CS294-43: Visual Object and Activity Recognition

Prof. Trevor Darrell

Feb 10th: Local Features

Today

- Scale selection [Lindeberg]
- Affine-invariance [Mikolajczyk and Schmid]
- MSER – Stable Regions [Matas et al.]
- SURF -Fast Approximate SIFT [Bay et al.]
- Spatio-Temporal Features [Laptev]
- Self-Similarilty [Shectman and Irani]

- Bonus: Temporal Self-Similarity [Laptev ECCV'08]

Local Invariant Features: What? Why? When? How?

Tinne Tuytelaars

Tutorial ECCV 2006

May 7th, 2006

Overview

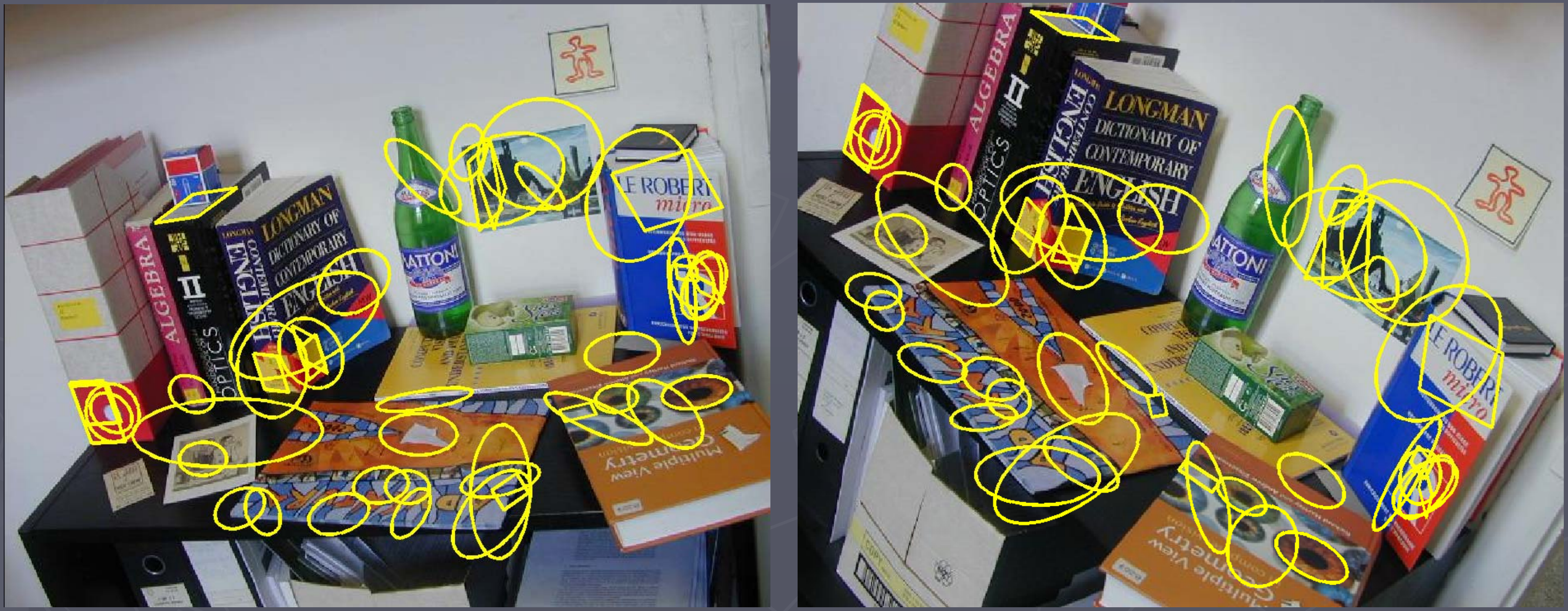
- Local Invariant Features: What? Why?
  - Introduction
  - Overview of existing detectors
  - Quantitative and qualitative comparison
- Local Invariant Features: When? How?
  - Feature descriptors
  - Applications
  - Conclusions

Overview

- ►Local Invariant Features: What? Why?
  - Introduction
  - Overview of existing detectors
  - Quantitative and qualitative comparison
- ►Local Invariant Features: When? How?
  - Feature descriptors
  - Applications
  - Conclusions

Introduction

- Wide baseline matching



Introduction

- Recognition of specific objects

Rothganger et al. '03

Lowe et al. '02

Ferrari et al. '04

Introduction

- Object class recognition

So what’s the novelty?

- ~~Local character~~

History

- History of interest point detectors goes a long way back...
  - Corner detectors
  - Blob detectors
  - Edgel detectors

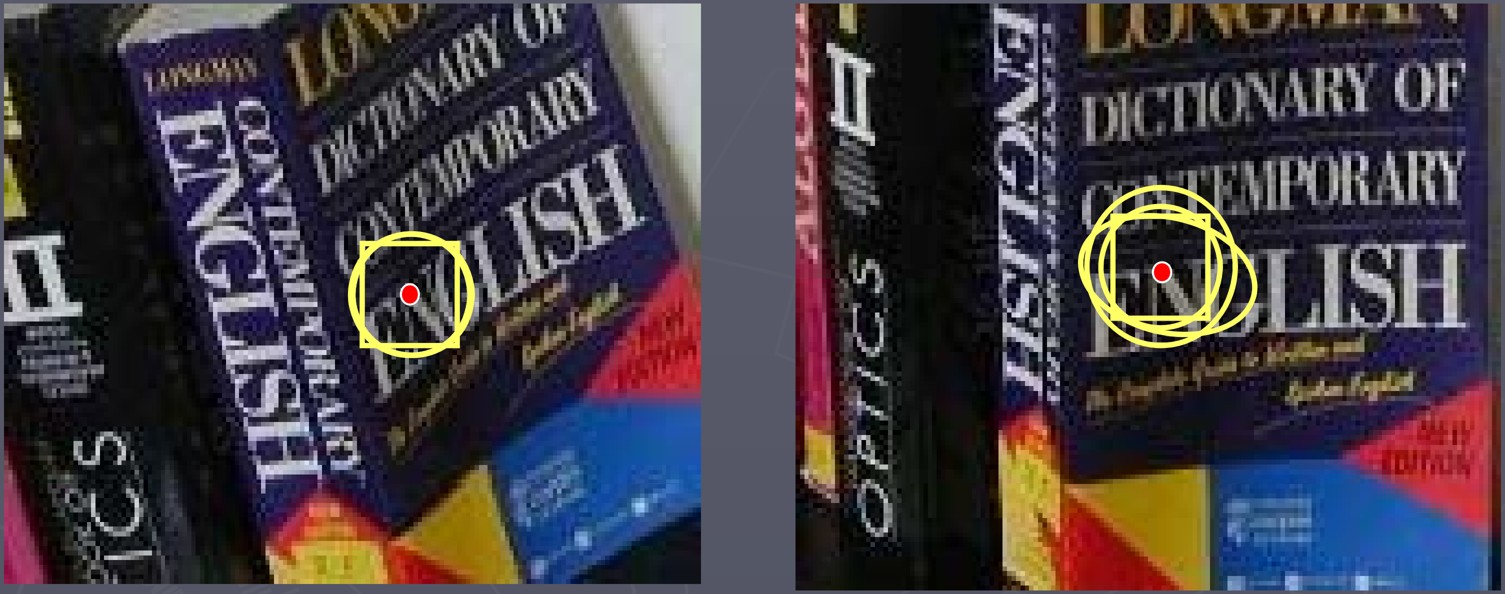
So what’s the novelty?

- ~~Local character~~
- Level of invariance
- Local invariant features: a new paradigm
  - Not just a method to select interesting locations in the image, or to speed up analysis
  - But rather a new image representation, that allows to describe the objects / parts without the need for segmentation

Properties of the ideal feature

- Local: features are local, so robust to occlusion and clutter (no prior segmentation)
- Invariant (or covariant)
- Robust: noise, blur, discretization, compression, etc. do not have a big impact on the feature
- Distinctive: individual features can be matched to a large database of objects
- Quantity: many features can be generated for even small objects
- Accurate: precise localization
- Efficient: close to real-time performance

The need for invariance



Terminology: Invariant or Covariant?

When a transformation is applied to an image,

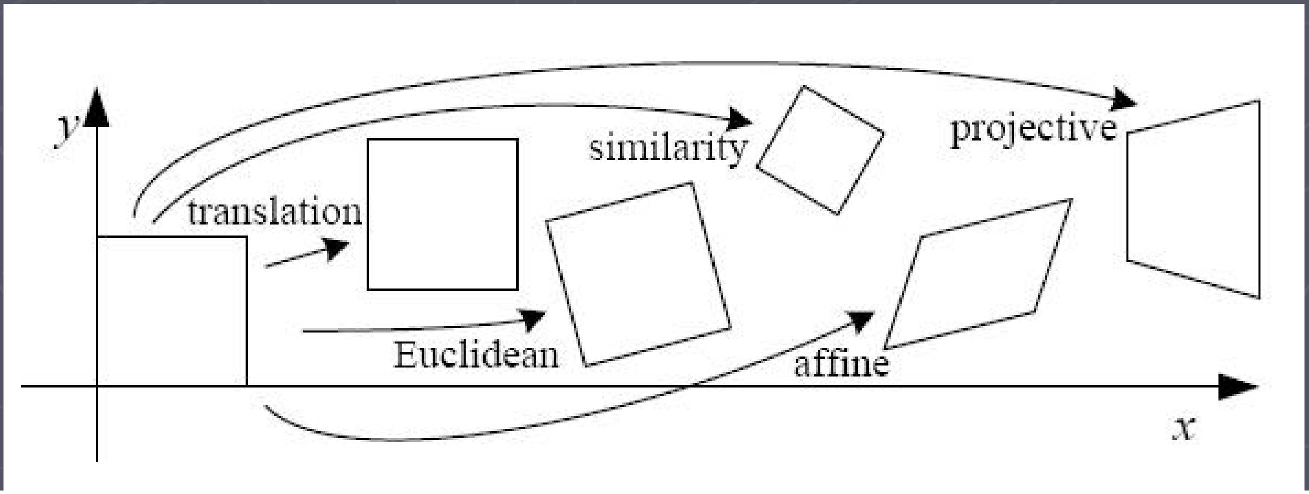
- an invariant measure remains unchanged.
- a covariant measure changes in a way consistent with the image transformation.

Terminology: 'detector' or 'extractor'

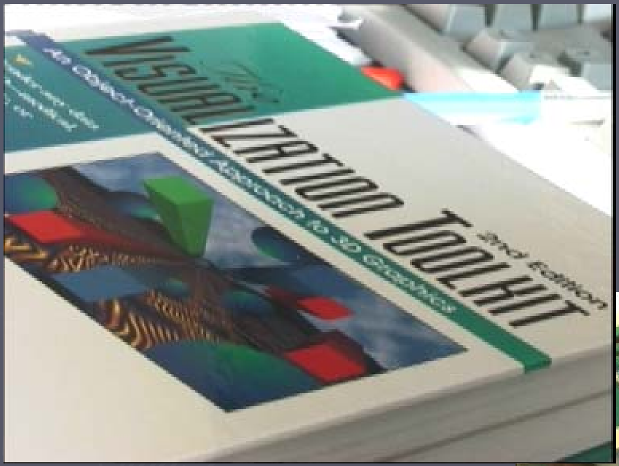
Geometric transformations

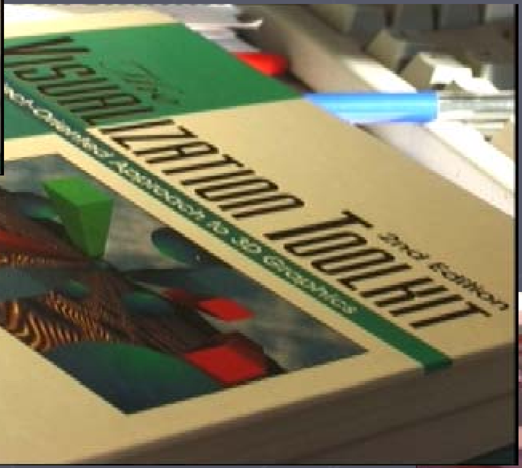
- Translation
- Euclidean (translation + rotation)
- Similarity (transl. + rotation + scale)
- Affine transformations
- Projective transformations

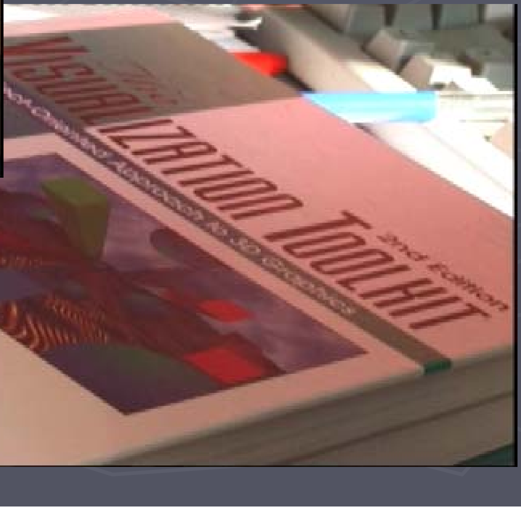
For planar patches:



Photometric transformations







Modelled as a linear transformation:

scaling + offset

Disturbances

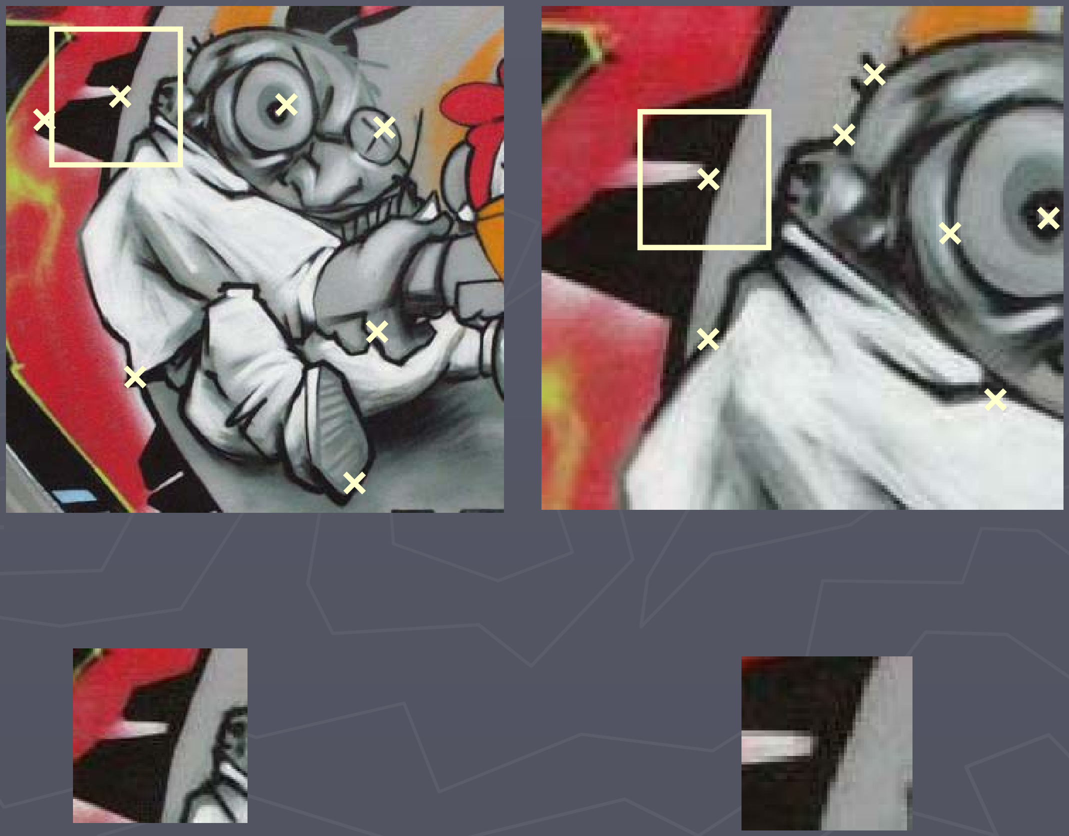
- Noise
- Image blur
- Discretization errors
- Compression artefacts
- Deviations from the mathematical model (non-linearities, non-planarities, etc.)
- Intra-class variations

How to cope with transformations?

- Exhaustive search
- Invariance
- Robustness

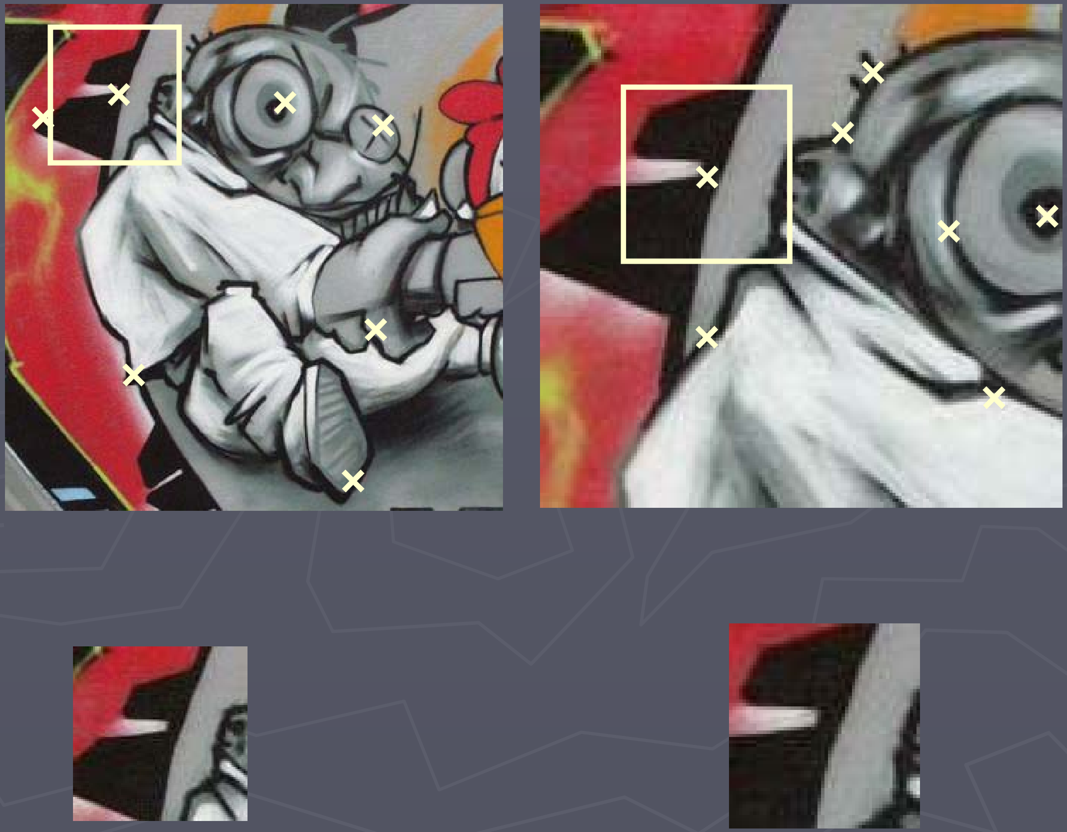
Exhaustive search

- Multi-scale approach



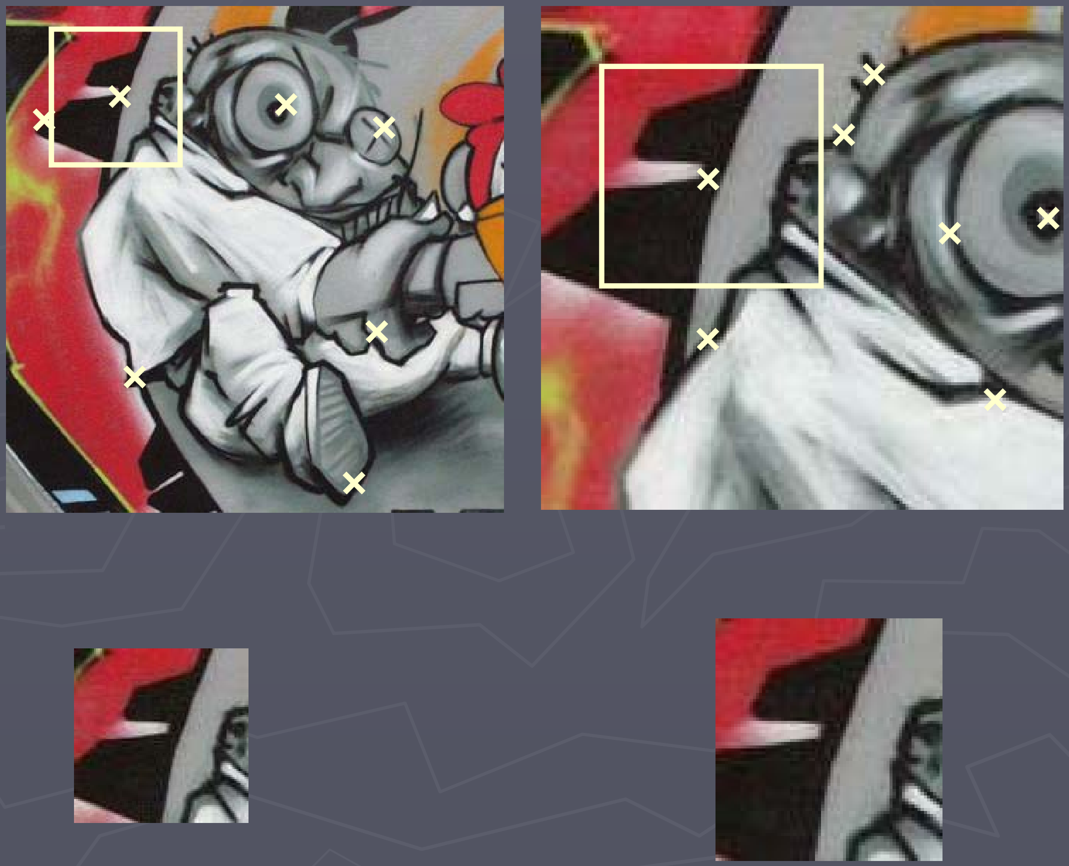
Exhaustive search

- Multi-scale approach



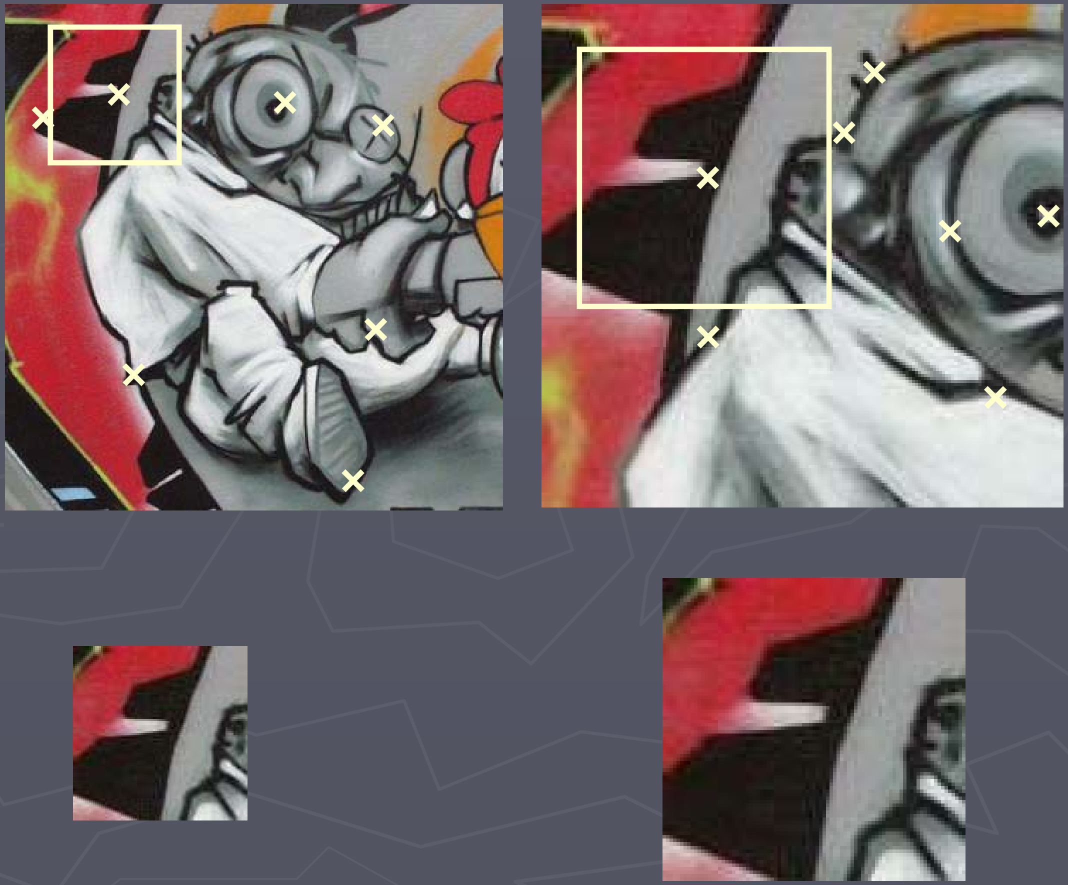
Exhaustive search

- Multi-scale approach



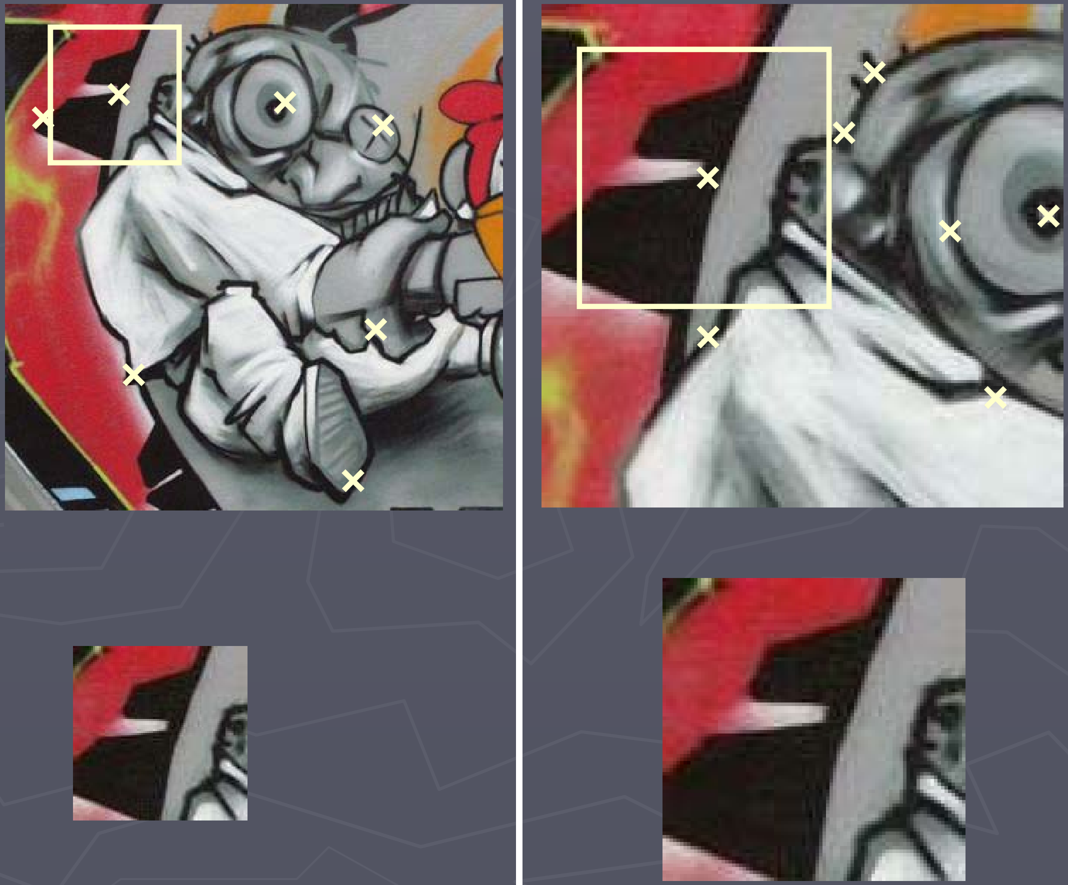
Exhaustive search

- Multi-scale approach



Invariance

- Extract patch from each image individually



Invariance

- Integration, e.g.
  - moment invariants, ...
- Heuristics, e.g.
  - Difference of intensity values for photom. offset
  - Ratio of intensity values for photom. scalefactor
- Selection and normalization, e.g.
  - Automatic scale selection (Lindeberg et al., 1996)
  - Orientation assignment
  - Affine normalization (‘deskewing’)
- ...

International Journal of Computer Vision 30(2), 79–116 (1998)

Feature Detection with Automatic Scale Selection

TONY LINDEBERG
Computational Vision and Active Perception Laboratory (CVAP), Department of Numerical Analysis and Computing Science, KTH (Royal Institute of Technology), S-100 44 Stockholm, Sweden
tony@nada.kth.se

Received February 1, 1994; Revised June 1, 1996; Accepted July 30, 1998

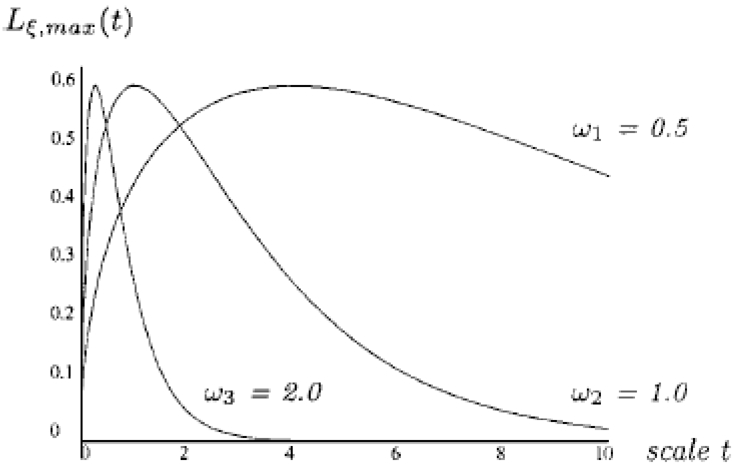


Figure 1. The amplitude of first order normalized derivatives as function of scale for sinusoidal input signals of different frequency ($\omega_1 = 0.5$, $\omega_2 = 1.0$ and $\omega_3 = 2.0$).

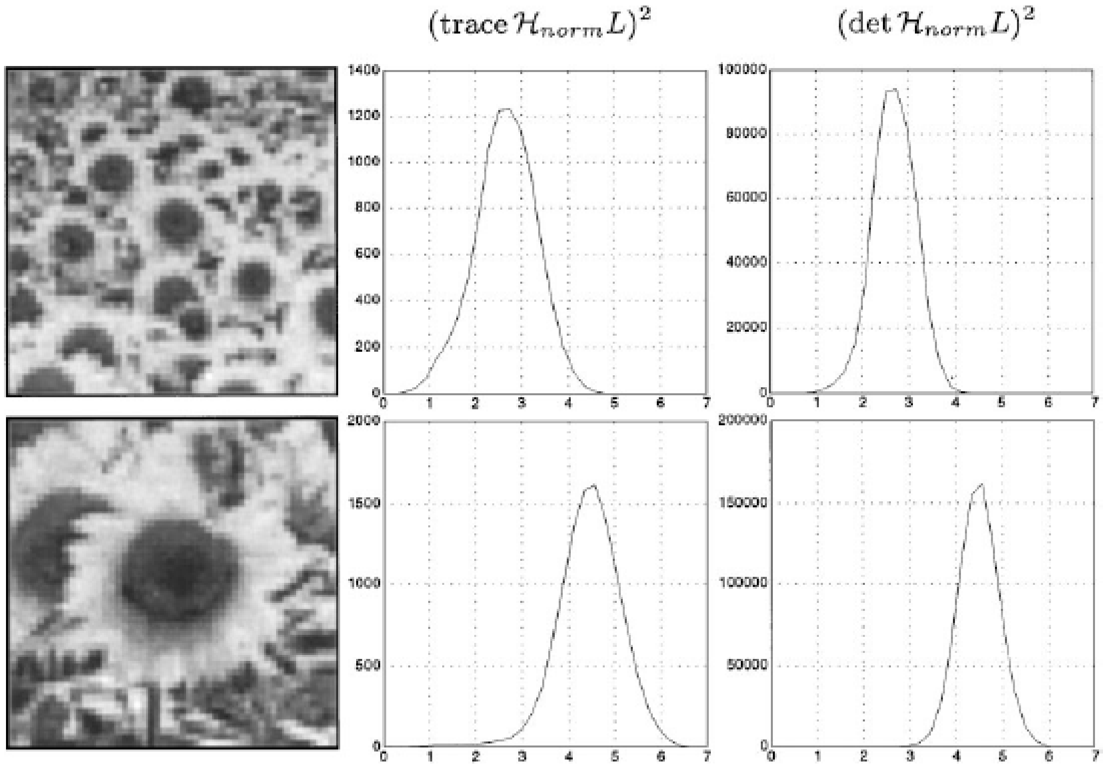


Figure 2. Scale-space signatures of the trace and the determinant of the normalized Hessian matrix computed for two details of a sunflower image; (left) grey-level image, (middle) signature of (trace $\mathcal{H}_{norm}L)^2$, (right) signature of (det $\mathcal{H}_{norm}L)^2$. (The signatures have been computed at the central point in each image. The horizontal axis shows effective scale, essentially the logarithm of the scale parameter, whereas the scaling of the vertical axis is linear in the normalized operator response.)

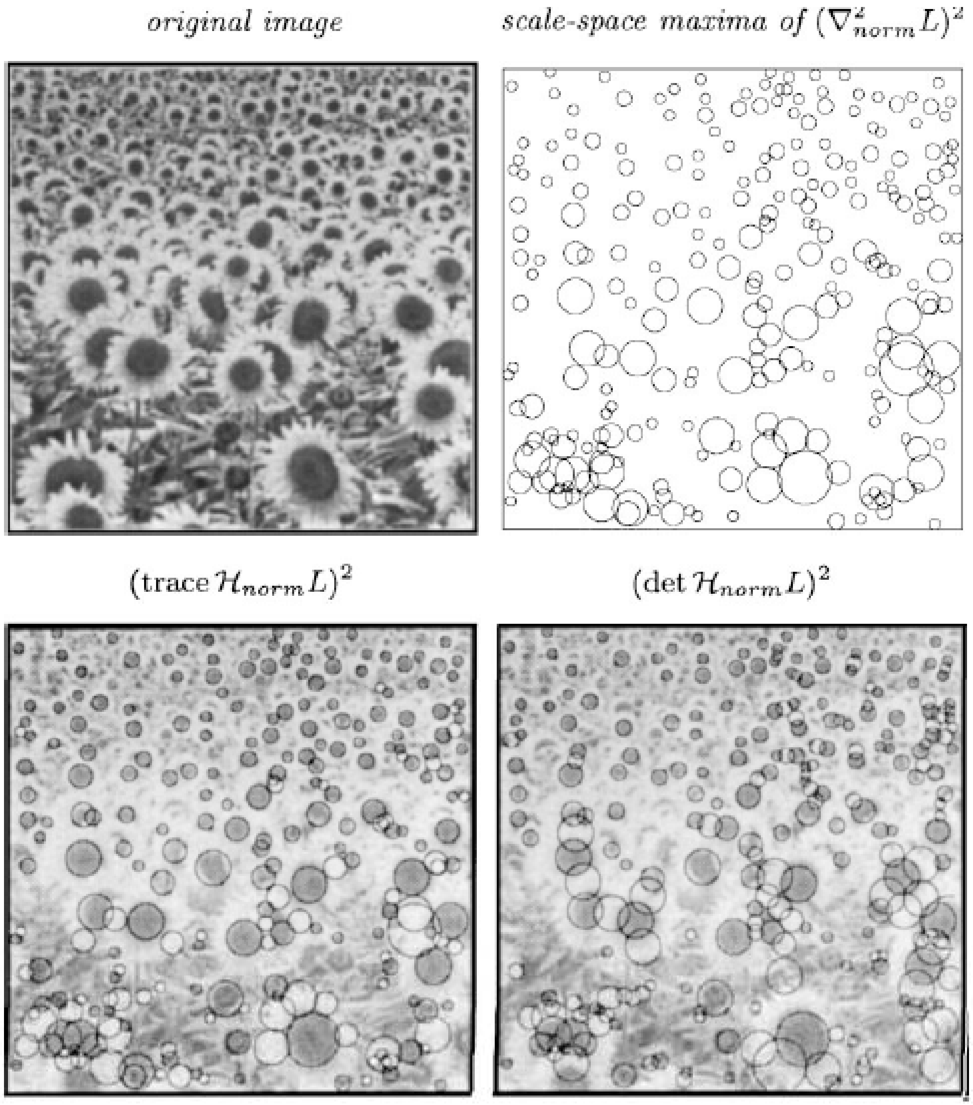
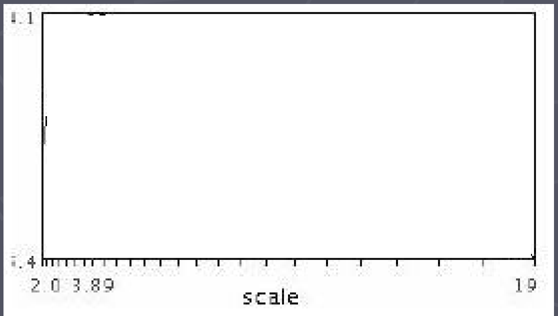

original image
scale-space maxima of $(\nabla^2_{norm} L)^2$
$(\mathrm{trace}\, \mathcal{H}_{norm} L)^2$
$(\det \mathcal{H}_{norm} L)^2$

Figure 3. Normalized scale-space maxima computed from an image of a sunflower field: (top left): Original image. (top right): Circles representing the 250 normalized scale-space maxima of $(\mathrm{trace}\, \mathcal{H}_{norm} L)^2$ having the strongest normalized response. (bottom left): Circles representing scale-space maxima of $(\mathrm{trace}\, \mathcal{H}_{norm} L)^2$ superimposed onto a bright copy of the original image. (bottom right): Corresponding results for scale-space maxima of $(\det \mathcal{H}_{norm} L)^2$.

Automatic scale selection

Lindeberg et al., 199

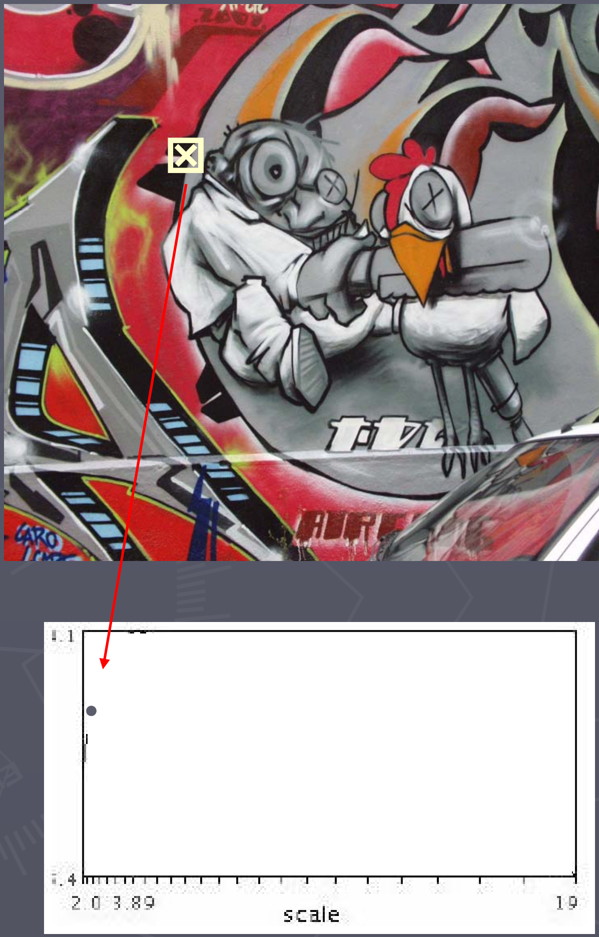


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection

Function responses for increasing scale

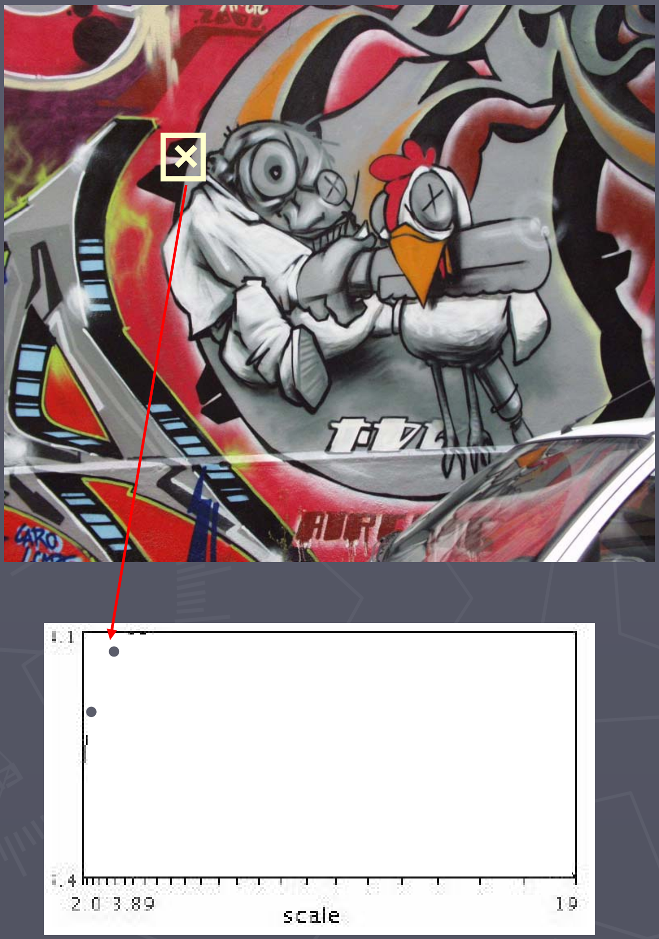
Scale trace (signature)



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

Automatic scale selection

Function responses for increasing scale
scale trace (signature)

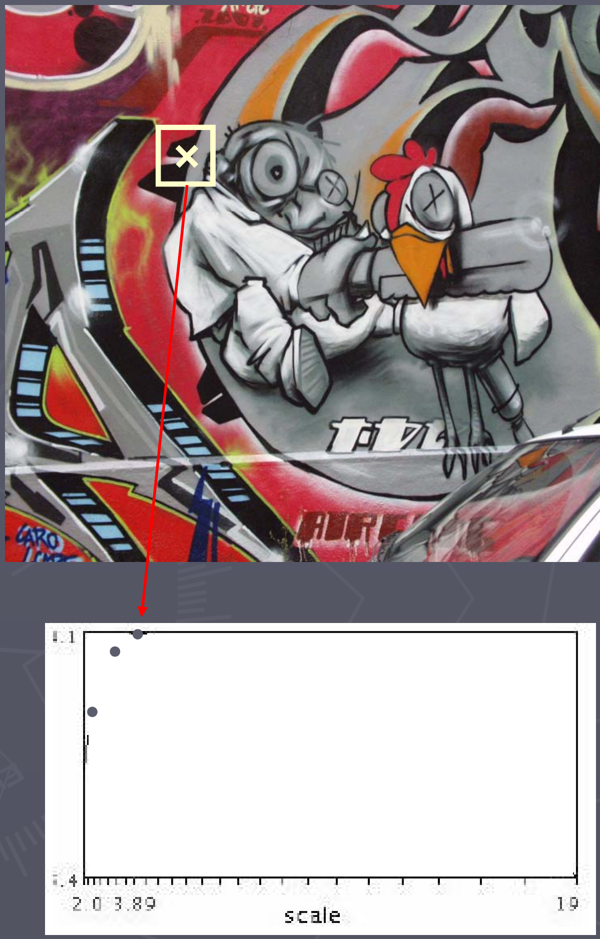


$f(I_{i_1 \ldots i_m}(x, \sigma))$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

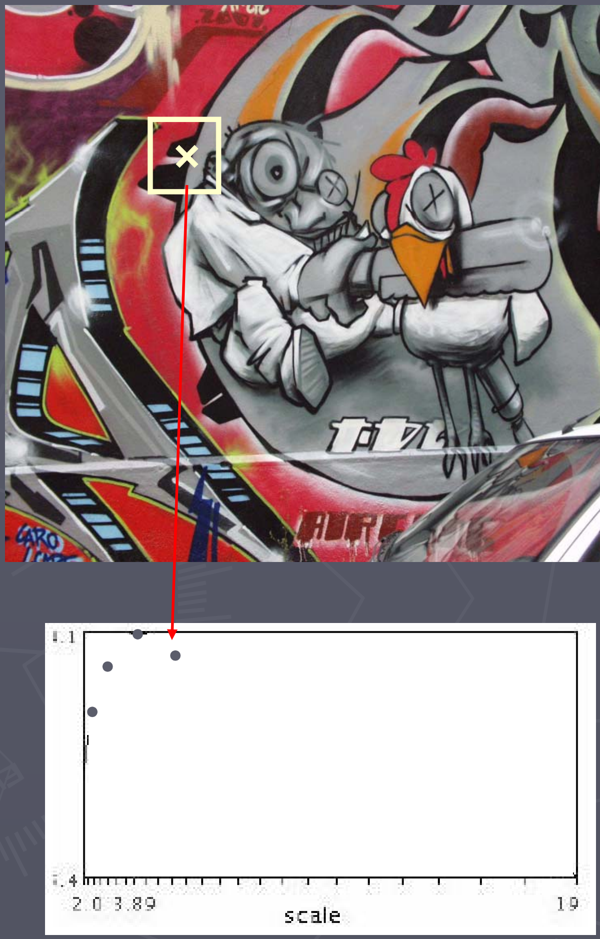


$f(I_{i_1 \ldots i_m}(x, \sigma))$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

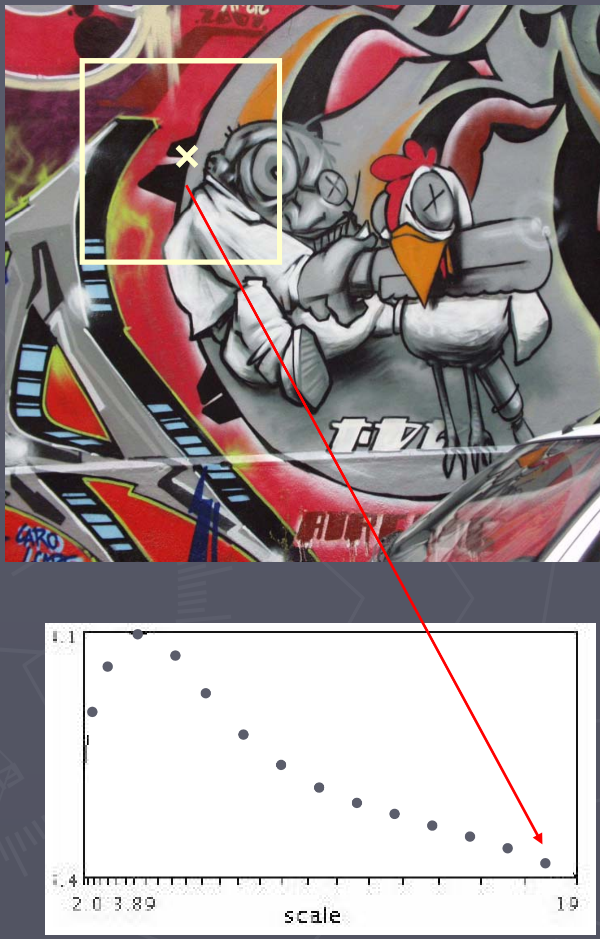


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

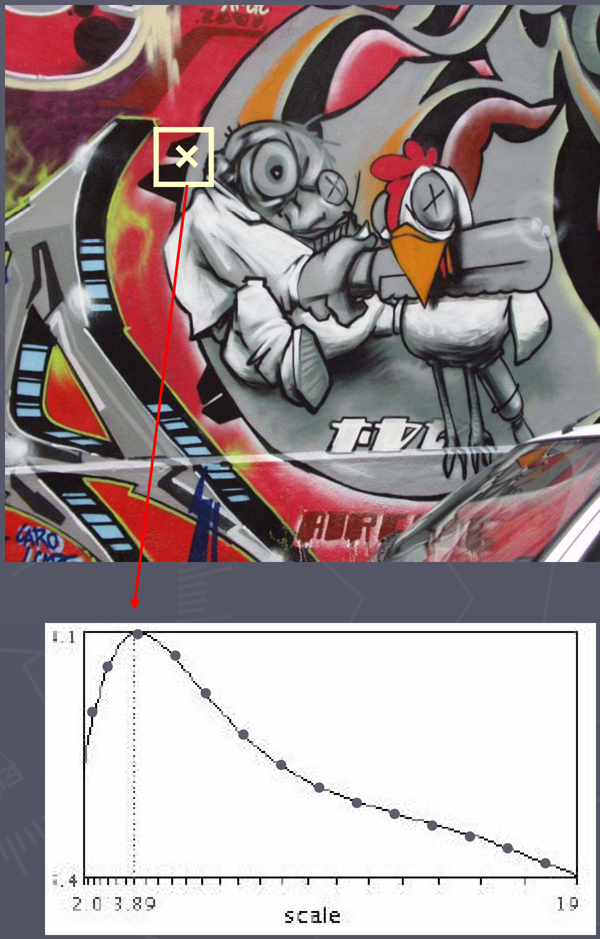


$f(I_{i_1 \ldots i_m}(x, \sigma))$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

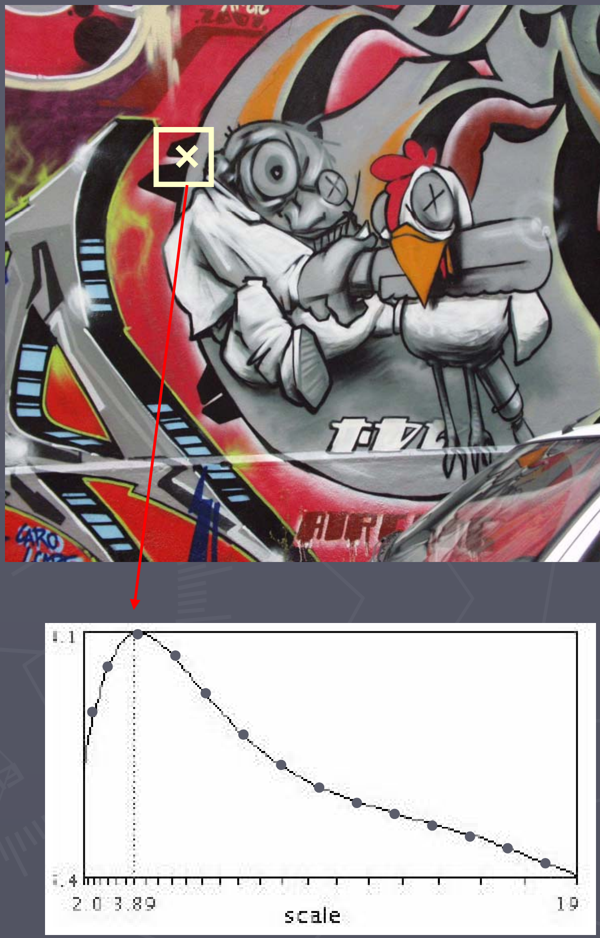


$f(I_{i_1 \ldots i_m}(x,\sigma))$

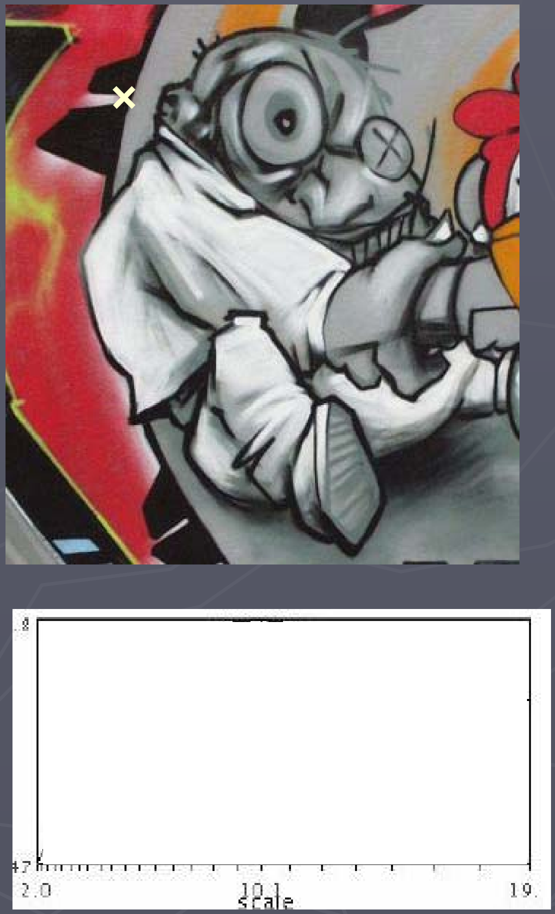
Automatic scale selection

Function responses for increasing scale

Scale trace (signature)



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

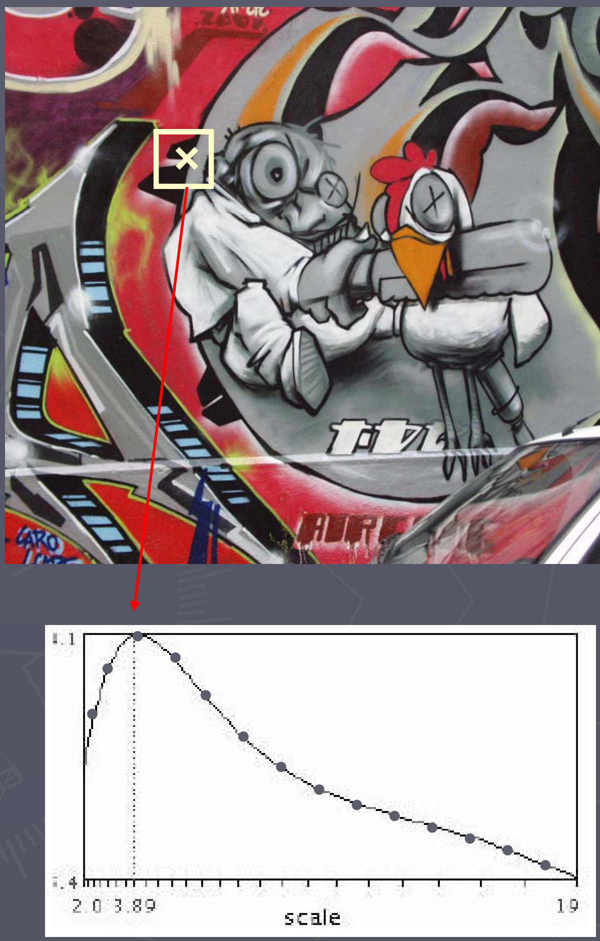


$f(I_{i_1 \ldots i_m}(x',\sigma))$

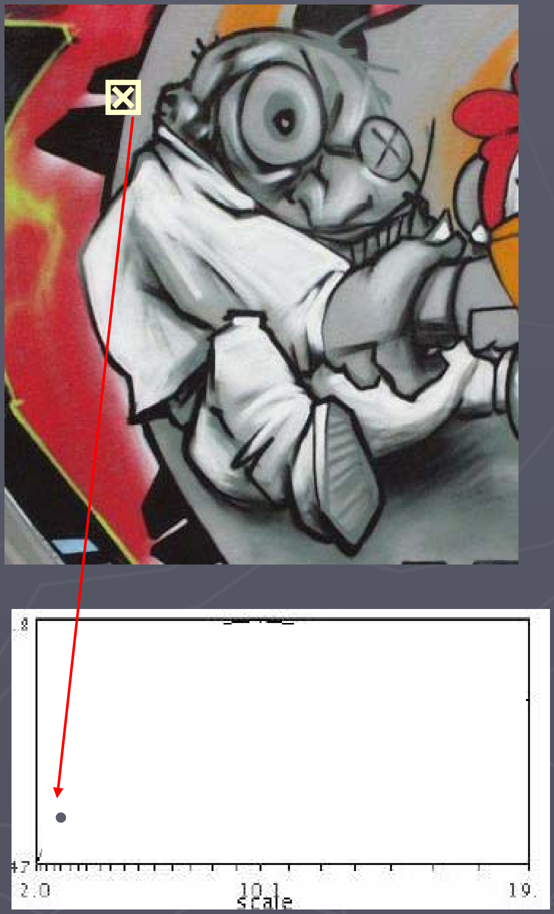
Automatic scale selection

Function responses for increasing scale

Scale trace (signature)



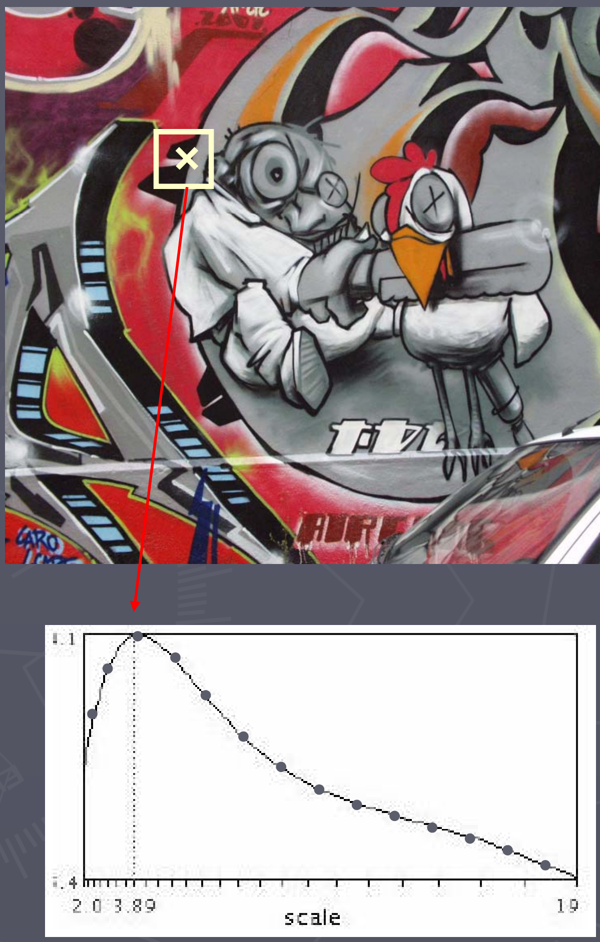
$f(I_{i_1 \ldots i_m}(x,\sigma))$



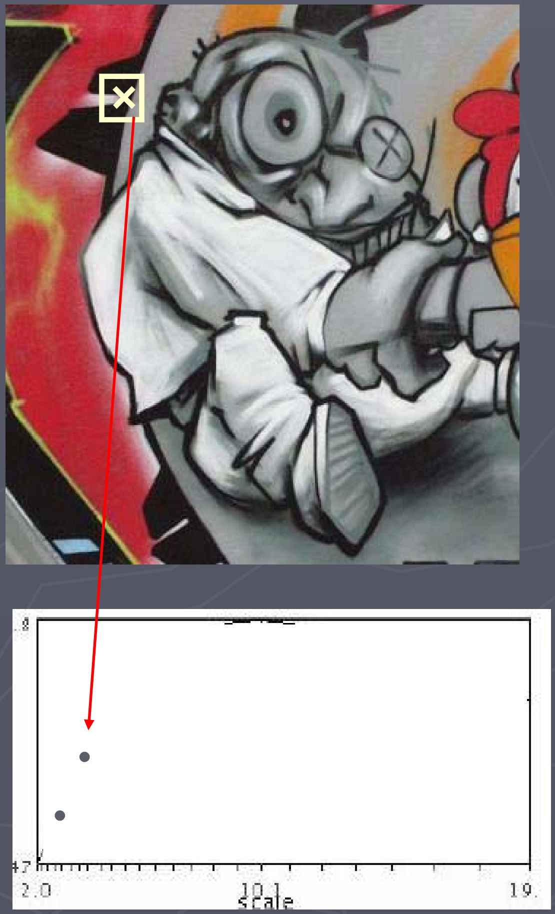
$f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)



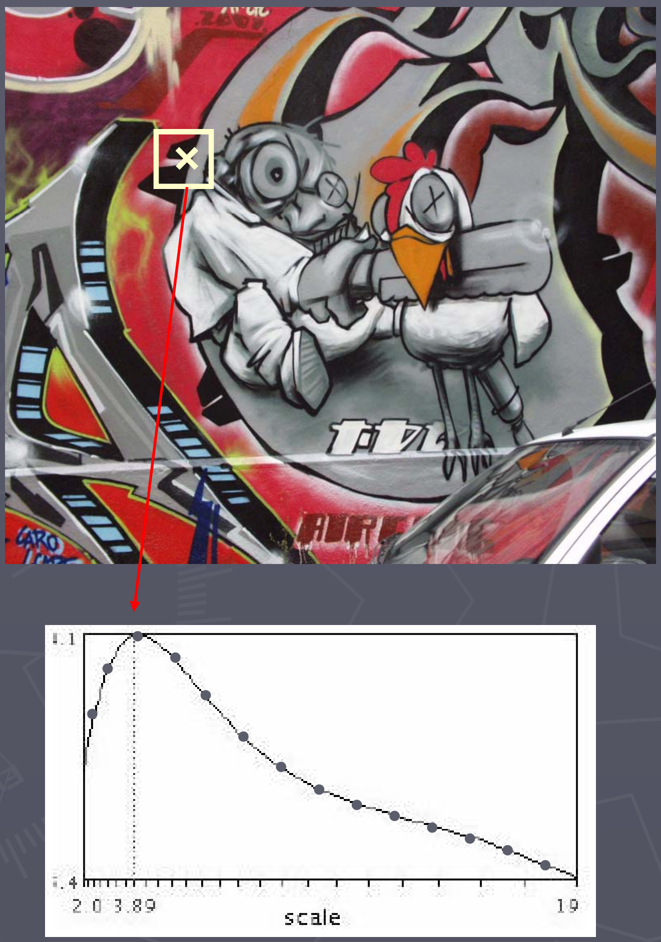
$$f(I_{i_1\ldots i_m}(x,\sigma))$$



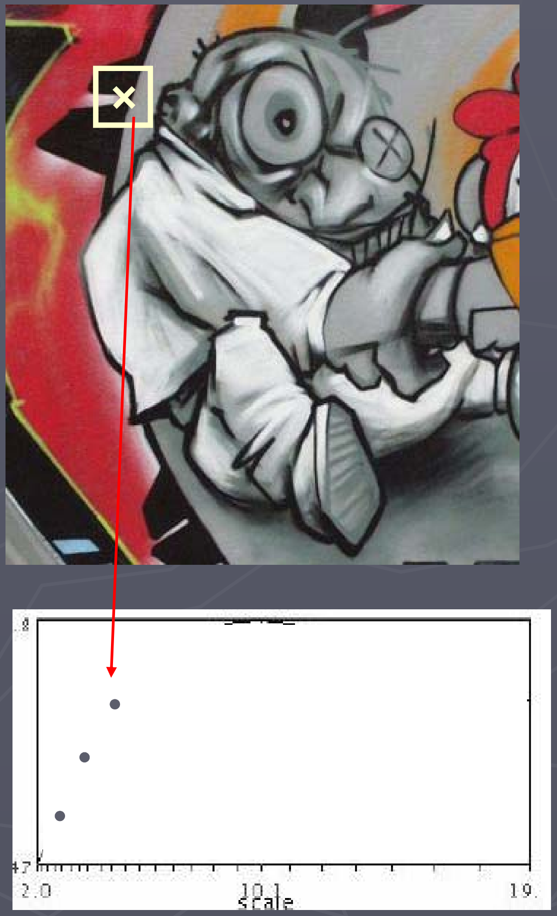
$$f(I_{i_1\ldots i_m}(x',\sigma))$$

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)



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

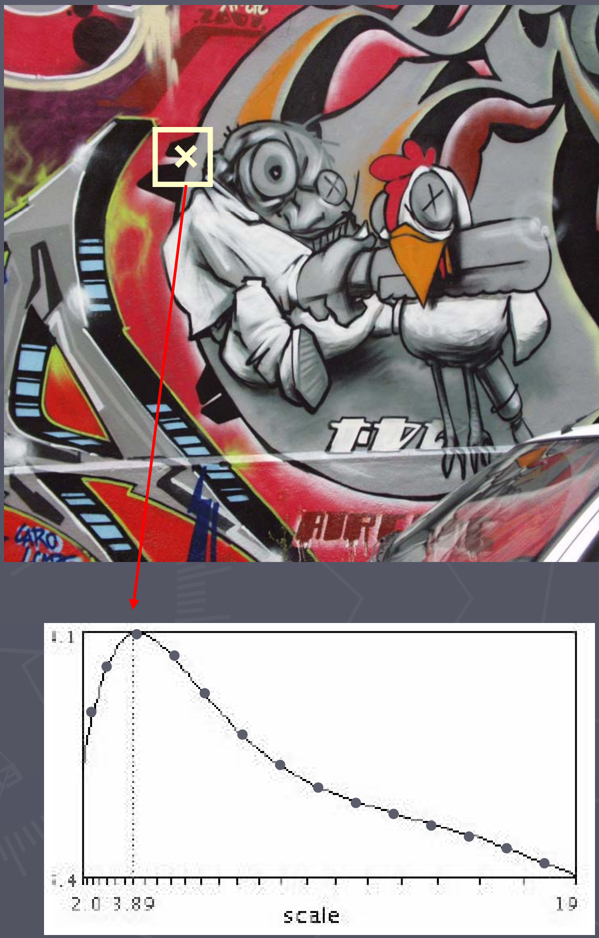


$f(I_{i_1 \ldots i_m}(x', \sigma))$

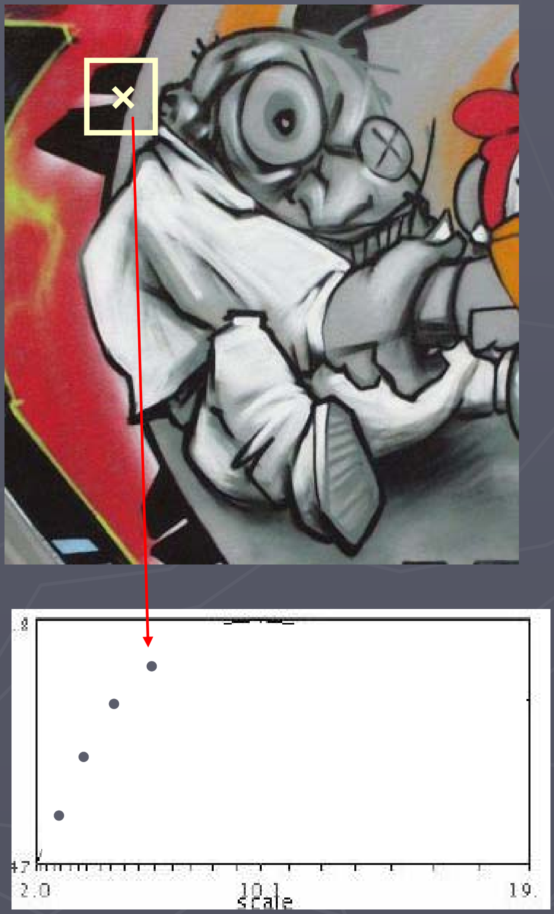
Automatic scale selection

Function responses for increasing scale

scale trace (signature)



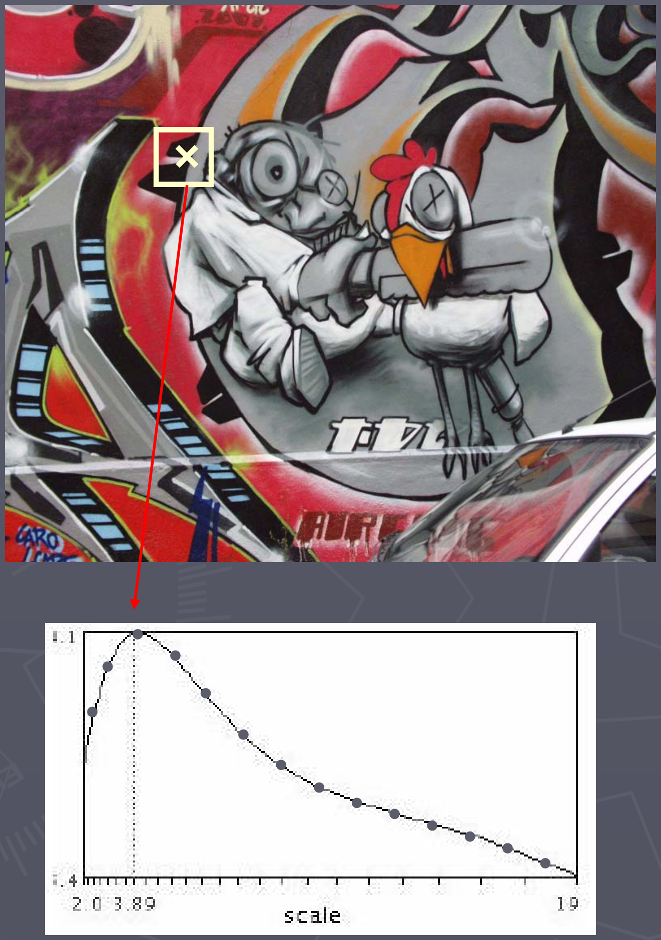
$f(I_{i_1 \ldots i_m}(x,\sigma))$



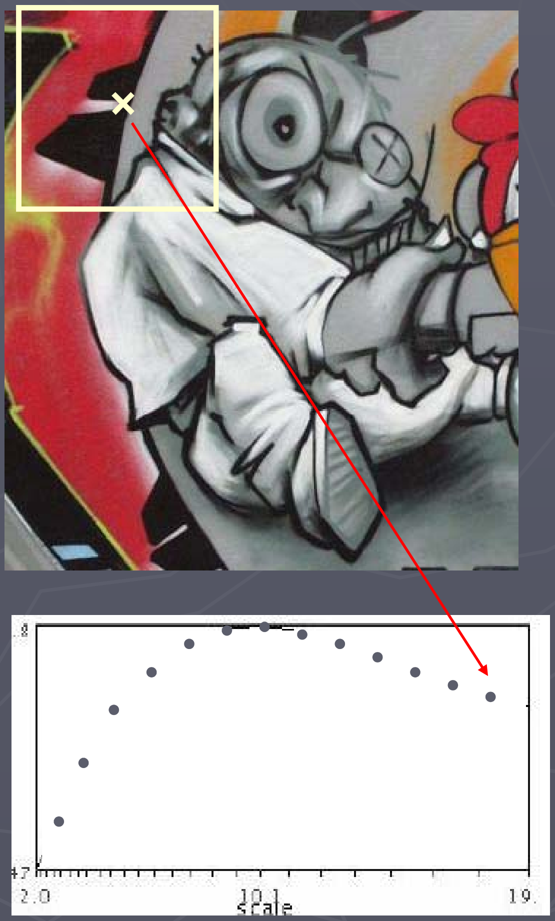
$f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)



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

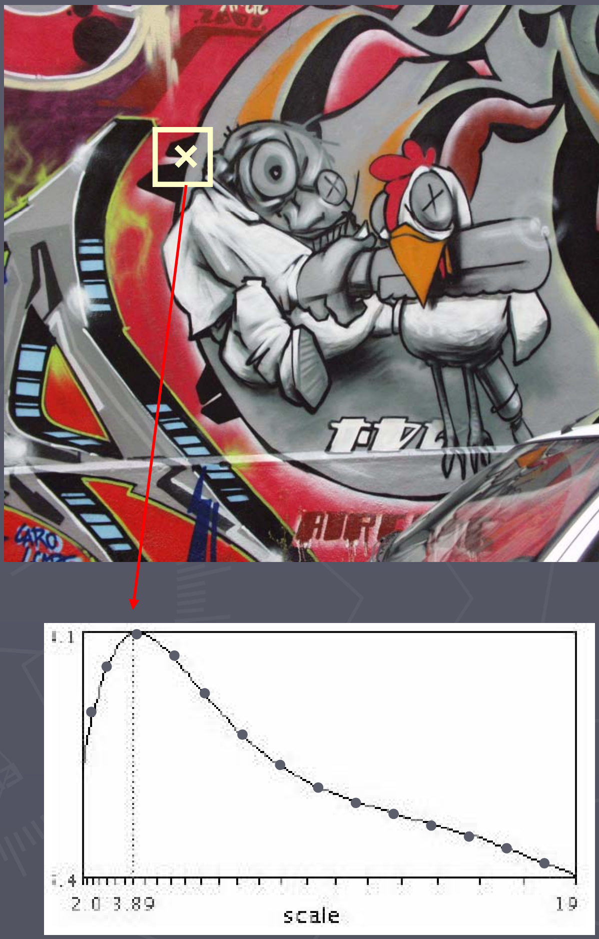


$f(I_{i_1 \ldots i_m}(x', \sigma))$

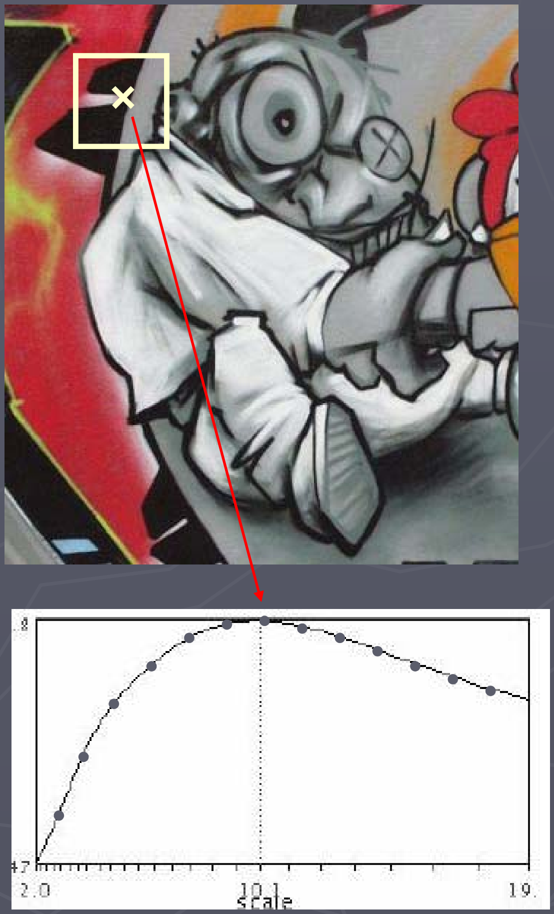
Automatic scale selection

Function responses for increasing scale

Scale trace (signature)



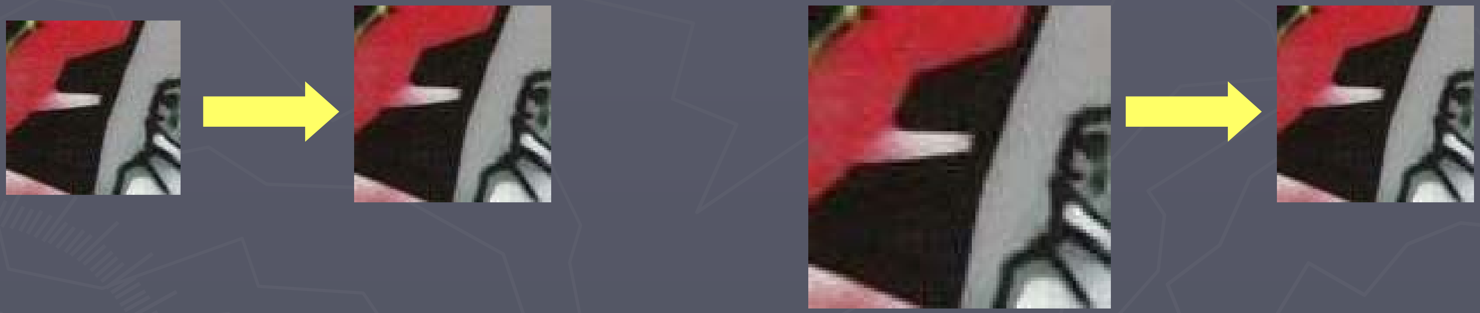
$f(I_{i_1 \ldots i_m}(x, \sigma))$



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

Automatic scale selection

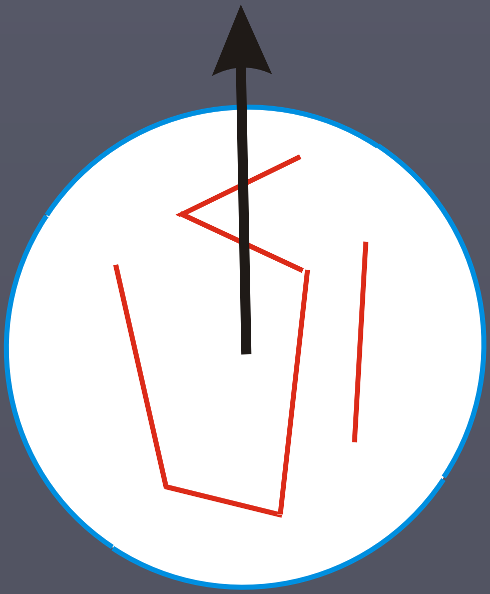
- Normalize: rescale to fixed size

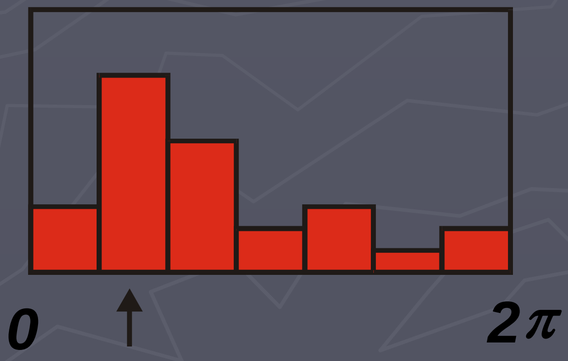


Orientation assignment

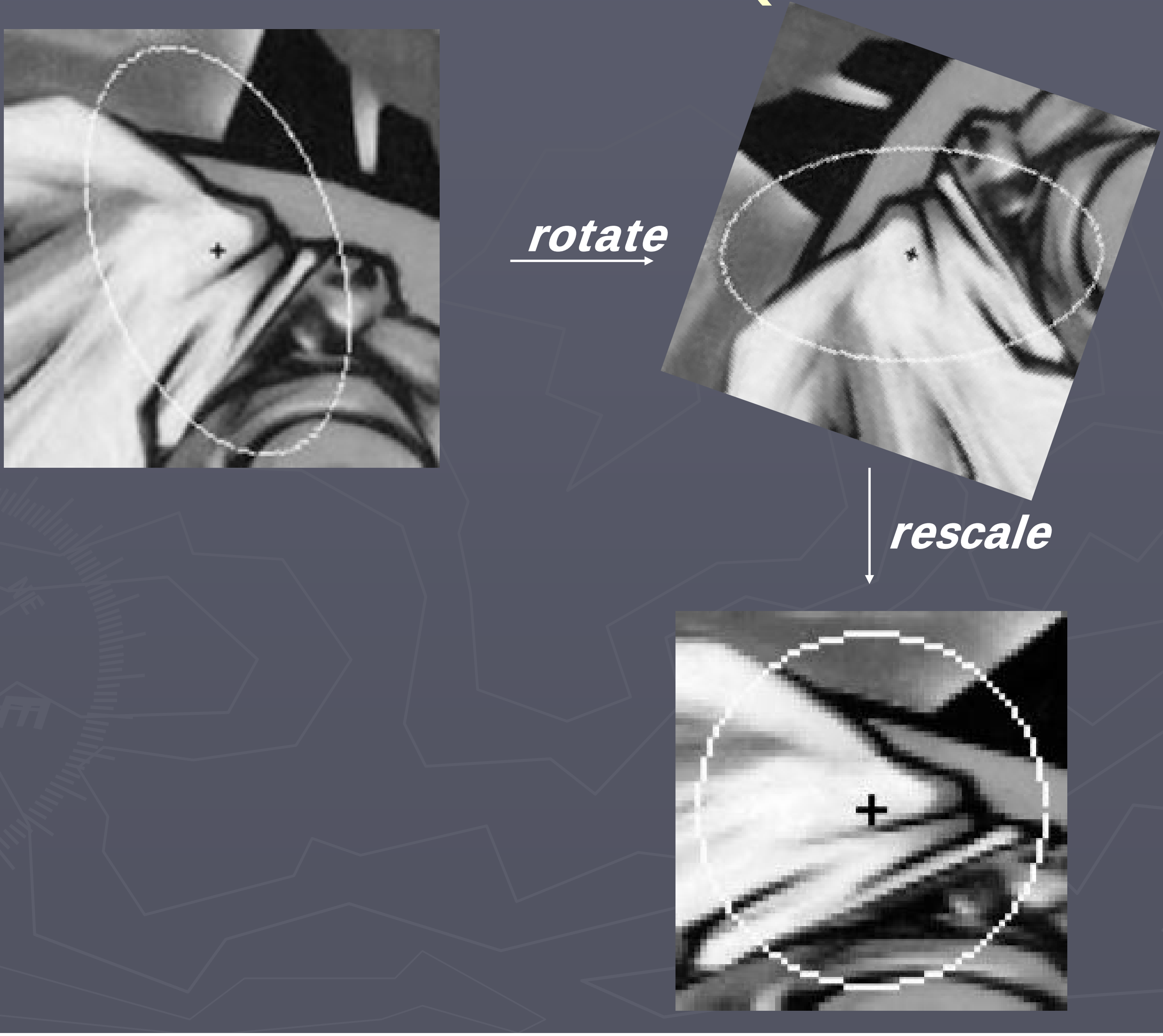
Lowe, SIFT, 1999

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





Affine normalization (‘deskewing’)



rotate

rescale

Overview

- Local Invariant Features: What? Why?
  - Introduction
  - Overview of existing detectors
  - Quantitative and qualitative comparison
- Local Invariant Features: When? How?
  - Feature descriptors
  - Applications
  - Conclusions

Overview of existing detectors

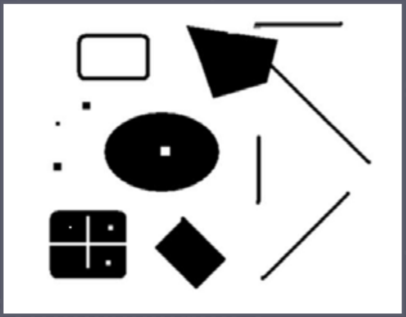
- Hessian & Harris
- Lowe: DoG
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- Others

Overview of existing detectors

- Hessian & Harris
- Lowe: DoG
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- Others

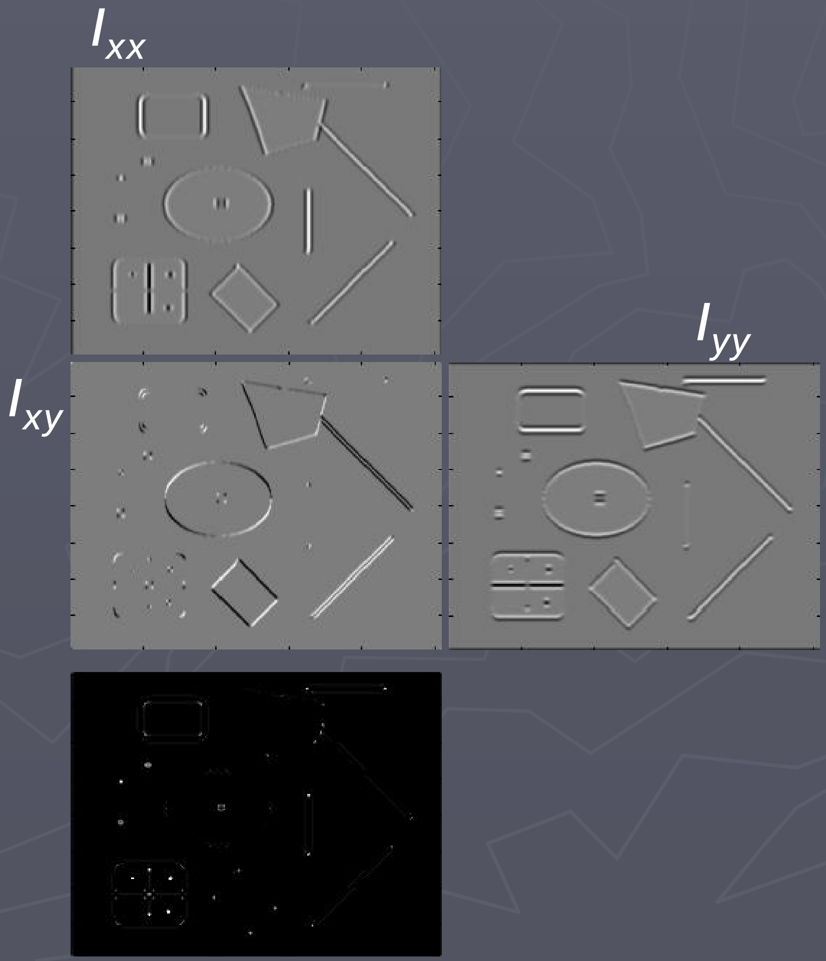
Hessian detector (Beaudet, 1978)

- Hessian determinant

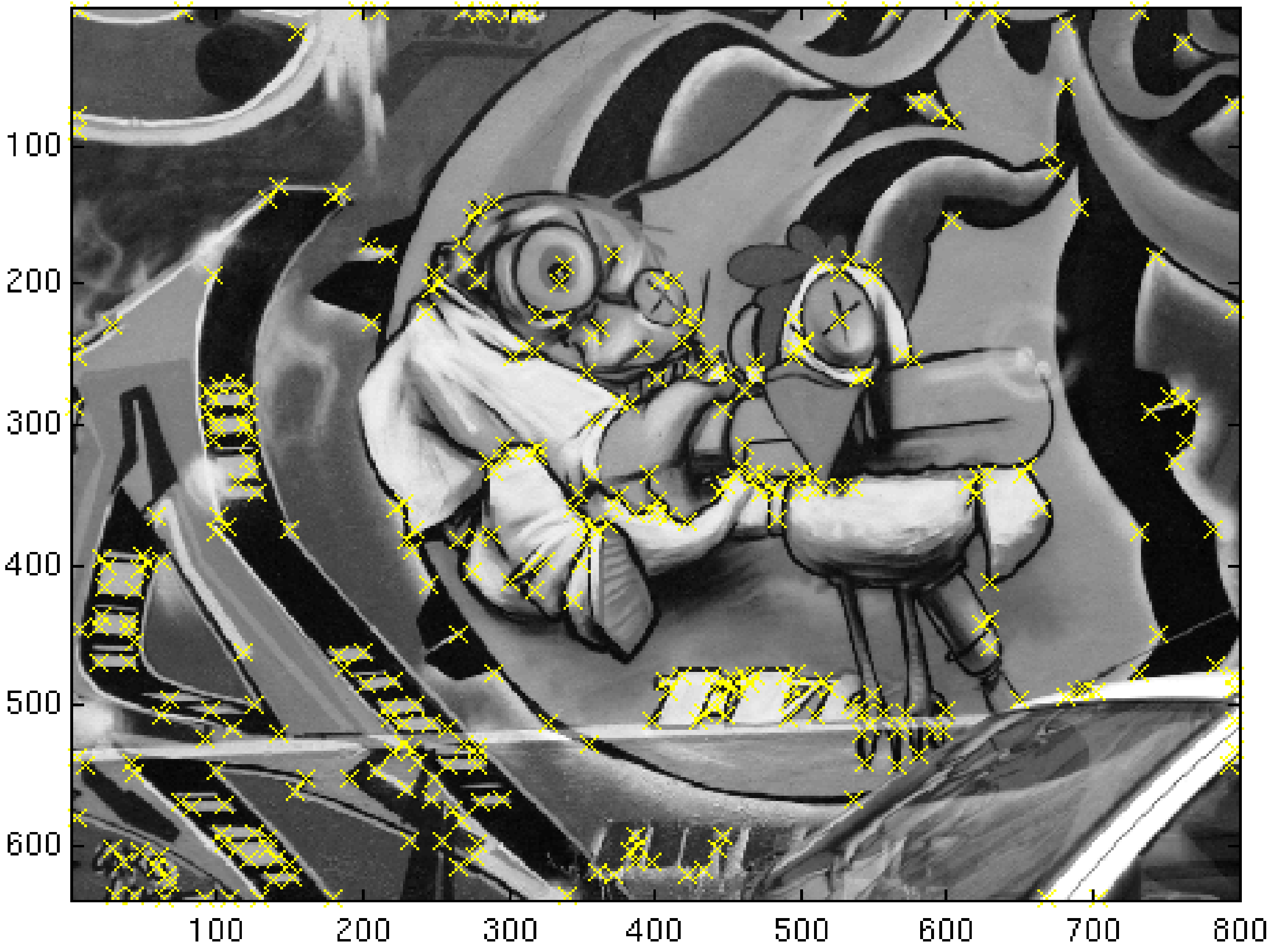


$$Hessian(I)=\begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

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



Hessian (Beaudet, 1978)

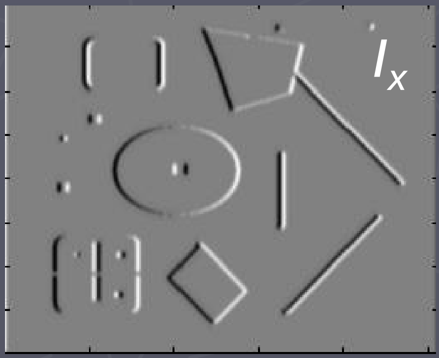


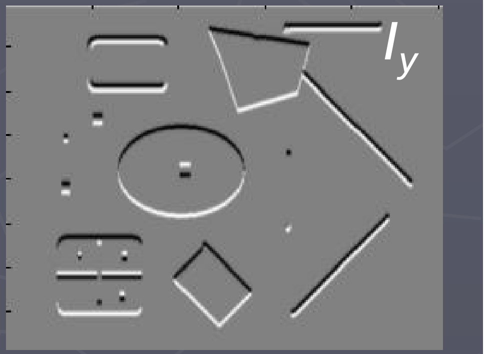
Harris detector (Harris, 1988)

- Second moment matrix / autocorrelation matrix

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

1. *Image derivatives*
   $g_x(\sigma_D)$, $g_y(\sigma_D)$,

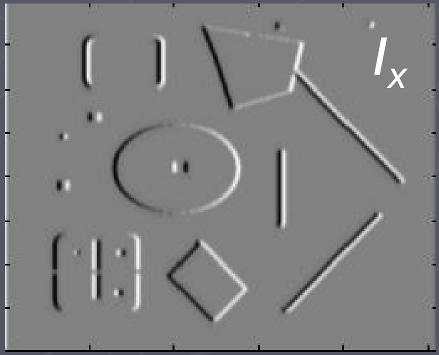


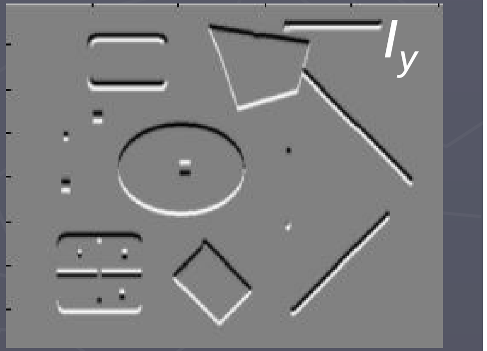


Harris detector (Harris, 1988)

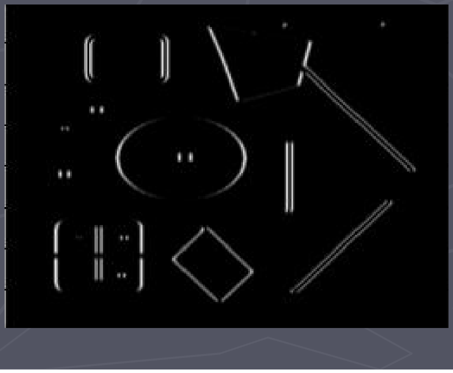
- Second moment matrix / autocorrelation matrix

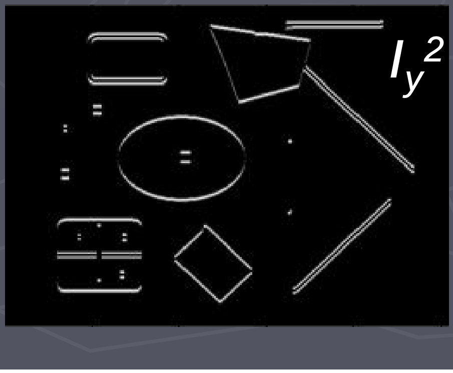
$$\mu(\sigma_I,\sigma_D)=g(\sigma_I)*\begin{bmatrix} I_x^2(\sigma_D) & I_xI_y(\sigma_D)\\ I_xI_y(\sigma_D) & I_y^2(\sigma_D)\end{bmatrix}$$

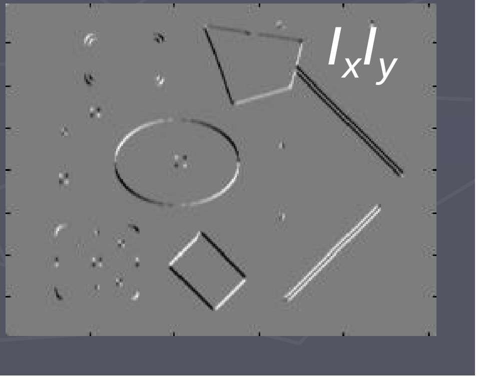




1. *Image derivatives*
   $g_x(\sigma_D)$, $g_y(\sigma_D)$,







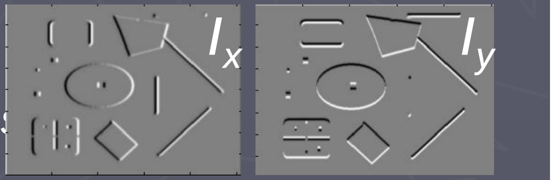
2. *Square of derivatives*

Harris detector (Harris, 1988)

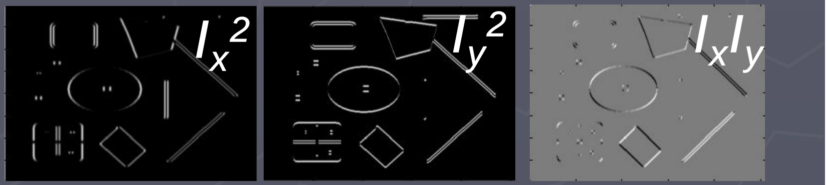
- Second moment matrix / autocorrelation matrix

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

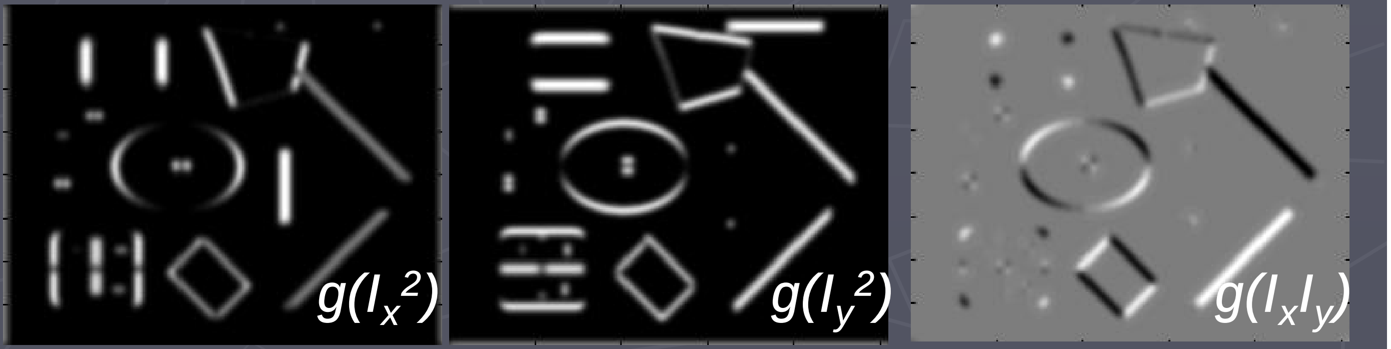
1. Image derivatives



2. Square of derivatives



3. Gaussian filter g(σ_I)



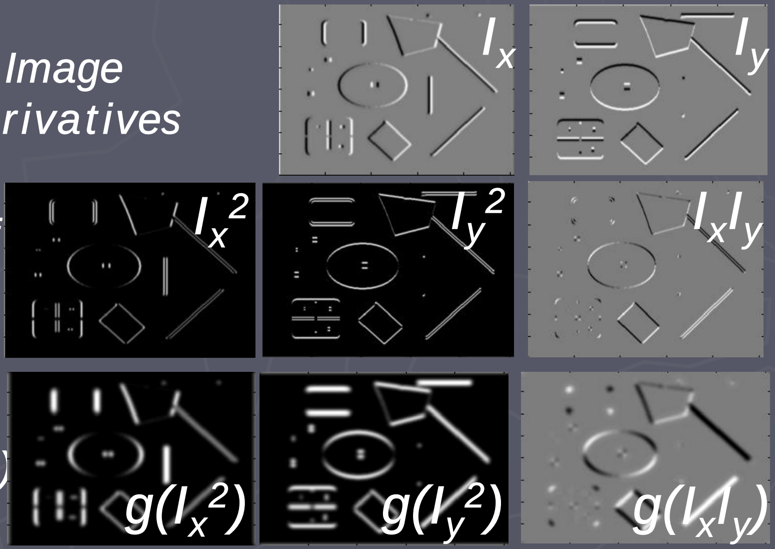
Harris detector (Harris, 1988)

- Second moment matrix autocorrelation matrix

1. Image derivatives

2. Square of derivatives

3. Gaussian filter $g(\sigma_I)$

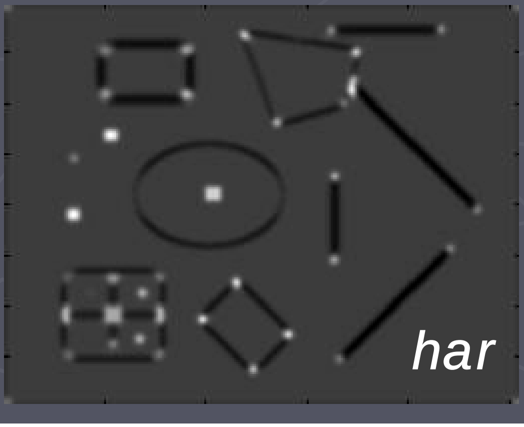


4. Cornerness function – both eigenvalues are strong

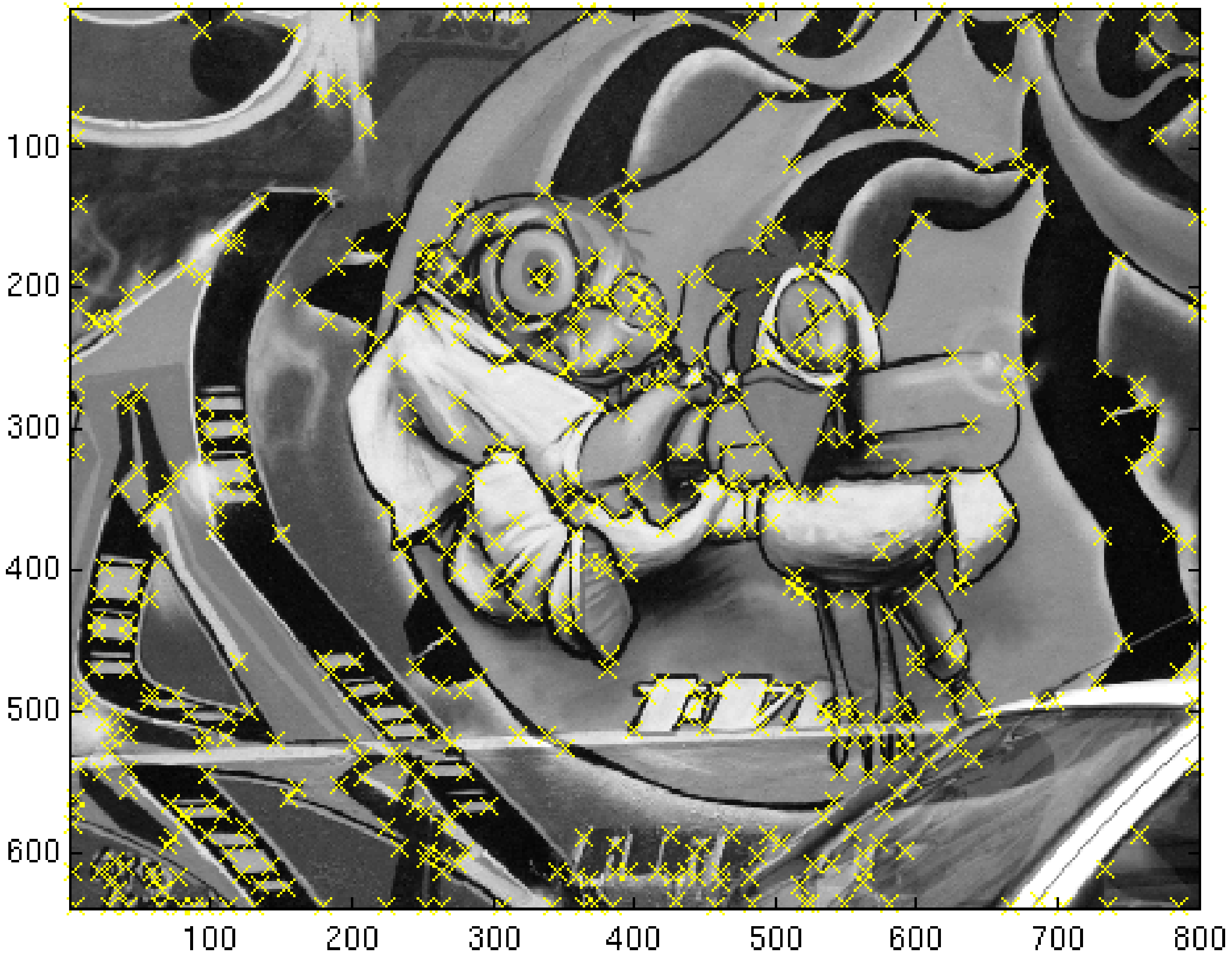
$$har = \det[\mu(\sigma_I,\sigma_D)] - \alpha[\mathrm{trace}(\mu(\sigma_I,\sigma_D))] =$$

$$g(I_x^2)g(I_y^2) - [g(I_xI_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

5. Non-maxima suppression



Harris detector (Harris, 1988)

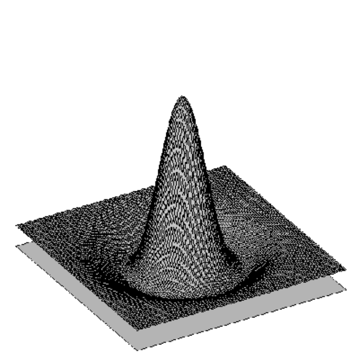


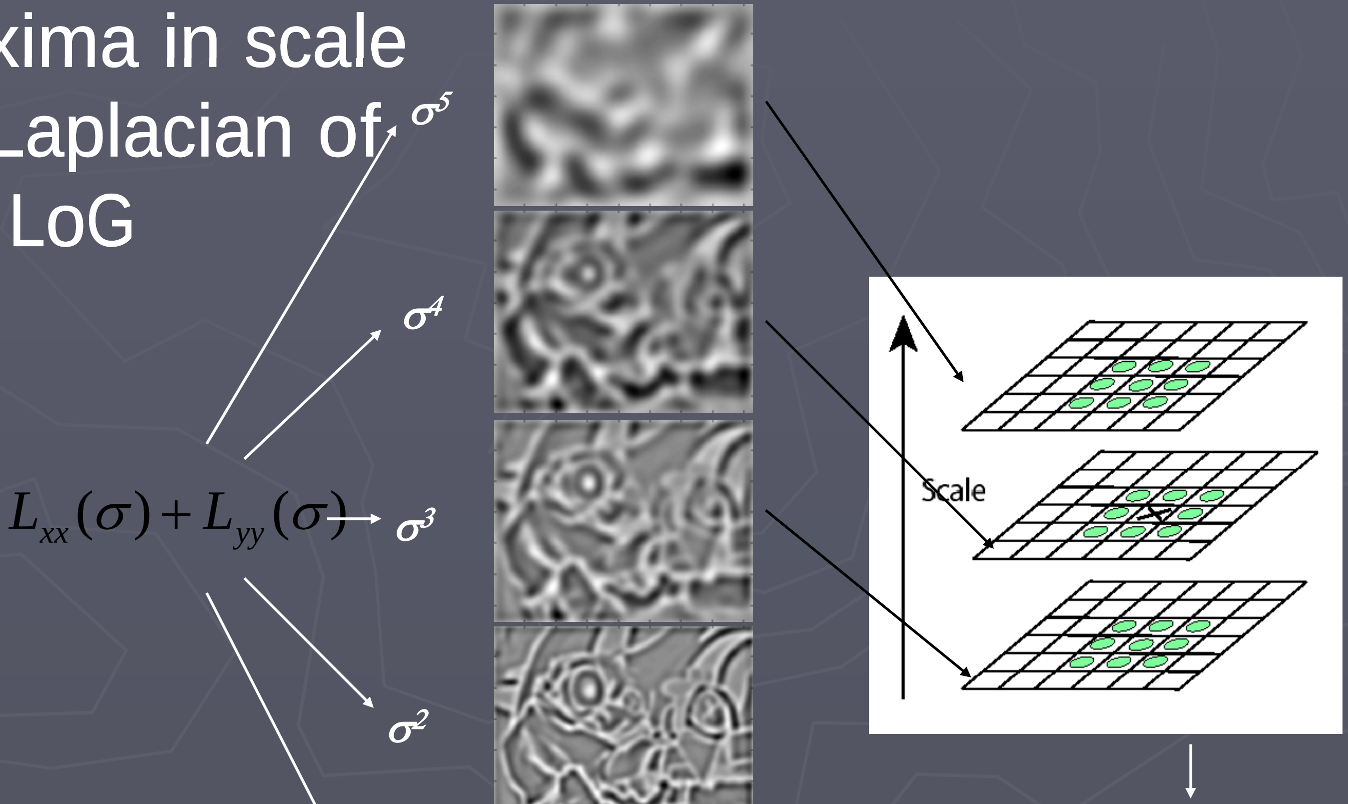
Overview of existing detectors

- Hessian & Harris
- Lowe: DoG
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- Others

Scale invariant detectors
Laplacian of Gaussian

- Local maxima in scale space of Laplacian of Gaussian LoG

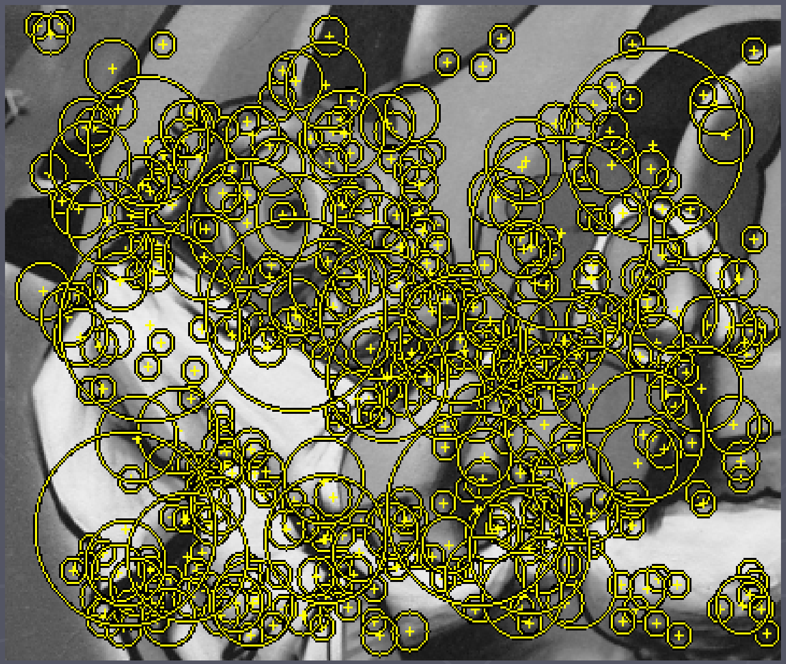




$L_{xx}(\sigma)+L_{yy}(\sigma)$

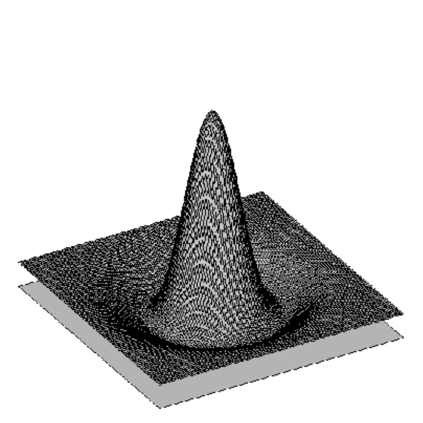
list of
(x, y, σ)

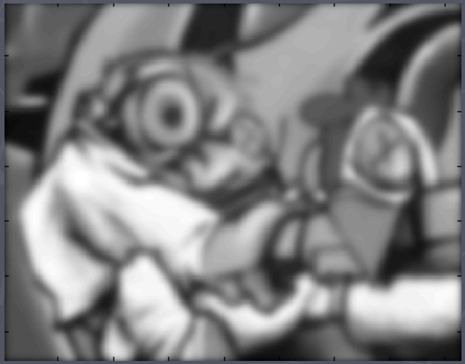
Scale invariant detectors Laplacean of Gaussian



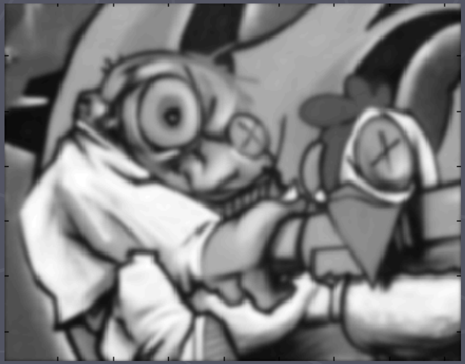
Lowe’s DoG

- Difference of Gaussians as approximation of the Laplacian of Gaussian

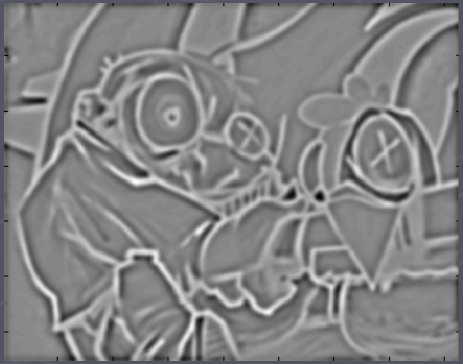




-

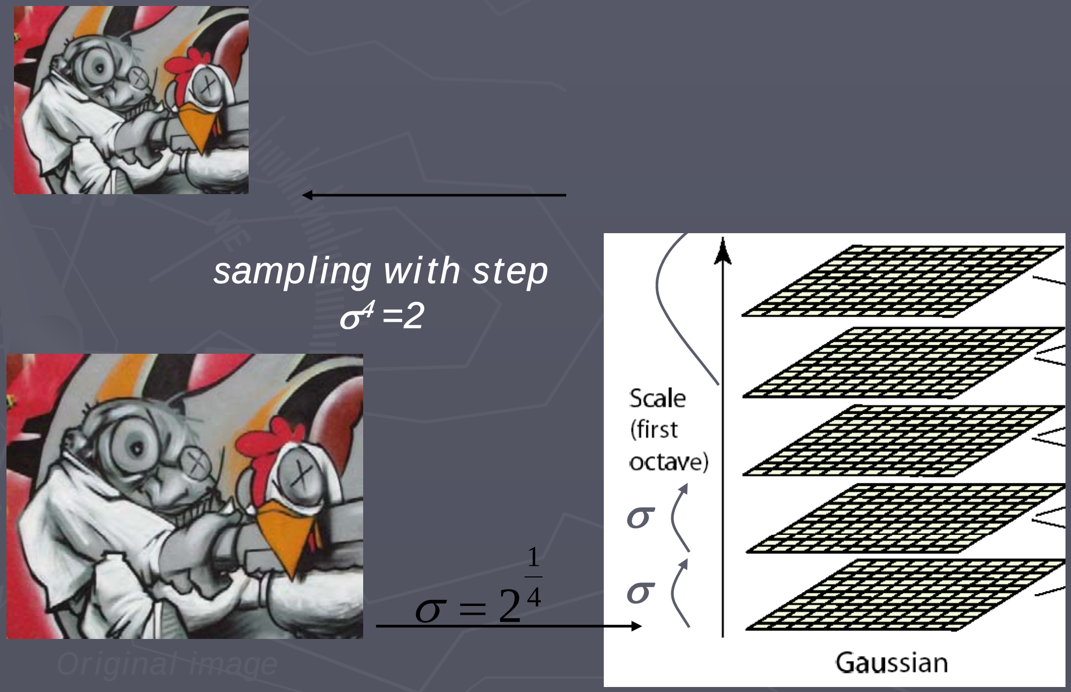


=



Lowe's DoG

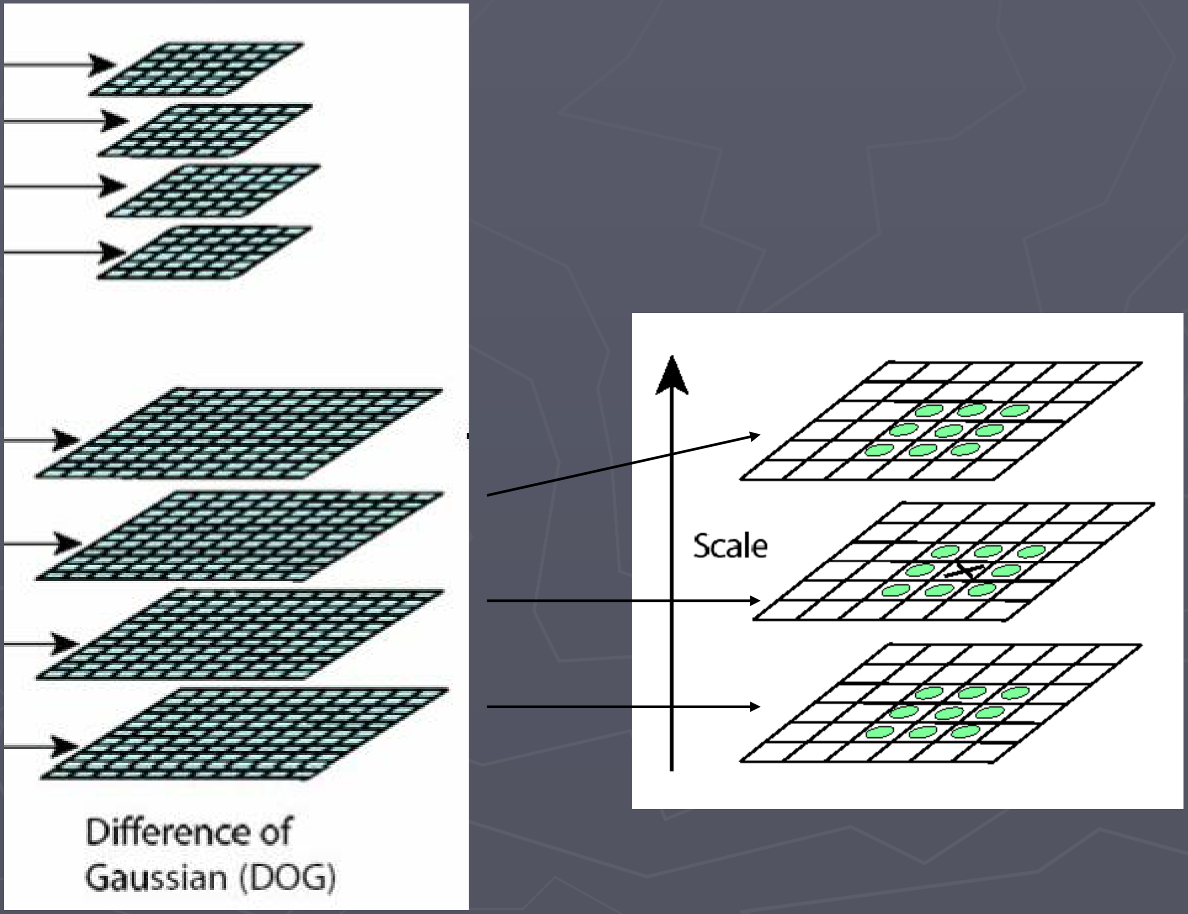
- Difference of Gaussians as approximation of the Laplacian of Gaussian



sampling with step
$\sigma^4 = 2$

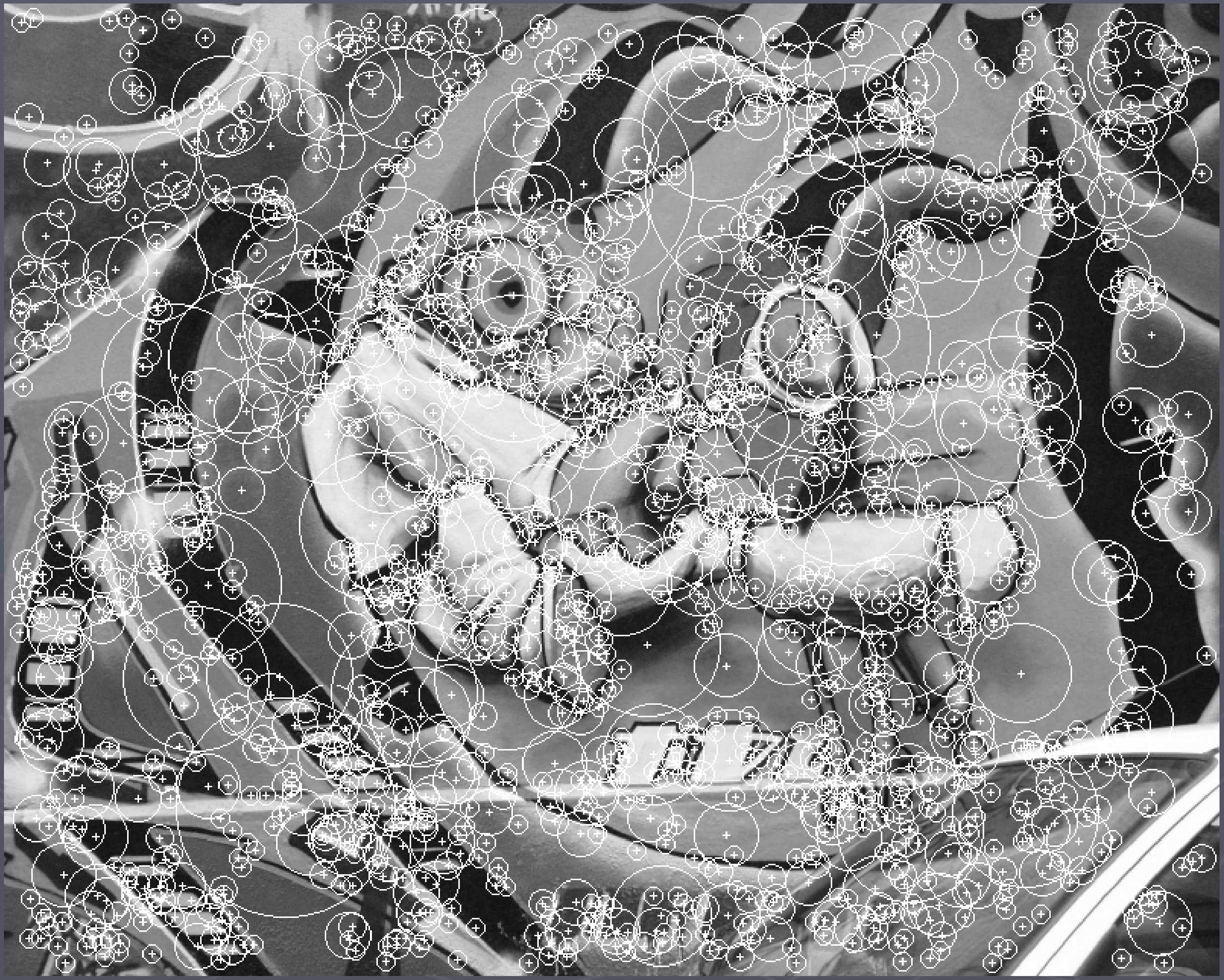
$\sigma = 2^{\frac{1}{4}}$

Original image



→ *list of*
(x, y, σ)

Lowe’s DoG



Appreciation

scale-invariant

🙂 simple, efficient scheme

😐 laplacian fires more on edges than determinant of hessian

Overview of existing detectors

- Hessian & Harris
- Lowe: DoG
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- Others

International Journal of Computer Vision 60(1), 63–86, 2004

Scale & Affine Invariant Interest Point Detectors

KRYSTIAN MIKOLAJCZYK AND CORDELIA SCHMID
INRIA Rhne-Alpes GRAVIR-CNRS, 655 av. de l'Europe, 38330 Montbonnot, France
Krystian.Mikolajczyk@inrialpes.fr
Cordelia.Schmid@inrialpes.fr

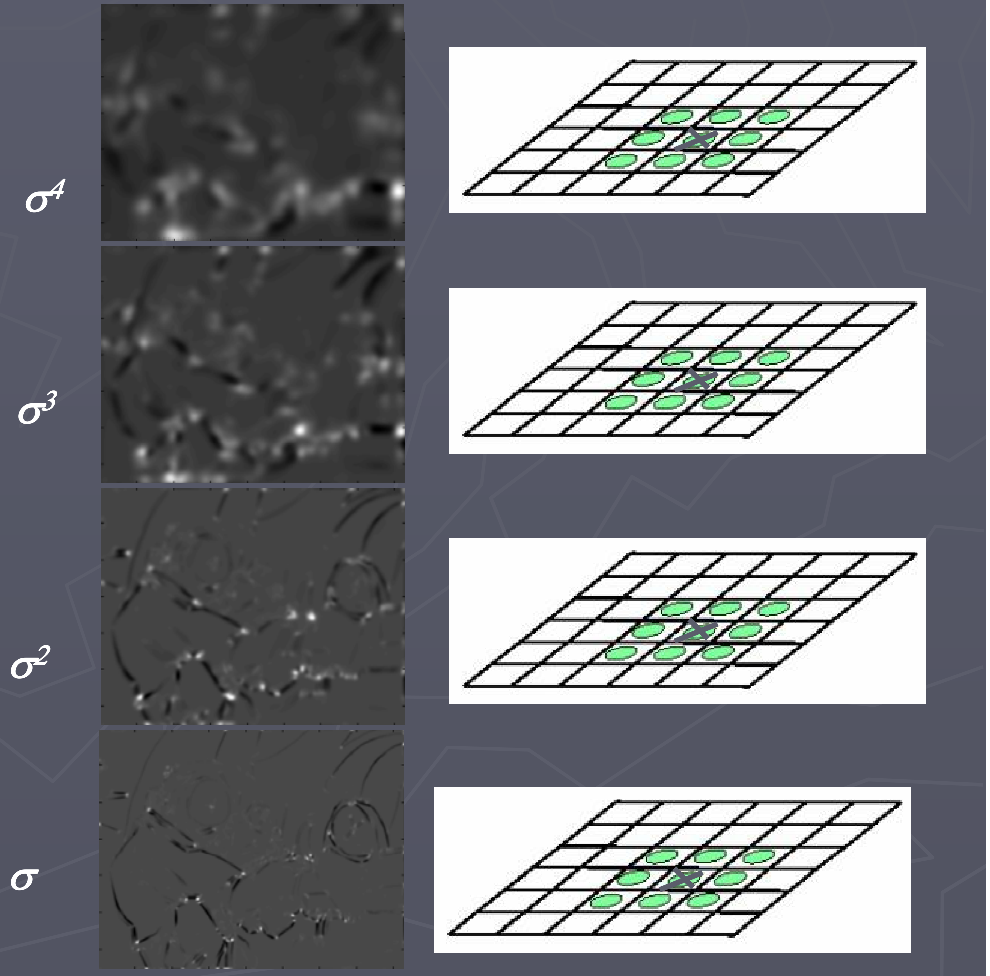
Received January 3, 2003; Revised September 24, 2003; Accepted January 22, 2004

Mikolajczyk & Schmid

- Harris Laplace
- Hessian Laplace
- Harris Affine
- Hessian Affine

Mikolajczyk: Harris Laplace

1. Initialization: Multiscale Harris corner detection

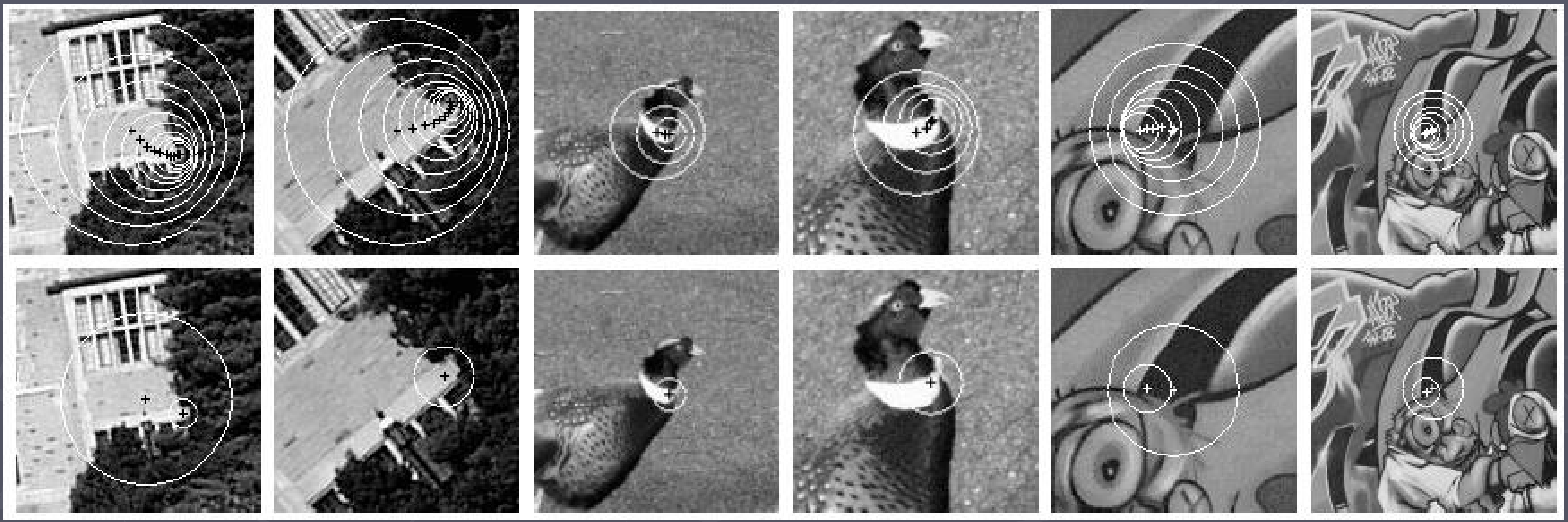


Computing Harris function *Detecting local maxima*

Mikolajczyk: Harris Laplace

1. ***Initialization: Multiscale Harris corner detection***
2. ***Scale selection based on Laplacian***

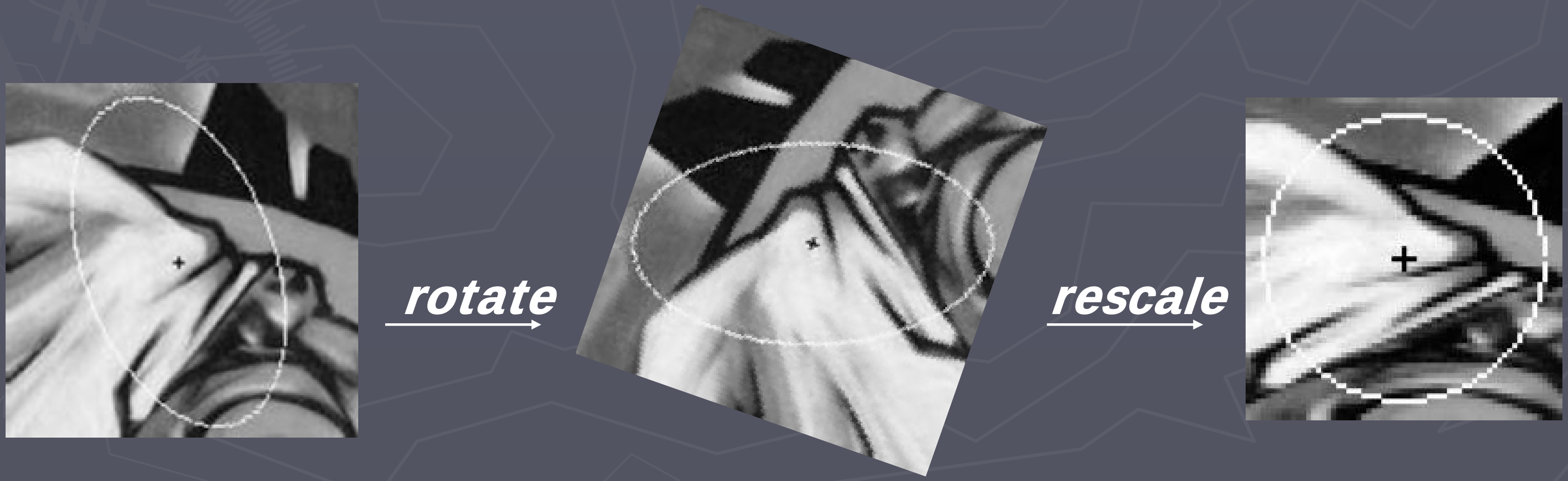
Harris points



Harris-Laplace points

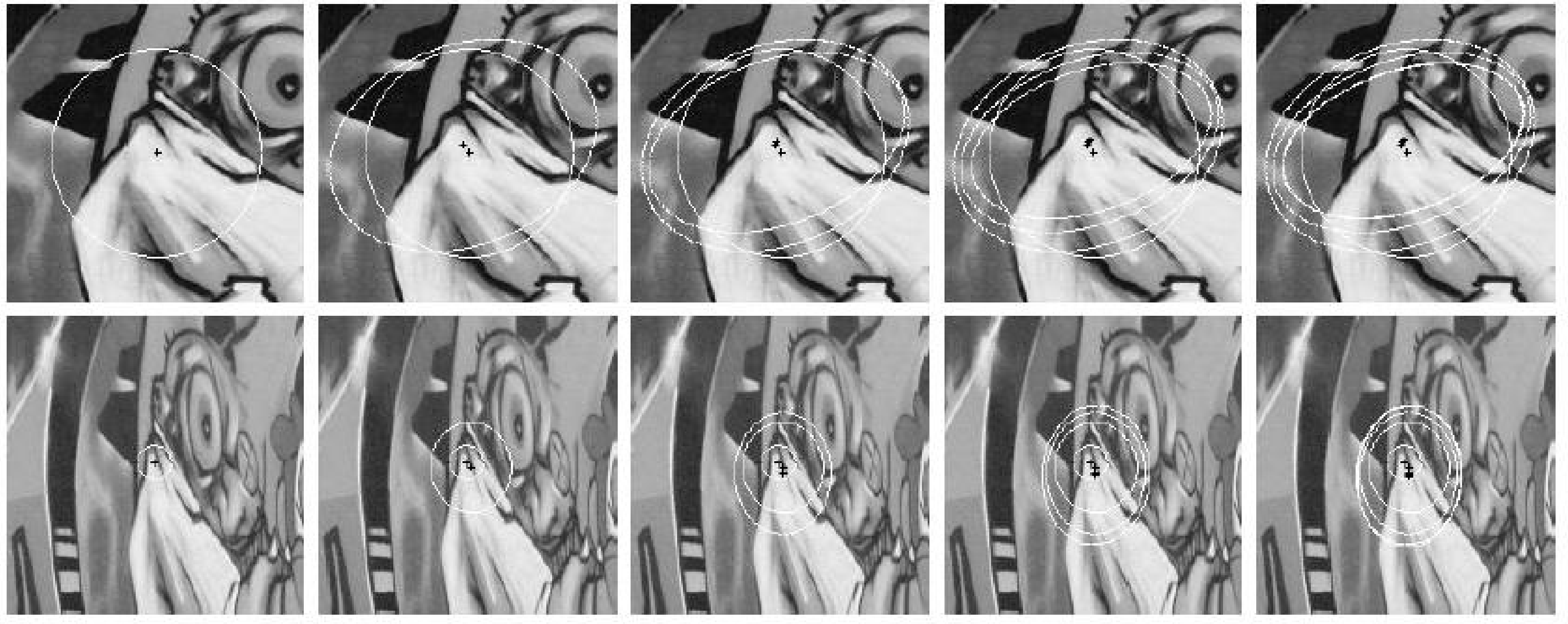
Mikolajczyk: Harris Affine

- Based on Harris Laplace
- Using normalization / deskewing



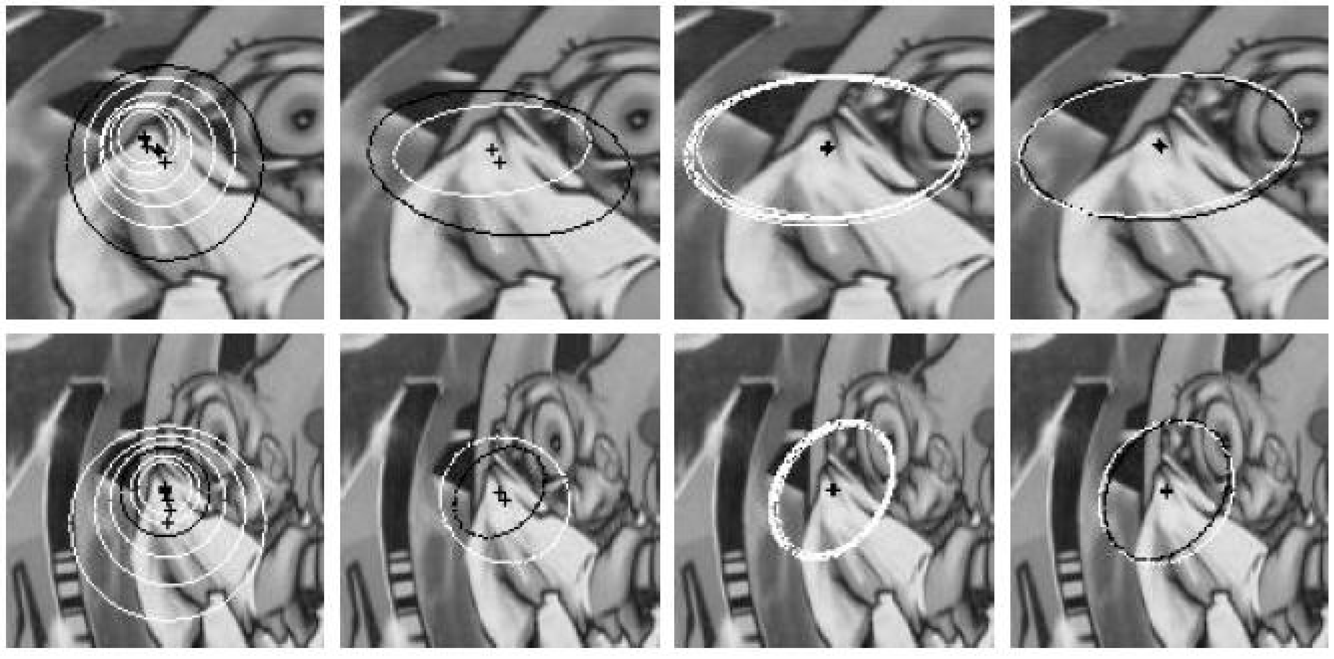
Mikolajczyk: Harris Affine

- Initialization with Harris Laplace
- Estimate shape based on second moment matrix
- Using normalization / deskewing
- Iterative algorithm



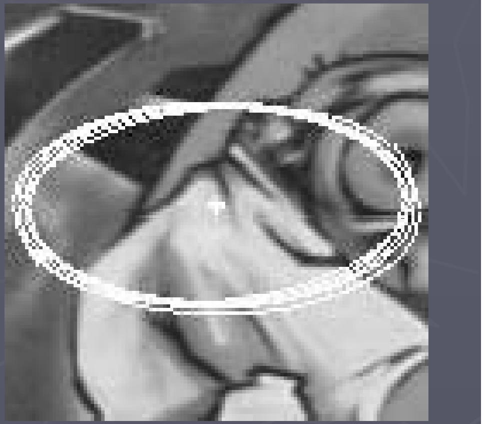
Mikolajczyk: Harris Affine

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

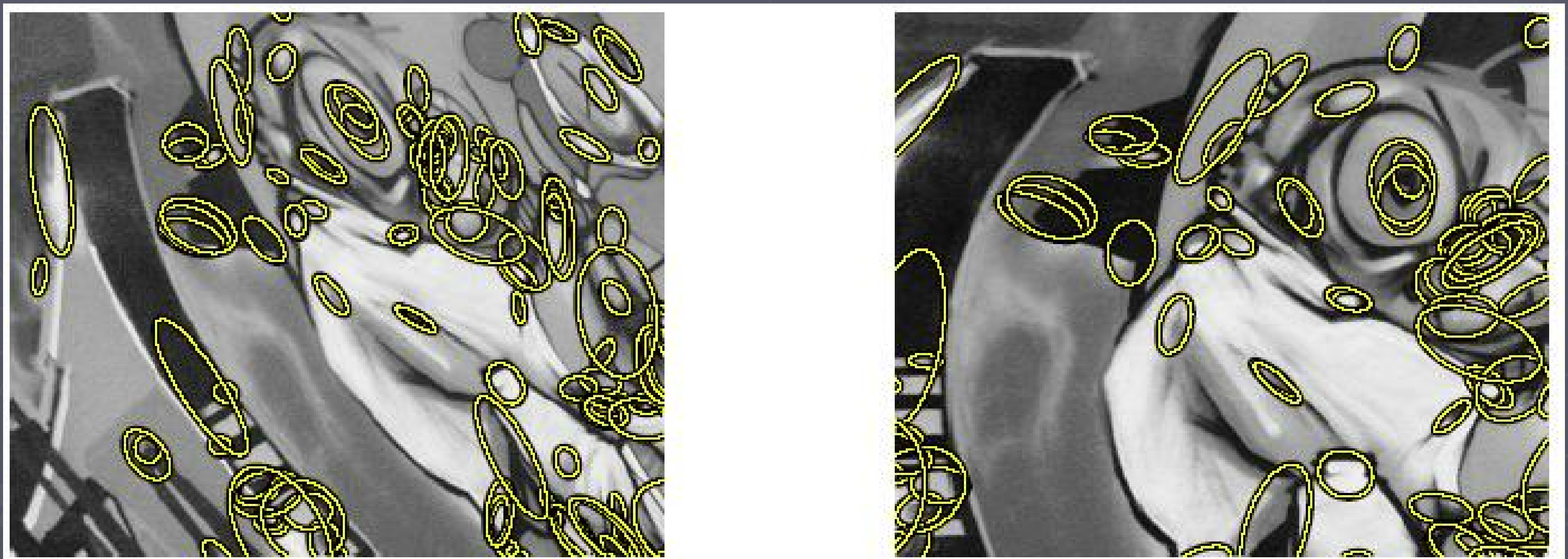


Mikolajczyk: affine invariant interest points

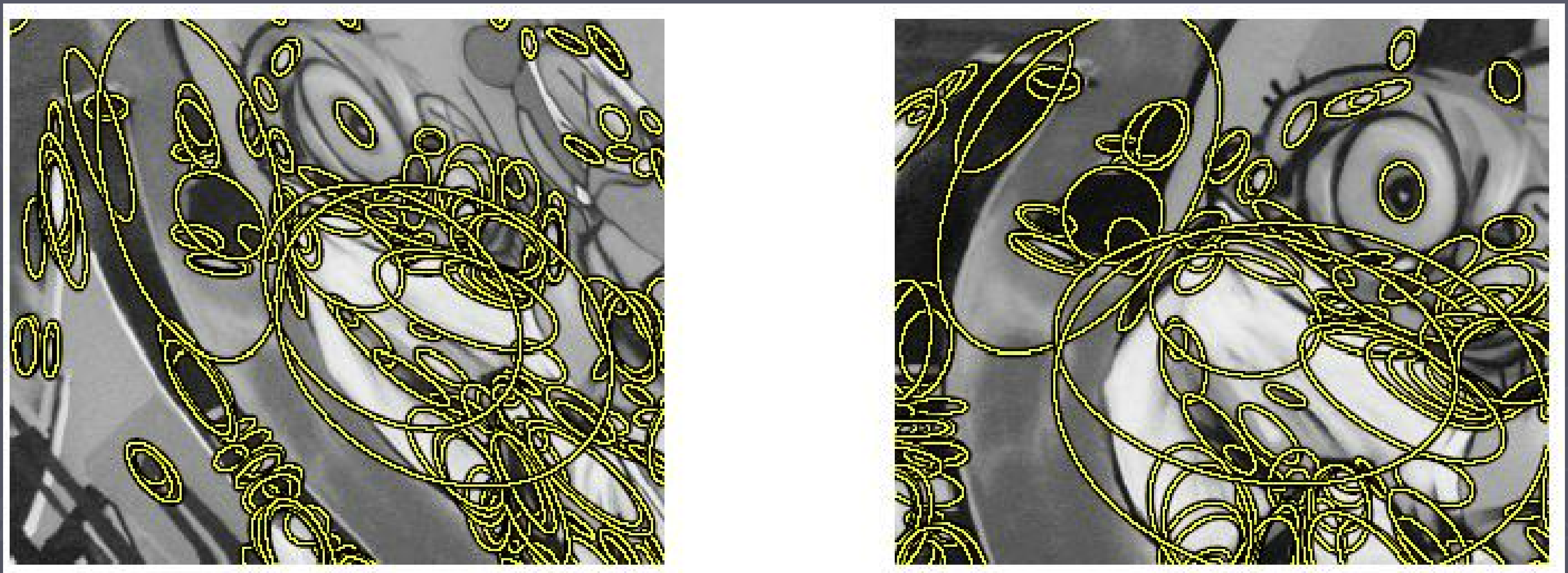
1. Initialization: Multiscale Harris corner detection
2. Iterative algorithm
   1. Normalize window (de$^{\mu}$skewing)
   2. Select integration scale (max. of LoG)
   3. Select differentiation scale (max. $\lambda_{\min} / \lambda_{\max}$)
   4. Detect spatial localization (Harris)
   5. Compute new affine transformation (μ)
   6. Go to step 2. (unless stop criterion)



Harris Affine



Hessian Affine



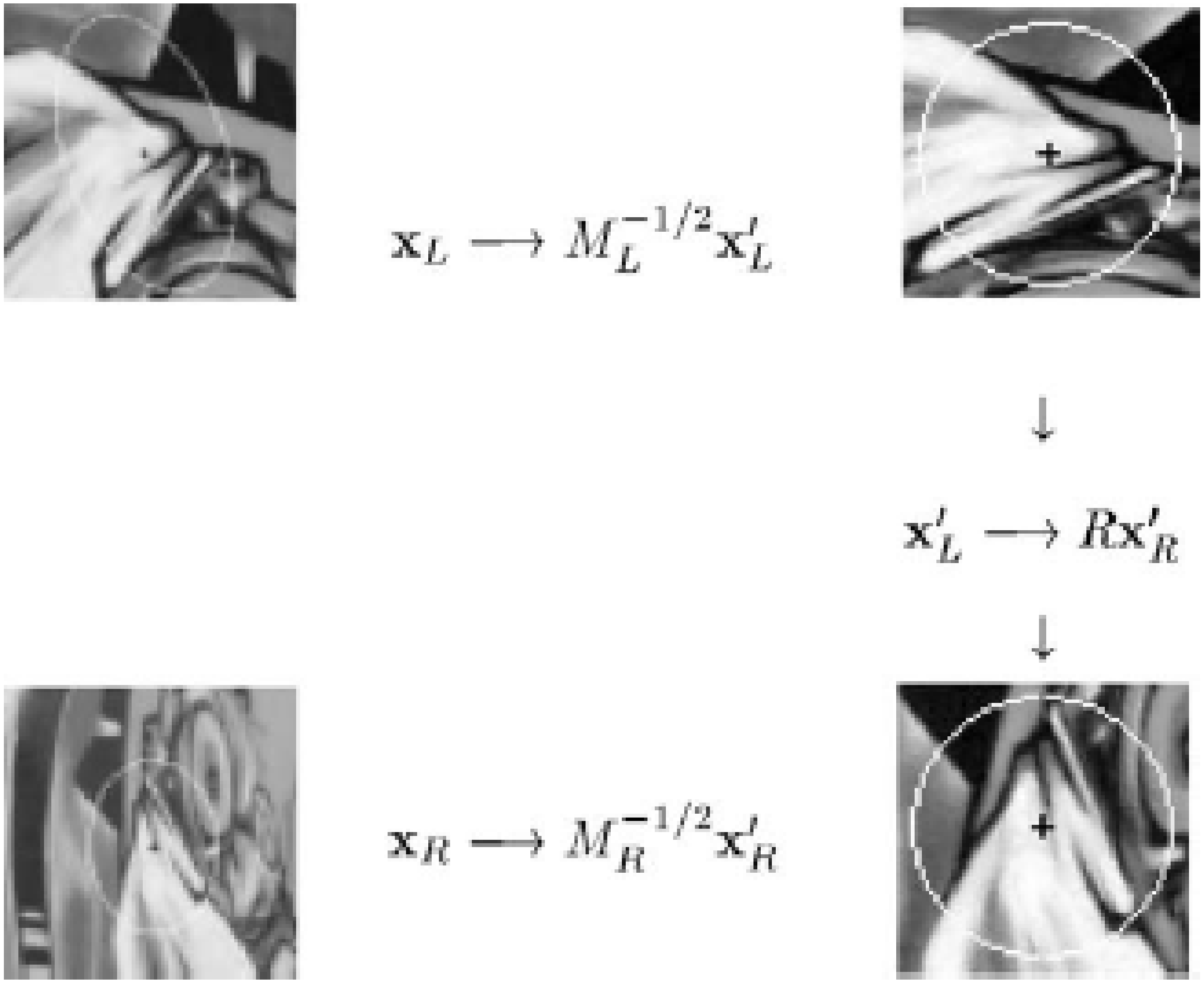

$\mathbf{x}_L \longrightarrow M_L^{-1/2}\mathbf{x}'_L$
$\mathbf{x}'_L \longrightarrow R\mathbf{x}'_R$
$\mathbf{x}_R \longrightarrow M_R^{-1/2}\mathbf{x}'_R$

Figure 4. Diagram illustrating the affine normalization based on the second moment matrices. Image coordinates are transformed with matrices $M_L^{-1/2}$ and $M_R^{-1/2}$. The transformed images are related by an orthogonal transformation.

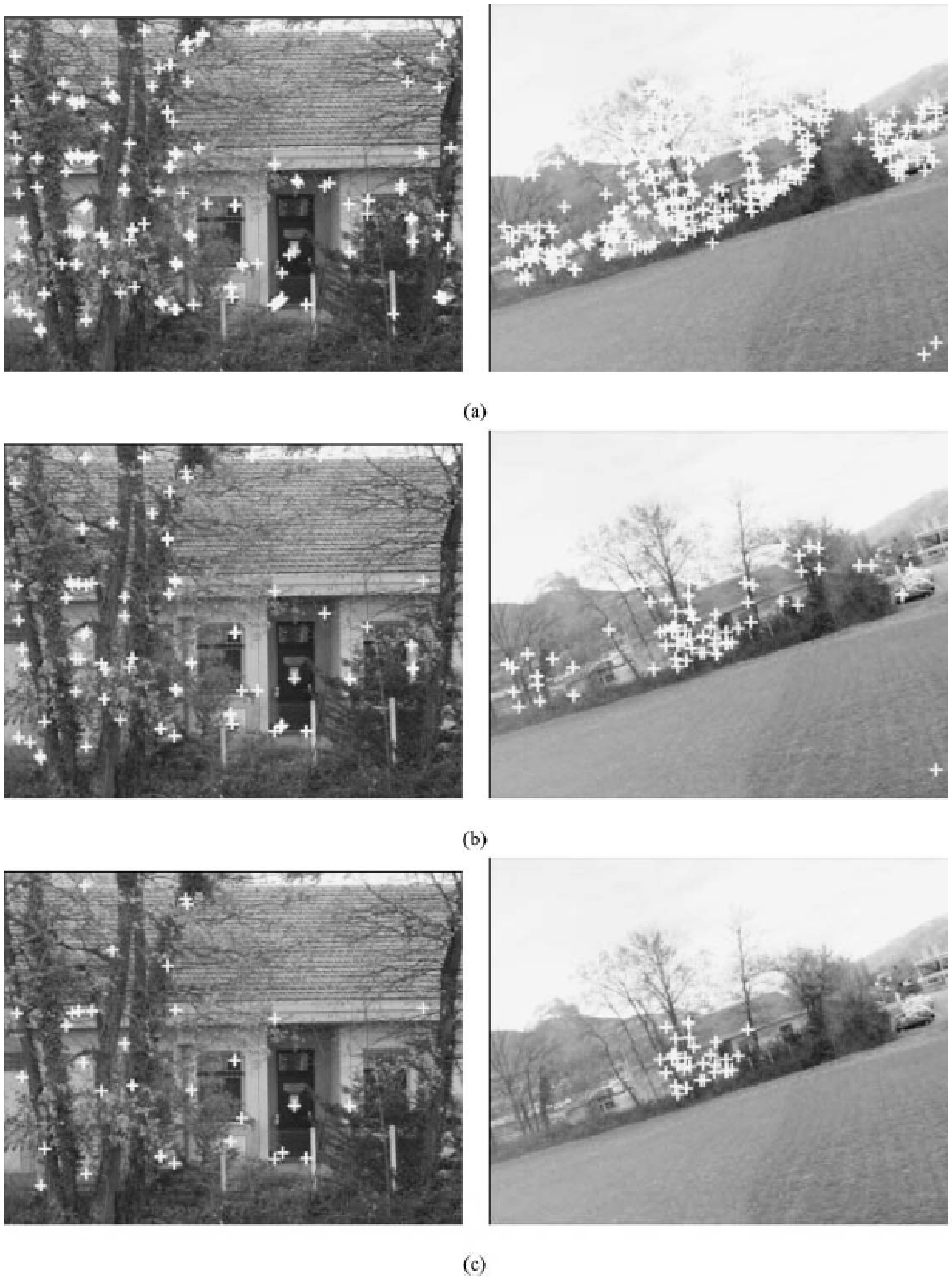
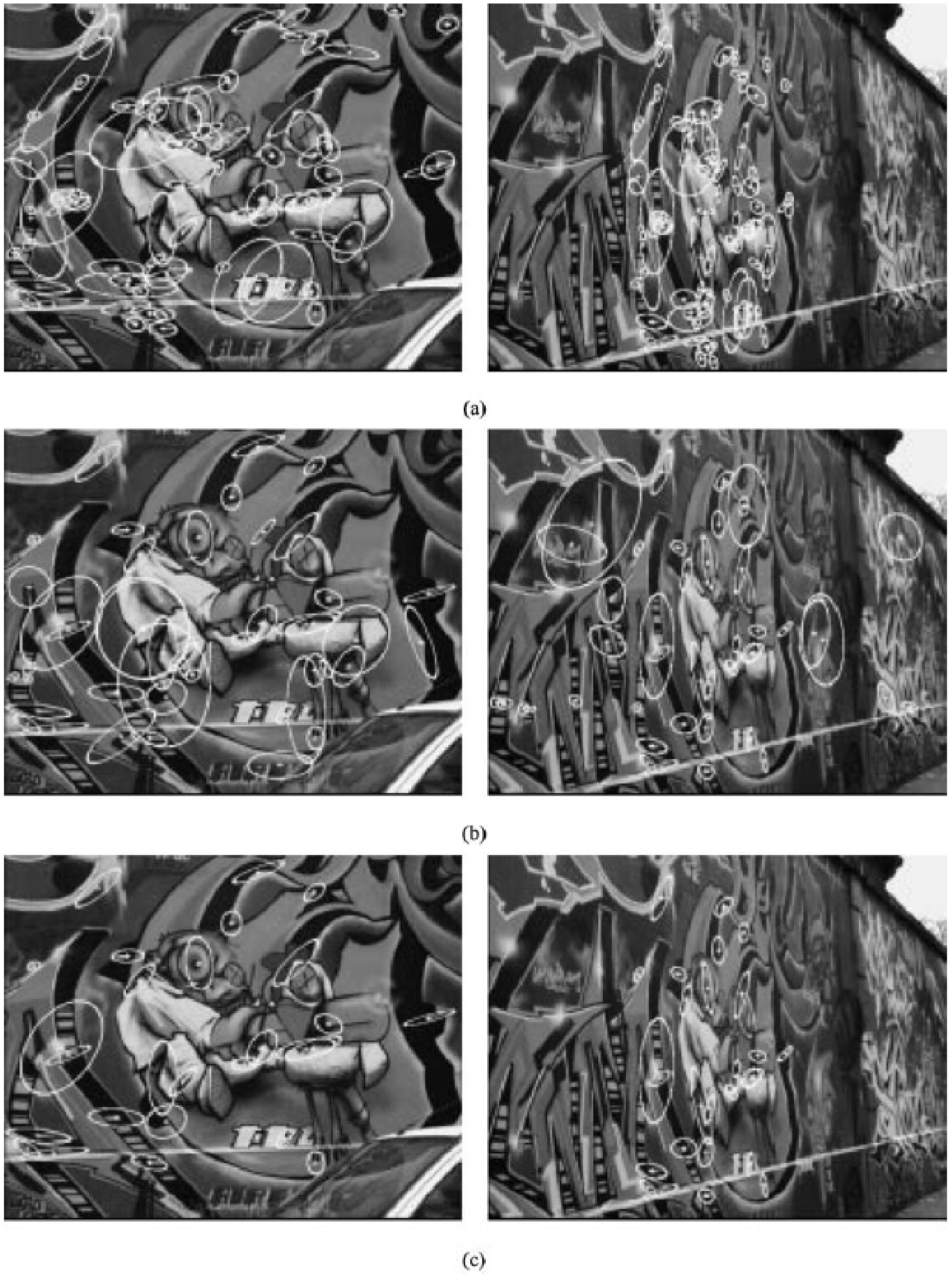


Figure 12. Robust matching: Harris-Laplace detects 190 and 213 points in the left and right images, respectively (a). 58 points are initially matched (b). There are 32 inliers to the estimated homography (c), all of which are correct. The estimated scale factor is 4.9 and the estimated rotation angle is 19 degrees.

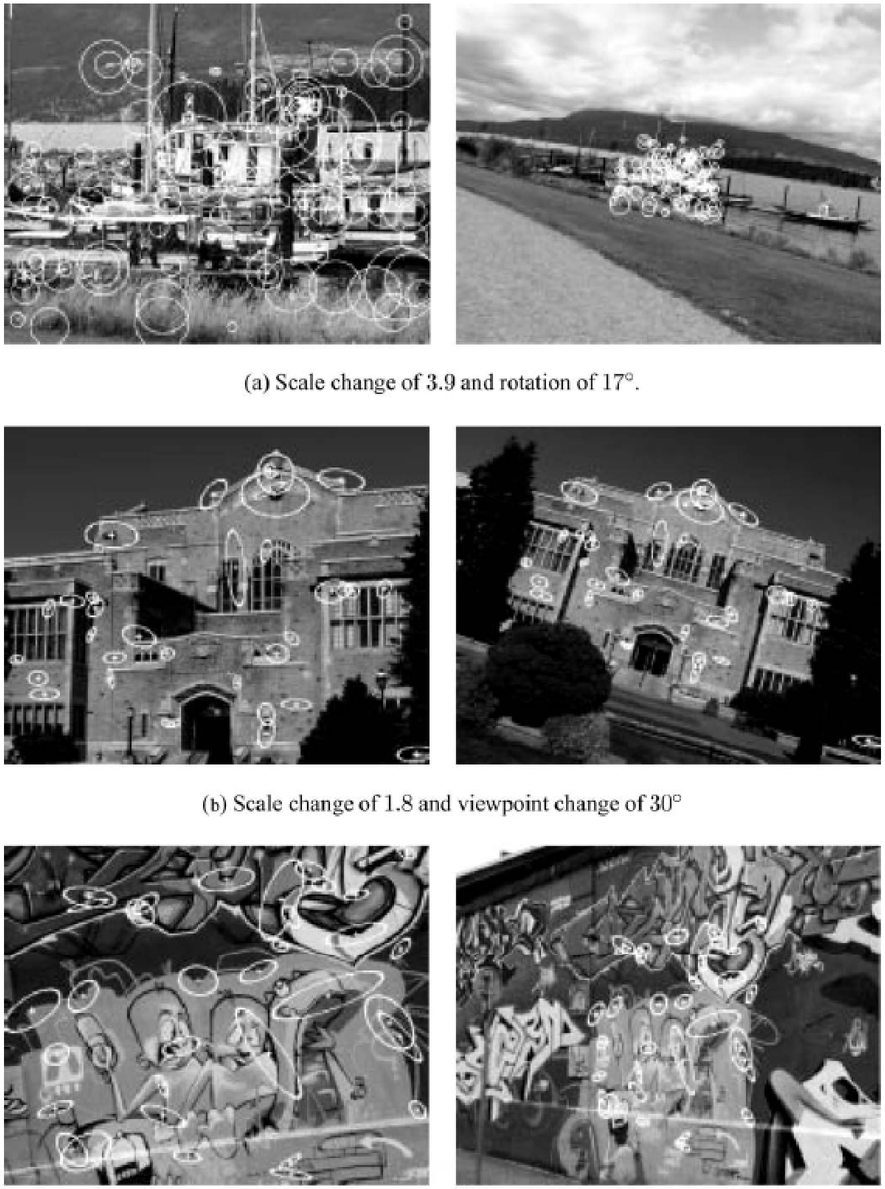


(a)

(b)

(c)

Figure 13. Robust matching: (a) 78 pairs of possible matches are found among the 287 and 325 points detected by Harris-Affine. (b) 43 points are matched based on the descriptors and the cross-correlation score. 27 of these matches are correct. (c) 27 are inliers to the estimated homography. All of them correct.



(a) Scale change of 3.9 and rotation of 17°.

(b) Scale change of 1.8 and viewpoint change of 30°

(c) Scale change of 1.7 and viewpoint change of 50°

Figure 14. Correctly matched images using scale and affine regions. The displayed matches are the inliers to a robustly estimated homography or fundamental matrix. There are (a) 118 matches (b) 34 matches and (c) 22 matches. All of them are correct.

Appreciation

Scale or affine invariant

Detects blob- and corner-like structures

☺ large number of regions

☺ well suited for object class recognition

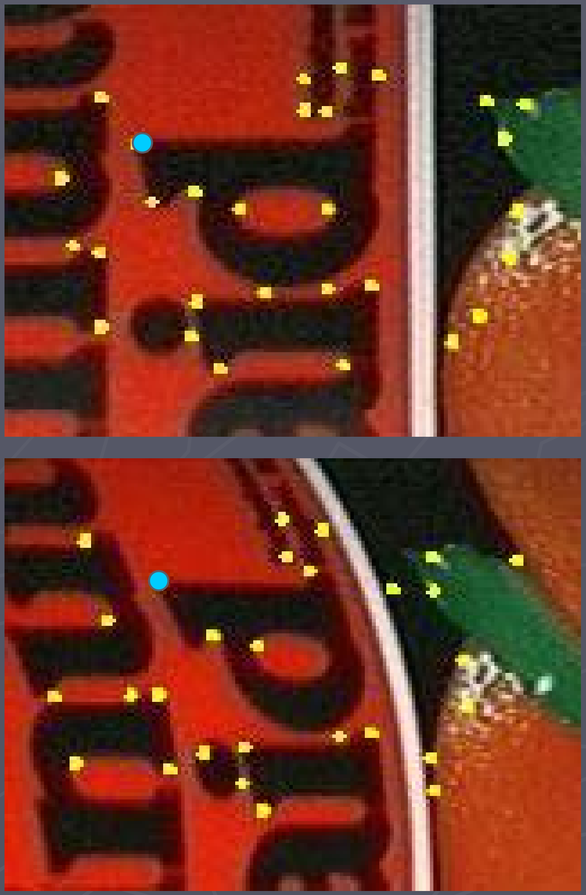
☹ less accurate than some competitors

Overview of existing detectors

- Lowe: DoG
- Lindeberg: scale selection
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- Others

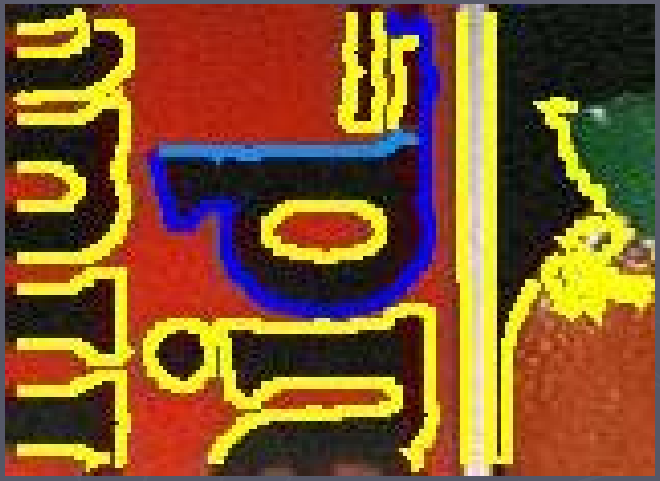
Tuytelaars: edge-based regions

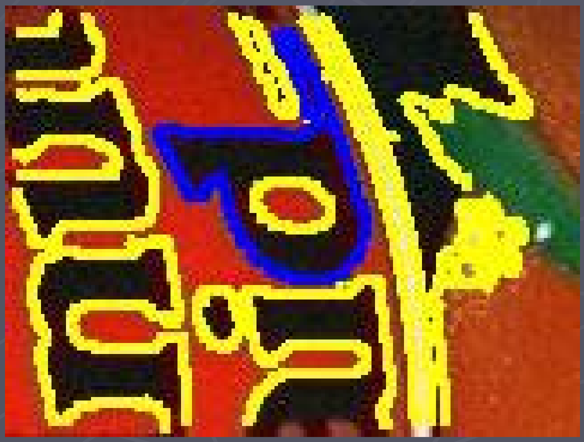
1. Select Harris corners



Tuytelaars: edge-based regions

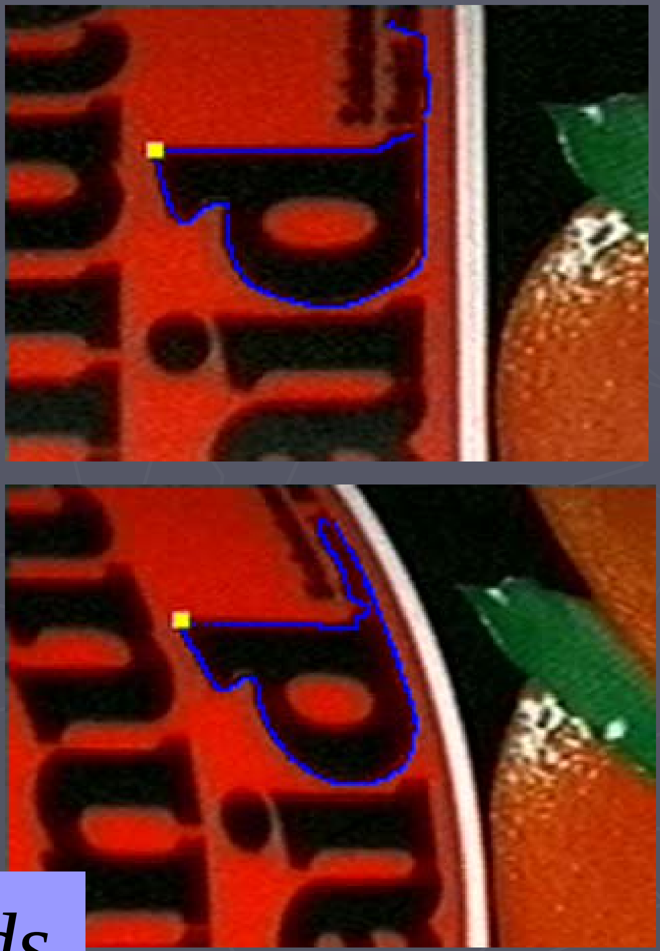
1. Select Harris corners
2. Find Canny edges





Tuytelaars: edge-based regions

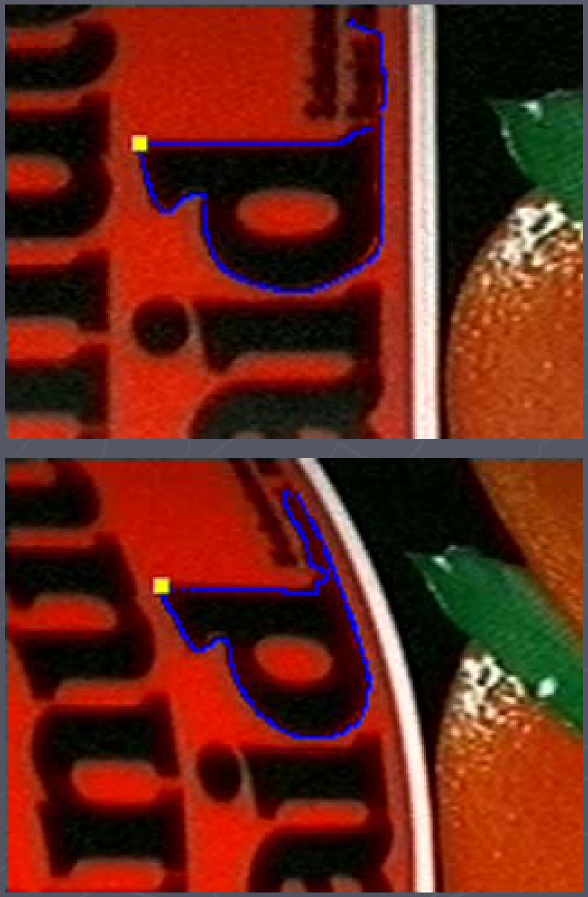
1. Select Harris corners
2. Find Canny edges
3. Evaluate relative affine invariant parameter along edges



$$l_i = \int abs(|\, p_i^{(1)}(s_i) \quad p - p_i(s_i)\, |) ds_i$$

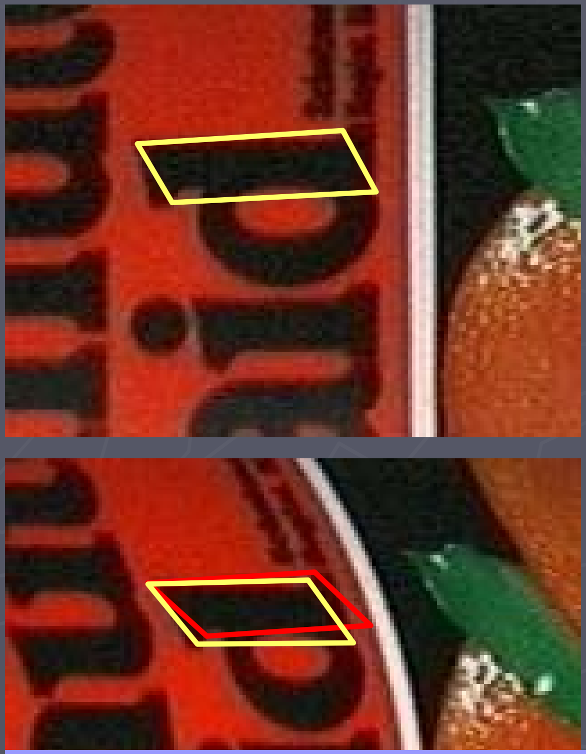
Tuytelaars: edge-based regions

1. Select Harris corners
2. Find Canny edges
3. Evaluate relative affine invariant parameter along edges
4. Construct 1-dimensional family of parallelograms



Tuytelaars: edge-based regions

1. Select Harris corners
2. Find Canny edges
3. Evaluate relative affine invariant parameter along edges
4. Construct 1-dimensional family of parallelograms
5. Select parallelogram based on local extrema of invariant function



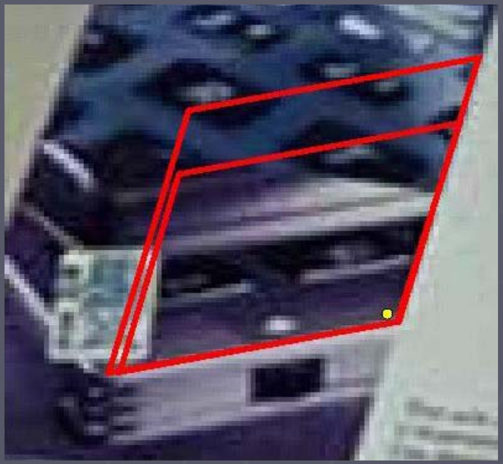
$$f(\Omega) = \frac{|\, p_1 - p_g \quad p_2 - p_g \,|}{|\, p - p_1 \quad p - p_2 \,|} \frac{M^1_{00}}{\sqrt{M^2_{00} M^0_{00} - M^1_{00} M^1_{00}}}$$

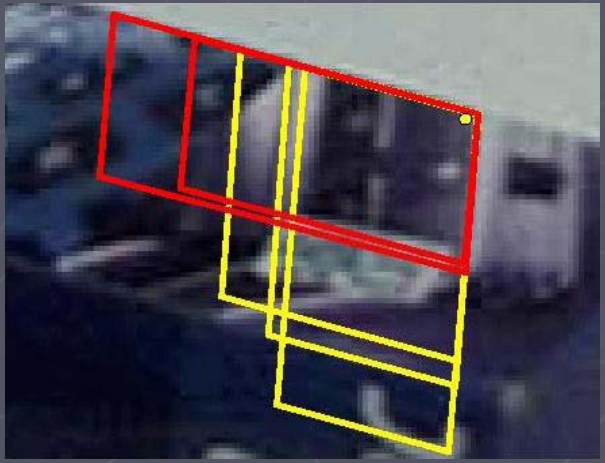
$$p_g = \left(\frac{M^1_{10}}{M^1_{00}}, \frac{M^1_{01}}{M^1_{00}}\right)$$

$$M^a_{pq} = \int [I(x,y)]^a x^p y^q \, dxdy$$

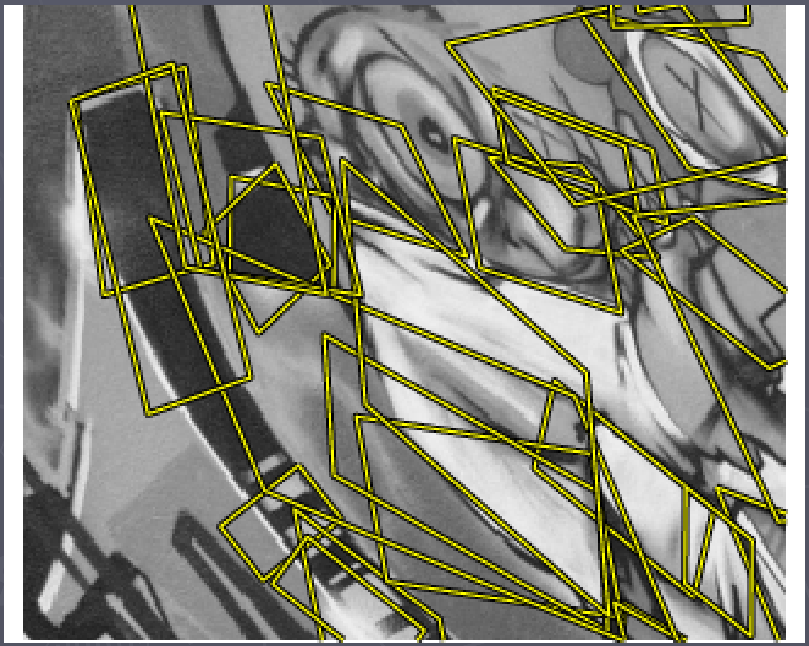
Tuytelaars: edge-based regions

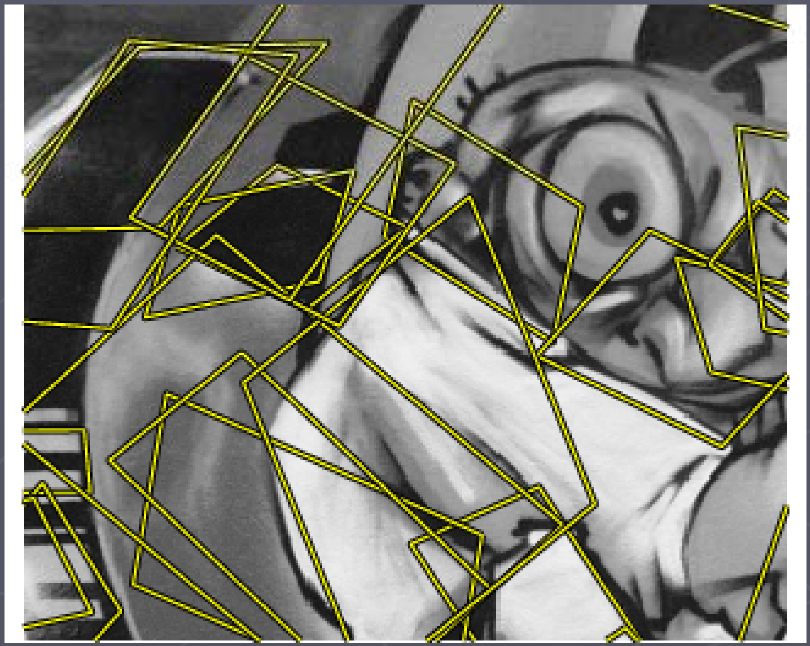
- Variant for straight lines...



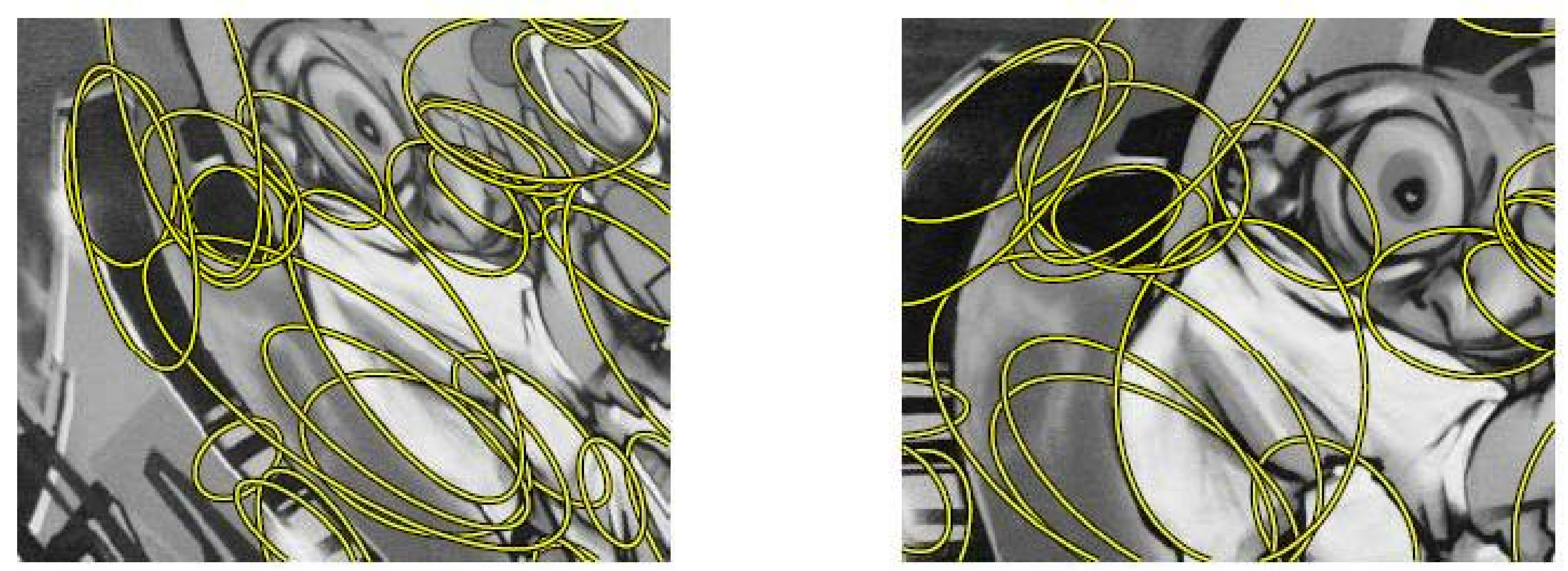


Edge-based regions





Edge-based regions



Appreciation

Affine invariant

Detects corner-like structures

🙂 Works well in structured scenes

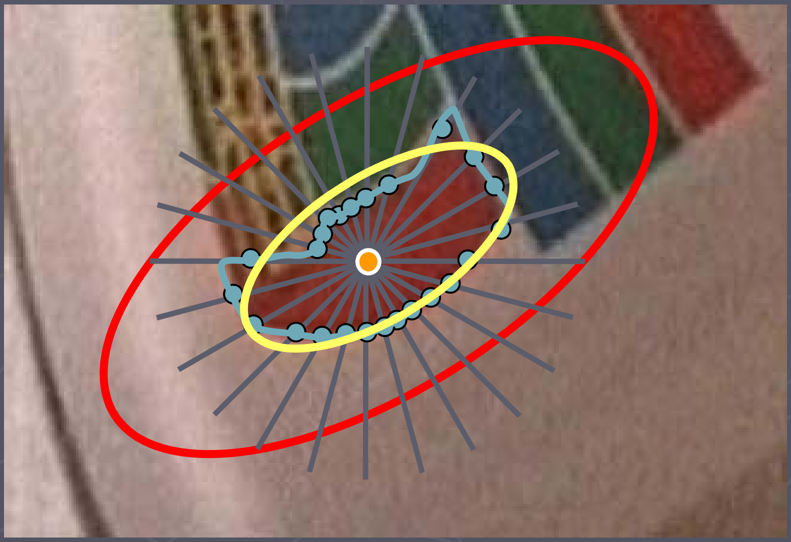
🙂 Doesn’t cross edges/object contours

🙁 Depends on presence of edges

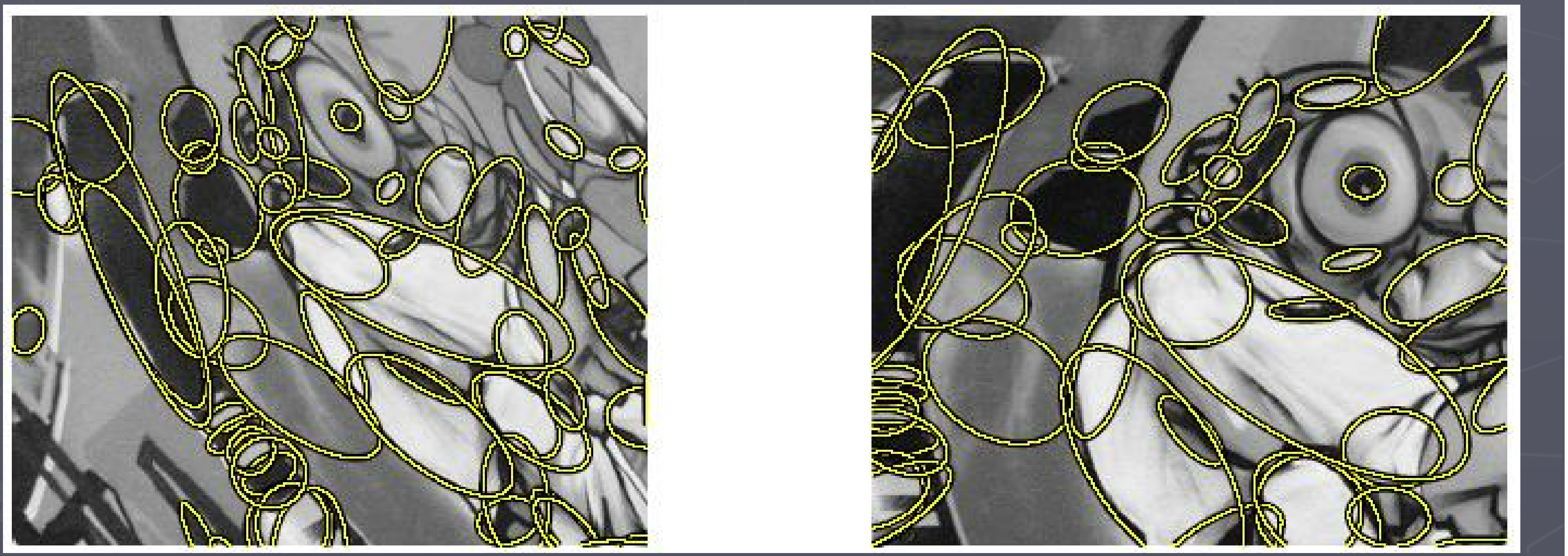
Tuytelaars: intensity-based regions

1. ***Select intensity extrema***
2. ***Consider intensity profile along rays***
3. ***Select maximum of invariant function f(t) along each ray***
4. ***Connect all local maxima***
5. ***Fit an ellipse***

$$f(t) = \frac{abs(I_0 - I)}{\max\left(\frac{\int abs(I_0 - I)dt}{t}, d\right)}$$



Intensity-based regions



Appreciation

Affine invariant

Detects ‘blob’-like structures

🙂 Accurate regions

😐 Especially good on printed material

Overview of existing detectors

- Lowe: DoG
- Lindeberg: scale selection
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- Others

Robust Wide Baseline Stereo from Maximally Stable Extremal Regions

J. Matas[1,2], O. Chum[1], M. Urban[1], T. Pajdla[1]

[1]Center for Machine Perception, Dept. of Cybernetics, CTU Prague, Karlovo nám 13, CZ 121 35

[2]CVSSP, University of Surrey, Guildford GU2 7XH, UK

[matas, chum]@cmp.felk.cvut.cz

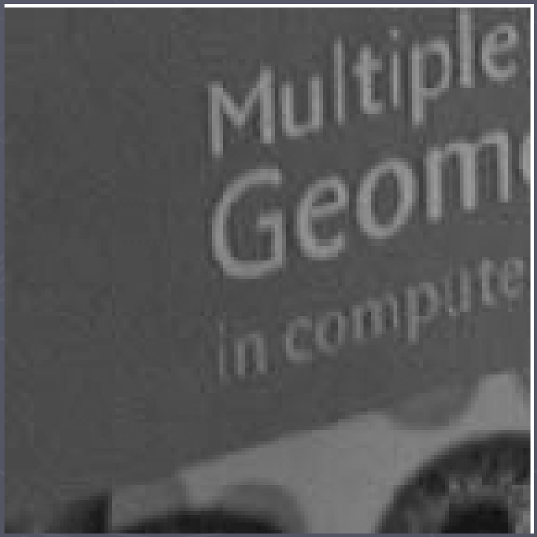
Abstract

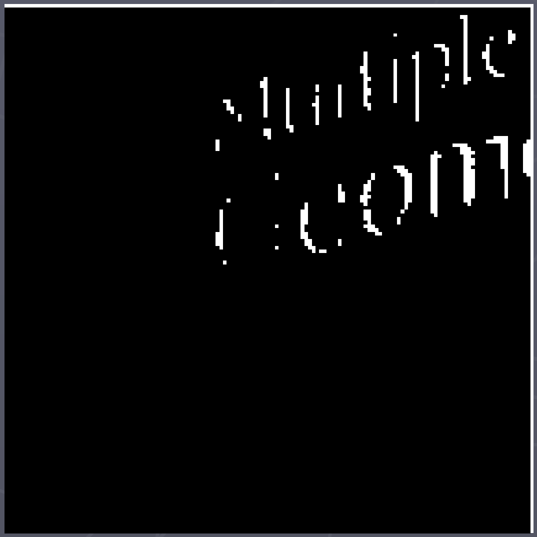
The wide-baseline stereo problem, i.e. the problem of establishing correspondences between a pair of images taken from different viewpoints is studied.

A new set of image elements that are put into correspondence, the so called *extremal regions*, is introduced. Extremal regions possess highly desirable properties: the set is closed under 1. continuous (and thus projective) transformation of image coordinates and 2. monotonic transformation of image intensities. An efficient (near linear complexity) and practically fast detection algorithm (near frame rate) is presented for an affinely-invariant stable subset of extremal regions, the maximally stable extremal regions (MSER).

Matas: Maximally Stable Extremal Regions (MSERs)

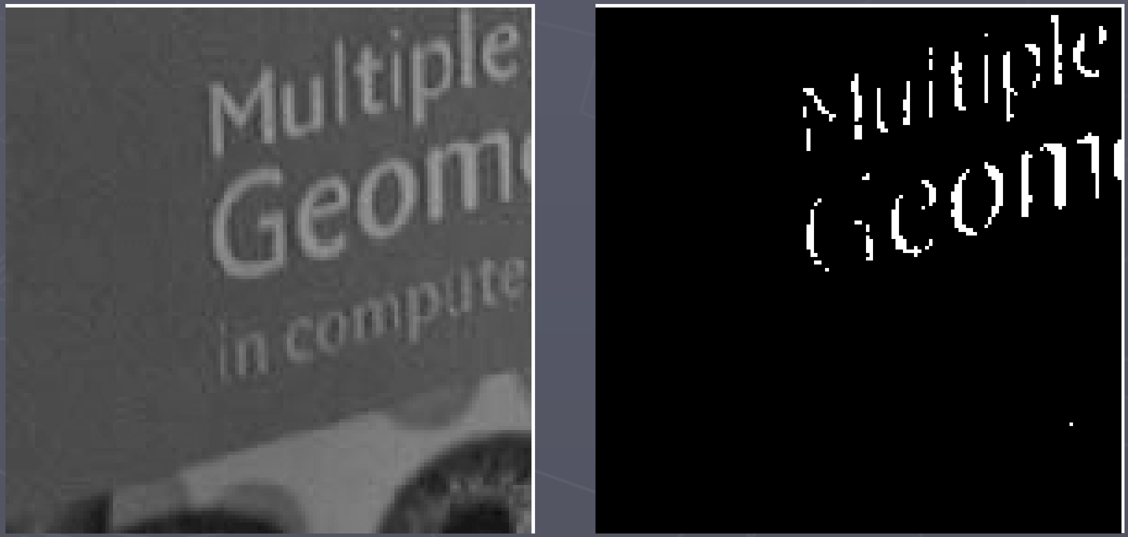
- Based on watershed algorithm





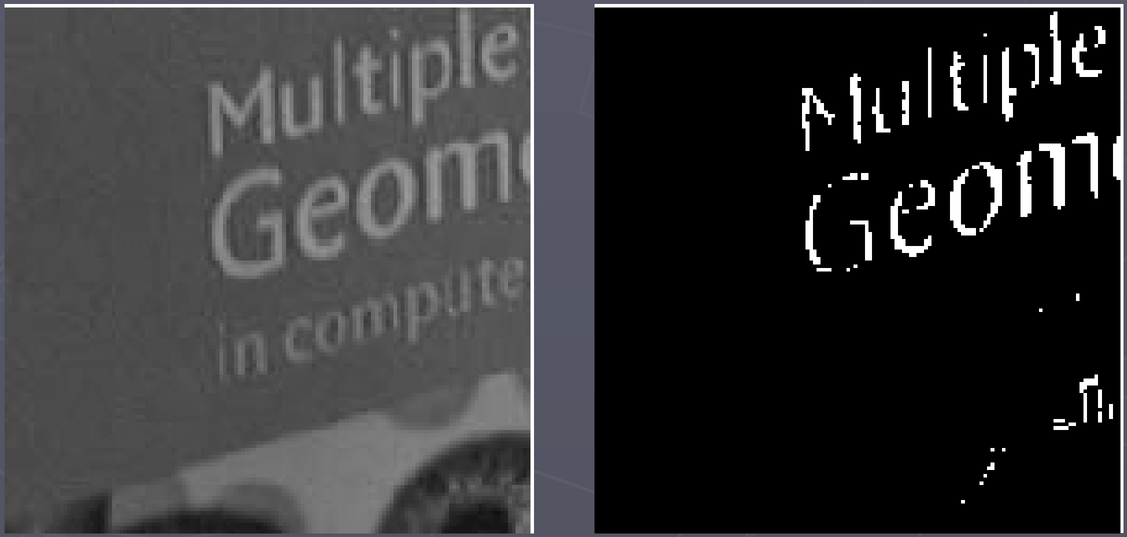
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



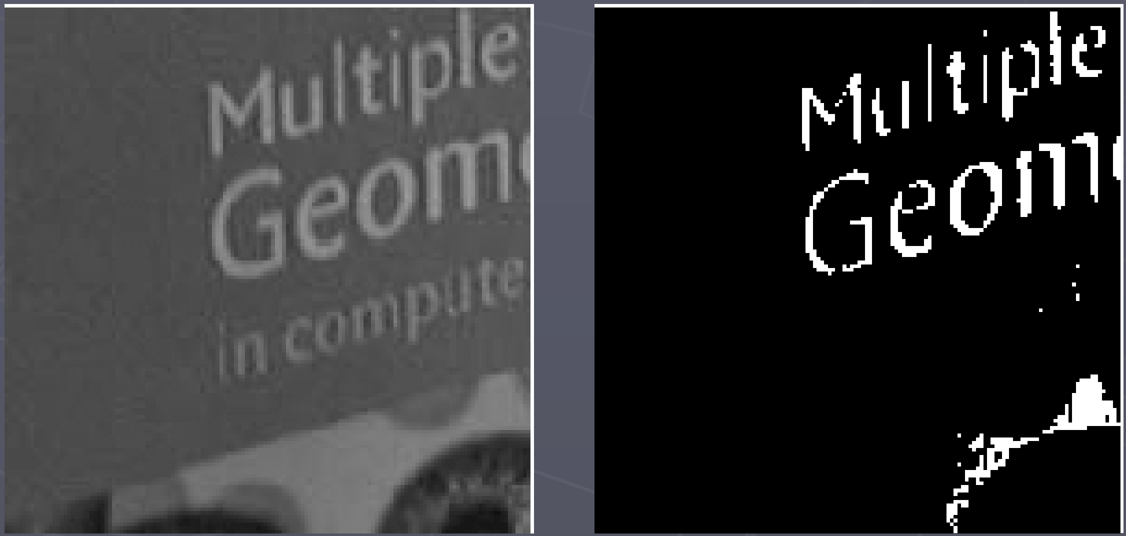
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



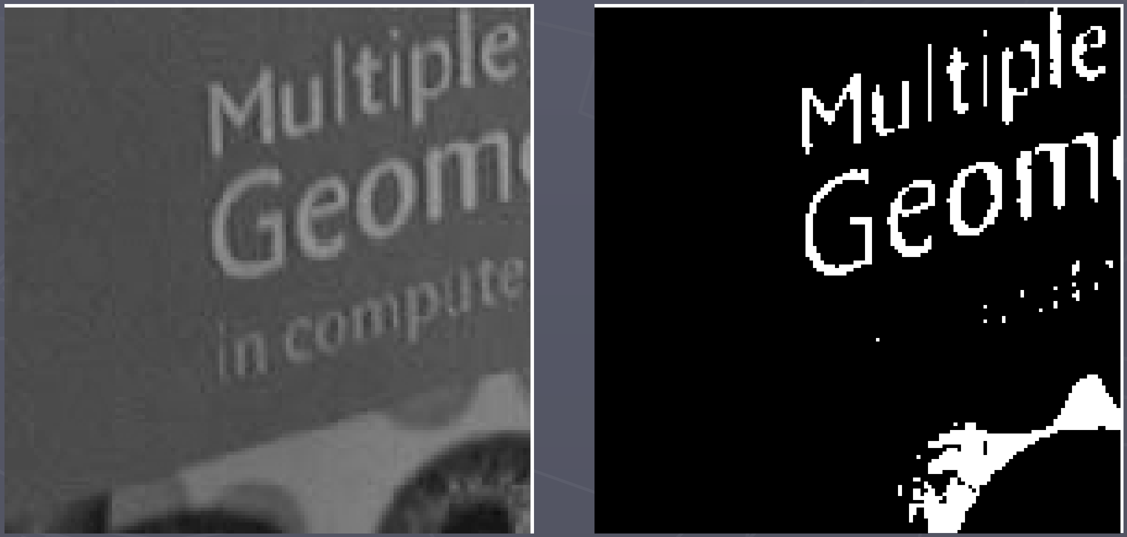
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



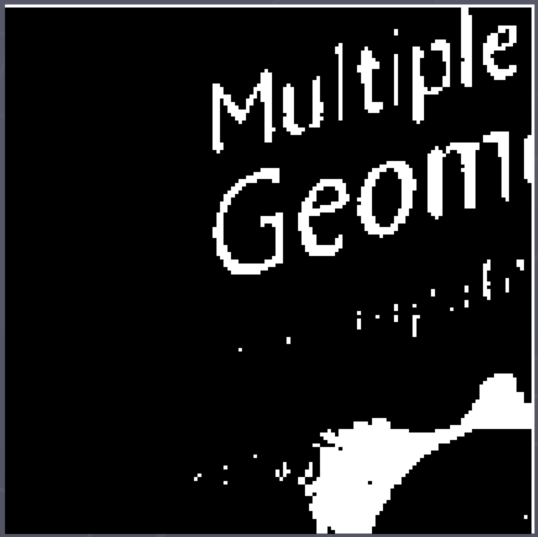
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



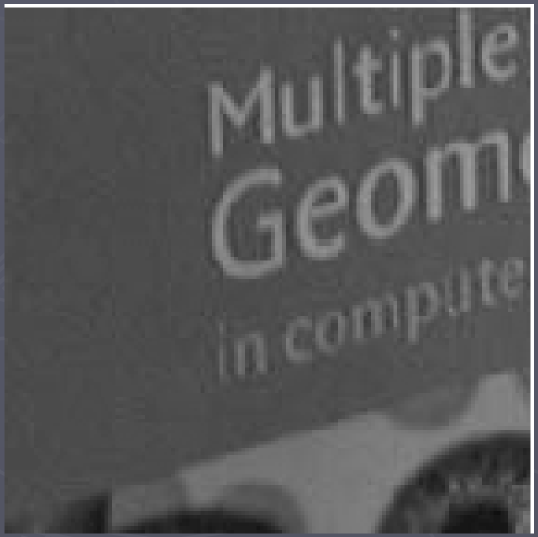
Matas: Maximally Stable Extremal Regions (MSERs)

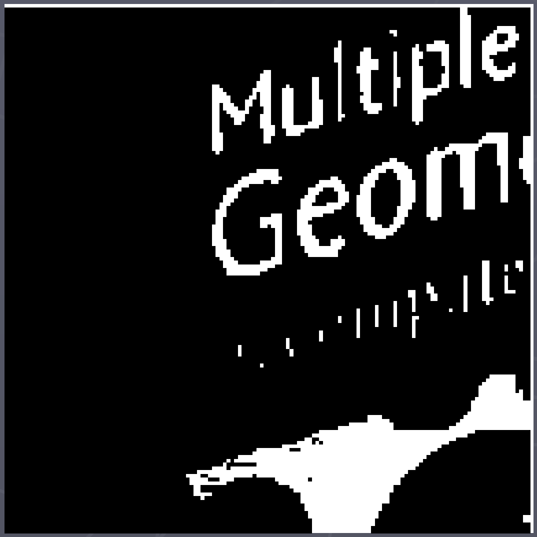
- Based on watershed algorithm



Matas: Maximally Stable Extremal Regions (MSERs)

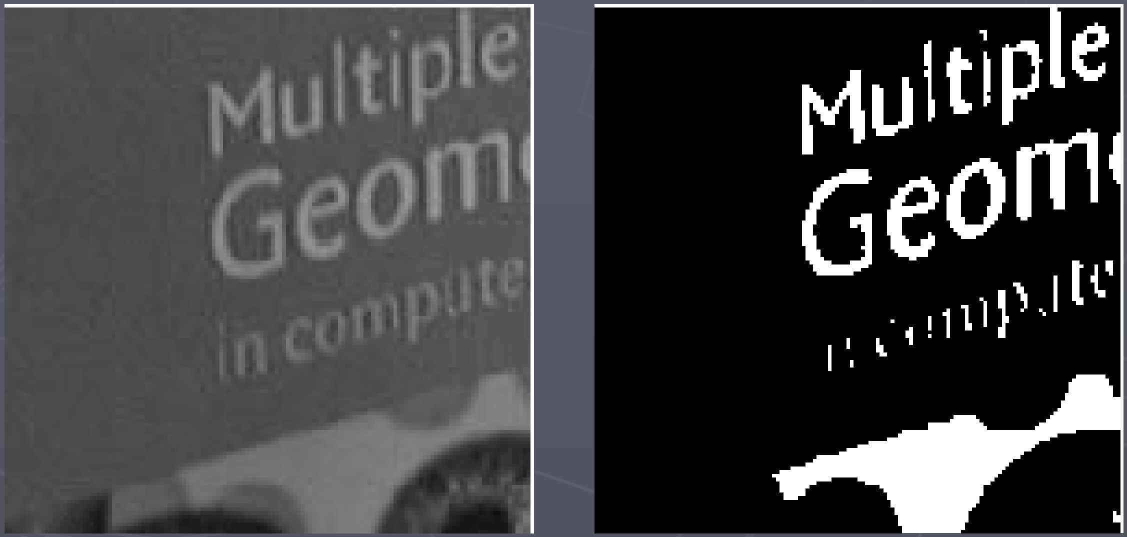
- Based on watershed algorithm





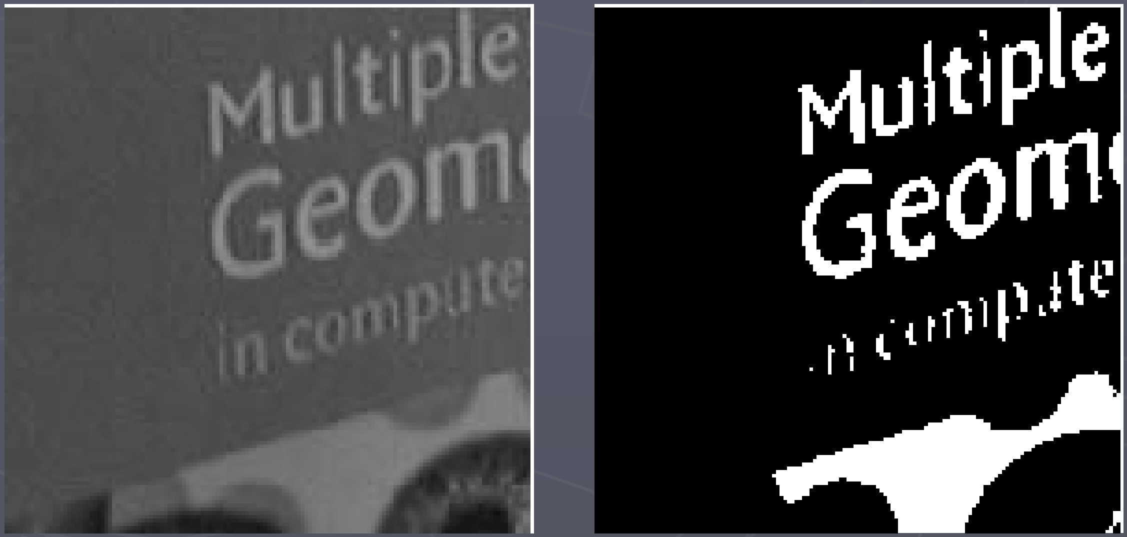
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



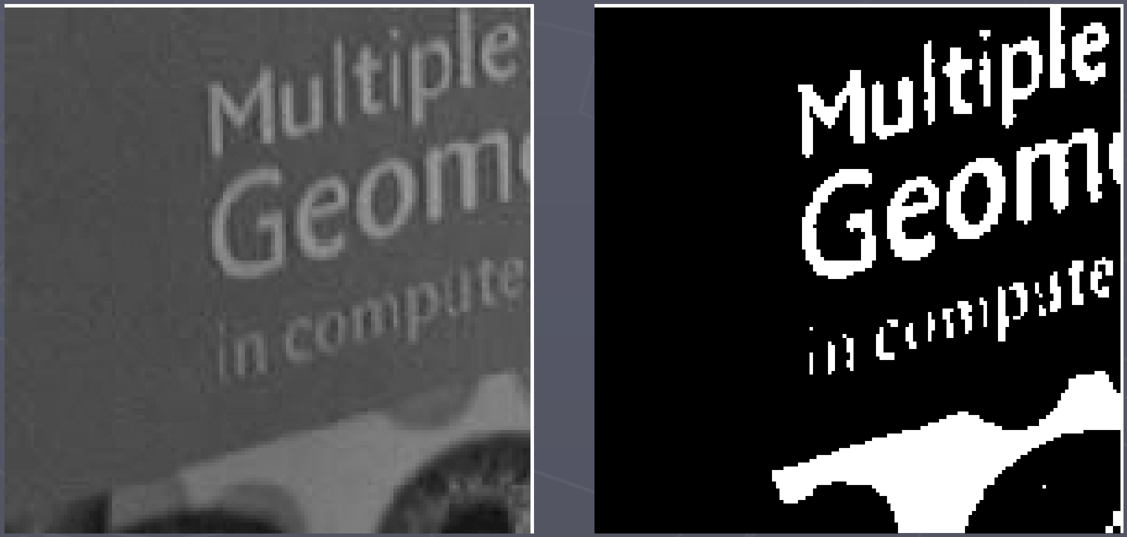
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



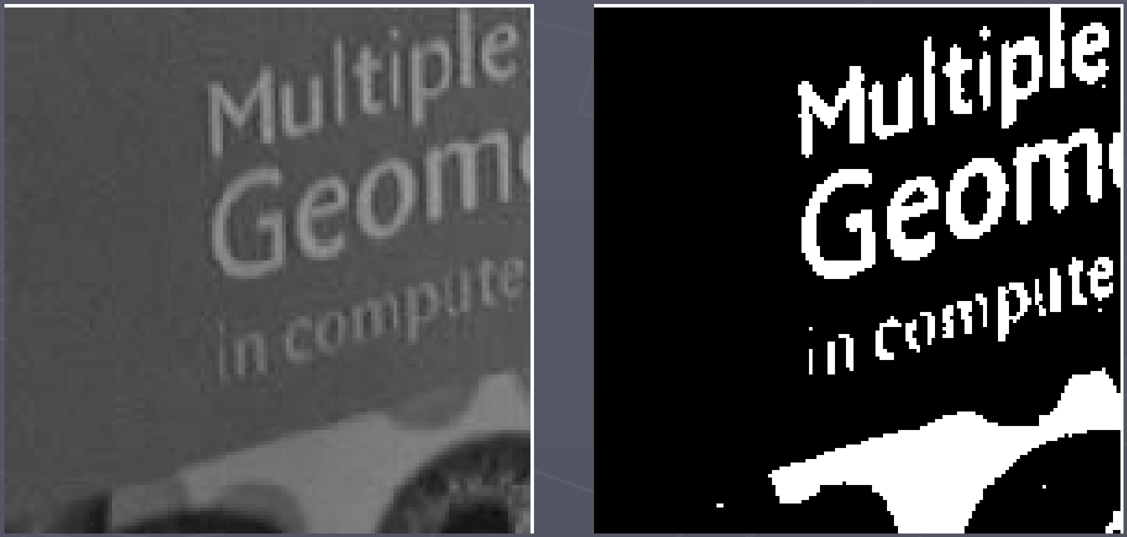
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



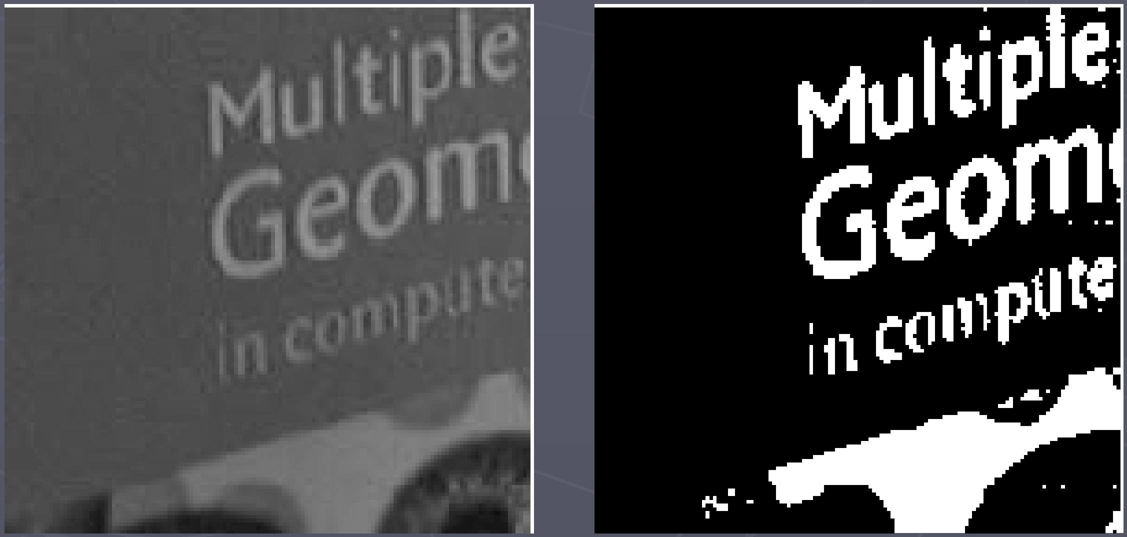
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



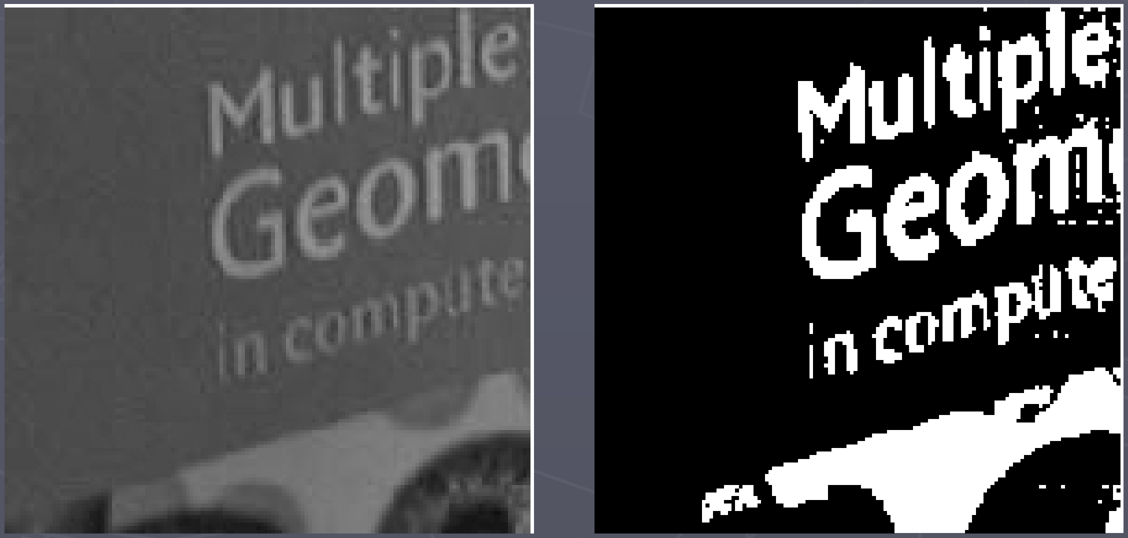
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



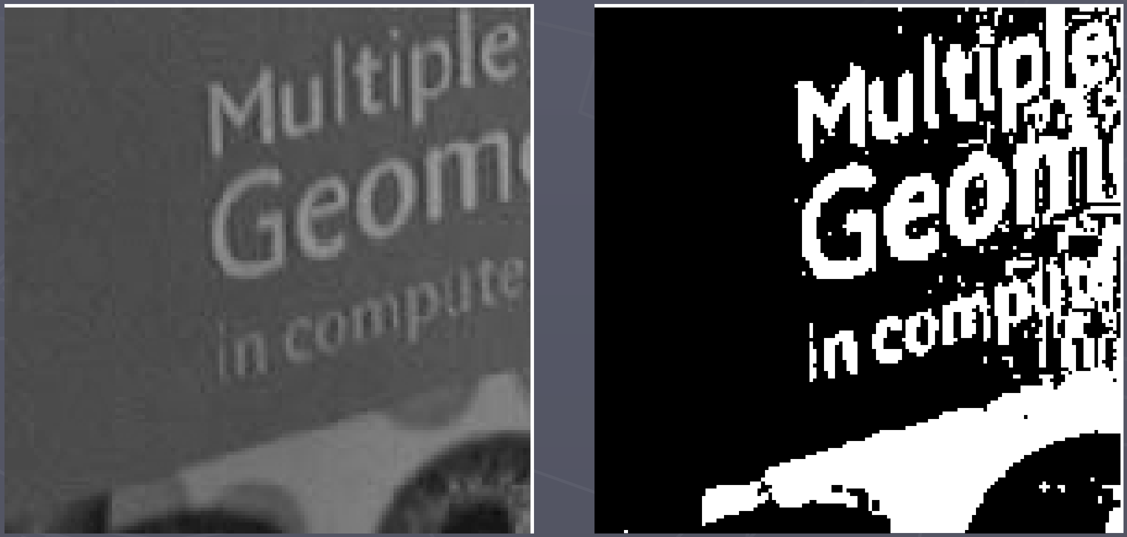
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



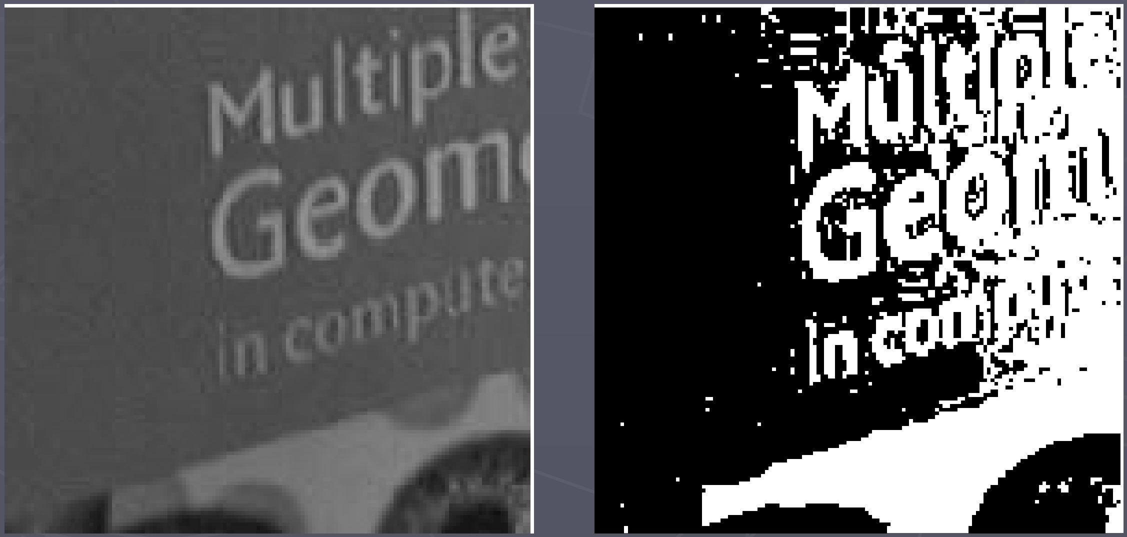
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



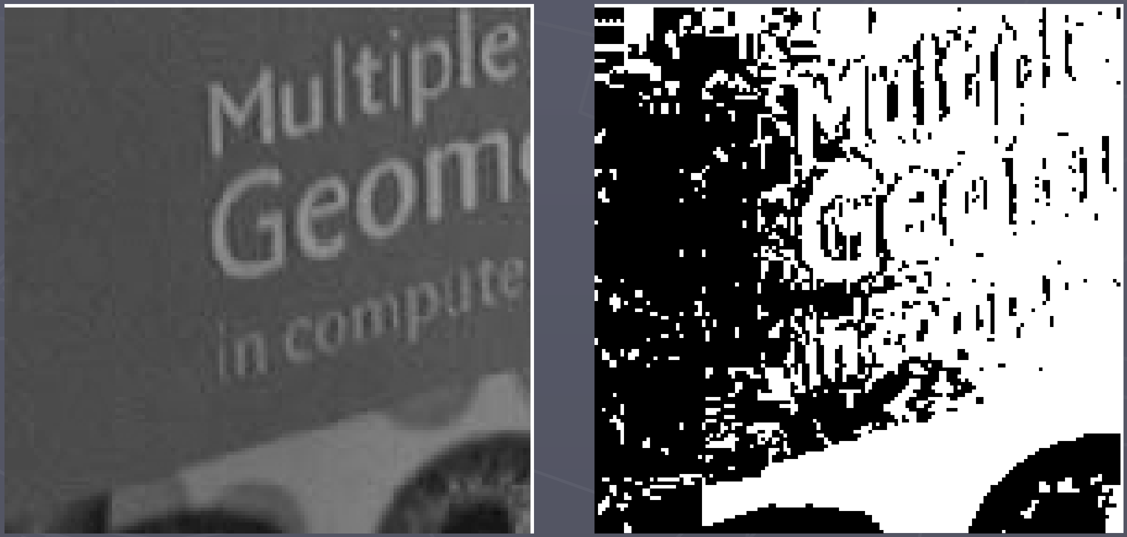
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



Matas: Maximally Stable Extremal Regions (MSERs)

- ***Extremal region: region such that***

$$\forall p \in Q, \forall q \in \delta Q : \begin{array}{l} I(p) > I(q) \\ I(p) < I(q) \end{array}$$

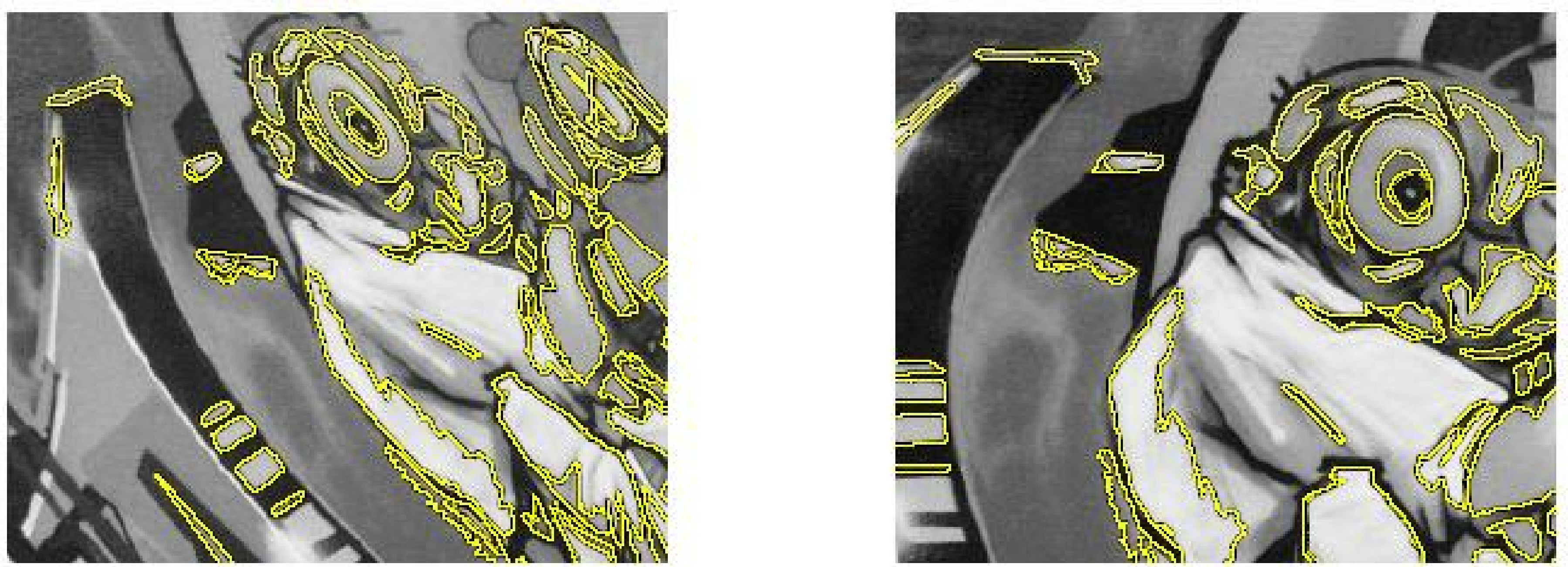
- ***Order regions***

$$Q_1 \subset \ldots \subset Q_i \subset Q_{i+1} \subset \ldots Q_n$$

- ***Maximally Stable Extremal Region:***
  local minimum of

$$q(i) = | Q_{i+\Delta} \setminus Q_{i-\Delta} | / Q_i$$

Maximally Stable Extremal Regions



Appreciation

Affine invariant

Detects blob-like structures

☺ Simple, efficient scheme

☺ High repeatability

😐 Fires on similar features as IBR
(regions need not be convex, but need to be closed)

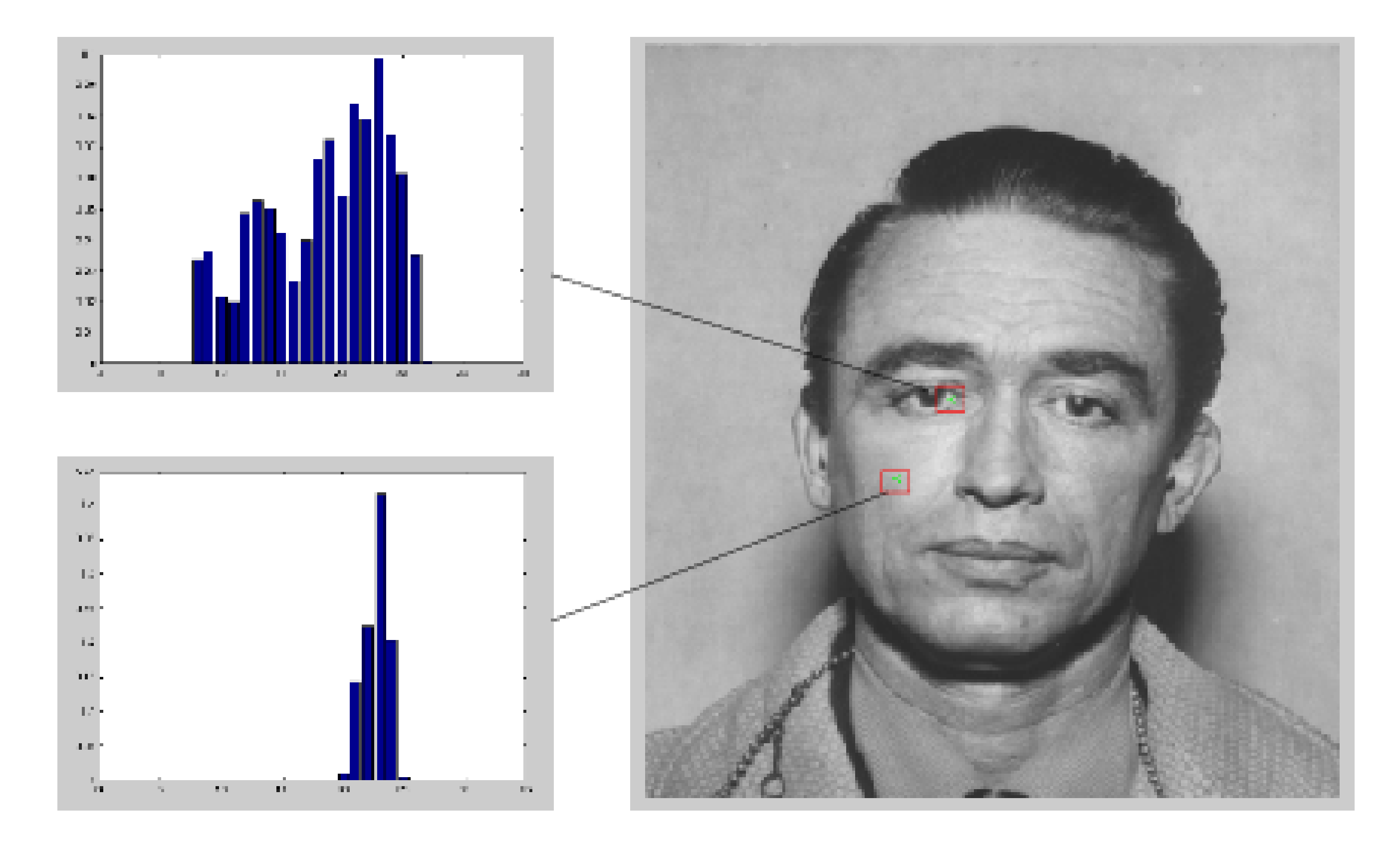
☹ Sensitive to image blur

Overview of existing detectors

- Lowe: DoG
- Lindeberg: scale selection
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- Others

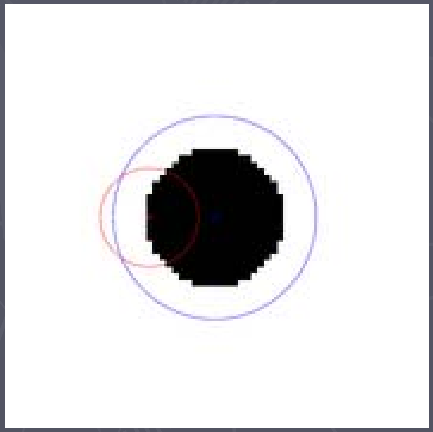
Kadir & Brady’s salient regions

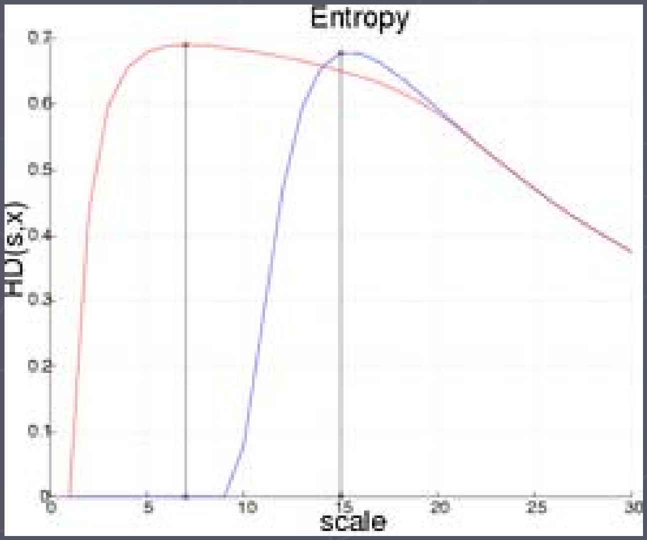
- Based on entropy

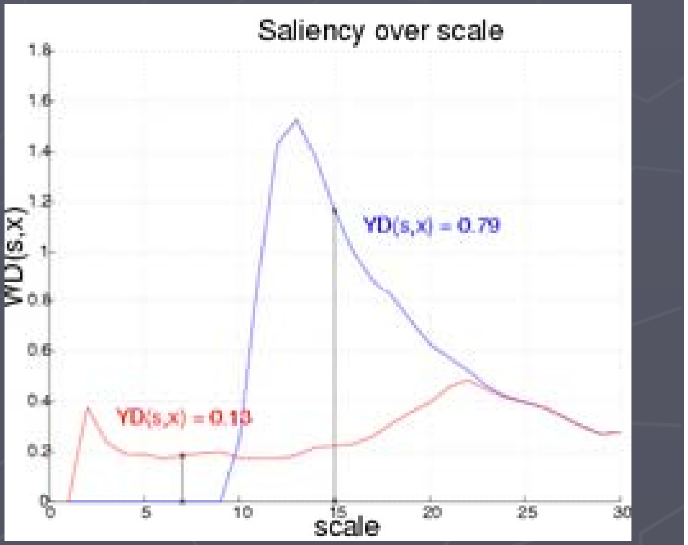


Kadir & Brady’s salient regions

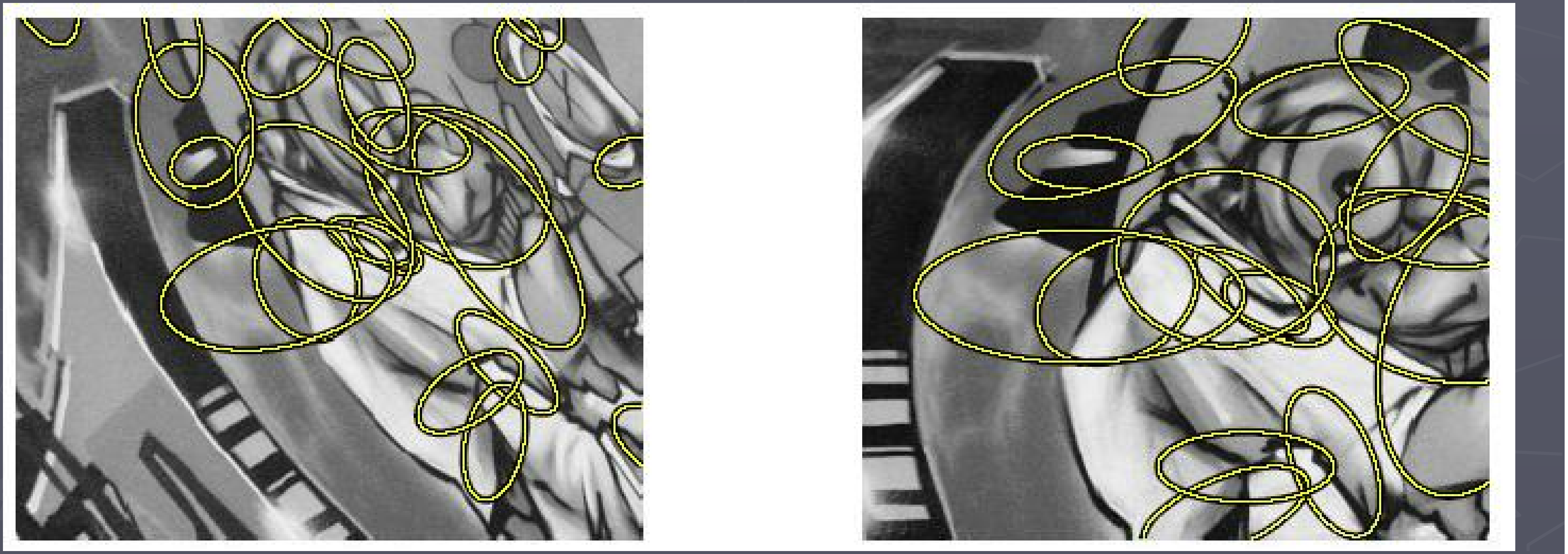
- Maxima in entropy, combined with inter-scale saliency
- Extended to affine invariance







Salient regions



Appreciation

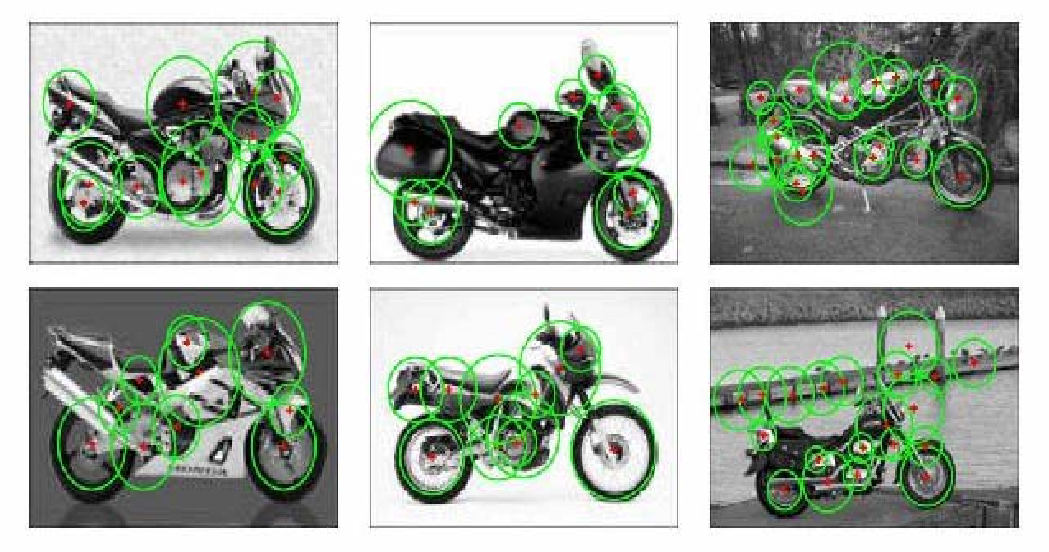
Scale or affine invariant

Detects blob-like structures

🙂 very good for object class recognition

😐 limited number of regions

🙁 slow to extract



Overview of existing detectors

- Lowe: DoG
- Lindeberg: scale selection
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- Others

Other feature detectors

- Edge-based detectors
  - Jurie et al., Mikolajczyk et al., ...
- Combinations of small-scale features
  - Brown & Lowe
- Vertical line segments
  - Goedeme et al.
- Speeded-Up Robust Features (SURF)
  - Bay et al.

SURF: Speeded Up Robust Features

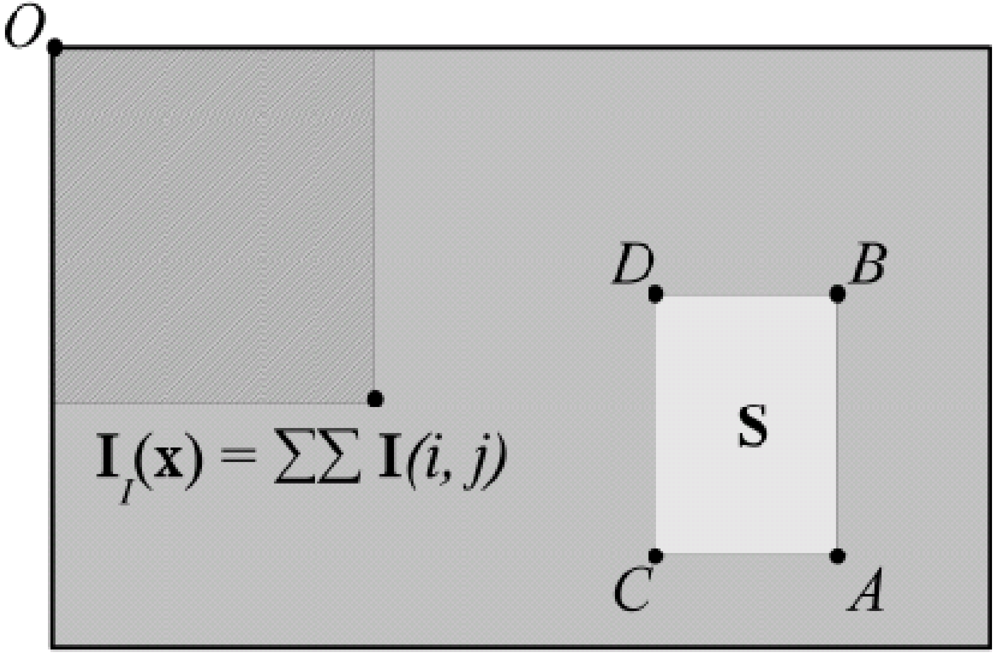
Herbert Bay[1], Tinne Tuytelaars[2], and Luc Van Gool[12]

[1] ETH Zurich
{bay, vangool}@vision.ee.ethz.ch
[2] Katholieke Universiteit Leuven
{Tinne.Tuytelaars, Luc.Vangool}@esat.kuleuven.be

Abstract. In this paper, we present a novel scale- and rotation-invariant interest point detector and descriptor, coined SURF (Speeded Up Robust Features). It approximates or even outperforms previously proposed schemes with respect to repeatability, distinctiveness, and robustness, yet can be computed and compared much faster.
This is achieved by relying on integral images for image convolutions; by building on the strengths of the leading existing detectors and descriptors (*in casu*, using a Hessian matrix-based measure for the detector, and a distribution-based descriptor); and by simplifying these methods to the essential. This leads to a combination of novel detection, description, and matching steps. The paper presents experimental results on a standard evaluation set, as well as on imagery obtained in the context of a real-life object recognition application. Both show SURF's strong performance.

Methodology

- Using integral images for major speed up
  - **Integral Image (summed area tables)** is an intermediate representation for the image and contains the **sum of gray scale pixel values of image**
  - Second order derivative and Haar-wavelet response



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

Cost four additions operation only

Slide Credit: Bay, Tuletaars, Van Gool, Wyman

Detection

- Hessian-based interest point localization

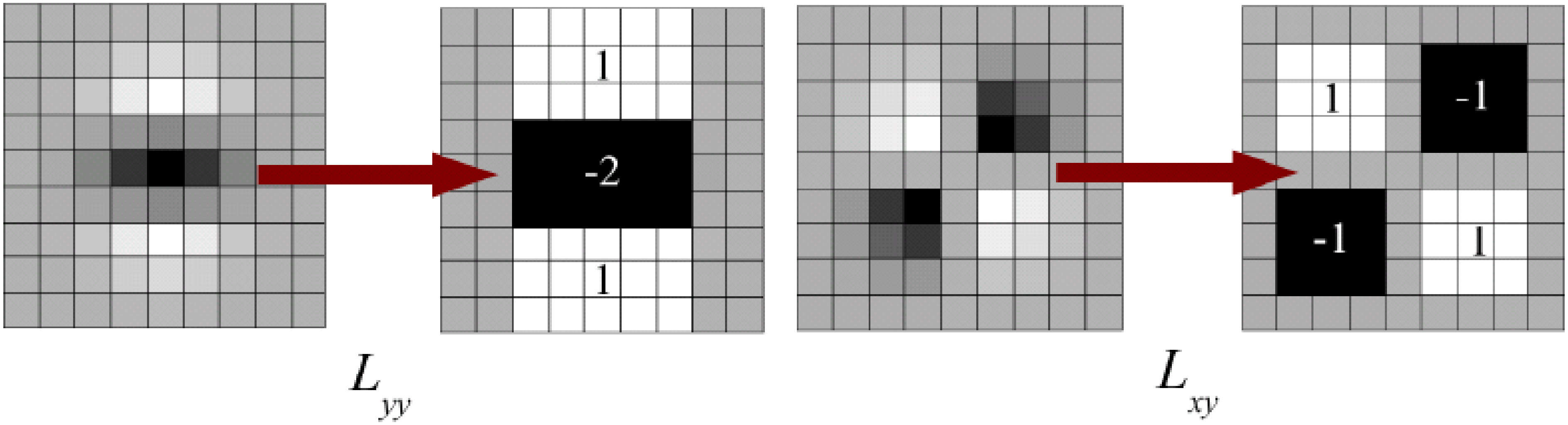
$$H = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix}$$

- $L_{xx}(x,y,\sigma)$ is the **Laplacian of Gaussian** of the image
- It is the convolution of the *Gaussian* second order derivative with the image
- Lindeberg showed Gaussian function is optimal for scale-space analysis
- This paper argues that Gaussian is overrated since the property that no new structures can appear while going to lower resolution is not proven in 2D case

Slide Credit: Bay, Tuletaars, Van Gool, Wyman

Detection

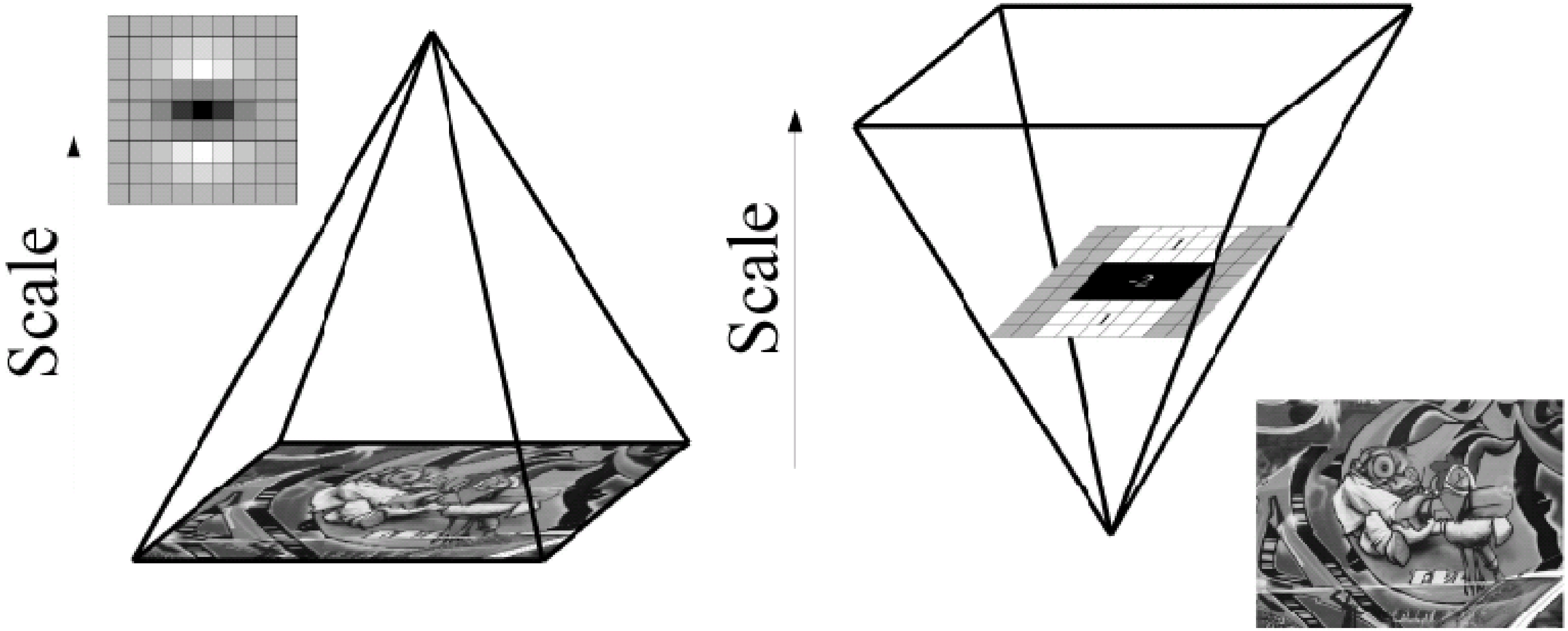
- Approximated second order derivatives with box filters (mean/average filter)



Slide Credit: Bay, Tuletaars, Van Gool, Wyman

Detection

- Scale analysis with constant image size



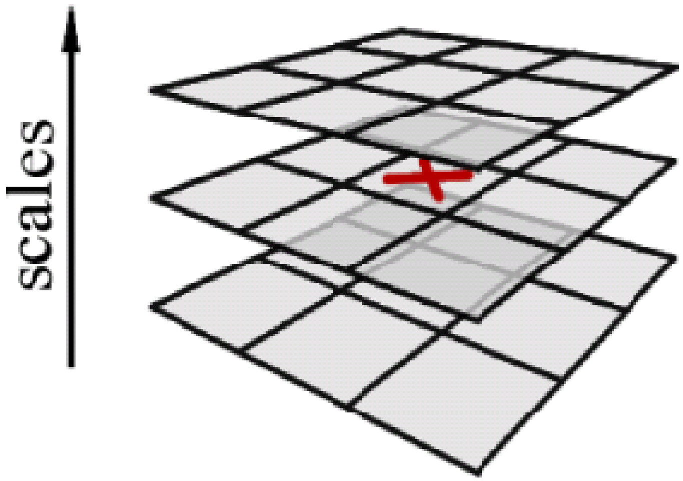
9 x 9, 15 x 15, 21 x 21, 27 x 27 → 39 x 39, 51 x 51 …

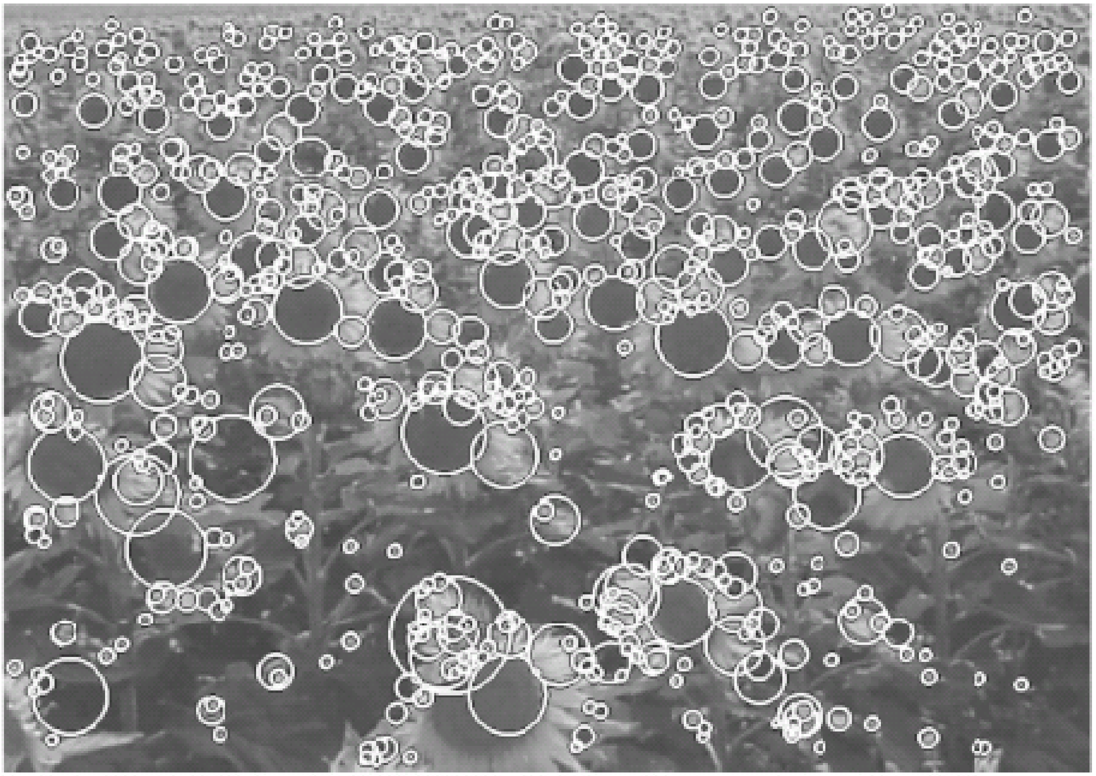
1st octave ***2nd octave***

Slide Credit: Bay, Tuletaars, Van Gool, Wyman

Detection

- Non-maximum suppression and interpolation
  - Blob-like feature detector

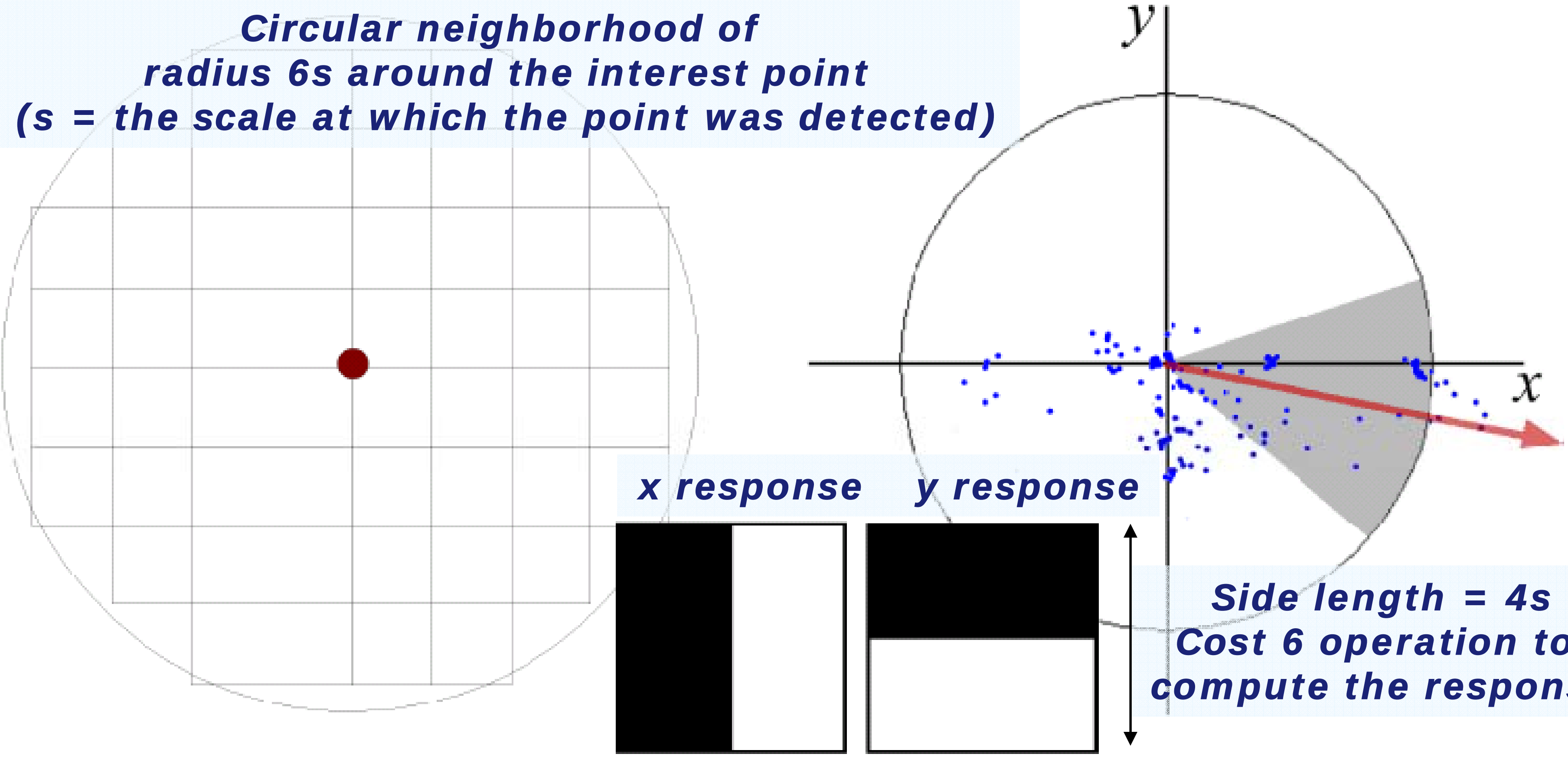




Slide Credit: Bay, Tuletaars, Van Gool, Wyman

Description

- Orientation Assignment



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

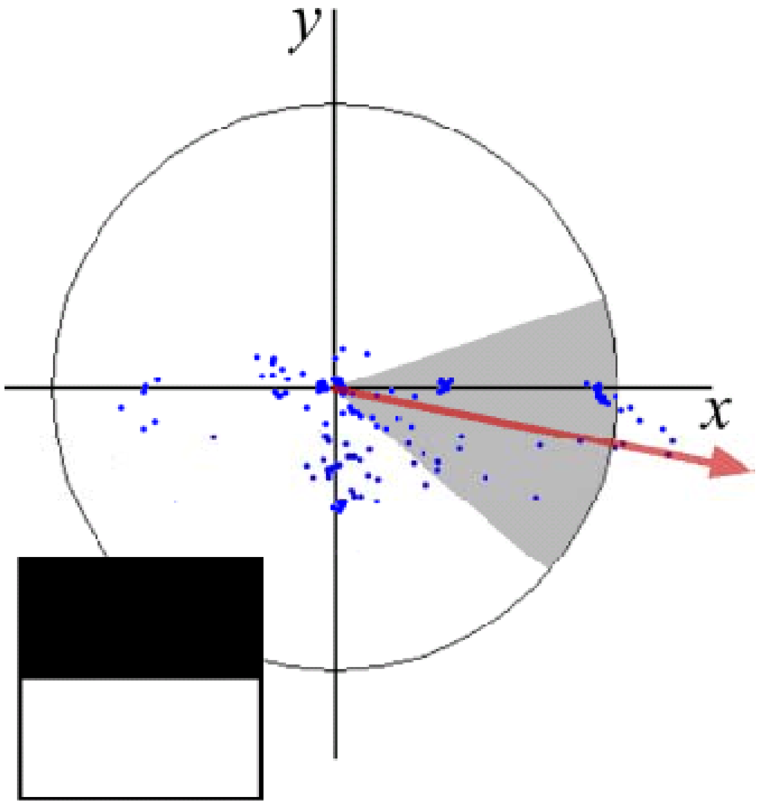
x response ***y response***

Side length = 4s Cost 6 operation to compute the response

Slide Credit: Bay, Tuletaars, Van Gool, Wyman

Description

- Dominant orientation
  - The Haar wavelet responses are represented as vectors
  - Sum all responses within a sliding orientation window covering an angle of 60 degree
  - The two summed response yield a new vector
  - **The longest vector** is the dominant orientation
  - Second longest is … **ignored**



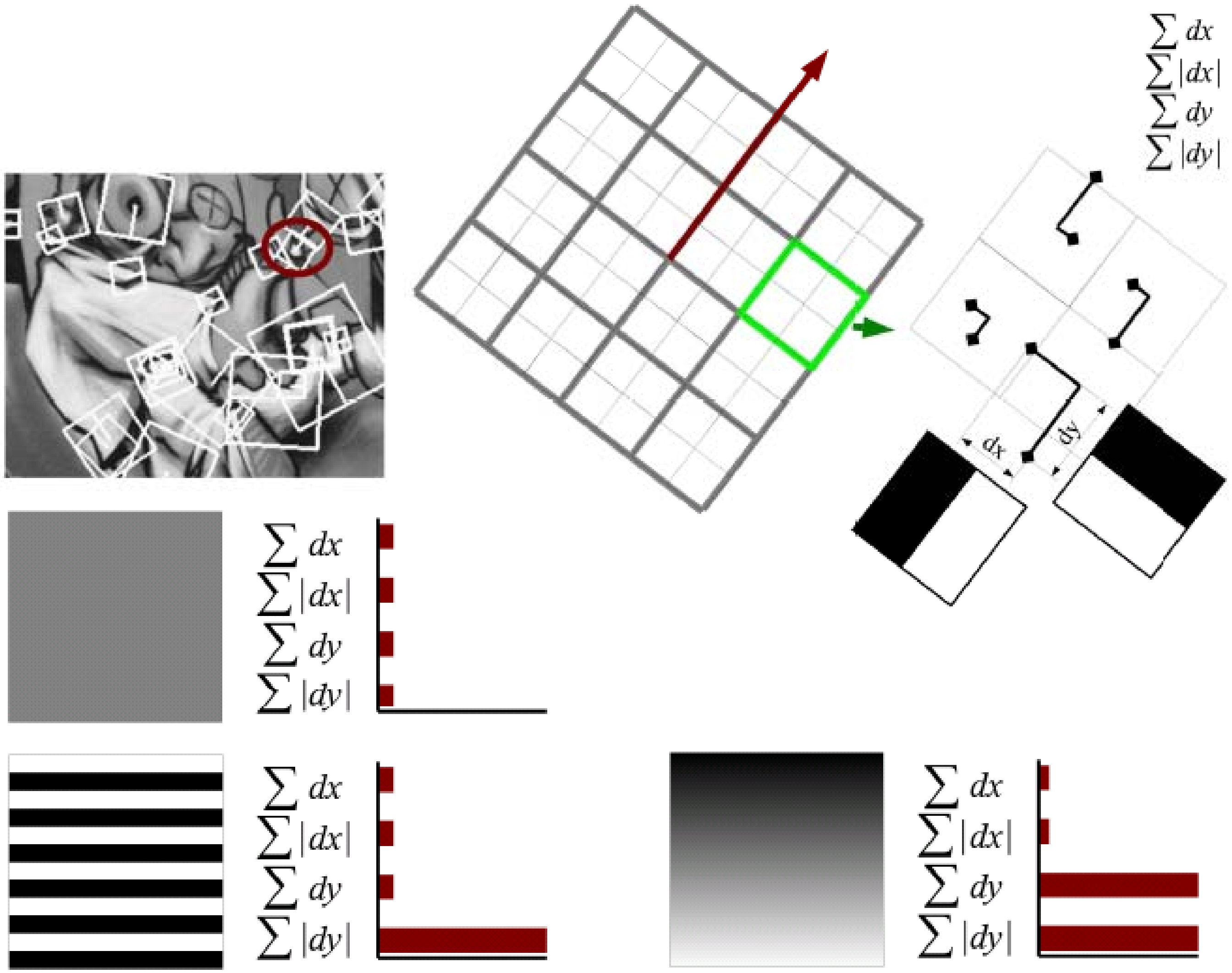
Slide Credit: Bay, Tuletaars, Van Gool, Wyman

Description

- Split the interest region up into 4 x 4 square sub-regions with 5 x 5 regularly spaced sample points inside
- Calculate Haar wavelet response d_x and d_y
- Weight the response with a Gaussian kernel centered at the interest point
- Sum the response over each sub-region for d_x and d_y separately → **feature vector of length 32**
- In order to bring in information about the polarity of the intensity changes, extract the sum of absolute value of the responses → **feature vector of length 64**
- Normalize the vector into unit length

Slide Credit: Bay, Tuletaars, Van Gool, Wyman

Description



Slide Credit: Bay, Tuletaars, Van Gool, Wyman

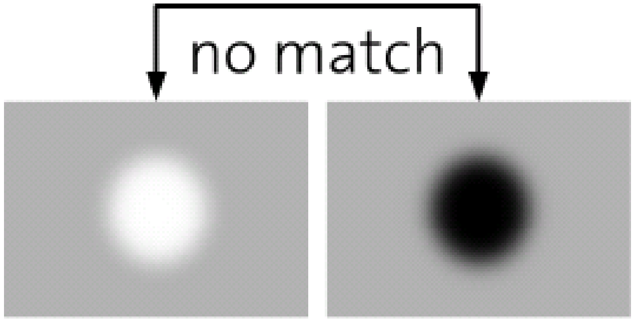
Description

- SURF-128
  - The sum of d_x and $|d_x|$ are computed separately for $d_y < 0$ and $d_y > 0$
  - Similarly for the sum of d_y and $|d_y|$
  - This doubles the length of a feature vector

Slide Credit: Bay, Tuletaars, Van Gool, Wyman

Matching

- Fast indexing through the sign of the Laplacian for the underlying interest point
  - The sign of trace of the Hessian matrix
  - Trace = $L_{xx} + L_{yy}$



- Either 0 or 1 (Hard thresholding, may have boundary effect …)
- In the matching stage, compare features if they have the same type of contrast (sign)

Slide Credit: Bay, Tuletaars, Van Gool, Wyman

Table 1. Thresholds, number of detected points and calculation time for the detectors in our comparison. (First image of Graffiti scene, 800 × 640).

detector	threshold	nb of points	comp. time (msec)
Fast-Hessian	600	1418	120
Hessian-Laplace	1000	1979	650
Harris-Laplace	2500	1664	1800
DoG	default	1520	400

Table 2. Computation times for the joint detector - descriptor implementations, tested on the first image of the Graffiti sequence. The thresholds are adapted in order to detect the same number of interest points for all methods. These relative speeds are also representative for other images.

	U-SURF	SURF	SURF-128	SIFT
time (ms):	255	354	391	1036

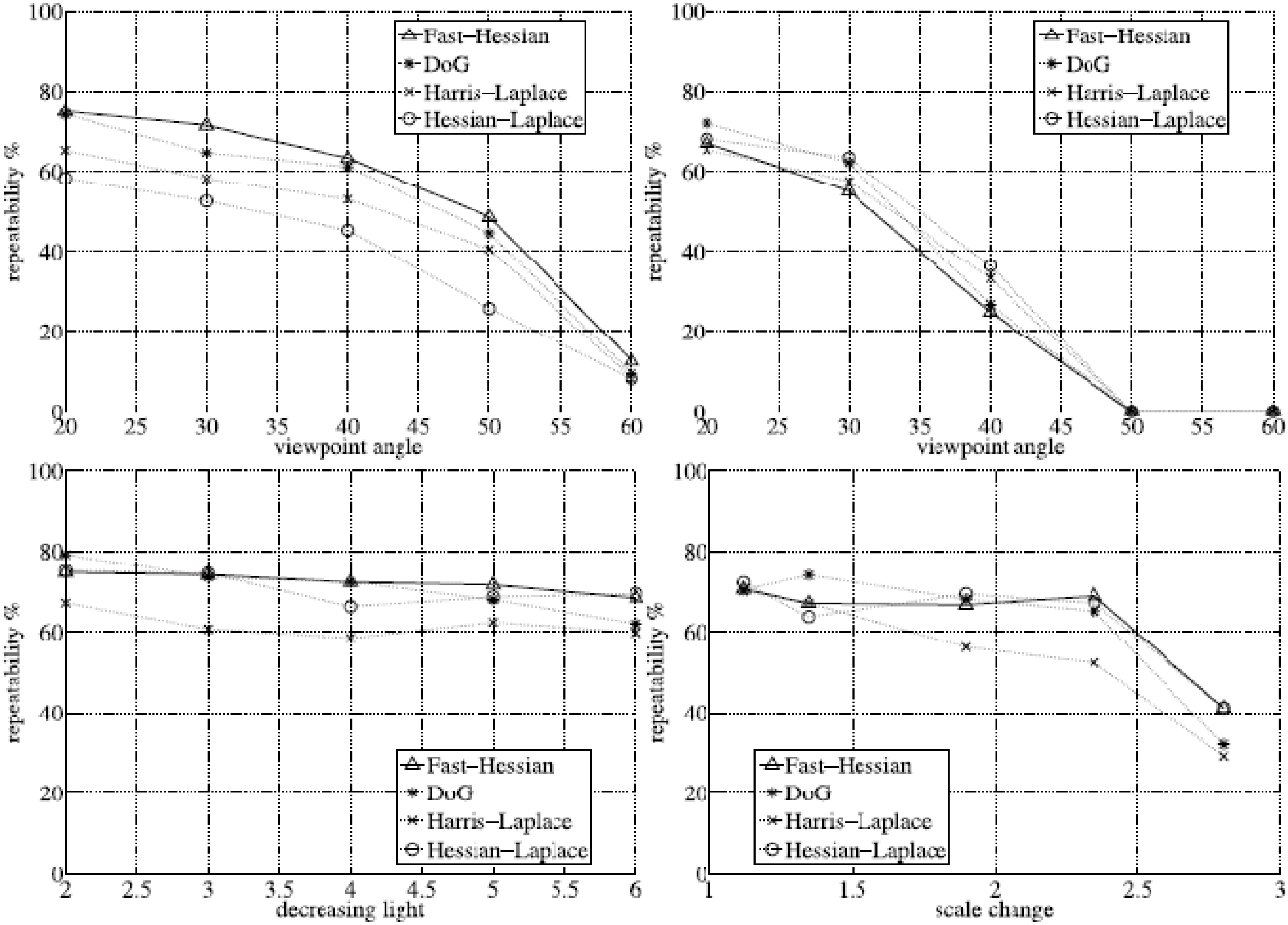


Fig. 6. Repeatability score for image sequences, from left to right and top to bottom, Wall and Graffiti (Viewpoint Change), Leuven (Lighting Change) and Boat (Zoom and Rotation)

Overview

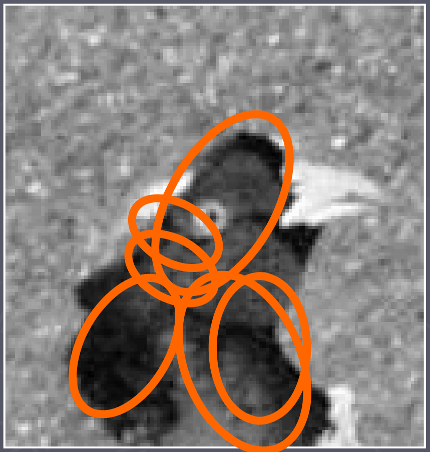
- ►Local Invariant Features: What? Why?
  - Introduction
  - Overview of existing detectors
  - Quantitative and qualitative comparison
- ►Local Invariant Features: When? How?
  - Feature descriptors
  - Applications
  - Conclusions

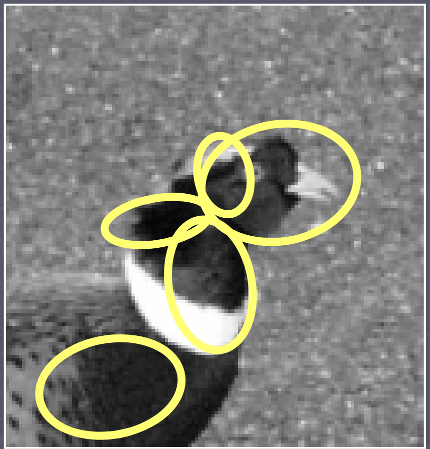
Quantitative comparisons

- Evaluation of interest points (Schmid & Mohr, ICCV98)
- Evaluation of descriptors (Mikolajczyk & Schmid, CVPR03)
- Evaluation of affine invariant features (Mikolajczyk et al., PAMI05)
- Evaluation on 3D objects (Moreels & Perona, ICCV05)
- Evaluation on 3D objects (Fraundorfer & Bischof, ICCV05)
- Evaluation in the context of object class recognition (Mikolajczyk et al., ICCV05)

Evaluation criteria: repeatability

- Repeatability rate : percentage of corresponding points

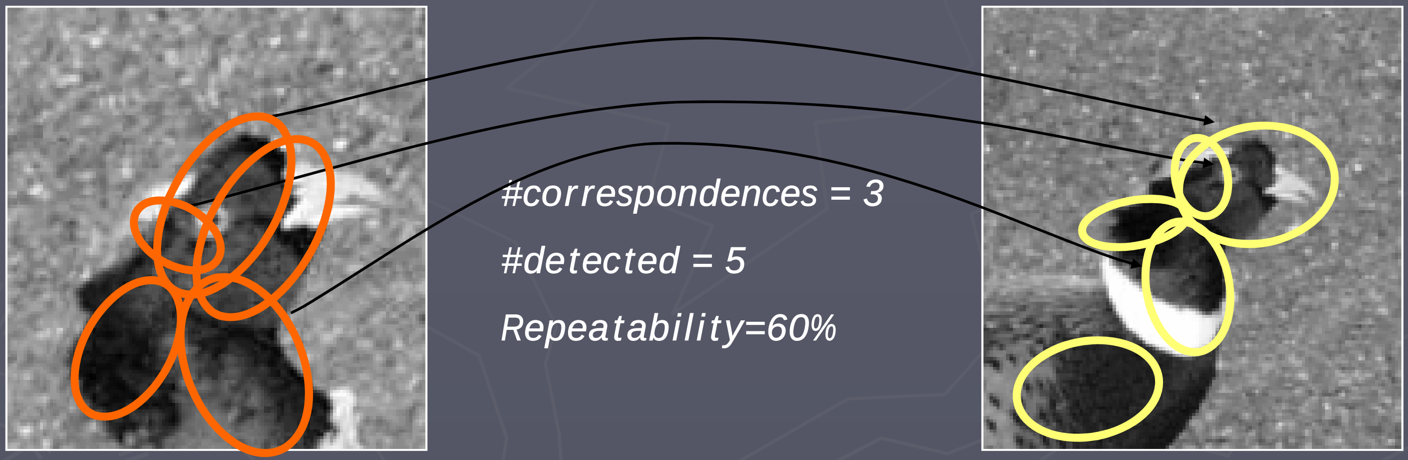




$$\mathit{repeatabil\ ity} = \frac{\#\mathit{correspond\ ences}}{\#\mathit{detected}} \cdot 100\%$$

Evaluation criteria: repeatability

- Repeatability rate : percentage of corresponding points



#correspondences = 3

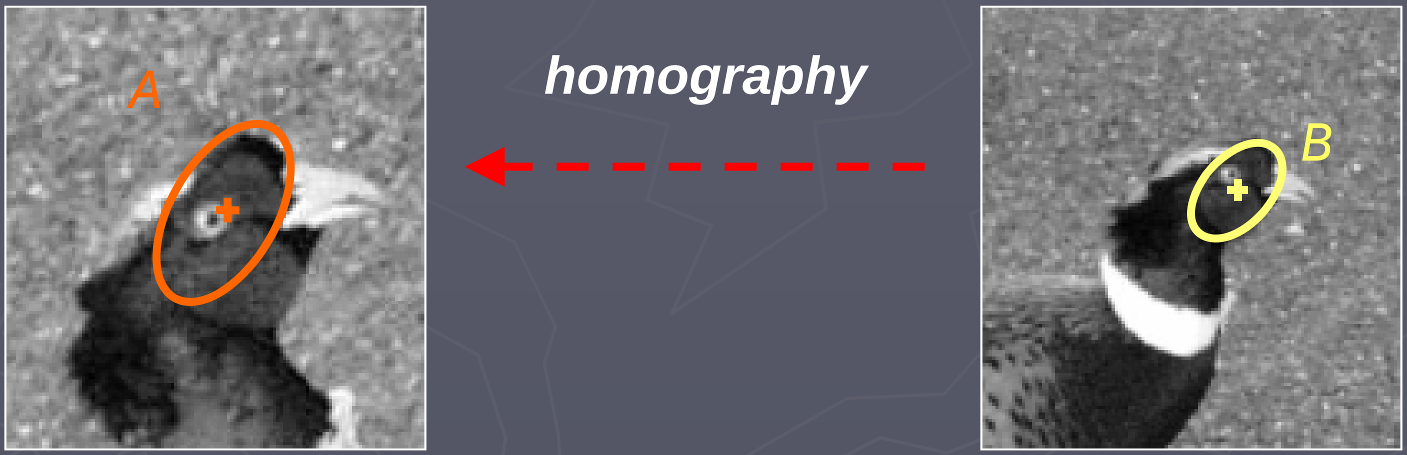
#detected = 5

Repeatability=60%

$$repeatability = \frac{\#\,correspondences}{\#\,detected} \cdot 100\%$$

Evaluation criteria: repeatability

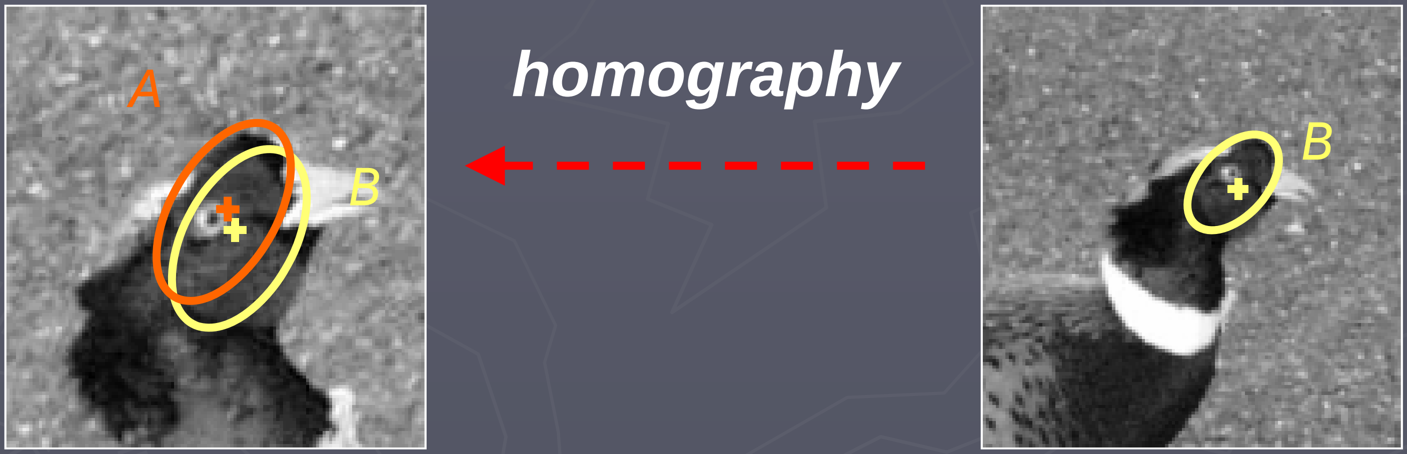
- Repeatability rate : percentage of corresponding points



homography

Evaluation criteria: repeatability

- Repeatability rate : percentage of corresponding points

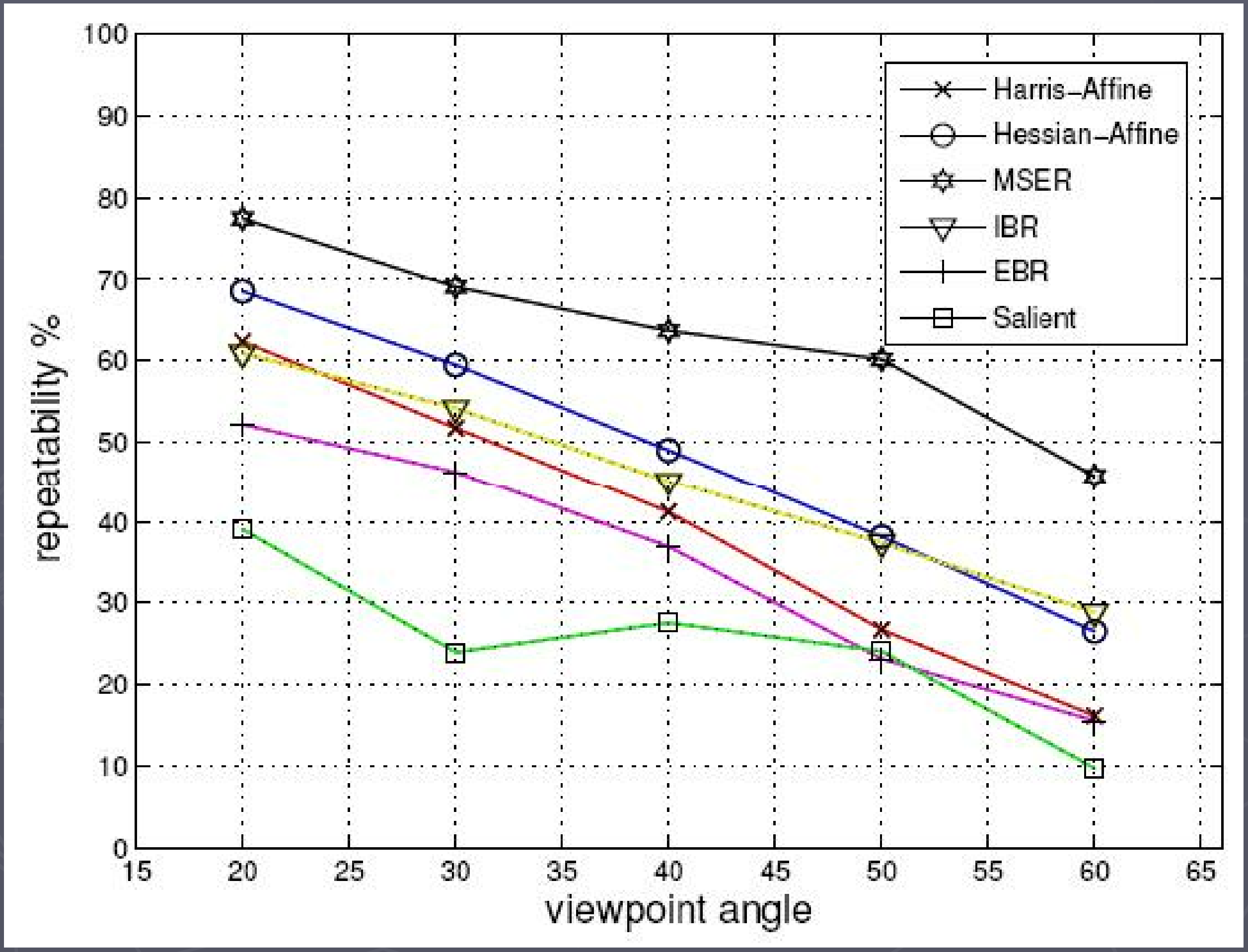


homography

- *Two points are corresponding if* $\frac{A \cap B}{A \cup B} > T$

  T=60%

Repeatability



Quantitative evaluation

- Repeatability often lower than 50%
- Performance often depends on scene type, different detectors are complementary
- Number of detected features varies greatly
- Accuracy of detected features varies
- Performance depends on application
- Speed

Qualitative Comparison

- Difficult to declare a ‘winner’
- Different methods are complementary
- ‘Best features’ depends on application:
  - Level of invariance needed
  - Number/density of features wanted
  - Typical scene types
  - Accuracy of features
  - Generalization power of features
  - ...

Matching Local Self-Similarities across Images and Videos

Eli Shechtman Michal Irani
Dept. of Computer Science and Applied Math
The Weizmann Institute of Science
76100 Rehovot, Israel

Abstract

We present an approach for measuring similarity between visual entities (images or videos) based on matching internal self-similarities. What is correlated across images (or across video sequences) is the internal layout of local self-similarities (up to some distortions), even though the patterns generating those local self-similarities are quite different in each of the images/videos. These internal self-similarities are efficiently captured by a compact local "self-similarity descriptor", measured densely throughout the image/video, at multiple scales, while accounting for local and global geometric distortions. This gives rise to matching capabilities of complex visual data, including detection of objects in real cluttered images using only rough hand-sketches, handling textured objects with no clear boundaries, and detecting complex actions in cluttered video data with no prior learning. We compare our measure to commonly used image-based and video-based similarity measures, and demonstrate its applicability to object detection, retrieval, and action detection.

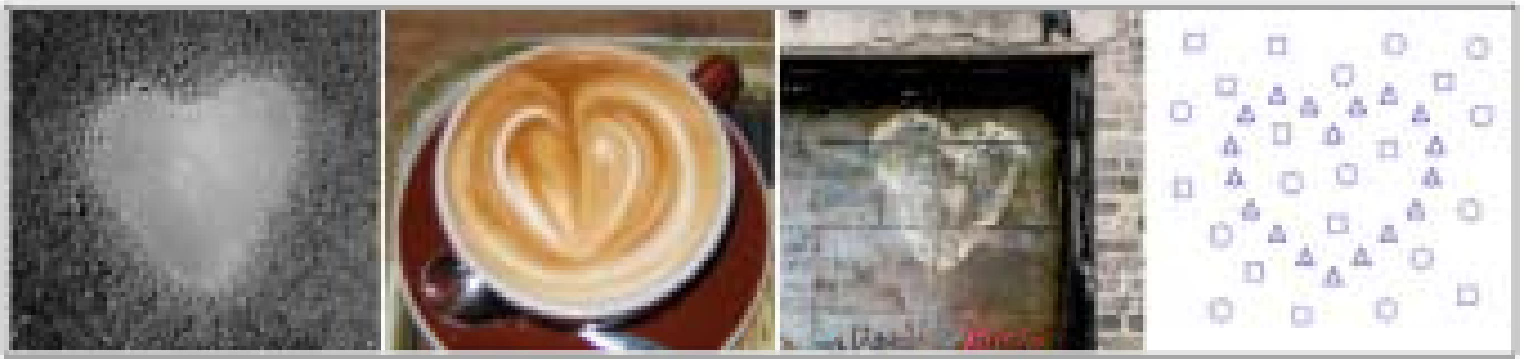
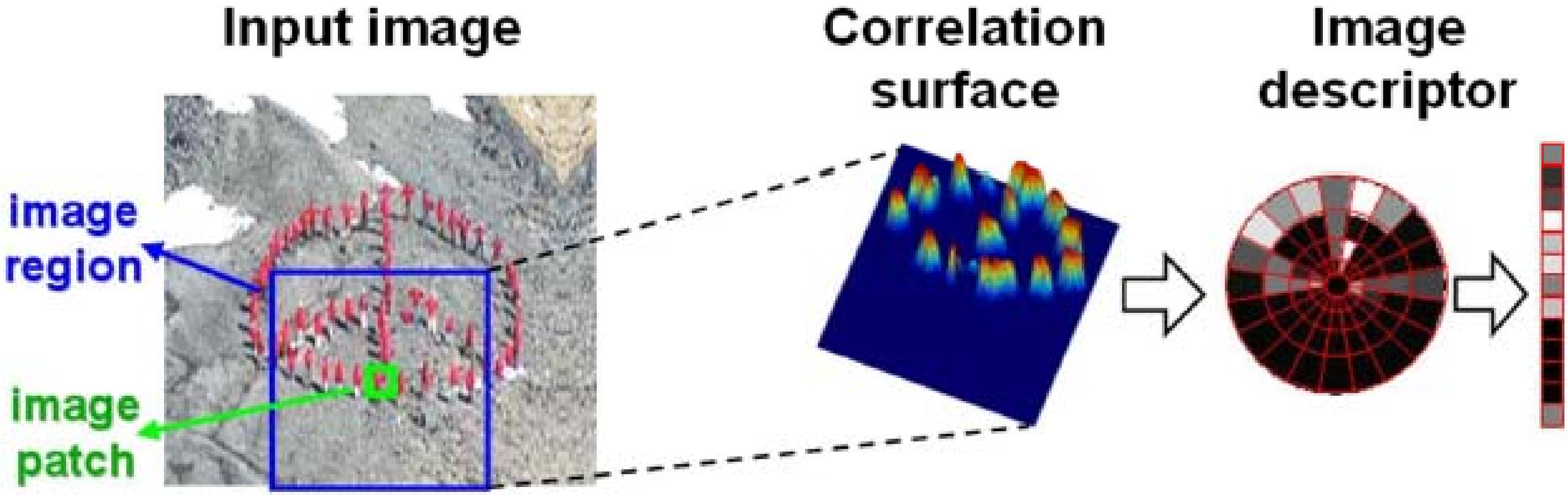


Figure 1. *These images of the same object (a heart) do NOT share common image properties (colors, textures, edges), but DO share a similar geometric layout of local internal self-similarities.*


Input image
Correlation surface
Image descriptor
image region
image patch

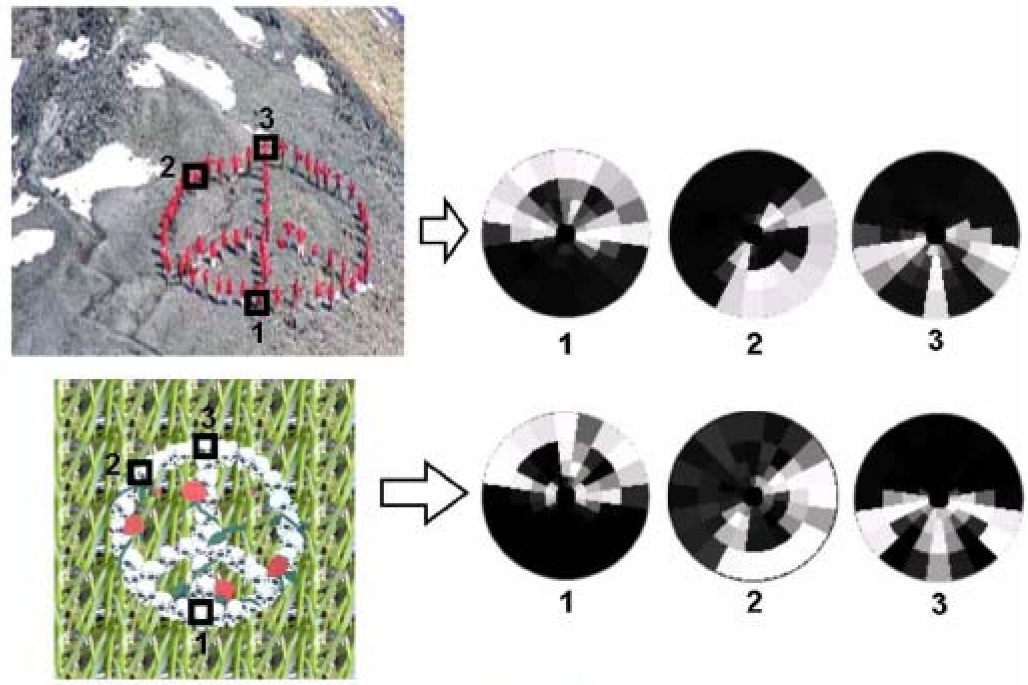


Figure 3. **Corresponding "Self-similarity descriptors".** *We show a few corresponding points (1,2,3) across two images of the same object, with their "self-similarity" descriptors. Despite the large difference in photometric properties between the two images, their corresponding "self-similarity" descriptors are quite similar.*

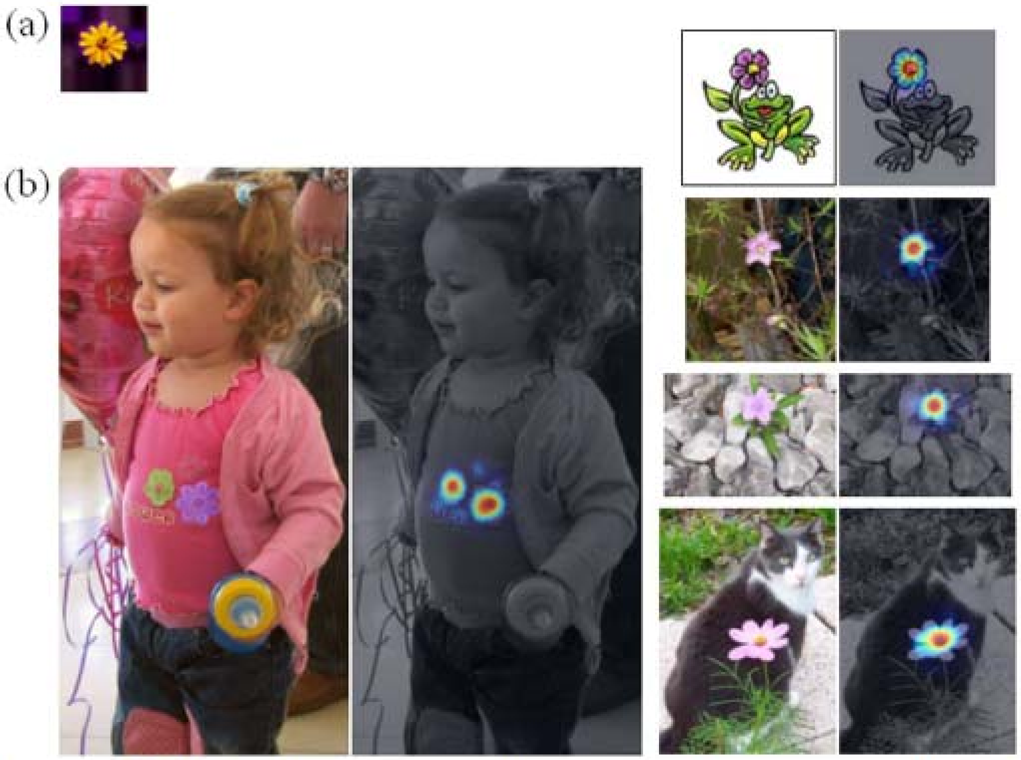


Figure 4. **Object detection.** *(a) A single template image (a flower). (b) The images against which it was compared with the corresponding detections. The continuous likelihood values above a threshold (same threshold for all images) are shown superimposed on the gray-scale images, displaying detections of the template at correct locations (red corresponds to the highest values).*

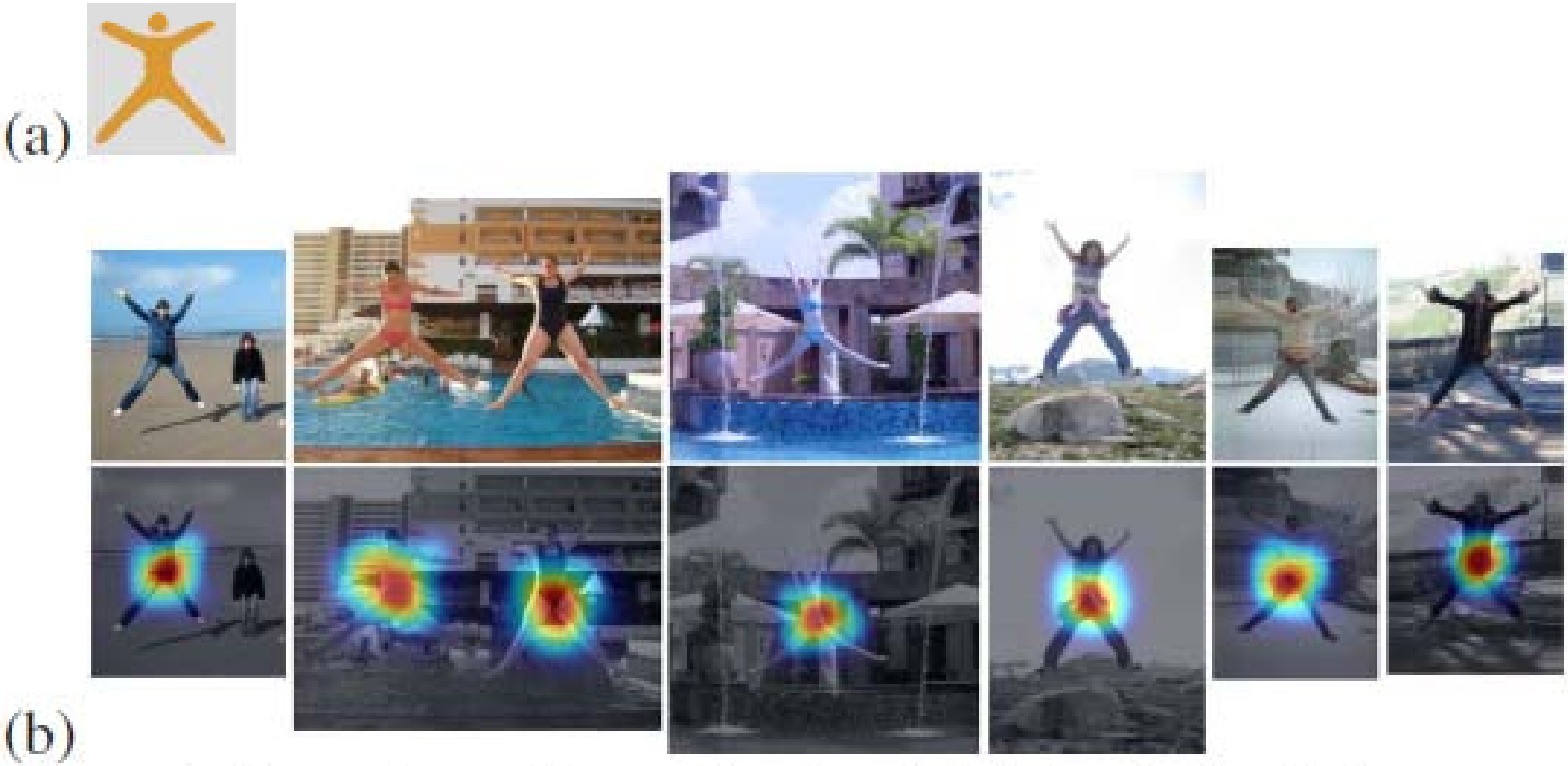


Figure 6. **Detection using a sketch.** *(a) A hand-sketched template. (b) The images against which it was compared with the corresponding detections.*

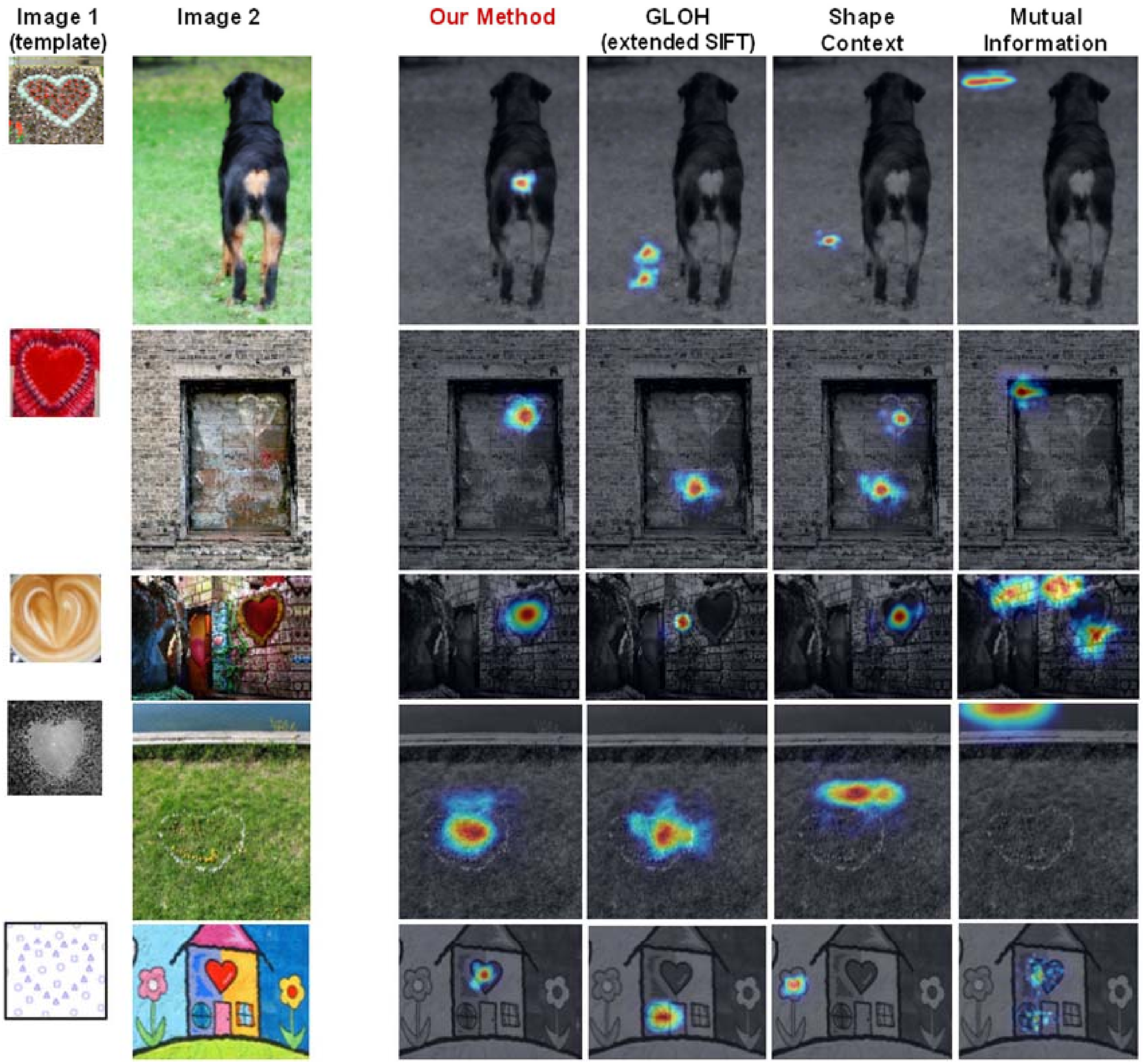


Image 1 (template)
Image 2
Our Method
GLOH (extended SIFT)
Shape Context
Mutual Information

International Journal of Computer Vision 64(2/3), 107–123, 2005

On Space-Time Interest Points

IVAN LAPTEV
IRISA/INRIA, Campus Beaulieu, 35042 Rennes Cedex, France
ilaptev@irisa.fr

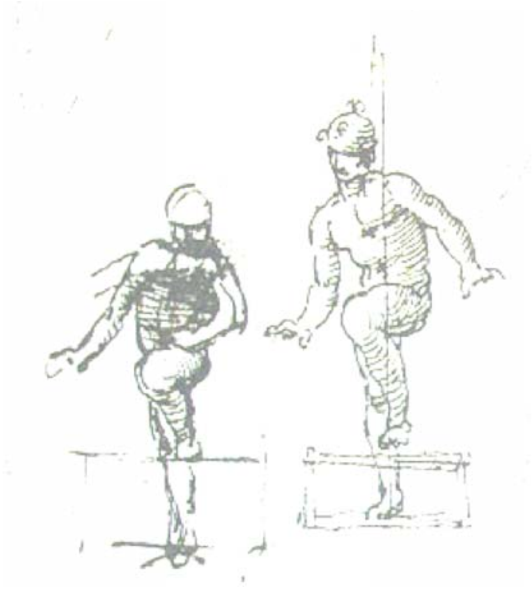
Received October 8, 2003; Revised October 8, 2003; Accepted June 23, 2004

First online version published in June, 2005

Abstract. Local image features or interest points provide compact and abstract representations of patterns in an image. In this paper, we extend the notion of spatial interest points into the spatio-temporal domain and show how the resulting features often reflect interesting events that can be used for a compact representation of video data as well as for interpretation of spatio-temporal events.

To detect spatio-temporal events, we build on the idea of the Harris and Förstner interest point operators and detect local structures in space-time where the image values have significant local variations in both space and time. We estimate the spatio-temporal extents of the detected events by maximizing a normalized spatio-temporal Laplacian operator over spatial and temporal scales. To represent the detected events, we then compute local, spatio-temporal, scale-invariant N-jets and classify each event with respect to its jet descriptor. For the problem of human motion analysis, we illustrate how a video representation in terms of local space-time features allows for detection of walking people in scenes with occlusions and dynamic cluttered backgrounds.

Keywords: interest points, scale-space, video interpretation, matching, scale selection



Human actions

in computer vision

Ivan Laptev

INRIA Rennes, France
ivan.laptev@inria.fr

Summer school, June 30 - July 11, 2008, Lotus Hill, China

Motivation

Goal: Interpretation of dynamic scenes

... non-rigid object motion ... camera motion ... complex background motion

Common methods:

- **Camera stabilization**
- **Segmentation** ?
- **Tracking** ?

Common problems:

- **Complex BG motion**
- **Changes in appearance**

⇒ ***No global assumptions about the scene***

Space-time

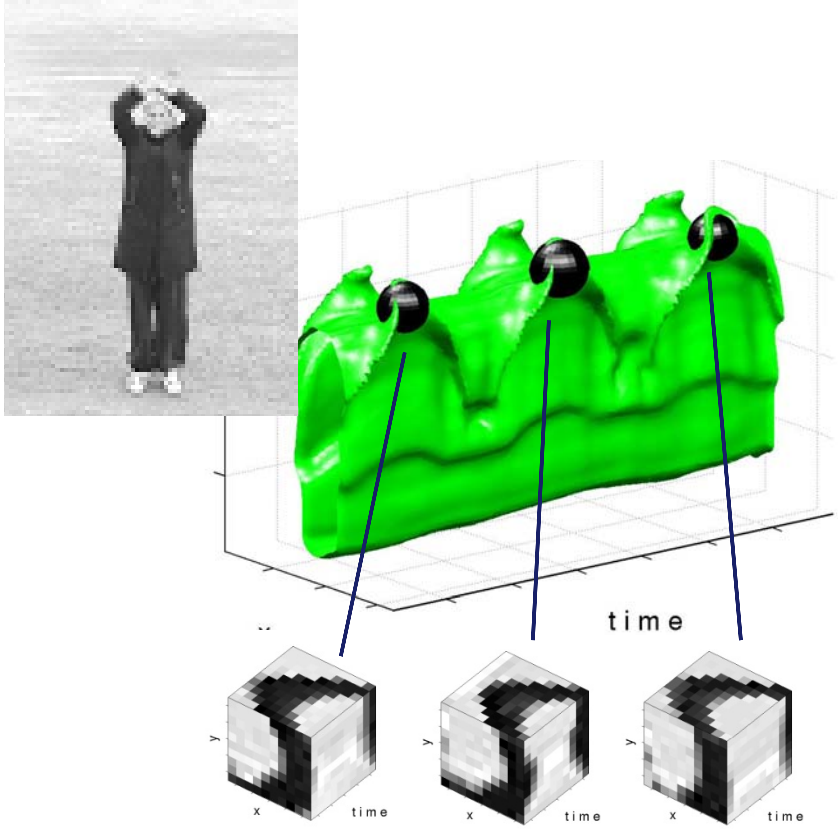
No **global** assumptions ⇒

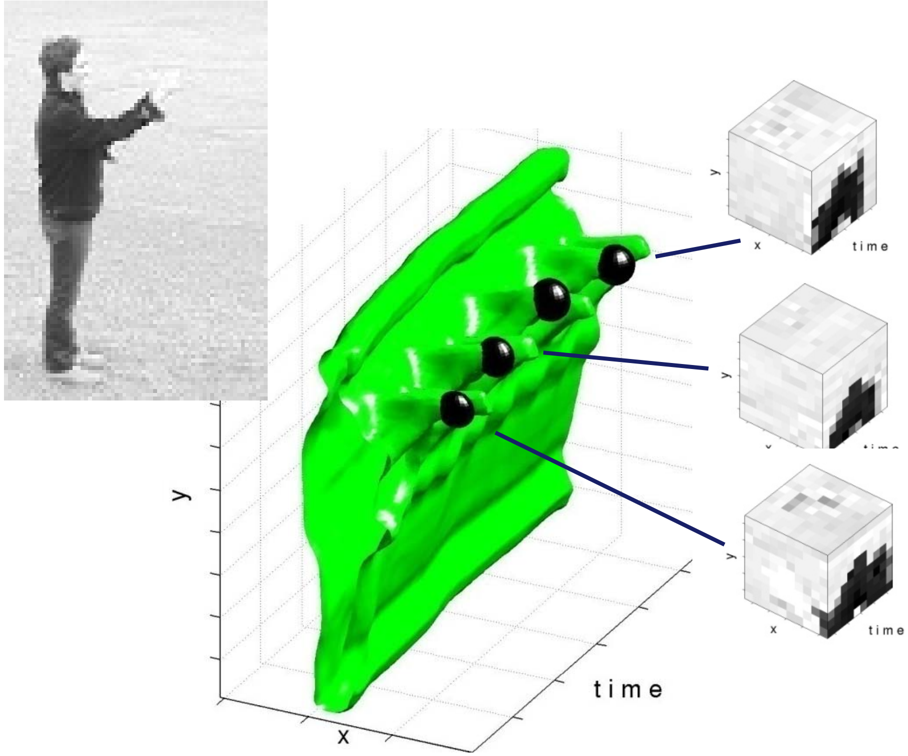
Consider **local** spatio-temporal neighborhoods

Space-time

No global assumptions ⇒

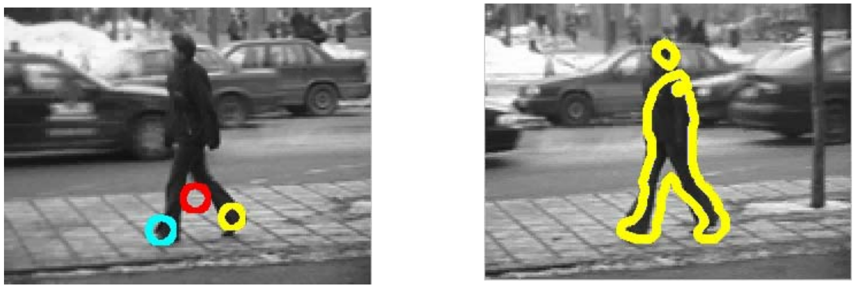
Consider local spatio-temporal neighborhoods



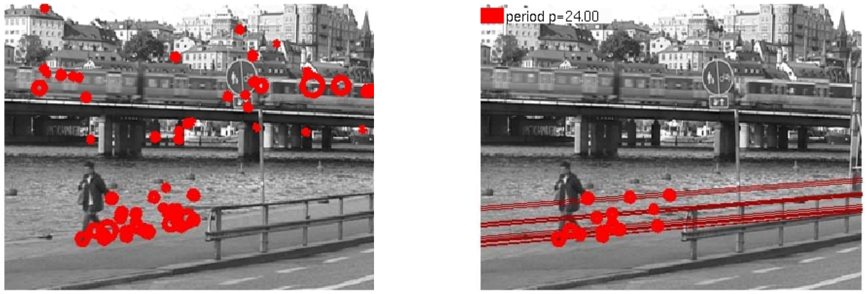


Applications: preview

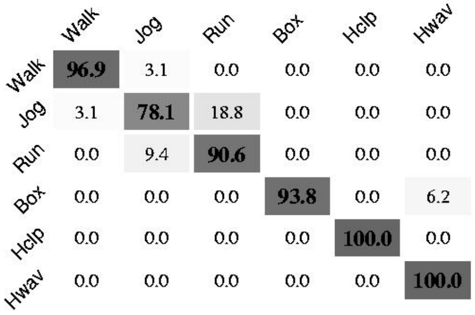
Sequence alignment



Periodic motion detection



Action recognition



	Walk	Jog	Run	Box	Hclp	Hwav
Walk	96.9	3.1	0.0	0.0	0.0	0.0
Jog	3.1	78.1	18.8	0.0	0.0	0.0
Run	0.0	9.4	90.6	0.0	0.0	0.0
Box	0.0	0.0	0.0	93.8	0.0	6.2
Hclp	0.0	0.0	0.0	0.0	100.0	0.0
Hwav	0.0	0.0	0.0	0.0	0.0	100.0

Questions

- How to find informative neighborhoods? (ICCV'03)
- How to deal with transformations in the data? (ICPR'04)
- How to describe the neighborhoods? (SCMVP'04)
- How to use obtained features in applications? (ICCV'03)
  (ICPR'04)
  (ICCV'05)

Questions

- **How to find informative neighborhoods?** —— (ICCV'03)
- **How to deal with transformations in the data?** (ICPR'04)
- **How to describe the neighborhoods?** —— (SCMVP'04)
- **How to use obtained features for applications?** (ICPR'04)
  (ICPR'04)
  (ICCV'05)

Space-Time interest points

What neighborhoods to consider?

Distinctive* neighborhoods** ⇒ **High image variation in *space* and *time ⇒ **Look at the distribution of the gradient**

Definitions:

$f: \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ Original image sequence

$g(x, y, t; \Sigma)$ Space-time Gaussian with covariance $\Sigma \in \mathrm{SPSD}(3)$

$L_\xi(\cdot;\Sigma) = f(\cdot) * g_\xi(\cdot;\Sigma)$ Gaussian derivative of f

$\nabla L = (L_x, L_y, L_t)^T$ Space-time gradient

$$\mu(\cdot;\Sigma) = \nabla L(\cdot;\Sigma)(\nabla L(\cdot;\Sigma))^T * g(\cdot; s\Sigma) = \begin{pmatrix} \mu_{xx} & \mu_{xy} & \mu_{xt} \\ \mu_{xy} & \mu_{yy} & \mu_{yt} \\ \mu_{xt} & \mu_{yt} & \mu_{tt} \end{pmatrix}$$

Second-moment matrix

Space-Time interest points

Properties of $\mu(\cdot;\Sigma)$:

$\mu(\cdot;\Sigma)$ defines second order approximation for the local distribution of ∇L within neighborhood Σ

$\mathrm{rank}(\mu) = 1 \quad \Rightarrow$ 1D space-time variation of f, e.g. *moving bar*

$\mathrm{rank}(\mu) = 2 \quad \Rightarrow$ 2D space-time variation of f, e.g. *moving ball*

$\mathrm{rank}(\mu) = 3 \quad \Rightarrow$ 3D space-time variation of f, e.g. *jumping ball*

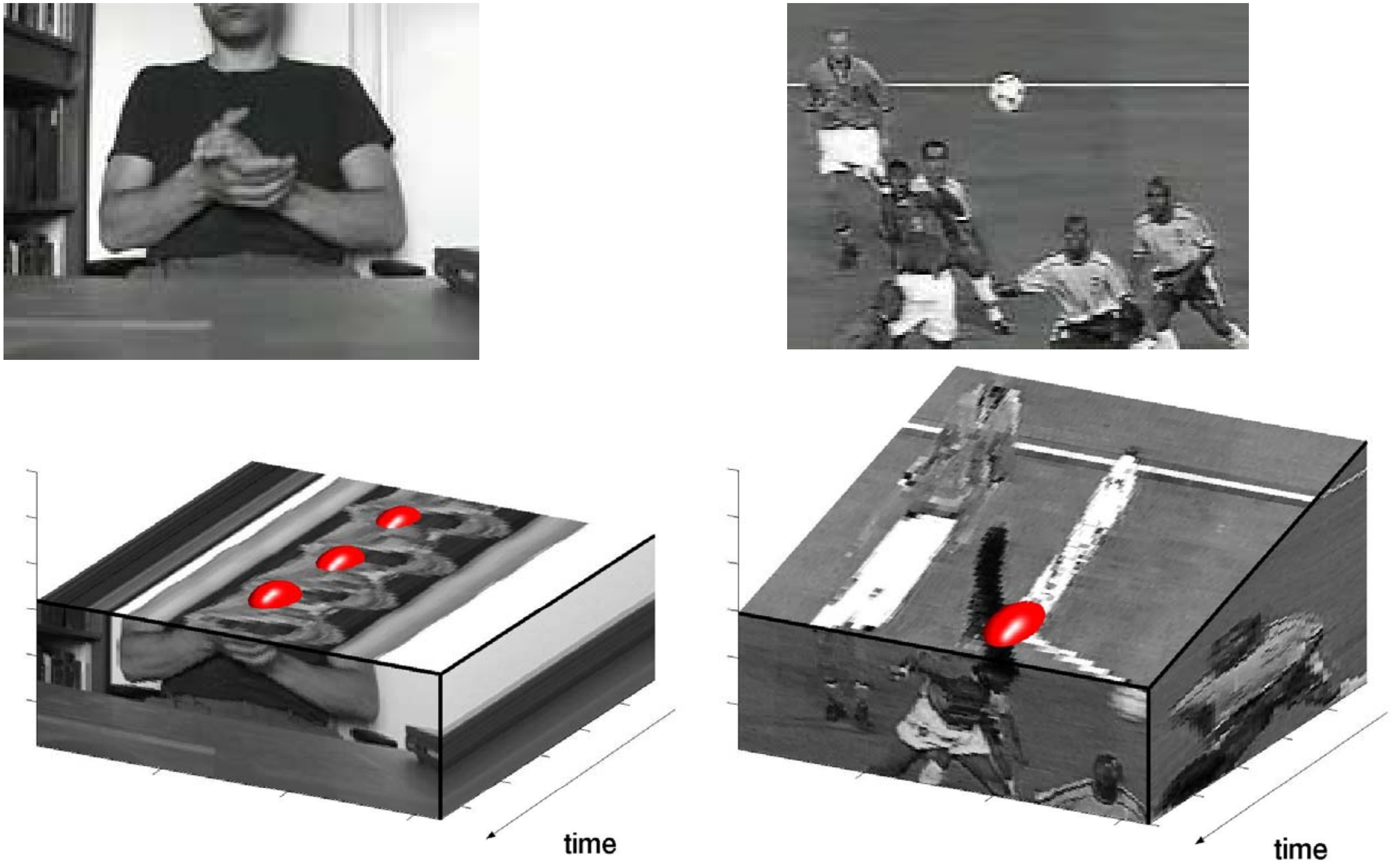
Large eigenvalues of μ can be detected by the local maxima of *H* over *(x,y,t)*:

$$\begin{aligned} H(p;\Sigma) &= \det(\mu(p;\Sigma)) + k\,\mathrm{trace}^3(\mu(p;\Sigma)) \\ &= \lambda_1\lambda_2\lambda_3 - k(\lambda_1+\lambda_2+\lambda_3)^3 \end{aligned}$$

(similar to Harris operator [Harris and Stephens, 1988])

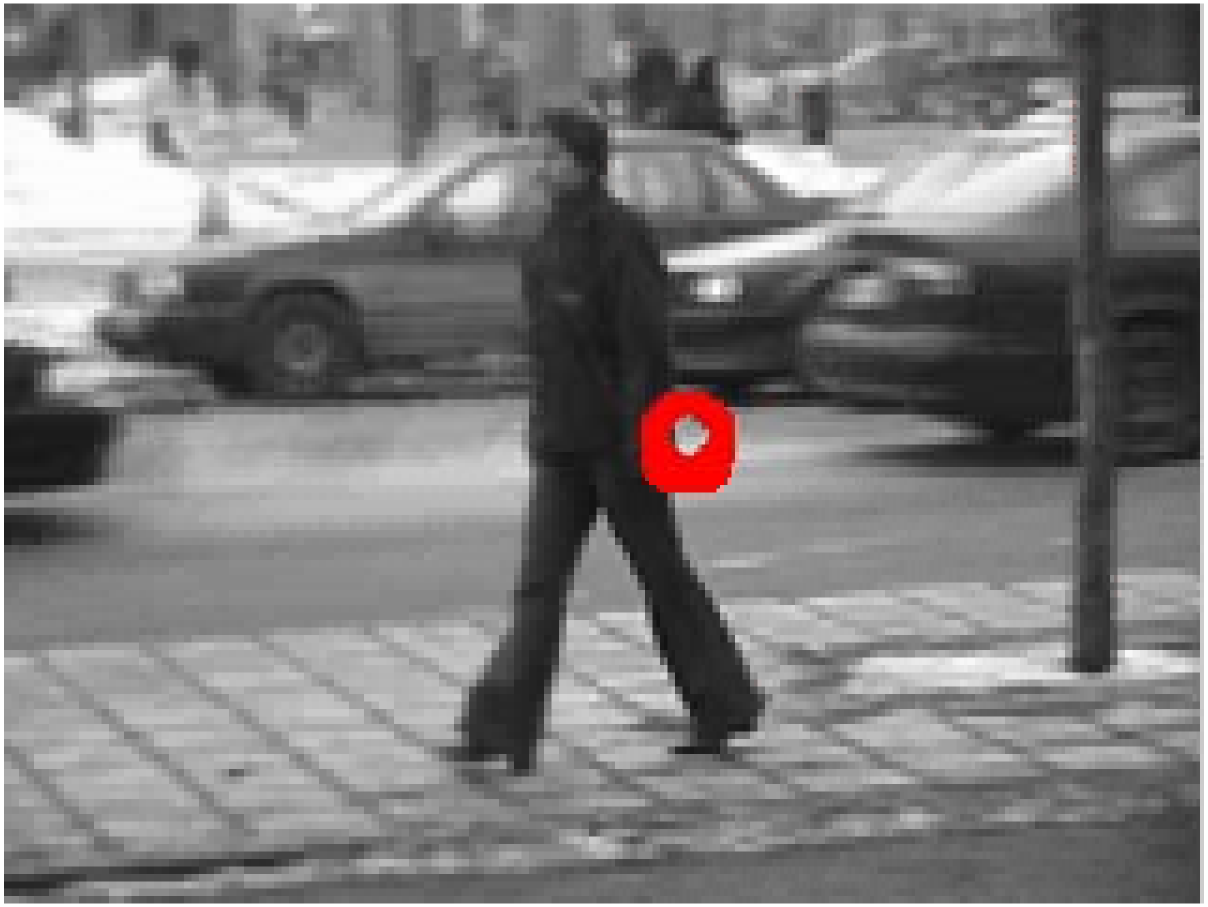
Space-Time interest points

Motion event detection



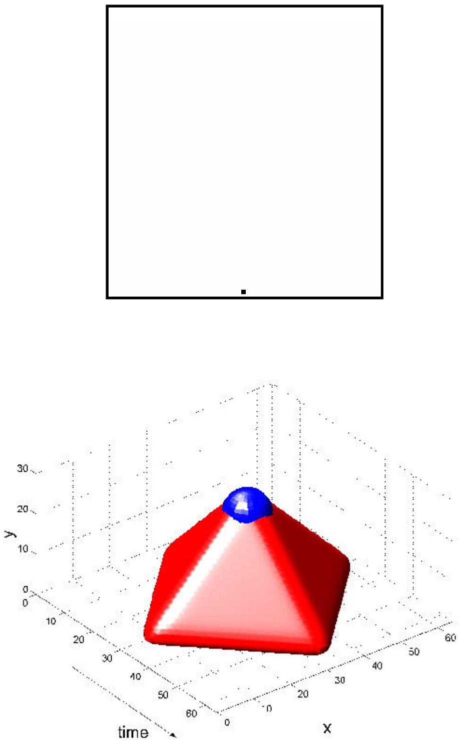
Space-Time interest points

Motion event detection: complex background

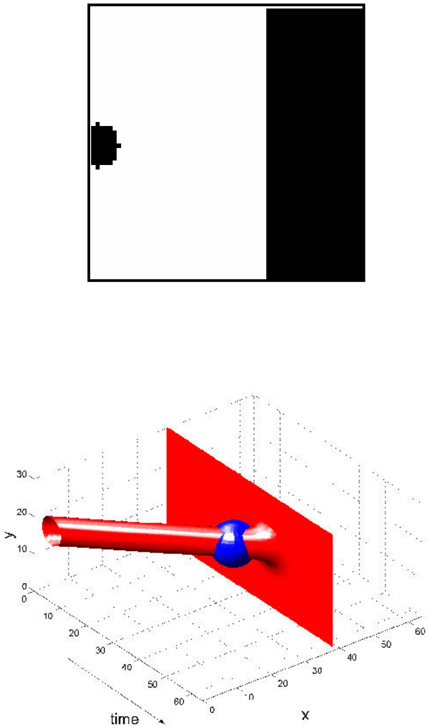


Space-Time interest points

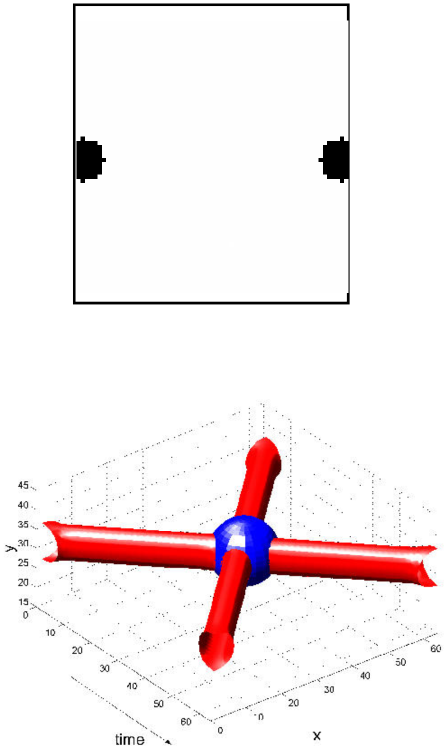
accelerations



appearance/ disappearance



split/merge



Relations to psychology

”… The world presents us with a continuous stream of activity which the mind parses into events. Like objects, they are bounded; they have beginnings, (middles,) and ends. Like objects, they are structured, composed of parts. However, in contrast to objects, events are structured in time…''

Tversky et.al.(2002), in *”The Imitative Mind”*

- **Events are well localized in time and are consistently identified by different people.**
- **The ability of memorizing activities has shown to be dependent on how fine we subdivide the motion into units.**

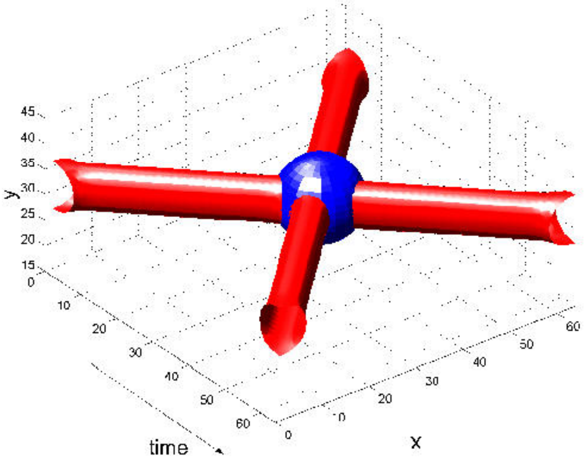
Questions

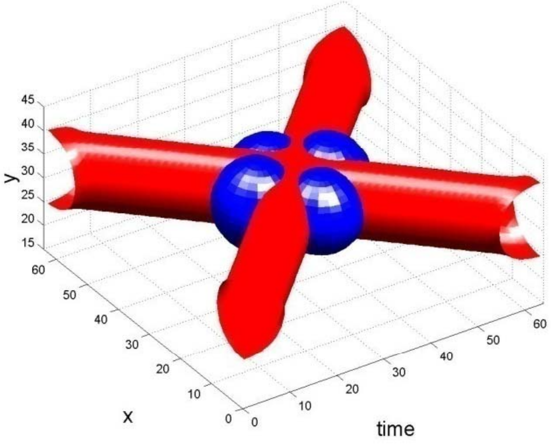
- How to **find** informative neighborhoods? —— (ICCV'03)
- How to deal with transformations in the data? (ICPR'04)
- How to describe the neighborhoods? —— (SCMVP'04)
- How to use obtained features for applications? (ICPR'04)

Questions

- How to find informative neighborhoods? (ICCV'03)
- How to deal with transformations in the data? (ICCV'03)
- How to describe the neighborhoods? (SCMVP'04)
- How to use obtained features for applications? (ICPR'04)

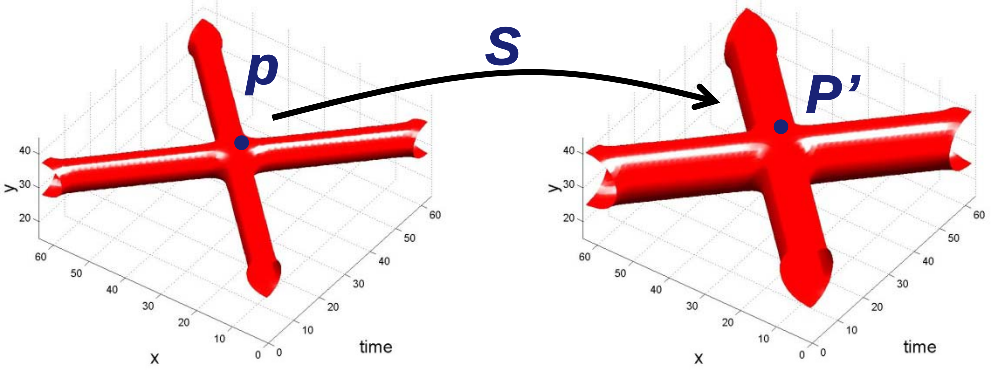
Scale and frequency transformations





Spatio-temporal scale selection

Image sequence *f* can be influenced by changes in *spatial and temporal resolution*



point transformation

$$p = S^{-1}p', \;\; S = \begin{pmatrix} s_\sigma & 0 & \\ 0 & s_\sigma & 0 \\ 0 & 0 & s_\tau \end{pmatrix}, \;\; p = \begin{pmatrix} x \\ y \\ t \end{pmatrix}$$

covariance transformation

$$\Sigma = pp^T = S^{-2}\Sigma' = \begin{pmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \tau^2 \end{pmatrix}$$

Spatio-temporal scale selection

Want to estimate *S* from the data

Estimate spatial and temporal extents of image structures ⇒ **Scale selection**

Scale-selection in *space* [Lindeberg IJCV'98]

$$\begin{cases} \nabla^2_{norm} L(p;\,\sigma) = \sigma^2 (L_{xx}(p;\,\sigma) + L_{yy}(p;\,\sigma)) \\ \partial_\sigma \left(\nabla^2_{norm} L(p;\,\sigma_0) \right) = 0 \end{cases}$$

Extension to *space-time:*

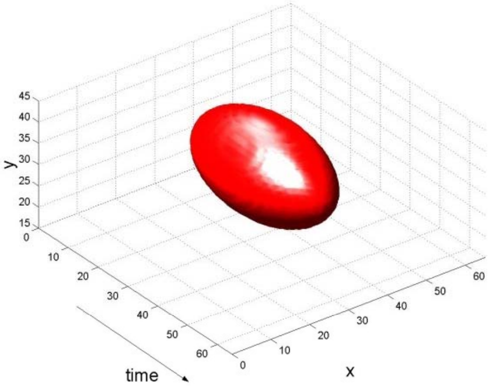
Find normalization parameters *a,b,c,d* for

$$\sigma^{2a} \tau^{2b} L_{xx}(p;\,\sigma_0, \tau_0)$$
$$\sigma^{2a} \tau^{2b} L_{yy}(p;\,\sigma_0, \tau_0)$$
$$\sigma^{2c} \tau^{2d} L_{tt}(p;\,\sigma_0, \tau_0)$$

Spatio-temporal scale selection

Analyze spatio-temporal blob

$$g(x, y, t;\, \sigma_l^2, \tau_l^2) =$$
$$\frac{1}{\sqrt{(2\pi)^3 \sigma_l^4 \tau_l^2}} \exp(-(x^2 + y^2)/2\sigma_l^2 - t^2/2\tau_l^2)$$



Extrema constraints

$$(\sigma^{2a}\tau^{2b}L_{xx})'_{\sigma^2} = 0 \qquad (\sigma^{2c}\tau^{2d}L_{tt})'_{\sigma^2} = 0$$
$$(\sigma^{2a}\tau^{2b}L_{xx})'_{\tau^2} = 0 \qquad (\sigma^{2c}\tau^{2d}L_{tt})'_{\tau^2} = 0$$

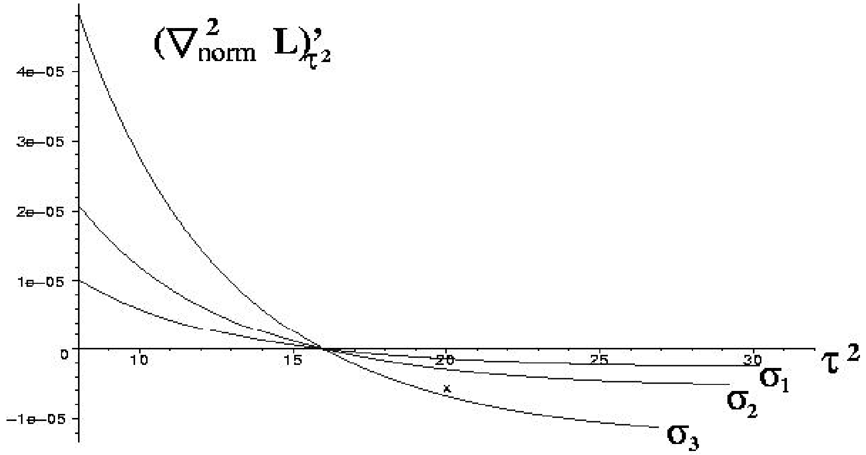
give parameter values
a=1, b=1/4, c=1/2, d=3/4

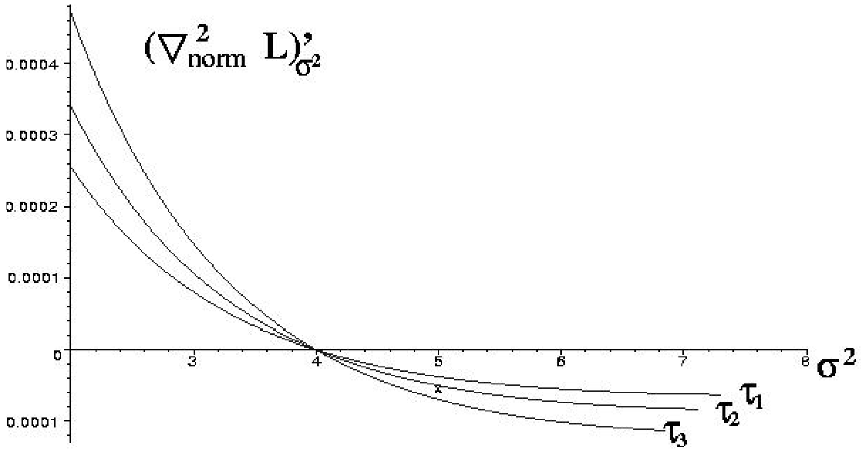
Spatio-temporal scale selection

⇒ **The normalized spatio-temporal Laplacian operator**

$$\nabla^2_{norm} L = \sigma^2 \tau^{1/2} (L_{xx} + L_{yy}) + \sigma \tau^{3/2} L_{tt}$$

Assumes extrema values at positions and scales corresponding to the centers and the spatio-temporal extent of a Gaussian blob





Space-Time interest points

H **depends on μ and, hence, on Σ and scale transformation *S***

⇒ **adapt interest points by iteratively computing:**

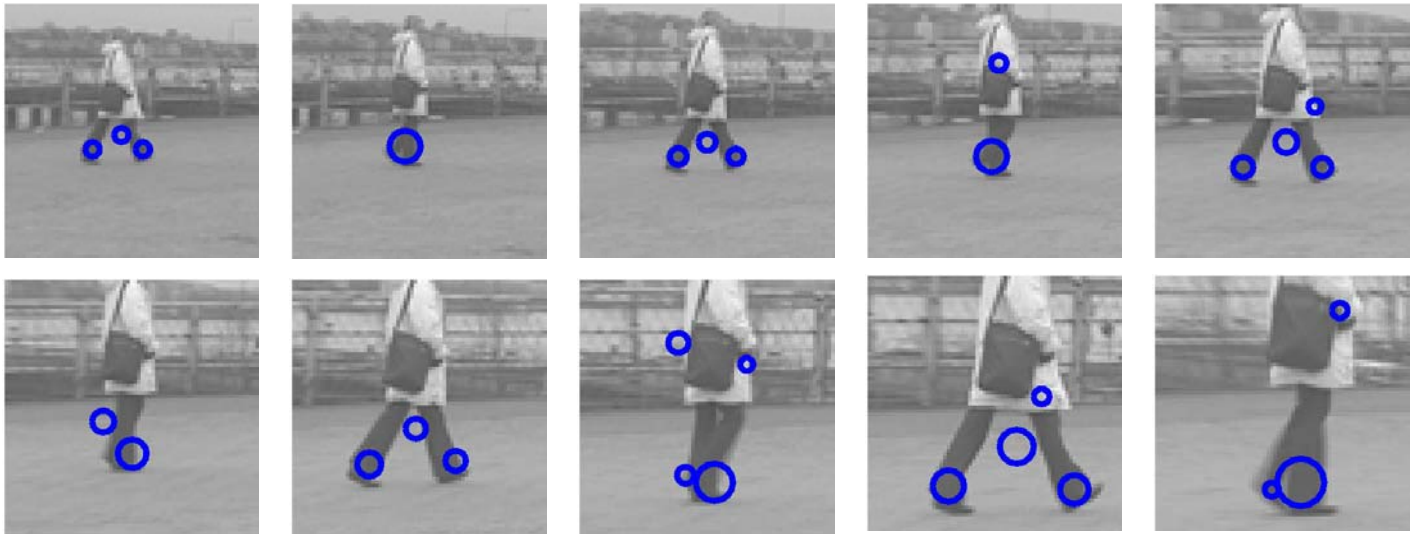
- **Scale estimation** $(\sigma_0, \tau_0) = \text{argmax}_{\sigma,\tau}(\nabla^2_{norm}L(p;\,\Sigma))^2$ **(∗)**

- **Interest point detection** $H(p;\,\Sigma) = \det(\mu(p;\,\Sigma)) + k\text{trace}^3(\mu(p;\,\Sigma))$ **(∗∗)**

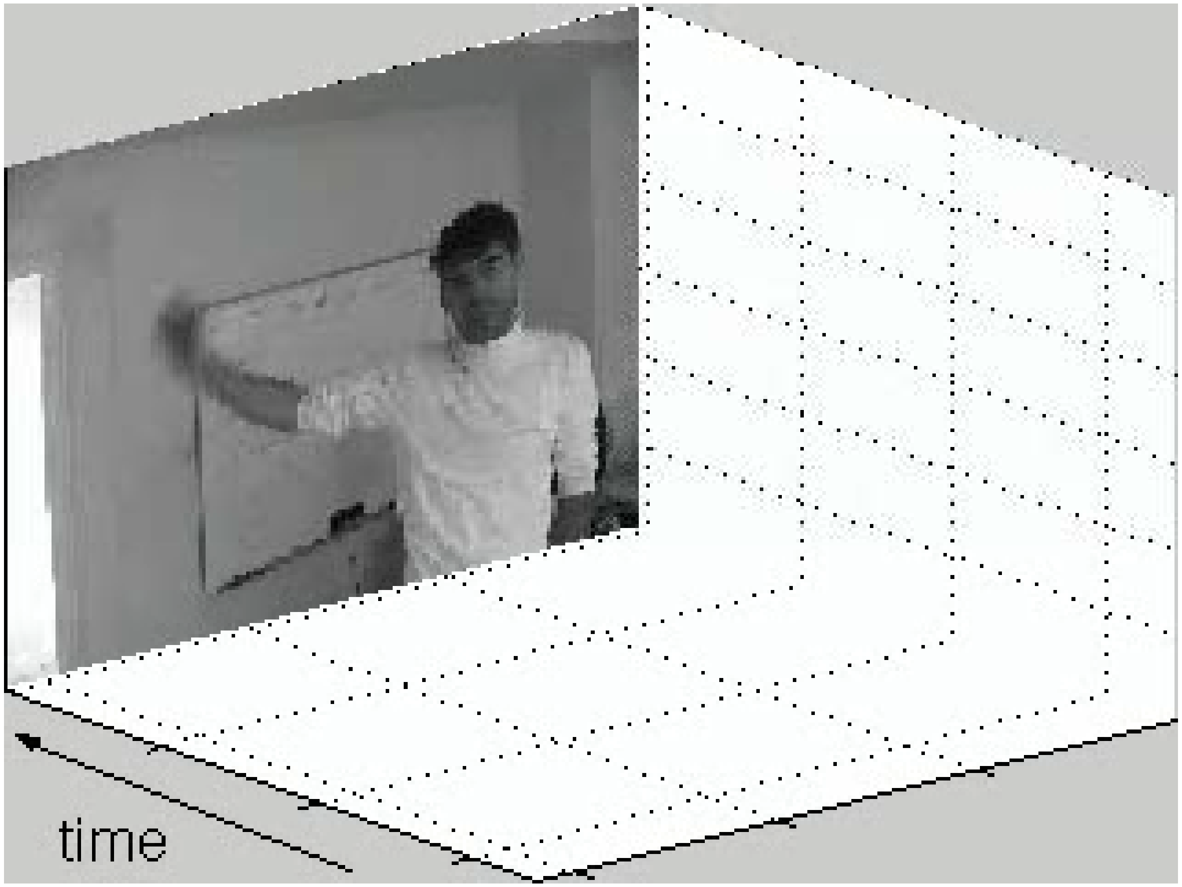
1. **Fix** Σ
2. **For each detected interest point** p_i **(∗∗)**
3. **Estimate** $S(\sigma,\tau)$ **(∗)**
4. **Update covariance** $\Sigma' = S^2$
5. **Re-detect** p_i **using** Σ'
6. **Iterate 3-6 until convergence of** σ,τ **and** p_i

Spatio-temporal scale selection

Stability to size changes, e.g. camera zoom



Spatio-temporal scale selection



Selection of temporal scales captures the *frequency* of events

Questions

- How to find informative neighborhoods? (ICCV'03)
- How to deal with transformations in the data? (ICCV'03)
- How to describe the neighborhoods? (SCMVP'04)
- How to use obtained features for applications? (ICPR'04)

Scale and frequency transformations

Questions

- How to find informative neighborhoods? (ICCV’03)
- How to deal with transformations in the data? (ICPR’04)
- How to describe the neighborhoods? (SCMVP’04)
- How to use obtained features for applications? (ICPR’04)

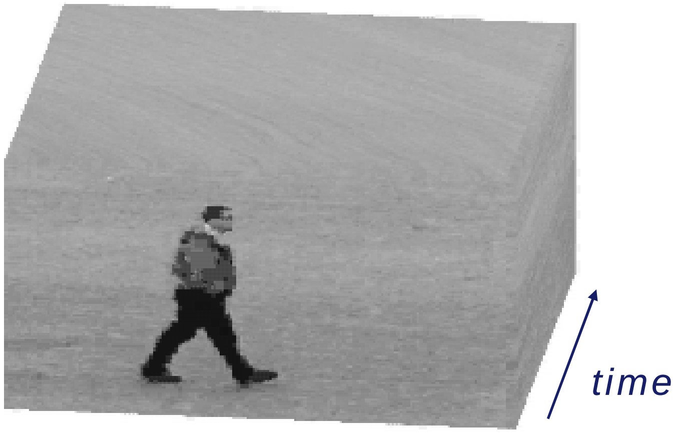
Transformations due to camera motion

Questions

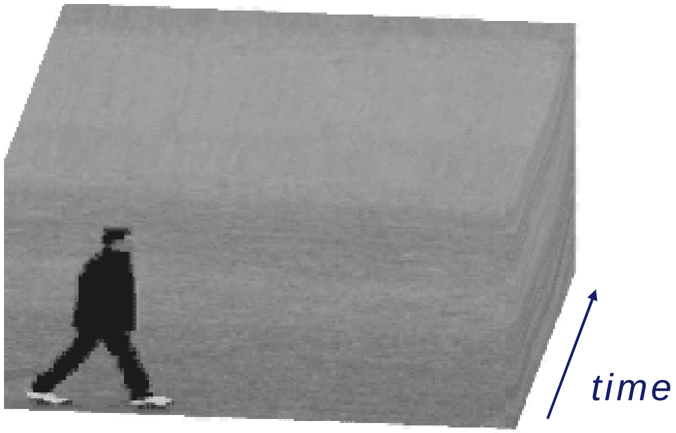
- How to find informative neighborhoods? (ICCV'03)
- **How to deal with transformations in the data? (ICPR'04)**
- How to describe the neighborhoods? (SCMVP'04)
- How to use obtained features for applications? (ICPR'04)

Transformations due to camera motion

Moving camera

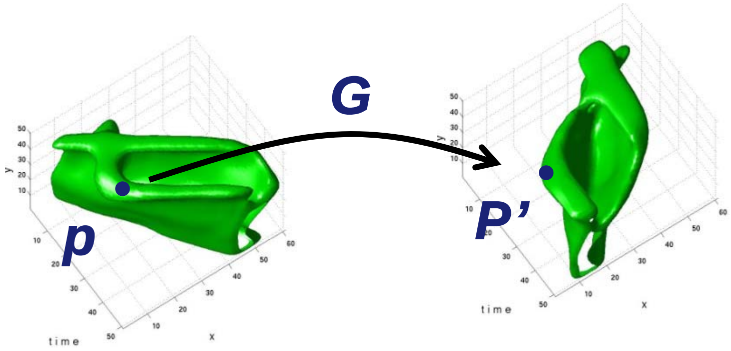


Stationary camera



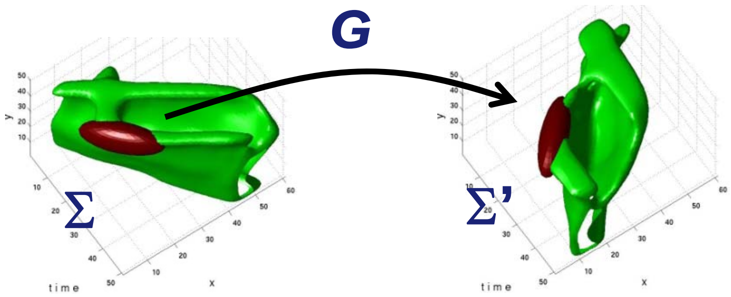
Galilean transformation

point transformation



$$p = G^{-1}p' \qquad G = \begin{pmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{pmatrix}, \qquad p = \begin{pmatrix} x \\ y \\ t \end{pmatrix}$$

covariance transformation



$$\Sigma = pp^T = G^{-1}\Sigma' G^{-T} \qquad \Sigma = \begin{pmatrix} c_{xx} & c_{xy} & c_{xt} \\ c_{xy} & c_{yy} & c_{yt} \\ c_{xt} & c_{yt} & c_{tt} \end{pmatrix}$$

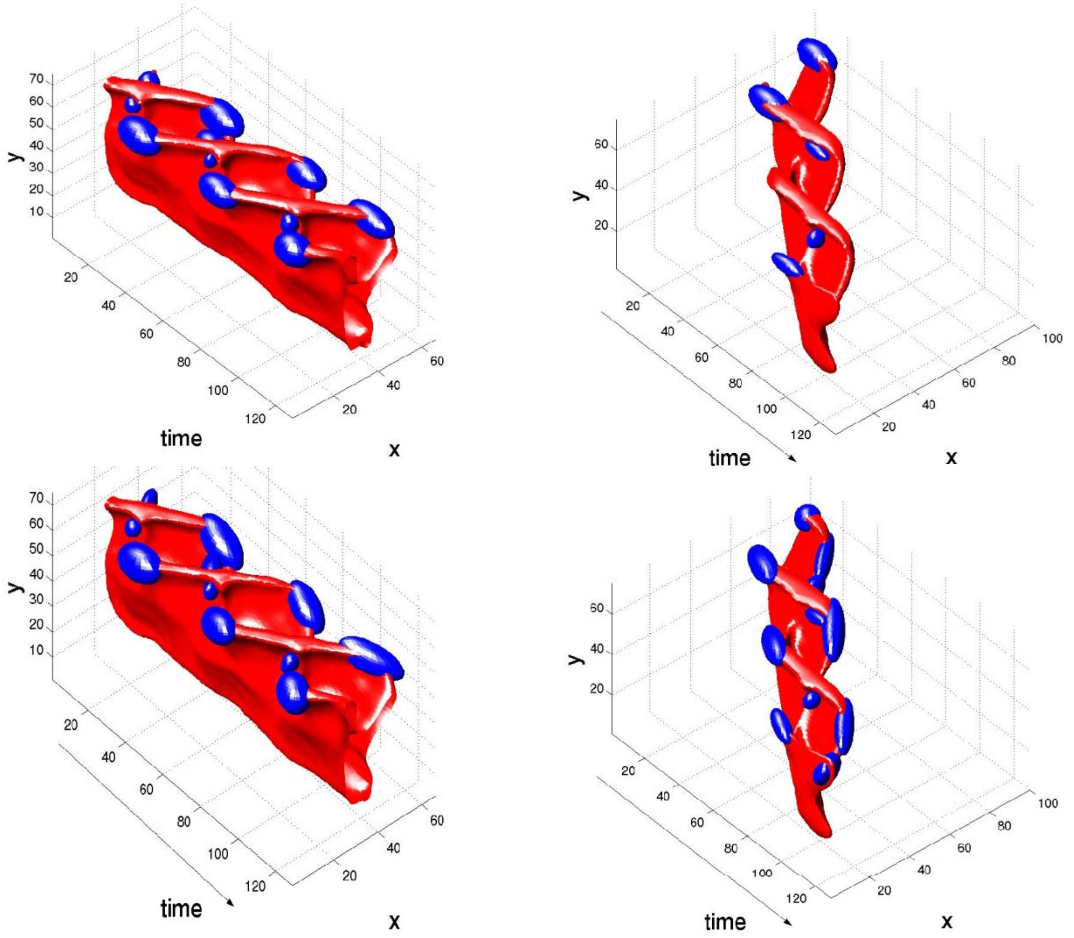
Adapted interest points

Stabilized camera

Stationary camera

Interest points

Velocity-adapted interest points



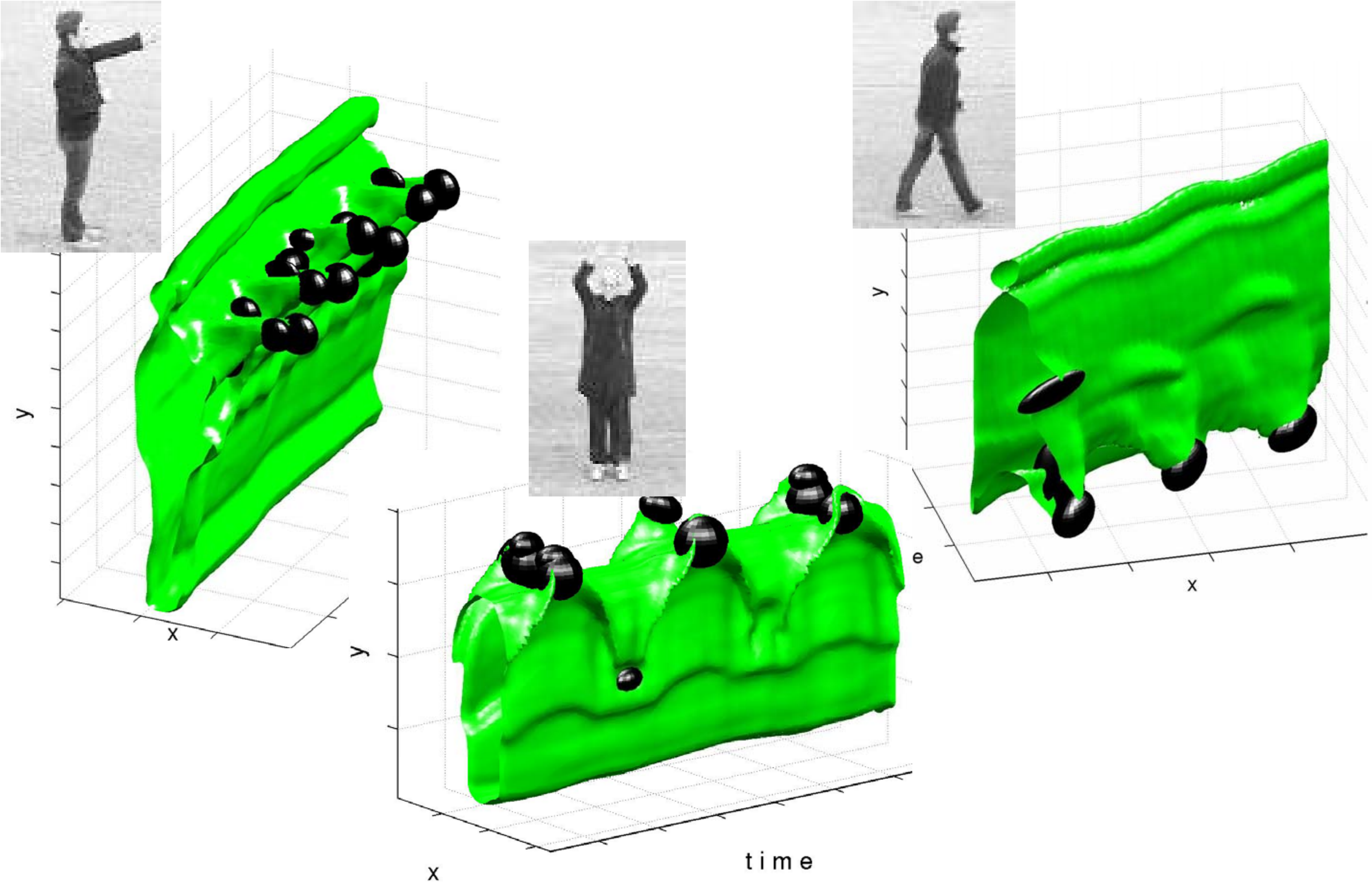
Questions

- How to find informative neighborhoods? (ICCV'03)
- **How to deal with transformations in the data?** (ICCV'03)
- How to describe the neighborhoods? (SCMVP'04)
- How to use obtained features for applications? (ICPR'04)

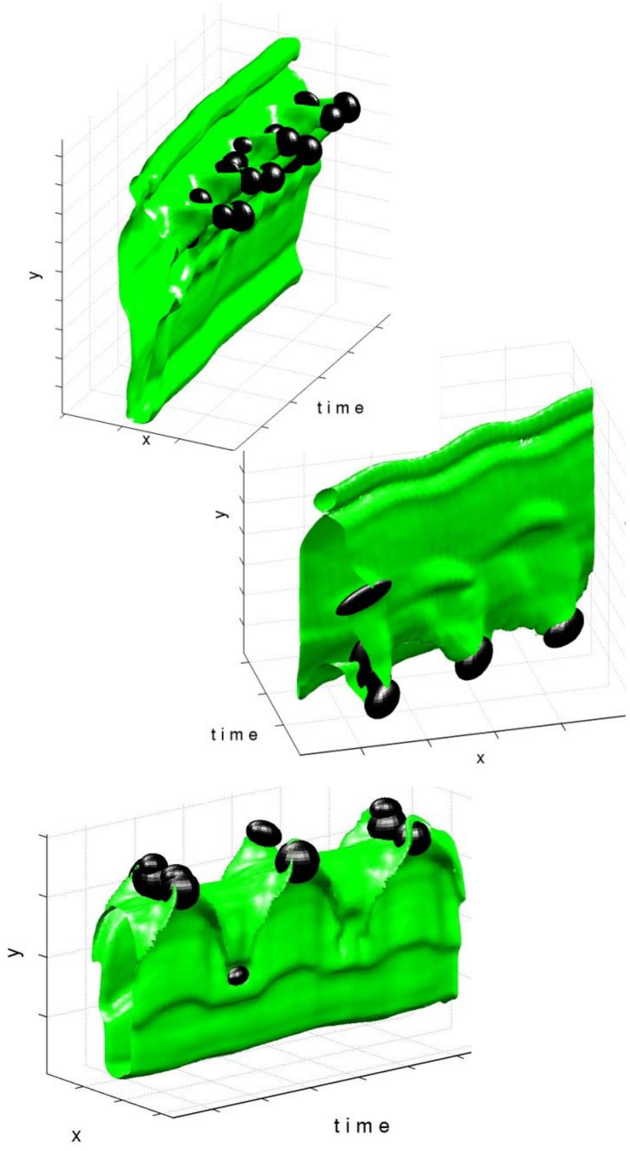
Questions

- How to find informative neighborhoods? (ICCV'03)
- How to deal with transformations in the data? (ICCV'03)
- How to **describe** the neighborhoods? (SCMVP'04)
- How to use obtained features for applications? (ICPR'04)

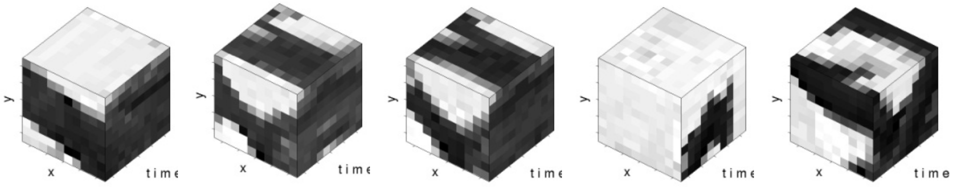
Features from human actions



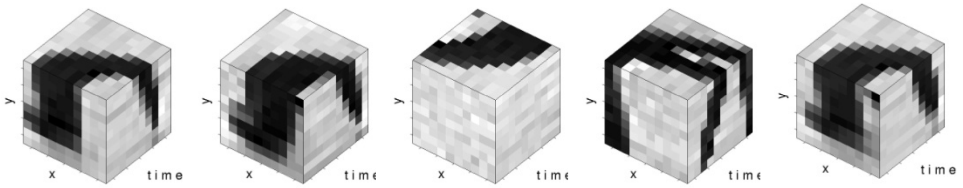
Space-time neighborhoods



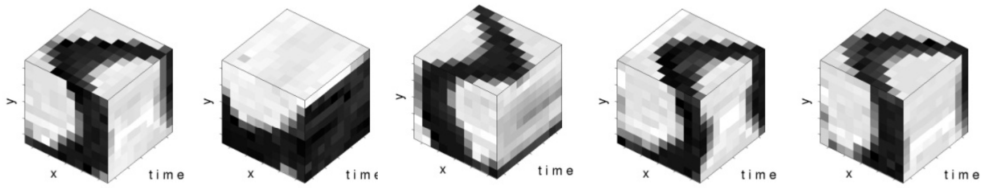
boxing



walking



hand waving



Local space-time descriptors

A common choice for local descriptors is a *local jet* (Koenderink and van Doorn, 1987) computed from spatio-temporal Gaussian derivatives (here at interest points *p*$_i$)

$$d_i = (L_{x'}, L_{y'}, L_{t'}, L_{x'x'}, L_{x'y'}, L_{x't'}, \ldots, L_{t't't't'})$$

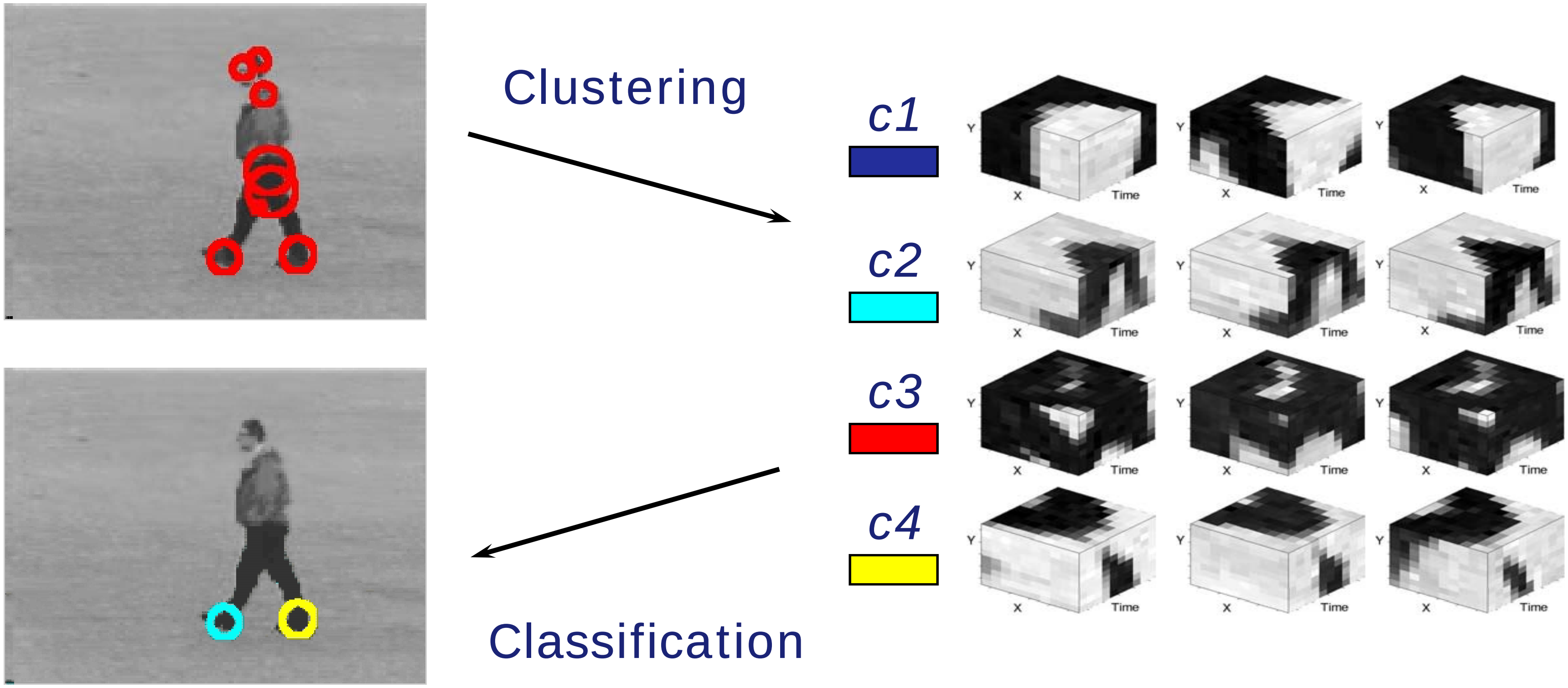
Covariance-normalization to obtain transformation-invriant Descriptors:

$$L_{x'^m y'^n t'^k}(\cdot;\, \Sigma) = \partial_{x'}^m(\partial_{y'}^n(\partial_{t'}^k(g(\cdot;\, \Sigma) * f)))$$

where $(\partial_{x'}, \partial_{y'}, \partial_{t'})^T = \Sigma^{-1/2}(\partial_x, \partial_y, \partial_t)^T$

Use of descriptors: Clustering

- Group similar points in the space of image descriptors using K-means clustering
- Select significant clusters



Clustering

Classification

c1 🟦

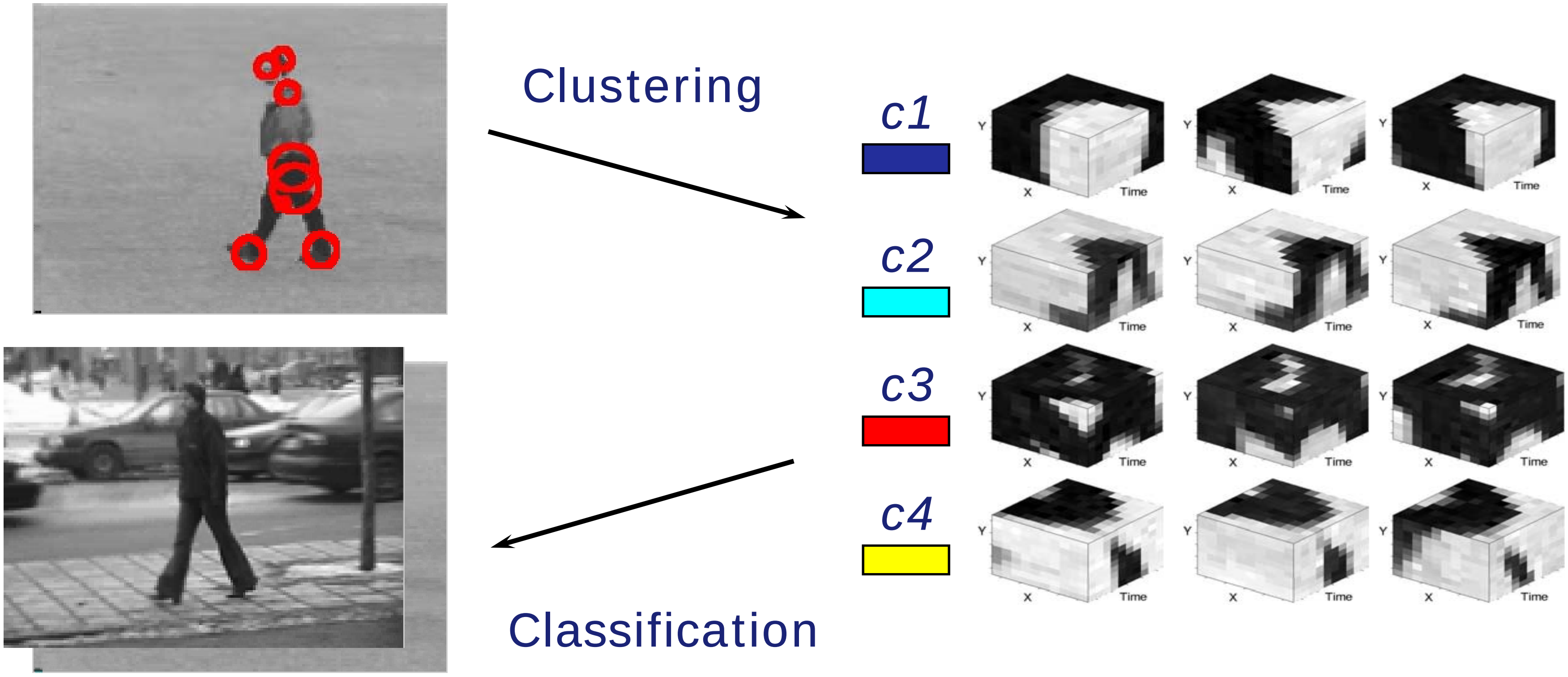
c2 🟦

c3 🟥

c4 🟨

Use of descriptors: Clustering

- Group similar points in the space of image descriptors using K-means clustering
- Select significant clusters



Clustering

c1 🟦

c2 🟦

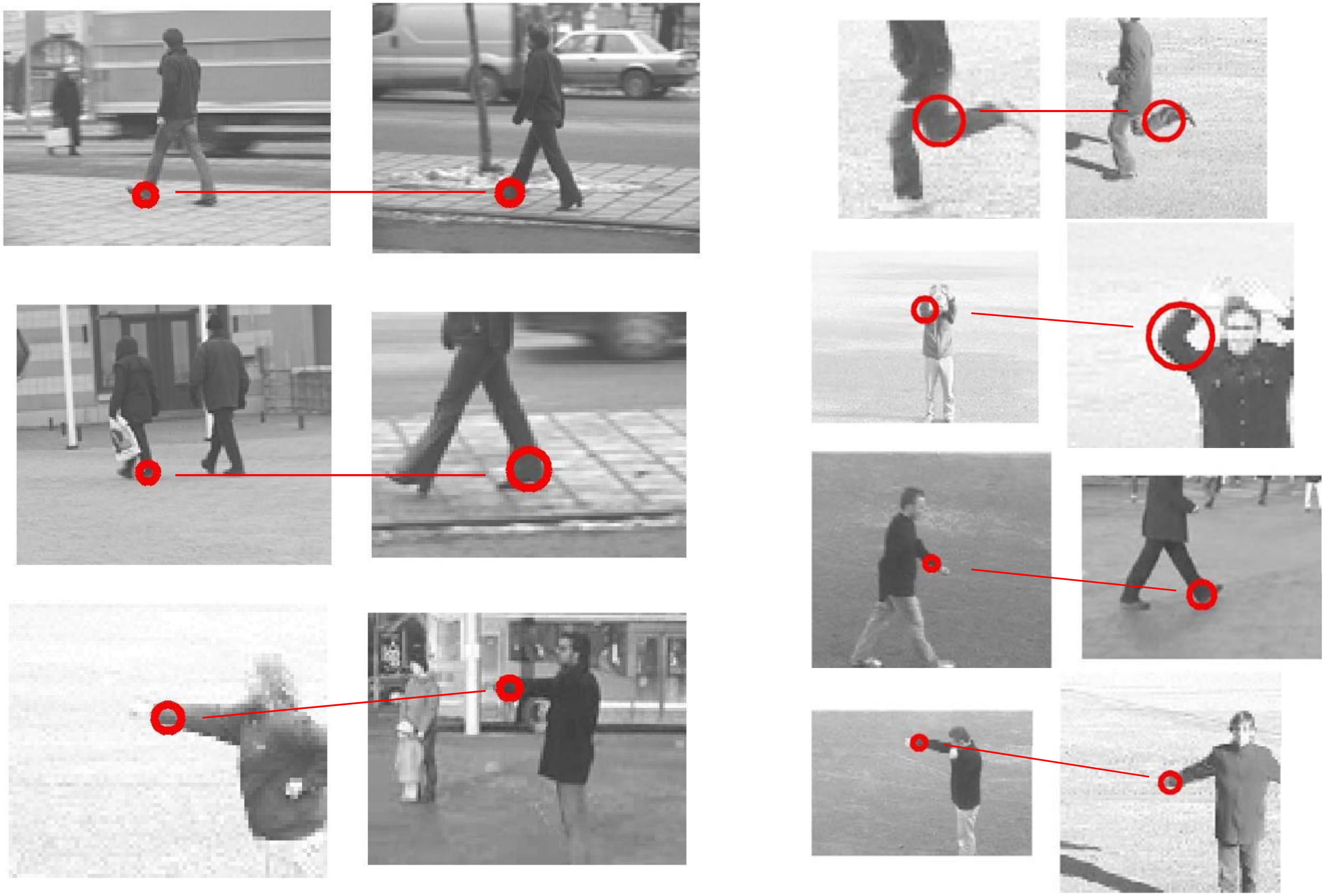
c3 🟥

c4 🟨

Classification

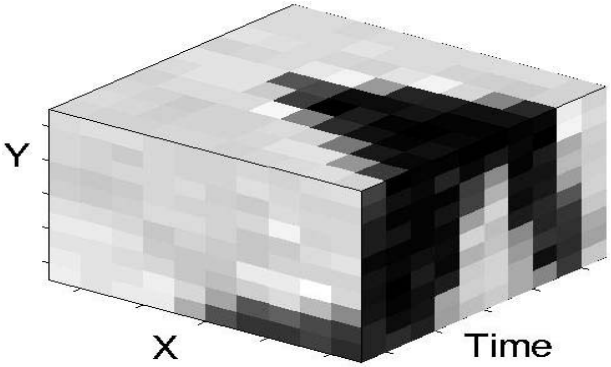
Use of descriptors: Matching

- Find similar events in pairs of video sequences



Other descriptors better?

Consider the following choices:



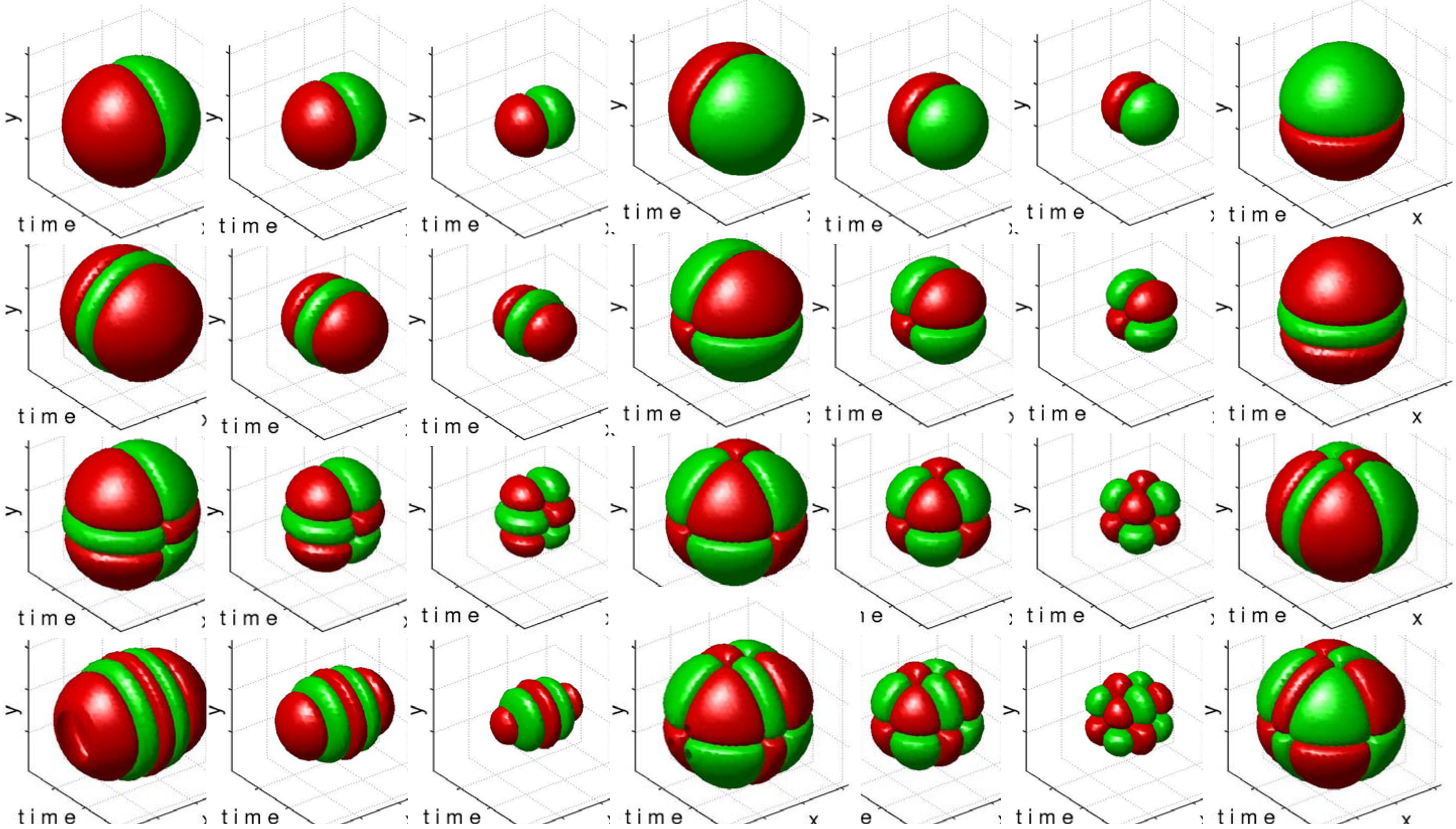
Spatio-temporal neighborhood

- Multi-scale spatio-temporal derivatives
- Projections to orthogonal bases obtained with PCA
- Histogram-based descriptors

Multi-scale derivative filters

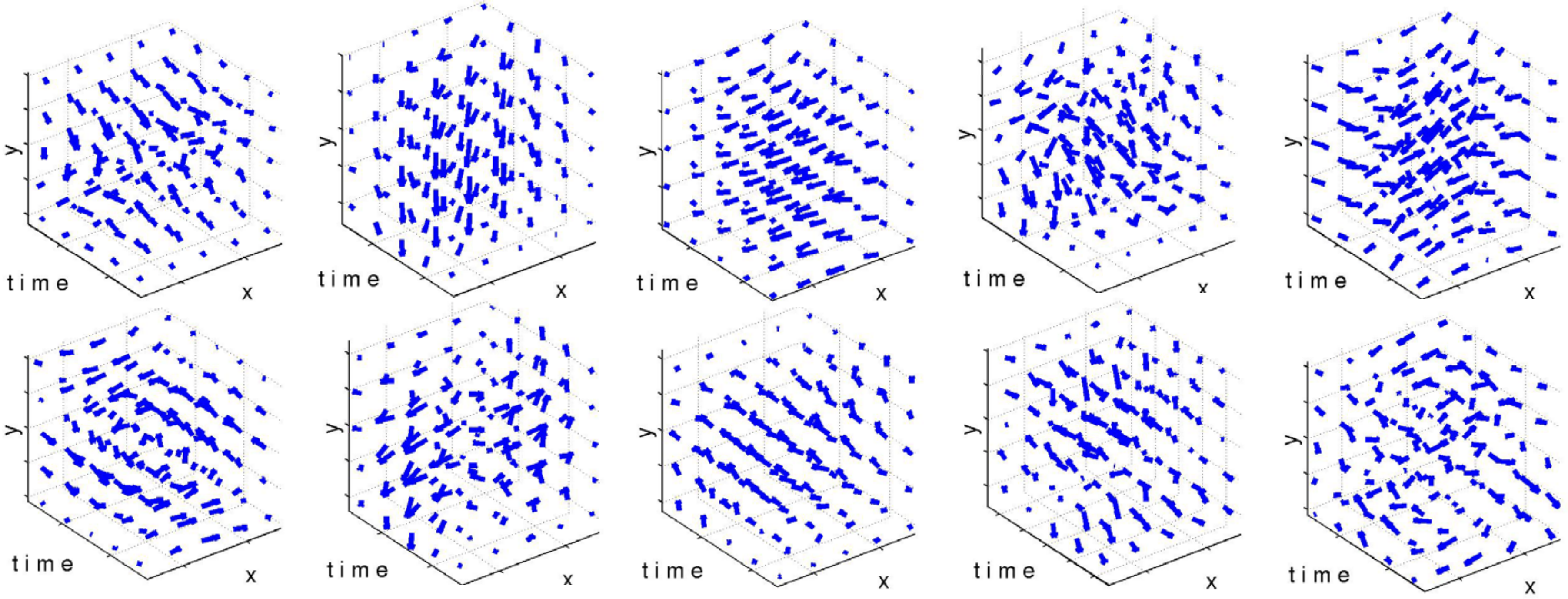
Derivatives up to order 2 or 4; 3 spatial scales; 3 temporal scales:

$\Rightarrow$ *9 x 3 x 3 = 81* or *34 x 3 x 3 = 306* dimensional descriptors



PCA descriptors

- Compute *normal flow* or *optic flow* in locally adapted spatio-temporal neighborhoods of features
- Subsample the flow fields to resolution 9x9x9 pixels
- Learn PCA basis vectors (separately for each flow) from features in training sequences
- Project flow fields of the new features onto the 100 most significant *eigen-flow-vectors*:

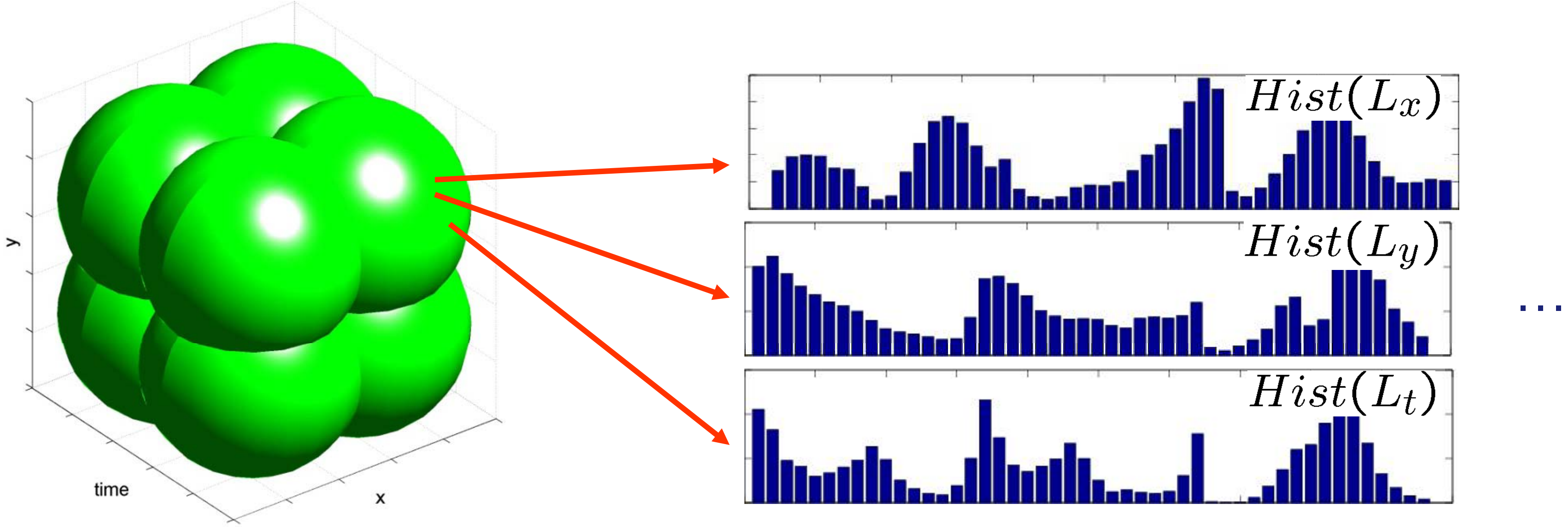


Position-dependent histograms

- Divide the neighborhood $\sum_i$ of each point p_i into M^3 *subneighborhoods*, here M=1,2,3
- Compute space-time gradients $(L_x, L_y, L_t)^T$ or optic flow $(v_x, v_y)^T$ at combinations of 3 temporal and 3 spatial scales

  $$\sigma \in \{0.5\sigma_0, \sigma_0, 2\sigma_0\}, \tau \in \{0.5\tau_0, \tau_0, 2\tau_0\}$$

  where σ_0, τ_0 are locally adapted detection scales
- Compute separable histograms over all subneighborhoods, derivatives/velocities and scales



Questions

- How to find informative neighborhoods? (ICCV'03)
- How to deal with transformations in the data? (ICCV'03)
- How to **describe** the neighborhoods? (SCMVP'04)
- How to use obtained features for applications? (ICPR'04)

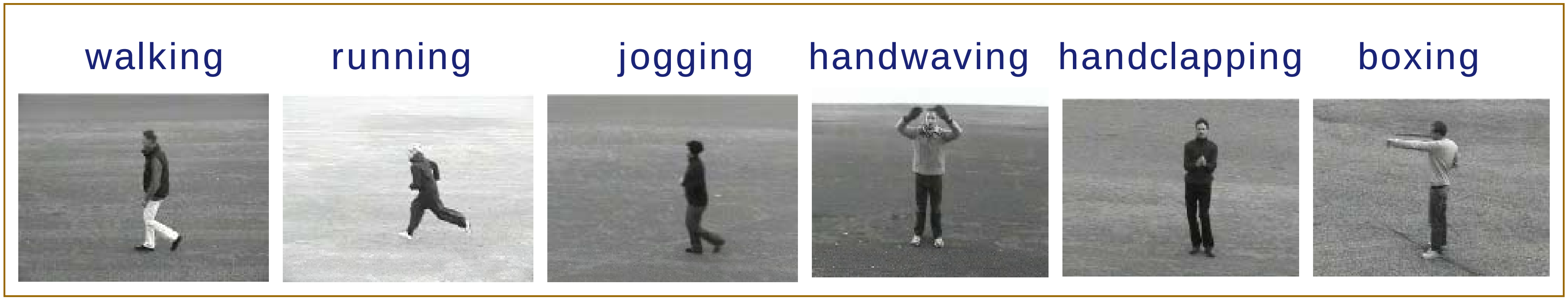
Questions

- How to find informative neighborhoods? (ICCV'03)
- How to deal with transformations in the data? (ICCV'03)
- How to describe the neighborhoods? (SCMVP'04)
- How to use obtained features for applications? (ICPR'04)

Action recognition

Evaluation: Action Recognition

Database:



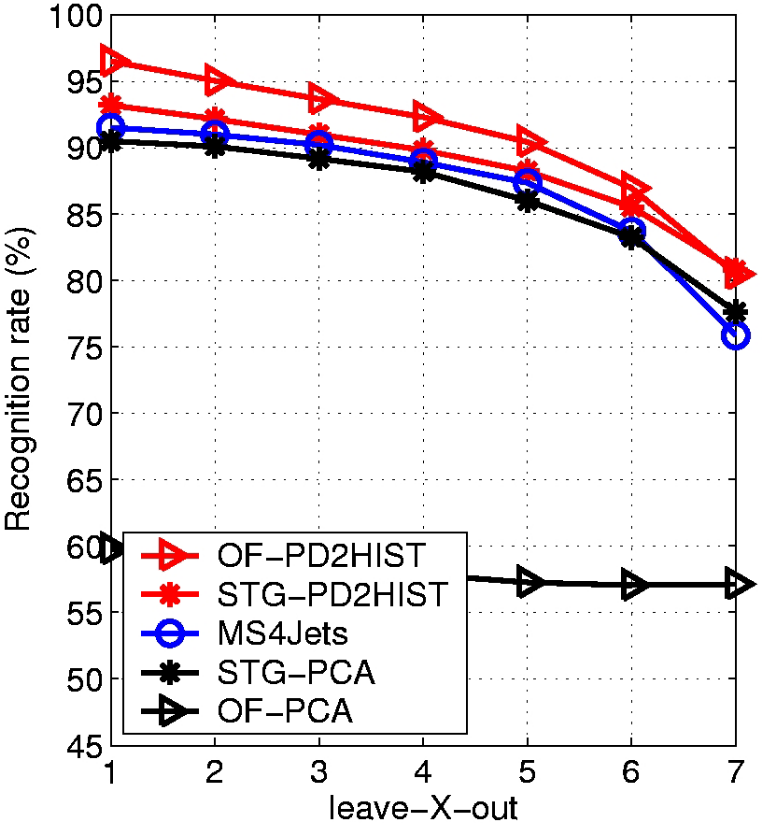
- Represent sequences as "*Bags of Local Features*"
- Compute similarity of two sequences as

$$D(s_1, s_2) = \text{GreedyMatch}(\{f_1^1, ..., f_1^n\}, \{f_2^1, ..., f_1^m\})$$

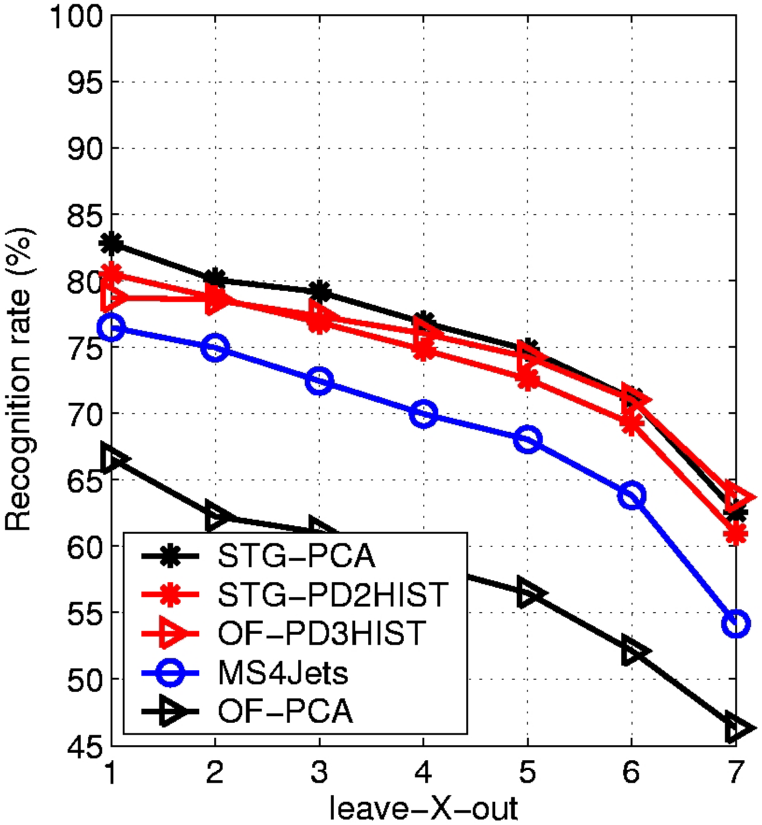
- Use e.g. Nearest Neighbor Classifier (NNC) to classify test actions given a set of training actions

Results: Recognition rates

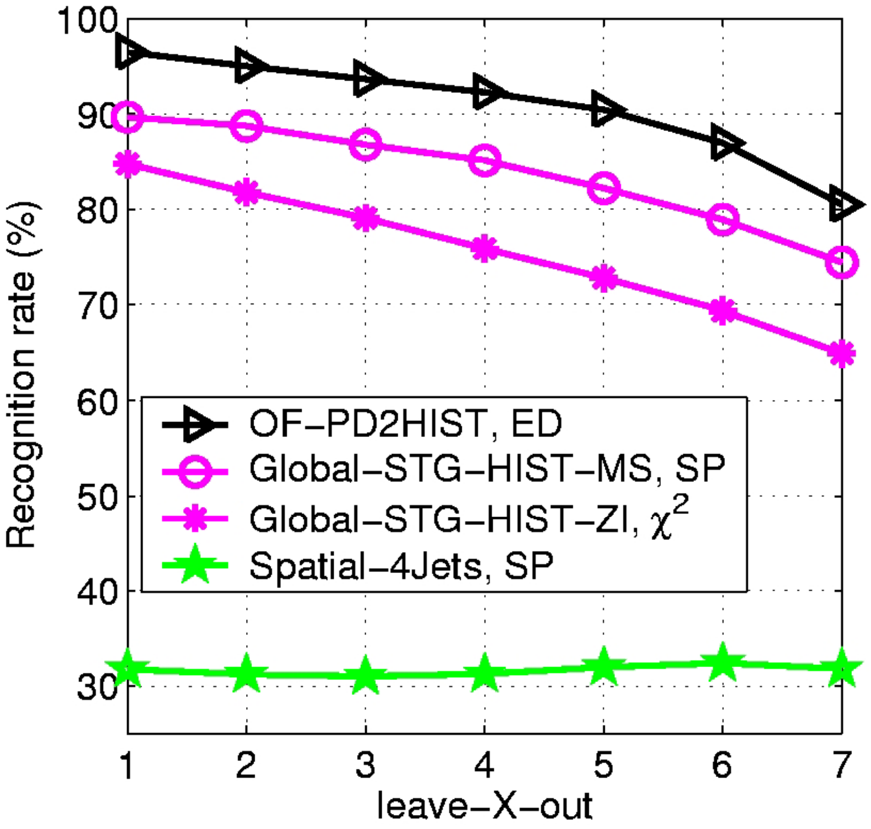
Scale-adapted features



Scale and *velocity* adapted features



Results: Comparison

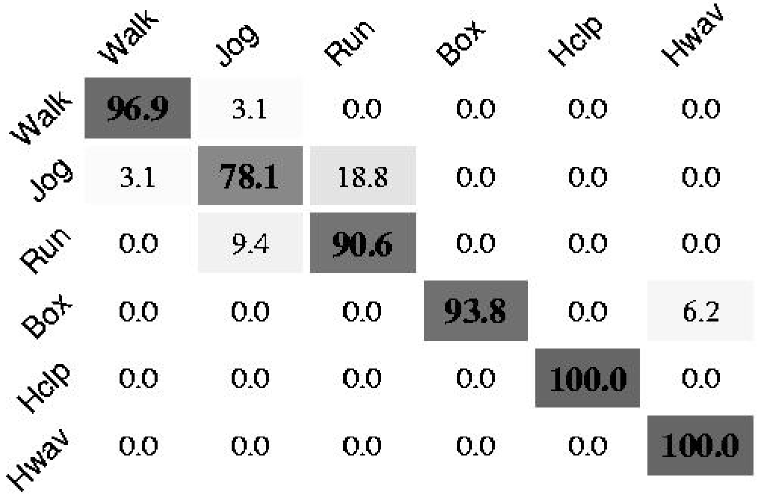


Global-STG-HIST: Zelnik-Manor and Irani CVPR'01

Spatial-4Jets: Spatial interest points (Harris and Stephens, 1988)

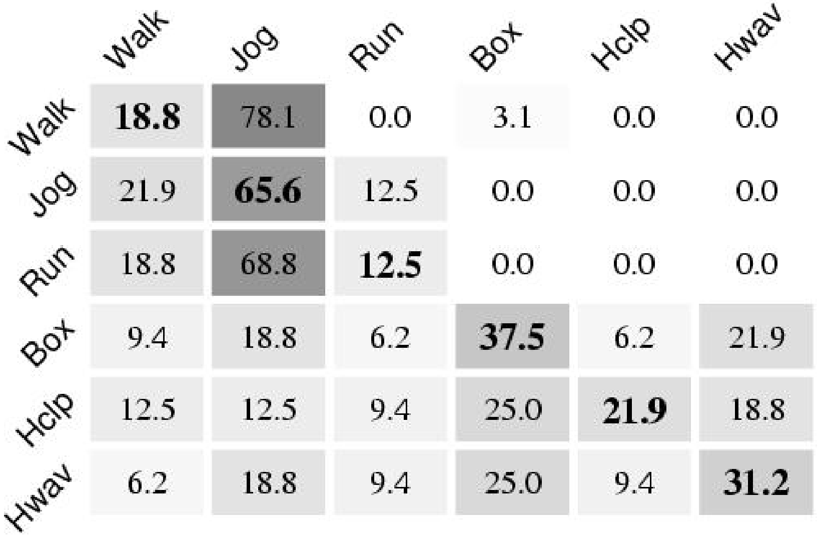
Confusion matrices

Position-dependent histograms for space-time interest points



	Walk	Jog	Run	Box	Hclp	Hwav
Walk	**96.9**	3.1	0.0	0.0	0.0	0.0
Jog	3.1	**78.1**	18.8	0.0	0.0	0.0
Run	0.0	9.4	**90.6**	0.0	0.0	0.0
Box	0.0	0.0	0.0	**93.8**	0.0	6.2
Hclp	0.0	0.0	0.0	0.0	**100.0**	0.0
Hwav	0.0	0.0	0.0	0.0	0.0	**100.0**

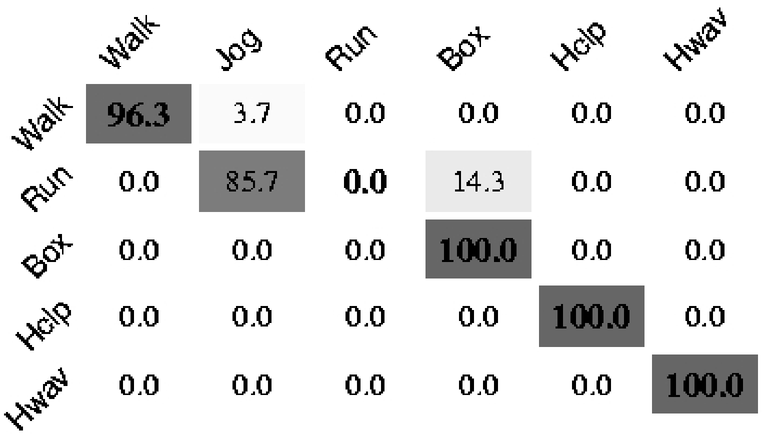
Local jets at *spatial interest points*



	Walk	Jog	Run	Box	Hclp	Hwav
Walk	**18.8**	78.1	0.0	3.1	0.0	0.0
Jog	21.9	**65.6**	12.5	0.0	0.0	0.0
Run	18.8	68.8	**12.5**	0.0	0.0	0.0
Box	9.4	18.8	6.2	**37.5**	6.2	21.9
Hclp	12.5	12.5	9.4	25.0	**21.9**	18.8
Hwav	6.2	18.8	9.4	25.0	9.4	**31.2**

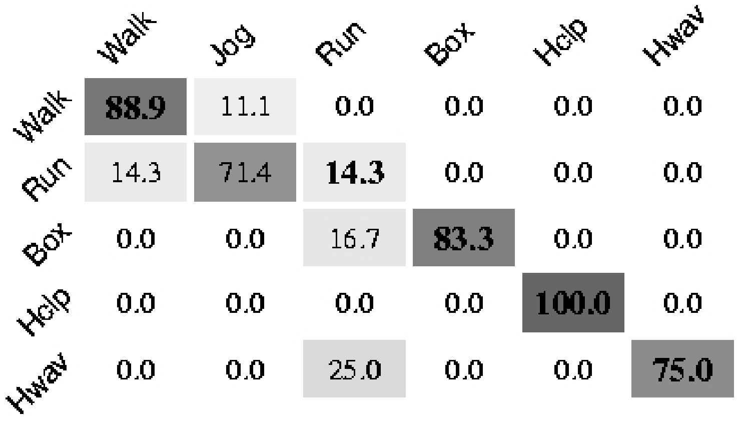
STG–PCA, ED	STG–PD2HIST, ED	4Jets, ED	2Jets, ED	STG–PD3HIST, ED	OF–PD2HIST, ED	OF–PD3HIST, ED	MS2Jets, ED	STG–HIST, SP	OF–PCA, SP	MS4Jets, ED	OF–HIST, ED	Global–STG–HIST–MS, SP
84.3	78.4	78.4	74.5	74.5	64.7	64.7	62.7	62.7	60.8	58.8	39.2	39.2

Confusion matrices



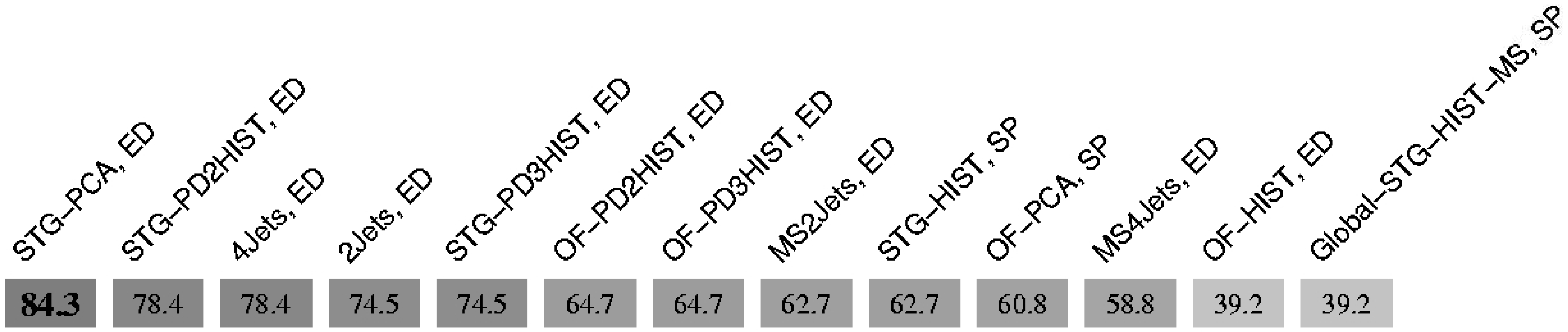
	Walk	Jog	Run	Box	Hclp	Hwav
Walk	**96.3**	3.7	0.0	0.0	0.0	0.0
Run	0.0	85.7	**0.0**	14.3	0.0	0.0
Box	0.0	0.0	0.0	**100.0**	0.0	0.0
Hclp	0.0	0.0	0.0	0.0	**100.0**	0.0
Hwav	0.0	0.0	0.0	0.0	0.0	**100.0**

STG-PCA, ED



	Walk	Jog	Run	Box	Hclp	Hwav
Walk	**88.9**	11.1	0.0	0.0	0.0	0.0
Run	14.3	71.4	**14.3**	0.0	0.0	0.0
Box	0.0	0.0	16.7	**83.3**	0.0	0.0
Hclp	0.0	0.0	0.0	0.0	**100.0**	0.0
Hwav	0.0	0.0	25.0	0.0	0.0	**75.0**

STG-PD2HIST, ED



STG–PCA, ED	STG–PD2HIST, ED	4Jets, ED	2Jets, ED	STG–PD3HIST, ED	OF–PD2HIST, ED	OF–PD3HIST, ED	MS2Jets, ED	STG–HIST, SP	OF–PCA, SP	MS4Jets, ED	OF–HIST, ED	Global–STG–HIST–MS, SP
84.3	78.4	78.4	74.5	74.5	64.7	64.7	62.7	62.7	60.8	58.8	39.2	39.2

Questions

- How to find informative neighborhoods? (ICCV'03)
- How to deal with transformations in the data? (ICCV'03)
- How to describe the neighborhoods? (SCMVP'04)
- How to use obtained features for applications? (ICPR'04)

Action recognition

Questions

- How to find informative neighborhoods? (ICCV'03)
- How to deal with transformations in the data? (ICCV'03)
- How to describe the neighborhoods? (SCMVP'04)
- How to use obtained features for applications? (ICCV'03)
  - Action recognition
  - Sequence alignment

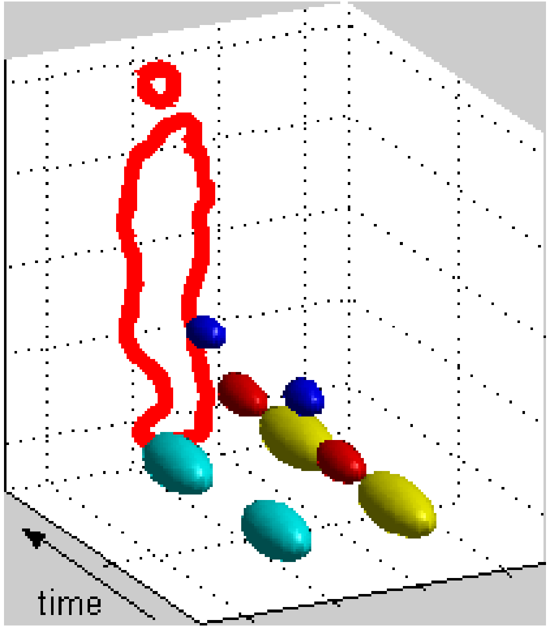
Sequence alignment

- Represent the gait pattern using classified spatio-temporal points corresponding the one gait cycle

- Define the state of the model *X* for the moment t_0 by the position, the size, the phase and the velocity of a person:

$$X_{t_0} = (x, y, s, \theta, v_x, v_y, v_s)$$

- Associate each phase θ with a silhouette of a person extracted from the original sequence



Sequence alignment

- Given a data sequence with the current moment t_0, detect and classify interest points in the time window of length t_w: (t_0, t_0-t_w)

- Transform model features according to X and for each model feature $f_{m,i}=(x_{m,i}, y_{m,i}, t_{m,i}, \sigma_{m,i}, \tau_{m,i}, c_{m,i})$ compute its distance d_i to the most close data feature $f_{d,j}$, $c_{d,j}=c_{m,i}$:

$$d_i = \sqrt{\frac{a}{\sigma^2_{m,i}}((x_{m,i} - x_{d,j})^2 + (y_{m,i} - y_{d,j})^2) + \frac{b}{\tau^2_{m,i}}(t_{m,i} - t_{d,j})^2}$$

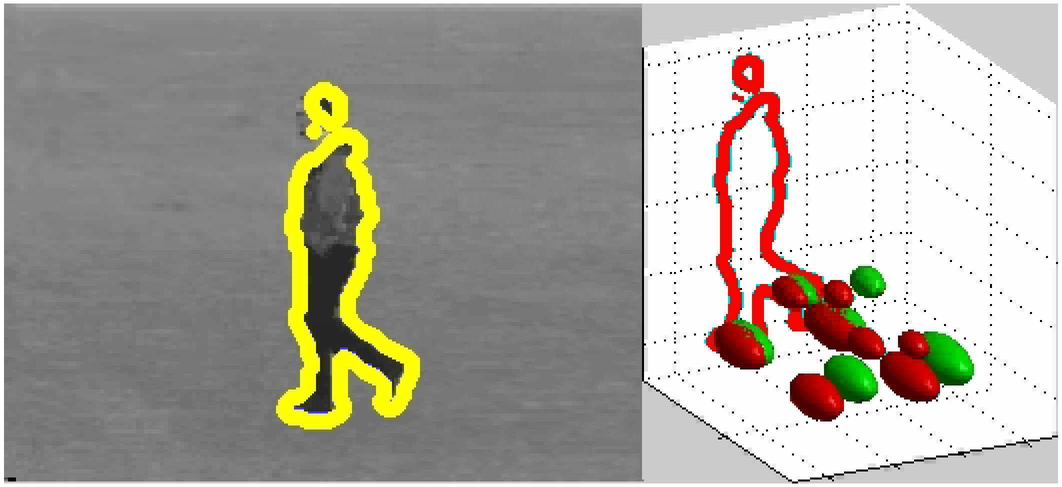
- Define the ”fit function” D of model configuration X *as a sum* of distances of all features weighted w.r.t. their ”age” (t_0-t_m) such that recent features get more influence on the matching

$$D(X) = \sum_i^N d_i \exp(-\frac{t_0 - t_{m,i}}{\rho^2})$$

Sequence alignment

At each moment *t₀* minimize *D* with respect to *X* using standard Gauss-Newton minimization method

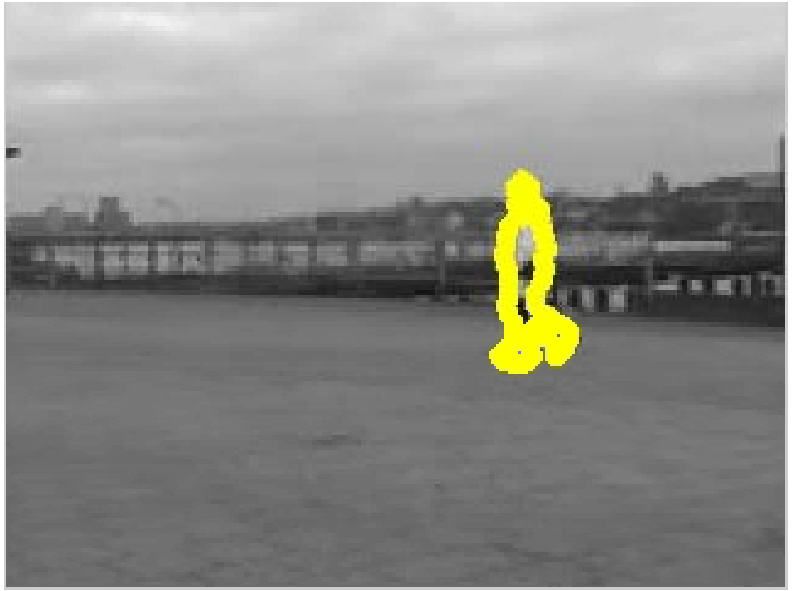
$$\tilde{X} = \text{argmin}_X \, D(X, t_0)$$



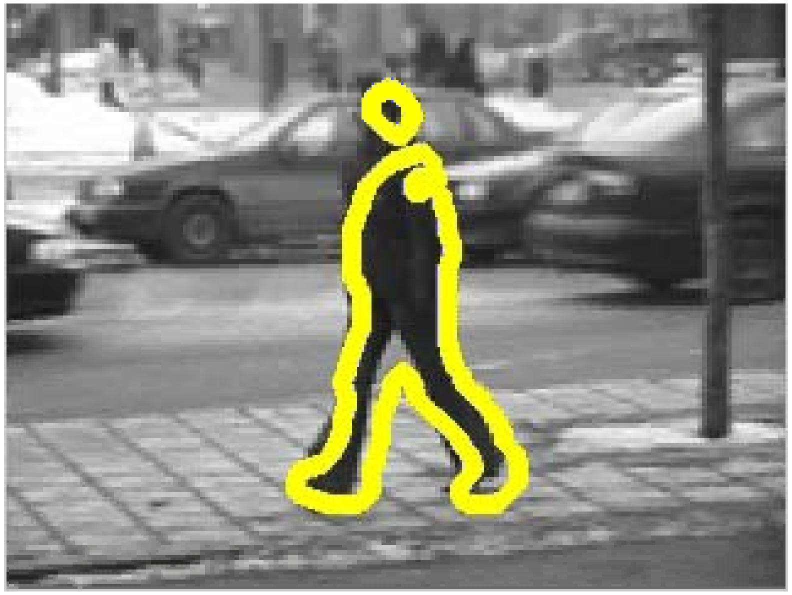
🟩 data features

🟥 model features

Experiments



Experiments

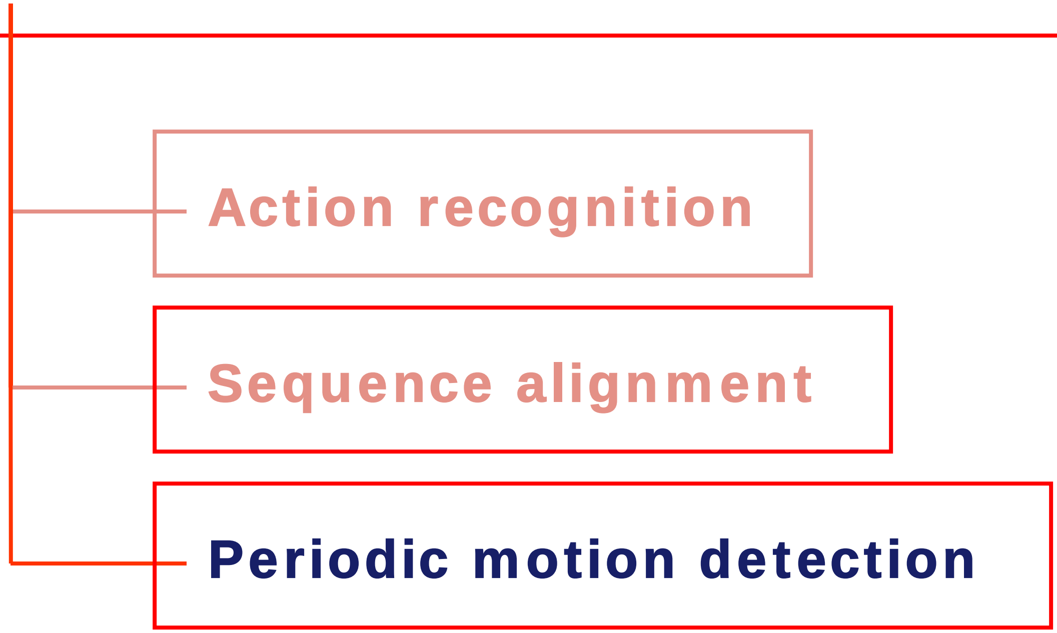


Questions

- How to find informative neighborhoods? (ICCV'03)
- How to deal with transformations in the data? (ICCV'03)
- How to describe the neighborhoods? (SCMVP'04)
- How to use obtained features for applications? (ICPR'04)
    - Action recognition
    - Sequence alignment

Questions

- How to find informative neighborhoods? (ICCV'03)
- How to deal with transformations in the data? (ICCV'03)
- How to describe the neighborhoods? (SCMVP'04)
- How to **use** obtained features for **applications**? (ICCV'05)



Action recognition

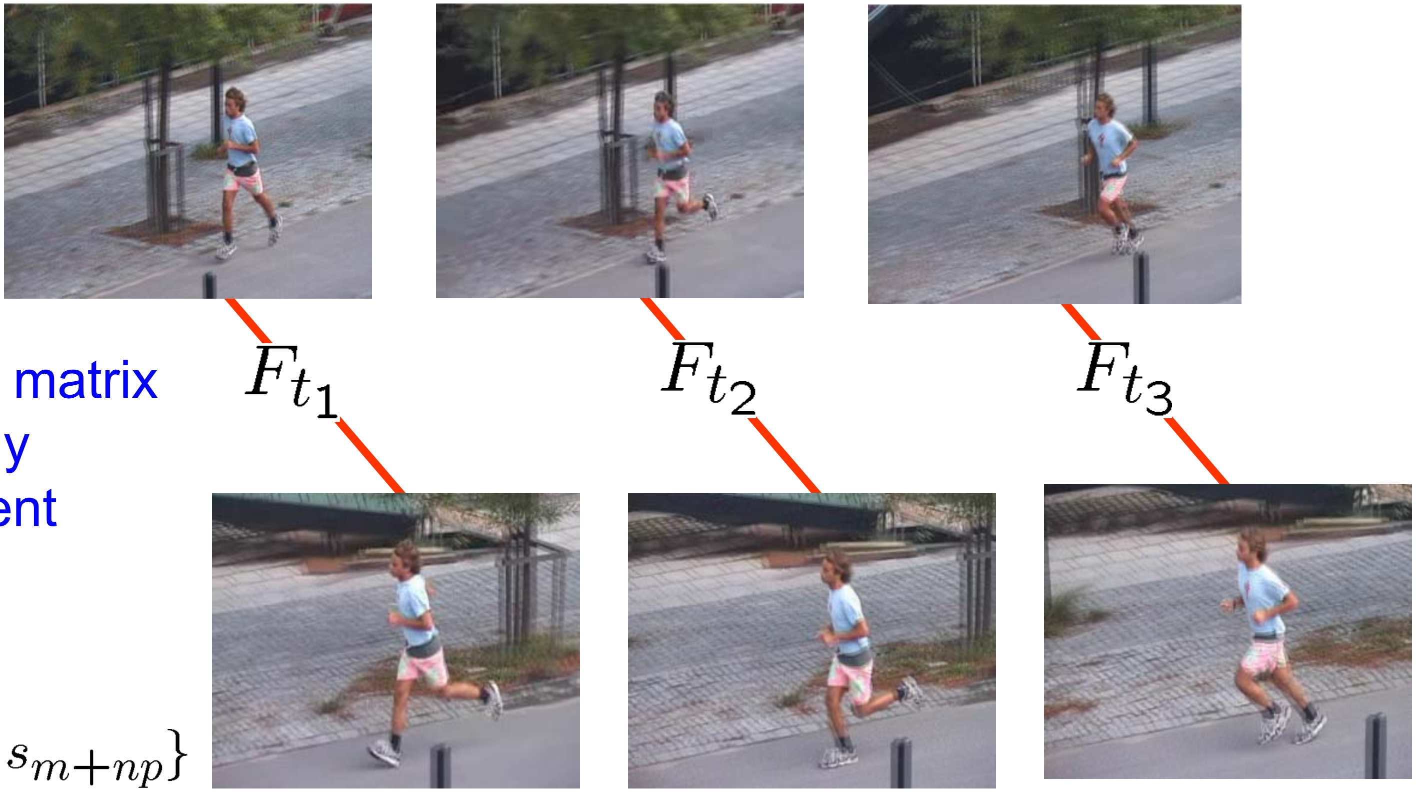
Sequence alignment

Periodic motion detection

Periodic motion detection

Periodic views can be approximately treated as stereopairs

$\{s_t, \ldots, s_m\}$



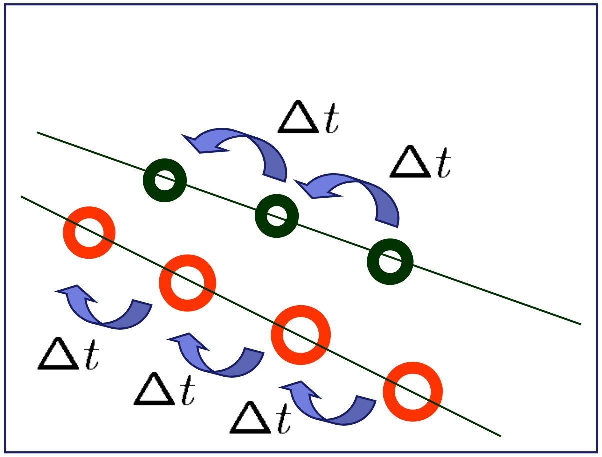
$\{s_{t+np}, \ldots, s_{m+np}\}$

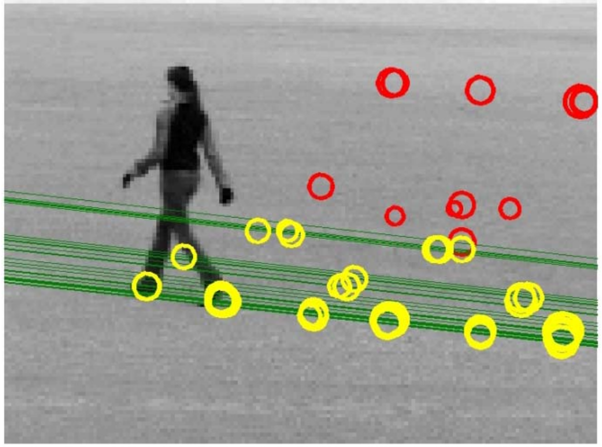
Fundamental matrix F is generally time-dependent

$\Rightarrow$ Periodic motion estimation ~ sequence alignment

Periodic motion detection

1. Corresponding points have similar descriptors





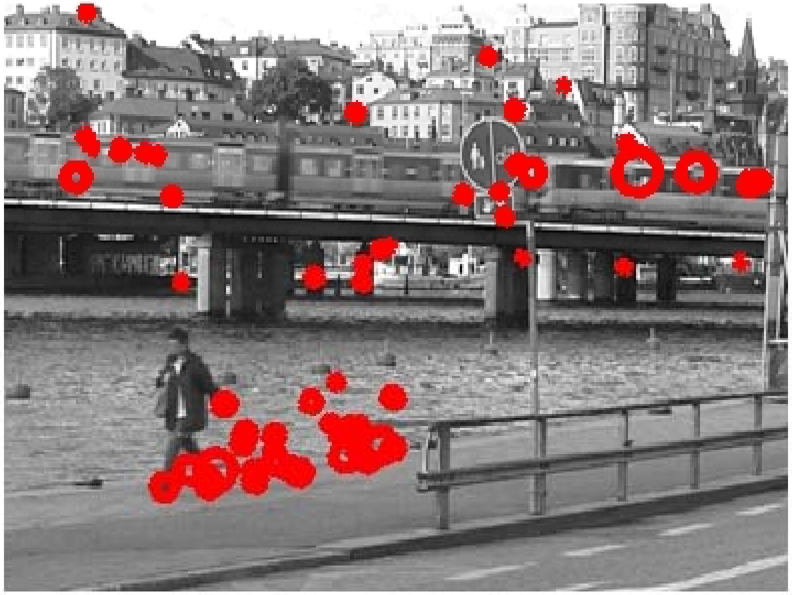
2. Same period $p = \Delta t$ for all features

3. For constant gross motion of the object, spatial arrangement of features across periods satisfy epipolar constraint: $[x^t]' F x^{t+p} = 0$

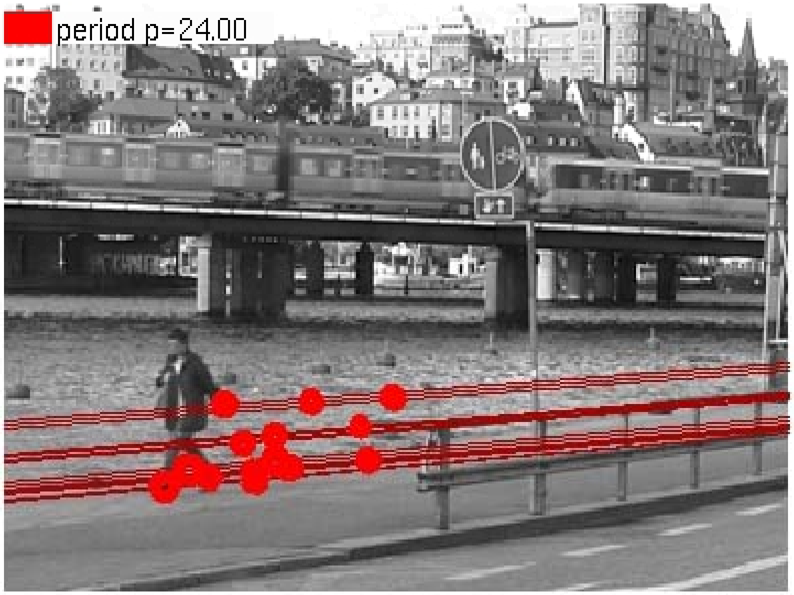
$\Rightarrow$ Use RANSAC to estimate *F* and *p*

Periodic motion detection

Original space-time features

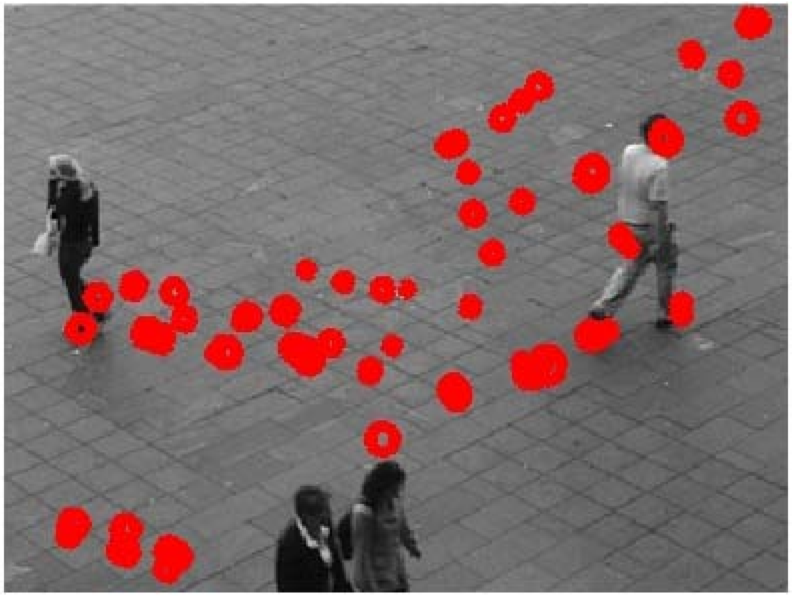


RANSAC estimation of F,p

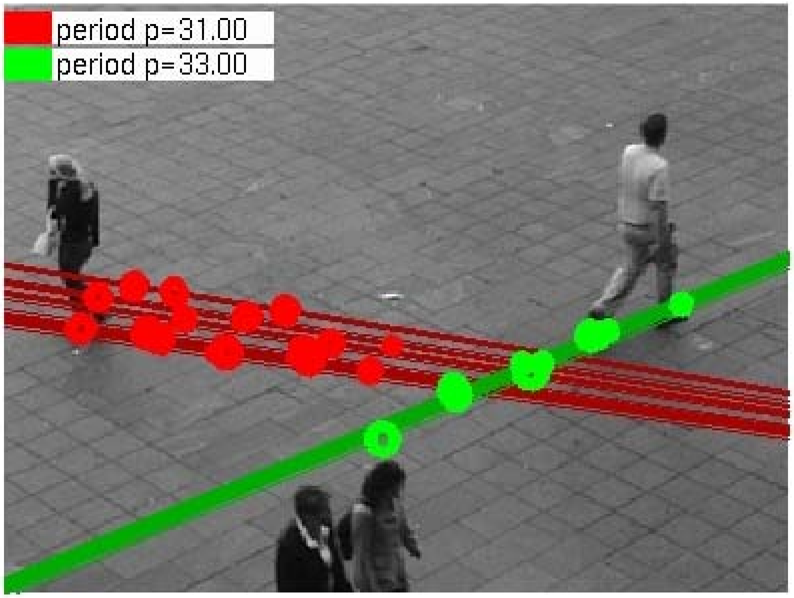


Periodic motion detection

Original space-time features

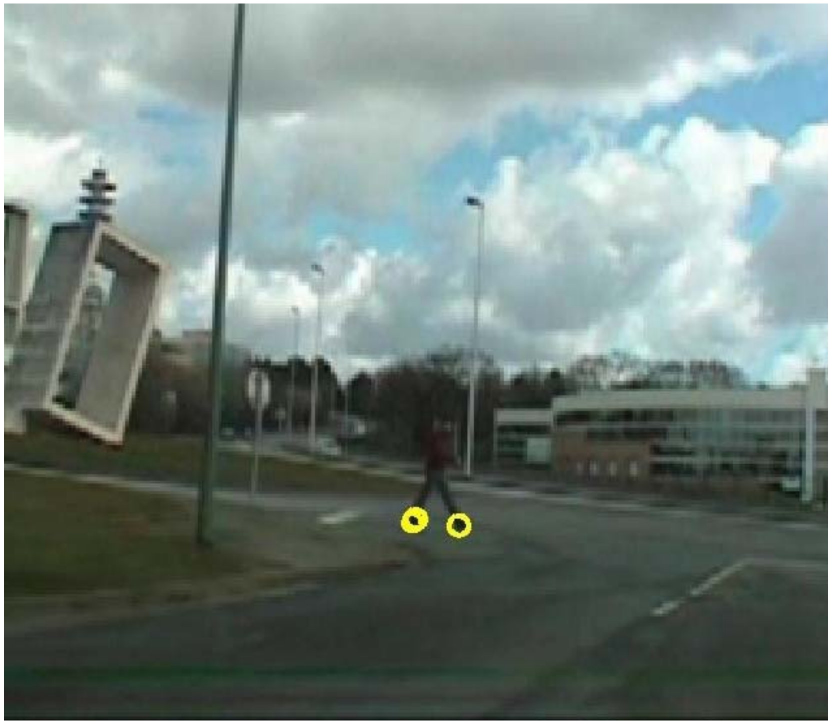


RANSAC estimation of F,p

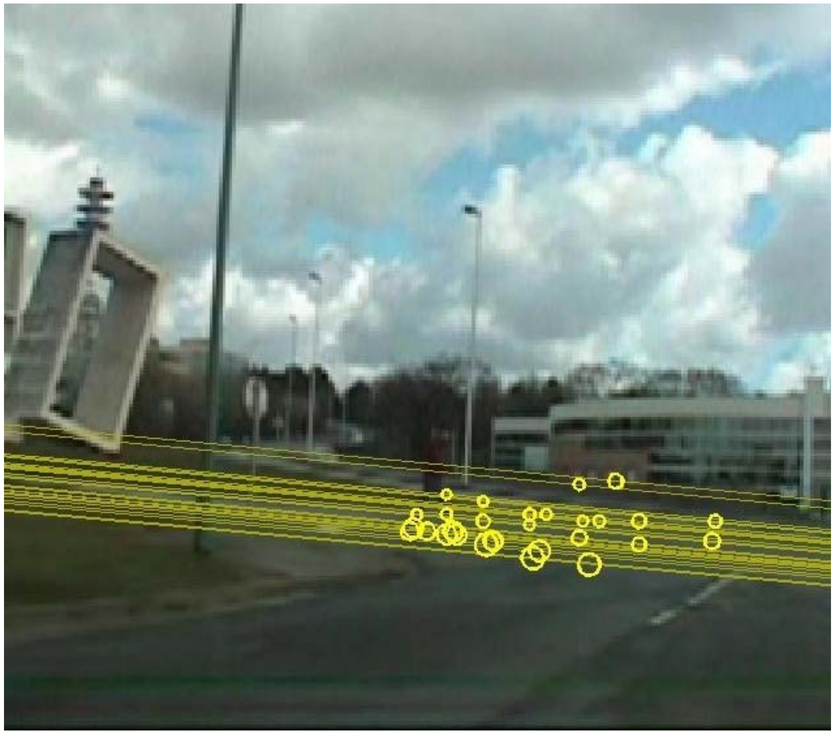


Periodic motion detection

Original space-time features



RANSAC estimation of F,p



Periodic motion segmentation

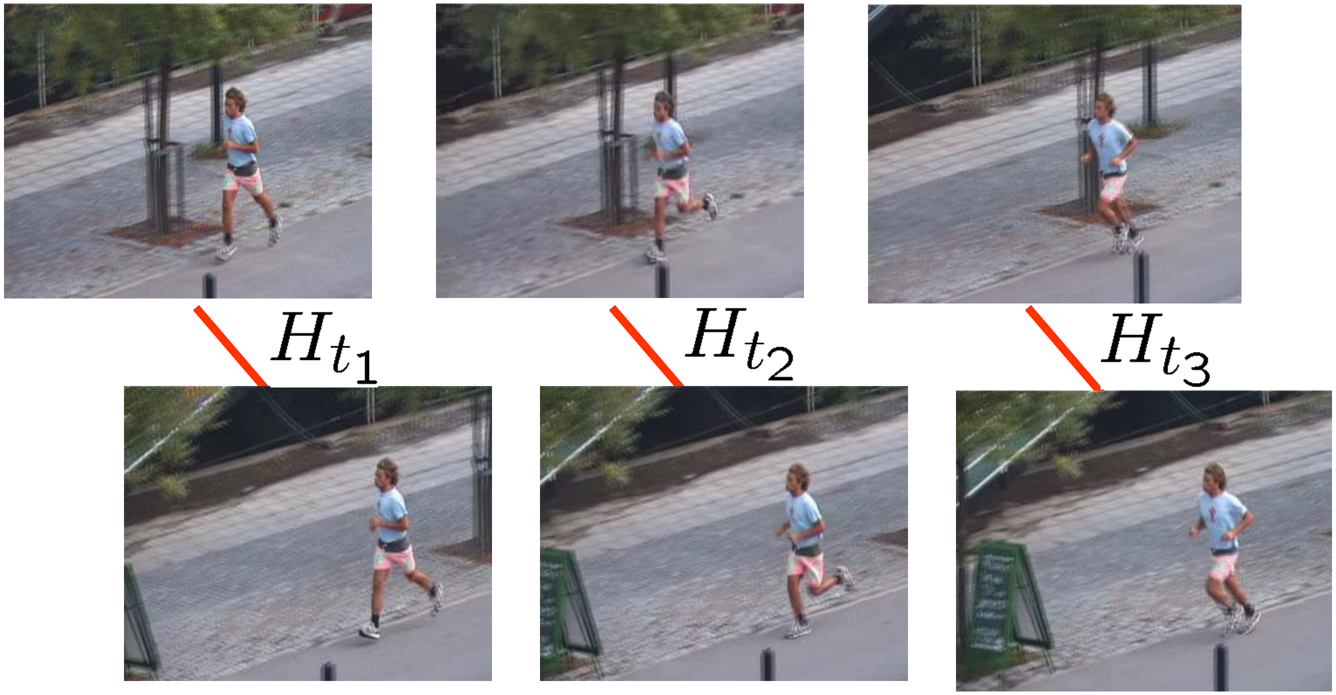
Assume periodic objects are planar

⇒ Periodic points can be related by a *dynamic homography:*

$x_t = Hx_{t+p}$ with

linear in time

$$H(t) = I + p(\mathbf{v}\mathbf{n}^\top - \mathbf{n}^\top\mathbf{v}I)/d - \boxed{t}\mathbf{n}^\top\mathbf{v}I/d$$



Periodic motion segmentation

Assume periodic objects are planar

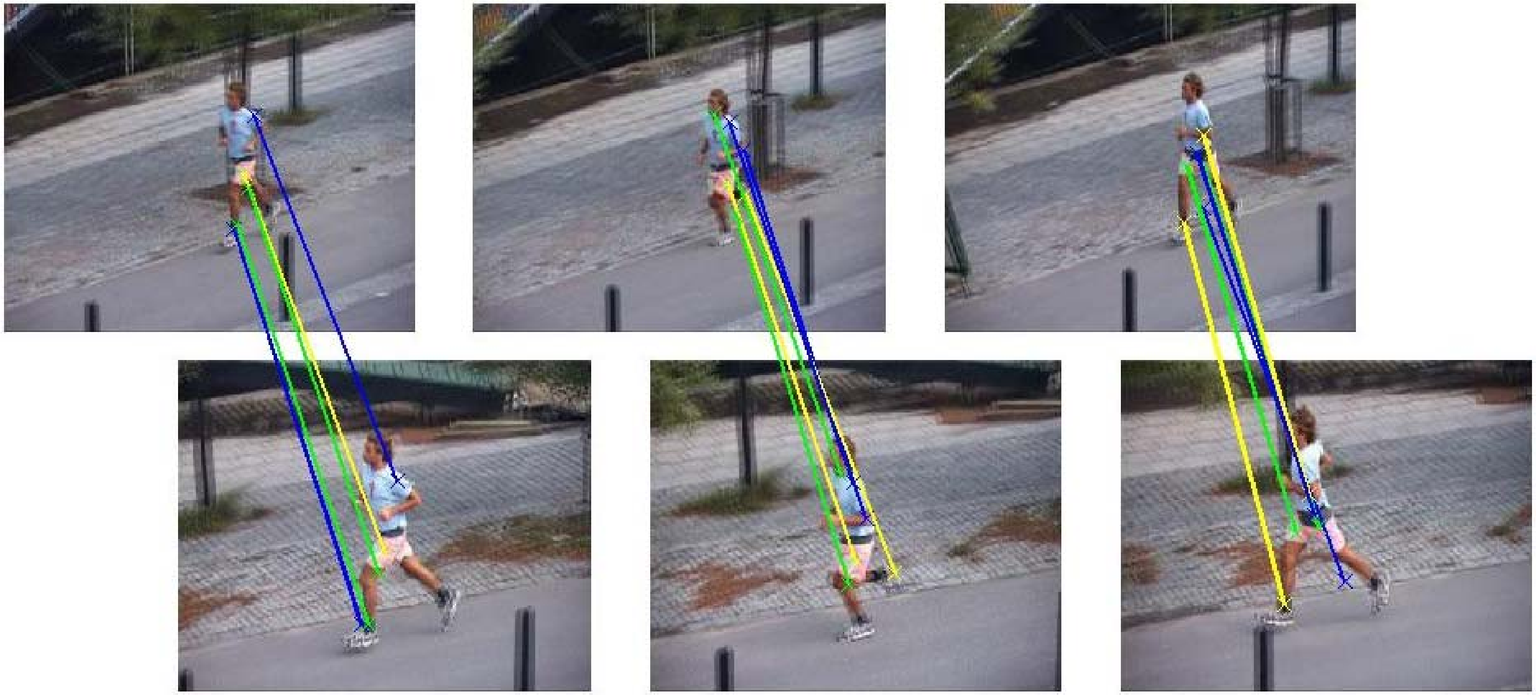
⇒ Periodic points can be related by a *dynamic homography:*

$x_t = H x_{t+p}$ with

linear in time

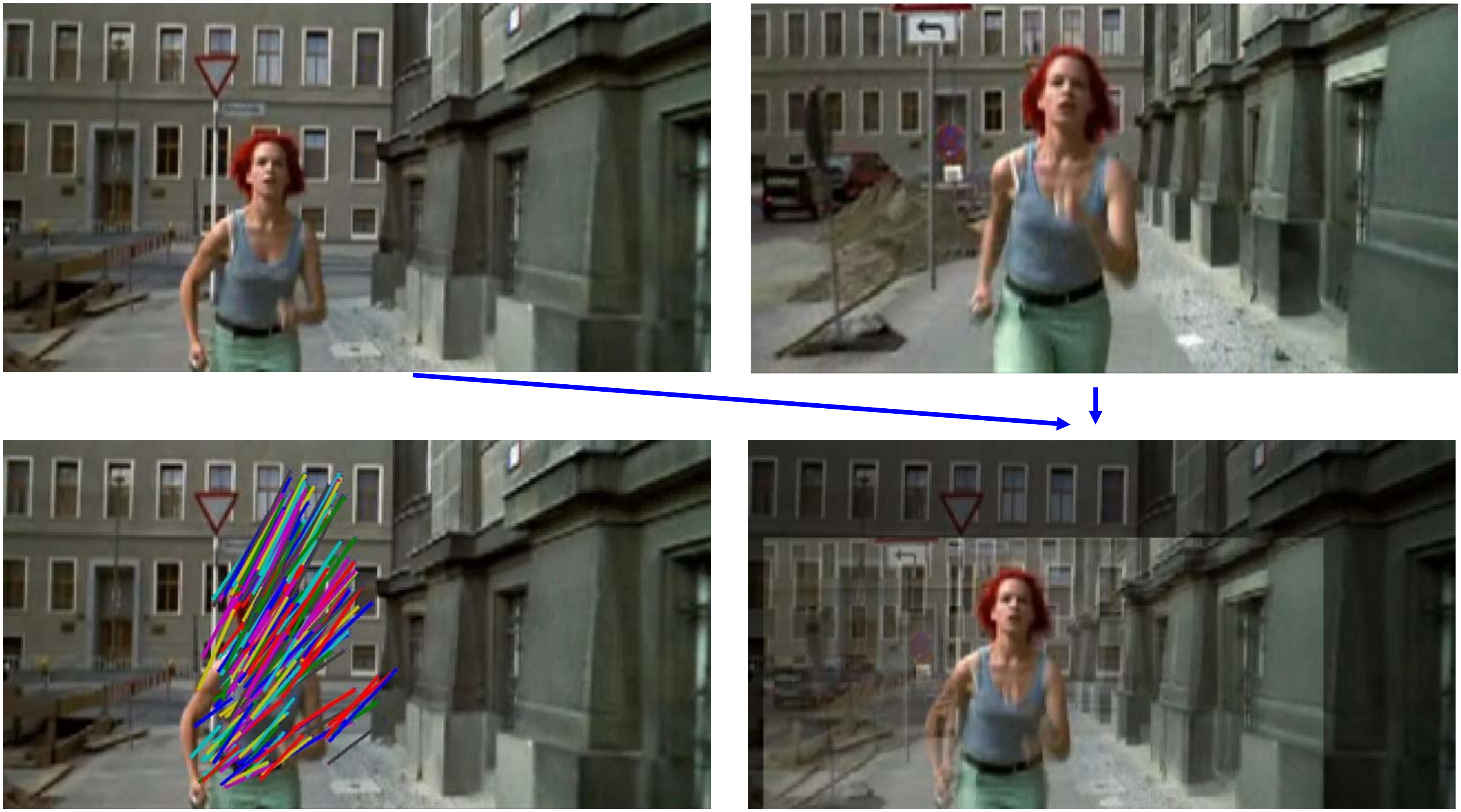
$$H(t) = I + p(\mathbf{v}\mathbf{n}^\top - \mathbf{n}^\top\mathbf{v}I)/d - \boxed{t}\mathbf{n}^\top\mathbf{v}I/d$$

⇒ RANSAC estimation of *H* and *p*

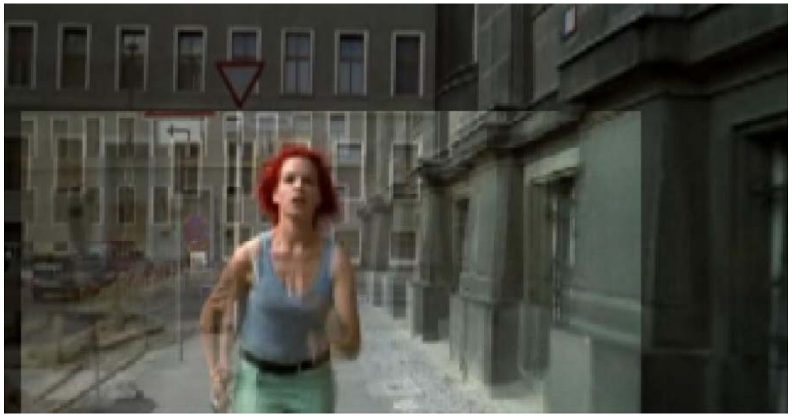


Periodic motion segmentation

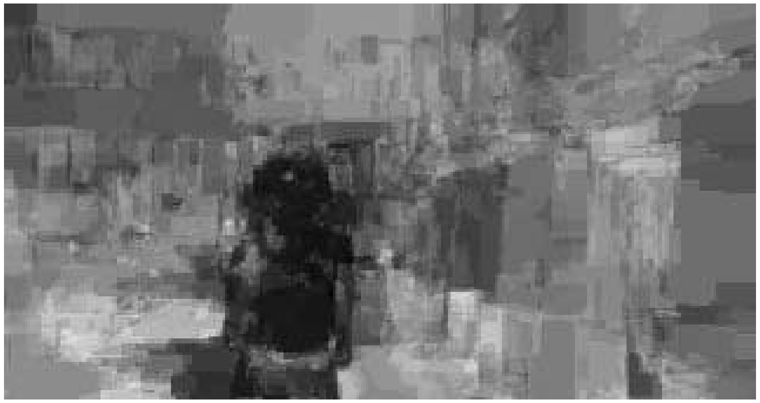
Object-centered stabilization



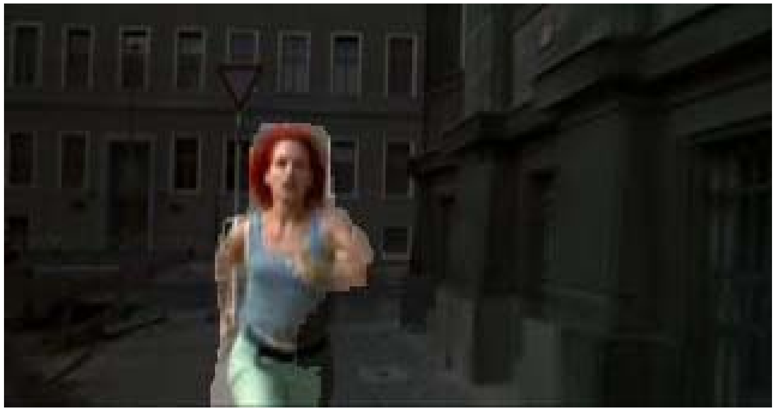
Periodic motion segmentation



Disparity estimation



Graph-cut segmentation



Periodic motion segmentation

Questions

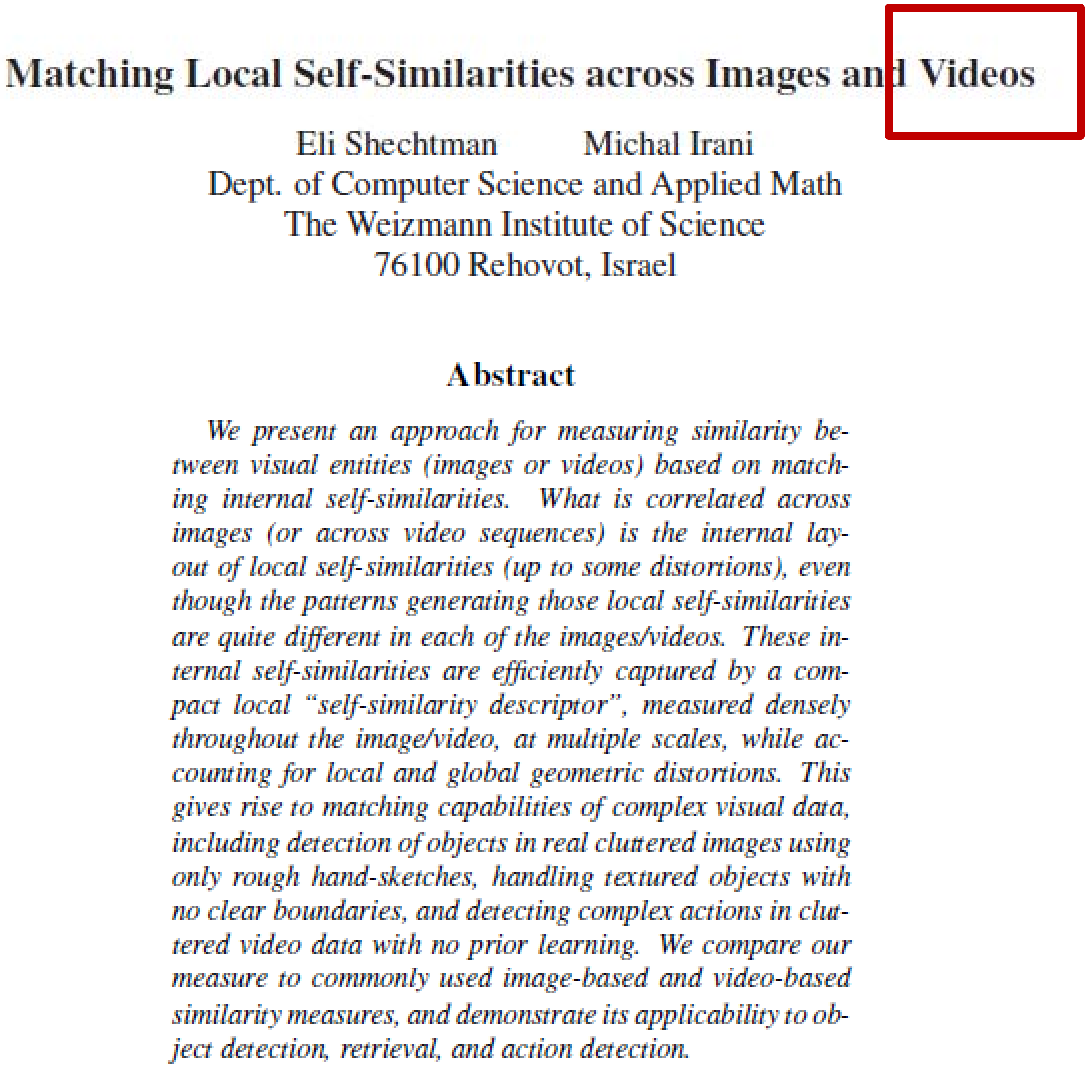
- How to find informative neighborhoods? (ICCV'03)
- How to deal with transformations in the data? (ICCV'03)
- How to describe the neighborhoods? (SCMVP'04)
- How to use obtained features for applications? (ICCV'05)
    - Action recognition
    - Sequence alignment
    - Periodic motion detection

Related work

- Zelnik and Irani CVPR’01
- Efros et.al. ICCV’03
- Lowe ICCV’99
- Mikolayczyk and Schmid CVPR’03, ECCV’02
- Fablet, Bouthemy and Peréz PAMI’02
- Harris and Stephens Alvey’88
- Koenderink and Doorn PAMI 1992
- Lindeberg IJCV 1998

Summary

- Detection of local space-time interest points
- Adaptation to scale and velocity transformations
- Evaluation of local space-time descriptors
- Applications: action recognition, sequence alignment, periodic motion detection, ... ?



Matching Local Self-Similarities across Images and Videos

Eli Shechtman Michal Irani
Dept. of Computer Science and Applied Math
The Weizmann Institute of Science
76100 Rehovot, Israel

Abstract

We present an approach for measuring similarity between visual entities (images or videos) based on matching internal self-similarities. What is correlated across images (or across video sequences) is the internal layout of local self-similarities (up to some distortions), even though the patterns generating those local self-similarities are quite different in each of the images/videos. These internal self-similarities are efficiently captured by a compact local "self-similarity descriptor", measured densely throughout the image/video, at multiple scales, while accounting for local and global geometric distortions. This gives rise to matching capabilities of complex visual data, including detection of objects in real cluttered images using only rough hand-sketches, handling textured objects with no clear boundaries, and detecting complex actions in cluttered video data with no prior learning. We compare our measure to commonly used image-based and video-based similarity measures, and demonstrate its applicability to object detection, retrieval, and action detection.

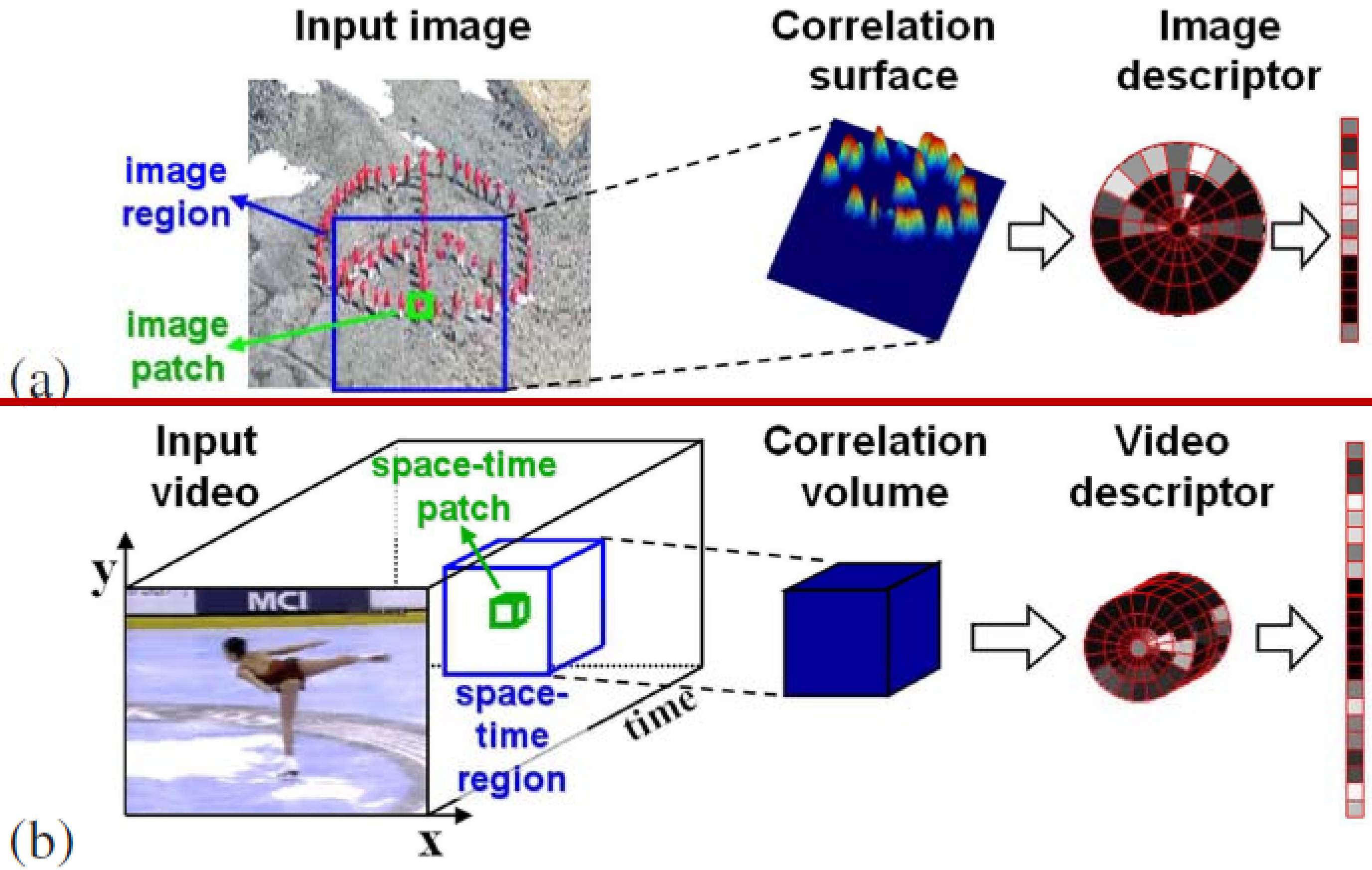
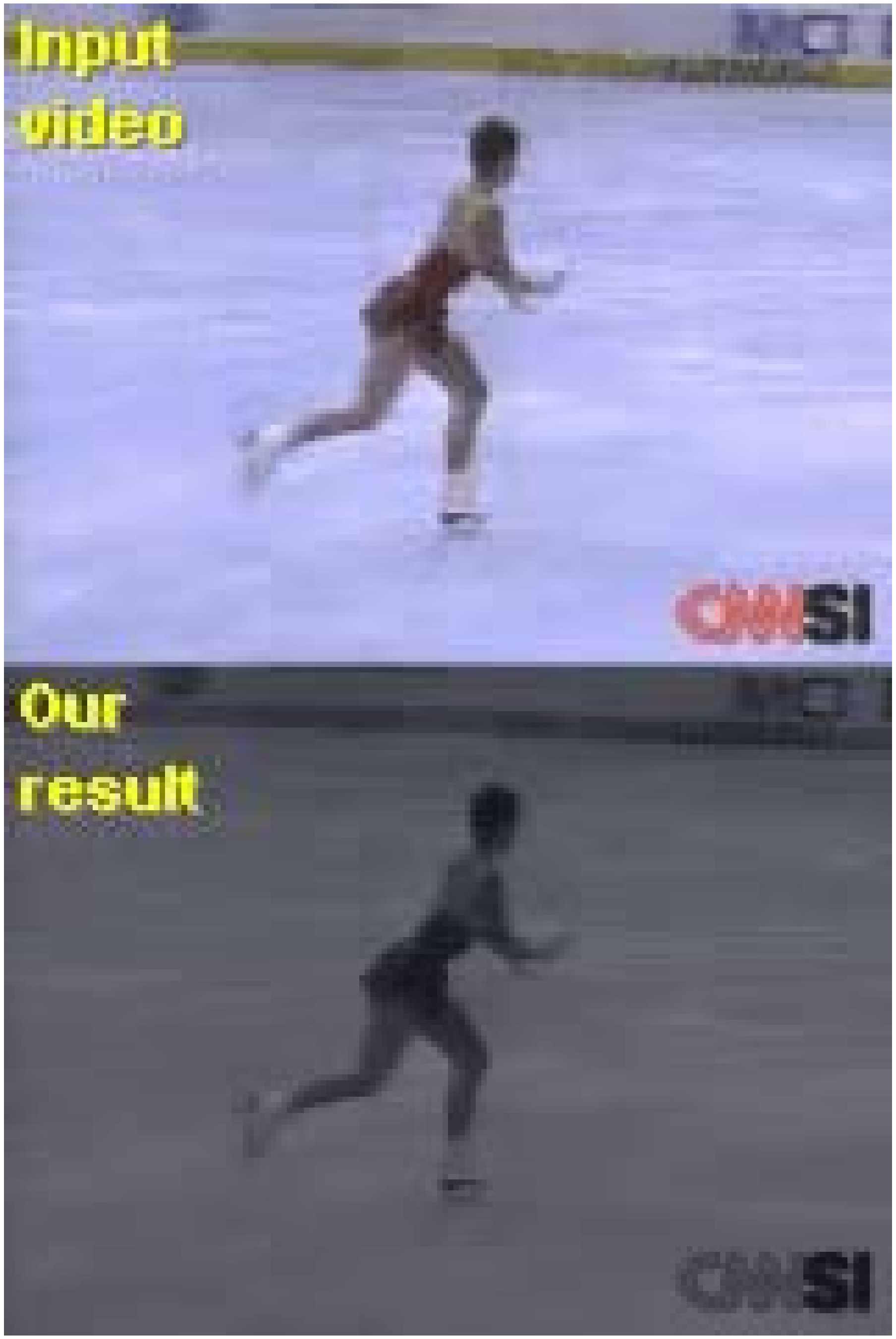
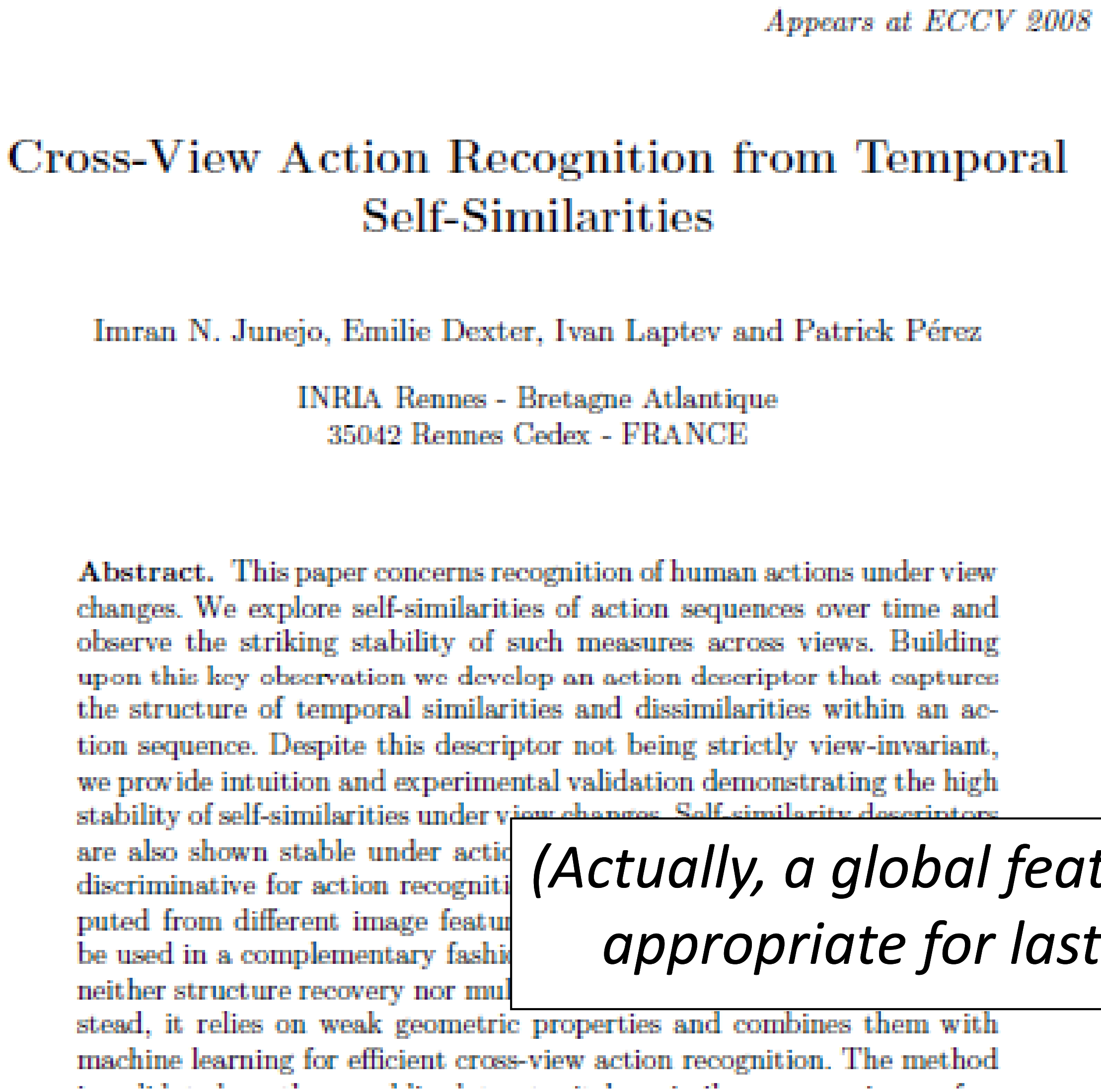


Figure 2. **Extracting the local “self-similarity” descriptor.** *(a) at an image pixel. (b) at a video pixel.*

Turn 1
Turn 2


Input video
Our result

(not on the reading list, but a nice ending to the lecture…)



Appears at ECCV 2008

Cross-View Action Recognition from Temporal Self-Similarities

Imran N. Junejo, Emilie Dexter, Ivan Laptev and Patrick Pérez

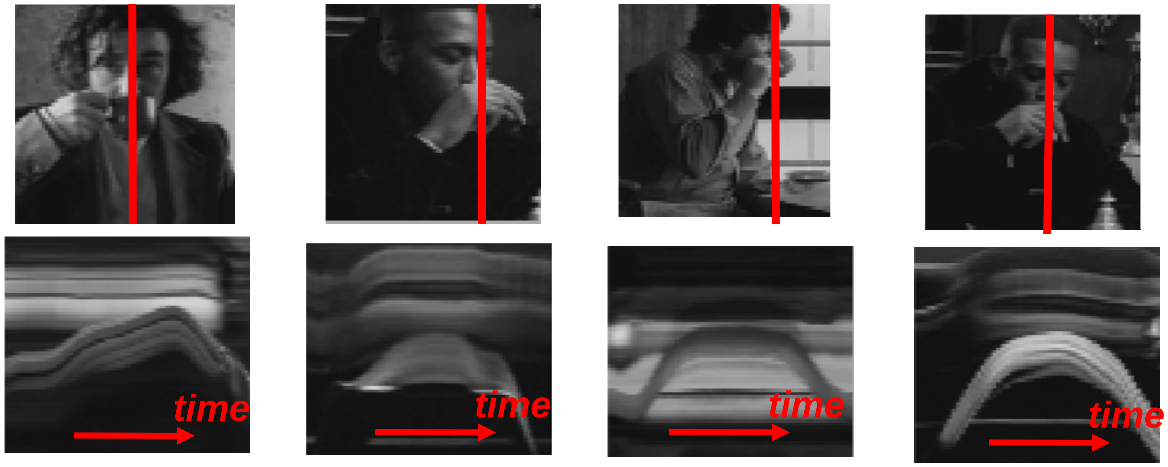
INRIA Rennes - Bretagne Atlantique
35042 Rennes Cedex - FRANCE

Abstract. This paper concerns recognition of human actions under view changes. We explore self-similarities of action sequences over time and observe the striking stability of such measures across views. Building upon this key observation we develop an action descriptor that captures the structure of temporal similarities and dissimilarities within an action sequence. Despite this descriptor not being strictly view-invariant, we provide intuition and experimental validation demonstrating the high stability of self-similarities under view changes. Self-similarity descriptors are also shown stable under actic
discriminative for action recogniti
puted from different image featu
be used in a complementary fashic
neither structure recovery nor mul
stead, it relies on weak geometric properties and combines them with machine learning for efficient cross-view action recognition. The method

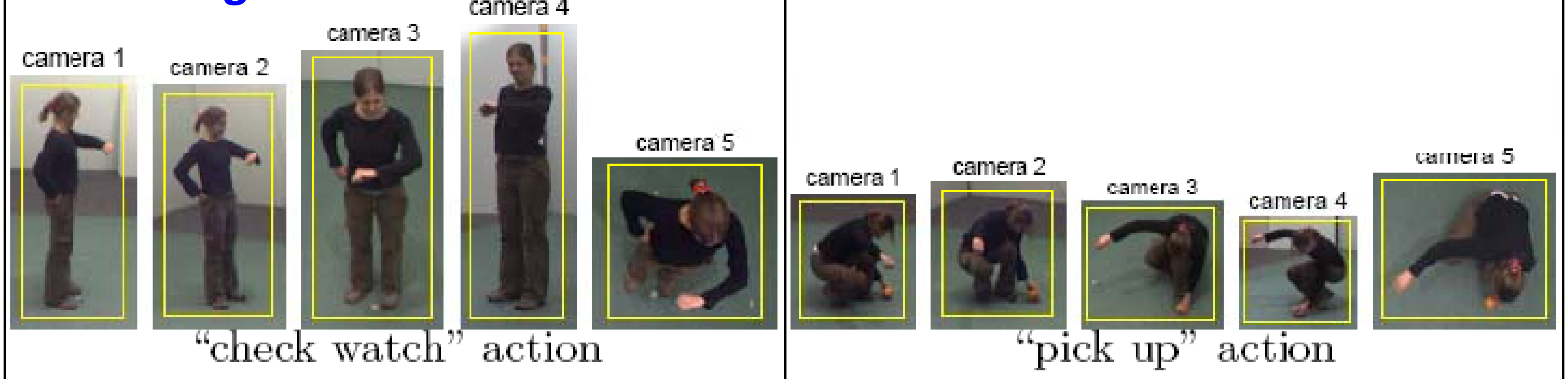
(Actually, a global feature…more appropriate for last week…)

Multi-view action recognition

Motion helps solving multi-view problems?



⇨ **Verify hypothesis and test methods in controlled multi-view settings**



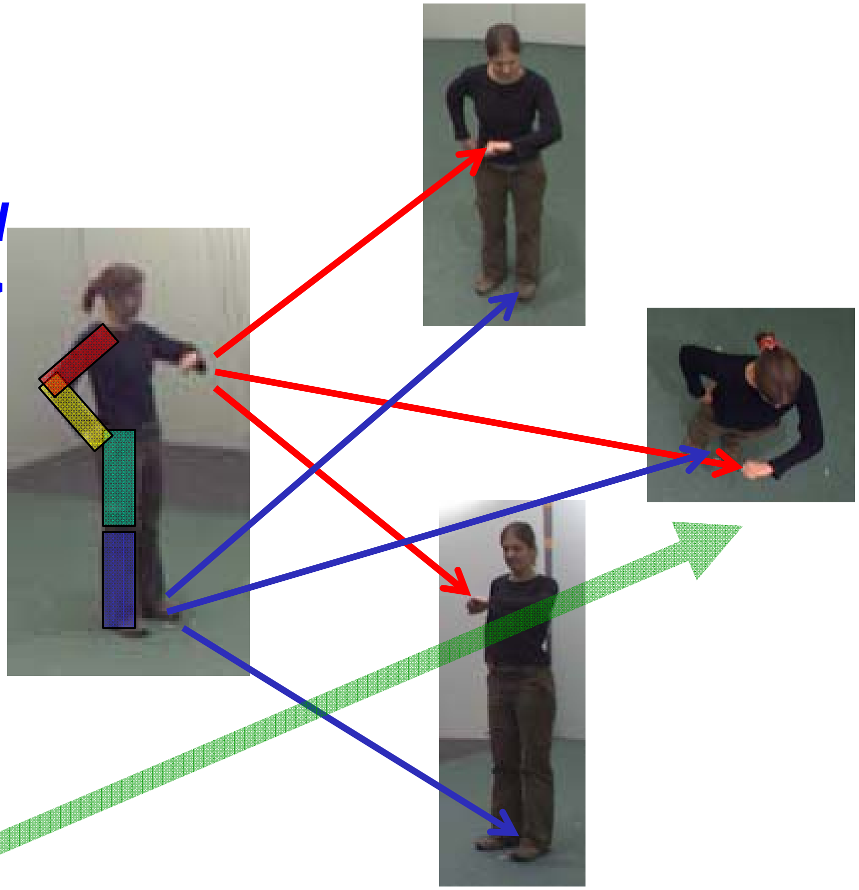
“check watch” action

“pick up” action

Multi-view action recognition

What we DO NOT want to do:

- *Do not want to search for multi-view point correspondence --- Non-rigid motion, cloth changes, … --> It's Hard!*
- *Do not want to identify body parts. Current methods are not reliable enough.*
- *Yet, want to learn actions from one view and to recognize actions in different views*

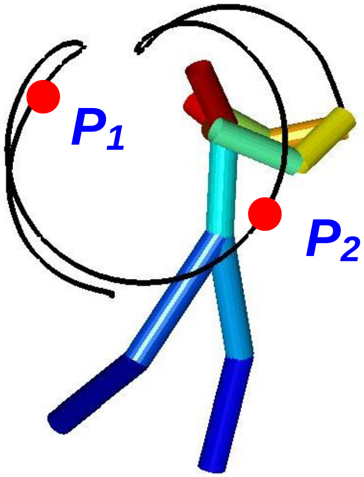


Temporal self-similarities

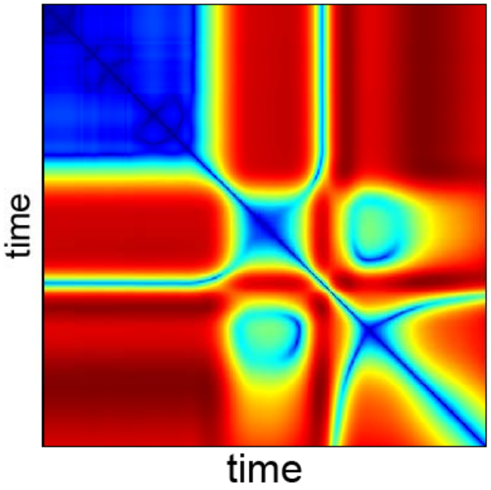
Ideas:

- ***Cross-view matching is hard but cross-time matching (tracking) is relatively easy.***
- ***Measure self-(dis)similarities across tin*** $\mathcal{D}(t_1,t_2),\ t_1,t_2 \in (1,...,T)$

Example: $\mathcal{D}(t_1,t_2) = ||P_1 - P_2||_2$

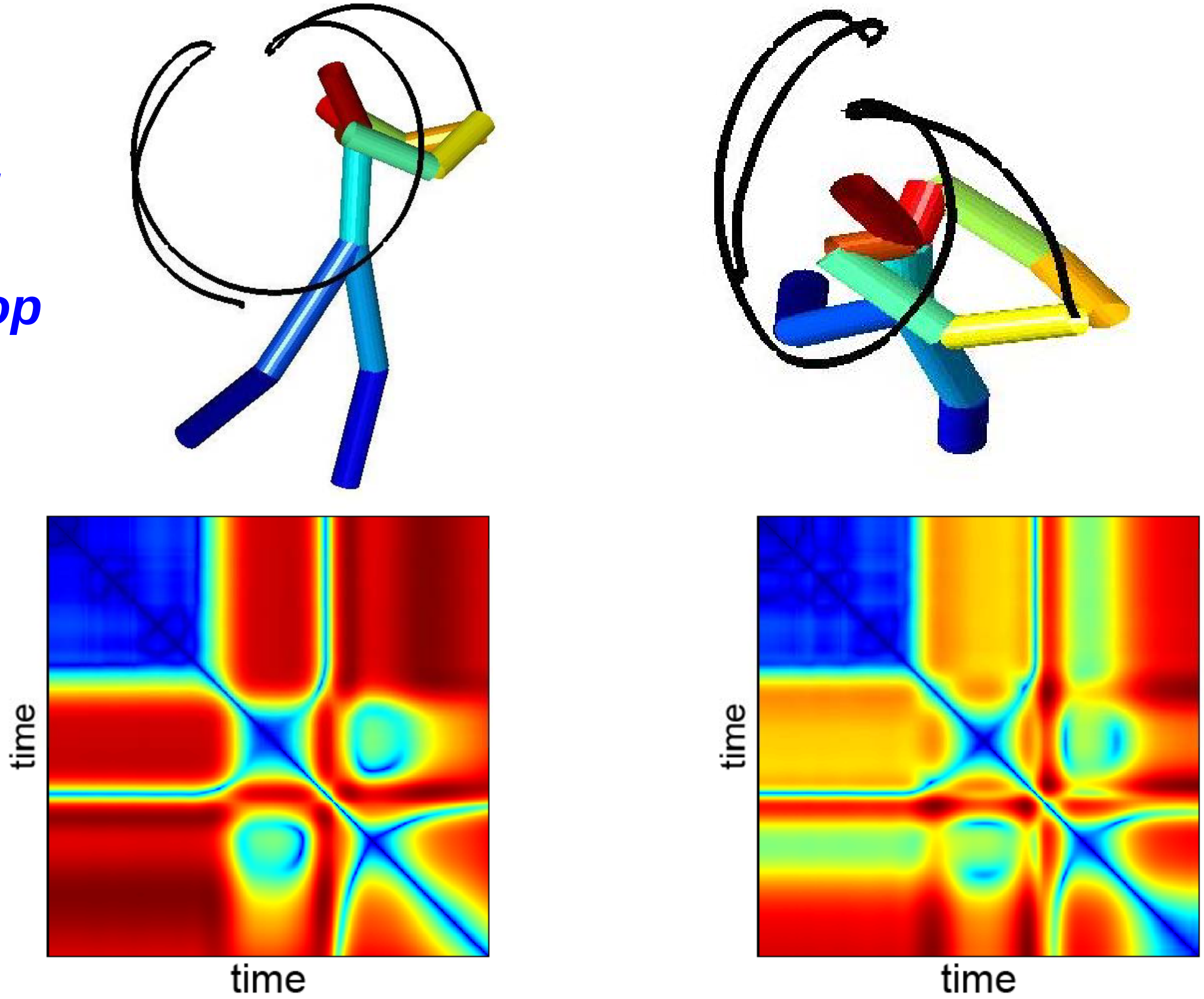


Distance matrix / self-similarity matrix (SSM):



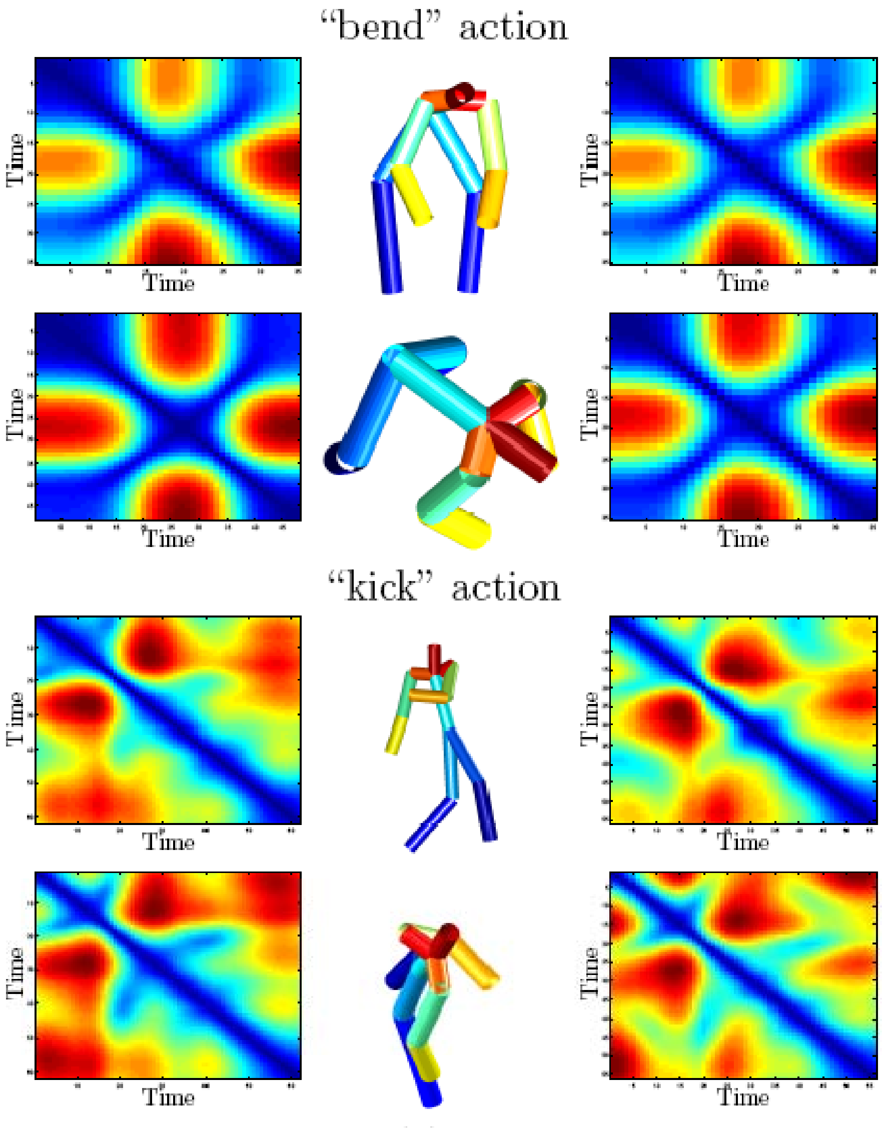
Temporal self-similarities: Multi-views

Example:
Golf swing
from the
side and top
views



Cross-View Action Recognition from Temporal Self-Similarities
I. Junejo, E. Dexter, I. Laptev, and P. Perez, ECCV 2008

Temporal self-similarities: MoCap



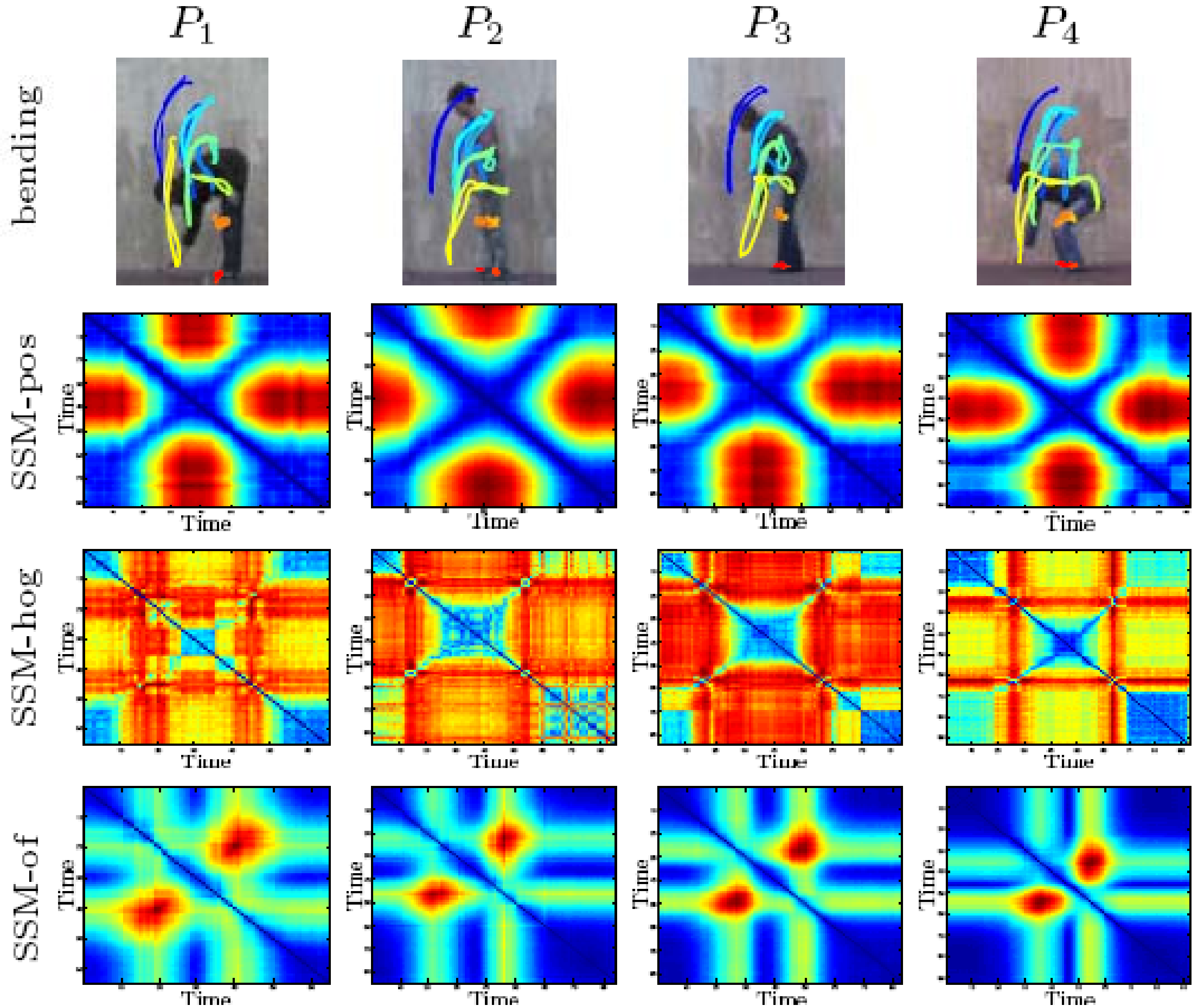
person 1

person 2

person 1

person 2

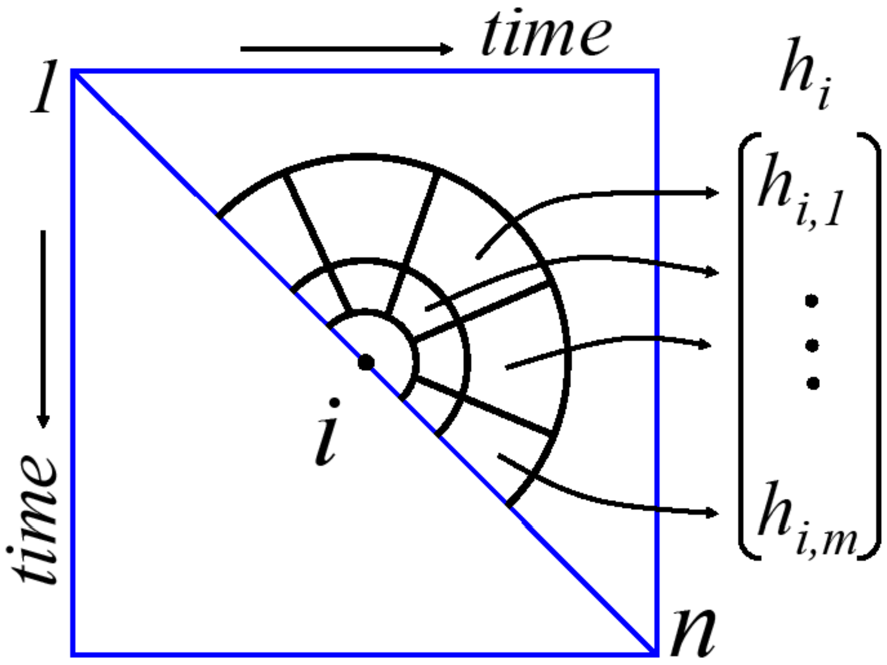
Temporal self-similarities: Video



Self-similarity descriptor

Properties of SSM:

- SPSD
- 0-valuaed diagonal
- uncertainty increases with the distance from the diagc$\Delta t \doteq t_2 - t_1$



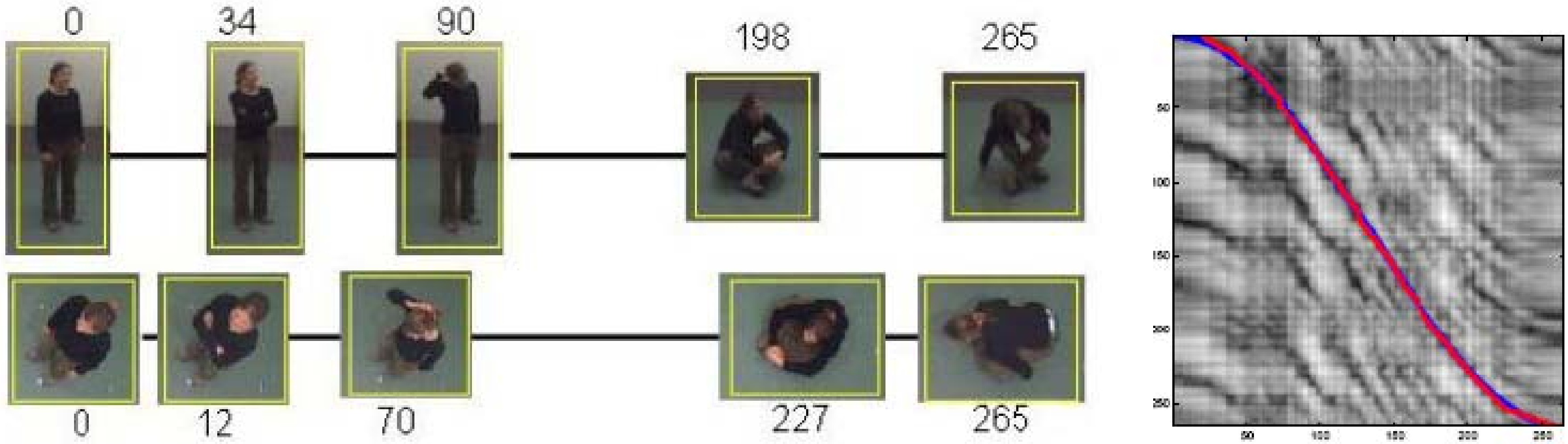
- Define a local histogram descriptor h_i for each point i on the diagonal.

- Sequence alignment:

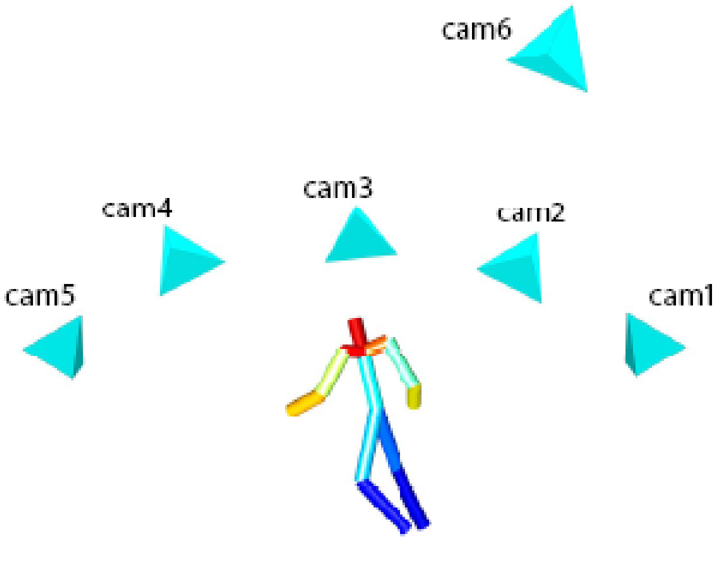
  DP for two sequences of descriptors {h_i}, {h_j}

- Action recognition:
  - Visual vocabulary for h
  - BoF representation of {h_i}
  - SVM

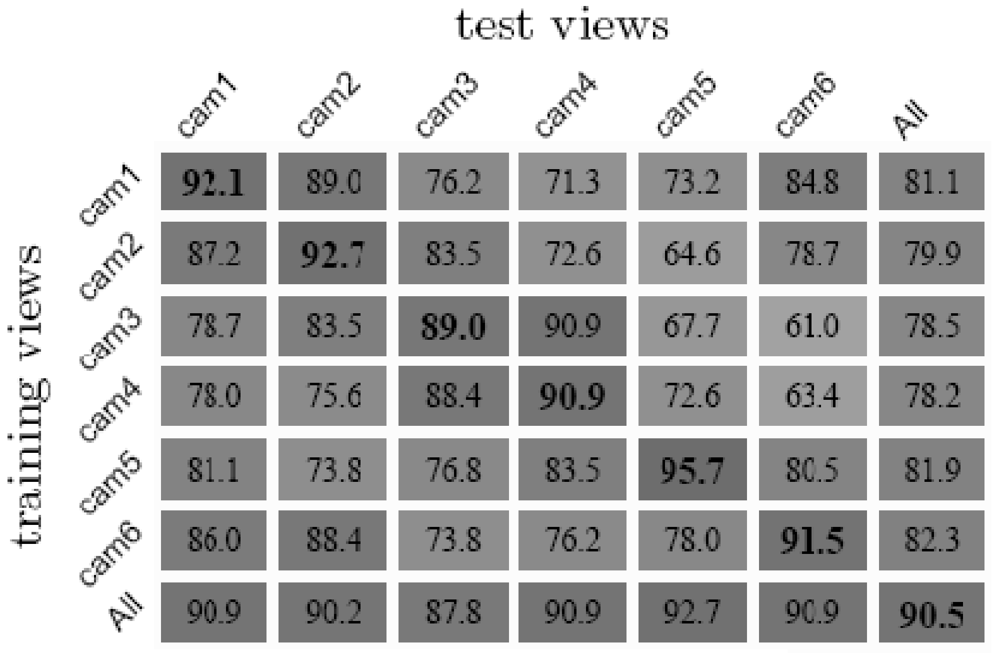
Multi-view alignment



Multi-view action recognition: MoCap



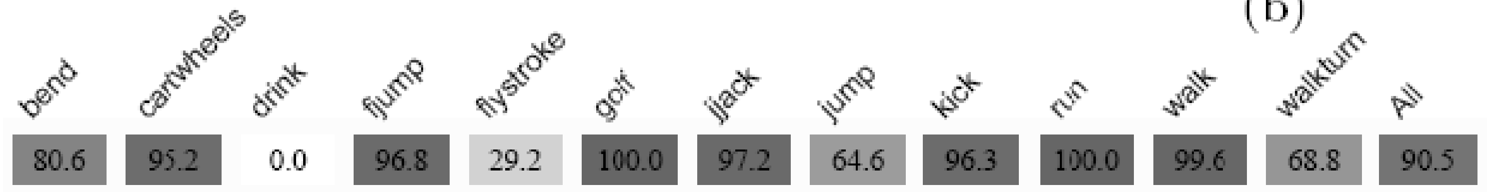
(a)



test views / training views

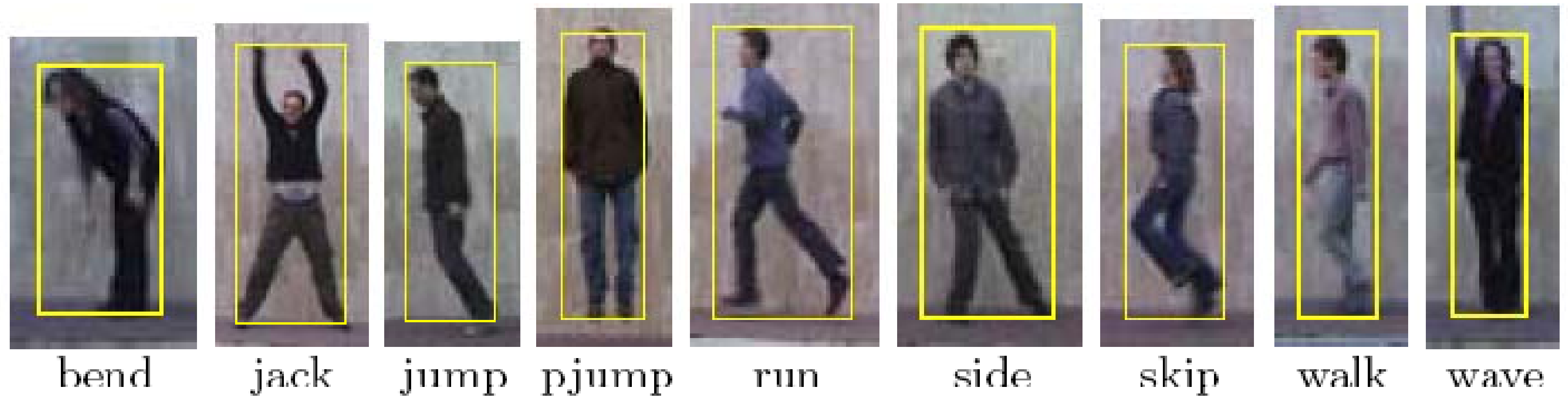
	cam1	cam2	cam3	cam4	cam5	cam6	All
cam1	**92.1**	89.0	76.2	71.3	73.2	84.8	81.1
cam2	87.2	**92.7**	83.5	72.6	64.6	78.7	79.9
cam3	78.7	83.5	**89.0**	90.9	67.7	61.0	78.5
cam4	78.0	75.6	88.4	**90.9**	72.6	63.4	78.2
cam5	81.1	73.8	76.8	83.5	**95.7**	80.5	81.9
cam6	86.0	88.4	73.8	76.2	78.0	**91.5**	82.3
All	90.9	90.2	87.8	90.9	92.7	90.9	**90.5**

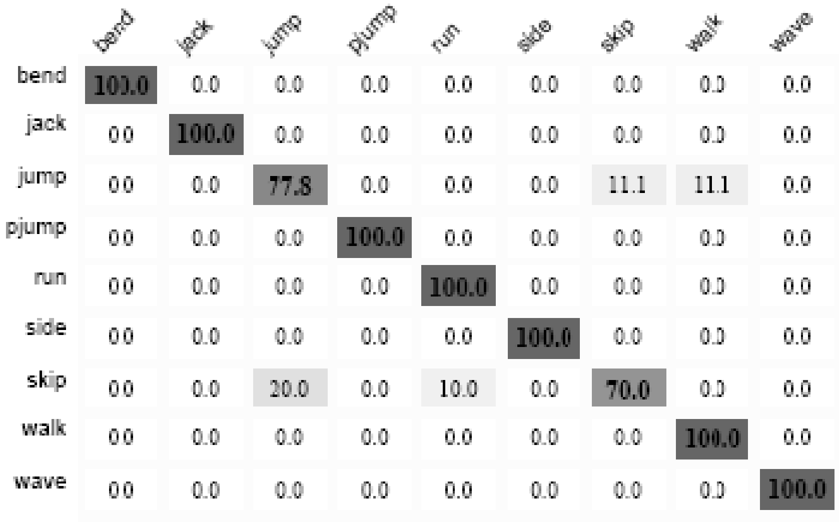
(b)



bend	cartwheels	drink	fjump	flystroke	golf	jjack	jump	kick	run	walk	walkturn	All
80.6	95.2	0.0	96.8	29.2	100.0	97.2	64.6	96.3	100.0	99.6	68.8	90.5

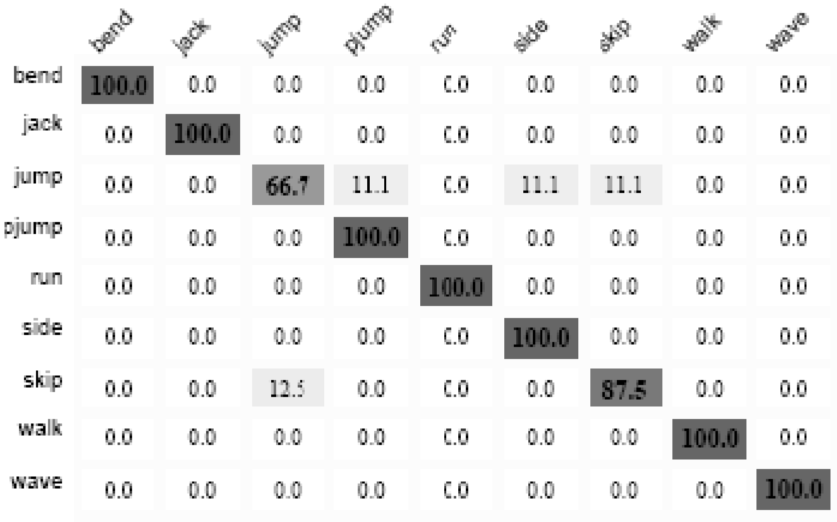
Single-view action recognition: Video





	bend	jack	jump	pjump	run	side	skip	walk	wave
bend	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jack	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jump	0.0	0.0	77.8	0.0	0.0	0.0	11.1	11.1	0.0
pjump	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0
run	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
side	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0
skip	0.0	0.0	20.0	0.0	10.0	0.0	70.0	0.0	0.0
walk	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0
wave	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0

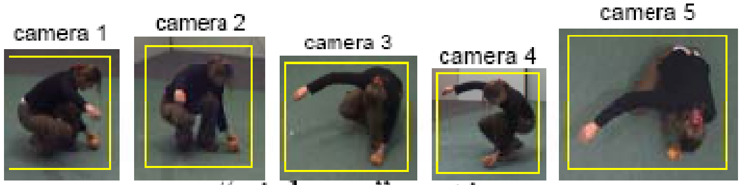
OF-based self-similarities



	bend	jack	jump	pjump	run	side	skip	walk	wave
bend	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jack	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jump	0.0	0.0	66.7	11.1	0.0	11.1	11.1	0.0	0.0
pjump	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0
run	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
side	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0
skip	0.0	0.0	12.5	0.0	0.0	0.0	87.5	0.0	0.0
walk	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0
wave	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0

Trajectory-based self-similarities

Multi-view action recognition: Video



“pick up” action

training views \ test views	cam1	cam2	cam3	cam4	cam5	All
cam1	**76.4**	77.6	69.4	70.3	44.8	67.2
cam2	77.3	**77.6**	73.9	67.3	43.9	67.4
cam3	66.1	70.6	**73.6**	63.6	53.6	65.0
cam4	69.4	70.0	63.0	**68.8**	44.2	63.9
cam5	39.1	38.8	51.8	34.2	**66.1**	45.2
All	74.8	74.5	74.8	70.6	61.2	**72.7**

	check–watch	cross–arms	scratch–head	sit–down	get–up	turn–around	walk	wave	punch	kick	pick–up
check–watch	83.3	0.0	0.7	1.3	0.7	1.3	8.0	0.7	0.0	0.0	4.0
cross–arms	0.0	94.0	2.0	1.3	0.7	0.7	0.0	0.7	0.0	0.0	0.7
scratch–head	0.0	0.0	68.7	2.0	9.3	2.0	1.3	4.7	10.0	2.0	0.0
sit–down	0.7	4.7	3.3	55.3	1.3	20.0	3.3	0.7	10.7	0.0	0.0
get–up	2.0	3.3	7.3	0.7	69.3	0.7	0.0	23.3	2.7	0.7	0.0
turn–around	3.3	1.3	0.0	27.3	0.0	56.7	3.3	2.0	2.7	0.0	3.3
walk	10.0	0.7	0.0	2.7	0.7	2.7	68.7	1.3	1.3	0.0	12.0
wave	3.3	0.7	6.7	2.0	14.7	0.0	0.7	63.3	8.7	0.0	0.0
punch	0.7	0.0	6.0	6.0	0.7	2.7	0.0	1.3	74.0	8.7	0.0
kick	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0
pick–up	2.0	0.0	0.0	2.7	0.7	4.7	13.3	0.7	0.0	0.0	76.0

	All-to-All
hog	57.8%
of	65.9%
of+ofx+ofy	66.5%
of+hog	71.9%
of+hog+ofx+ofy	**72.7%**

	cam1	cam2	cam3	cam4	cam5
This paper	**76.4%**	**77.6%**	**73.6%**	**68.8%**	**66.1%**
Weinland et al. [12] 3D	65.4%	70.0%	54.3%	66.0%	33.6%
Weinland et al. [12] 2D	55.2%	63.5%	—	60.0%	—

Properties

- No correspondence across views needed
- No body-part identification needed
- Relies on assumptions of person detection and tracking
- SSMs can be computed from different and complementary image measurements: trajectories, OF, HOG, etc.
- Provides only approximate view-invariance but under weak assumptions

Today

- Scale selection [Lindeberg]
- Affine-invariance [Mikolajczyk and Schmid]
- MSER – Stable Regions [Matas et al.]
- SURF -Fast Approximate SIFT [Bay et al.]
- Spatio-Temporal Features [Laptev]
- Self-Similarilty [Shectman and Irani]

- Bonus: Temporal Self-Similarity [Laptev ECCV’08]

Feb 17th – Generative approaches (Constellation, Topic Models, etc.) – *Sudderth guest lecture*

- R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2003, pp. 264-271. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1211479
- J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering object categories in image collections," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2005. http://publications.csail.mit.edu/tmp/MIT-CSAIL-TR-2005-012.ps
- J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," International Journal of Computer Vision. 79(3): 299-318. 2008 Available: http://dx.doi.org/10.1007/s11263-007-0122-4 (Buchsbaum presentation)
- E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, "Describing visual scenes using transformed objects and parts," International Journal of Computer Vision, vol. 77, no. 1, pp. 291-330, May 2008. Available: http://dx.doi.org/10.1007/s11263-007-0069-5

Optional Readings:

- F.-F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2. Washington, DC, USA: IEEE Computer Society, 2005, pp. 524-531. Available: http://dx.doi.org/10.1109/CVPR.2005.16
- P. Moreels and P. Perona, "A probabilistic cascade of detectors for individual object recognition," European Conference on Computer Vision , vol III, pp. 426-439, 2008. Available: http://dx.doi.org/10.1007/978-3-540-88690-7_32

Reminder

Please sign up via email for a paper that you would like to present or show a demonstration of.

- can show demos next week from this week's papers (e.g.,GIST / spatial envelope on some images collected around campus)

- but otherwise should show demo on day of paper (could show Laptev or self-similarity features on Berkeleyish action examples next week…)

DEADLINE FEB 17th

I'll expect two demos or one presentation per person taking the course for credit…

N.B., a demo is more than showing author's videos or canned matlab example…must try on something new or extend…