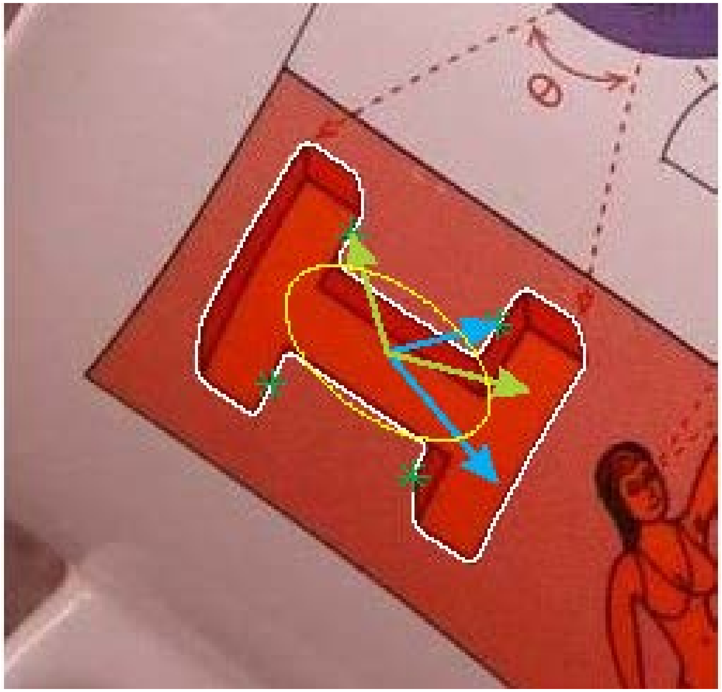
Constructions of Local Affine Frames

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



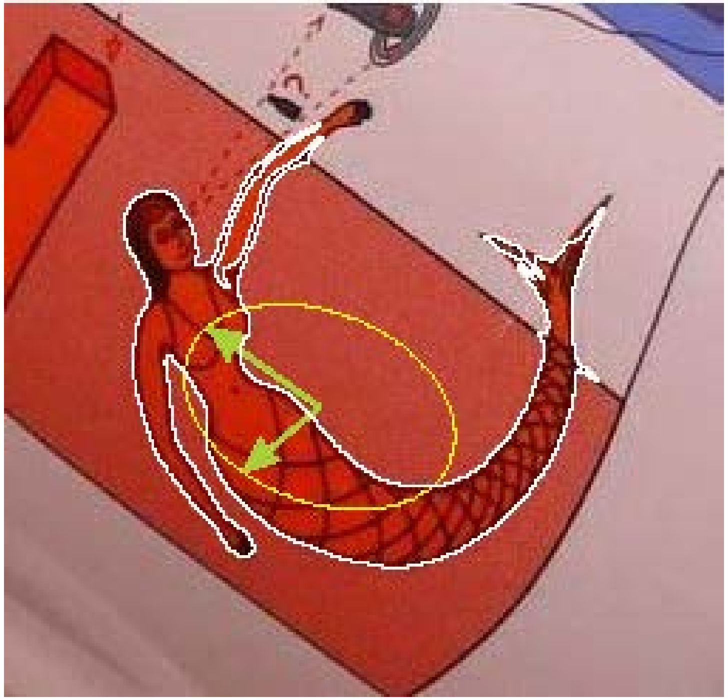
Constructions of Local Affine Frames

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



Constructions of Local Affine Frames

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



Common Structure of "Local Feature" Algorithms

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

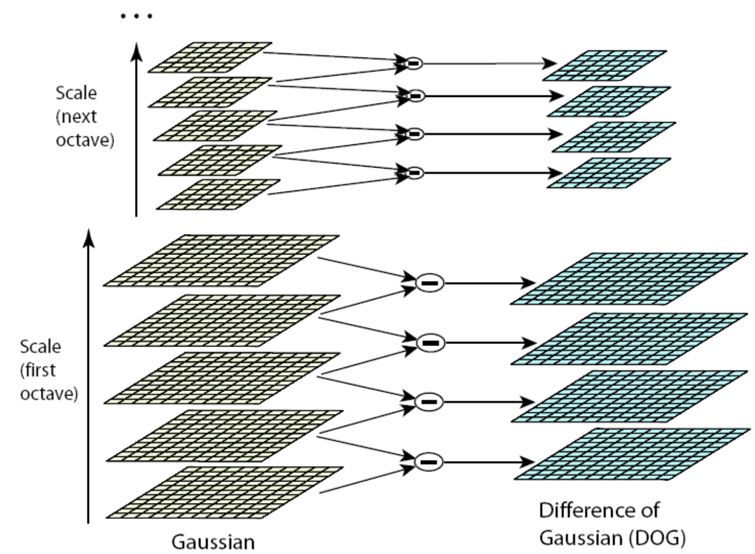
Matching:

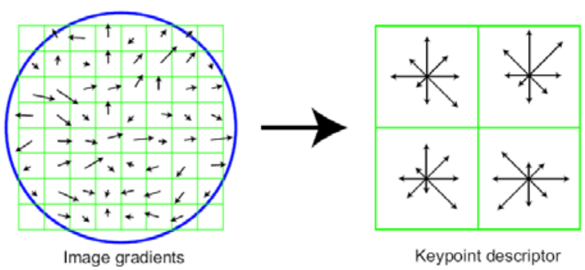
- test on Euclidean distance of 1st and 2nd match

Verification:

- Hough transform based clustering of correspondences with similar transformations

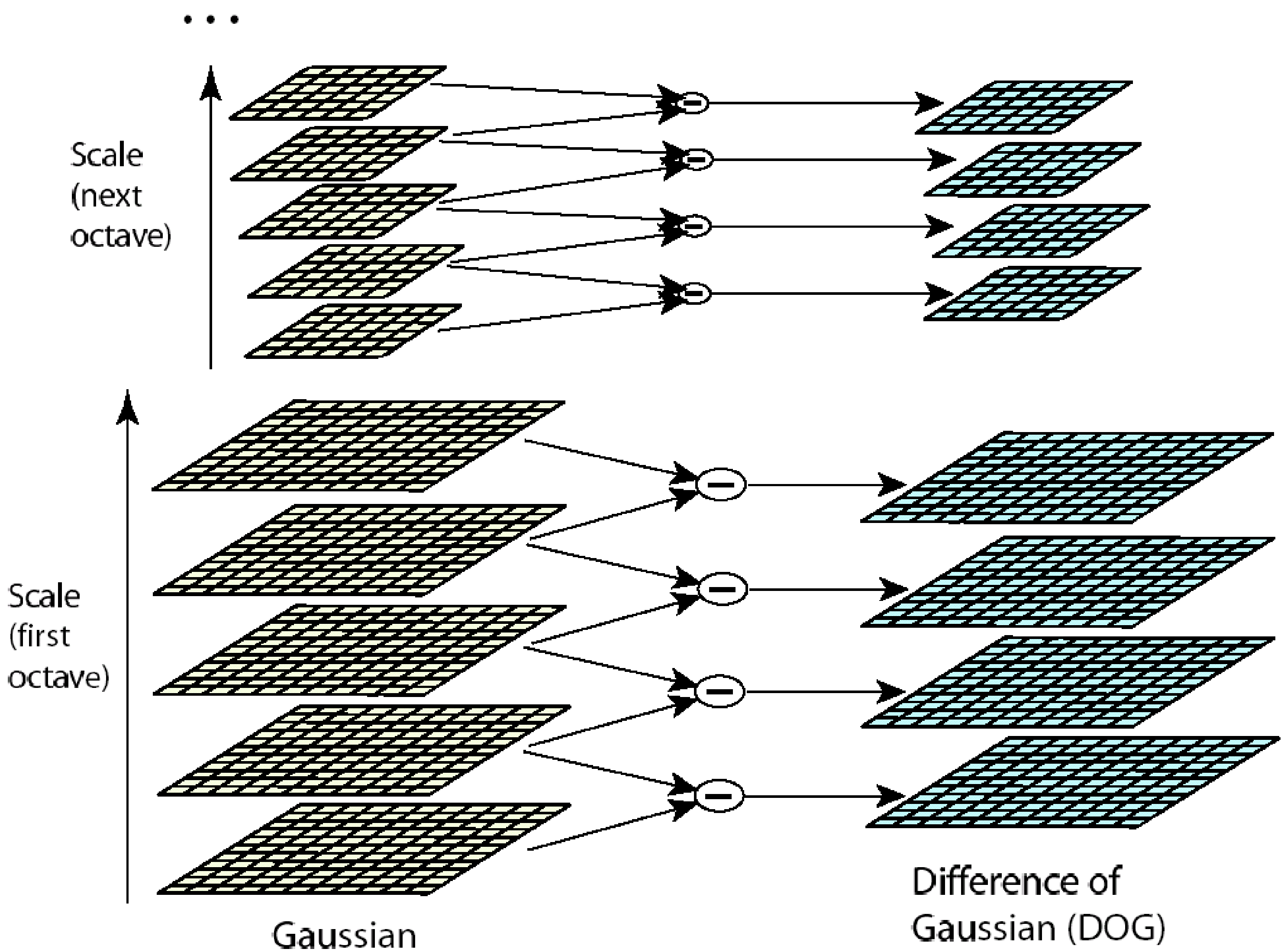
Fast, efficient implementation**, real-time recognition**





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

Scale space processed one octave at a time



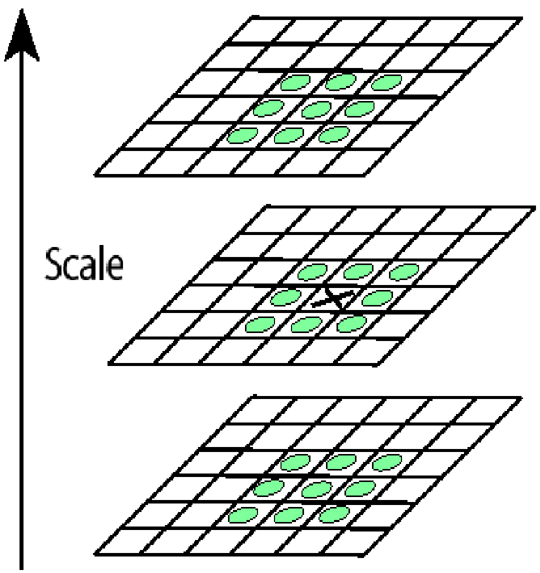
Sub-pixel/ Sub-level Keypoint Localization

- 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:

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

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

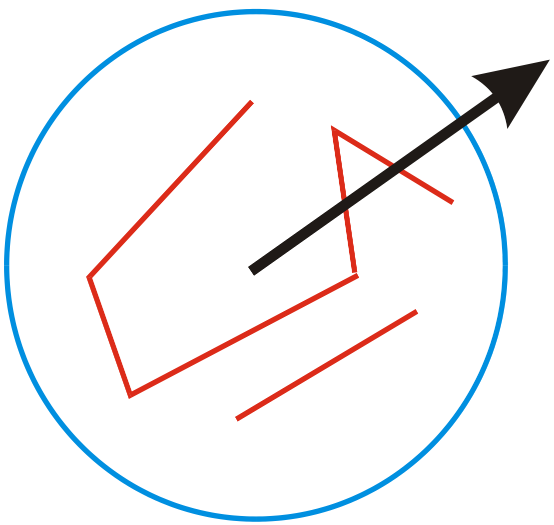
$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}$$

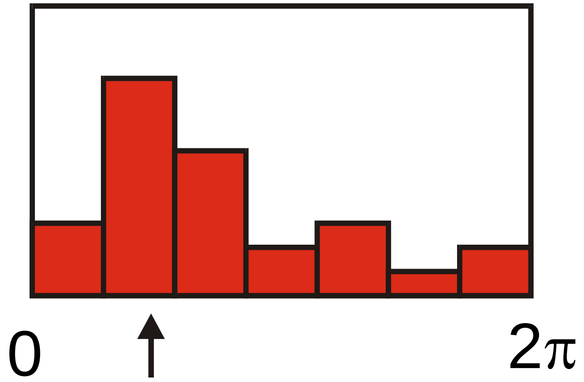


Select canonical orientation(s)

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

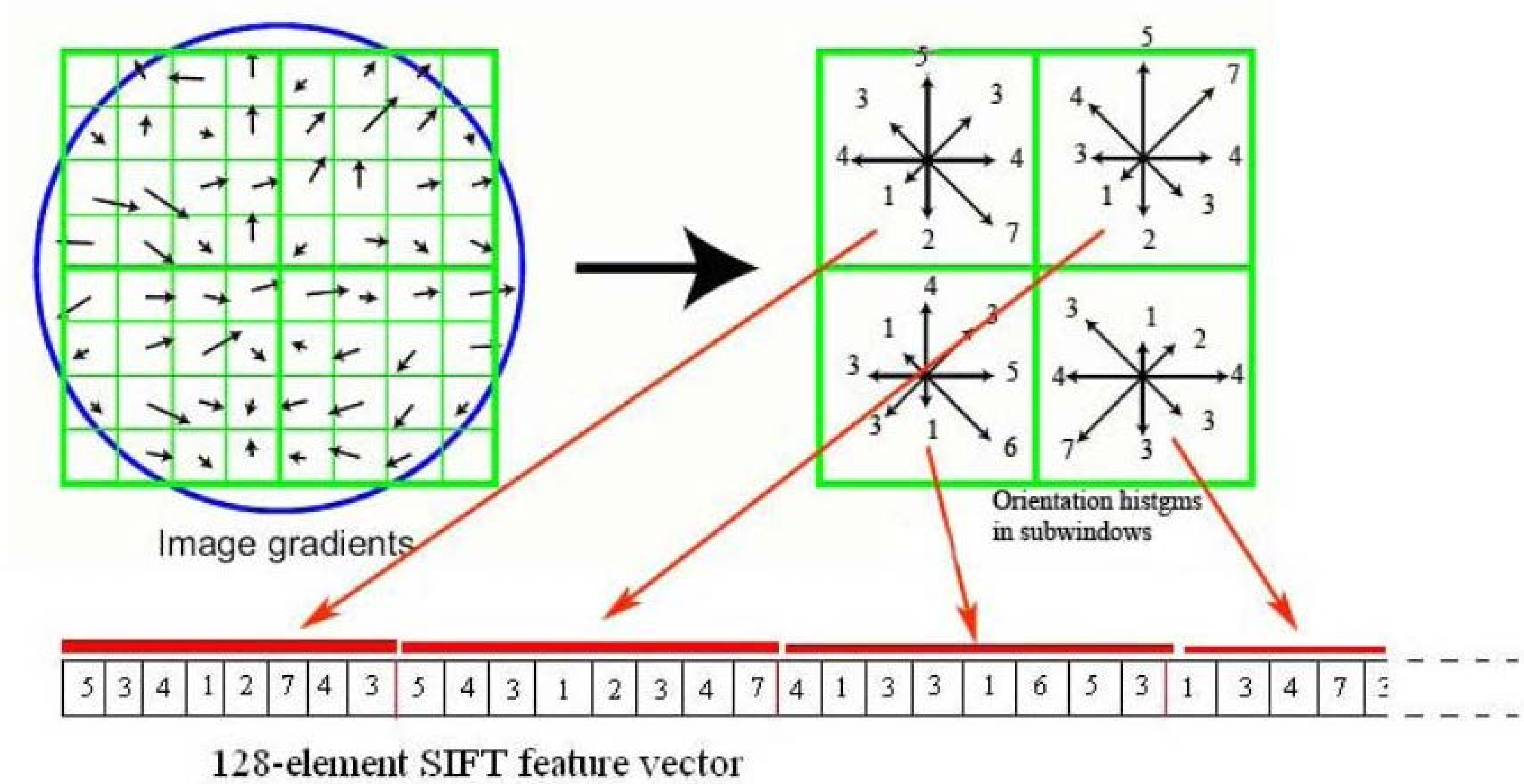
Note: if orientation of the object (image) is known, it may replace this construction





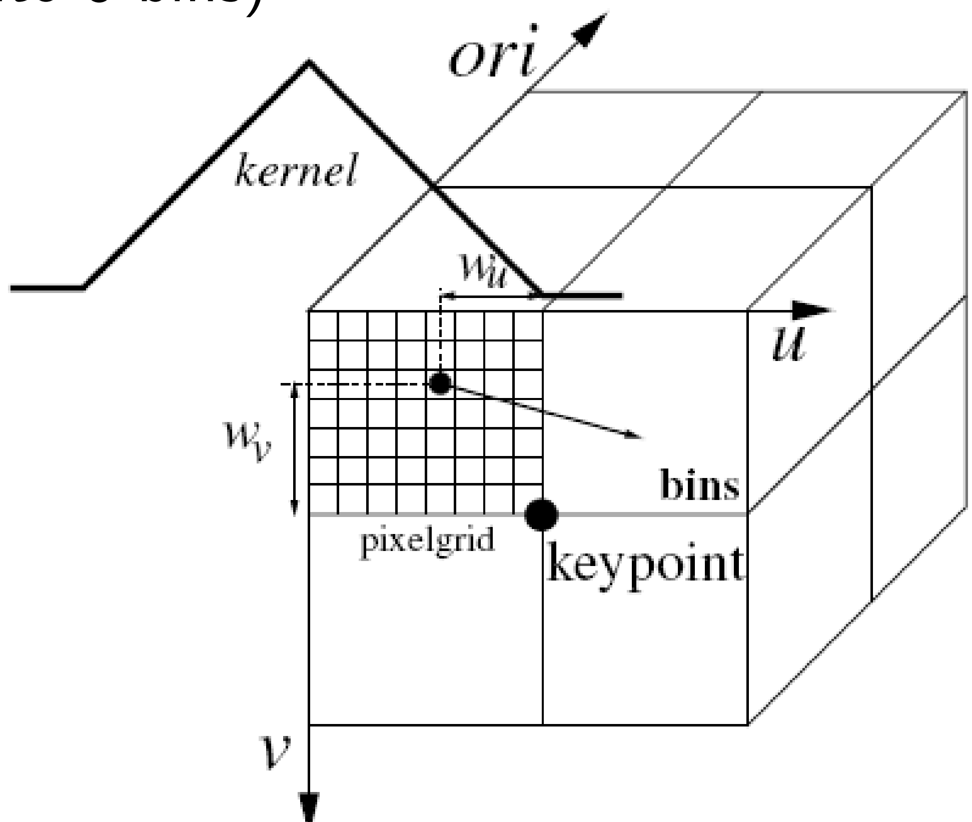
SIFT Descriptor

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



SIFT Descriptor

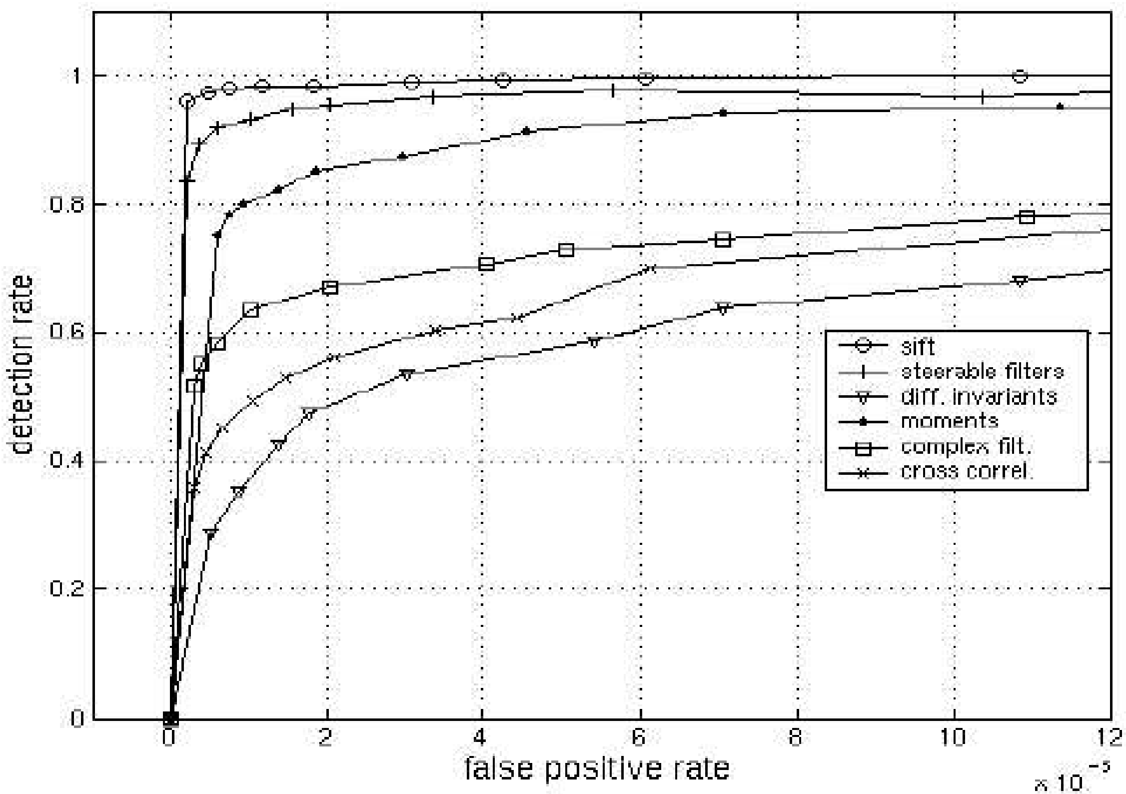
- SIFT descriptor can be viewed as a 3–D histogram in which two dimensions correspond to image spatial dimensions and the additional dimension to the image gradient direction (normally discretised into 8 bins)



SIFT – Scale Invariant Feature Transform[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

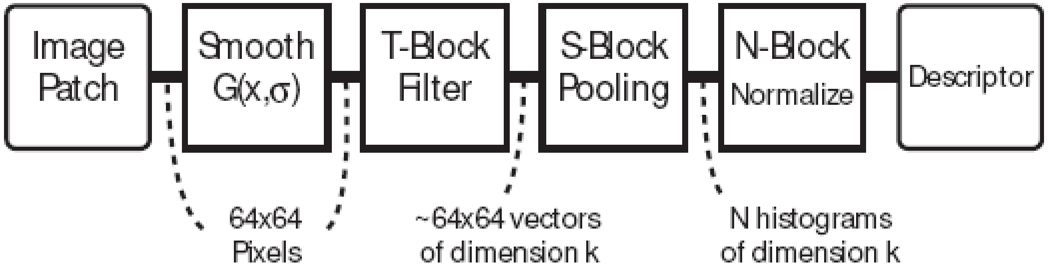
SIFT invariances

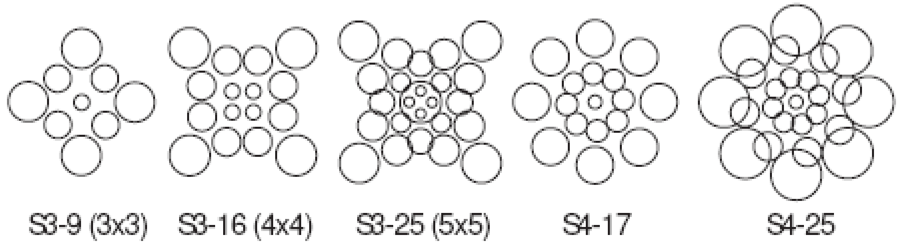
- Based on gradient orientations, which are robust to illumination changes
- Spatial binning gives tolerance to small shifts in location and scale, affine change.
- Explicit orientation normalization
- Photometric normalization by making all vectors unit norm
- Orientation histogram gives robustness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

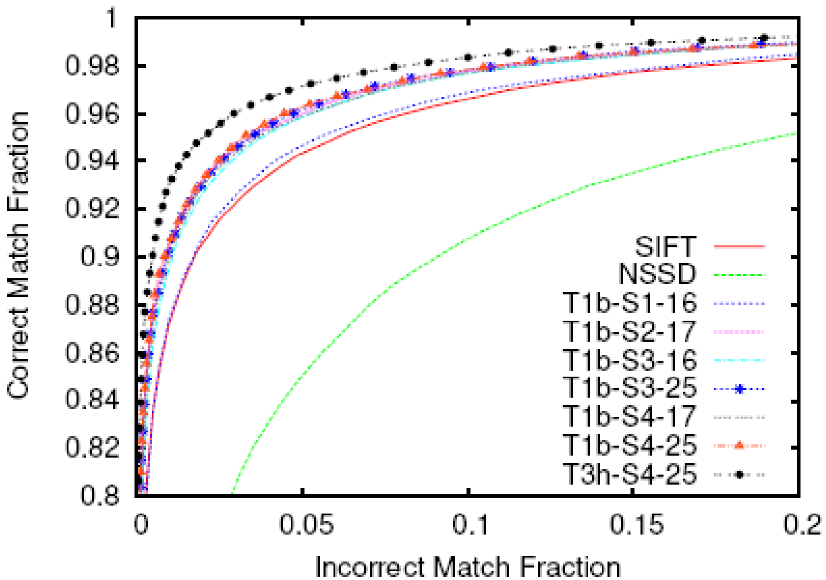
SIFT Descriptor

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

Learning Local Image Descriptors







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

Courtesy of Simon A.J.Winder, Matthew Brown, Microsoft Research, Redmont, USA

DAISY local image descriptor

I. Histograms at every pixel location are computed

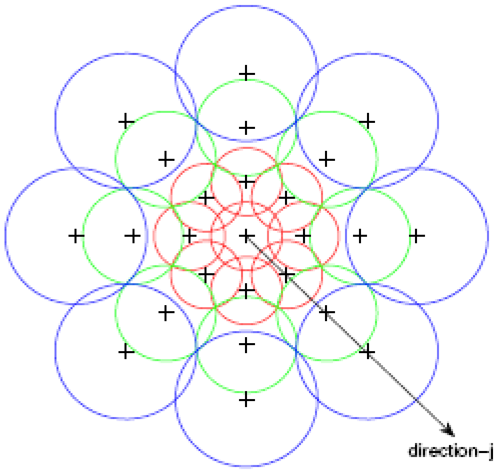
$$\mathbf{h}_{\Sigma}(u,v) = \left[\mathbf{G}_1^{\Sigma}(u,v), \dots, \mathbf{G}_8^{\Sigma}(u,v)\right]^{\top},$$

$\mathbf{h}_{\Sigma}(u,v)$: histogram at location (u, v)

$\mathbf{G}_1^{\Sigma}$: Gaussian convolved orientation maps

II. Histograms are normalized to unit norm

III. Local image descriptor is computed as



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

DAISY v. SIFT: computational complexity

CENTER FOR MACHINE PERCEPTION

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

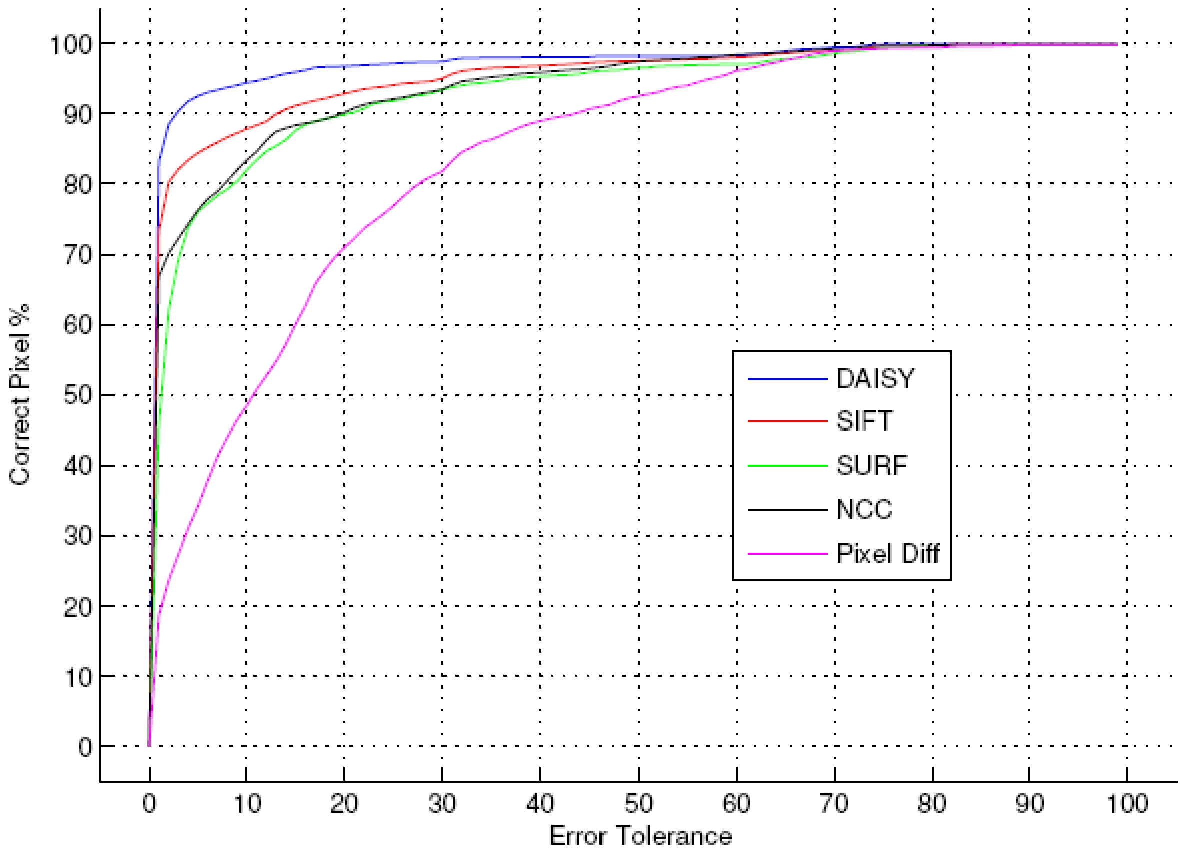
$$\mathbf{G}_o^{\Sigma_2} = G_{\Sigma_2} * \left(\frac{\partial \mathbf{I}}{\partial o}\right)^+ = G_{\Sigma} * G_{\Sigma_1} * \left(\frac{\partial \mathbf{I}}{\partial o}\right)^+ = G_{\Sigma} * \mathbf{G}_o^{\Sigma_1},$$

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

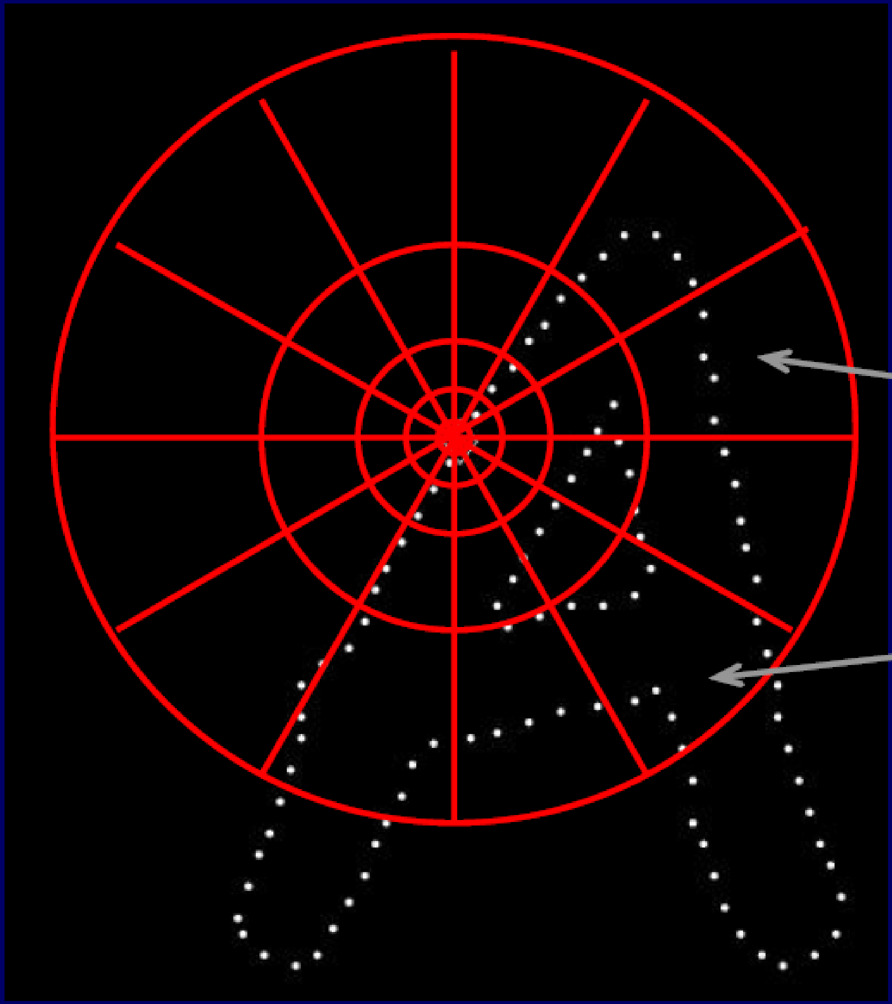
Image Size	DAISY	SIFT
800x600	5	252
1024x768	10	432
1290x960	13	651

Table 1. Computation Time Comparison (in seconds)

Results


100
90
80
70
60
50
40
30
20
10
0
Correct Pixel %
0
10
20
30
40
50
60
70
80
90
100
Error Tolerance
DAISY
SIFT
SURF
NCC
Pixel Diff

Shape Context



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

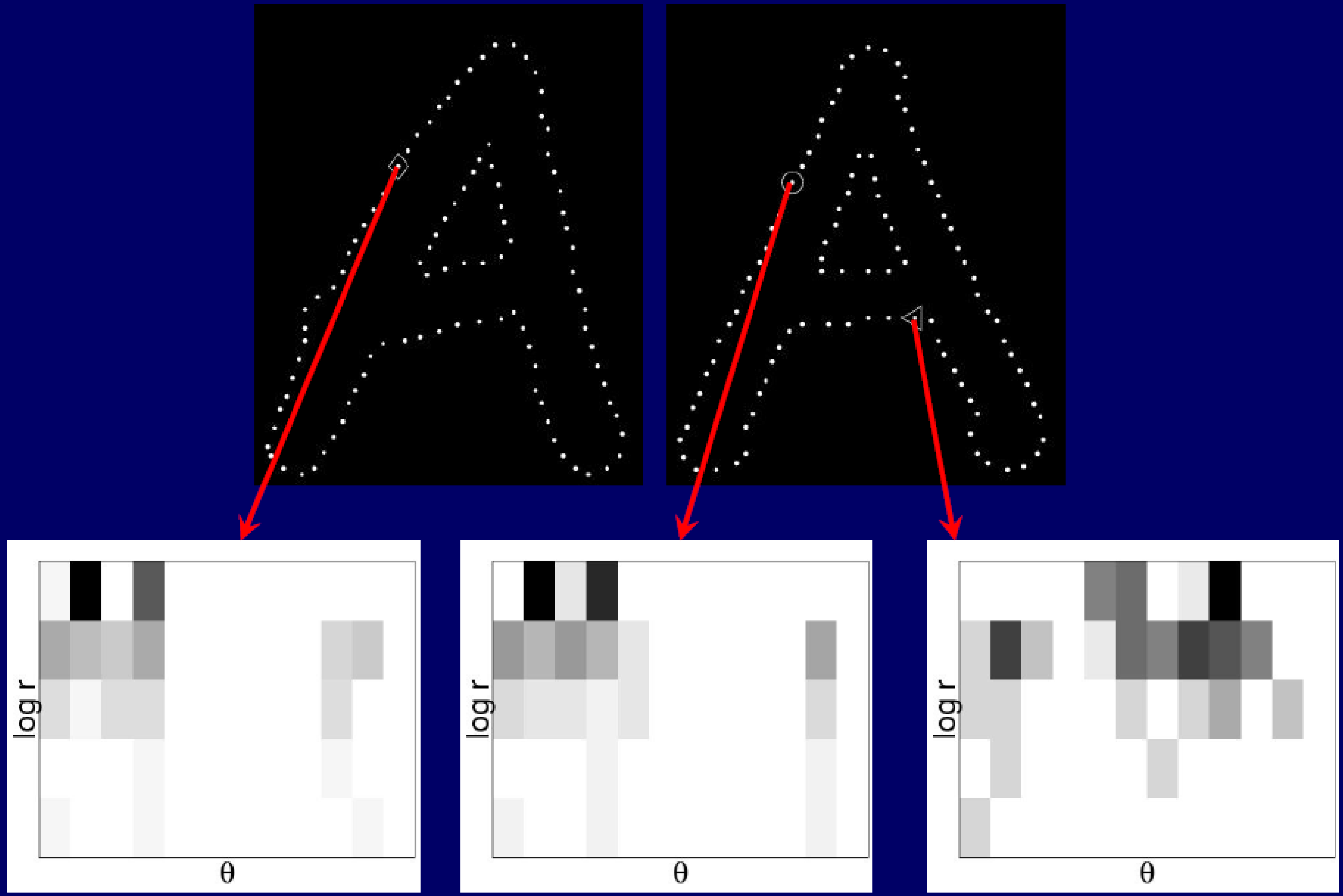
Count = 4

⋮

Count = 10

☞ Compact representation of distribution of points relative to each point

Shape Context



University of California
Berkeley

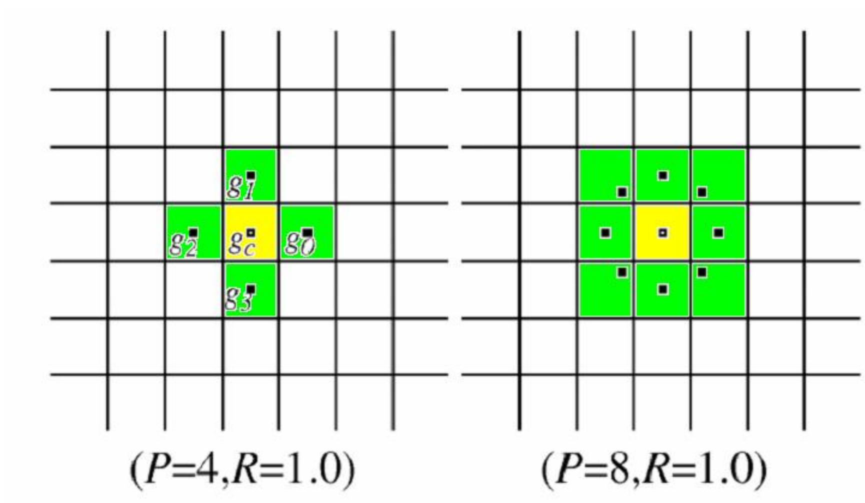
Computer Vision Group

Local Binary Pattern (LBP) Descriptor

CENTER FOR MACHINE PERCEPTION

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

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



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

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

LBP values in a 3 x 3 block

The LBP descriptor is invariant to any monotonic transformation of image

slide credit: Sara Arasteh et al

Rotation Invariant LBP ...

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

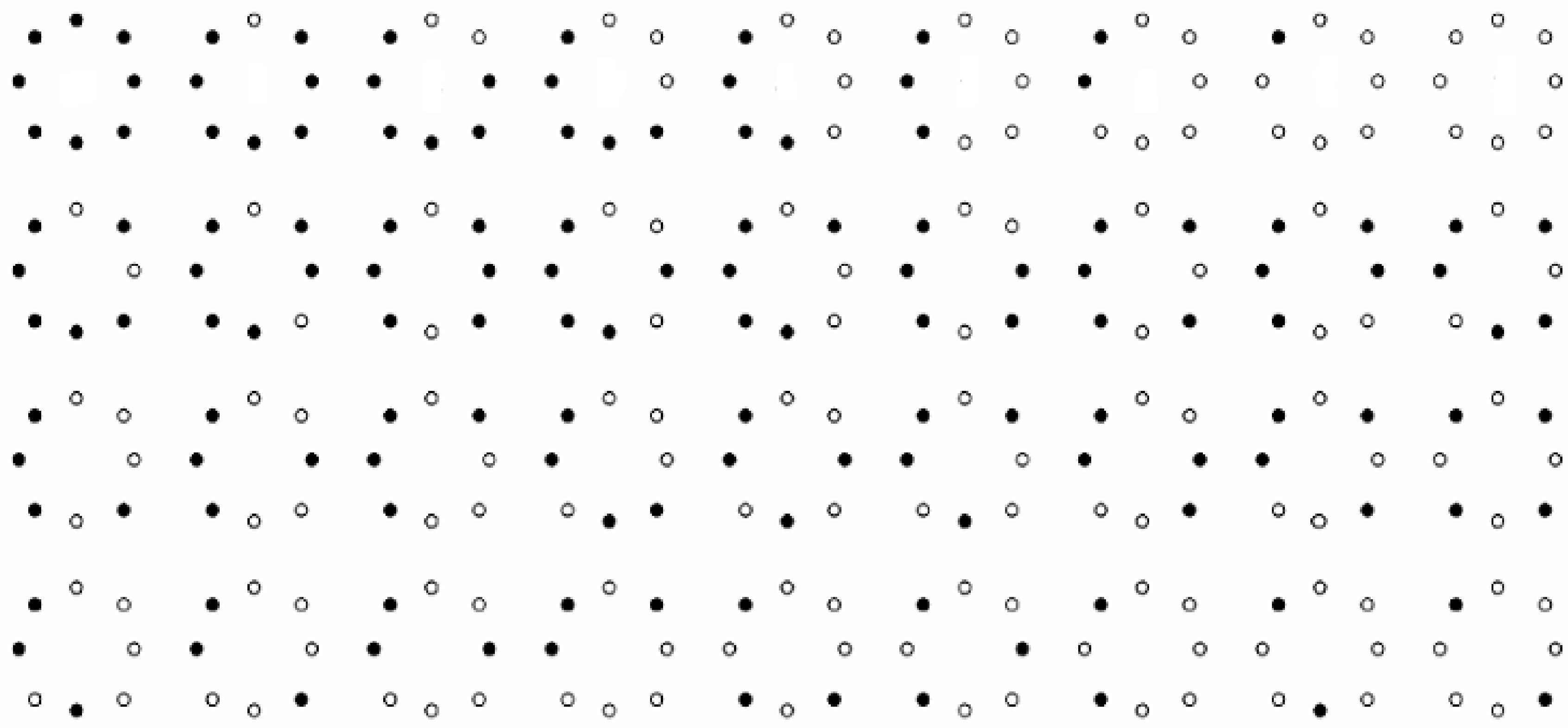
$$LBP_{P,R}^{ri} = \min\left\{ ROR(LBP_{P,R}, i) \quad \middle| \quad i = 0,1,\ldots, P-1 \right\}$$

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 ...

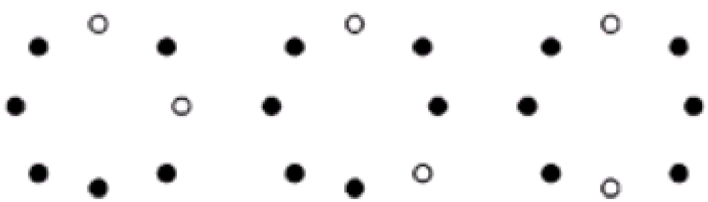
- This figure shows 36 unique rotation invariant binary patterns.



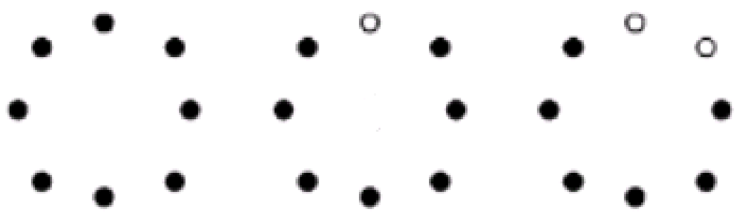
slide credit: Sara Arasteh et al

Rotation Invariant LBP ...

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



Samples of non-uniform patterns

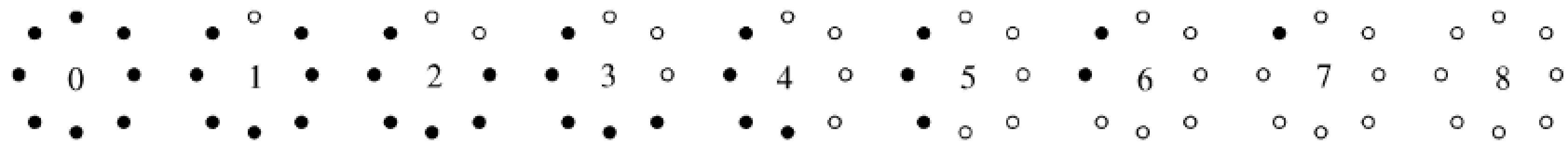


Samples of uniform patterns

slide credit: Sara Arasteh et al

Uniform LBP (ULBP)

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



Uniform Local Binary Patterns

LBPs are popular, numerous modifications exist

slide credit: Sara Arasteh et al

Matching Descriptors

Nearest-neighbor matching

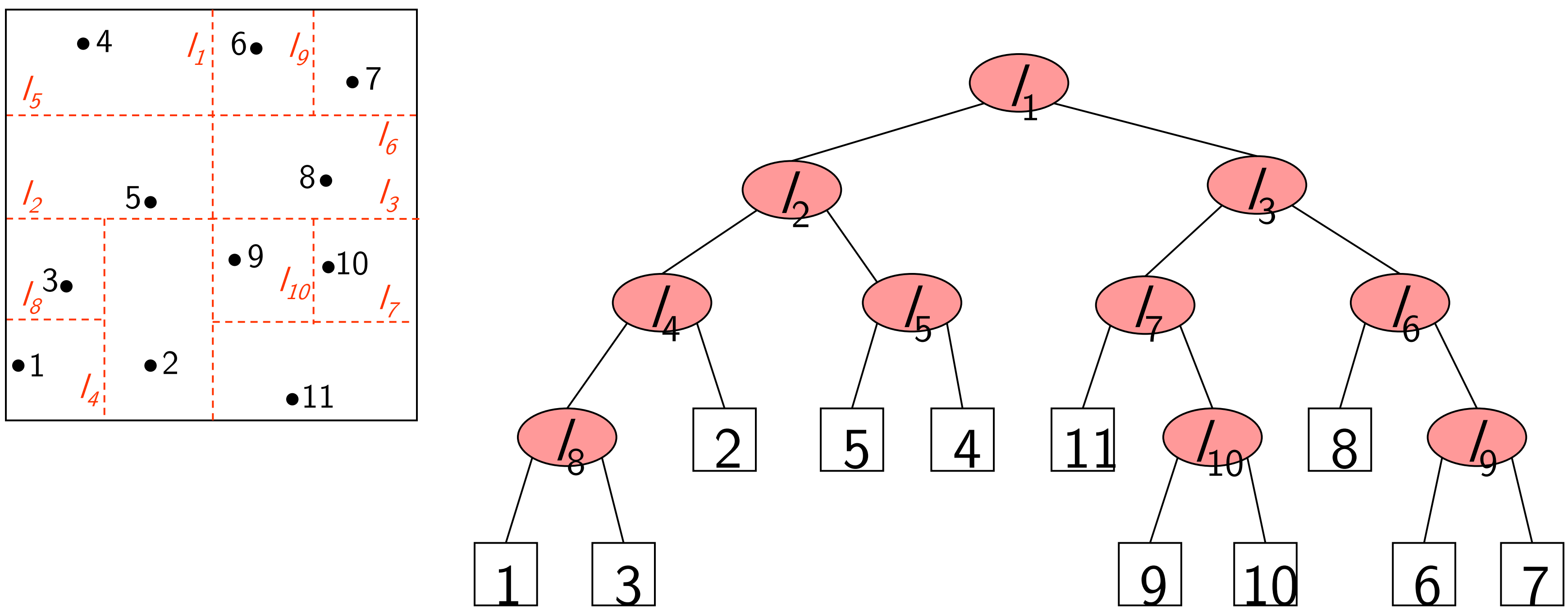
- Solve following problem for all feature vectors, **x**:

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

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

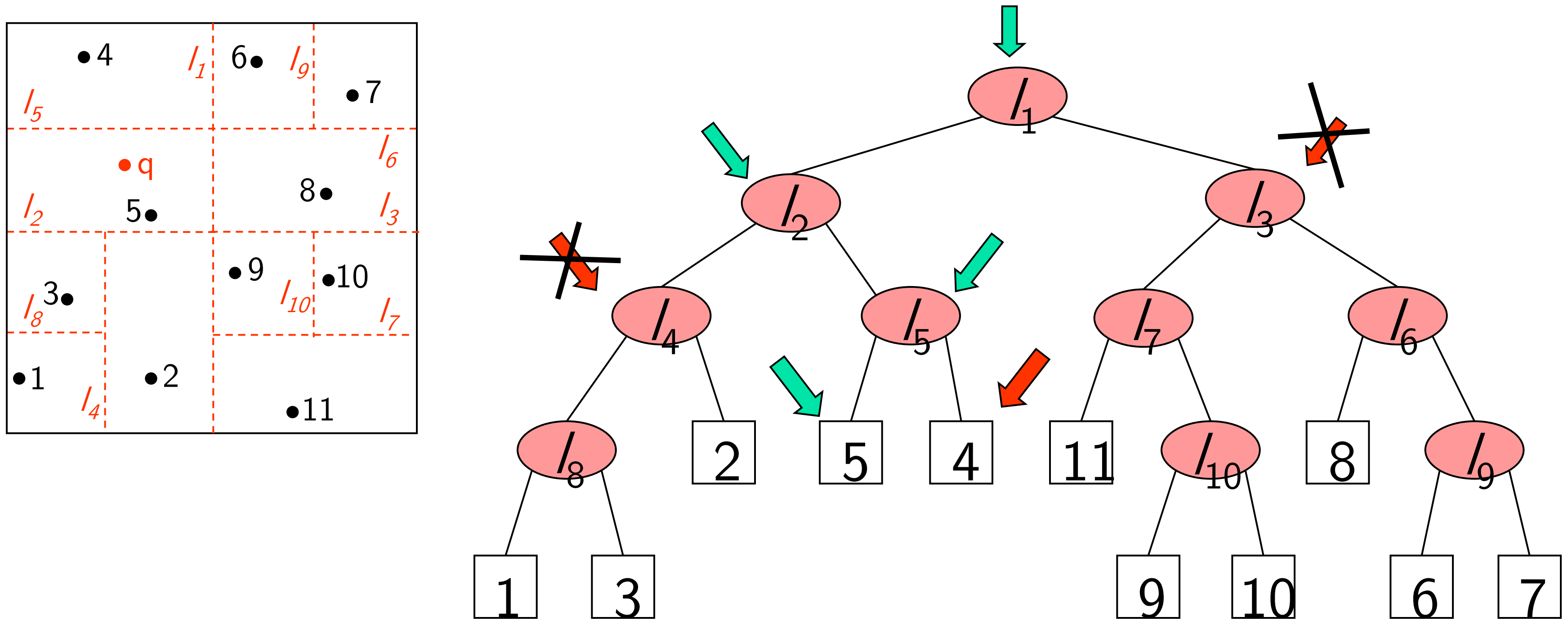
K-d tree construction

Simple 2D example



Slide credit: Anna Atramentov

K-d tree query

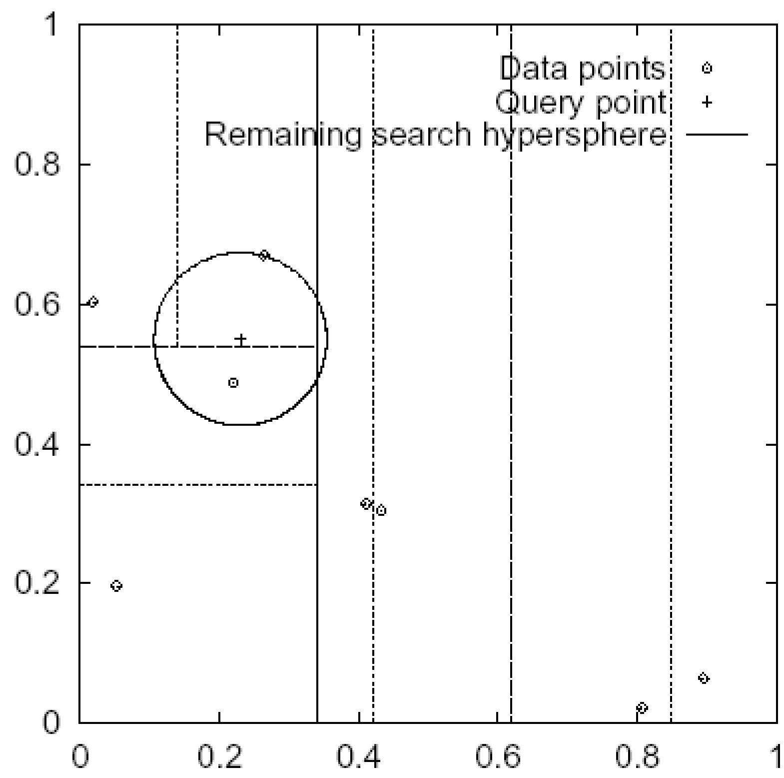


Slide credit: Anna Atramentov

Approximate k-d tree matching

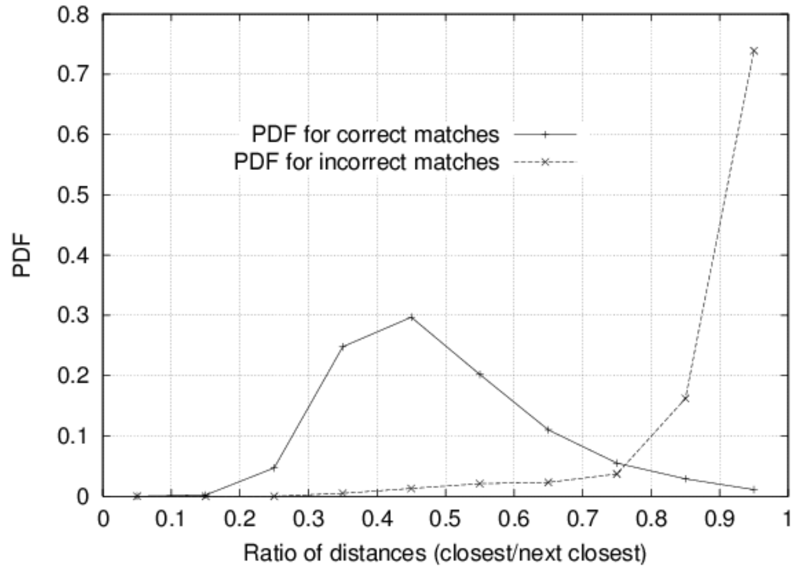
Key idea:

- 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

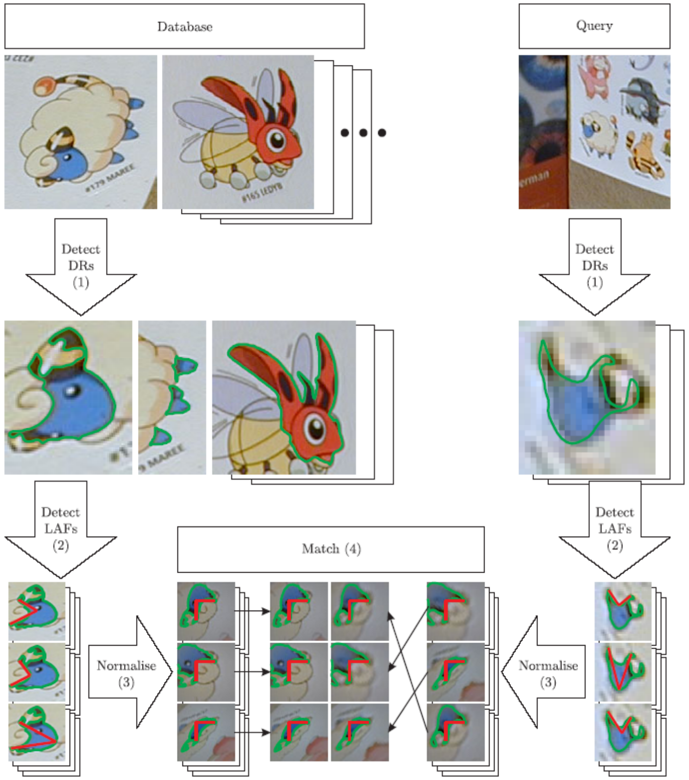
- How can we tell which putative matches are more reliable?
- Heuristic: compare distance of **nearest** neighbor to that of **second** nearest neighbor
  - Ratio will be high for features that are not distinctive
  - Threshold of 0.8 provides good separation



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

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

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.)

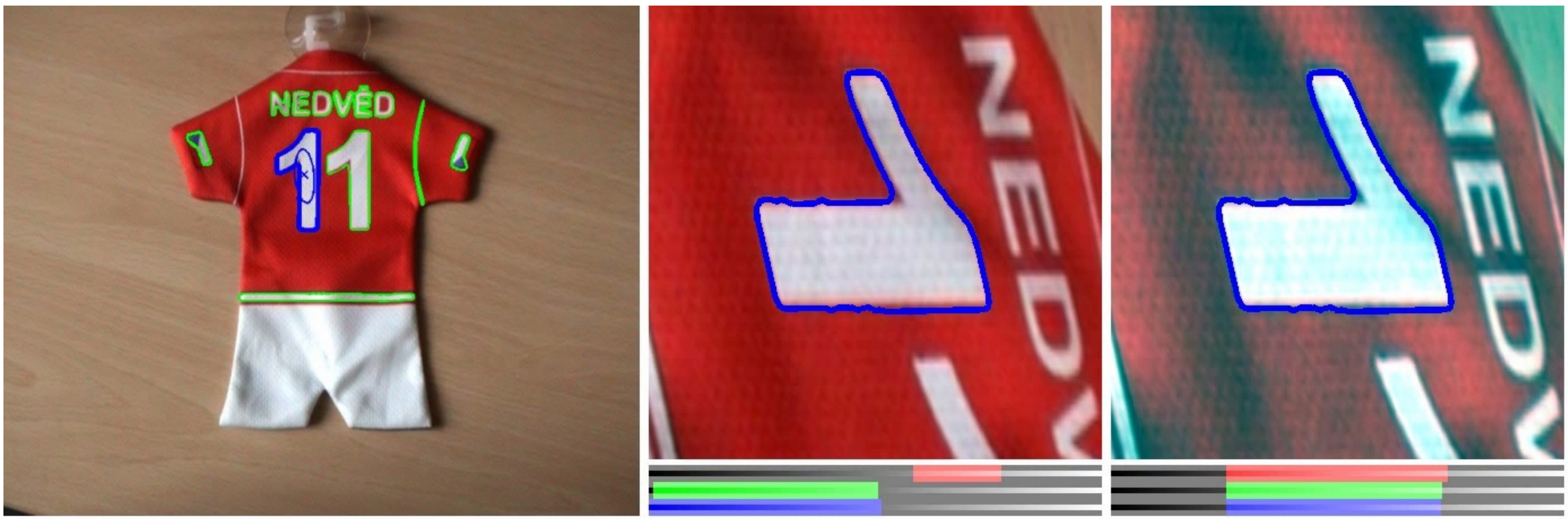


Matas, Chum, Urban, Pajdla: "Robust wide baseline stereo from maximally stable extremal regions". BMVC2002
Obdržálek and Matas: "Object recognition using local affine frames on distinguished regions". BMVC02
Obdržálek and Matas: "Sub-linear Indexing for Large Scale Object Recognition", BMVC 2005

MSER-LAF-Tree, Obdrzalek and Matas, 2005

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

[video-1, video-2]



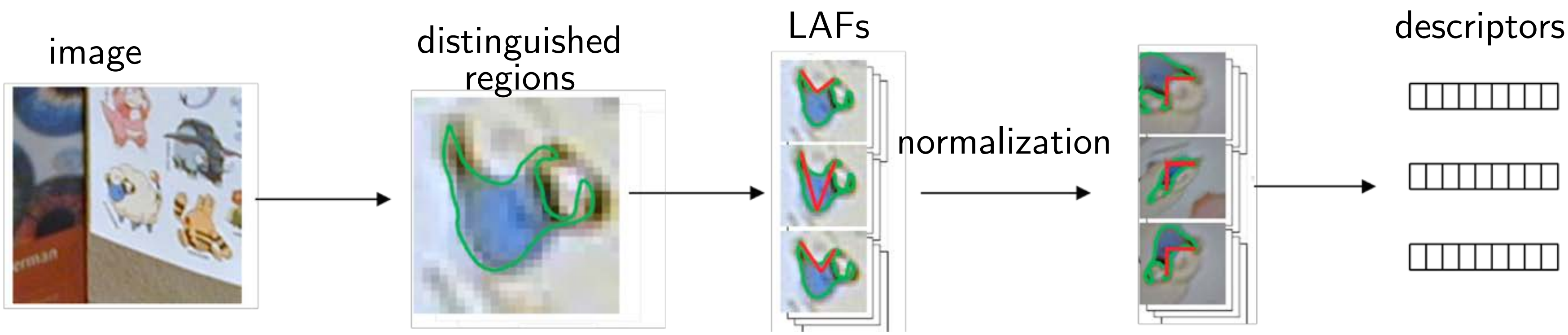
D. Nistér, H. Stewénius. *Scalable Recognition with a Vocabulary Tree, CVPR 2006* 300 citations

However:

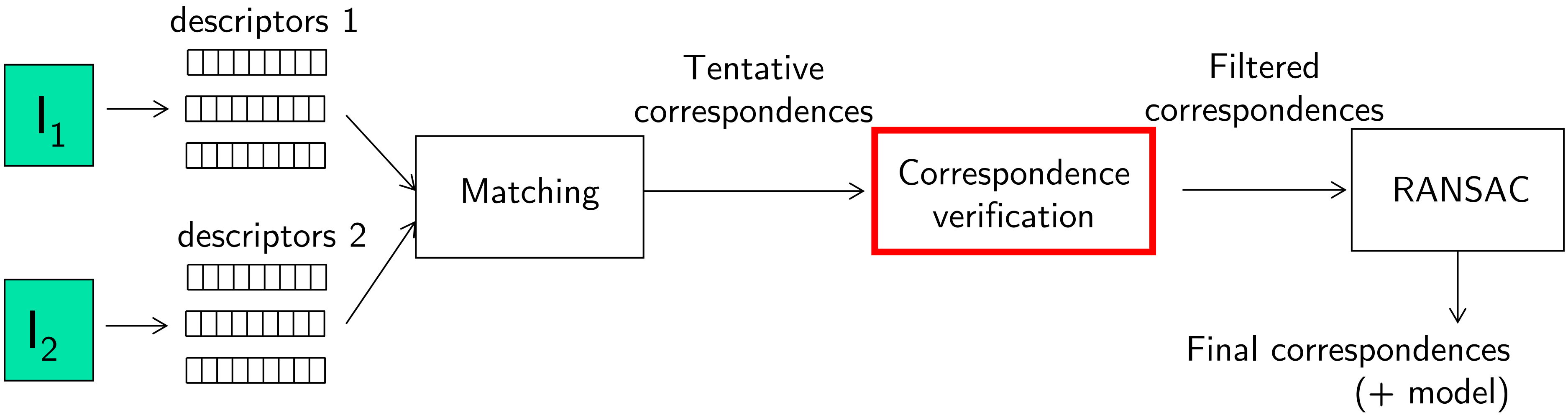
- Recognition of images, not objects
- Some of the object have no chance of being recognized via MSER+SIFT on different background

Correspondence Verification

- From image to local invariant descriptors



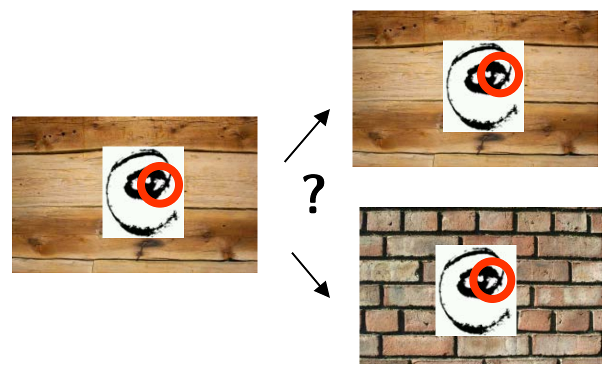
- Correspondence between two images



Correspondence Verification

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

measurement region too large



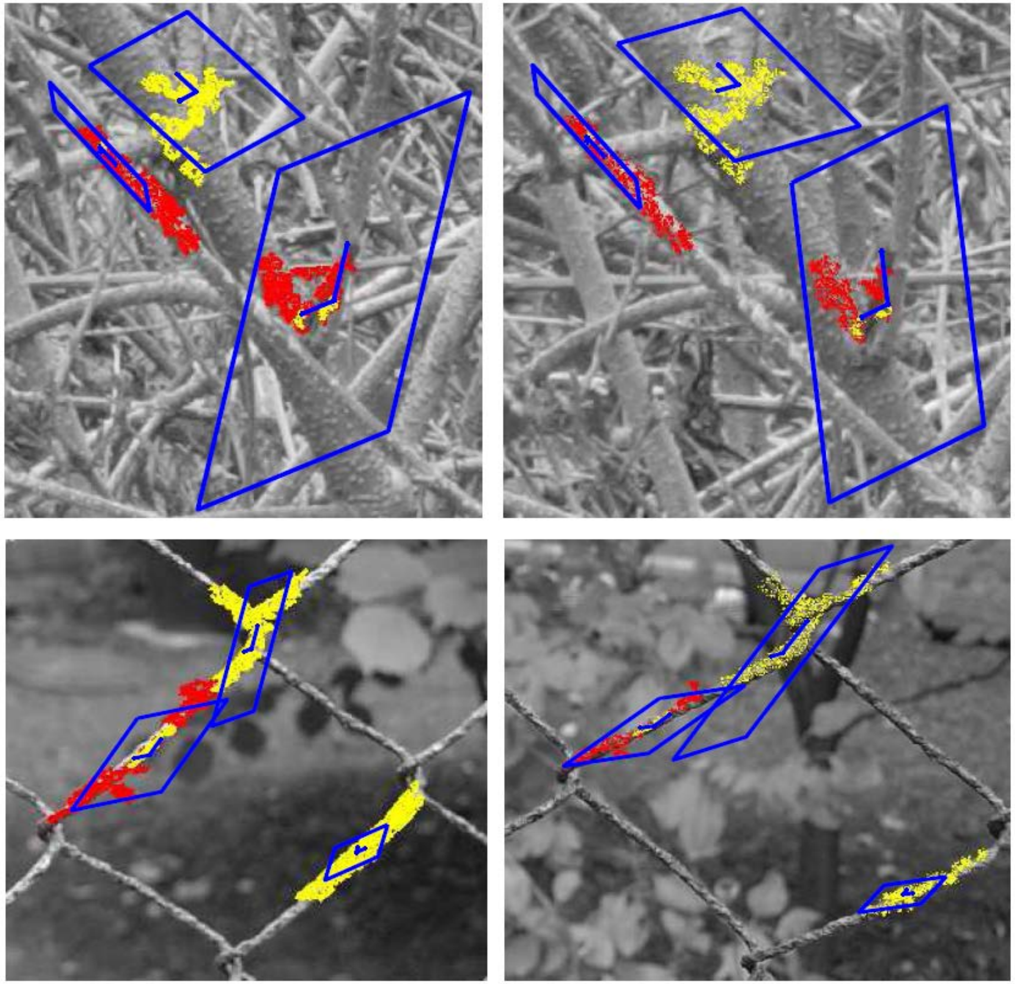
measurement region too small

Correspondence Verification

- Idea: "Look at both images simultaneously"

  => Sequential Correspondence Verification by Cosegmentaion

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

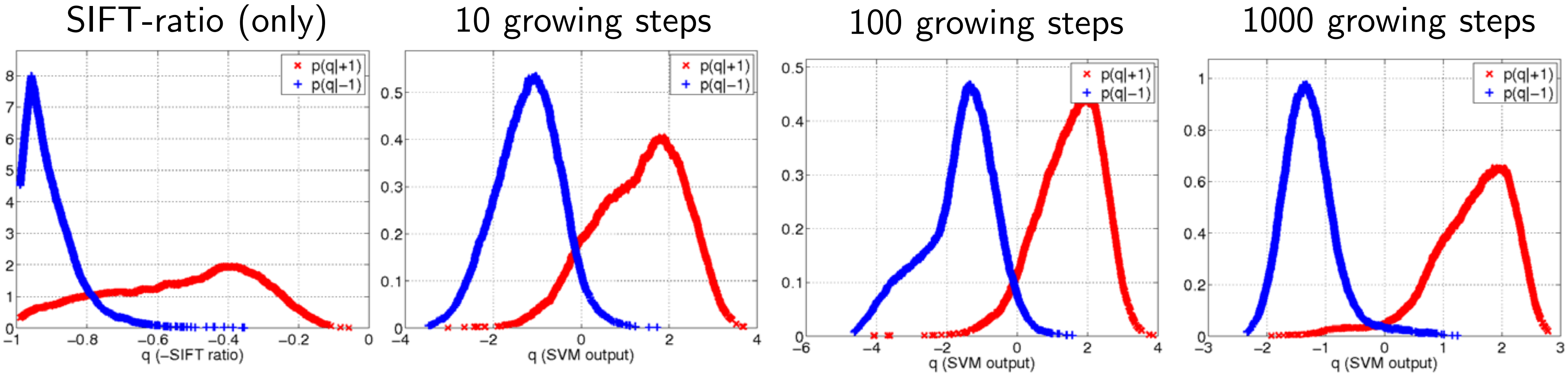


- **Input**: fixed number of tentative correspondences
- **Output**: Statistical Correspondence quality

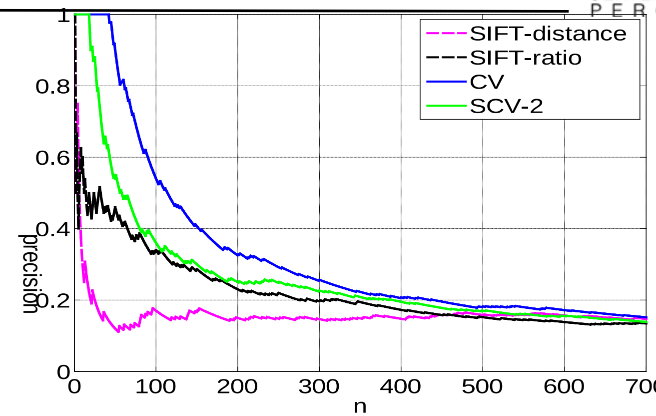
- A cosegmentaion process starts from LAF-correspondences to grow corresponding regions
- Various statistics are collected
- (Learned) Classifier to decide corresponding/non-correspond.

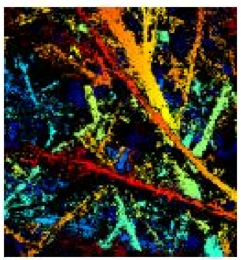
Correspondence Verification

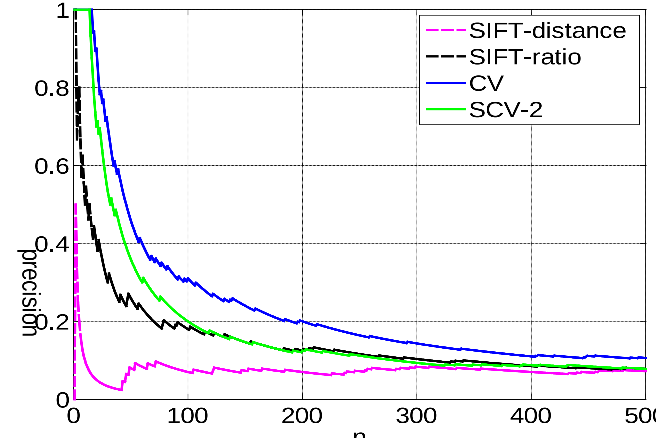
- Learning a (sequential) classifier
  - Training set from WBS images
  - 16k LAF correspondences (40 % correct)

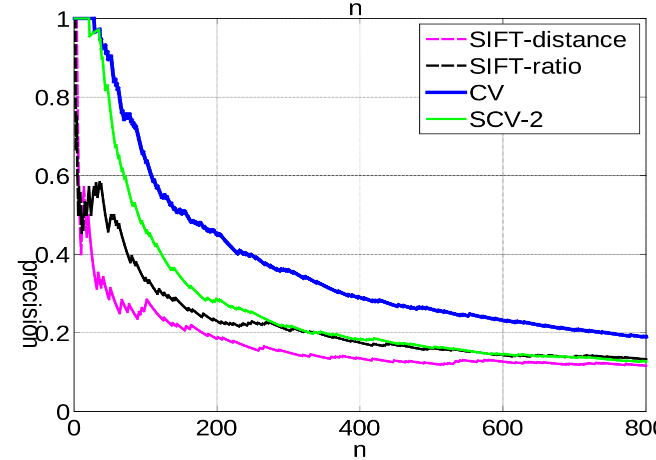


Correspondence Verification: Experiments

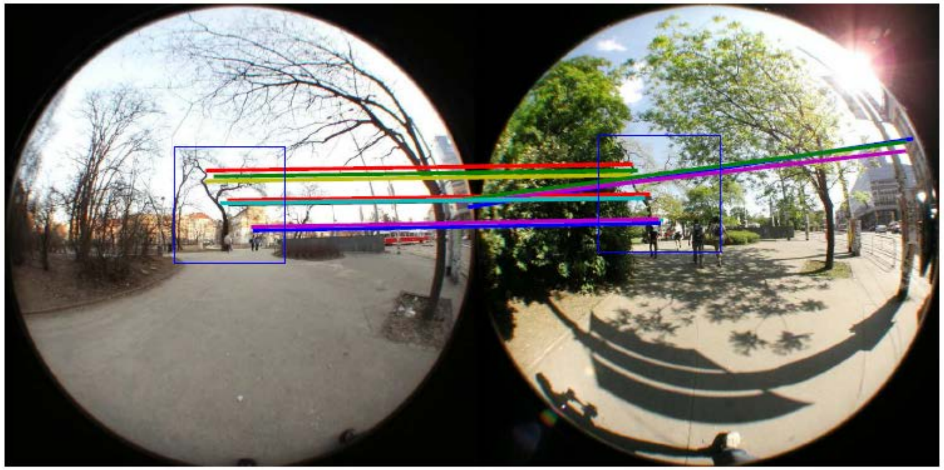


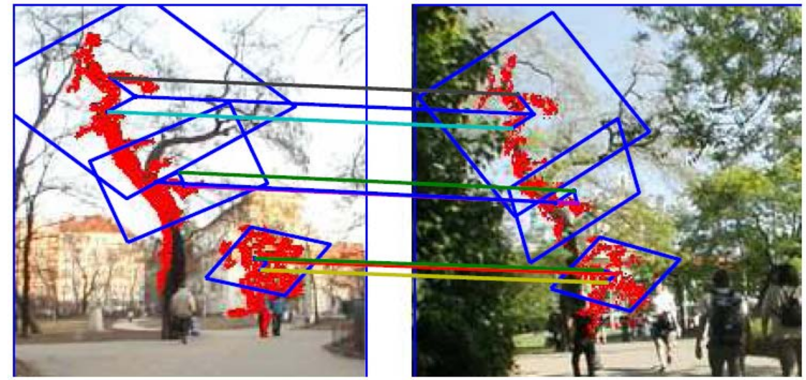






Correspondence Verification: Summary





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

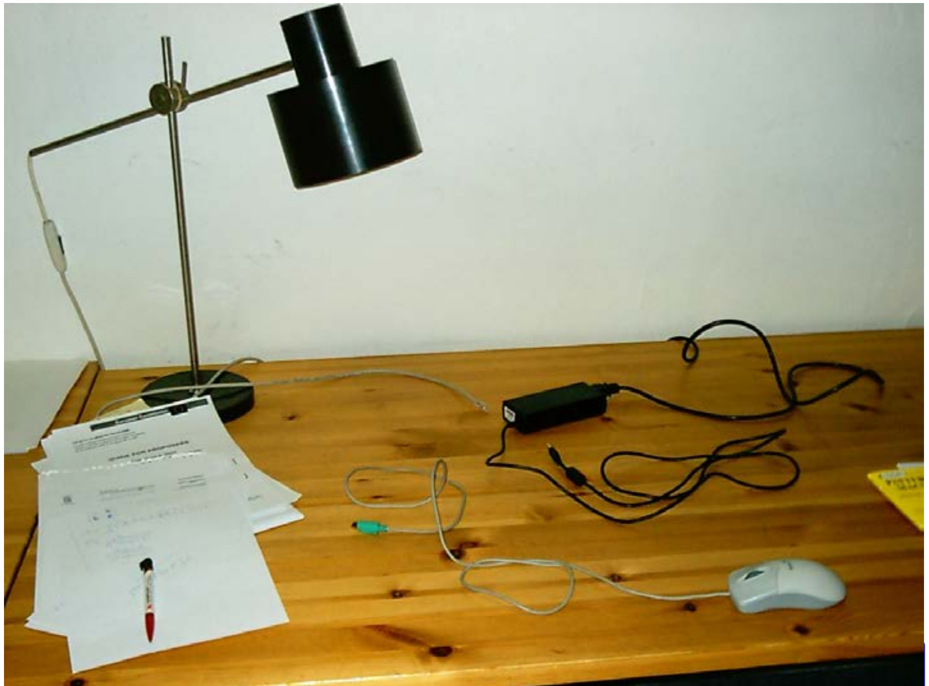
Local Feature Methods: Analysis

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).**

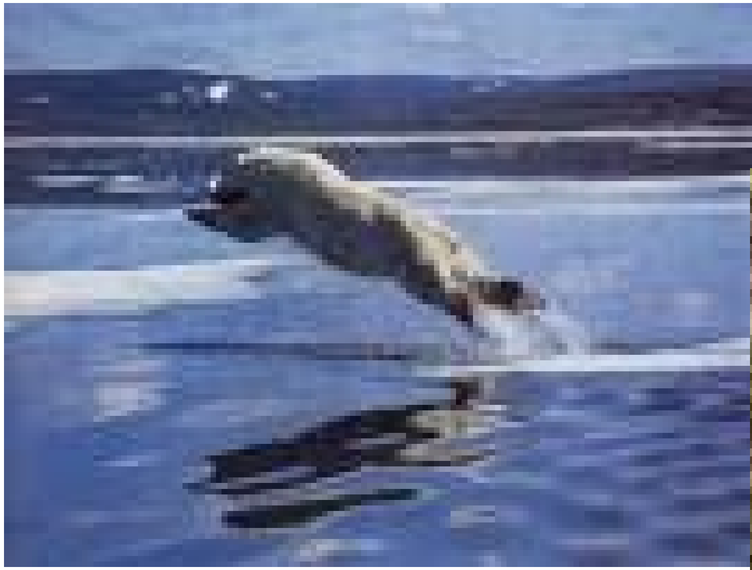
Where Local Features Fail:



Challenge: Elongated, "Wiry" and Flexible Objects

In this case: "no recognition without segmentation"?

Where Local Features Fail:



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

macros.tex
sfmath.sty
cmpitemize.tex

Thank you for your attention.