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

Tinne Tuytelaars

Tutorial ECCV 2006

May 7th, 2006

Overview

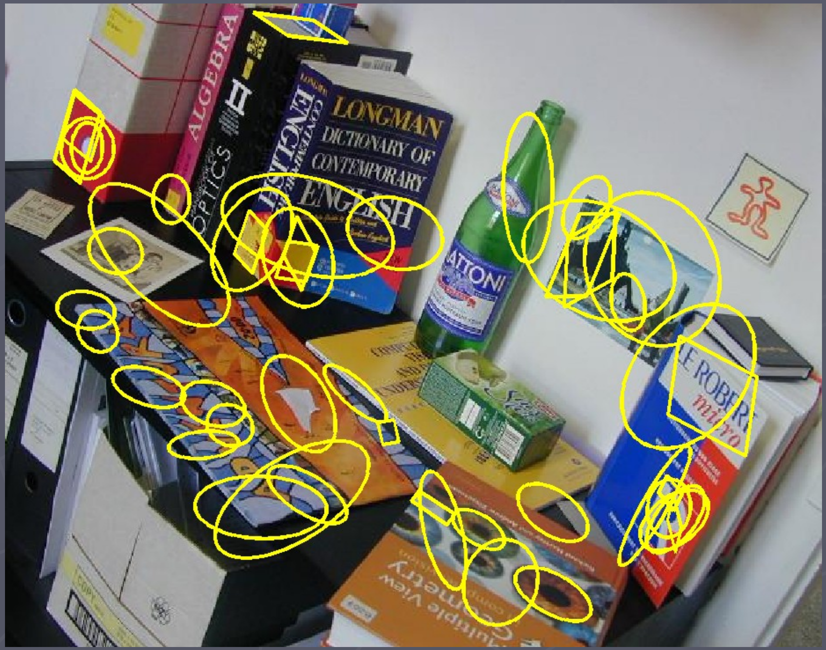
- Local Invariant Features: What? Why?
  - Introduction
  - Overview of existing detectors
  - Quantitative and qualitative comparison
- Local Invariant Features: When? How?
  - Feature descriptors
  - Applications
  - Conclusions

Overview

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

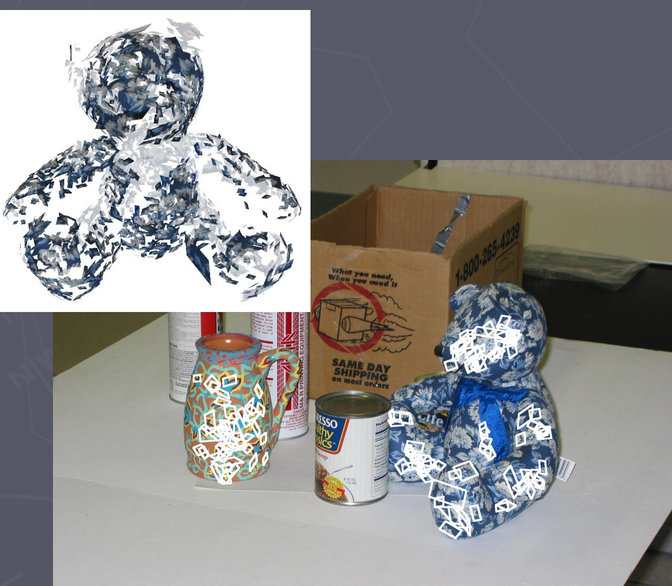
Introduction

Wide baseline matching



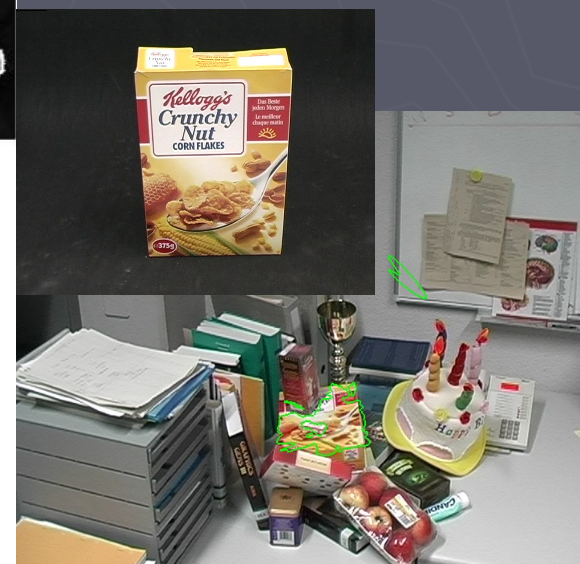
Introduction

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

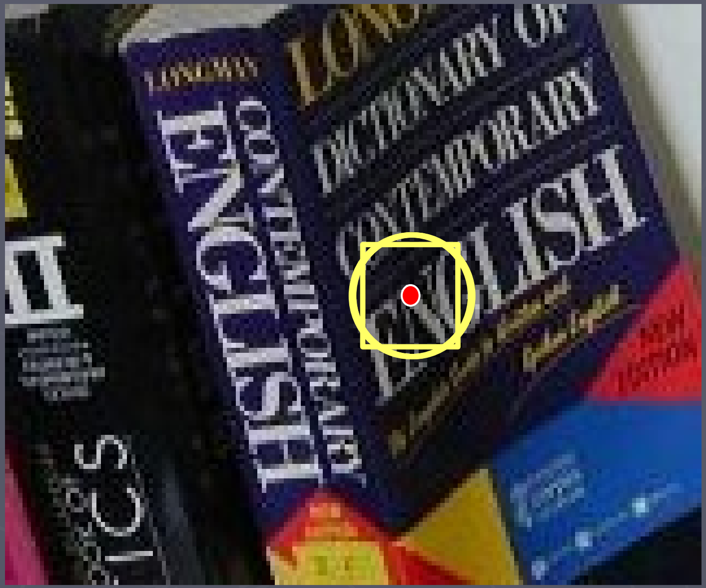
So what's the novelty?

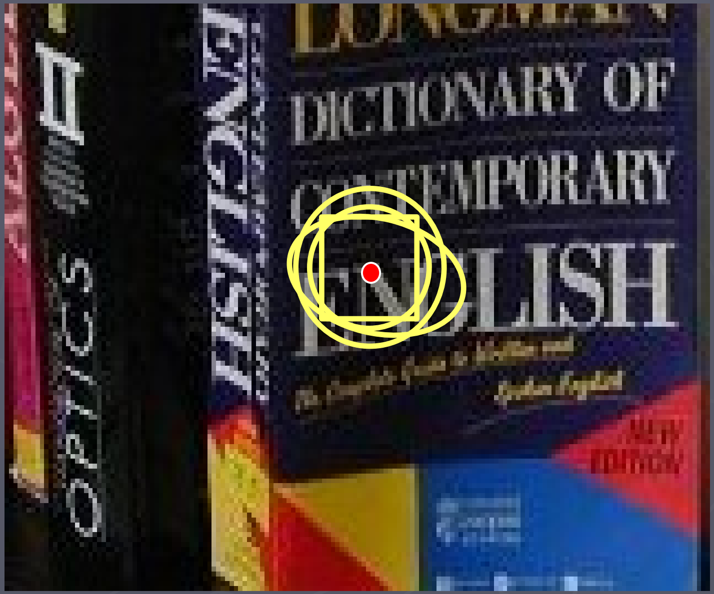
- ~~Local character~~
- Level of invariance
- Local invariant features: a new paradigm
  - Not just a method to select interesting locations in the image, or to speed up analysis
  - But rather a new image representation, that allows to describe the objects / parts without the need for segmentation

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.

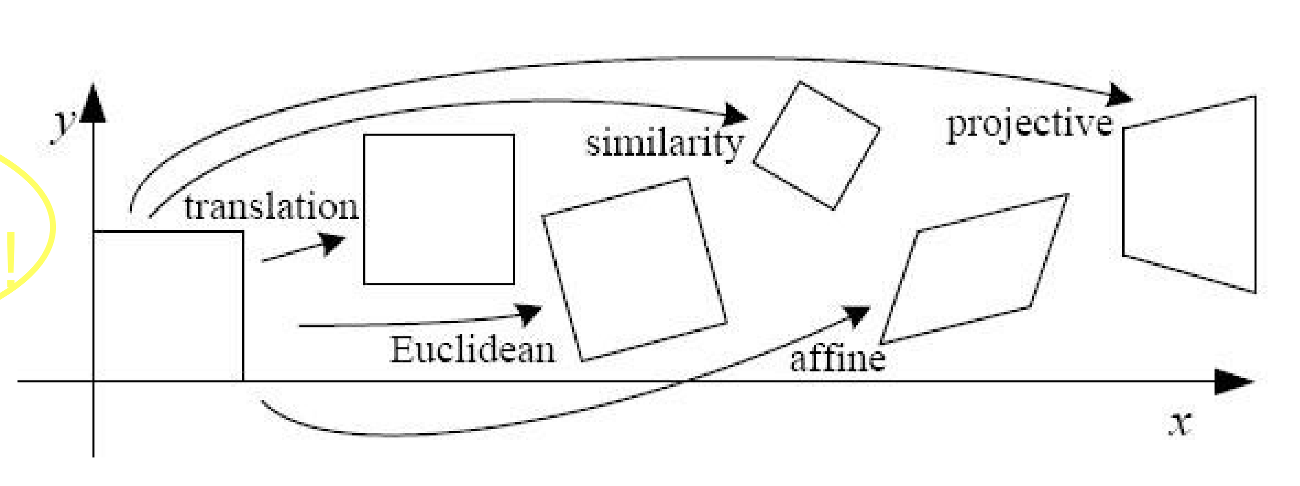
a covariant measure changes in a way consistent with the image transformation.

Terminology: ‘detector’ or ‘extractor’

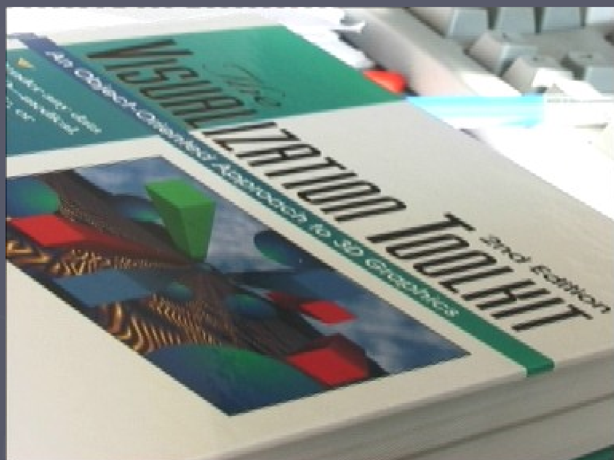
Geometric transformations

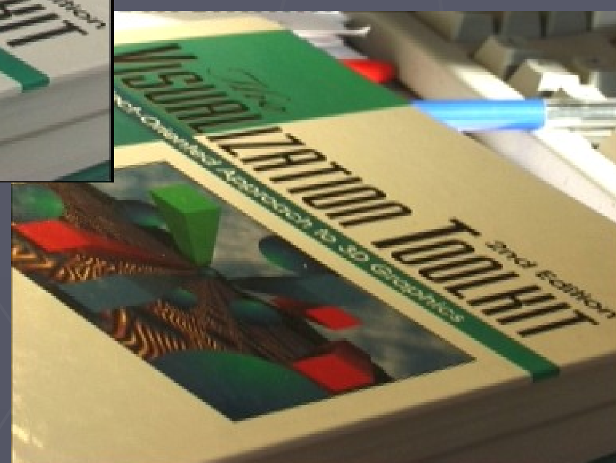
- Translation
- Euclidean (translation + rotation)
- Similarity (transl. + rotation + scale)
- Affine transformations
- Projective transformations

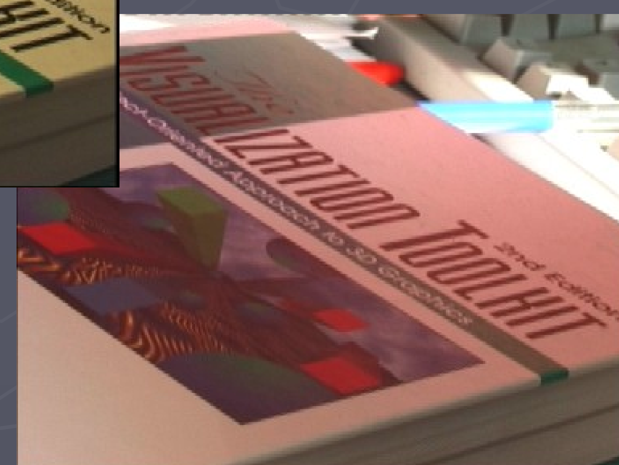
Only holds for planar patches!



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

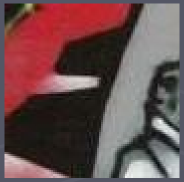
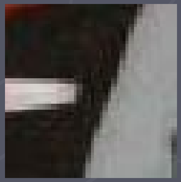
How to cope with transformations?

- Exhaustive search
- Invariance
- Robustness

Exhaustive search

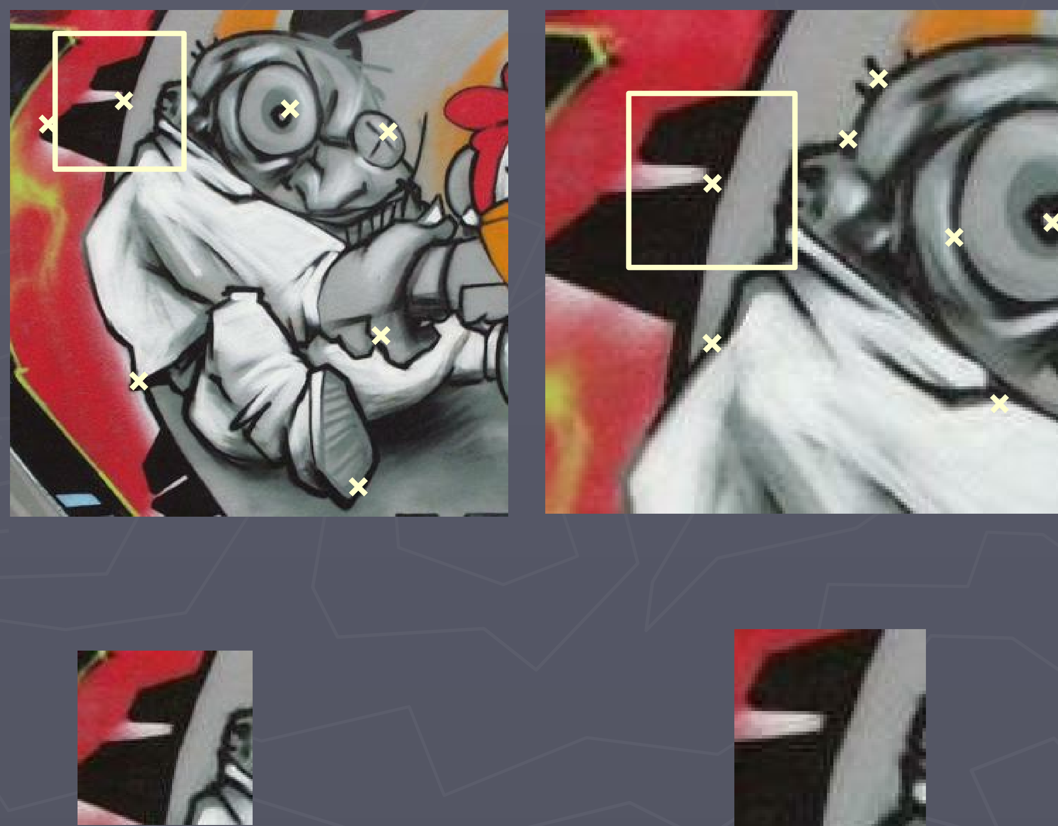
Multi-scale approach



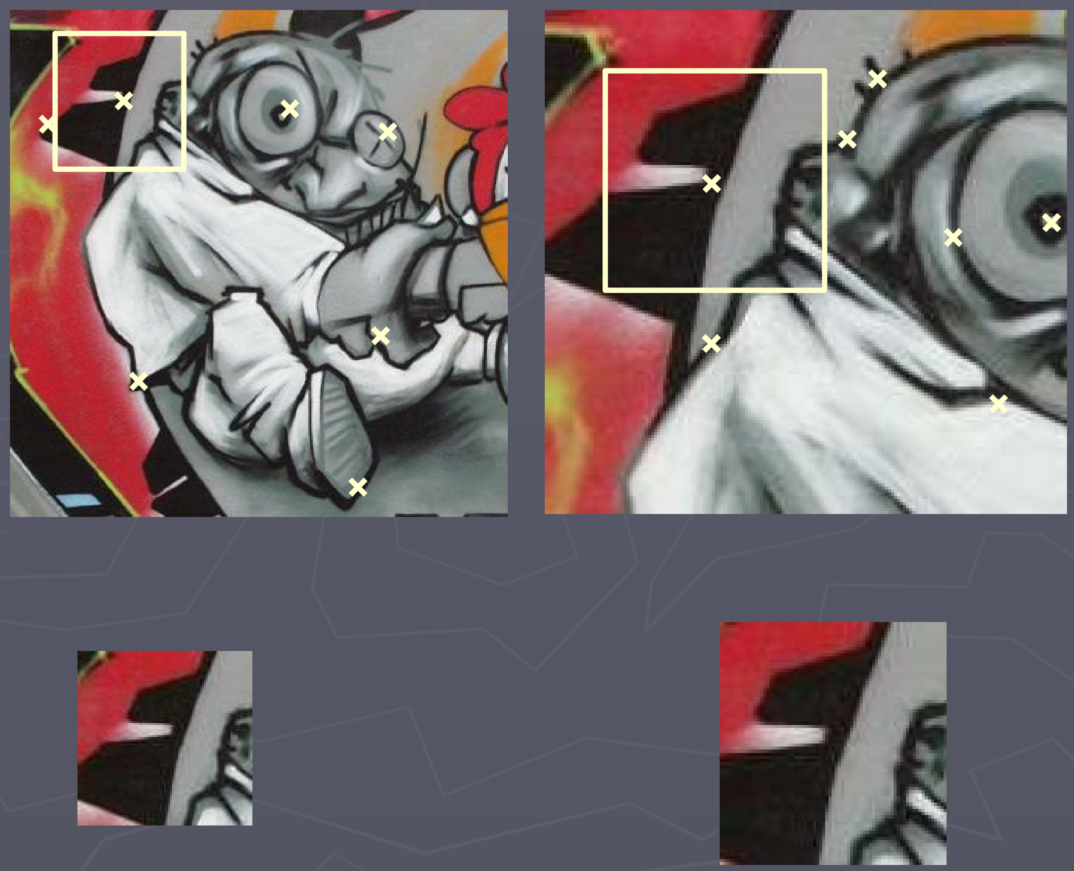
Exhaustive search

Multi-scale approach



Exhaustive search

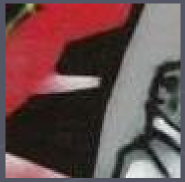
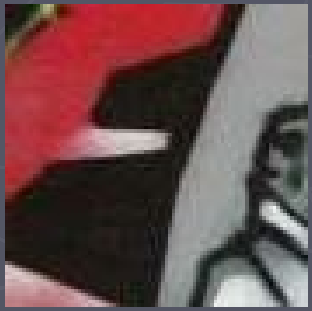
Multi-scale approach



Exhaustive search

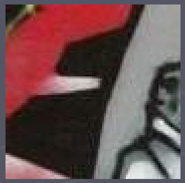
Multi-scale approach



Invariance

Extract patch from each image individually



Invariance

Integration, e.g.
- moment invariants, …

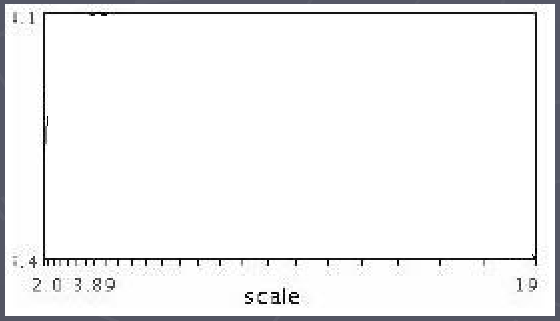
Heuristics, e.g.
- Difference of intensity values for photom. offset
- Ratio of intensity values for photom. scalefactor

Selection and normalization, e.g.
- Automatic scale selection (Lindeberg et al., 1996)
- Orientation assignment
- Affine normalization (‘deskewing’)

…

Automatic scale selection

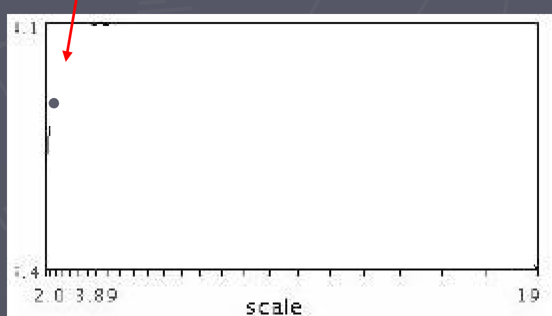
Lindeberg et al., 1996



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

Automatic scale selection

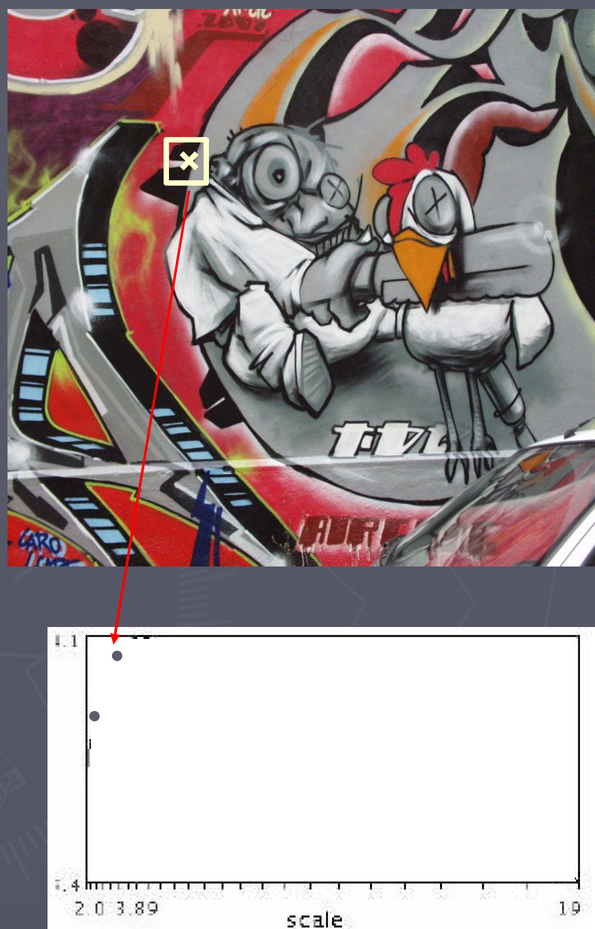
Function responses for increasing scale
Scale trace (signature)



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

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)



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

Automatic scale selection

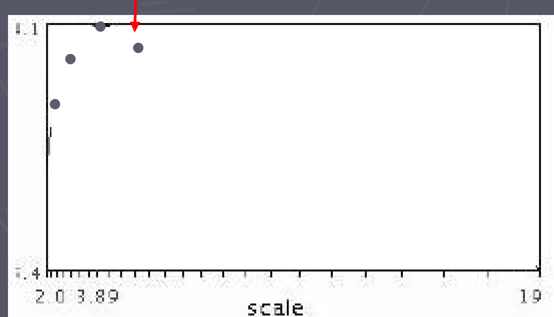
Function responses for increasing scale

Scale trace (signature)

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

Automatic scale selection

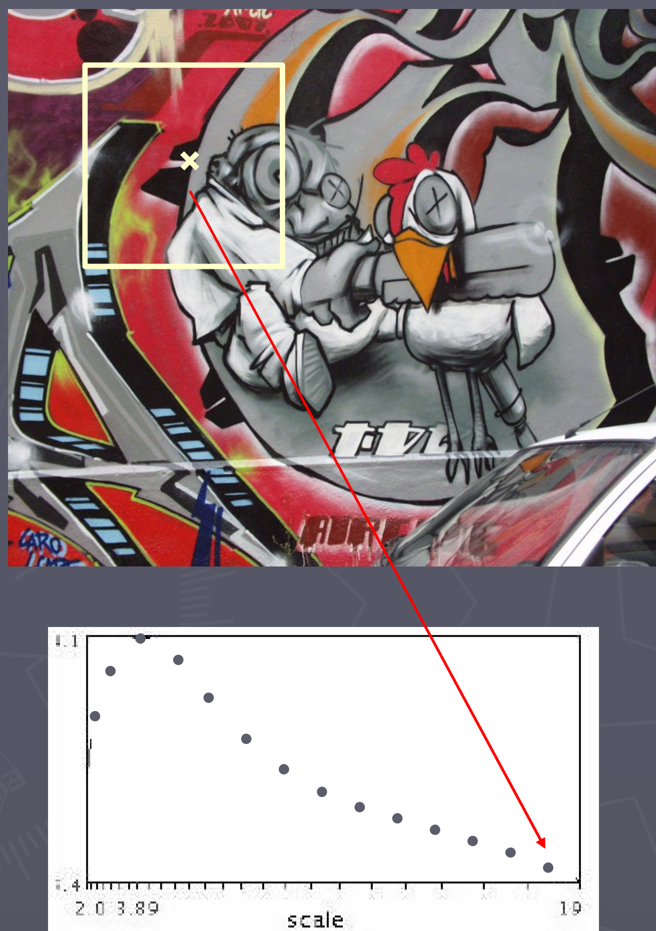
Function responses for increasing scale
Scale trace (signature)



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

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)

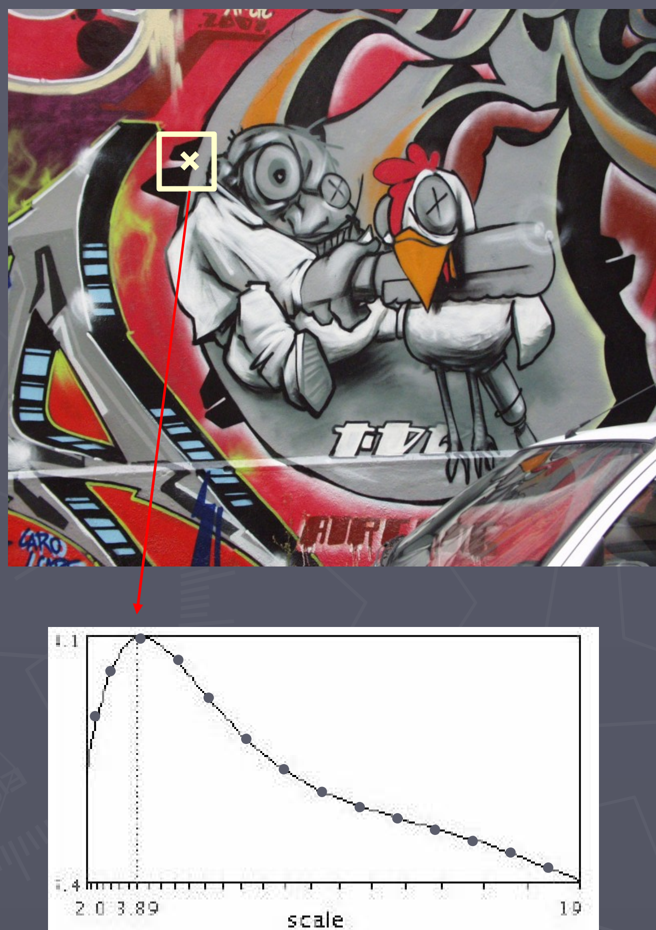


$$f(I_{i_1 \ldots i_m}(x, \sigma))$$

Automatic scale selection

Function responses for increasing scale

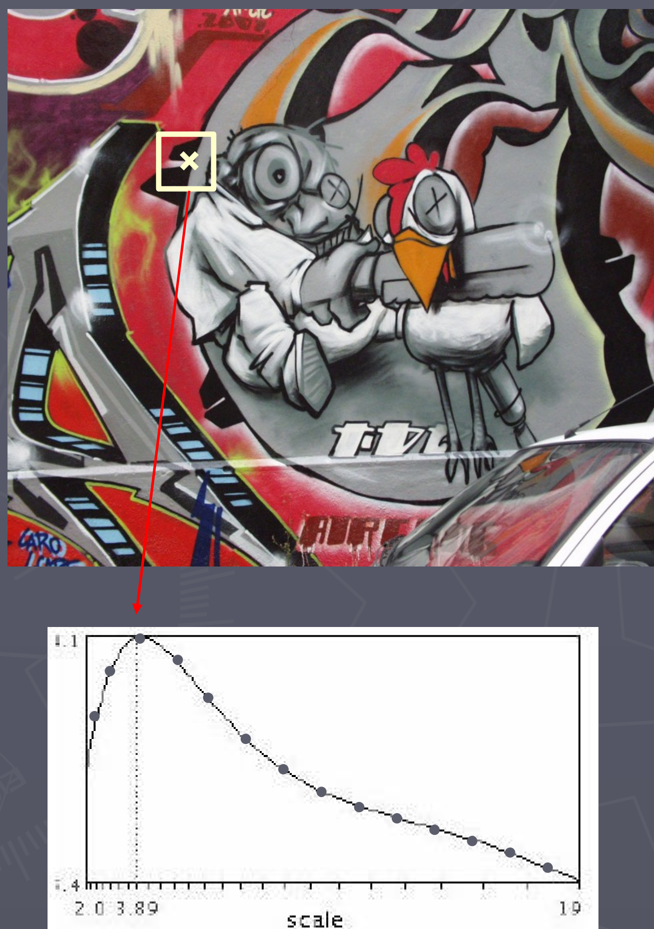
Scale trace (signature)



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

Automatic scale selection

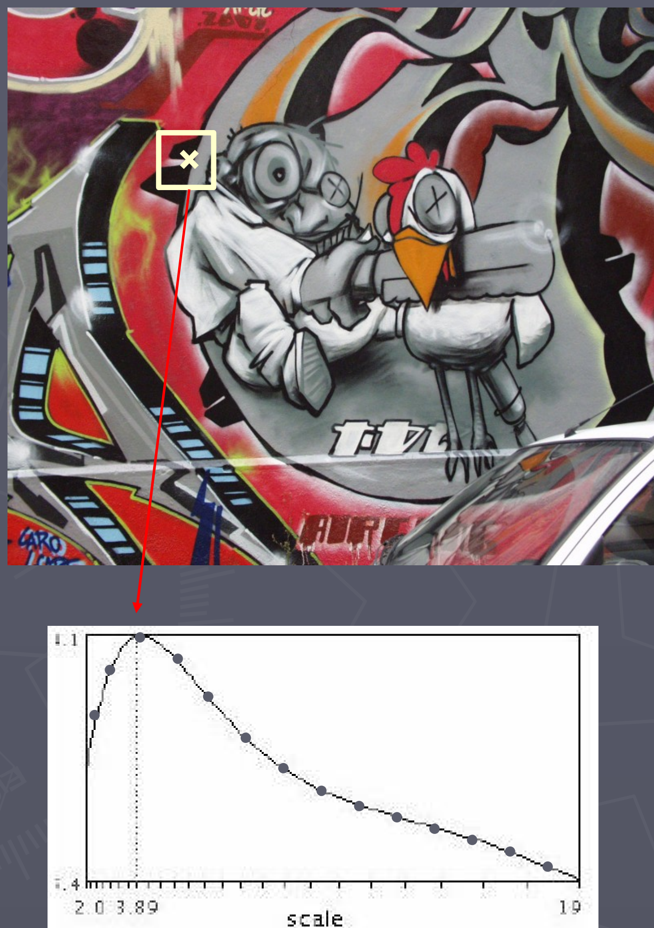
Function responses for increasing scale
Scale trace (signature)



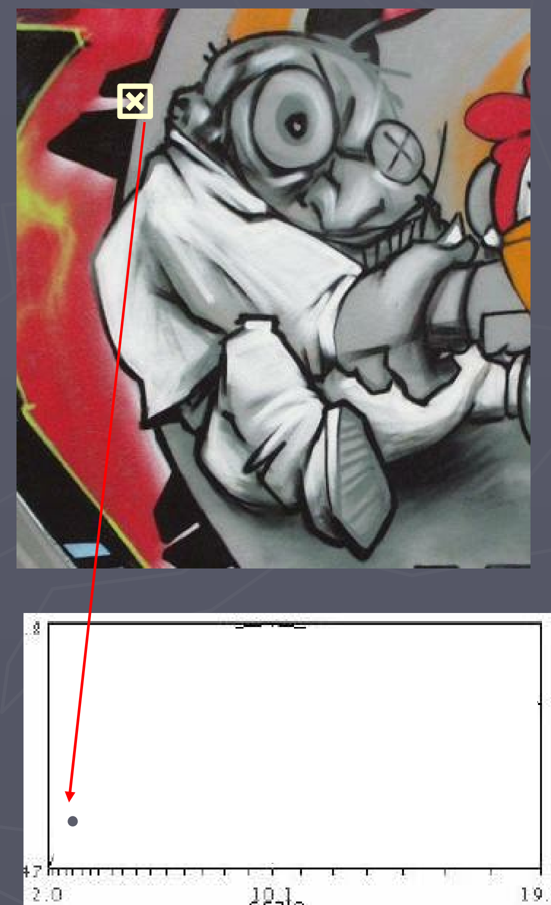
$f(I_{i_1 \ldots i_m}(x,\sigma))$

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

Automatic scale selection



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

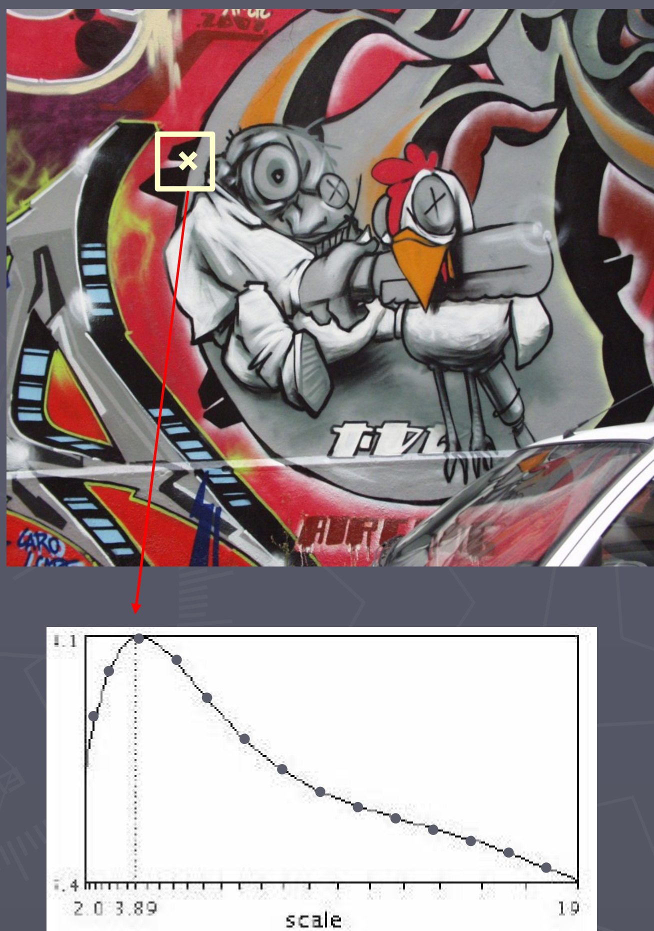


$$f(I_{i_1 \ldots i_m}(x', \sigma))$$

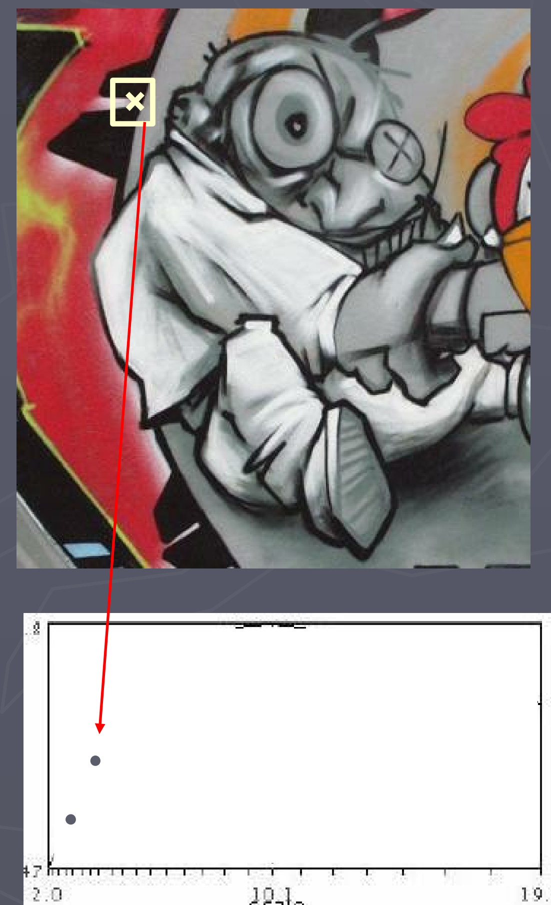
Automatic scale selection

Function responses for increasing scale

Scale trace (signature)



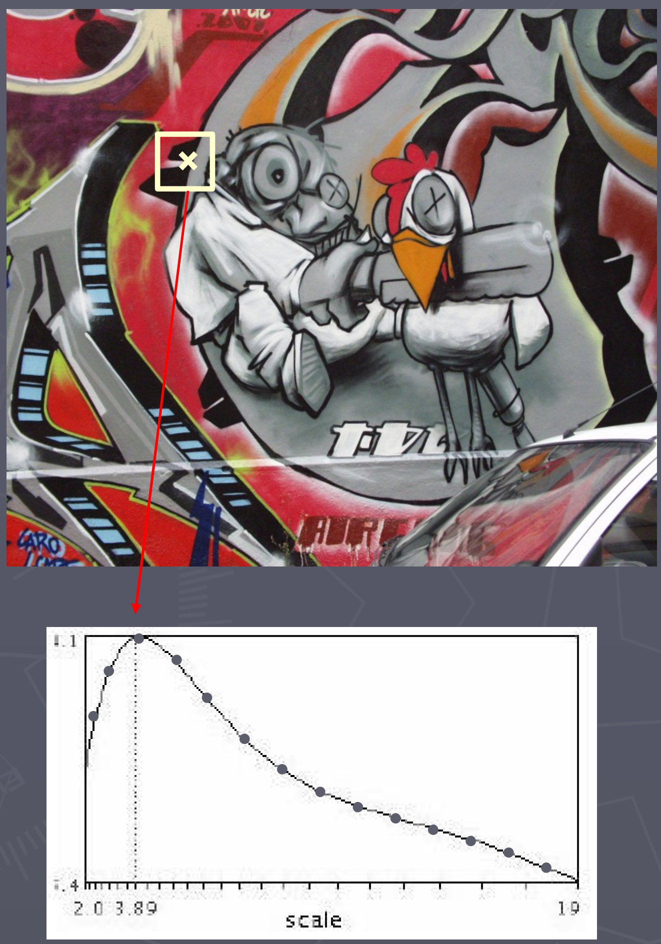
$$f(I_{i_1 \ldots i_m}(x, \sigma))$$



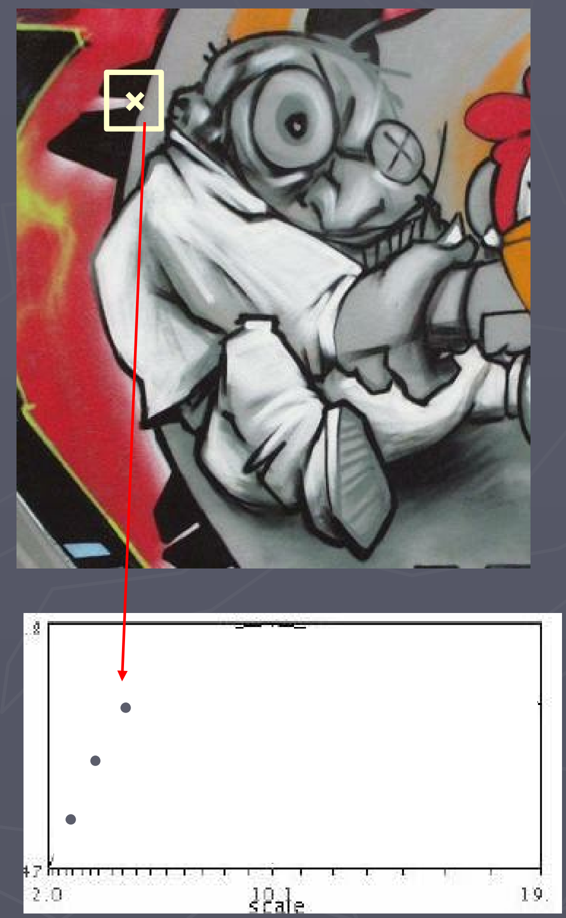
$$f(I_{i_1 \ldots i_m}(x', \sigma))$$

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)



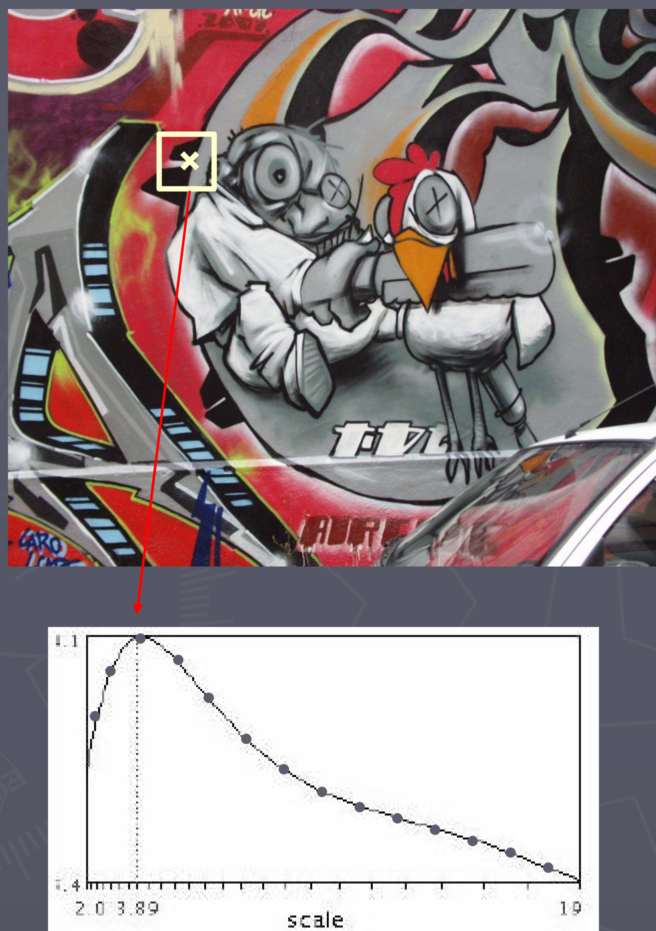
$f(I_{i_1 \ldots i_m}(x,\sigma))$



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

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)

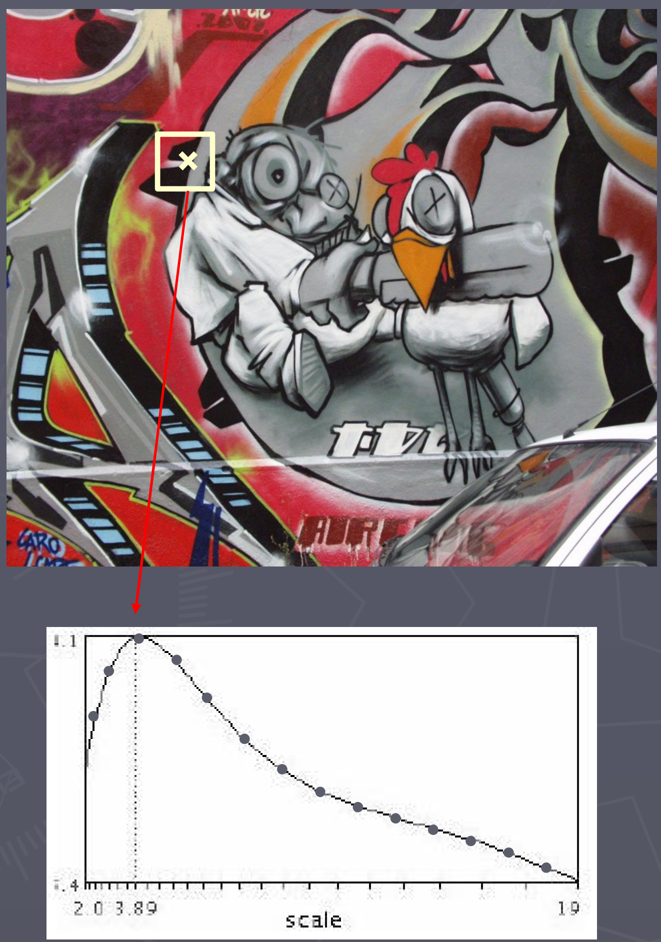


$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

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

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)

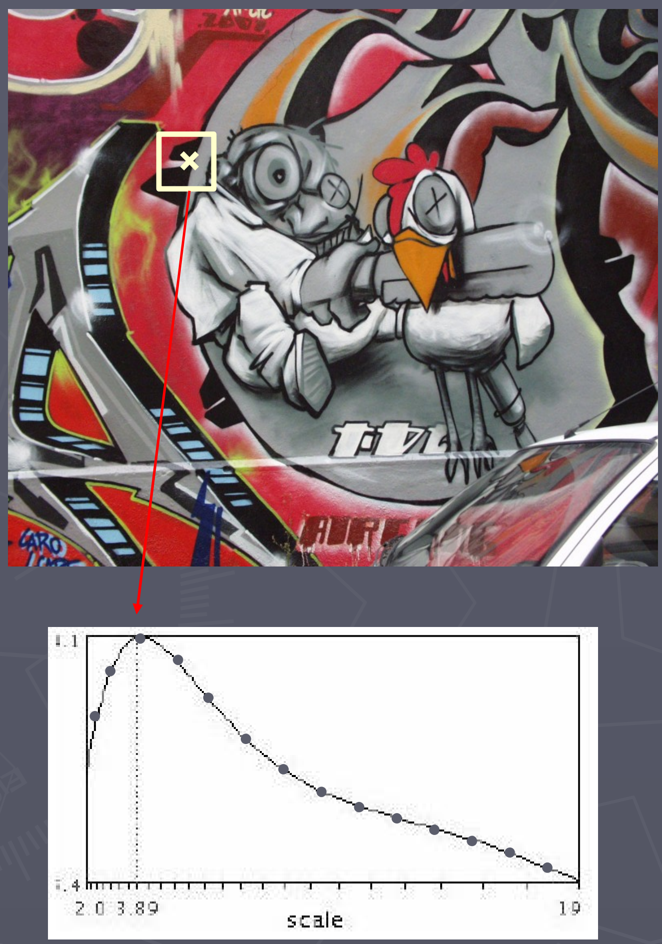


$f(I_{i_1 \ldots i_m}(x, \sigma))$

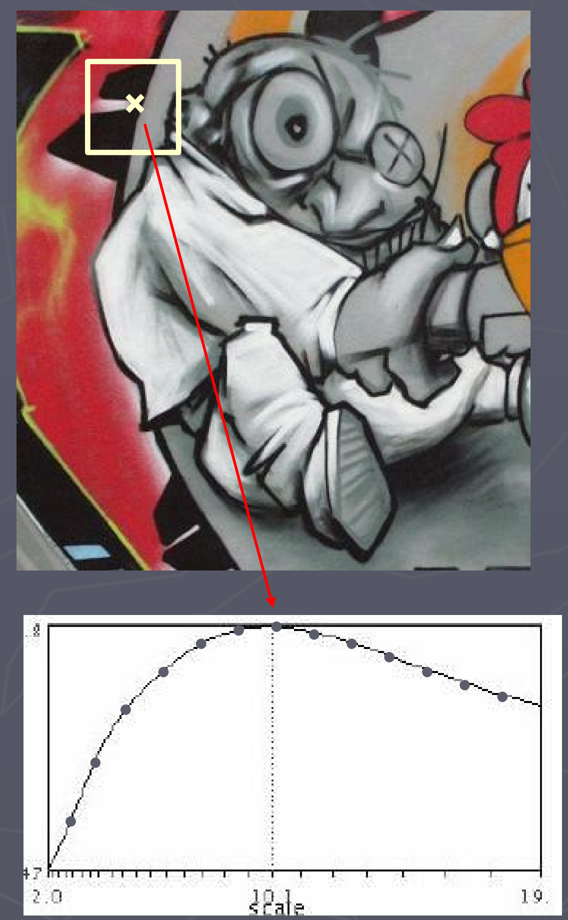
$f(I_{i_1 \ldots i_m}(x', \sigma))$

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)



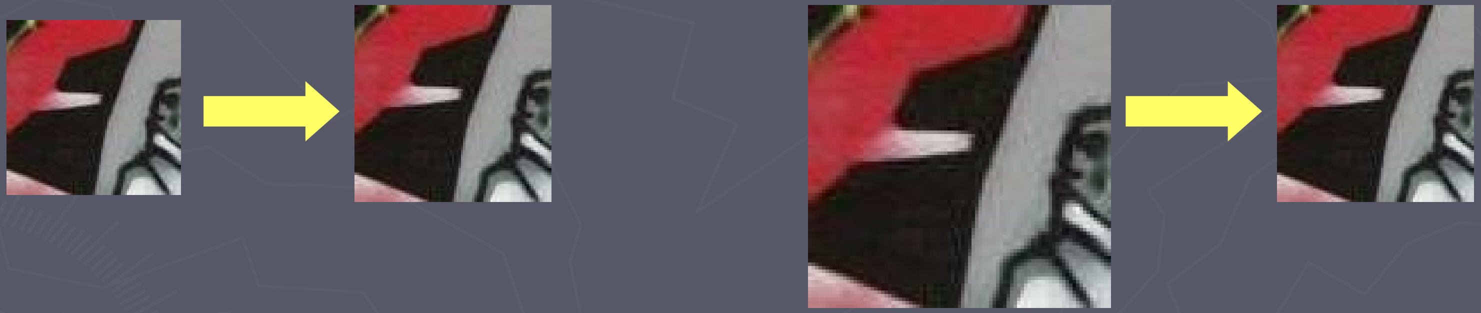
$f(I_{i_1 \ldots i_m}(x,\sigma))$



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

Automatic scale selection

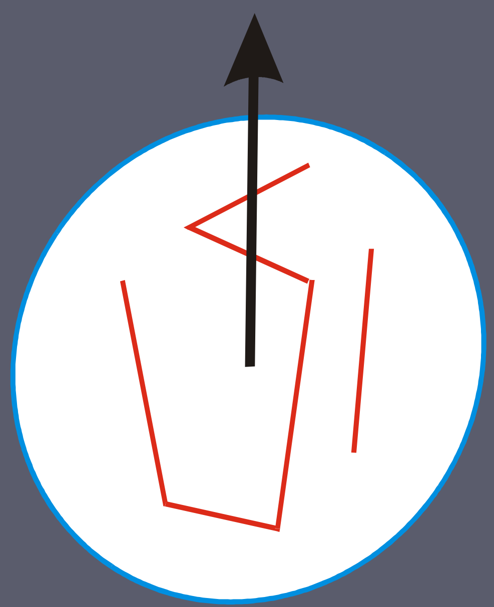
Normalize: rescale to fixed size

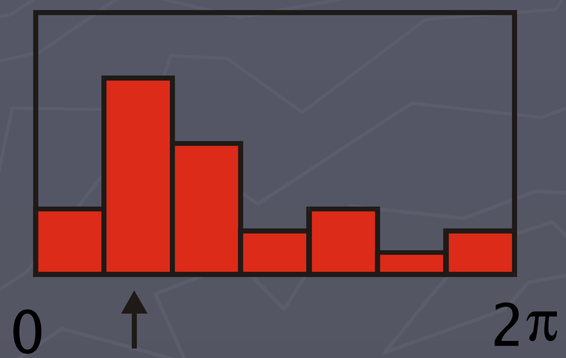


Orientation assignment

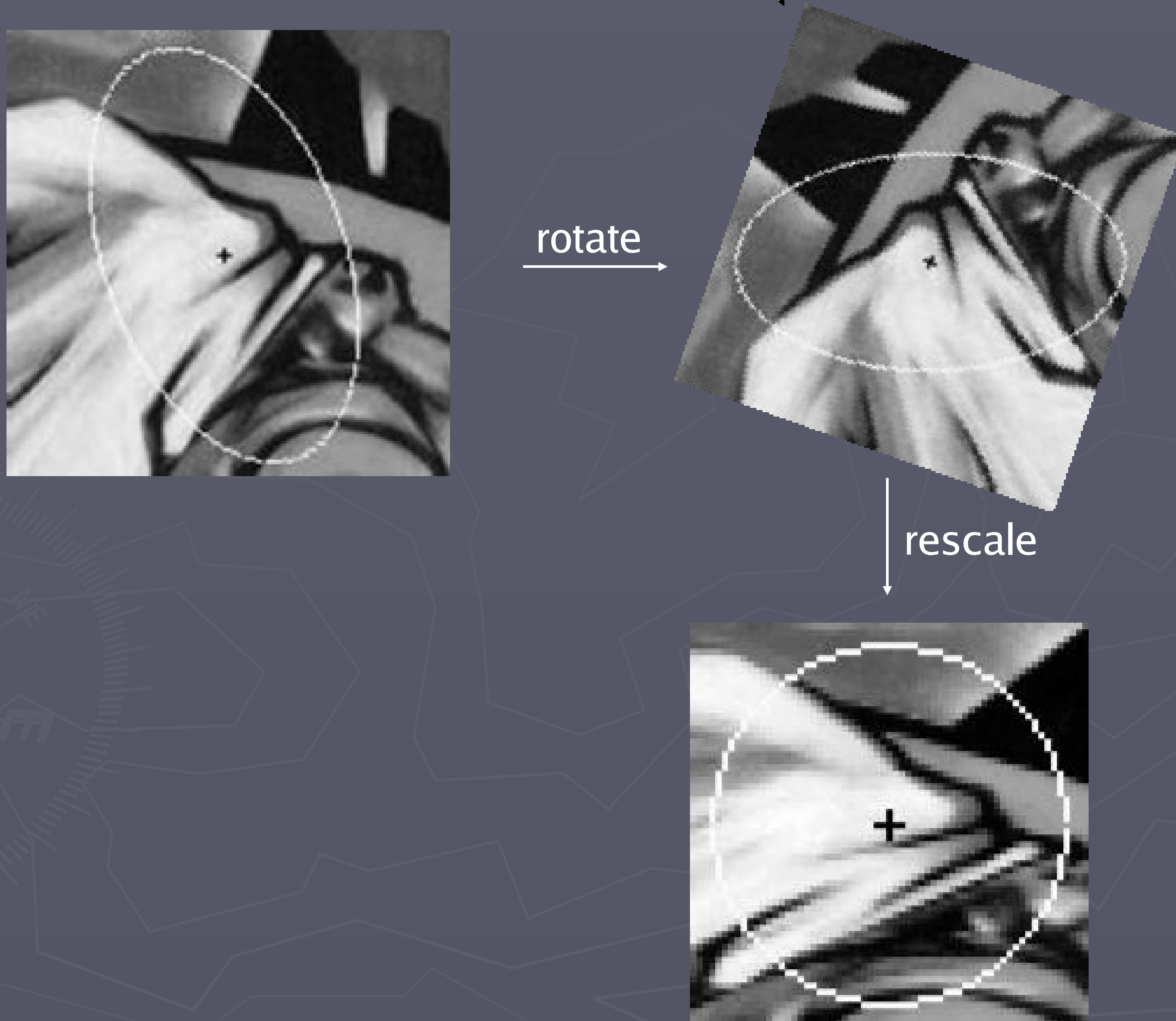
Lowe, SIFT, 1999

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





Affine normalization (‘deskewing’)



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

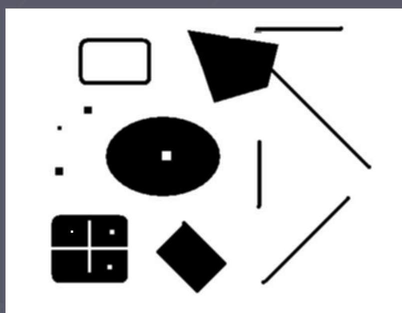
- Hessian & Harris
- Lowe: DoG
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- Others

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

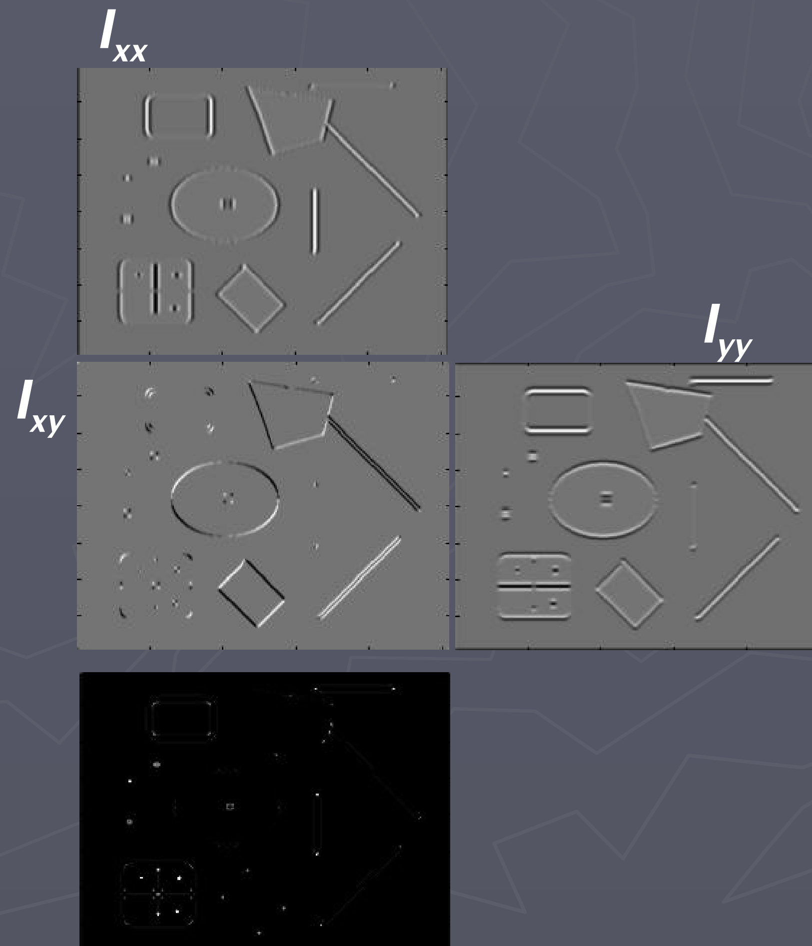
Hessian detector (Beaudet, 1978)

Hessian determinant

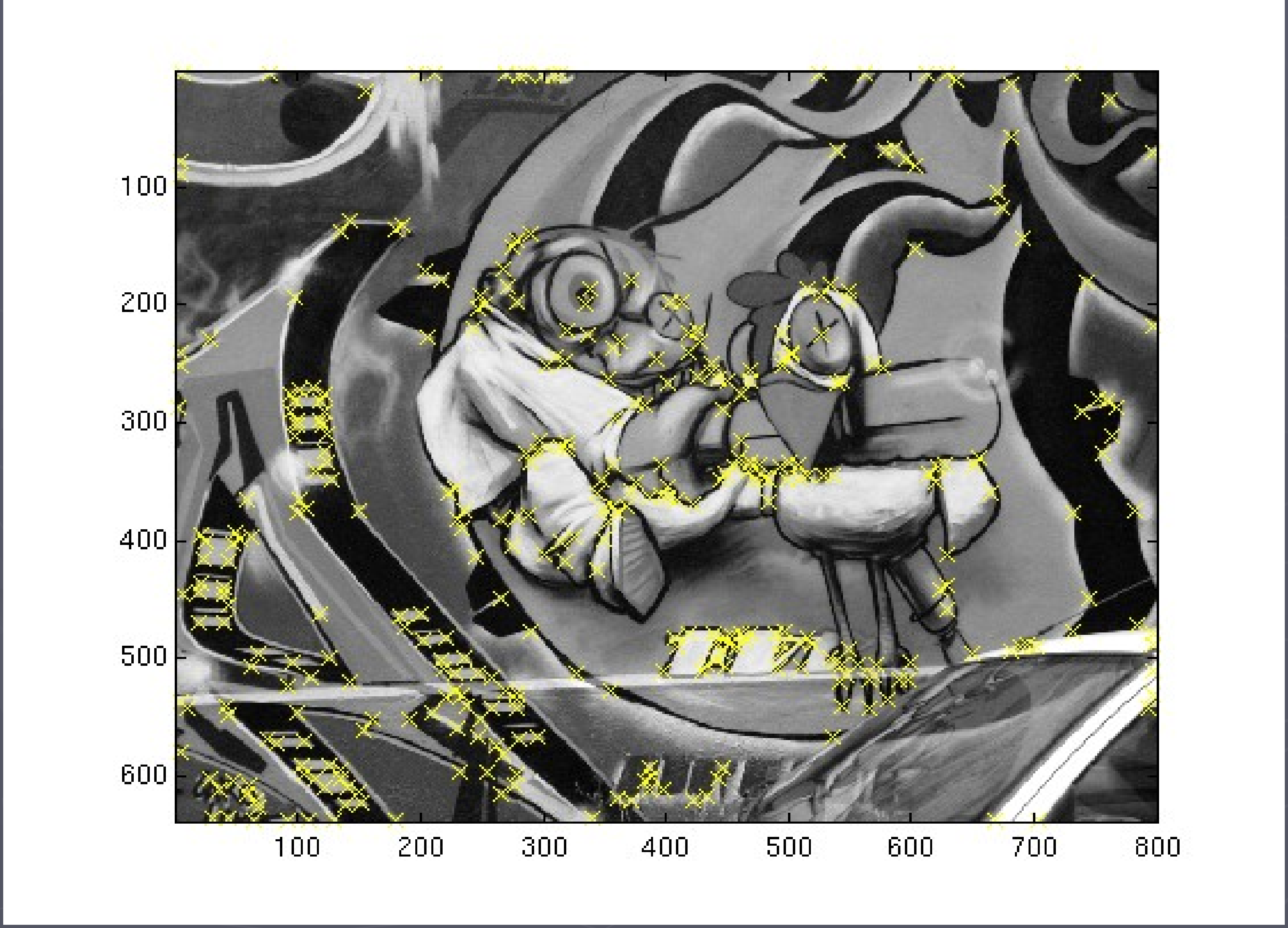


$$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

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



Hessian (Beaudet, 1978)

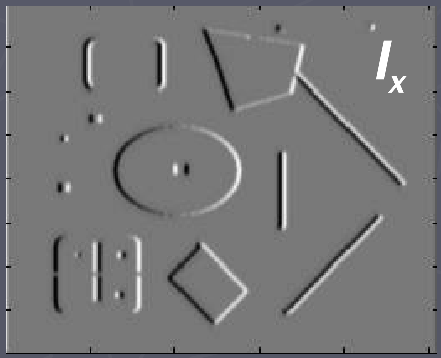


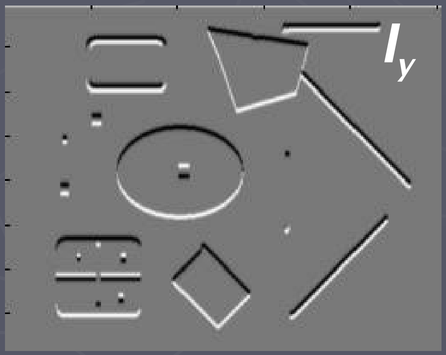
Harris detector (Harris, 1988)

Second moment matrix / autocorrelation matrix

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

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

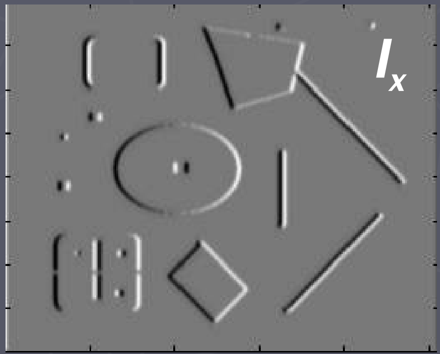


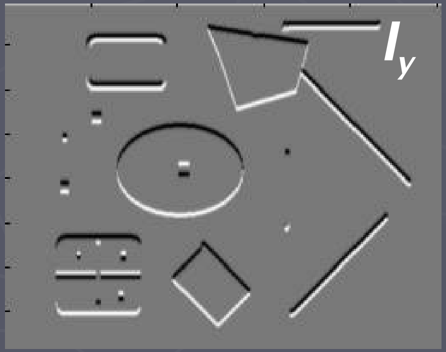


Harris detector (Harris, 1988)

Second moment matrix / autocorrelation matrix

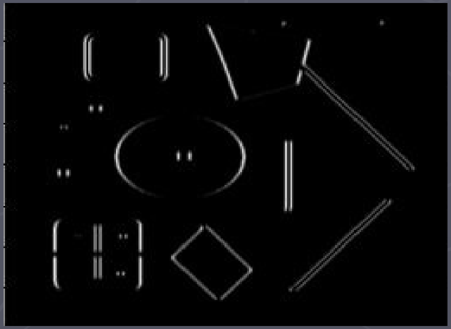
$$\mu(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

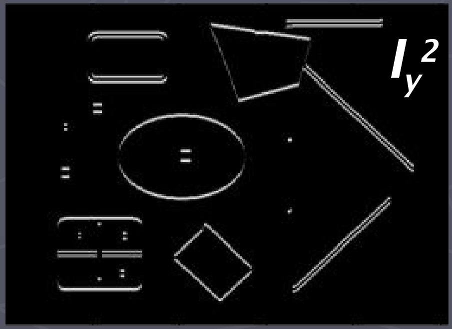


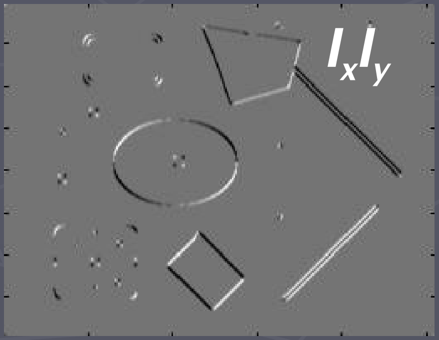


1. **Image derivatives**
   $g_x(\sigma_D)$, $g_y(\sigma_D)$,

2. **Square of derivatives**





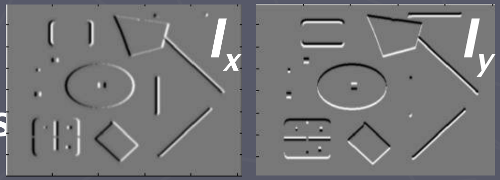


Harris detector (Harris, 1988)

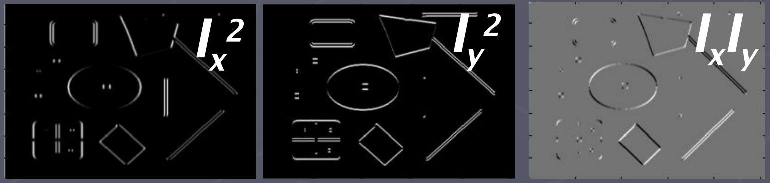
Second moment matrix / autocorrelation matrix

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

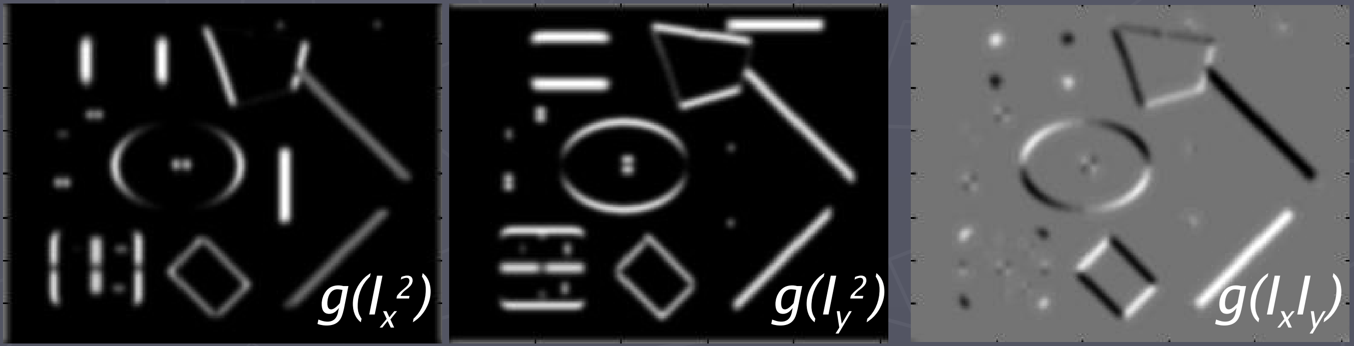
1. Image derivatives



2. Square of derivatives



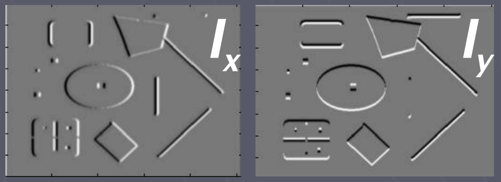
3. Gaussian filter $g(\sigma_I)$



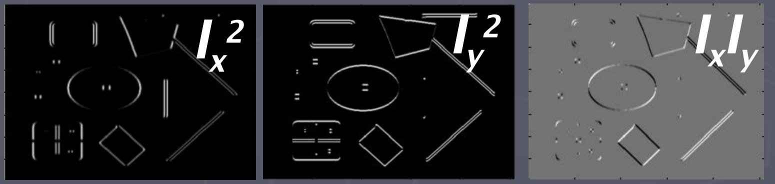
Harris detector (Harris, 1988)

Second moment matrix
autocorrelation matrix

1. Image derivatives



2. Square of derivatives



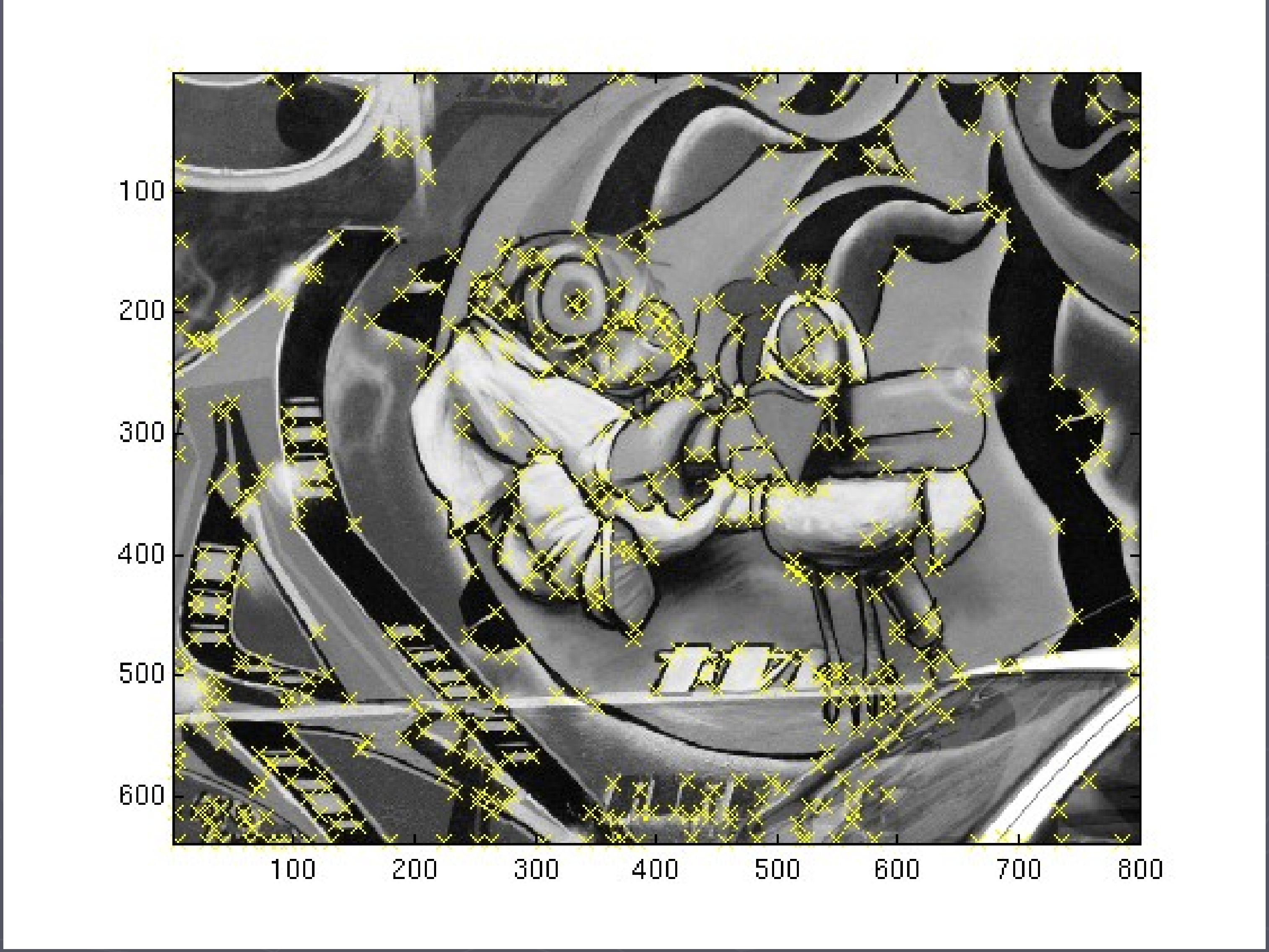
3. Gaussian filter $g(\sigma_I)$

4. Cornerness function – both eigenvalues are strong

$$har = \det[\mu(\sigma_I, \sigma_D)] - \alpha[\text{trace}(\mu(\sigma_I, \sigma_D))] =$$
$$g(I_x^2)g(I_y^2) - [g(I_xI_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

5. Non-maxima suppression

Harris detector (Harris, 1988)

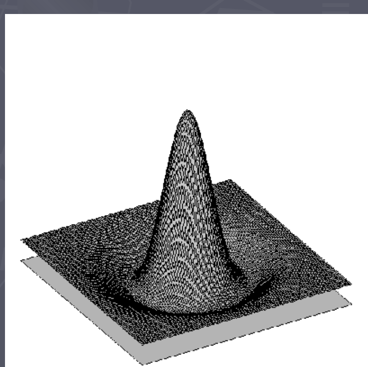


Overview of existing detectors

- 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



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

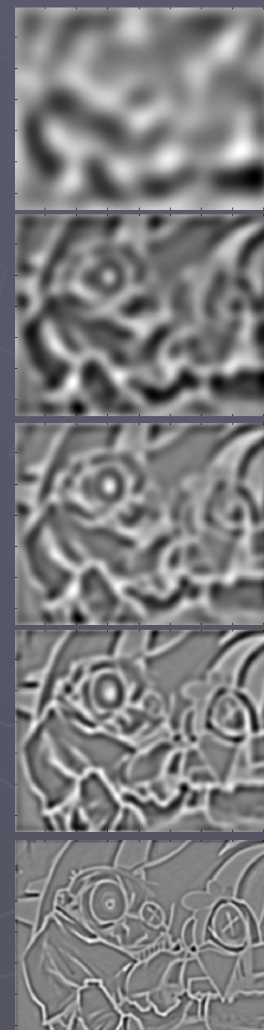
σ^5

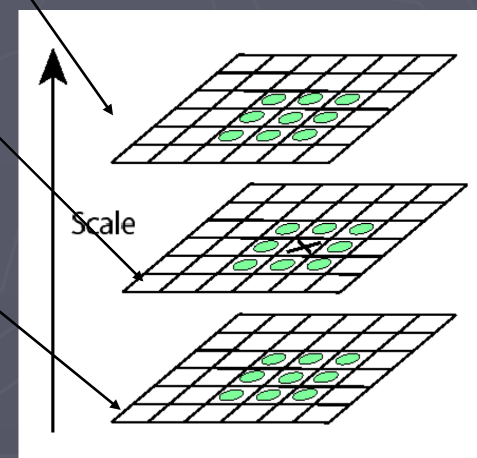
σ^4

σ^3

σ^2

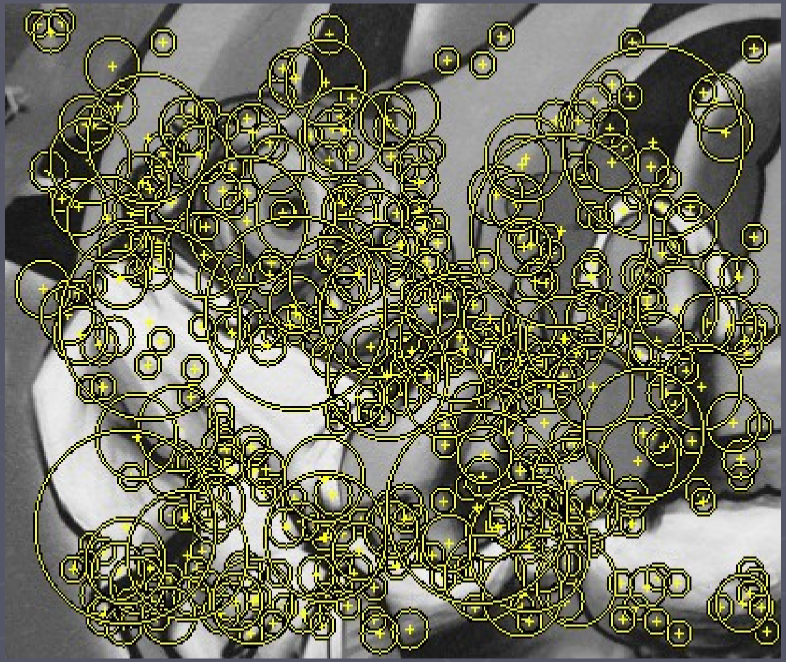
σ





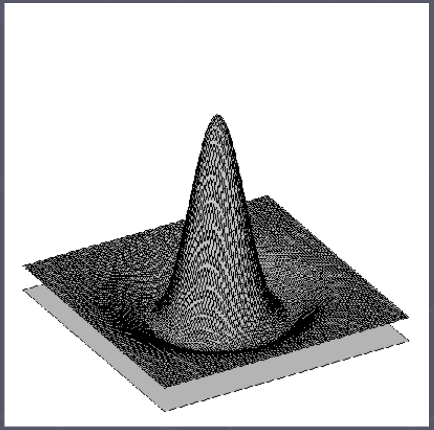
list of (x, y, σ)

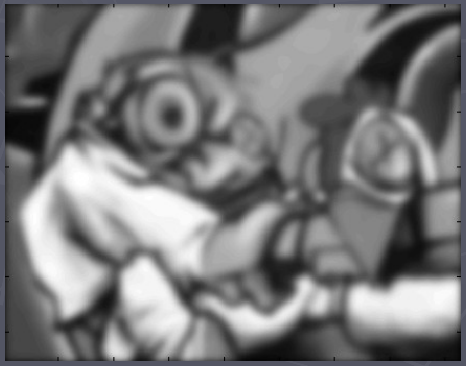
Scale invariant detectors
Laplacean of Gaussian



Lowe's DoG

Difference of Gaussians as approximation of the Laplacian of Gaussian



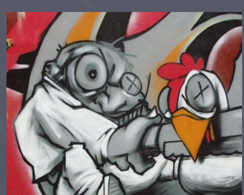


-

=

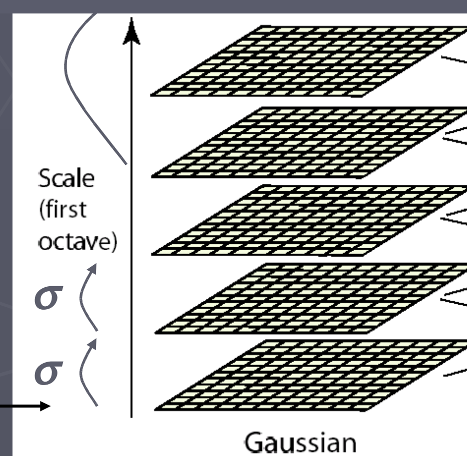
Lowe's DoG

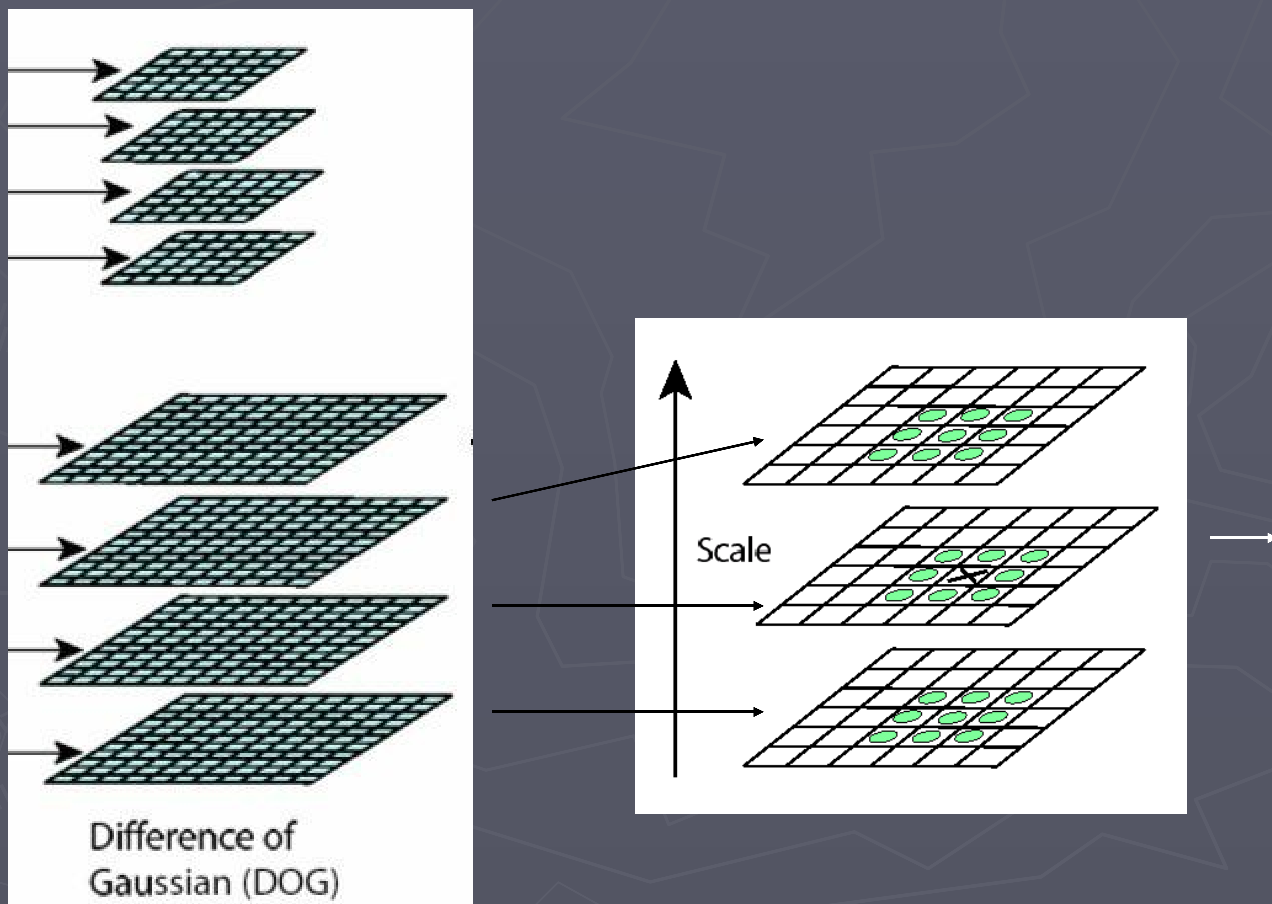
Difference of Gaussians as approximation of the Laplacian of Gaussian



sampling with step
$\sigma^4 = 2$

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





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

Mikolajczyk & Schmid

Harris Laplace

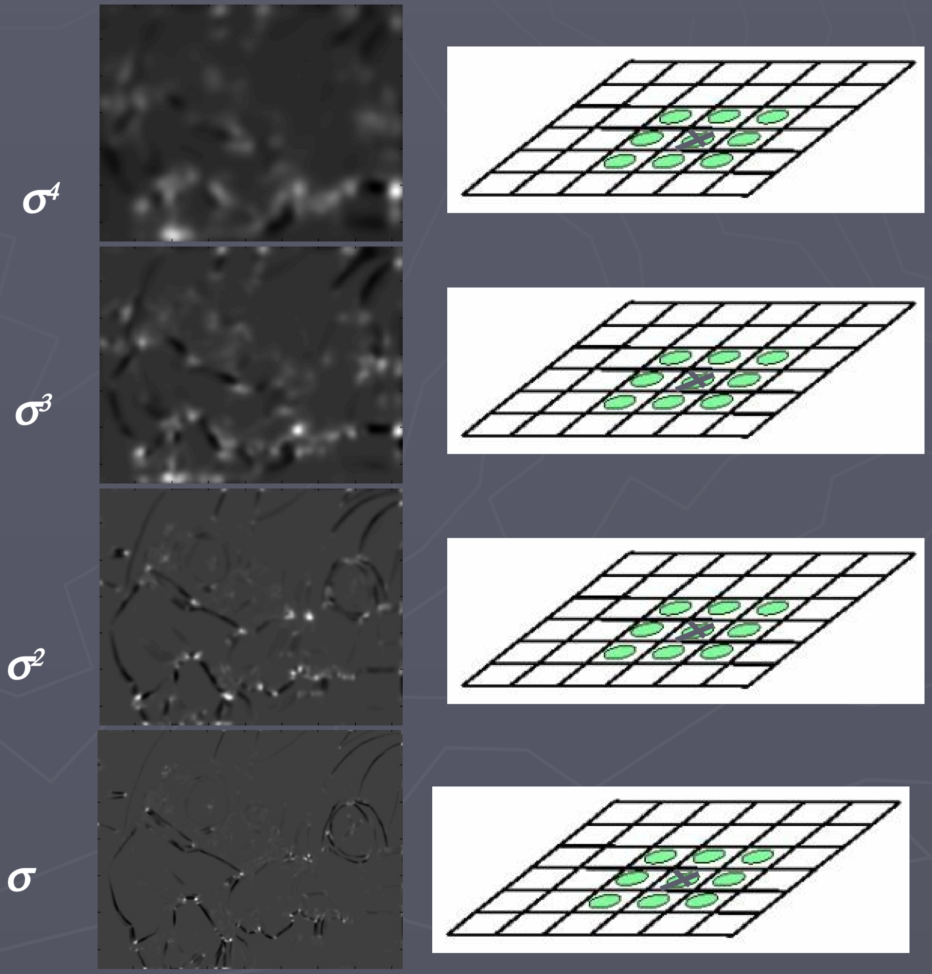
Hessian Laplace

Harris Affine

Hessian Affine

Mikolajczyk: Harris Laplace

1. Initialization: Multiscale Harris corner detection

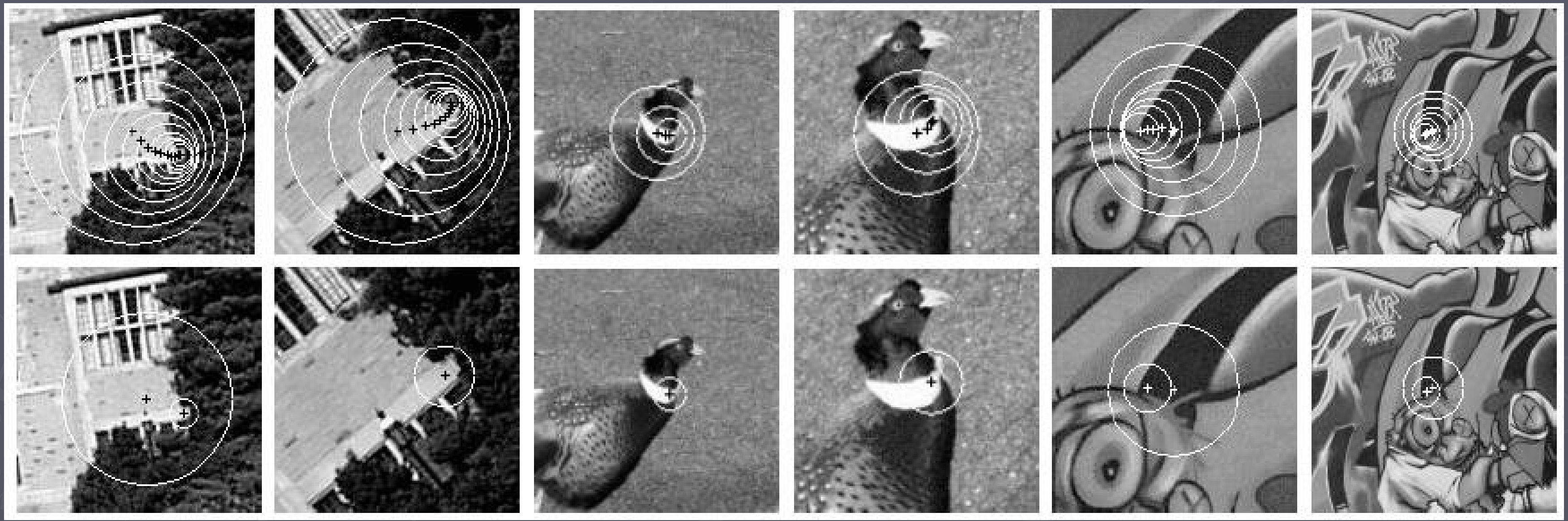


Computing Harris function **Detecting local maxima**

Mikolajczyk: Harris Laplace

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

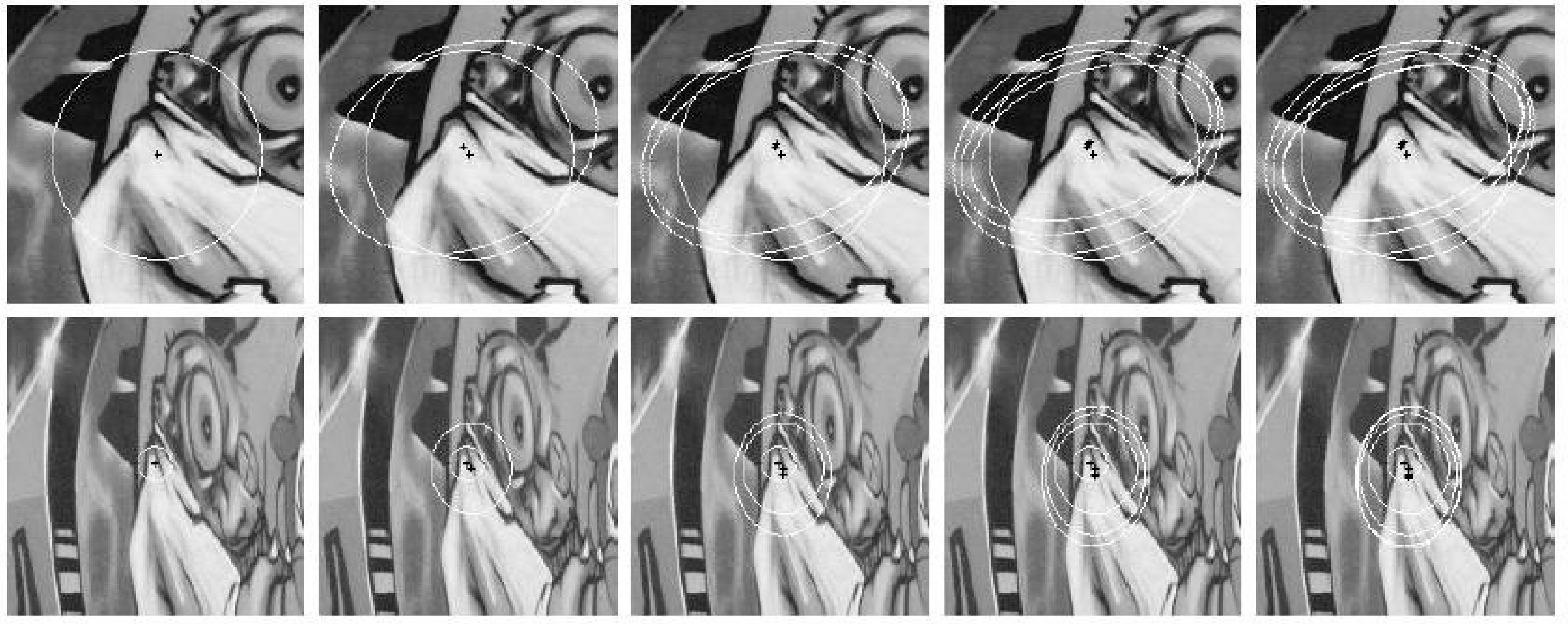
Harris points



Harris-Laplace points

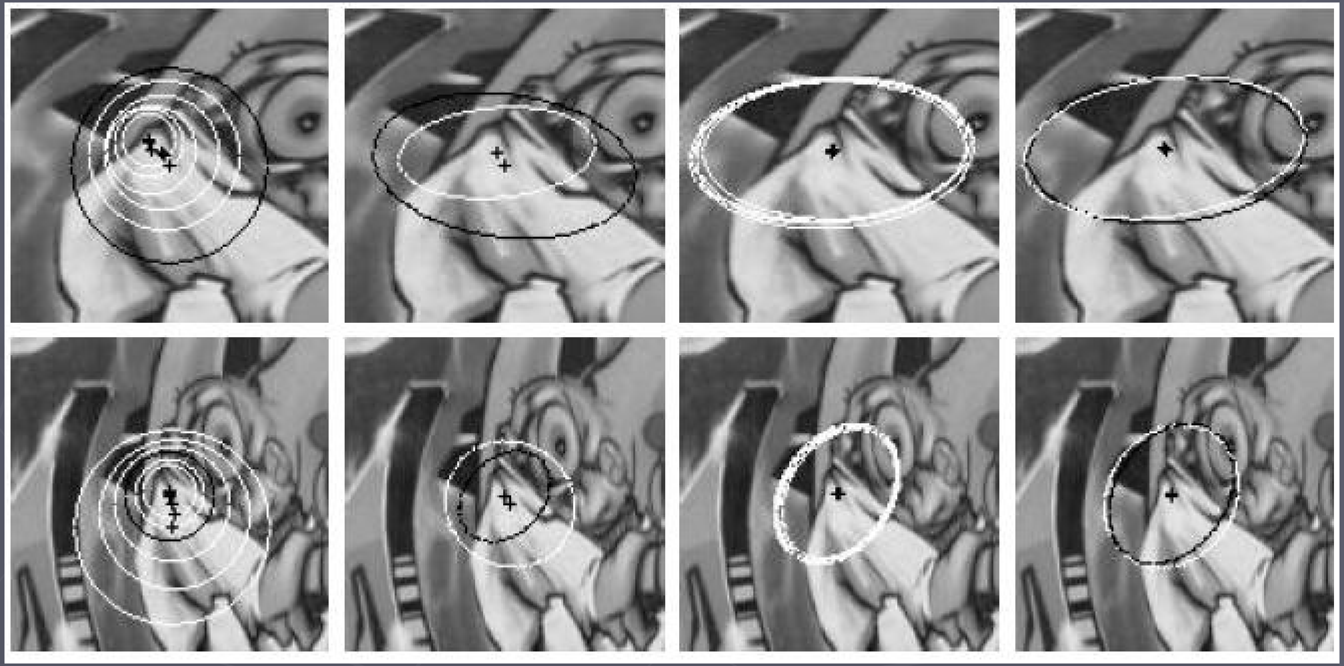
Mikolajczyk: Harris Affine

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

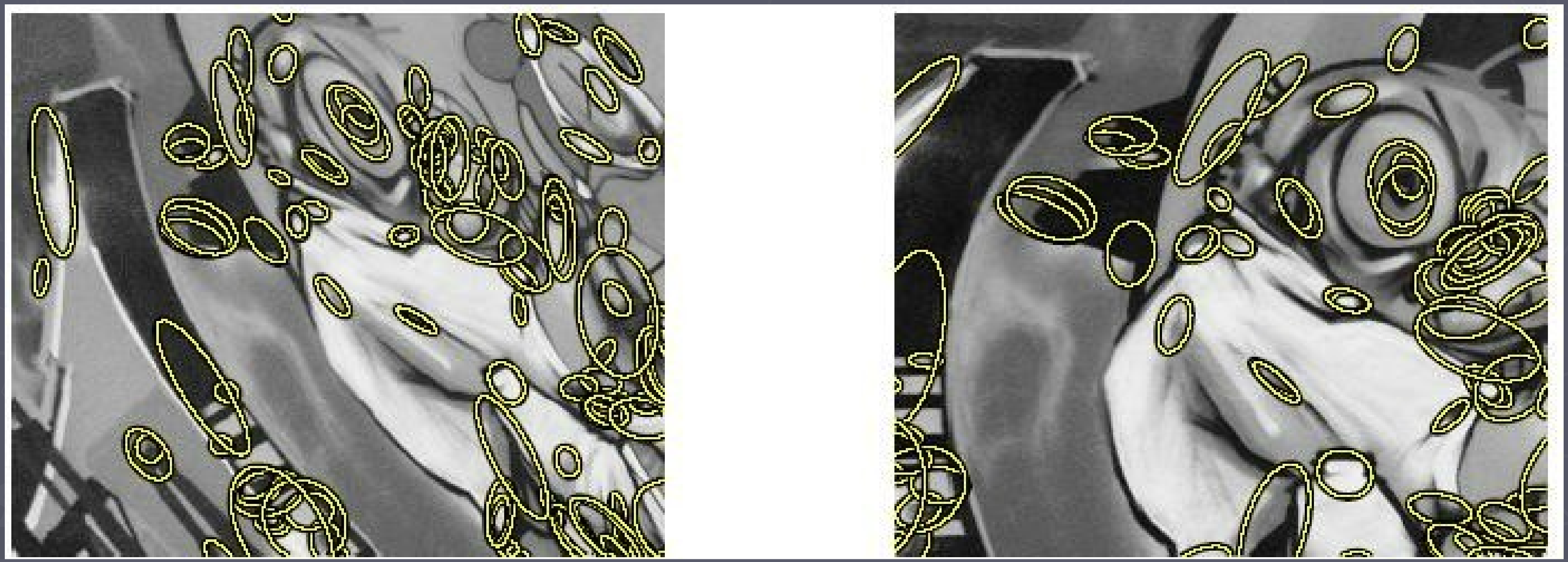


Mikolajczyk: Harris Affine

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



Harris Affine



Hessian Affine

Appreciation

Scale or affine invariant

Detects blob- and corner-like structures

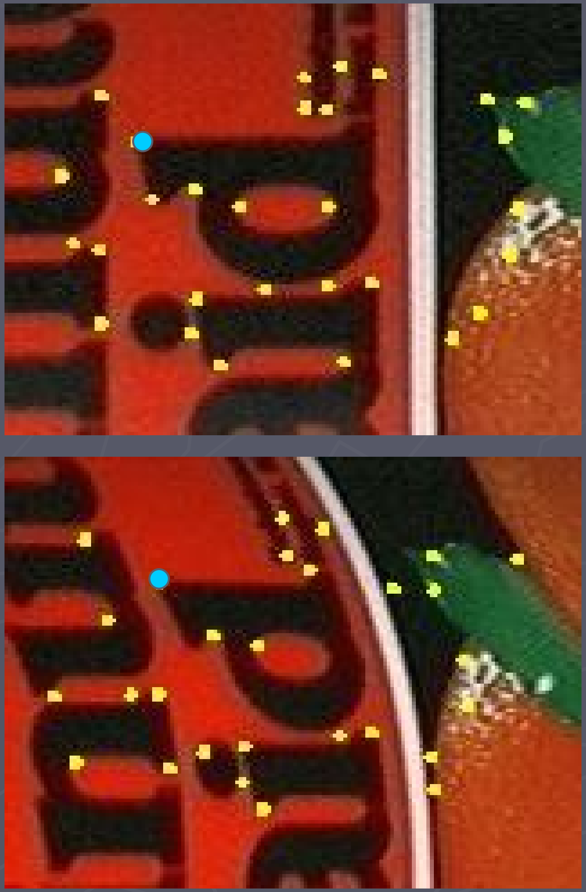
- large number of regions
- well suited for object class recognition
- less accurate than some competitors

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

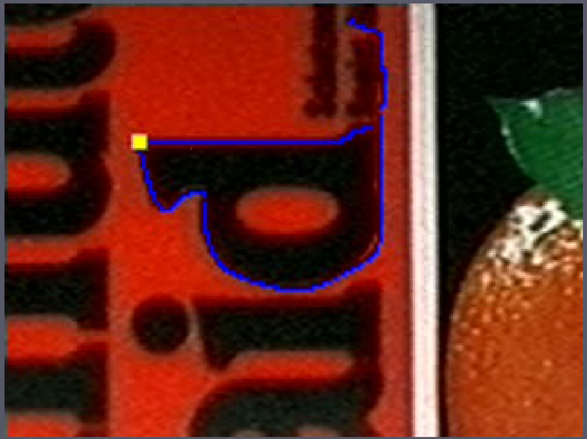
Tuytelaars: edge-based regions

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

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

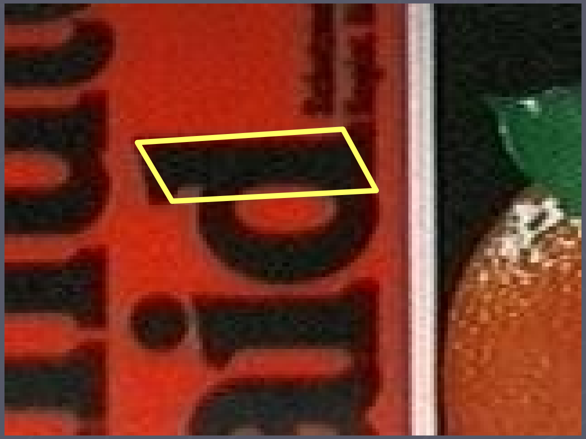
Tuytelaars: edge-based regions

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



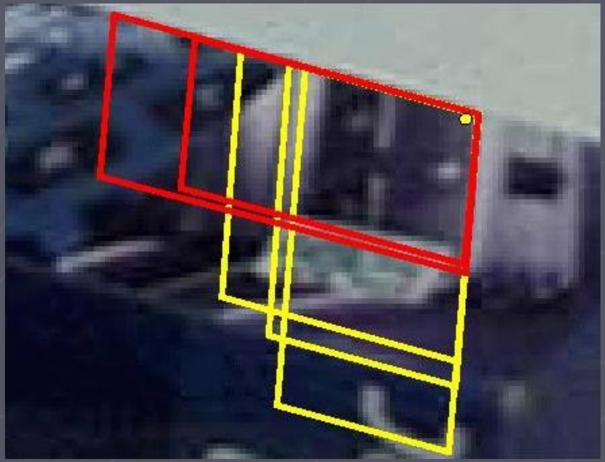
$$f(\Omega) = \frac{|\, p_1 - p_g \quad p_2 - p_g \,|}{|\, p - p_1 \quad p - p_2 \,|} \frac{M_{00}^{1}}{\sqrt{M_{00}^{2} M_{00}^{0} - M_{00}^{1} M_{00}^{1}}}$$

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

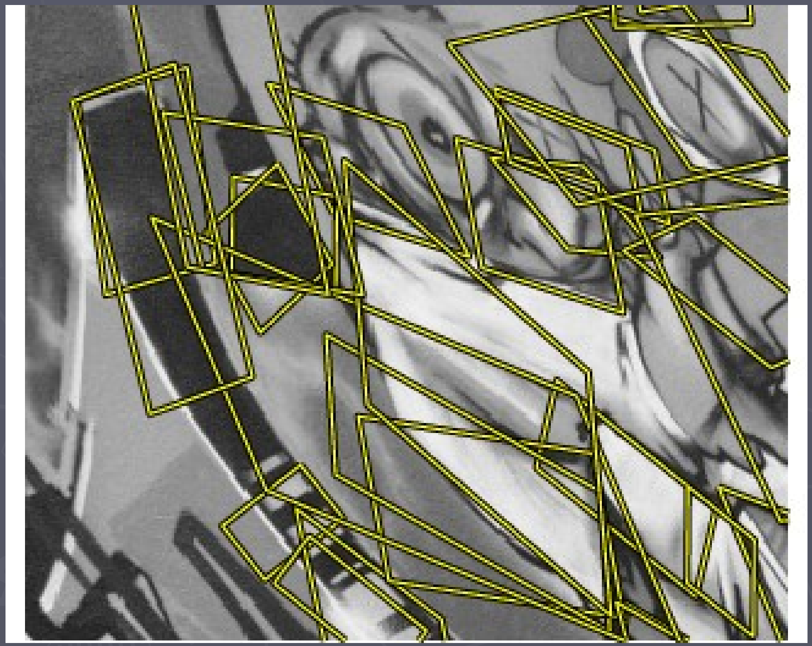
$$M_{pq}^{a} = \int [I(x,y)]^{a} x^{p} y^{q} \, dx\, dy$$

Tuytelaars: edge-based regions

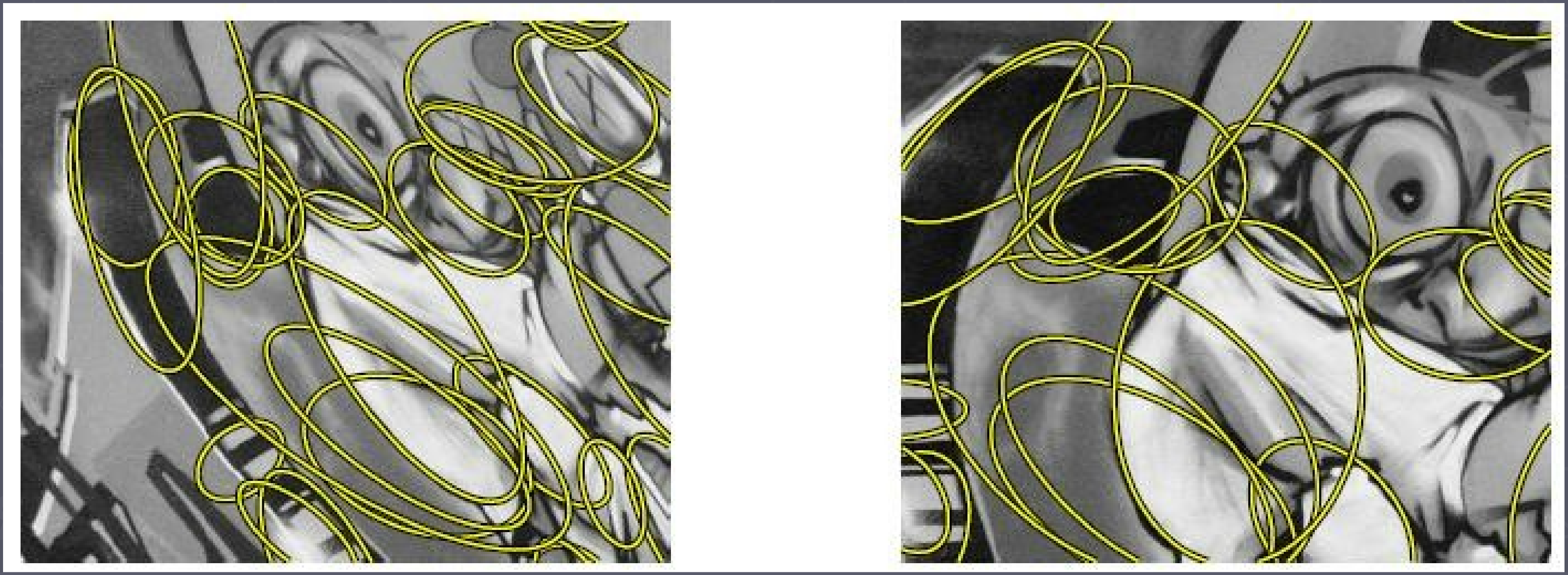
Variant for straight lines…



Edge-based regions



Edge-based regions



Appreciation

- Affine invariant
- Detects corner-like structures

 Works well in structured scenes

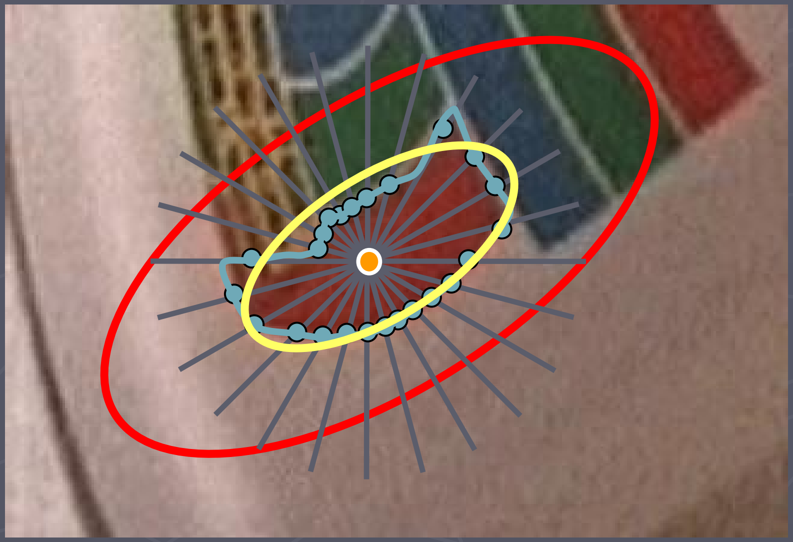
 Doesn’t cross edges/object contours

 Depends on presence of edges

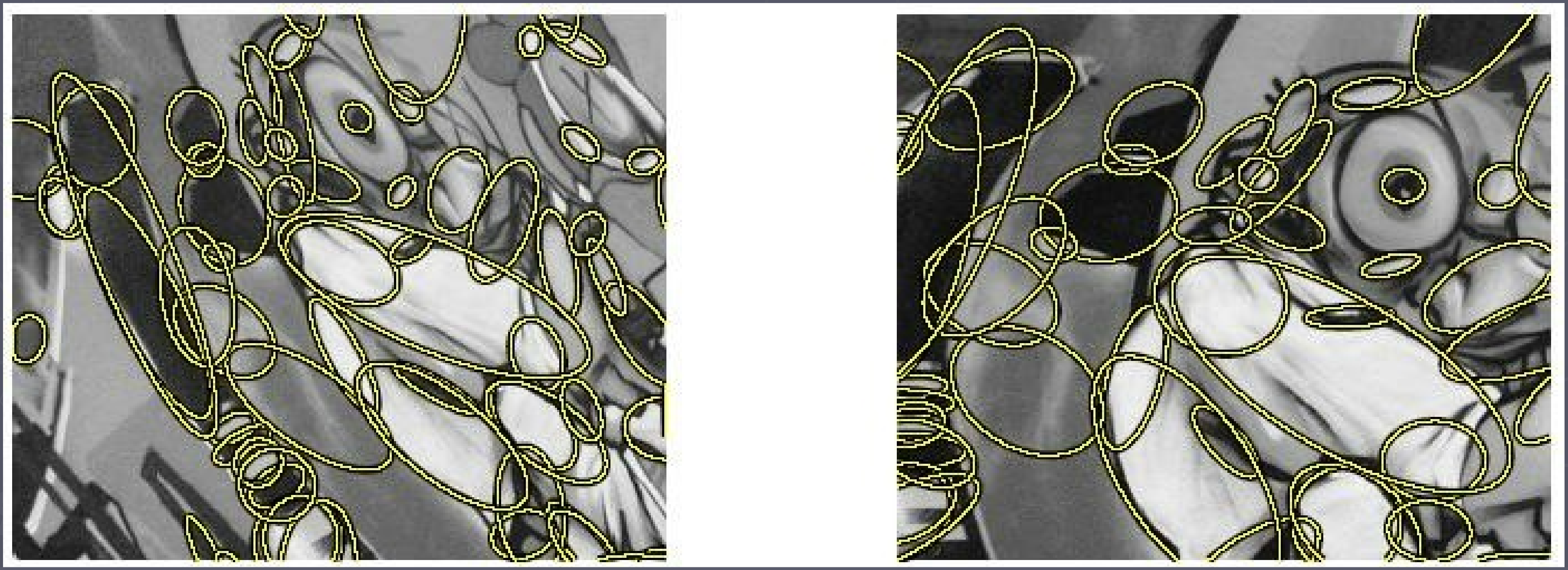
Tuytelaars: intensity-based regions

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

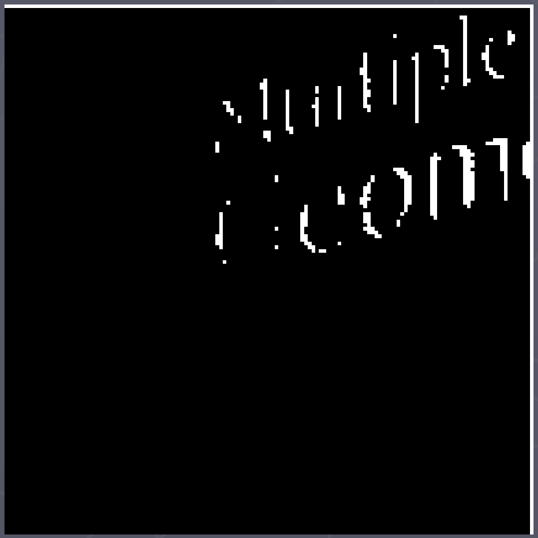
 Especially good on printed material

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

Matas: Maximally Stable Extremal Regions (MSERs)

Based on watershed algorithm

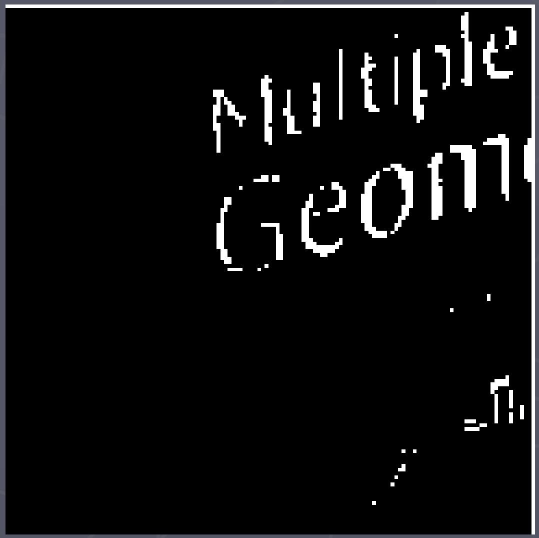


Matas: Maximally Stable Extremal Regions (MSERs)

Based on watershed algorithm

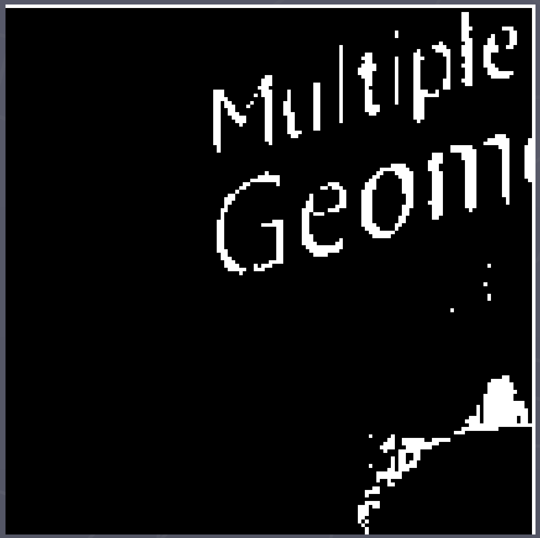
Matas: Maximally Stable Extremal Regions (MSERs)

- Based on watershed algorithm



Matas: Maximally Stable Extremal Regions (MSERs)

Based on watershed algorithm



Matas: Maximally Stable Extremal Regions (MSERs)

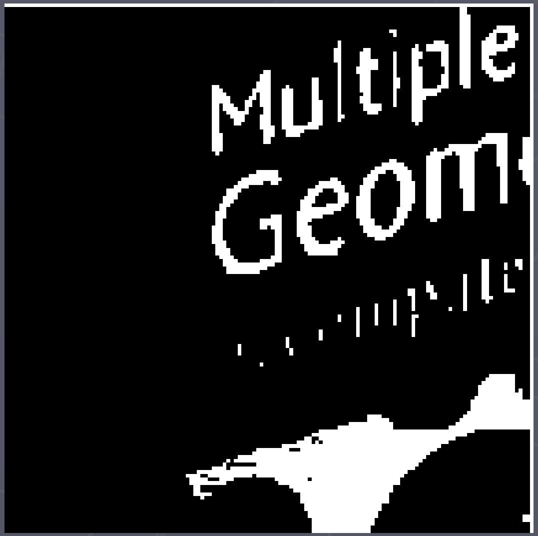
Based on watershed algorithm

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)

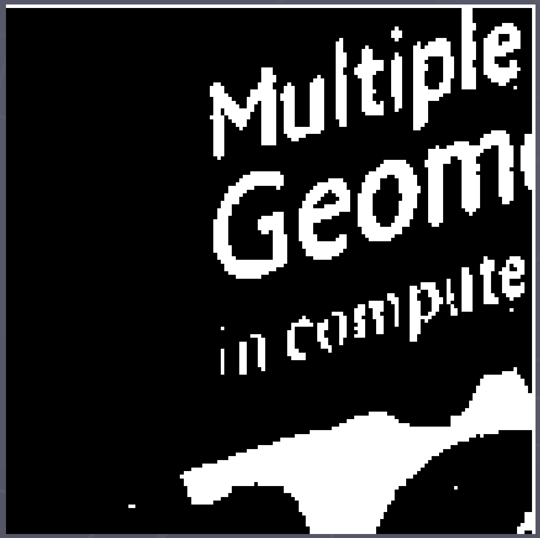
Based on watershed algorithm

Matas: Maximally Stable Extremal Regions (MSERs)

Based on watershed algorithm

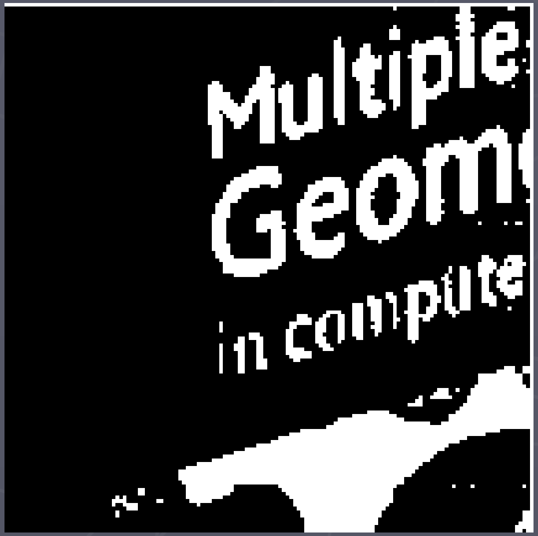
Matas: Maximally Stable Extremal Regions (MSERs)

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

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

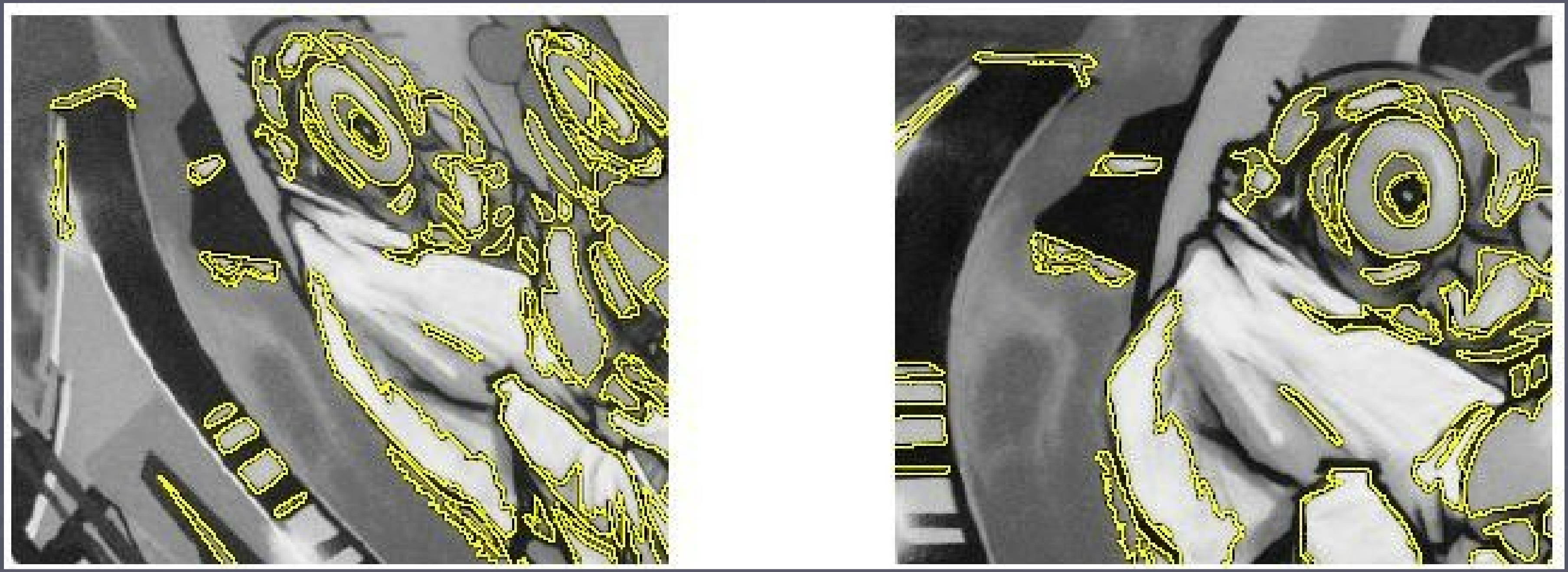
Order regions

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

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)

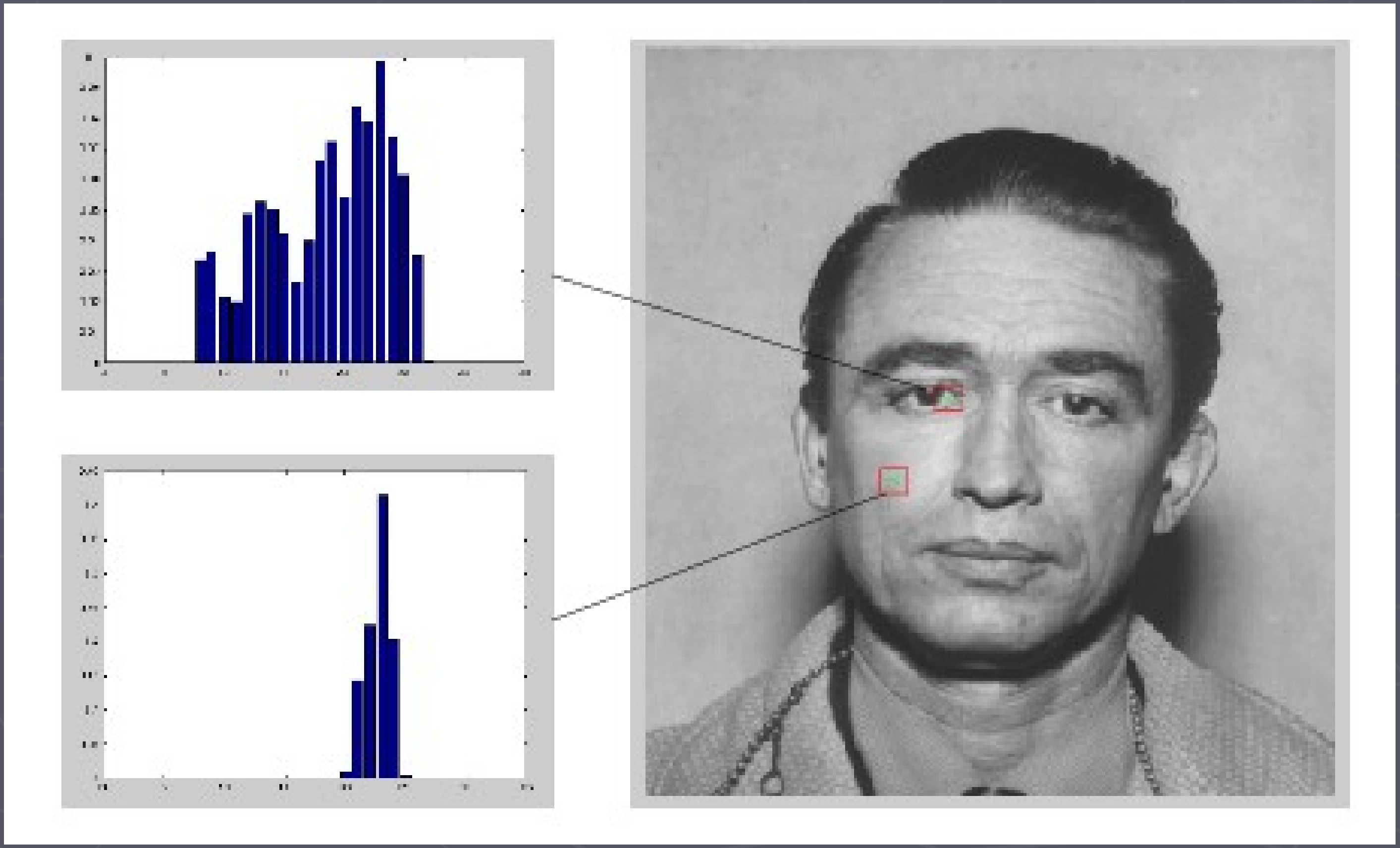
 Sensitive to image blur

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

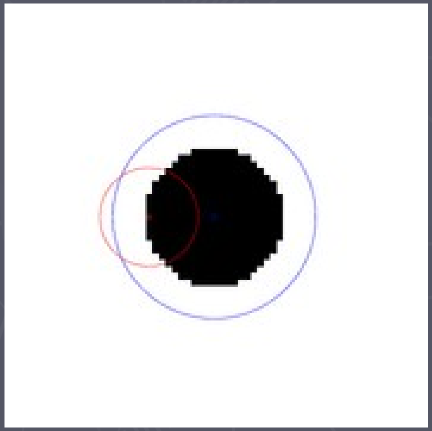
Based on entropy

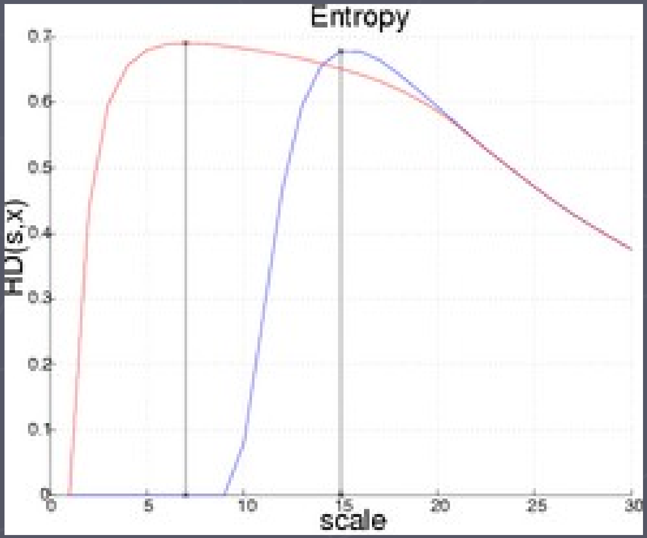


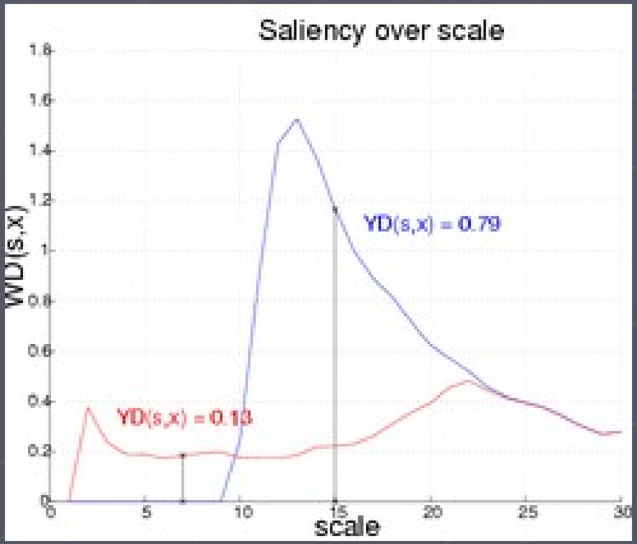
Kadir & Brady's salient regions

Maxima in entropy, combined with inter-scale saliency

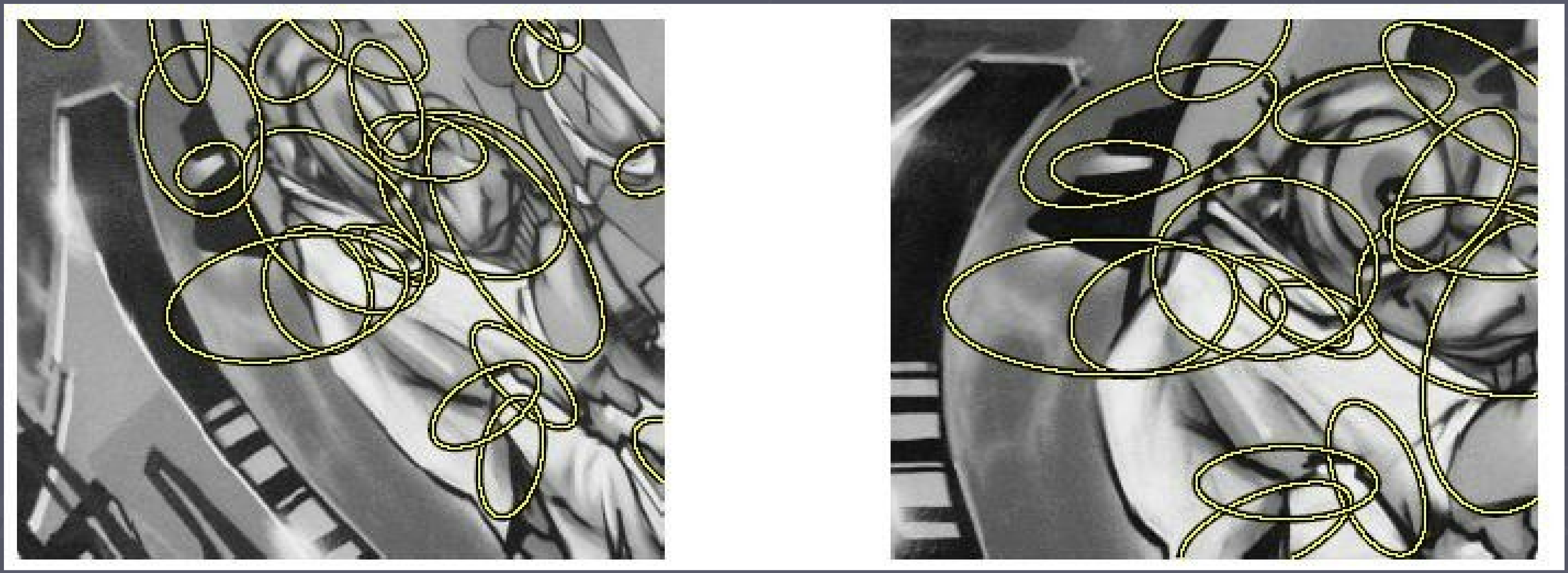
Extended to affine invariance







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.

Overview

Local Invariant Features: What? Why?

- Introduction
- Overview of existing detectors
- Quantitative and qualitative comparison

Local Invariant Features: When? How?

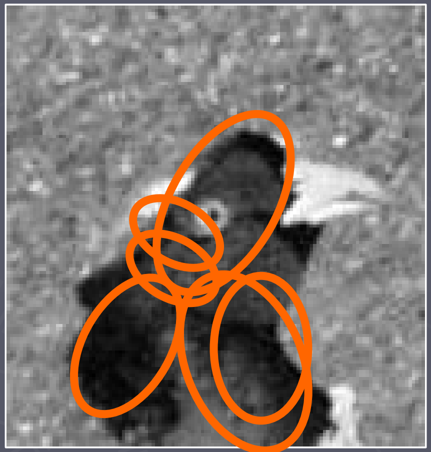
- Feature descriptors
- Applications
- Conclusions

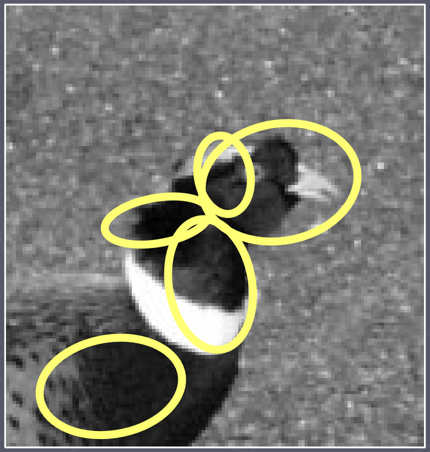
Quantitative comparisons

- 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

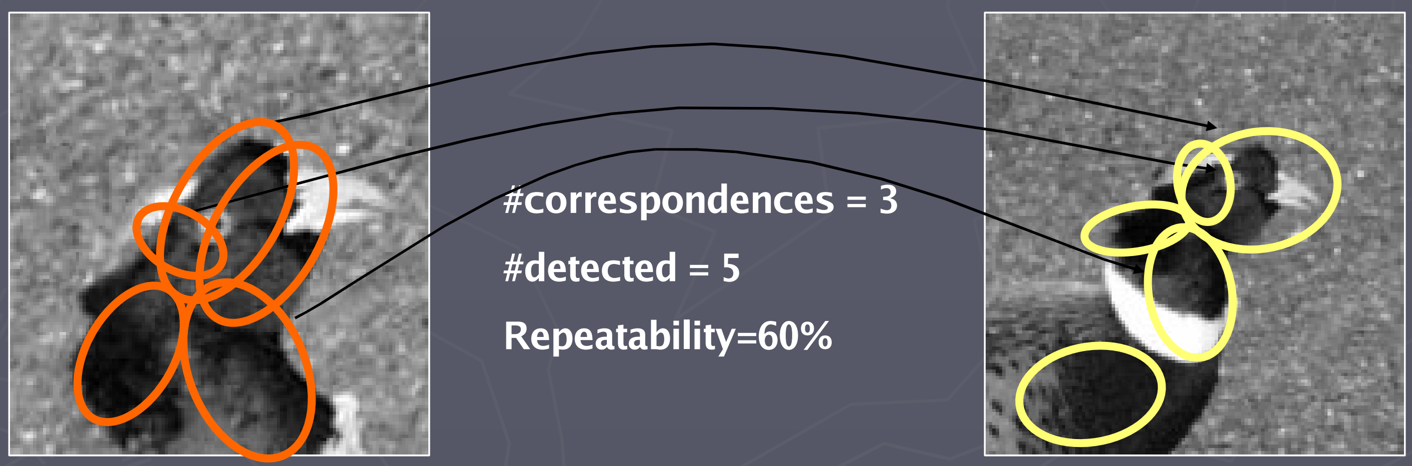




$$repeatability = \frac{\#correspondences}{\#detected} \cdot 100\%$$

Evaluation criteria: repeatability

Repeatability rate : percentage of corresponding points



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

Evaluation criteria: repeatability

Repeatability rate : percentage of corresponding points

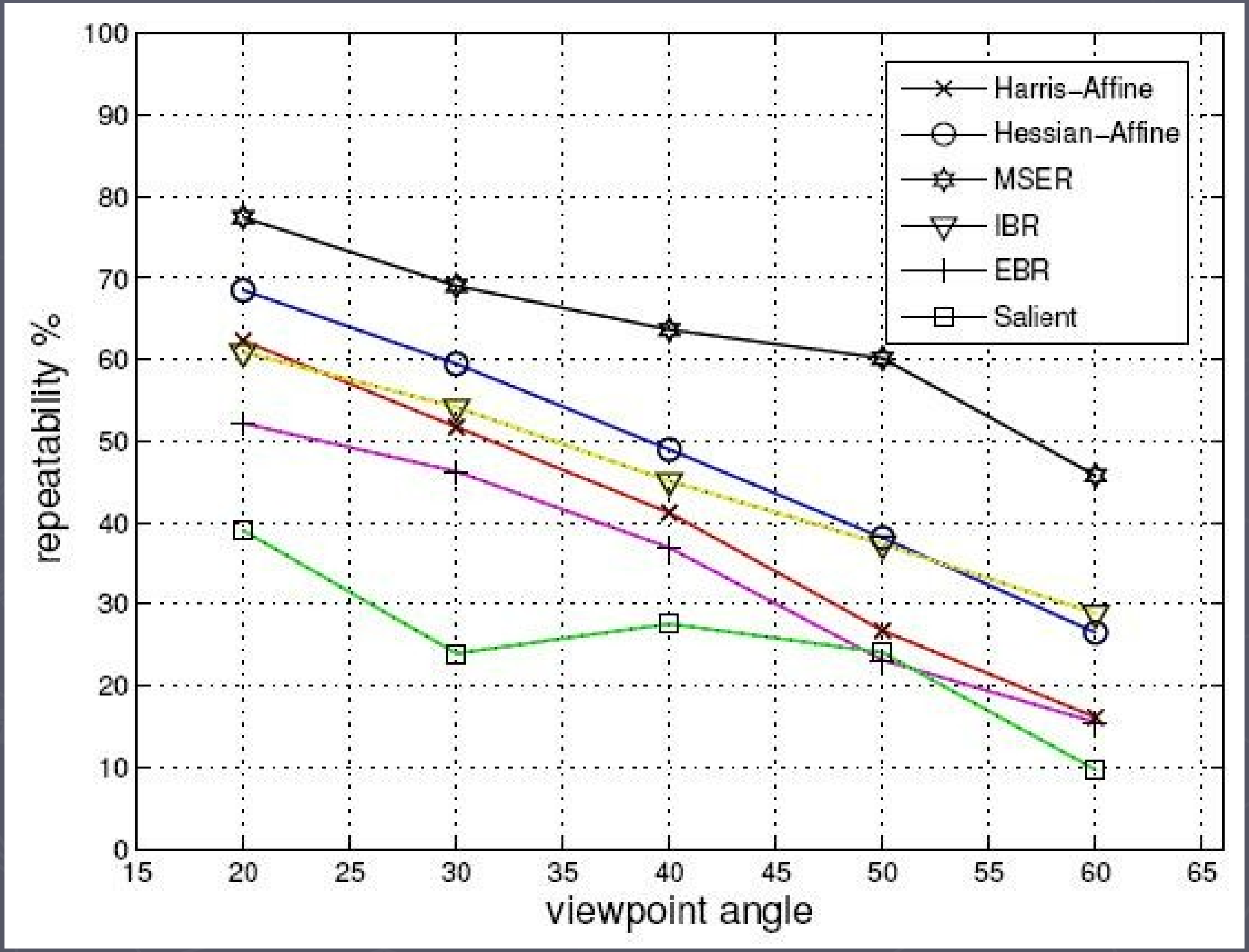
Evaluation criteria: repeatability

Repeatability rate : percentage of corresponding points

•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
  - …

BREAK

Overview

Local Invariant Features: What? Why?

- Introduction
- Overview of existing detectors
- Quantitative and qualitative comparison

Local Invariant Features: When? How?

- Feature descriptors
- Applications
- Conclusions

The ideal feature descriptor

Repeatable (invariant/robust)

Distinctive

Compact

Efficient

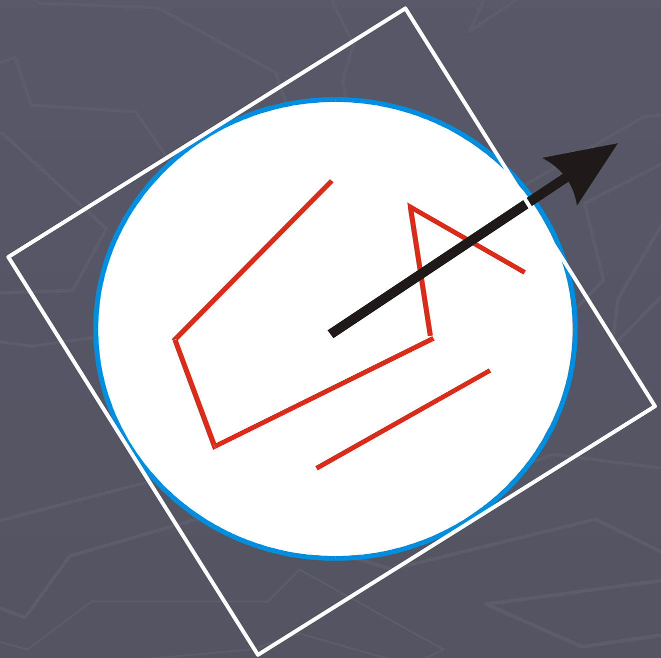
Normalized crosscorrelation

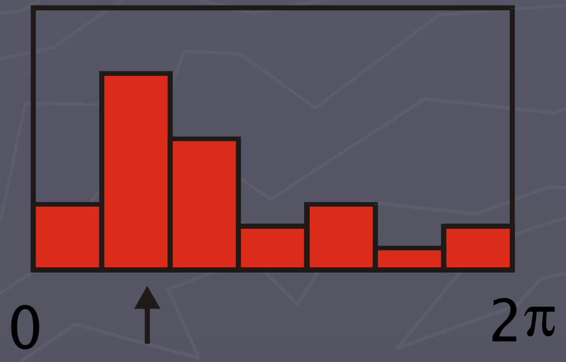
$$NCC = \frac{\sum_{x=-N}^{N}\sum_{y=-N}^{N}(I_1(x,y)-\bar{I}_1)(I_2(x,y)-\bar{I}_2)}{\sqrt{\sum_{x=-N}^{N}\sum_{y=-N}^{N}(I_1(x,y)-\bar{I}_1)^2\sum_{x=-N}^{N}\sum_{y=-N}^{N}(I_2(x,y)-\bar{I}_2)^2}}$$

After 'deskewing' the region:

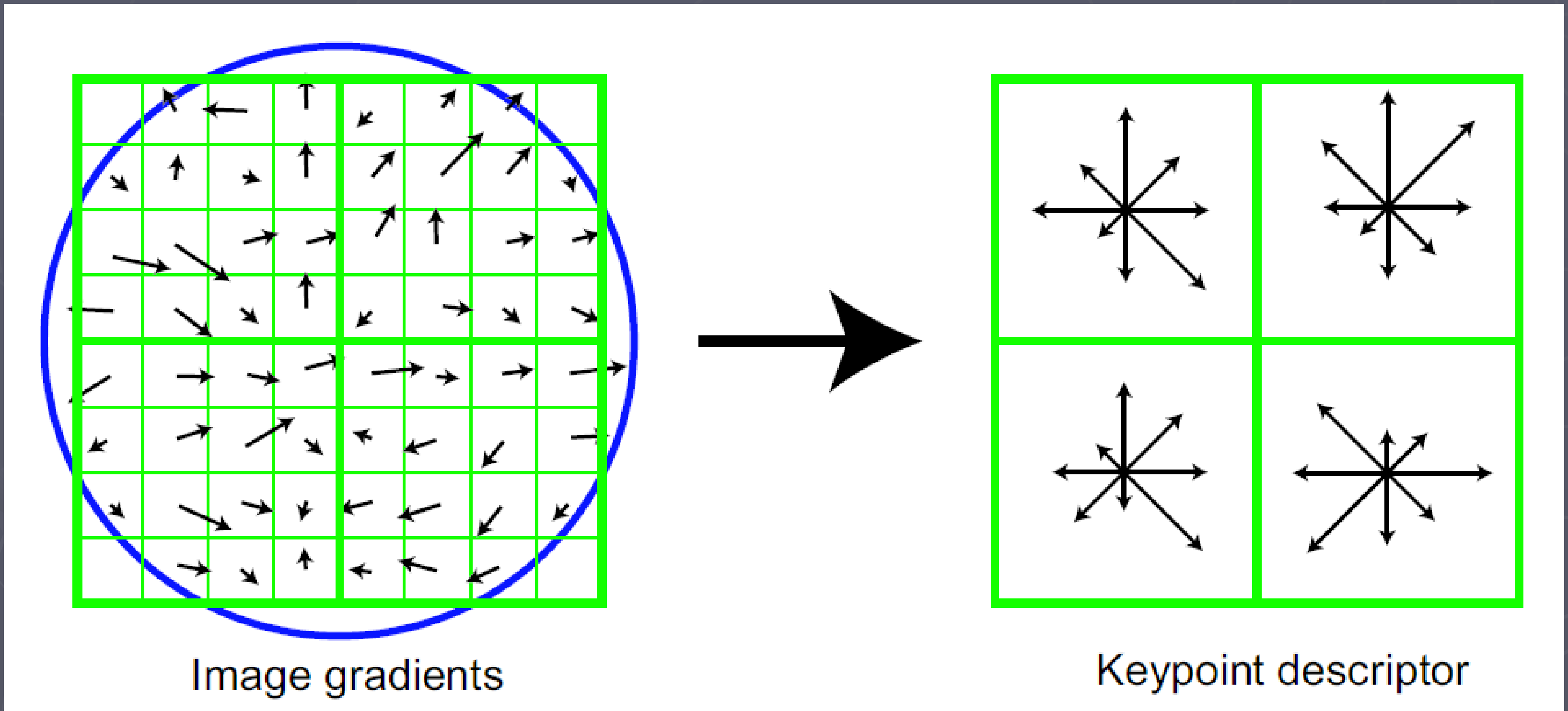
SIFT descriptor

- Orientation assignment
- Distribution-based
- Focusing on image gradients





SIFT descriptor



Others

Steerable filters, moment invariants, local jet, complex filters, shape contexts,

PCA-SIFT, GLOH, HOG,

SURF

Distance measures

Euclidean distance

Mahalanobis distance

$$d_M = \sqrt{(x - x')^T \mathbf{C}^{-1} (x - x')}$$

Overview

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

Applications

- Wide baseline matching
- Recognition of specific objects
- Recognition of object classes

Applications

- Wide baseline matching
- Recognition of specific objects
- Recognition of object classes

Wide baseline matching

Wide baseline matching

- Extract features in each image
- Compute feature descriptors
- Find correspondences
  - Matching strategy
- Check consistency – filter out mismatches
- (Refined matching)

Wide baseline matching

Which features to use ?

- Affine invariant features if large viewpoint changes are expected (>30degrees)
- Accurate features
- Limited number of good matches > large number of medium quality matches
- Take into account typical image content (blobs/corners/prints/…)

MSER, IBR, EBR, …

Matching strategy

- Match to nearest neighbour
- Match to nearest neighbour if distance below a threshold
- Match to nearest neighbour if much closer than second-best match (Lowe, 1999)

- Possibly match in both directions

Consistency checks

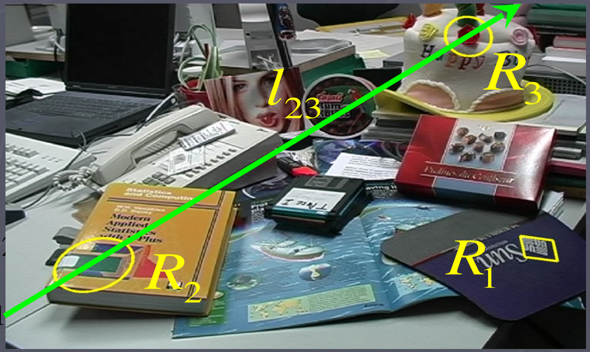
Global constraints

- Epipolar geometry (ransac)
- Homography (ransac)

Semi-local constraints

- (Same neighboring regions) (Schmid, 1998)
- Geometric constraints (Tuytelaars & Van Gool, 2000)
- (Topologic constraints) (Ferrari et al., 2004)
- Photometric constraints

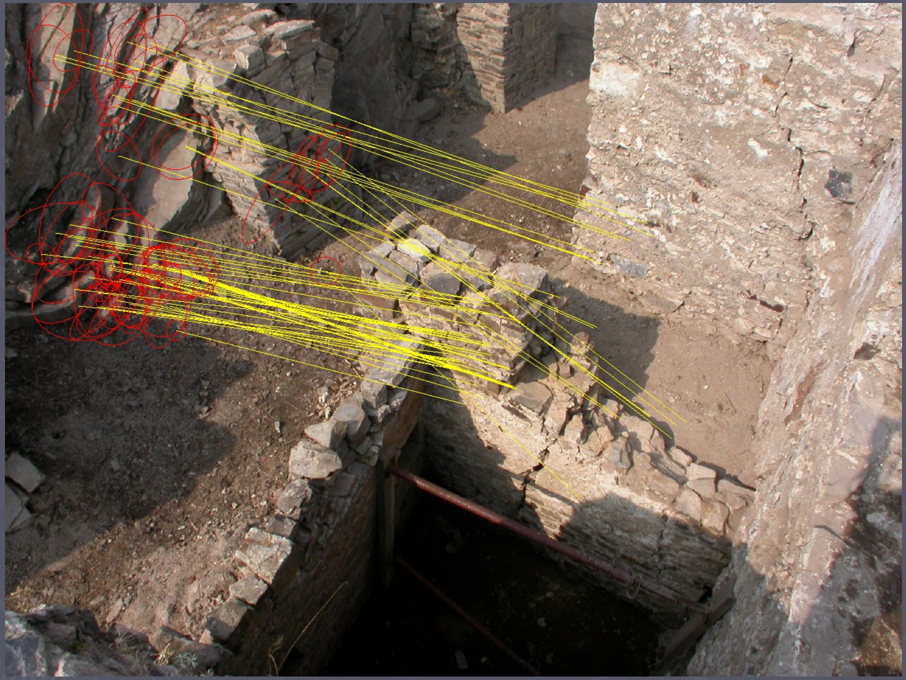
$$\det \begin{bmatrix} a_{23} - b_{23} & b_{13} - a_{13} & \\ a_{22} - b_{22} & b_{12} - a_{12} & a_{1} \\ a_{21} - b_{21} & b_{11} - a_{11} & a_{1} \end{bmatrix}$$

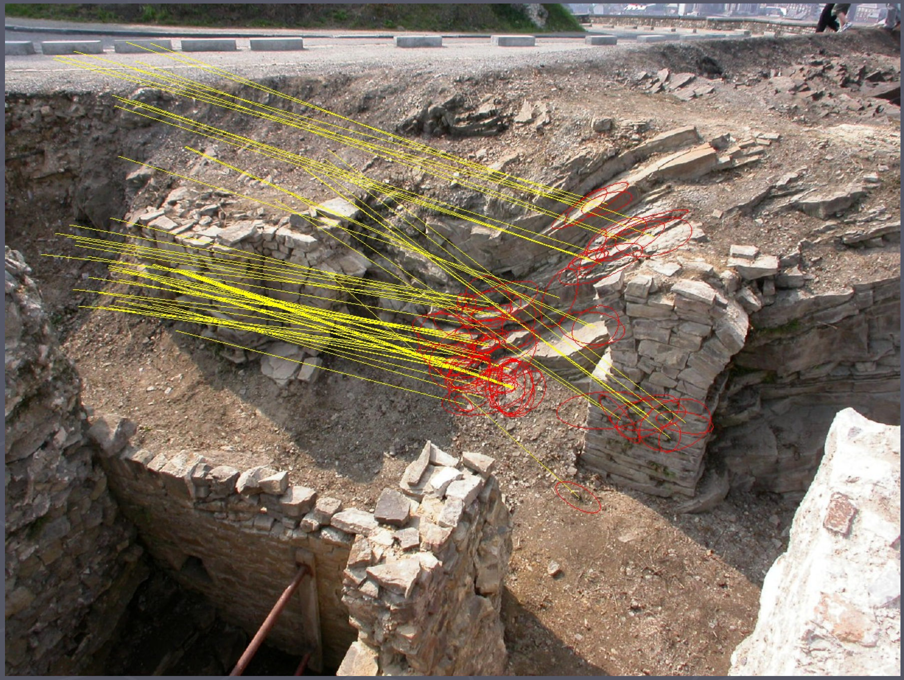


Refined matching

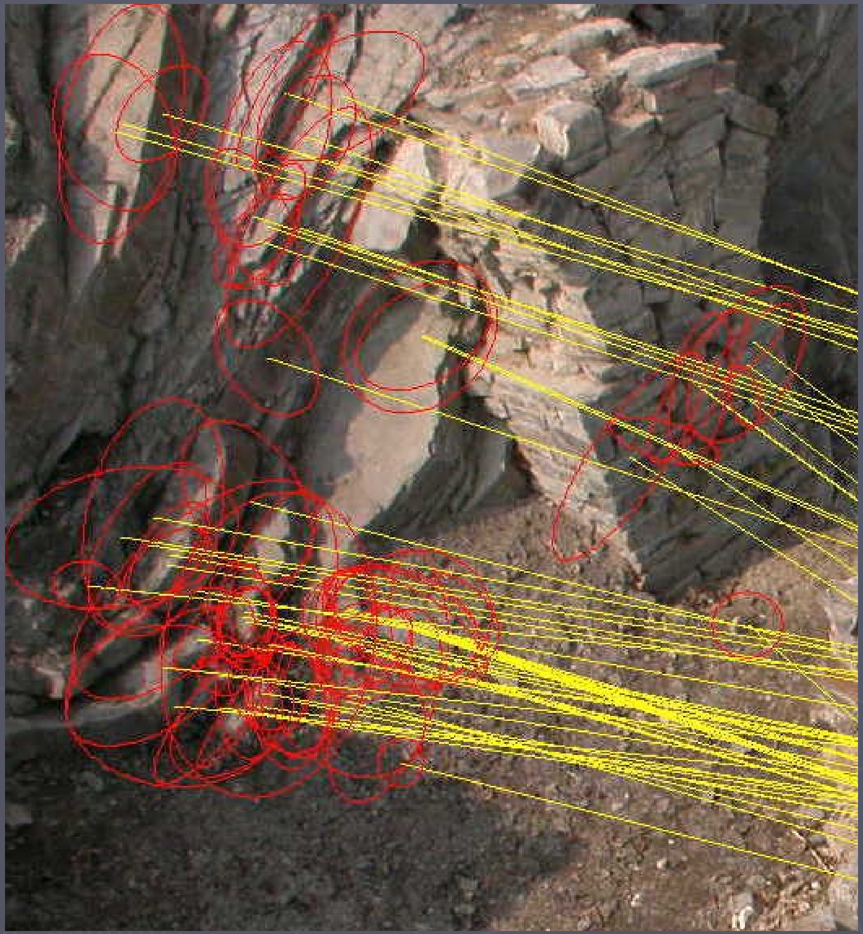
- Search only along epipolar lines
- Construct additional matches (Ferrari et al., 2004)

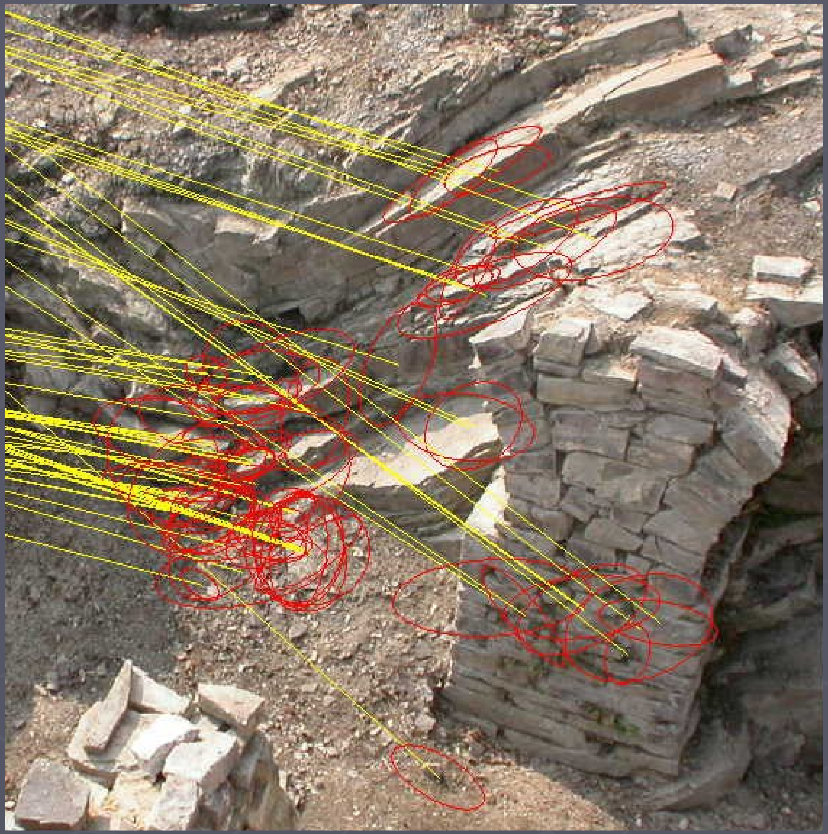
Wide baseline matching





Wide baseline matching





Applications

- Wide baseline matching
- Recognition of specific objects
- Recognition of object classes

Recognition of specific objects

Object recognition can be cast as feature matching problem

Recognition of specific objects

Training:
- Extract features in each model image
- Compute feature descriptors
- Store in database
  - Efficient search structures

Testing:
- Extract features
- Compute feature descriptors
- Match features to database
- Count number of votes
- Post-processing (Lowe, 1999; Rothganger & Ponce, 2003; Ferrari et al., 2004)

Recognition of specific objects

Which features to use ?

- Affine invariant features if large viewpoint changes are expected (>30 degrees)
- Level of invariance needed depends on number of model images
- Features need to be distinctive:
  risk for false matches is much larger
- At least a few good matches
  (if time for post-processing is not an issue)
- Take into account typical image content
  (blobs/corners/prints/)..

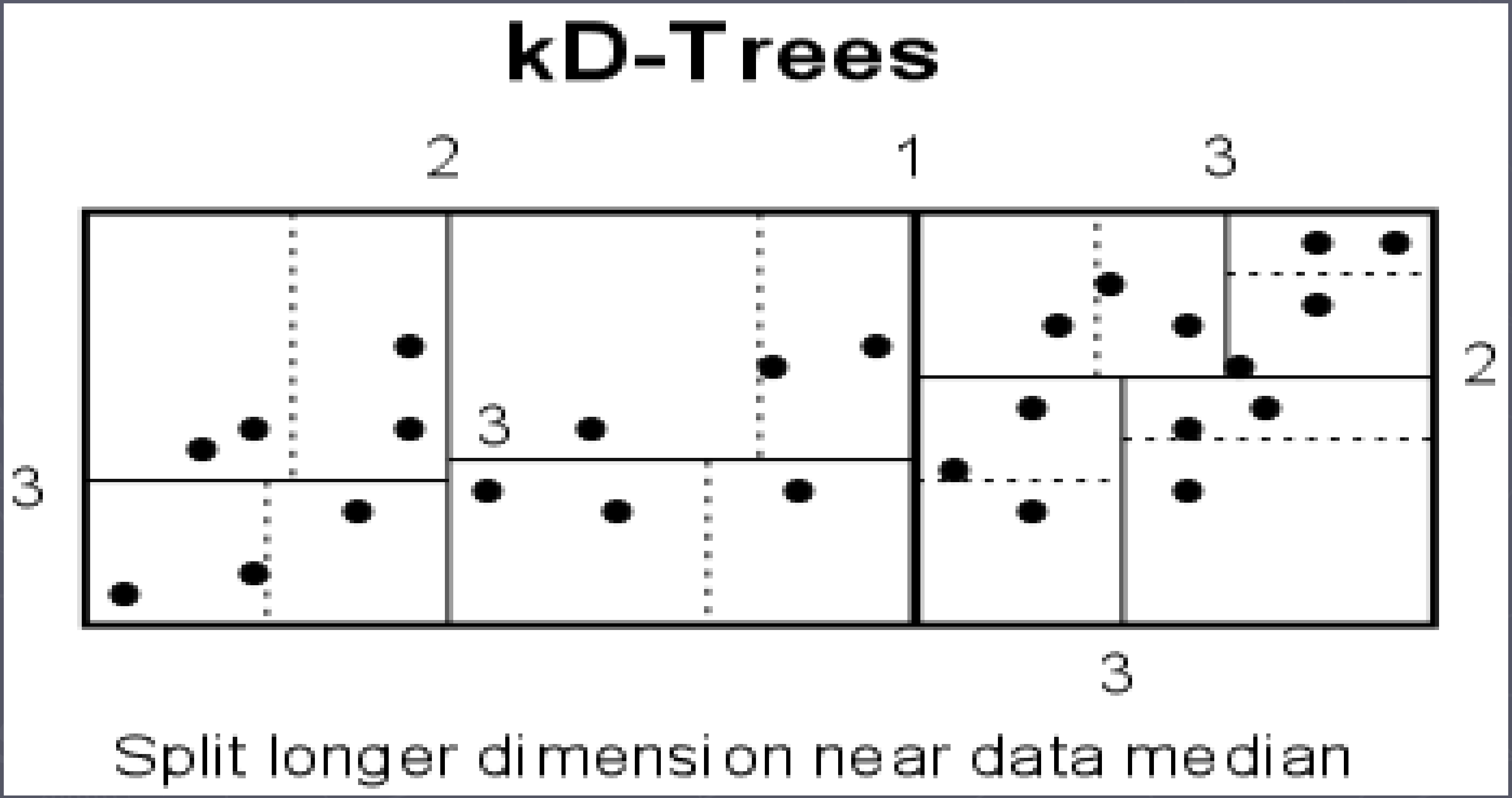
MSER, IBR, EBR, DoG, Harris/Hessian-Laplace/Affine

Image Retrieval

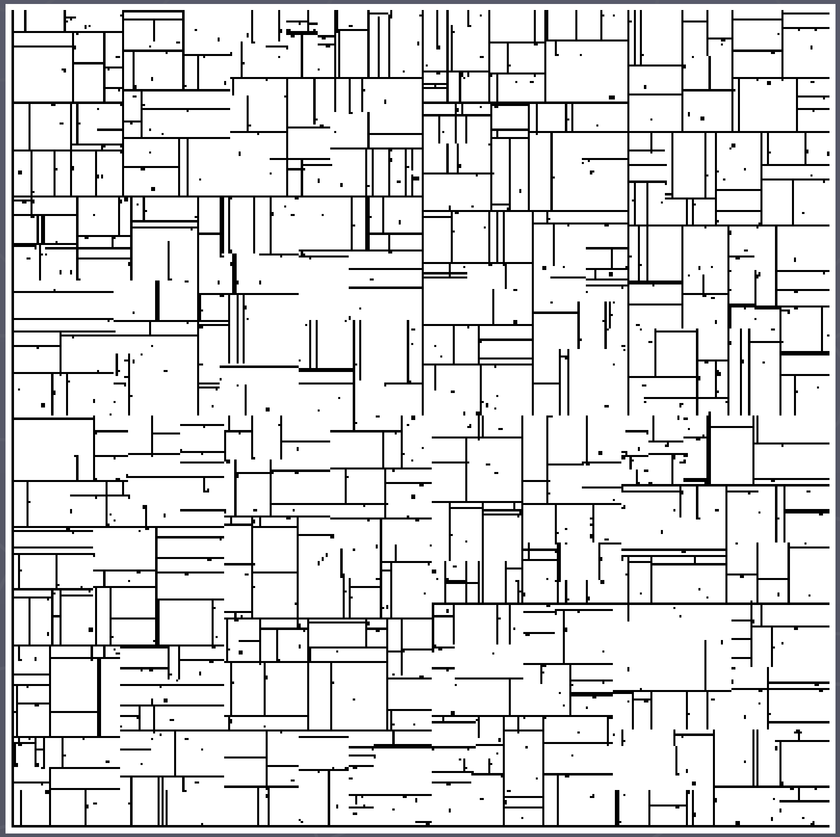
Efficient matching to a database of images

- Kd-tree
- Best bin first (Lowe, 1999)
- Visual vocabulary & inverted files (Sivic & Zisserman, 2003)

Kd-tree



Kd-tree



Best bin first

- Kd-tree less effective in high-dimensional spaces.
- Examine only the N closest bins of the kd-tree

Recognition of specific objects

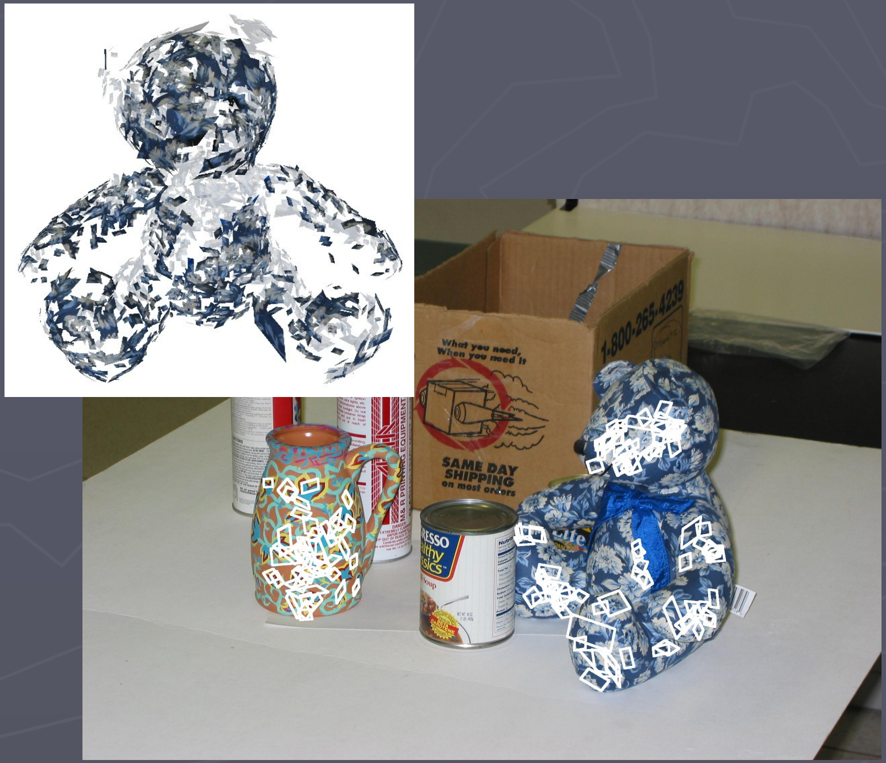
Postprocessing:

- Hough-like scheme
- 3D model
- Image exploration

Recognition of specific objects

Postprocessing:

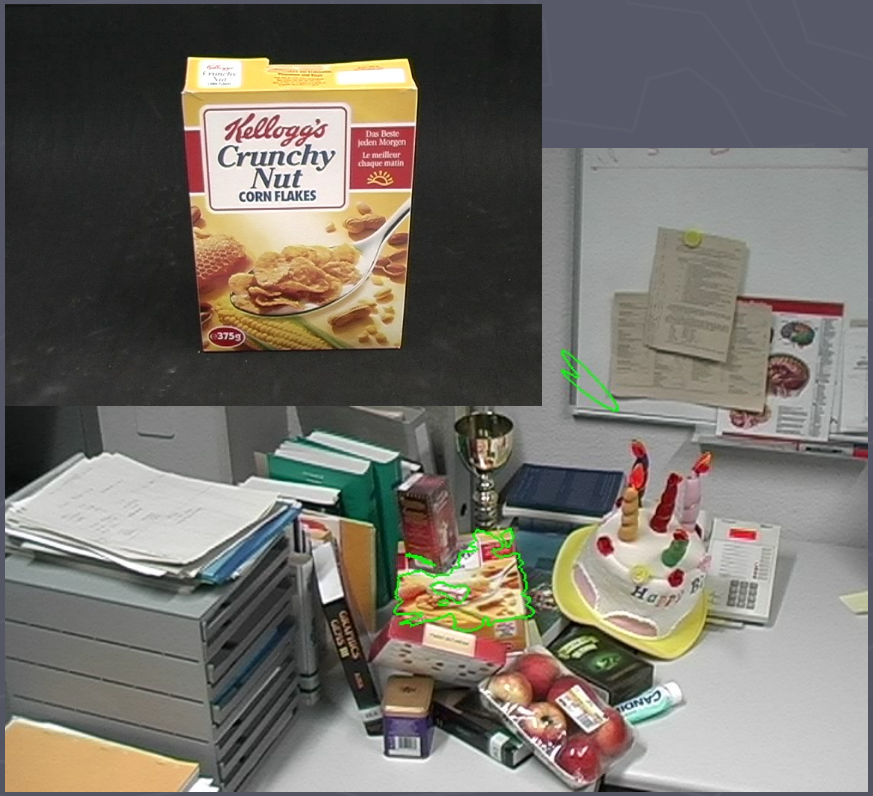
- Hough-like scheme
- 3D model
- Image exploration



Recognition of specific objects

Postprocessing:

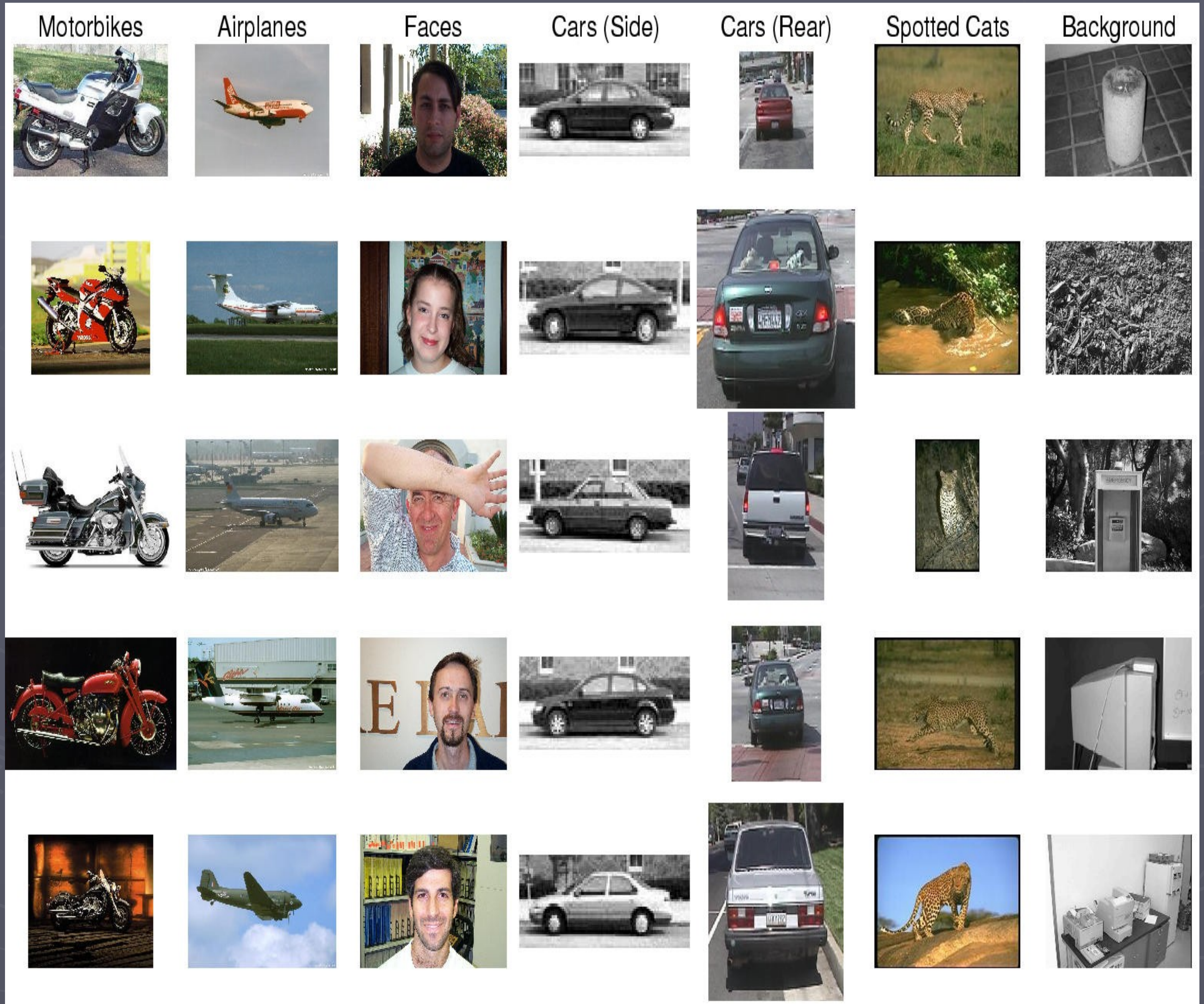
- Hough-like scheme
- 3D model
- Image exploration



Applications

- Wide baseline matching
- Recognition of specific objects
- Recognition of object classes

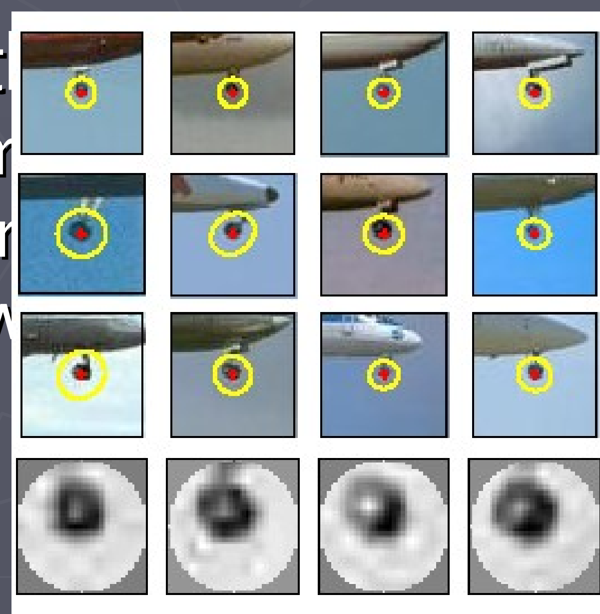
Recognition of object classes

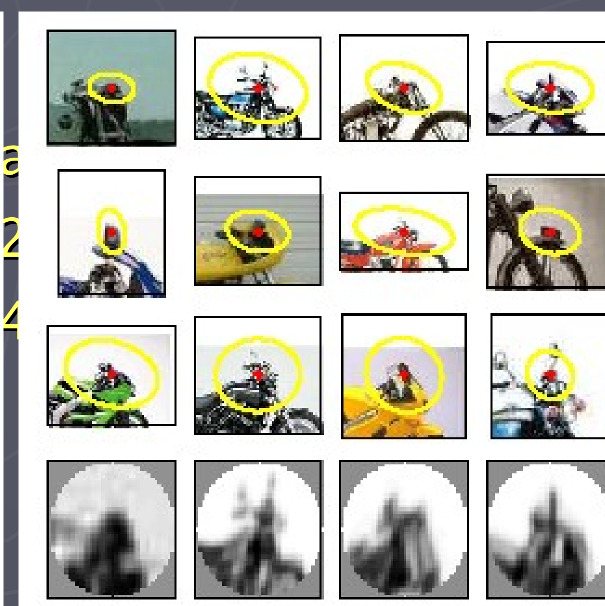


Recognition of object classes

Training

- Extract local features
- Compute feature descriptors
- Cluster features in object parts / codebooks / visual words
- Build model wit
  - Constellation m
  - Implicit shape r
  - Bag-of-visual-w
- Train classifier





Recognition of object classes

Which features to use ?

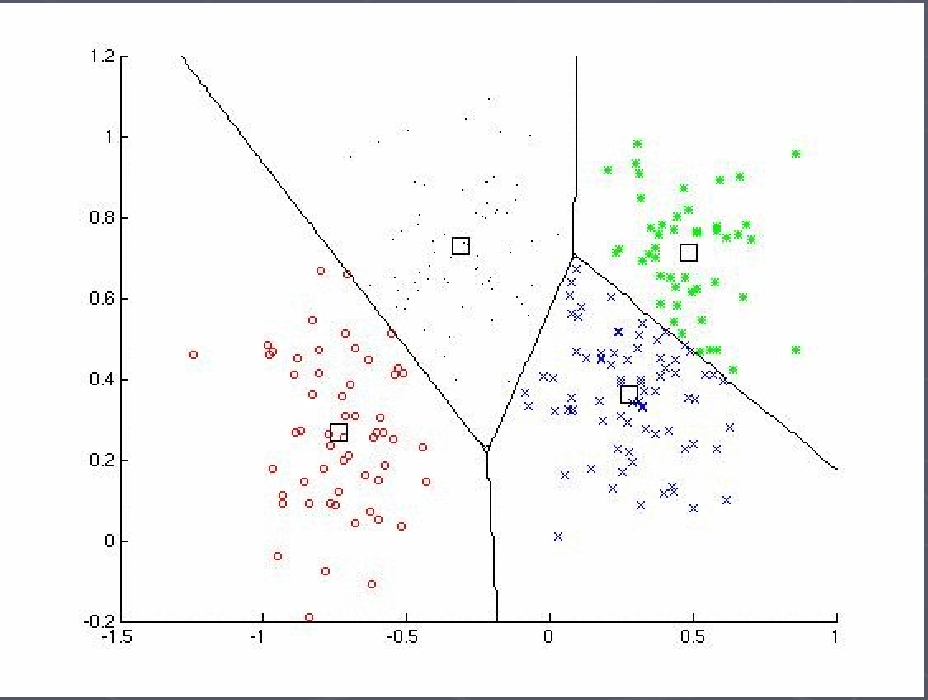
- Scale invariant features
- Robust features
- Large number of features (depends on model used)
- Accuracy not important

Salient Regions, Harris/Hessian-Laplace

Recognition of object classes

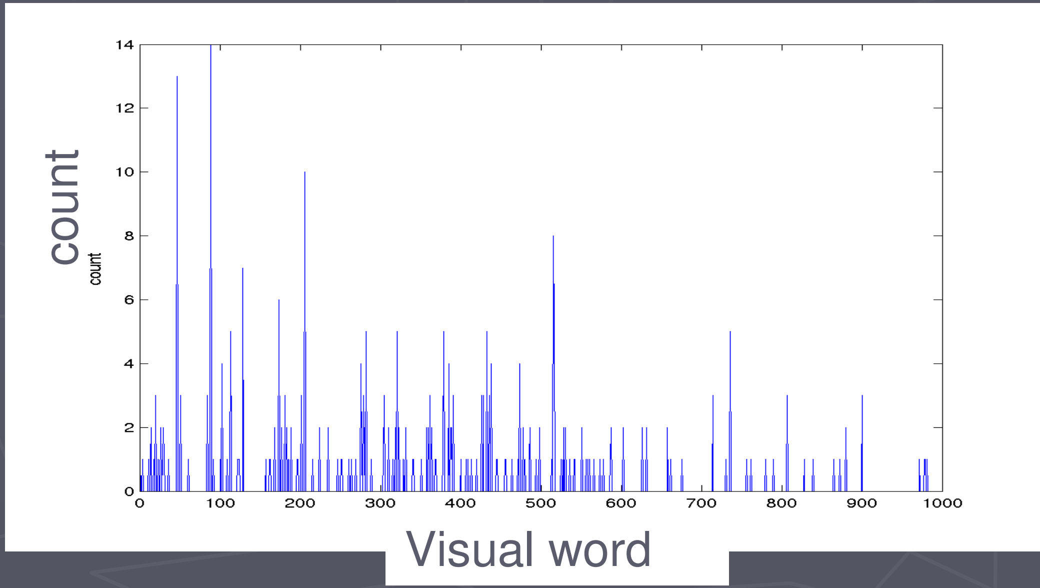
Clustering features into 'visual words'

- K-means
- Jurie & Triggs, ICCV05



Recognition of object classes

Bag-of-visual-words image representation:



Other applications

- Image mosaicking
- Mobile robot navigation
- Scene classification
- Texture classification
- Video data mining
- Object discovery
- 3D reconstruction
- …

Overview

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

Do's and Don'ts

DO

- Think about the right level of invariance
- Rely on statistics

DO NOT

- Expect wonders
- Rely on a single local feature
- Evaluate methods based on a single image

Questions ?

Tinne.Tuytelaars@esat.kuleuven.be

http://homes.esat.kuleuven.be/~tuytelaa/ECCV06tutorial.html

http://www.robots.ox.ac.uk/~vgg/research/affine