Computer Vision I

Bags of Visual Words – 05.06.2013

TECHNISCHE UNIVERSITÄT DARMSTADT

with slides from:

Bernt Schiele
Li Fei-Fei
Rick Szeliski
Steve Seitz
Thomas Hofmann

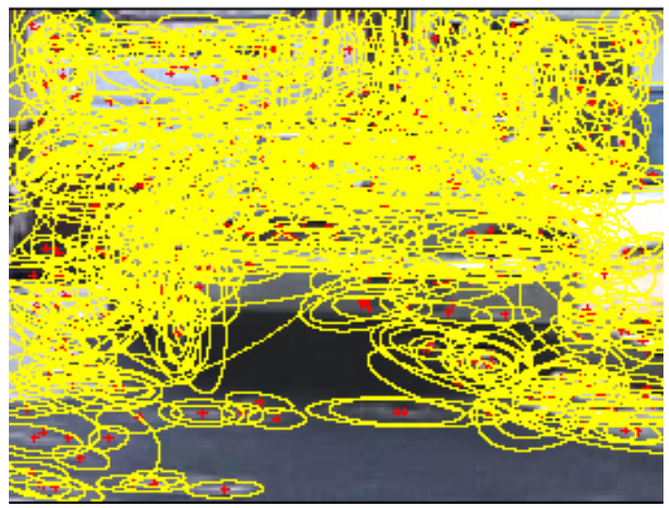
visual inference

Feature representation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Dense representation (regular grid):
  - Color histogram approach [Swain & Ballard '91]
  - Multidimensional receptive field histograms [Schiele & Crowley '96-'00]

- Sparse feature representations:
  - Find key points with interest point detector
    - e.g. scale- or affine-invariant
  - Represent key points with feature vectors
    - e.g. SIFT, ShapeContext



[Fei-Fei]

Overview

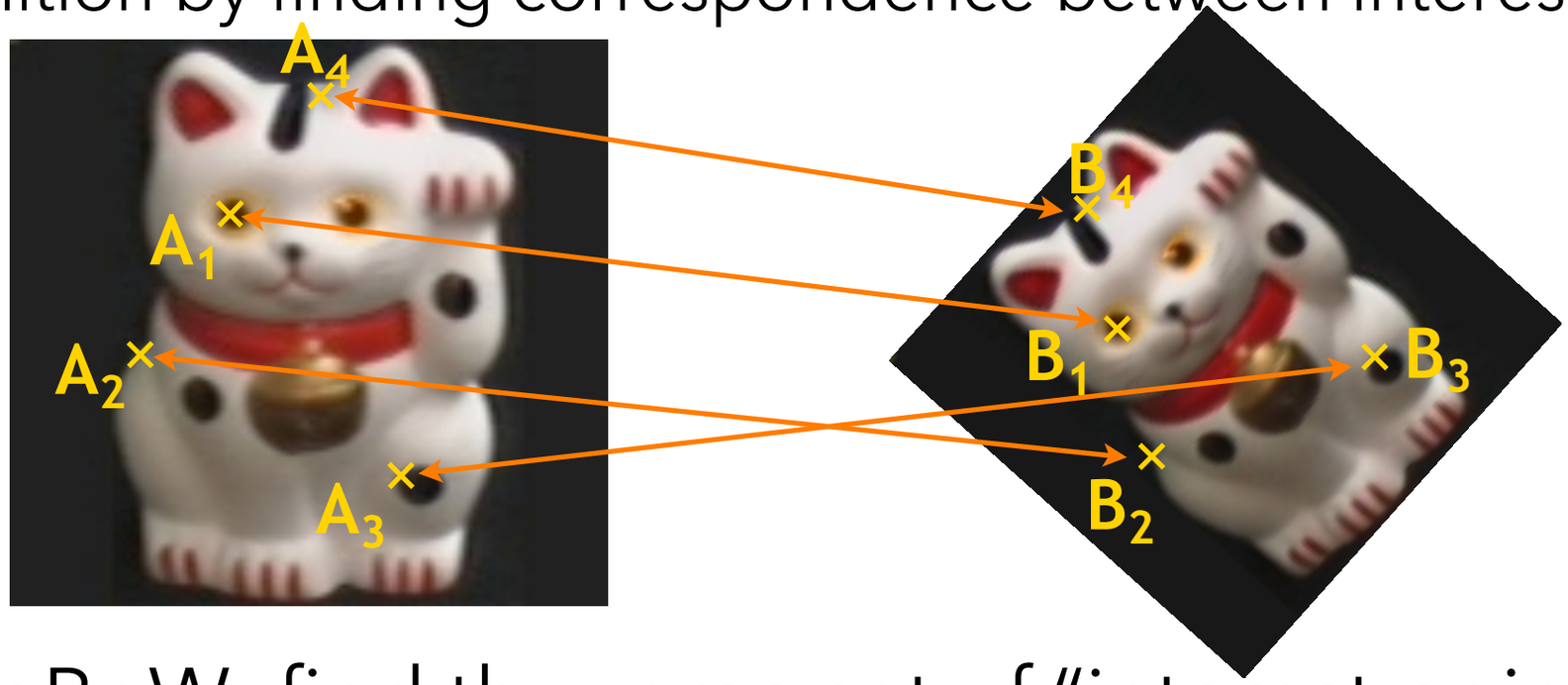
TECHNISCHE UNIVERSITÄT DARMSTADT

- **1 - Local Interest Point Detection**
  - Finding discriminative points (Harris, Hessian)
  - Scale invariant interest point detection (Harris-Laplace)
- 2 - Local Descriptors (= Features)
- 3 - Bag-of-Words Model (BoW)
  - BoW for Object Categorization

Motivation for Local Interest Point Detection

TECHNISCHE UNIVERSITÄT DARMSTADT

- General idea of interest point detection:
  - Recognition by finding correspondence between interest points

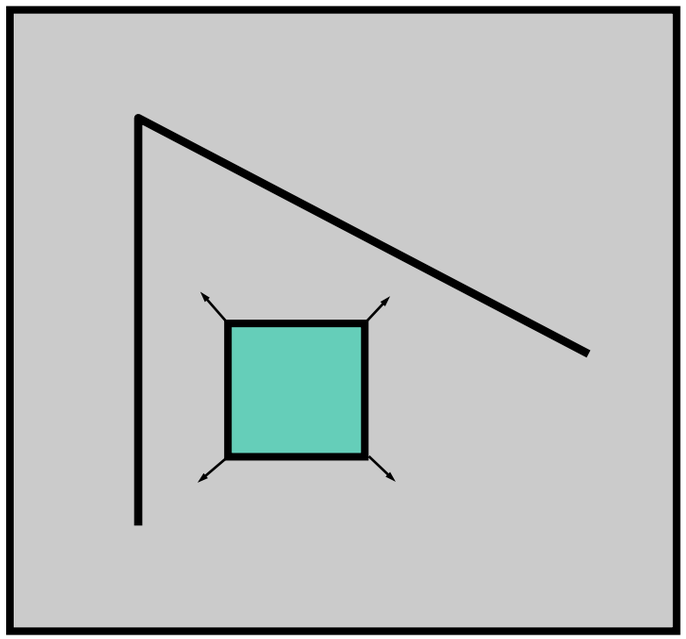


- Goal for BoW: find the same set of “interest points” that are
  - discriminative for object recognition
  - can be always found - even in the presence of arbitrary geometric and photometric transformations

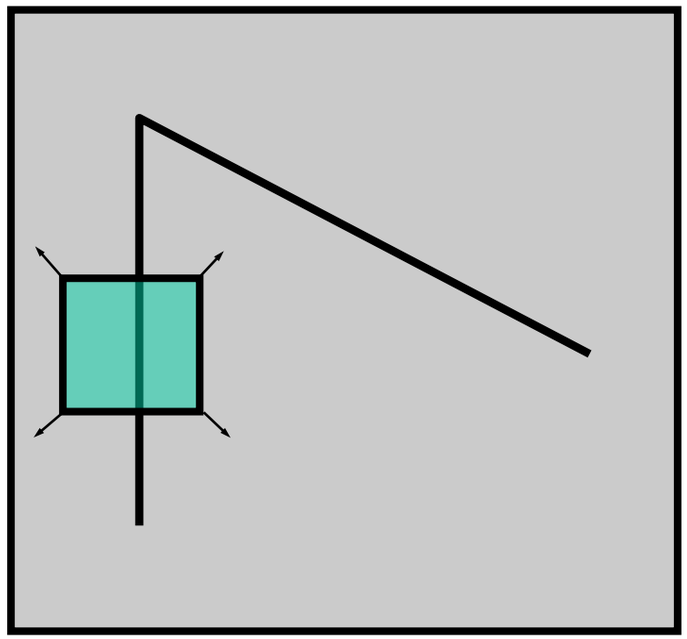
Feature detection

TECHNISCHE UNIVERSITÄT DARMSTADT

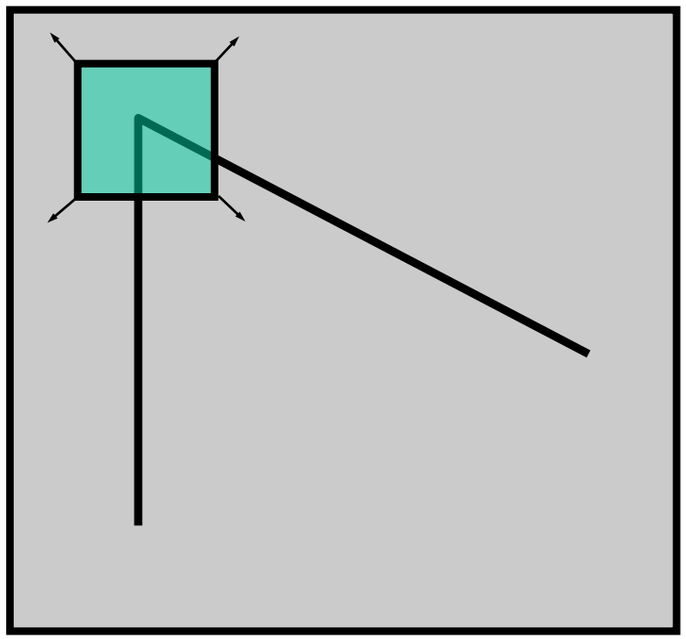
- Local measure of feature uniqueness
  - How does the window change when you shift it?
  - Shifting the window in any direction causes a big change



"flat" region:
no change in all directions



"edge":
no change along the edge direction



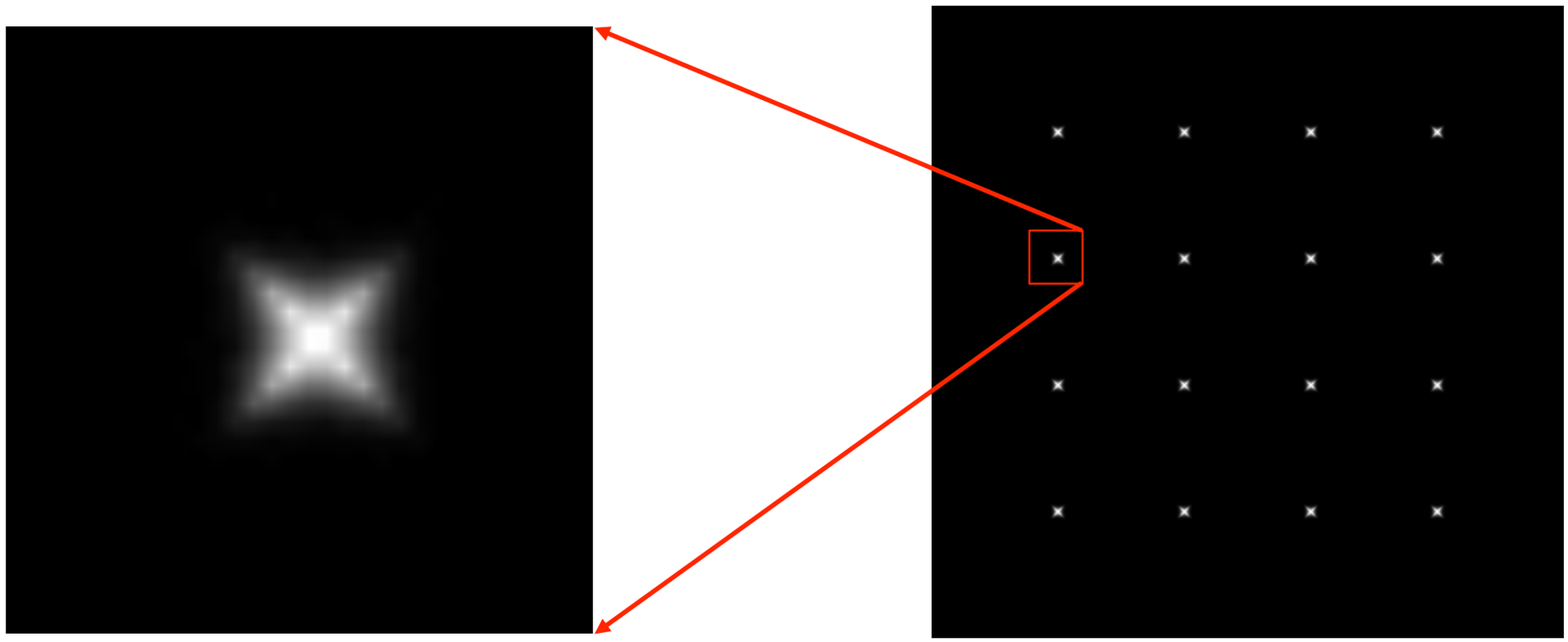
"corner":
significant change in all directions

[Szeliski & Seitz]

Feature detection summary

TECHNISCHE UNIVERSITÄT DARMSTADT

- Here's what you do
  - Compute the gradient at each point in the image
  - Create the H matrix from the entries in the gradient
  - Compute the eigenvalues.
  - Find points with large response ($\lambda\_$ > threshold)
  - Choose those points where $\lambda\_$ is a local maximum as features



The Harris operator

TECHNISCHE UNIVERSITÄT DARMSTADT

- λ_- is a variant of the "Harris operator" for feature detection

$$f = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$
$$= \det(H) - \alpha \cdot \mathrm{trace}(A)^2$$

- The trace is the sum of the diagonals, i.e., $\mathrm{trace}(H) = h_{11} + h_{22}$
- Very similar to λ_- but less expensive (no square root)
- Called the "Harris Corner Detector" or "Harris Operator"
- Lots of other detectors, this is one of the most popular

[Szeliski & Seitz]

Harris features (in red)

TECHNISCHE UNIVERSITÄT DARMSTADT

[Szeliski & Seitz]

Discussion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Interest-point detection so far:
  - Based on Harris or Hessian detector
  - Finds interesting, i.e. discriminative points
    (Harris detector was the “de-facto” standard for a long time)
  - Used for recognition, correspondence for stereo, sparse optical flow/motion, etc.

- Remember the goals of interest point detection:
  - Distinctiveness vs. invariance to transformation
  - Harris & Hessian find distinctive points - but they are **not** invariant to scale, affine and projective transformations

Geometric Transformations

TECHNISCHE UNIVERSITÄT DARMSTADT

- Example of different geometric transformations:
  - (1) original
  - (2) similarity transformation (translation, image plane rotation, scaling)
  - (3) projective transformation

(1)

(2)

(3)

Next

TECHNISCHE UNIVERSITÄT DARMSTADT

- Scale-Invariant Interest Point Detection
  - Matching images of different scales
  - Automatic scale selection
  - Scale invariant methods for feature extraction
    - Example: Harris-Laplace

Matching image patches

TECHNISCHE UNIVERSITÄT DARMSTADT

- of different scales

Matching image patches

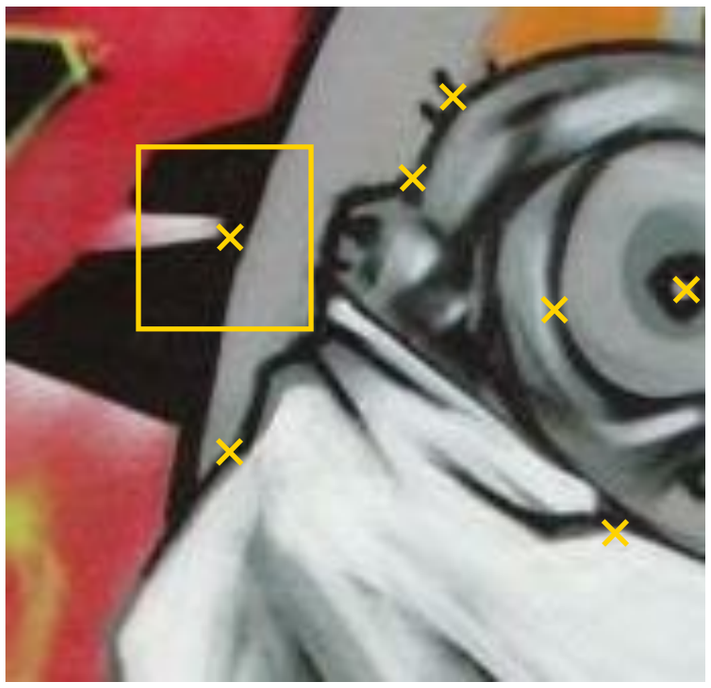
TECHNISCHE UNIVERSITÄT DARMSTADT

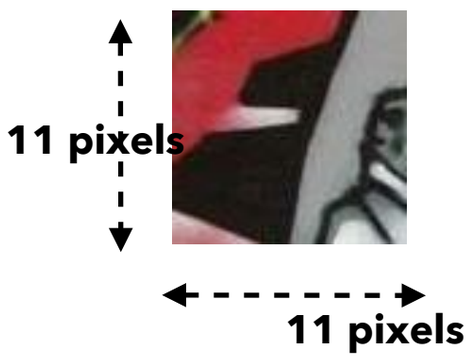
- Detecting interest points

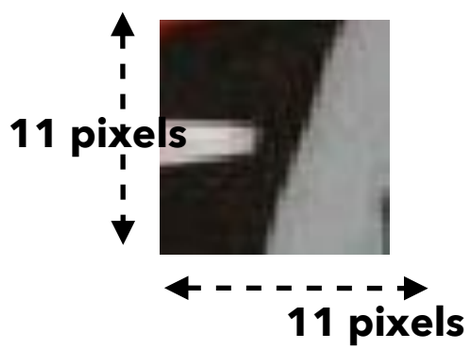
Matching image patches

TECHNISCHE UNIVERSITÄT DARMSTADT

- Extracting patches



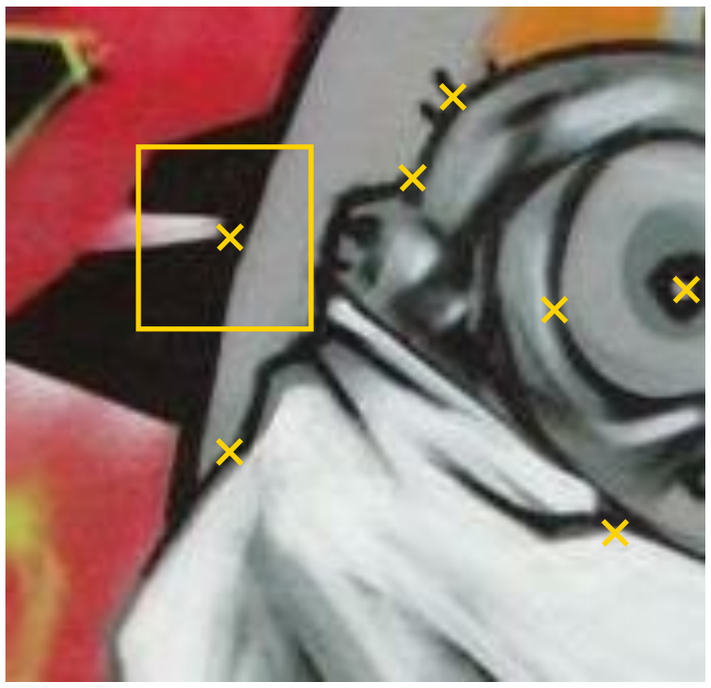


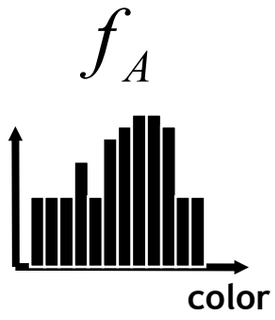


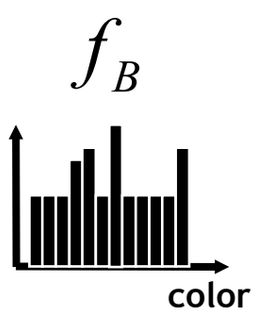
Matching image patches

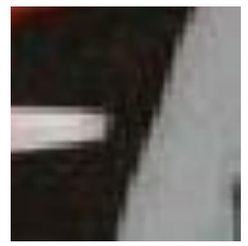
TECHNISCHE UNIVERSITÄT DARMSTADT

- Computing descriptors





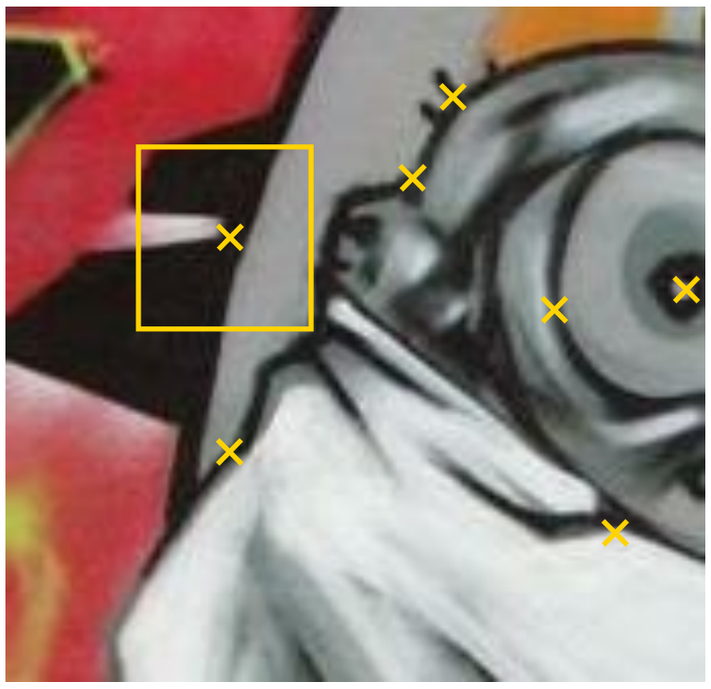


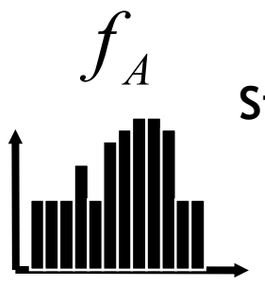


Matching image patches

TECHNISCHE UNIVERSITÄT DARMSTADT

- Comparing descriptors
  - Impossible to match – different histograms due to different patch content

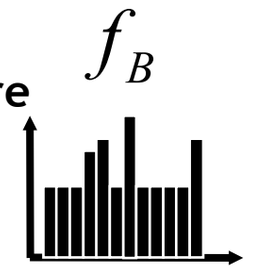


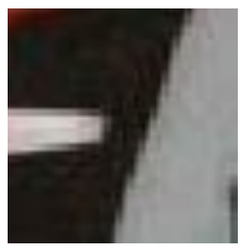


Similarity measure

$$\neq$$

$$d(f_A, f_B)$$

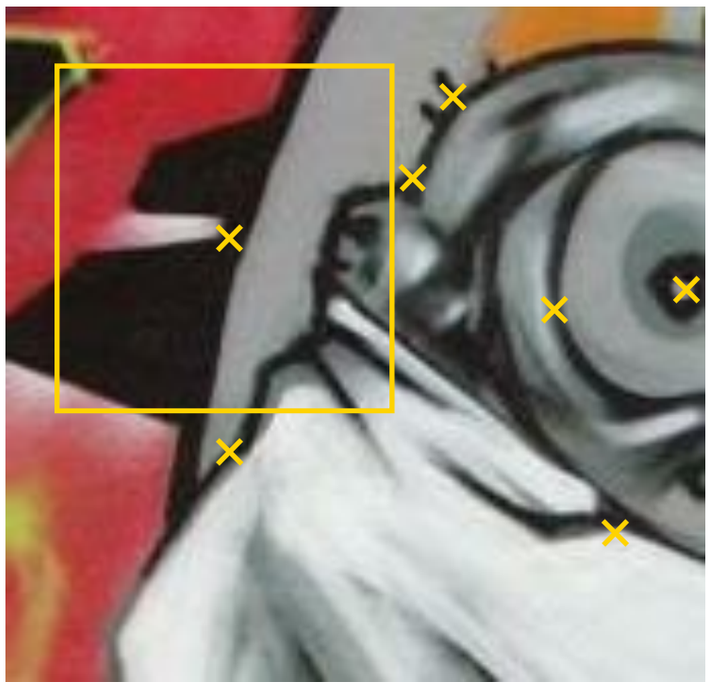


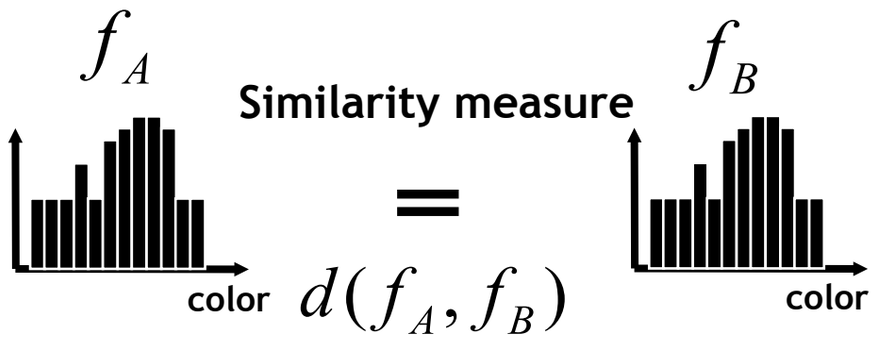


Matching image patches

TECHNISCHE UNIVERSITÄT DARMSTADT

- Comparing descriptors
  - The patch should contain the same image – how to find the correct size?

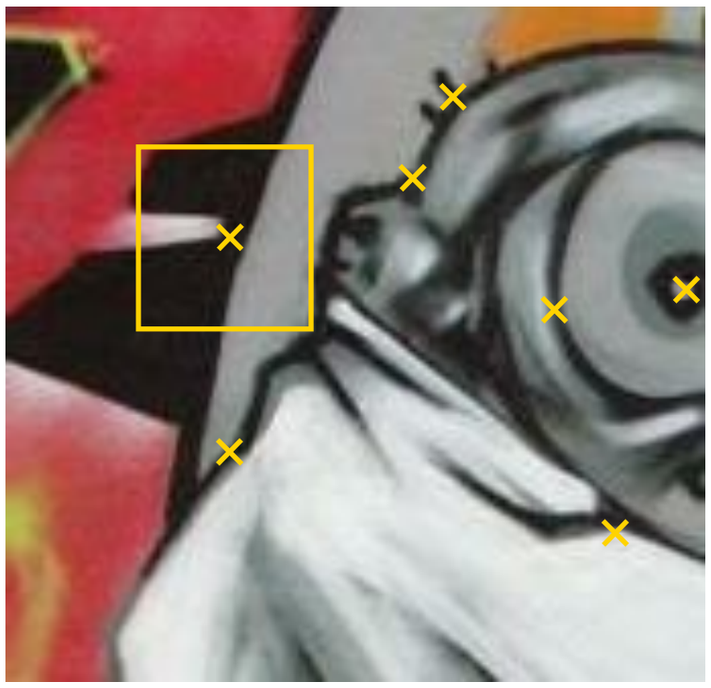


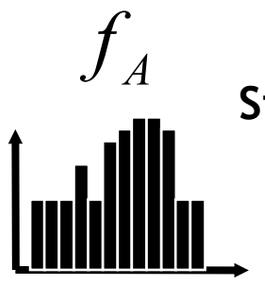


Matching image patches

TECHNISCHE UNIVERSITÄT DARMSTADT

- Comparing descriptors while varying the patch size

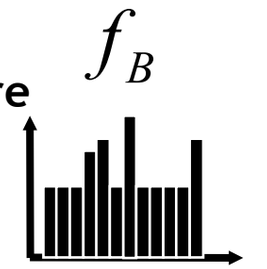


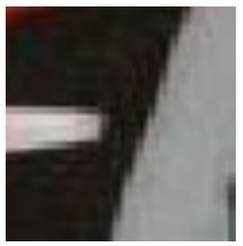


Similarity measure

$$\neq$$

$$d(f_A, f_B)$$

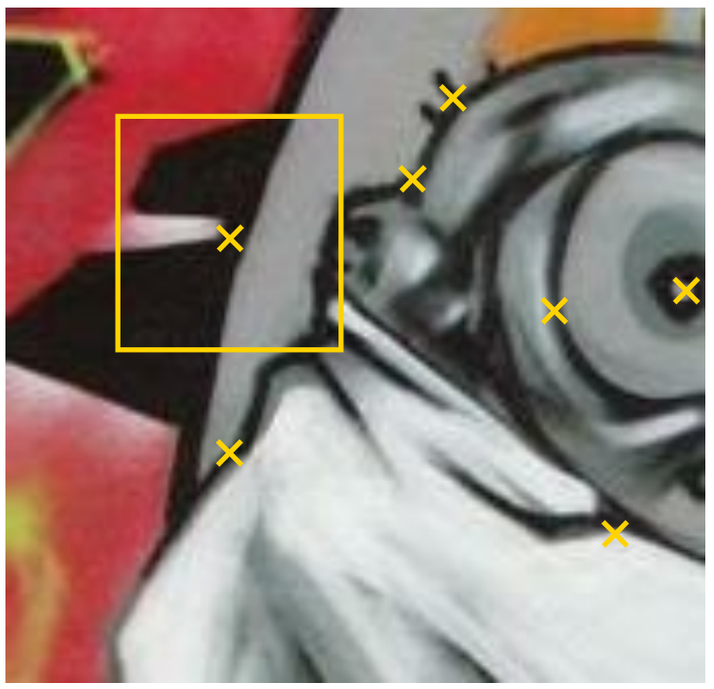


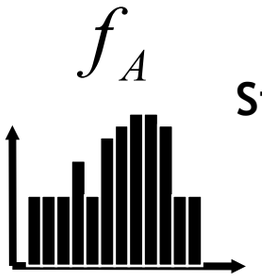


Matching image patches

TECHNISCHE UNIVERSITÄT DARMSTADT

- Comparing descriptors while varying the patch size

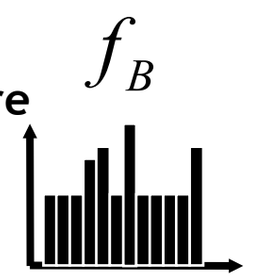




Similarity measure

$$\neq$$

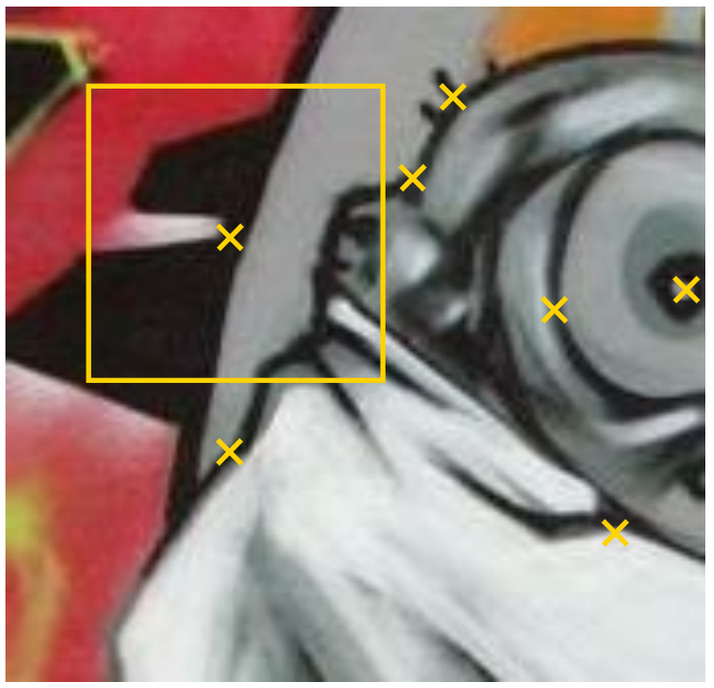
$$d(f_A, f_B)$$

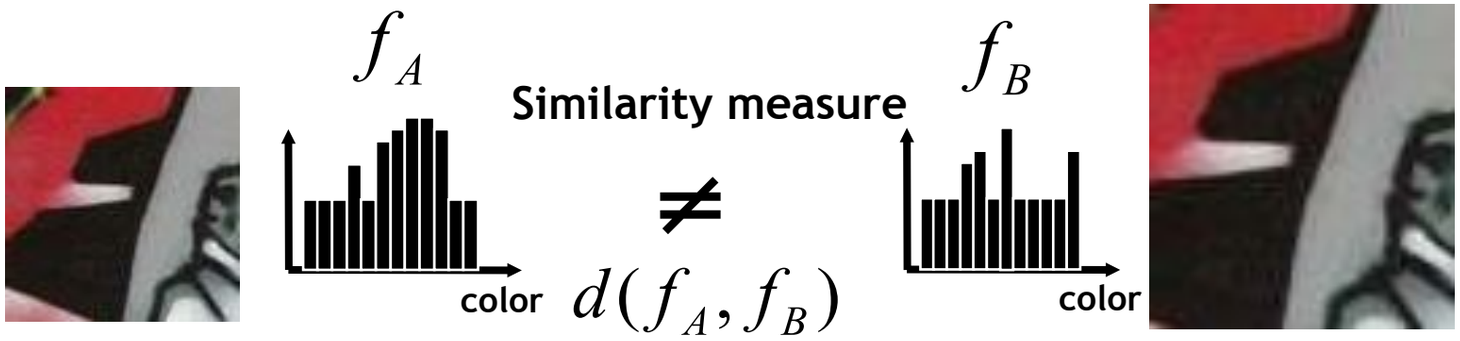


Matching image patches

TECHNISCHE UNIVERSITÄT DARMSTADT

- Comparing descriptors while varying the patch size

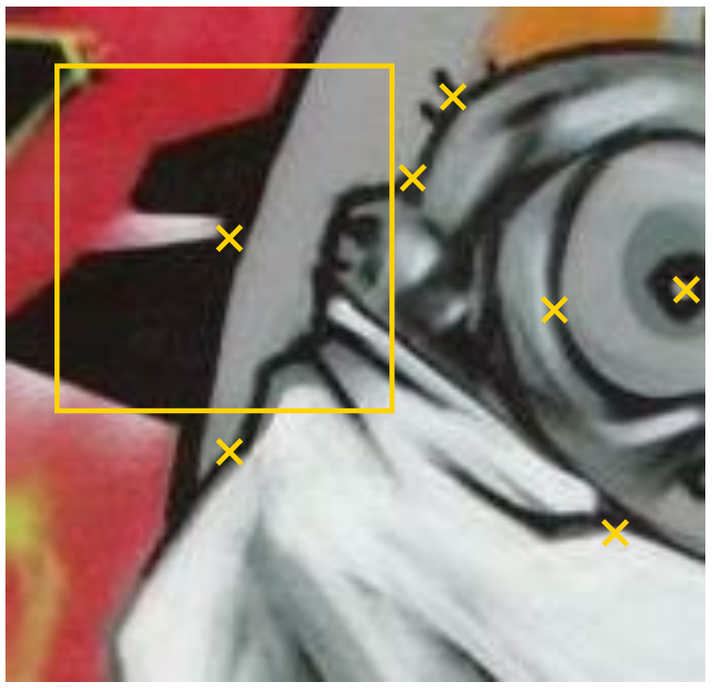


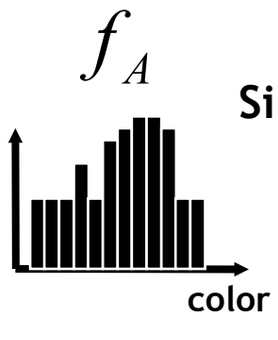


Matching image patches

TECHNISCHE UNIVERSITÄT DARMSTADT

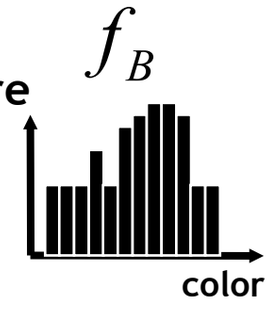
- Comparing descriptors while varying the patch size





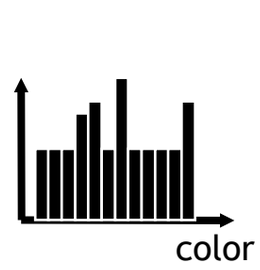
Similarity measure

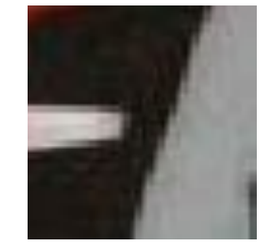
$= d(f_A, f_B)$

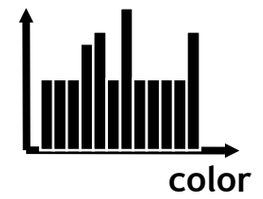


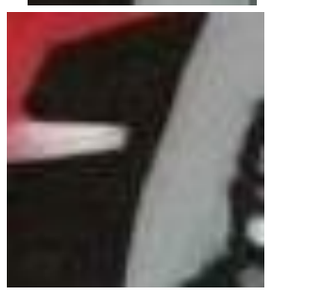
Matching image patches

- Comparing descriptors while varying the patch size
  - Computationally inefficient/prohibitive
  - Inefficient, but possible for matching
  - Prohibitive for retrieval in large databases
  - Prohibitive for recognition









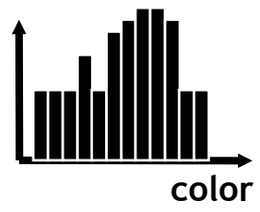
f_A

Similarity measure

=

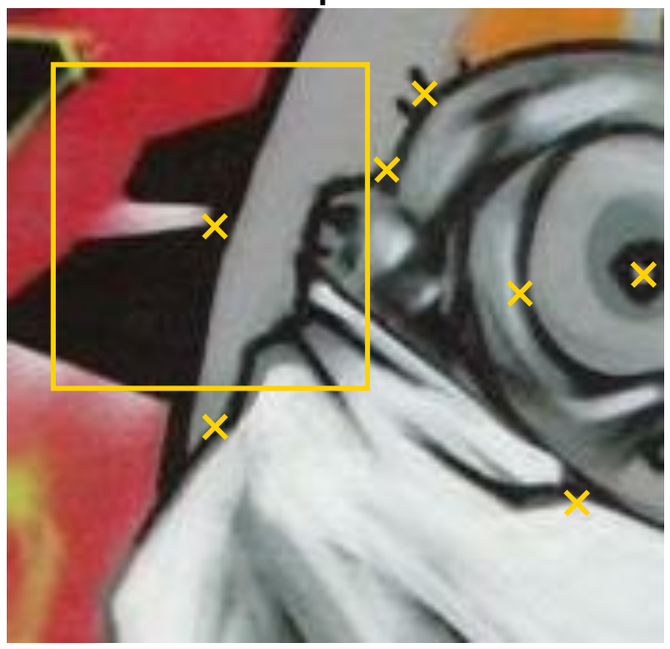
$d(f_A, f_B)$

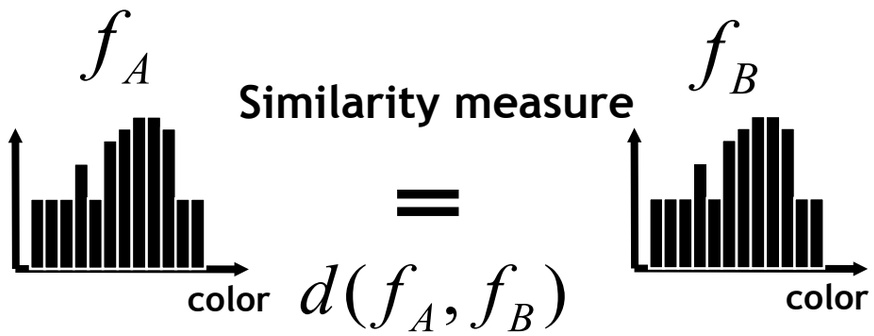
f_B



Matching image patches

- Detector finds location and scale of interest points
  - In both images: independent automatic scale detection
  - by finding “characteristic” scale of an interest point

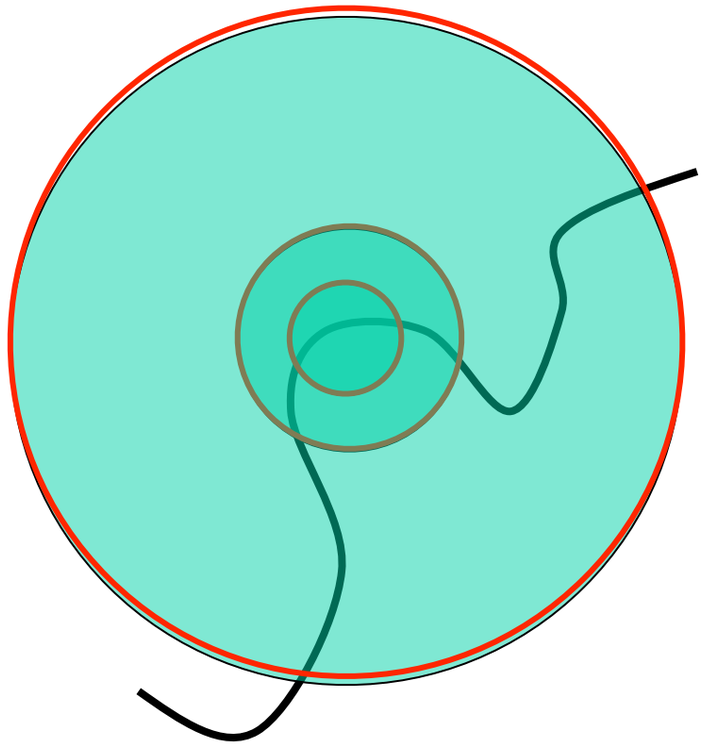




Automatic scale selection

TECHNISCHE UNIVERSITÄT DARMSTADT

- Key idea:



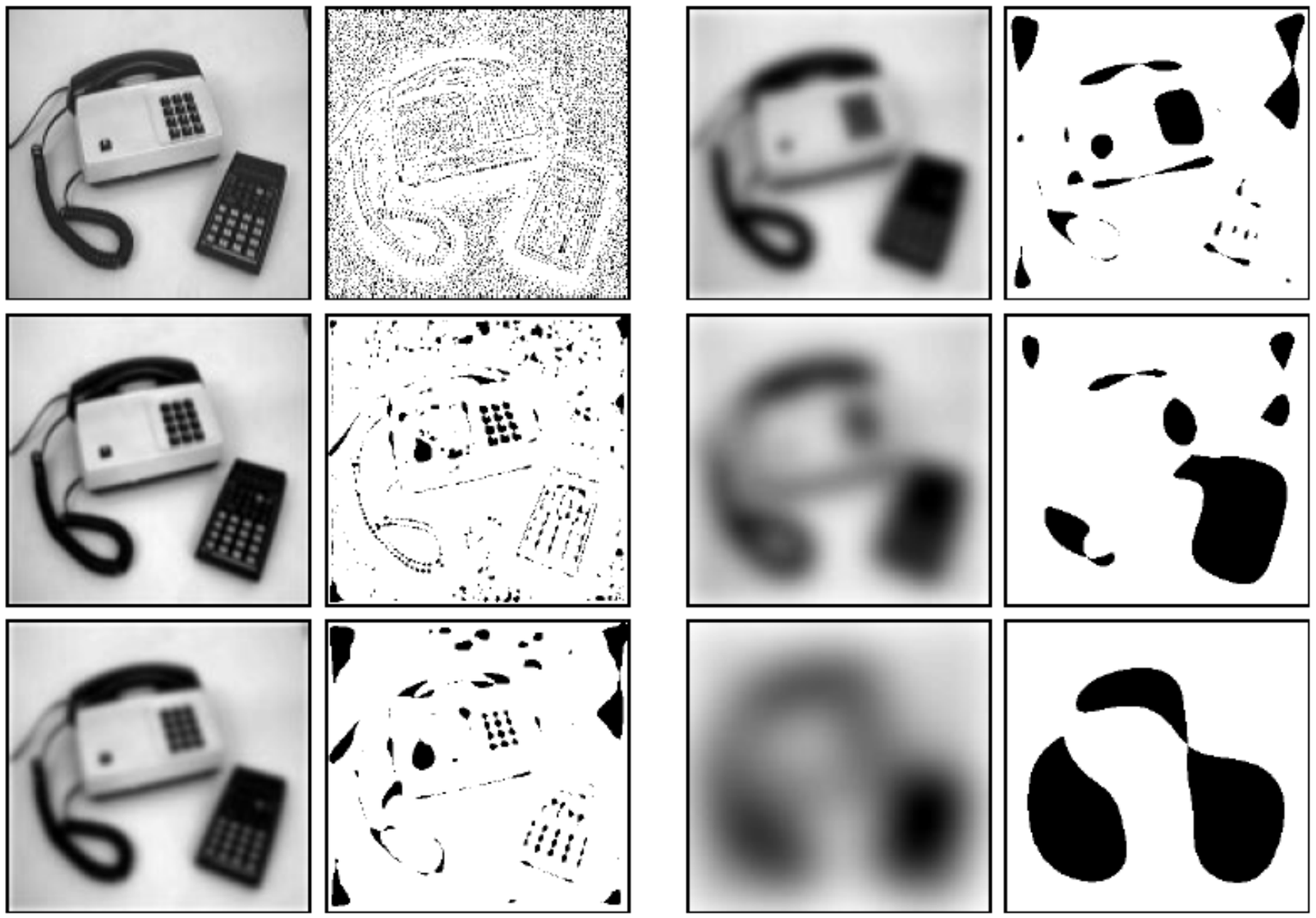
- Find scale that gives local maximum of some criterion.
- Which criterion?

[Seitz & Szeliski]

Scale Space

TECHNISCHE UNIVERSITÄT DARMSTADT

- Level of details decreases monotonically
  - as the scale of Gaussian smoothing is increased



Automatic scale selection

$$I_{r_1 \ldots r_m}(x, \sigma) = I_{r_1 \ldots r_m}(x', \sigma')$$

The same derivative responses if the patch contains the same image up to scale factor

Automatic scale selection

$$f\left(I_{r_1 \ldots r_m}(x, \sigma)\right) = f\left(I_{r_1 \ldots r_m}(x', \sigma')\right)$$

The same operator responses if the patch contains the same image up to scale factor

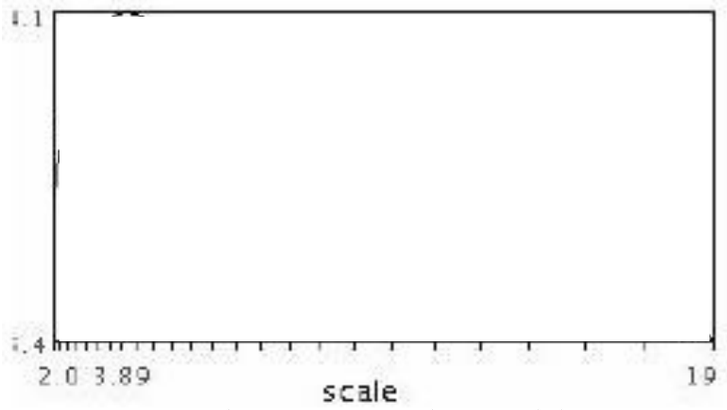
How to find corresponding patch sizes?

Automatic scale selection

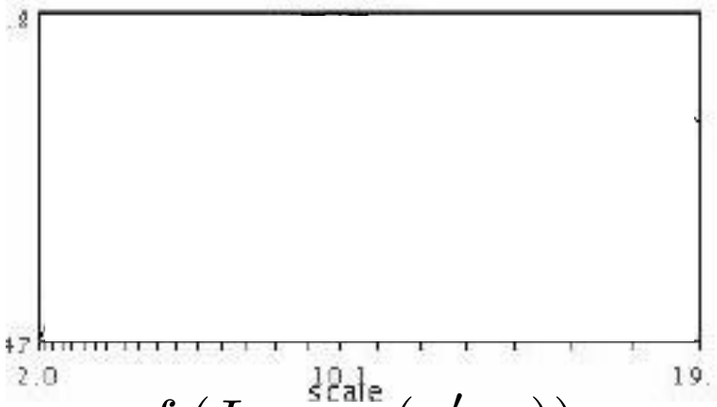
TECHNISCHE UNIVERSITÄT DARMSTADT

Function responses for increasing scale

Scale trace (signature)



$f\left(I_{r_1 \ldots r_m}(x, \sigma)\right)$



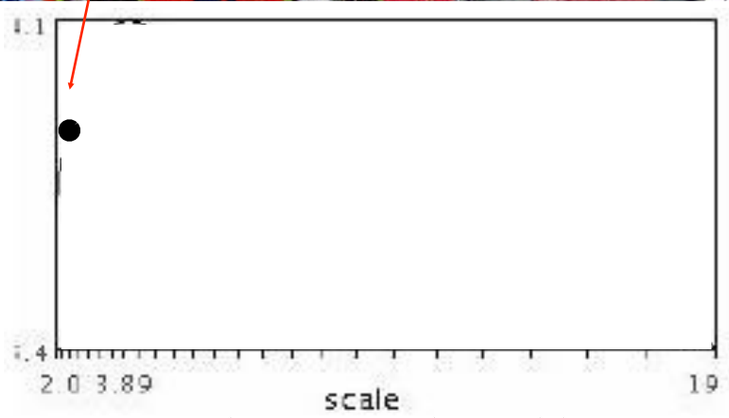
$f\left(I_{r_1 \ldots r_m}(x', \sigma)\right)$

Automatic scale selection

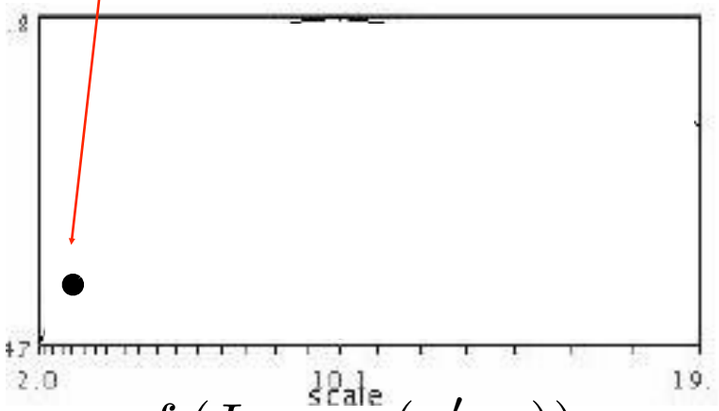
TECHNISCHE UNIVERSITÄT DARMSTADT

Function responses for increasing scale

Scale trace (signature)



$$f\left(I_{r_1 \ldots r_m}(x, \sigma)\right)$$



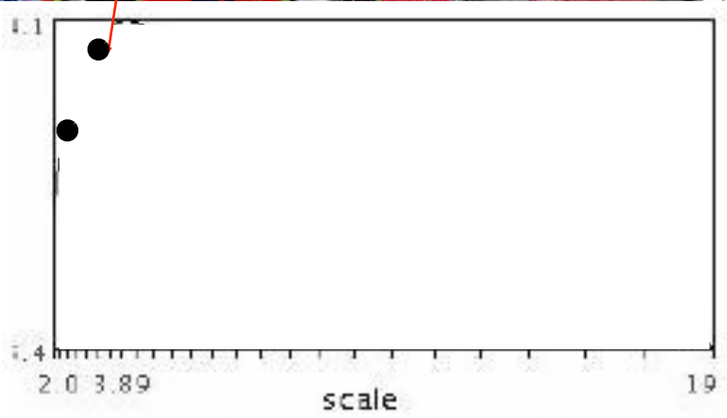
$$f\left(I_{r_1 \ldots r_m}(x', \sigma)\right)$$

Automatic scale selection

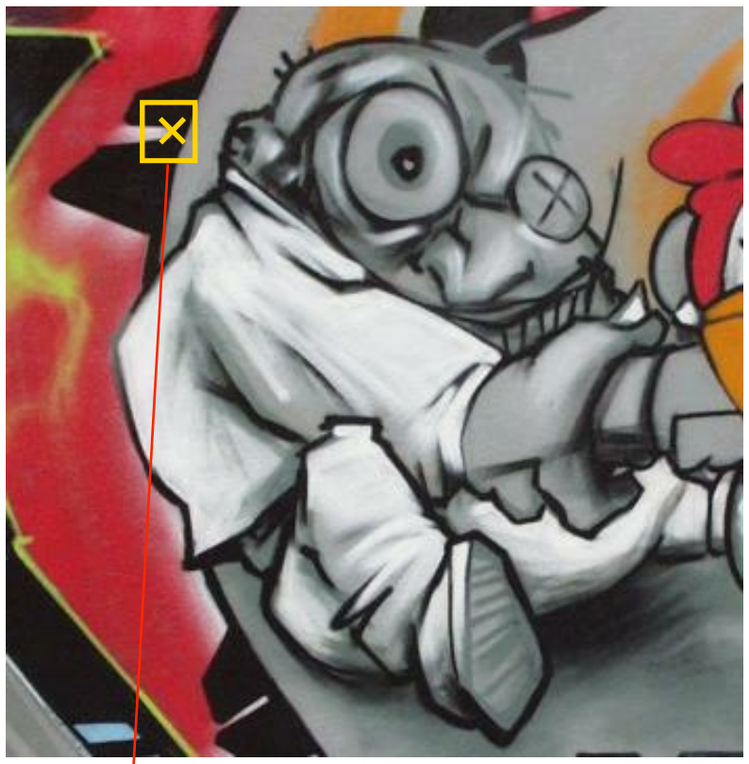
TECHNISCHE UNIVERSITÄT DARMSTADT

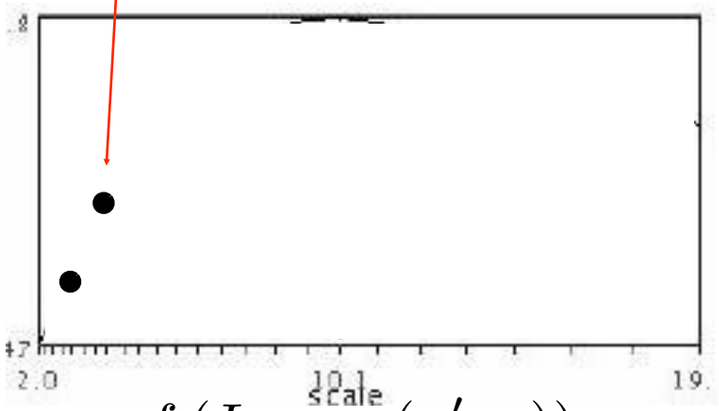
Function responses for increasing scale

Scale trace (signature)



$$f\left(I_{r_1 \ldots r_m}(x, \sigma)\right)$$





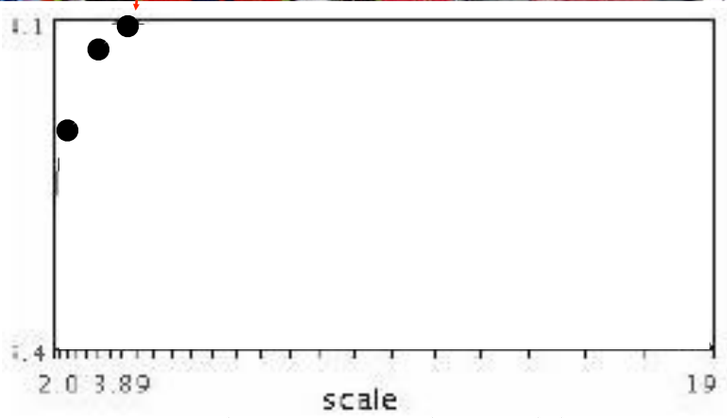
$$f\left(I_{r_1 \ldots r_m}(x', \sigma)\right)$$

Automatic scale selection

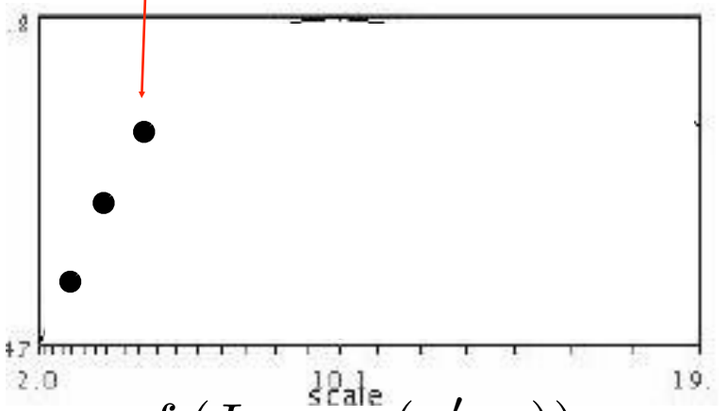
TECHNISCHE UNIVERSITÄT DARMSTADT

Function responses for increasing scale

Scale trace (signature)



$$f\left(I_{r_1 \ldots r_m}(x, \sigma)\right)$$



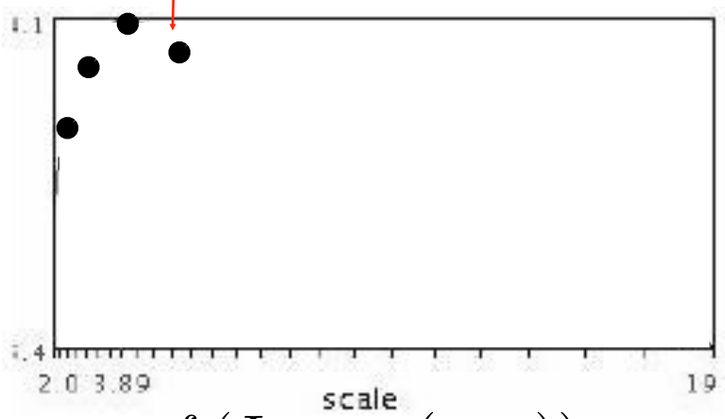
$$f\left(I_{r_1 \ldots r_m}(x', \sigma)\right)$$

Automatic scale selection

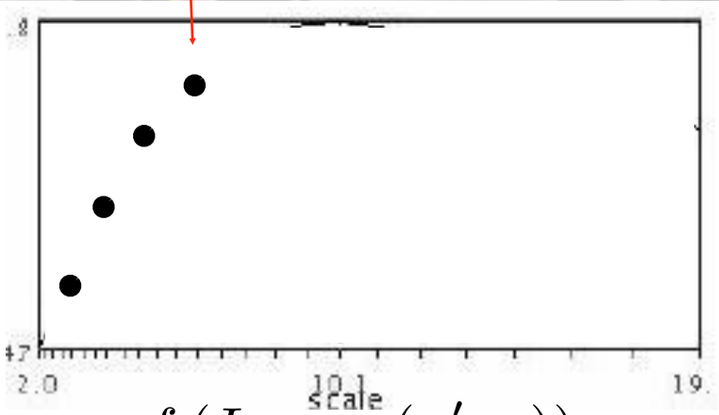
TECHNISCHE UNIVERSITÄT DARMSTADT

Function responses for increasing scale

Scale trace (signature)



$$f\left(I_{r_1 \ldots r_m}(x, \sigma)\right)$$



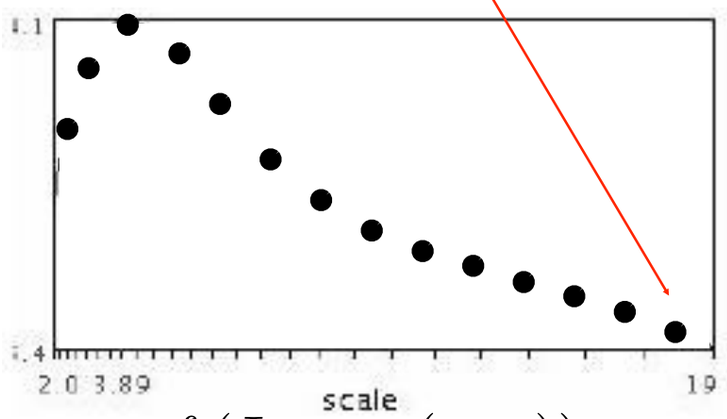
$$f\left(I_{r_1 \ldots r_m}(x', \sigma)\right)$$

Automatic scale selection

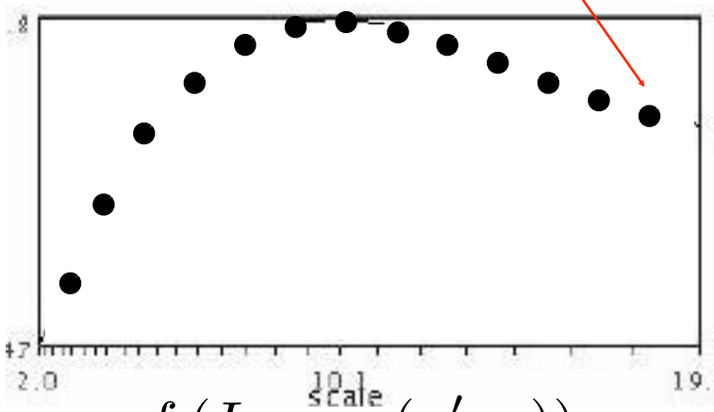
TECHNISCHE UNIVERSITÄT DARMSTADT

Function responses for increasing scale

Scale trace (signature)



$f\left(I_{r_1 \ldots r_m}(x, \sigma)\right)$



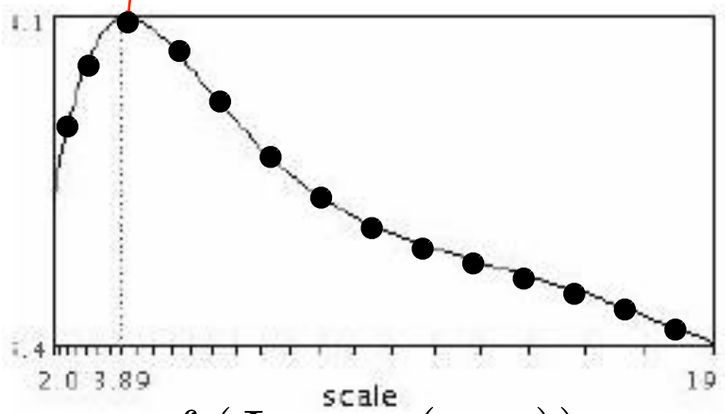
$f\left(I_{r_1 \ldots r_m}(x', \sigma)\right)$

Automatic scale selection

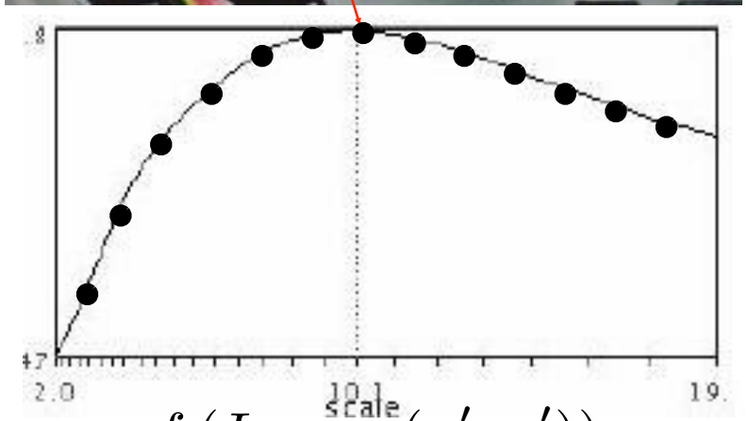
TECHNISCHE UNIVERSITÄT DARMSTADT

Function responses for increasing scale

Scale trace (signature)



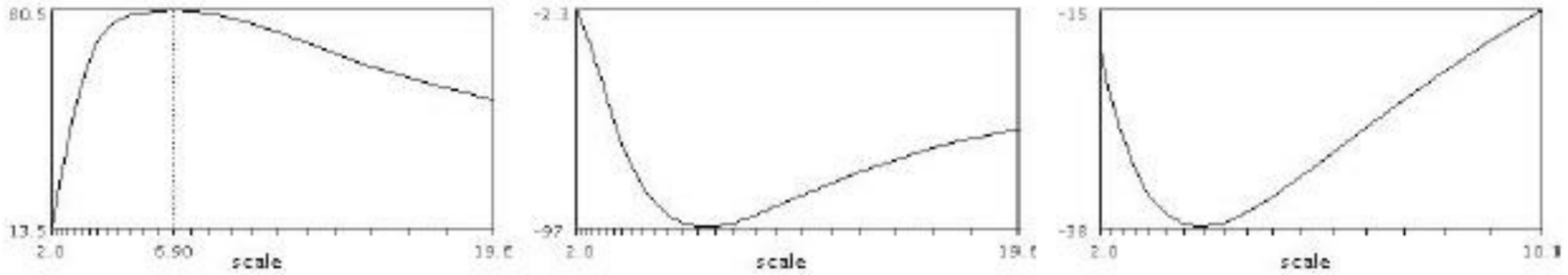
$$f\left(I_{r_1 \ldots r_m}(x, \sigma)\right)$$



$$f\left(I_{r_1 \ldots r_m}(x', \sigma')\right)$$

Automatic scale selection

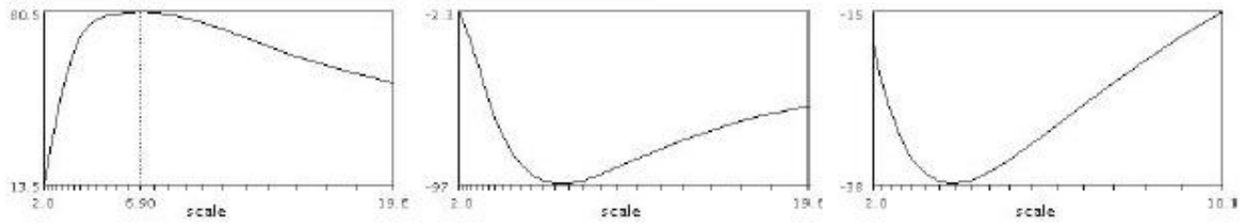
$$\text{Laplacian} = \text{trace}(\mathcal{H})$$



$$I_{xx} + I_{yy}$$

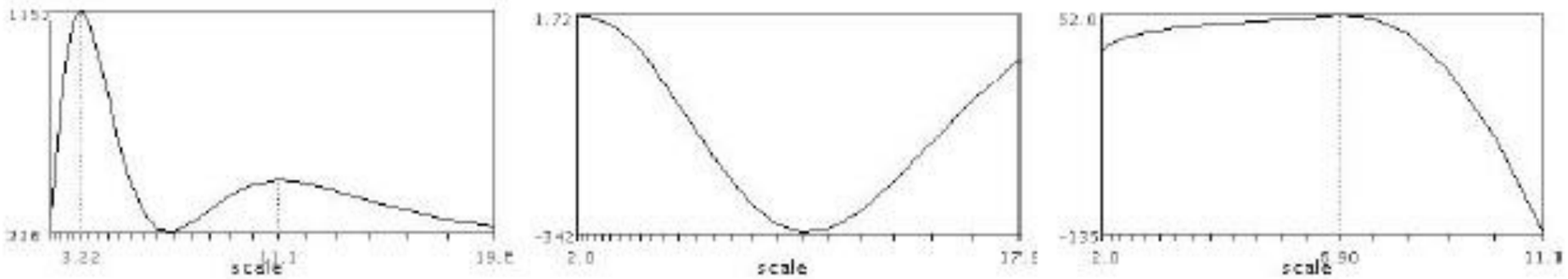
Automatic scale selection

Laplacian $= \text{trace}(\mathcal{H})$



$$I_{xx} + I_{yy}$$

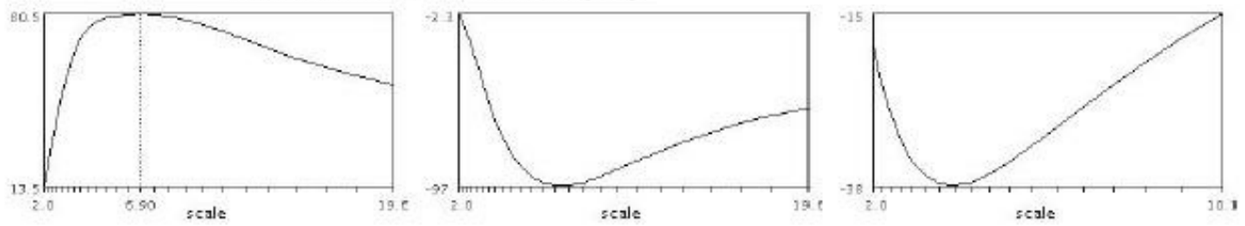
$\det(\mathcal{H})$



$$I_{xx}I_{yy} - I_{xy}^2$$

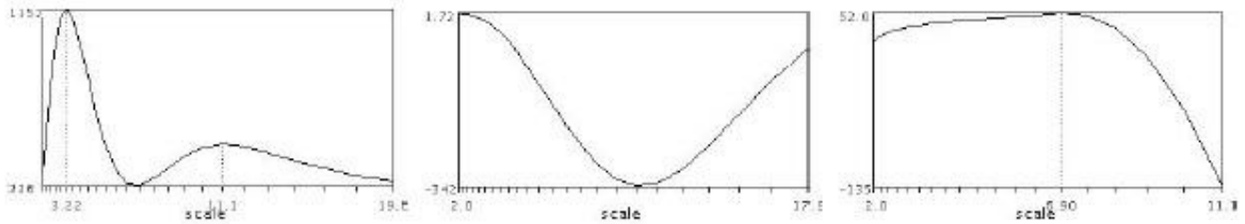
Automatic scale selection

Laplacian $= \text{trace}(\mathcal{H})$



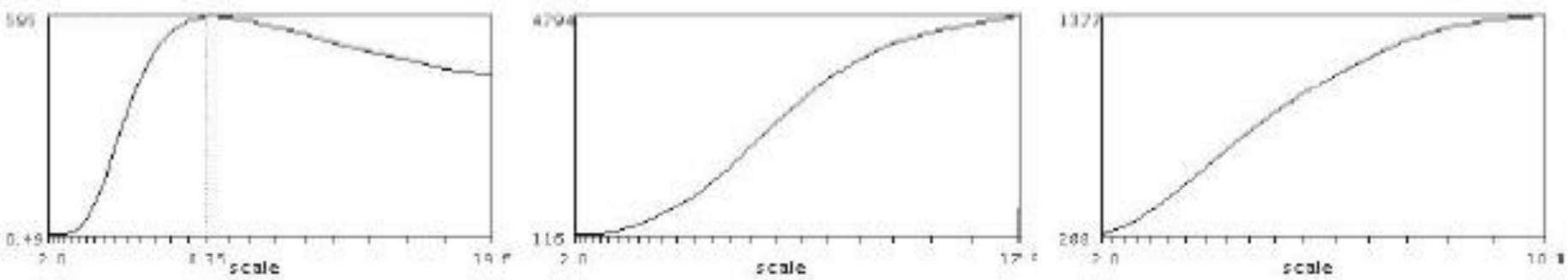
$$I_{xx} + I_{yy}$$

$\det(\mathcal{H})$



$$I_{xx}I_{yy} - I_{xy}^2$$

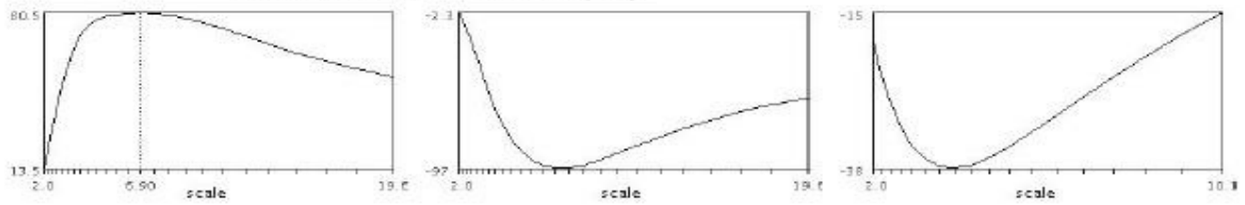
Squared gradient



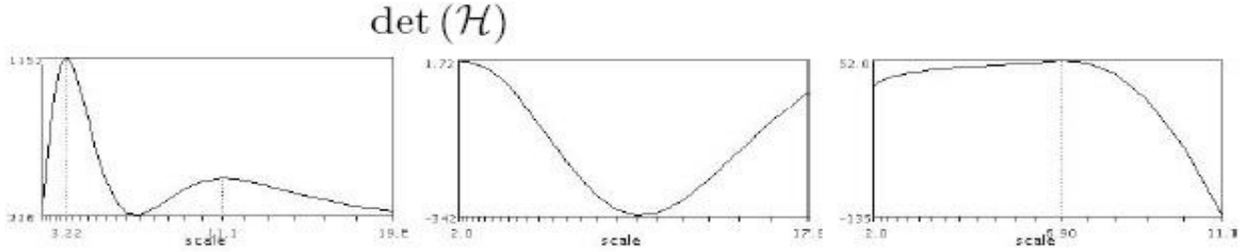
$$I_x^2 + I_y^2$$

Automatic scale selection

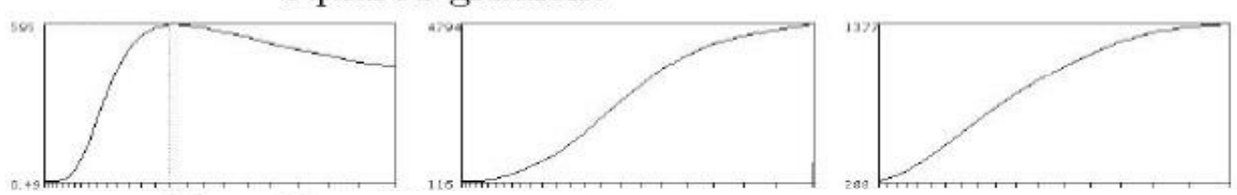
Laplacian $= \text{trace}(\mathcal{H})$



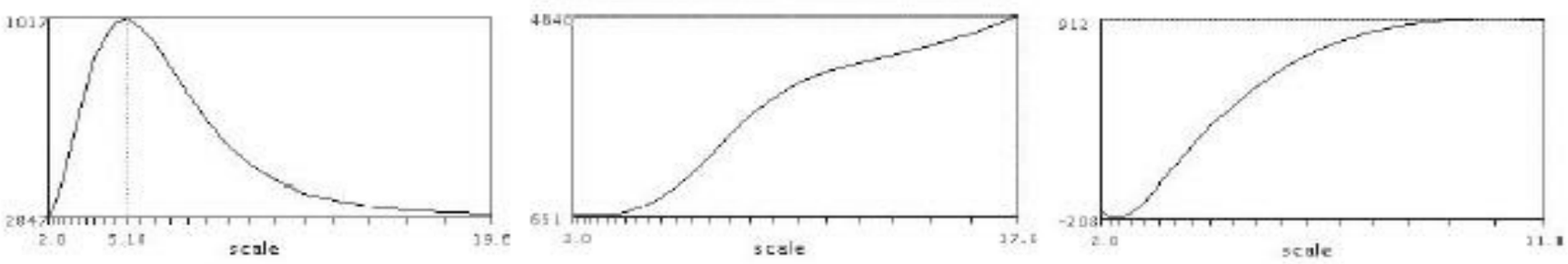
$\det(\mathcal{H})$



Squared gradient



Harris measure



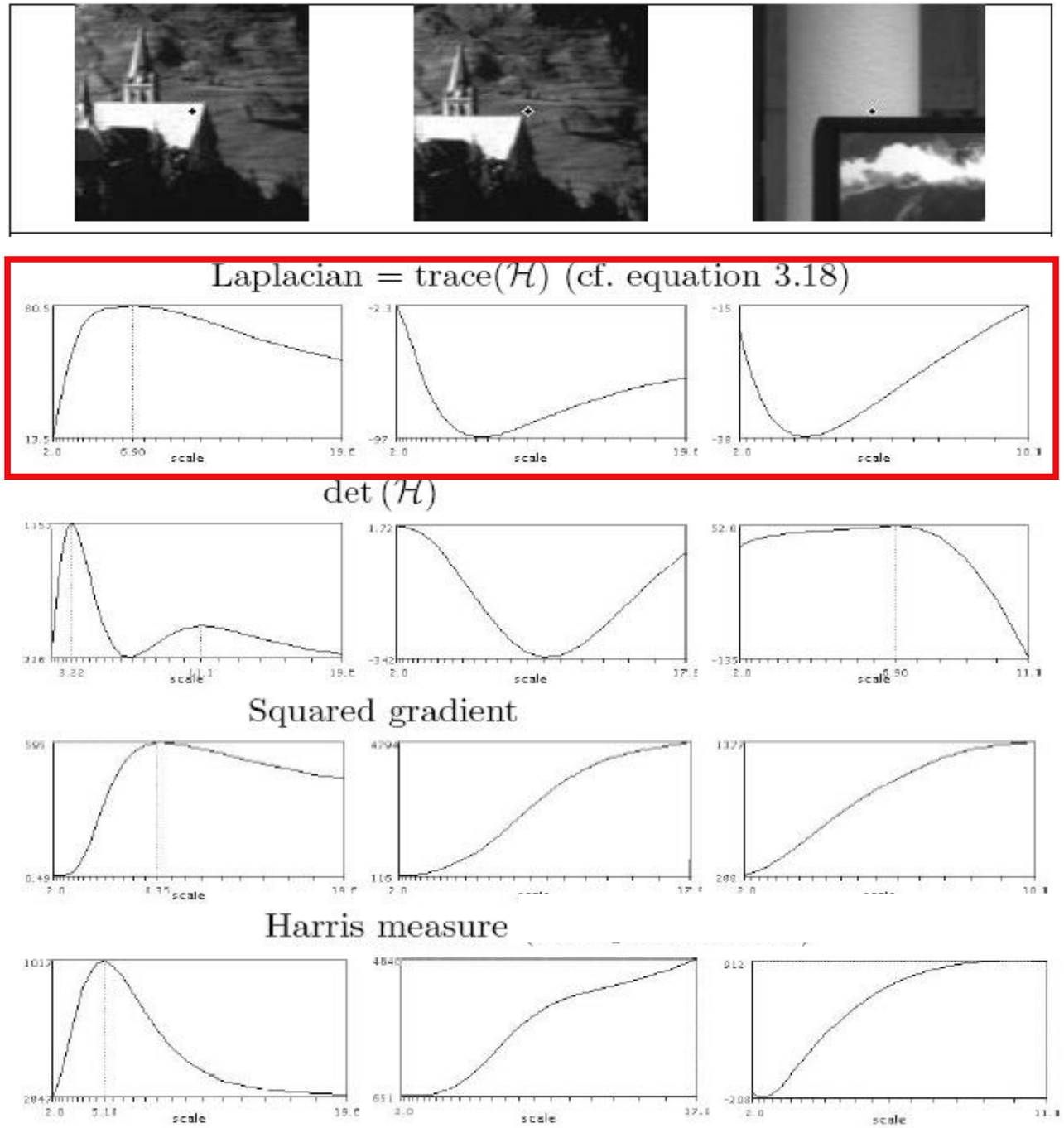
$$I_{xx} + I_{yy}$$

$$I_{xx}I_{yy} - I_{xy}^2$$

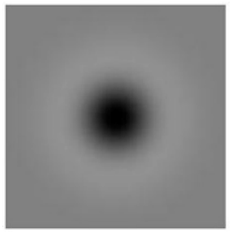
$$I_x^2 + I_y^2$$

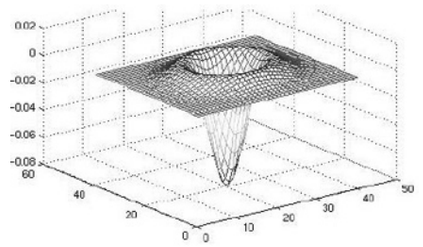
$$\det(H) - \alpha^2 \text{trace}(H)$$

Automatic scale selection



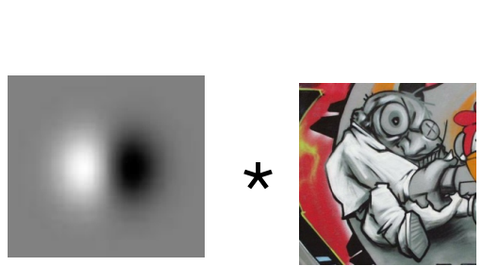
$$I_{xx} + I_{yy}$$

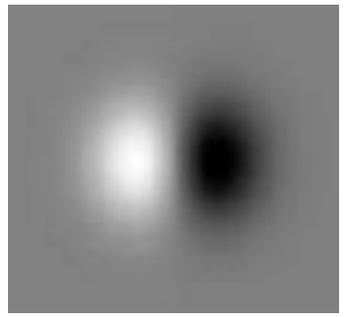




Scale invariance – normalization

- The amplitude of spatial derivatives, in general, decreases with scale, due to the response being smoother at a larger scale
  - In the case of the structures present at a large range of scales, like a corner or a step-edge, we would like to have the derivative constant over scale.





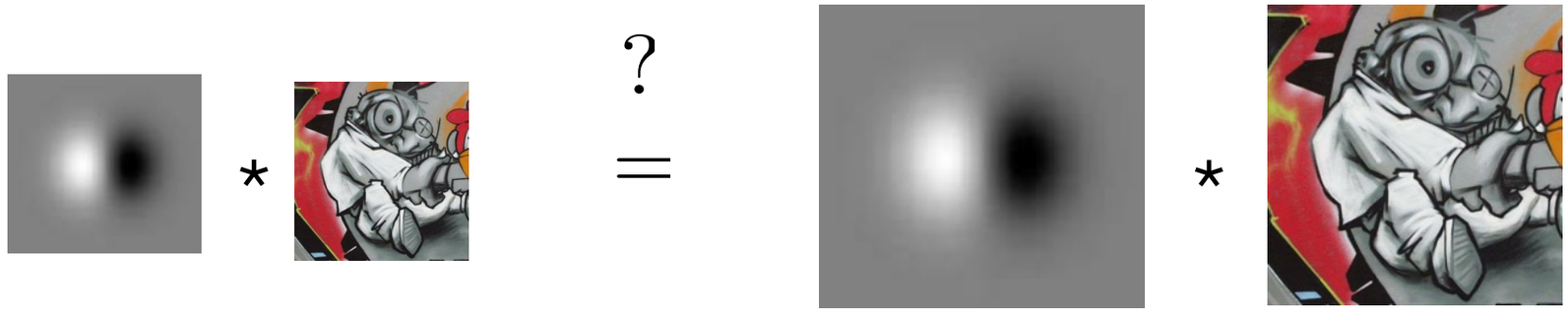

$$x' = sx \qquad \sigma \qquad I(x) \qquad \sigma' = s\sigma \qquad I(x') = I(sx)$$

- then:

$$\frac{\partial}{\partial x} I(x') = \frac{\partial}{\partial x} I(sx) = s \frac{\partial}{\partial x} I(x)$$

Scale invariance – normalization

- The amplitude of spatial derivatives, in general, decreases with scale, due to the response being smoother at a larger scale
  - In the case of the structures present at a large range of scales, like a corner or a step-edge, we would like to have the derivative constant over scale.



$\sigma \qquad I(x) \qquad \stackrel{?}{=} \qquad \sigma' = s\sigma \qquad I(x') = I(sx)$

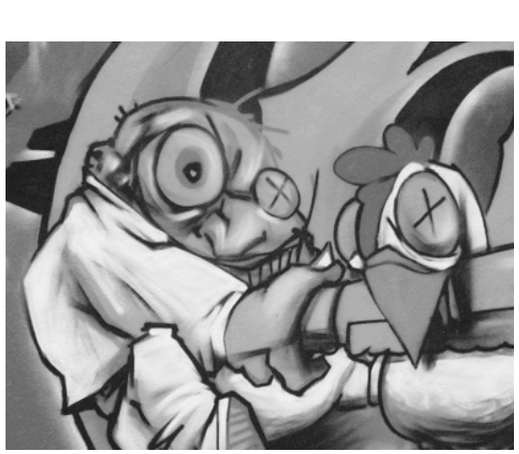
- $$I_{r_1 \ldots r_m}(x') = \boxed{s^m} I_{r_1 \ldots r_m}(sx)$$
  - In order to maintain the property of scale invariance the derivative function must be normalized with respect to the scale of derivation

$$I_{r_1 \ldots r_m}(x') = \boxed{s^m} g_{r_1 \ldots r_m}(\boxed{s\sigma}) * I(sx) = s^m g_{r_1 \ldots r_m}(\sigma') * I(x')$$

Laplacian of Gaussian (LoG)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Local optima in scale space of Laplacian of Gaussian

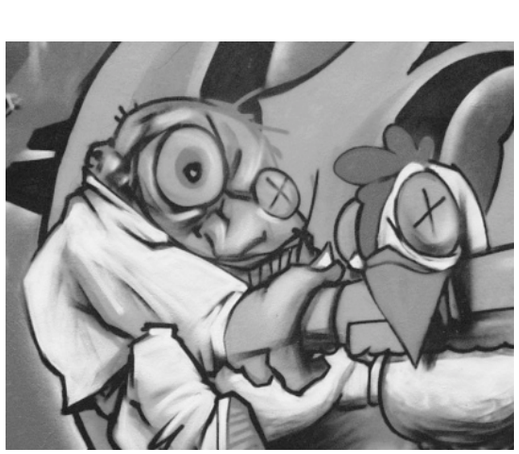


$$I_{xx}(\sigma) + I_{yy}(\sigma)$$

Laplacian of Gaussian (LoG)

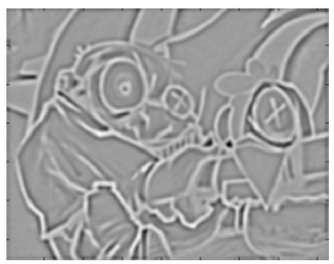
TECHNISCHE UNIVERSITÄT DARMSTADT

- Local optima in scale space of Laplacian of Gaussian



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

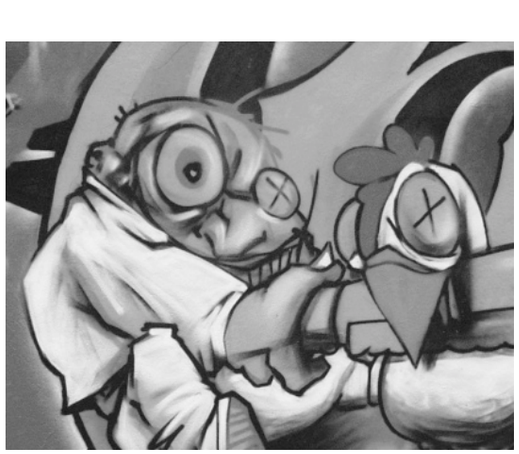
σ



Laplacian of Gaussian (LoG)

TECHNISCHE UNIVERSITÄT DARMSTADT

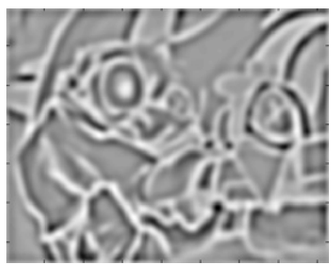
- Local optima in scale space of Laplacian of Gaussian

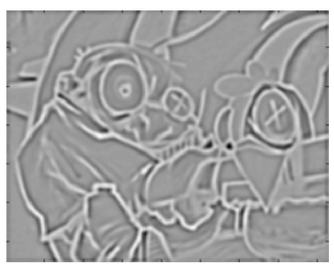


$$I_{xx}(\sigma) + I_{yy}(\sigma)$$

σ^2

σ



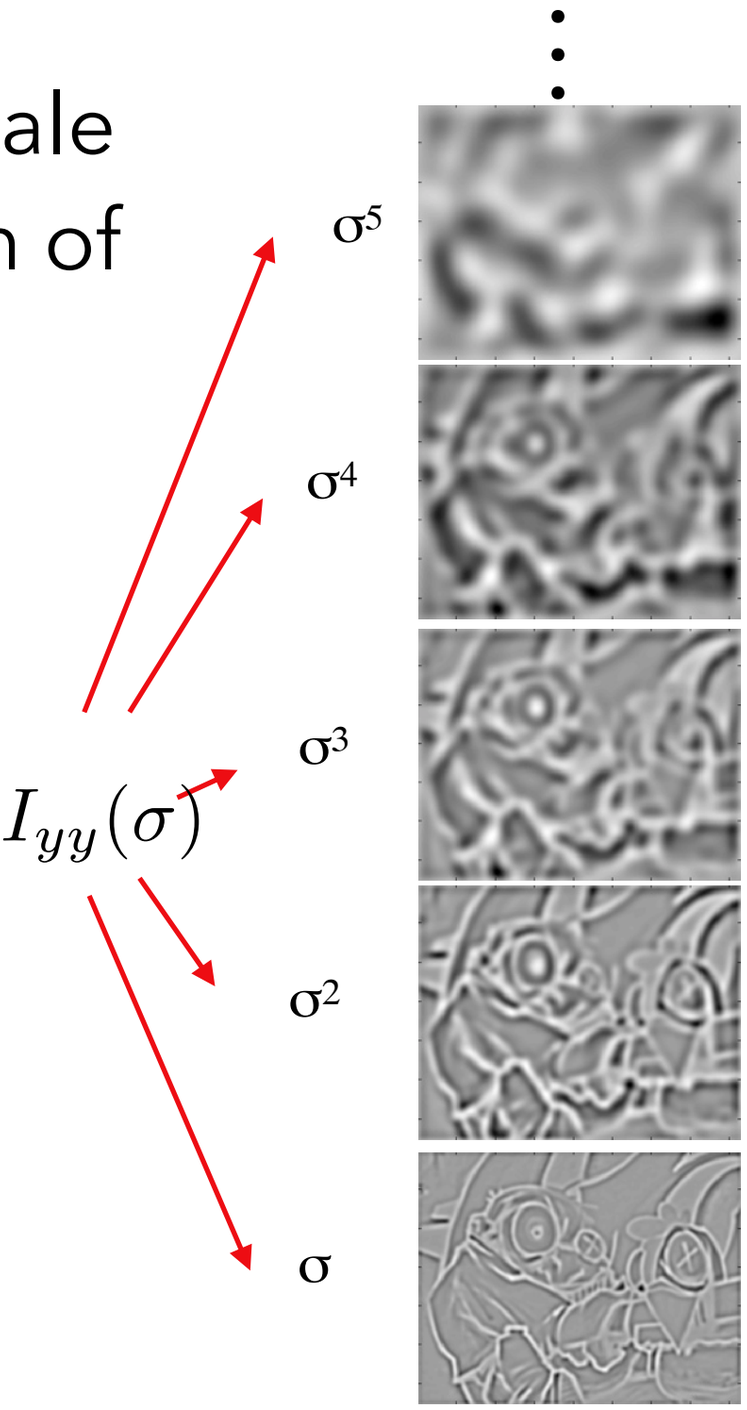


Laplacian of Gaussian (LoG)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Local optima in scale space of Laplacian of Gaussian

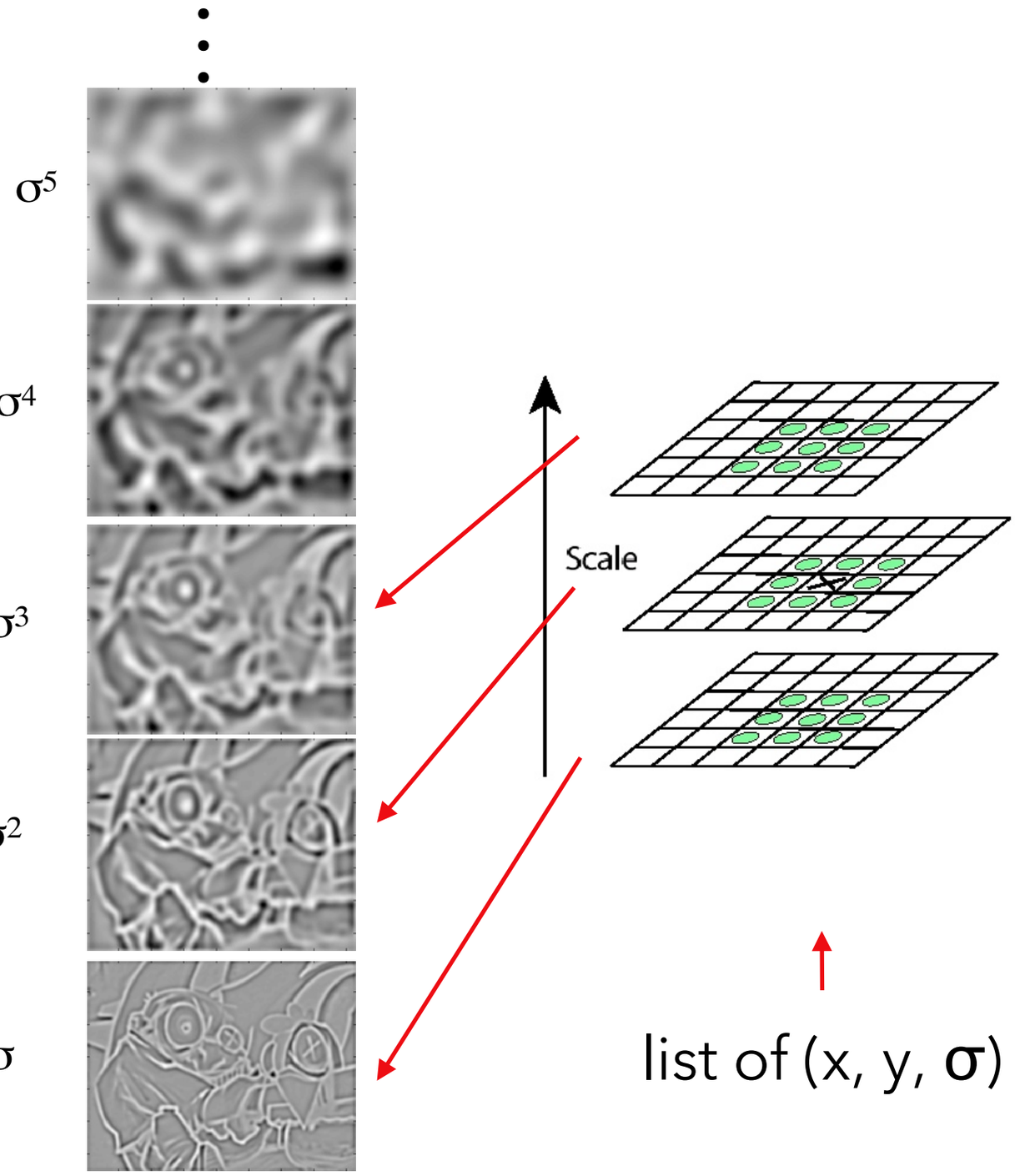
$$I_{xx}(\sigma) + I_{yy}(\sigma)$$



Laplacian of Gaussian (LoG)

- Local optima in scale space of Laplacian of Gaussian

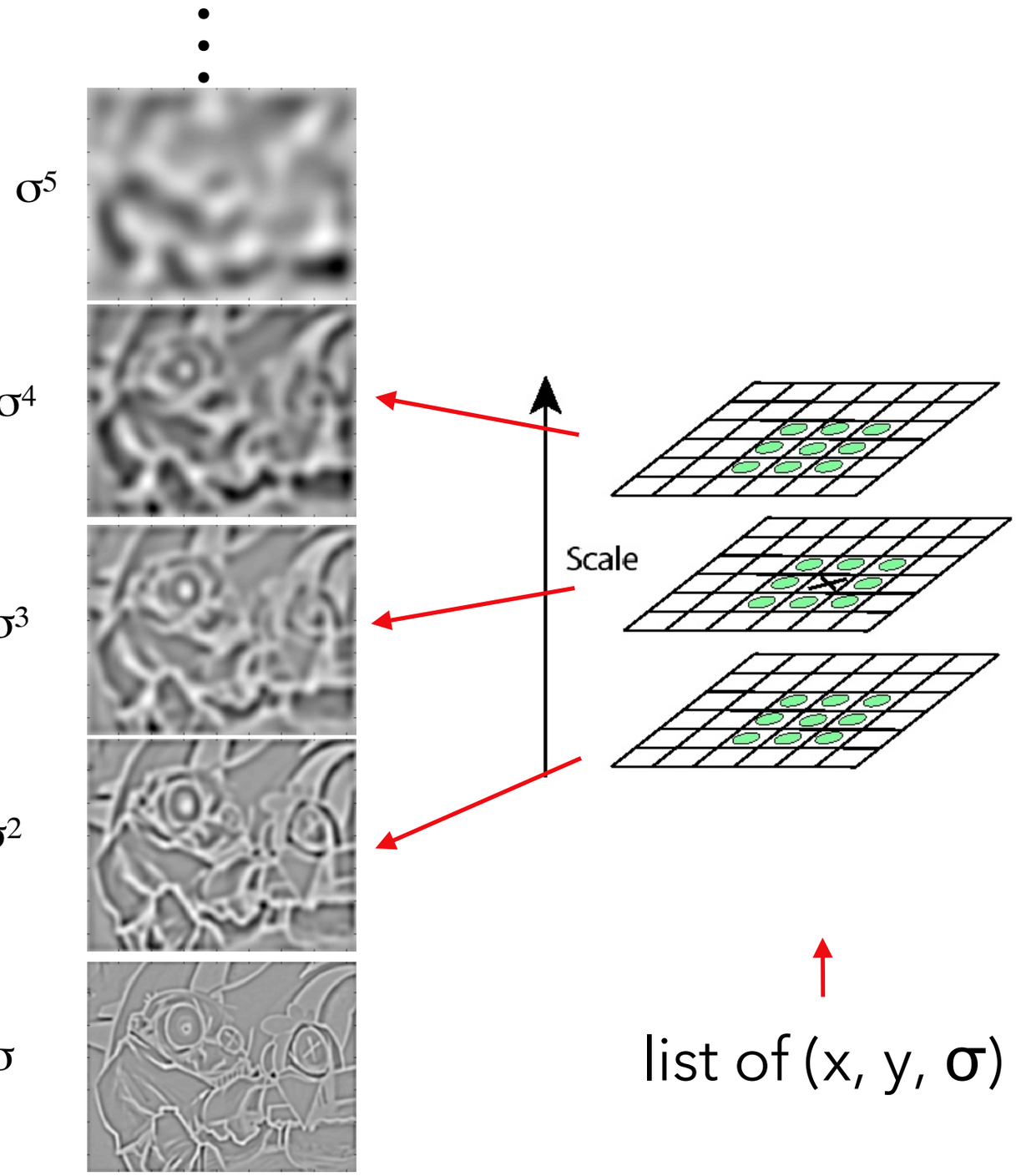
$$I_{xx}(\sigma) + I_{yy}(\sigma)$$



Laplacian of Gaussian (LoG)

- Local optima in scale space of Laplacian of Gaussian

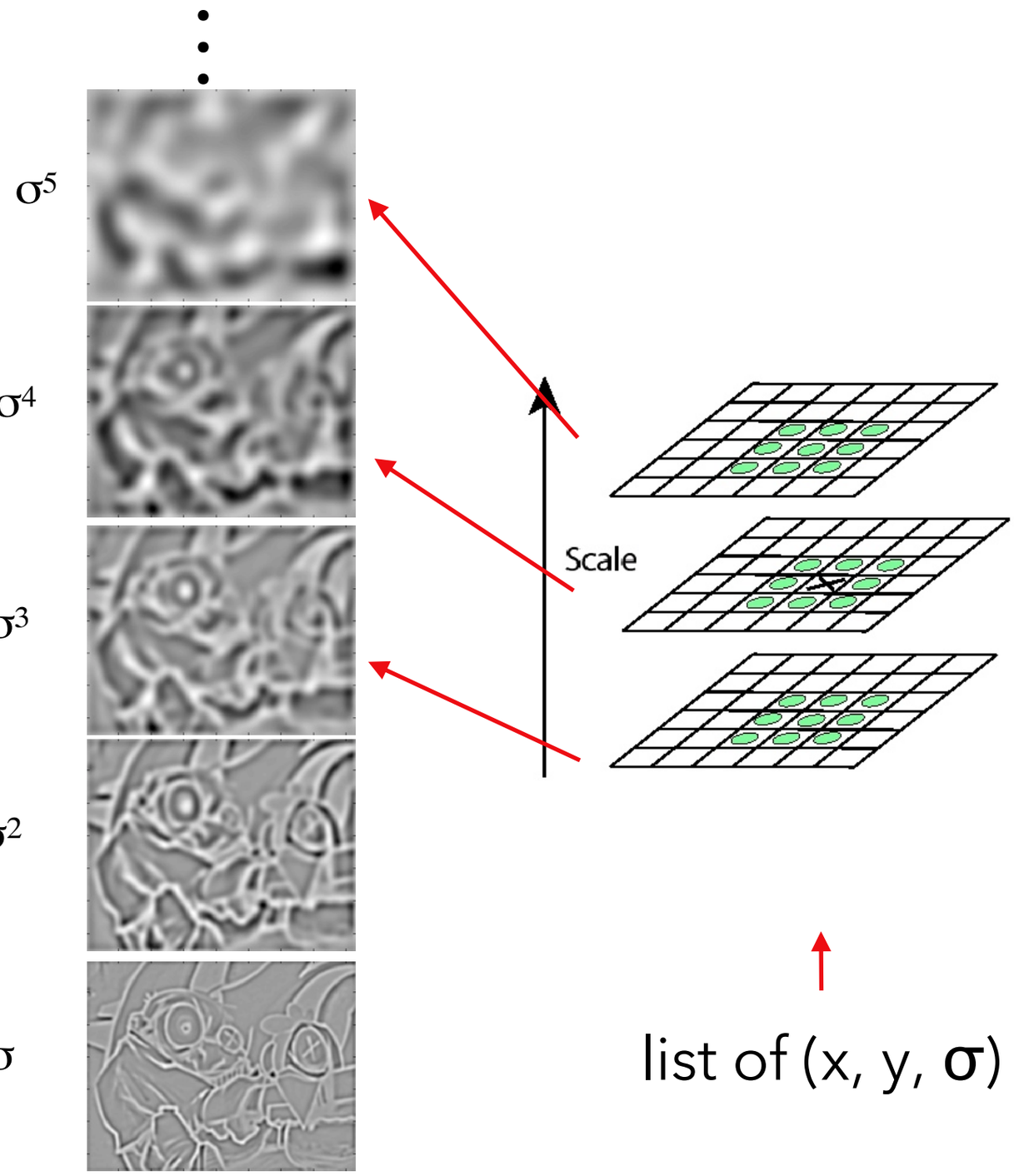
$$I_{xx}(\sigma) + I_{yy}(\sigma)$$



Laplacian of Gaussian (LoG)

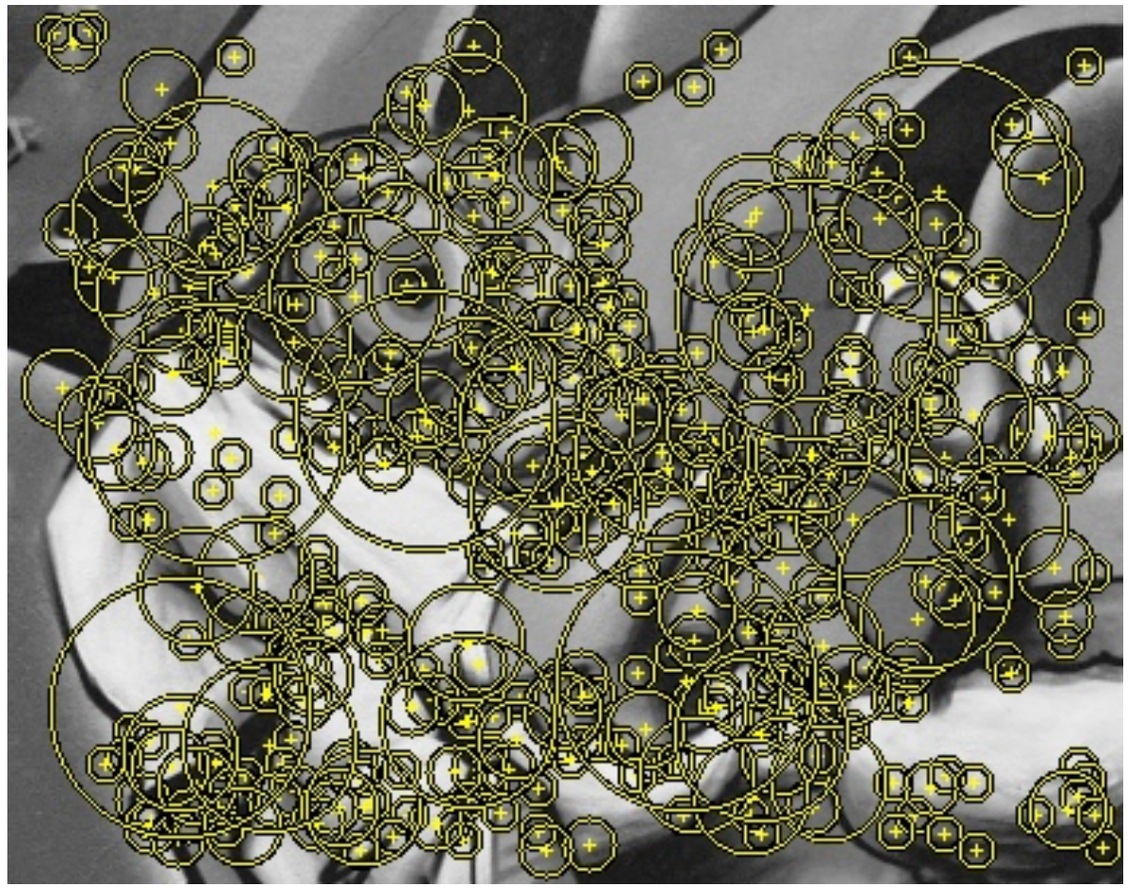
- Local optima in scale space of Laplacian of Gaussian

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



Laplacian of Gaussian (LoG)

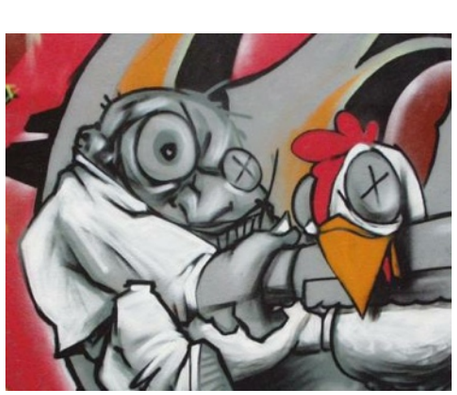
TECHNISCHE UNIVERSITÄT DARMSTADT



Harris-Laplace (HarLap)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Detecting multiscale Harris points



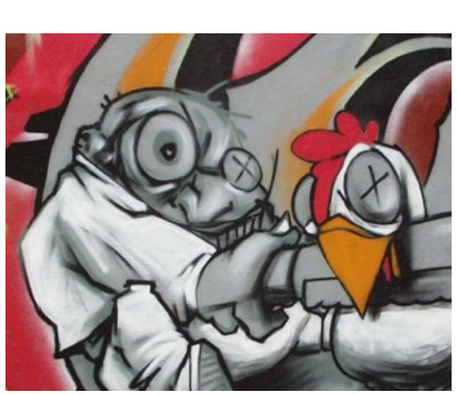
$$H = \sum_{(x,y)\in R} w(x,y)\nabla I(x,y)\nabla I(x,y)^T$$

$$f = \det(H) - \alpha^2 \mathrm{trace}(H)$$

Harris-Laplace (HarLap)

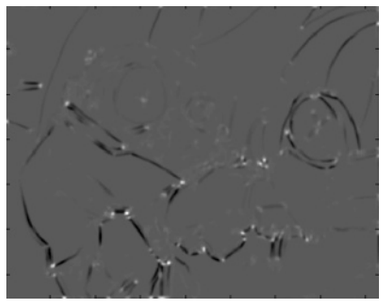
TECHNISCHE UNIVERSITÄT DARMSTADT

- Detecting multiscale Harris points



$$H = \sum_{(x,y)\in R} w(x,y)\nabla I(x,y)\nabla I(x,y)^T$$

$$f = \det(H) - \alpha^2 \mathrm{trace}(H)$$



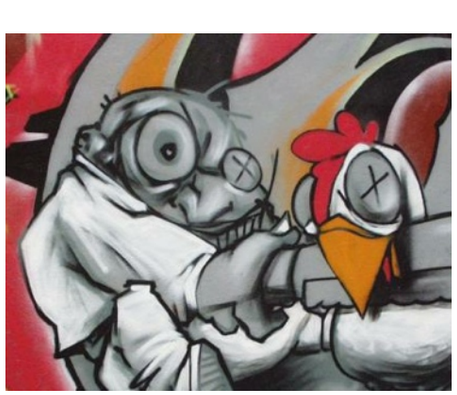
σ

Computing Harris function

Harris-Laplace (HarLap)

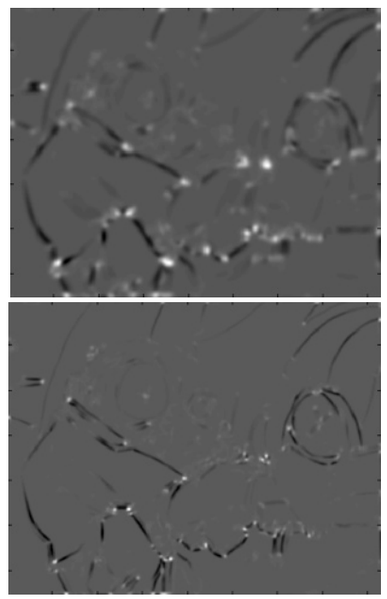
TECHNISCHE UNIVERSITÄT DARMSTADT

- Detecting multiscale Harris points



$$H = \sum_{(x,y)\in R} w(x,y)\nabla I(x,y)\nabla I(x,y)^T$$

$$f = \det(H) - \alpha^2 \mathrm{trace}(H)$$



Computing Harris function

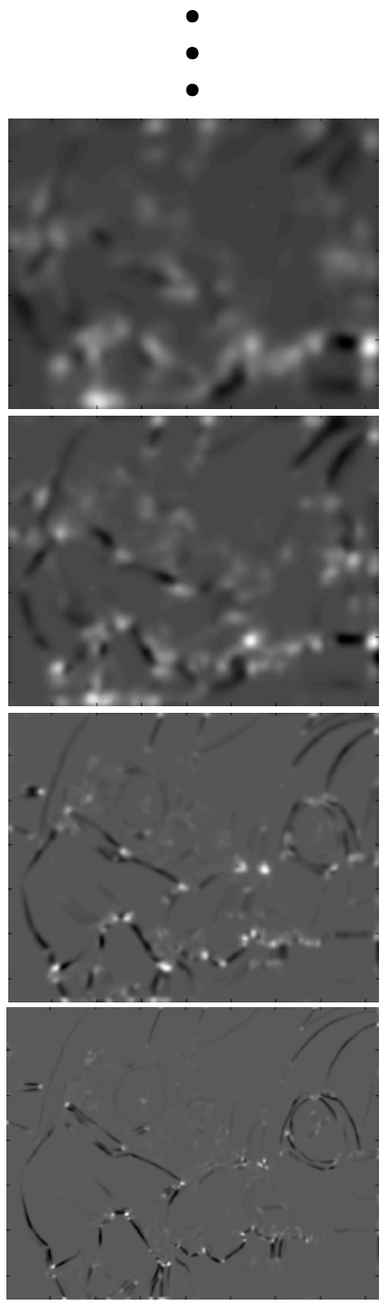
Harris-Laplace (HarLap)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Detecting multiscale Harris points – thousands of interest points

$$H = \sum_{(x,y)\in R} w(x,y)\nabla I(x,y)\nabla I(x,y)^T$$

$$f = \det(H) - \alpha^2 \mathrm{trace}(H)$$

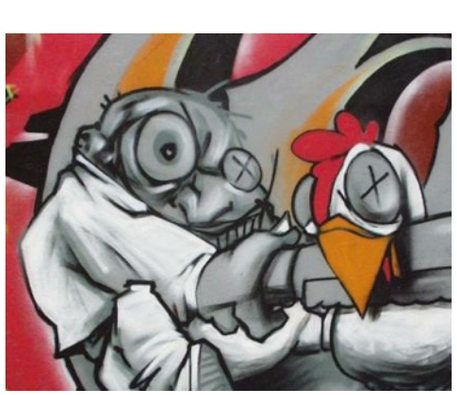


Computing Harris function

Harris-Laplace (HarLap)

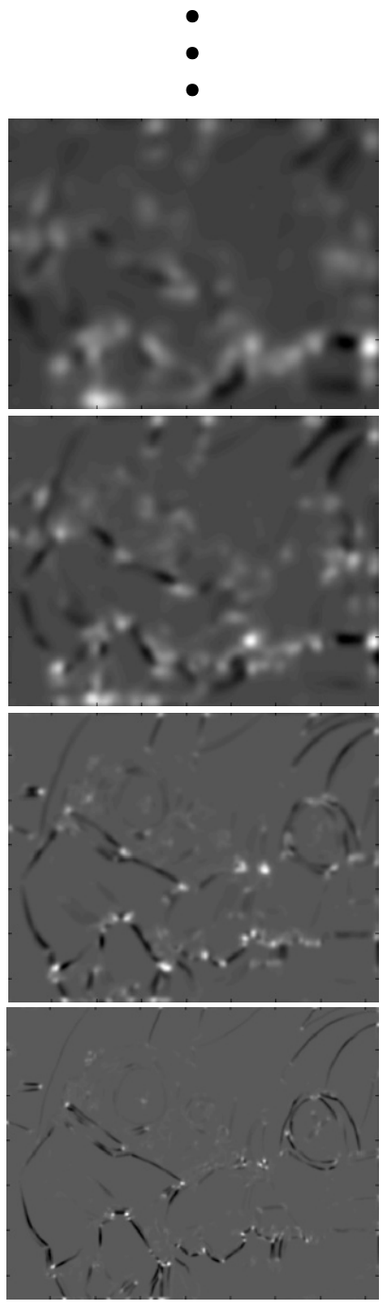
TECHNISCHE UNIVERSITÄT DARMSTADT

- Detecting multiscale Harris points – thousands of interest points



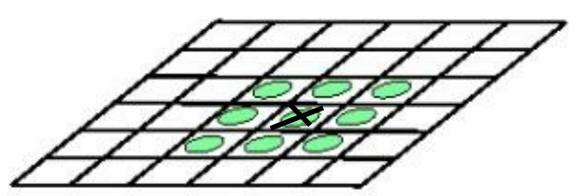
$$H = \sum_{(x,y)\in R} w(x,y)\nabla I(x,y)\nabla I(x,y)^T$$

$$f = \det(H) - \alpha^2 \mathrm{trace}(H)$$



Computing Harris function

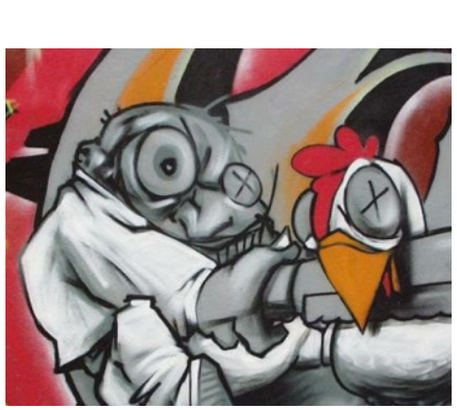
list of
(x, y, σ)



Detecting local maxima

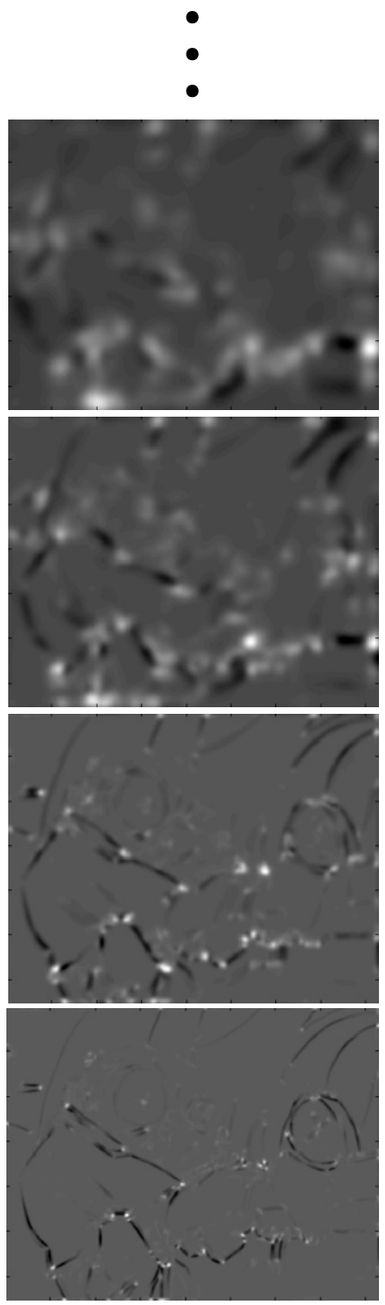
Harris-Laplace (HarLap)

- Detecting multiscale Harris points – thousands of interest points



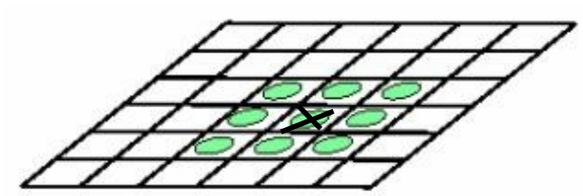
$$H = \sum_{(x,y)\in R} w(x,y)\nabla I(x,y)\nabla I(x,y)^T$$

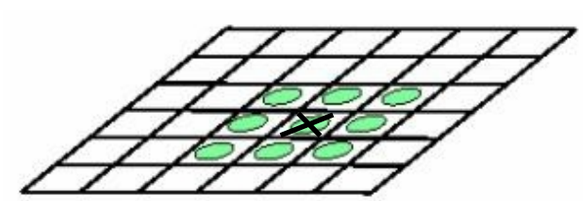
$$f = \det(H) - \alpha^2 \mathrm{trace}(H)$$



Computing Harris function

list of
(x, y, σ)

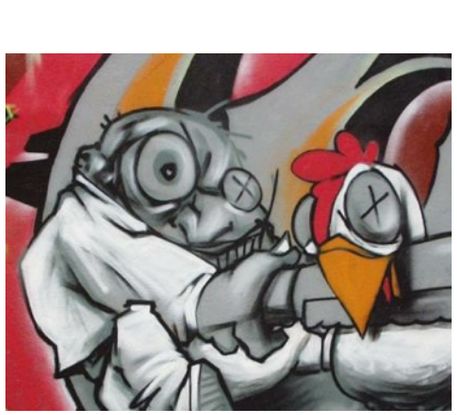




Detecting local maxima

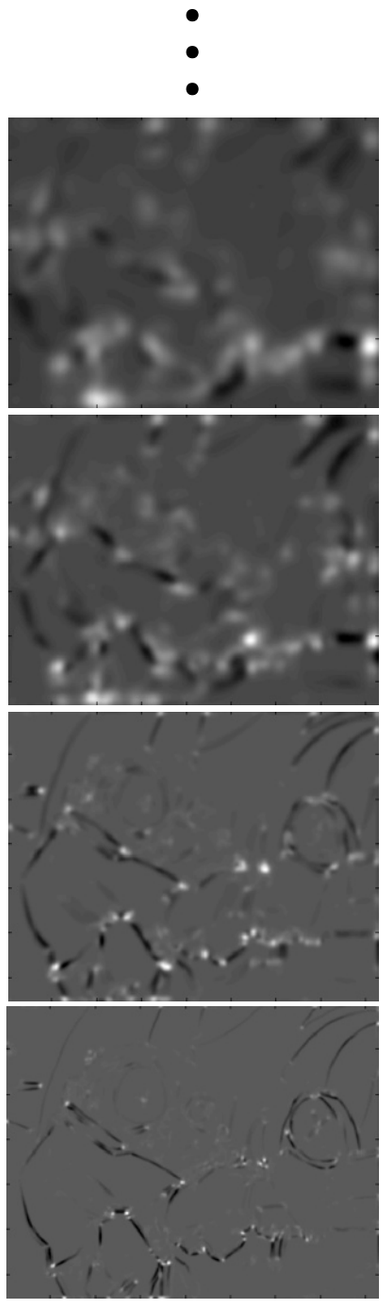
Harris-Laplace (HarLap)

- Detecting multiscale Harris points – thousands of interest points

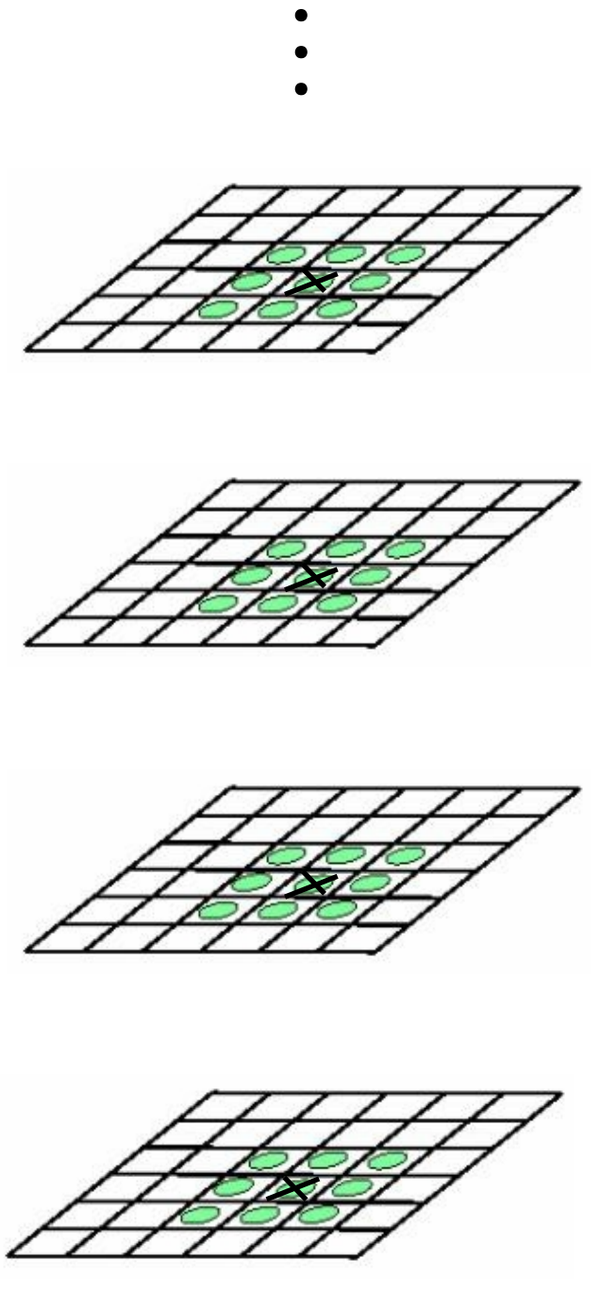


$$H = \sum_{(x,y)\in R} w(x,y)\nabla I(x,y)\nabla I(x,y)^T$$

$$f = \det(H) - \alpha^2 \mathrm{trace}(H)$$



Computing Harris function



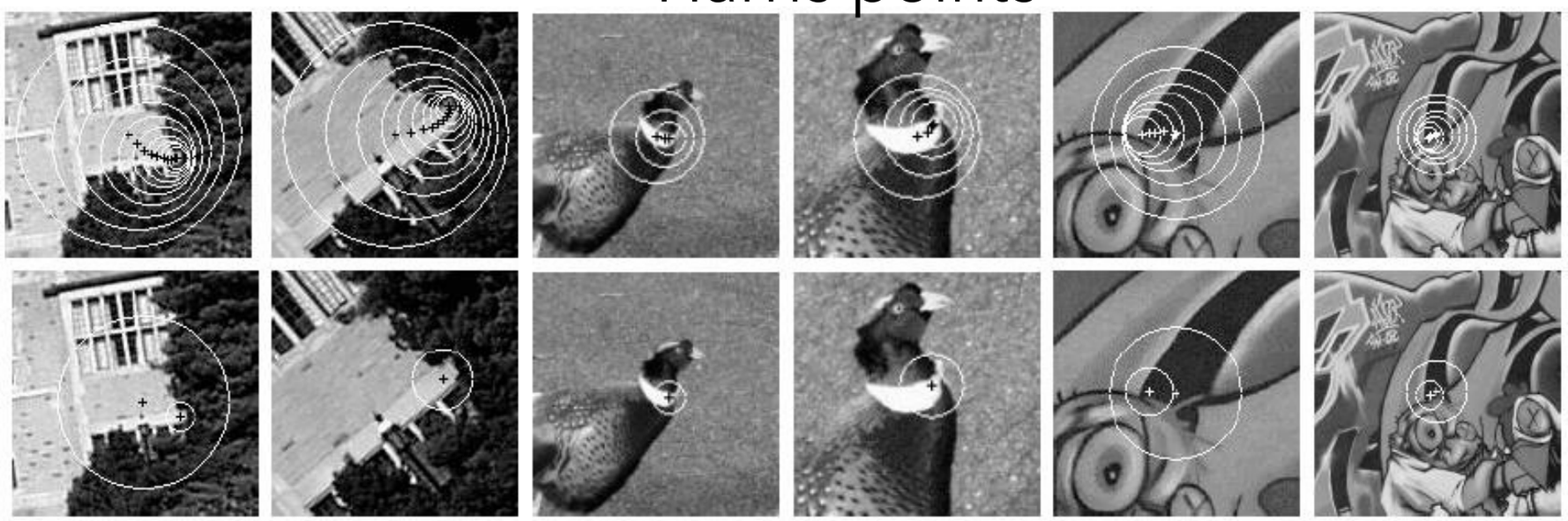
Detecting local maxima

Harris-Laplace (HarLap)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Detecting multiscale Harris points
- Selecting points which maximize the Laplacian
  - Automatic scale selection
    - Given a point (x, y, σ_n)
    - If $(L_{xx}(\sigma_n) + L_{yy}(\sigma_n)) > (L_{xx}(\sigma_{n-1}) + L_{yy}(\sigma_{n-1}))$ and
      $(L_{xx}(\sigma_n) + L_{yy}(\sigma_n)) > (L_{xx}(\sigma_{n+1}) + L_{yy}(\sigma_{n+1}))$ keep the point,
      otherwise reject

Harris points

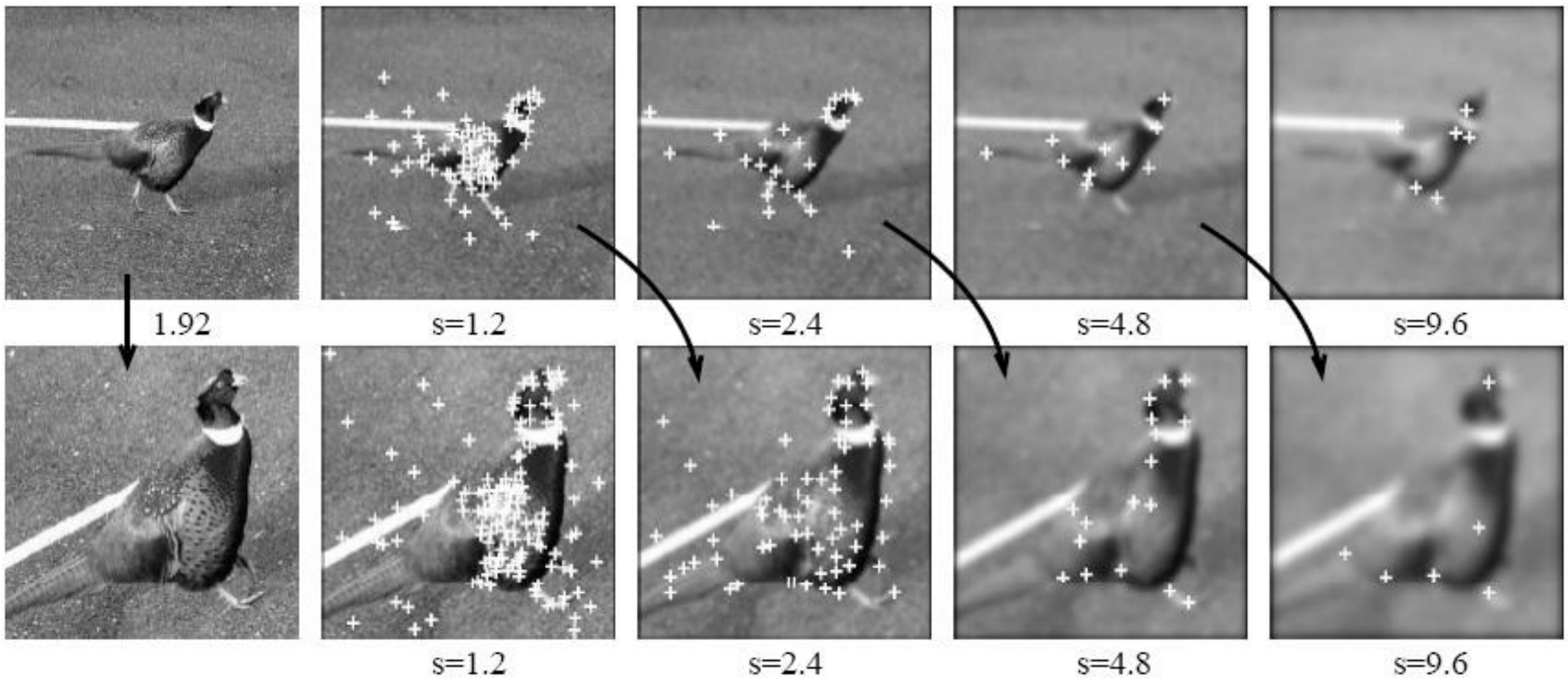


Harris-Laplace points

Harris-Laplace (HarLap)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Detecting multiscale Harris points
- Selecting Harris points which maximize the Laplacian
- Automatic scale selection

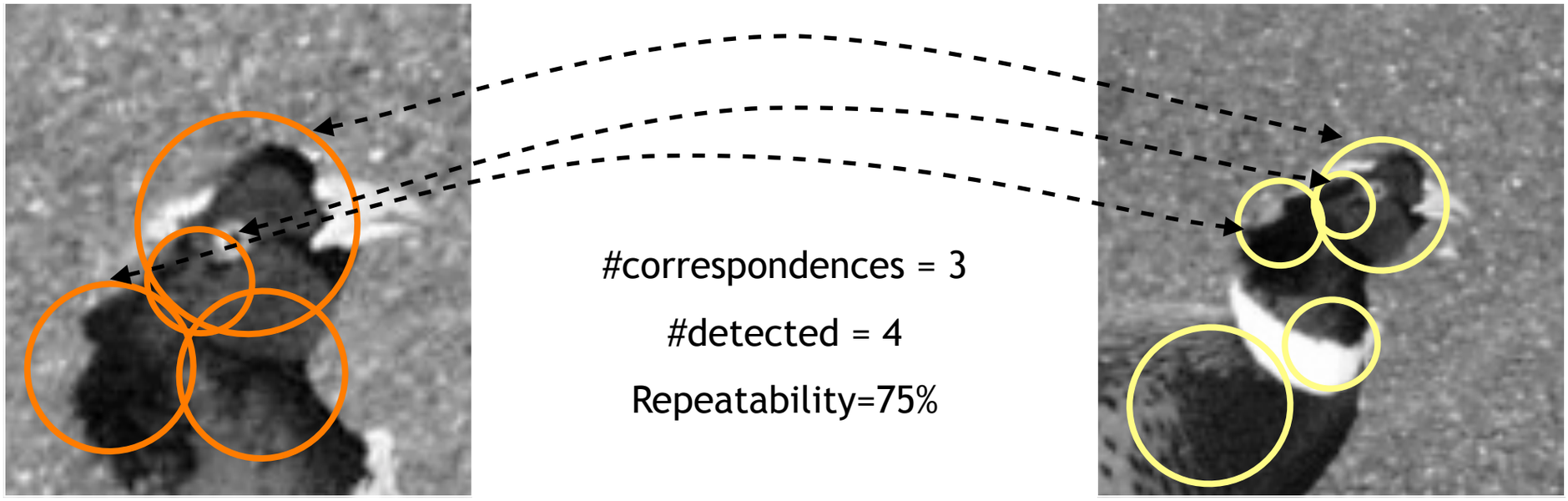


Harris-Laplace points

Evaluation criteria

TECHNISCHE UNIVERSITÄT DARMSTADT

- Repeatability rate : percentage of corresponding points

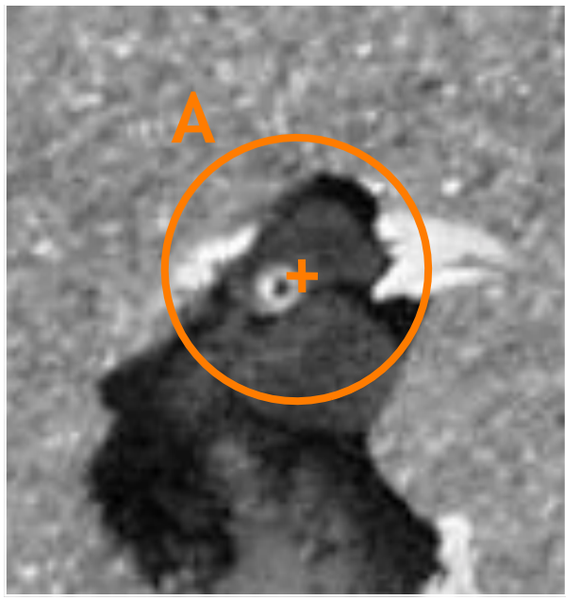


$$repeatability = \frac{\#correspondences}{\#detected} \cdot 100\%$$

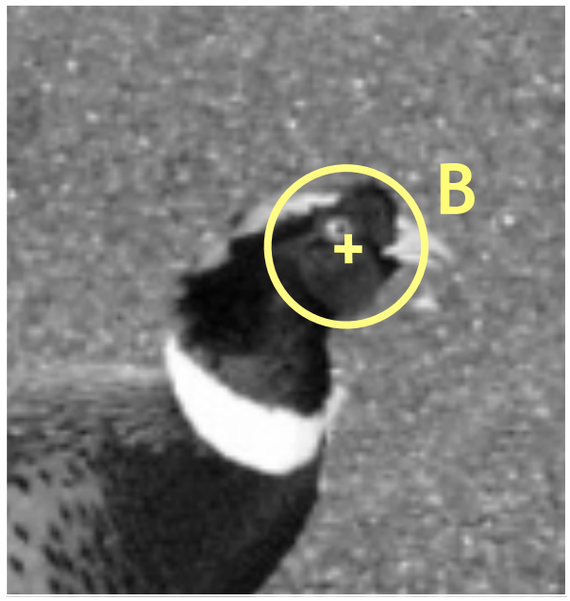
Evaluation criteria

TECHNISCHE UNIVERSITÄT DARMSTADT

- Repeatability rate : percentage of corresponding points



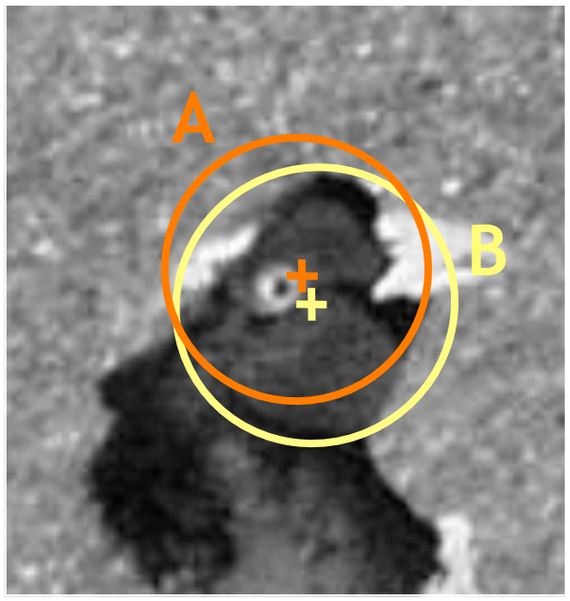
homography



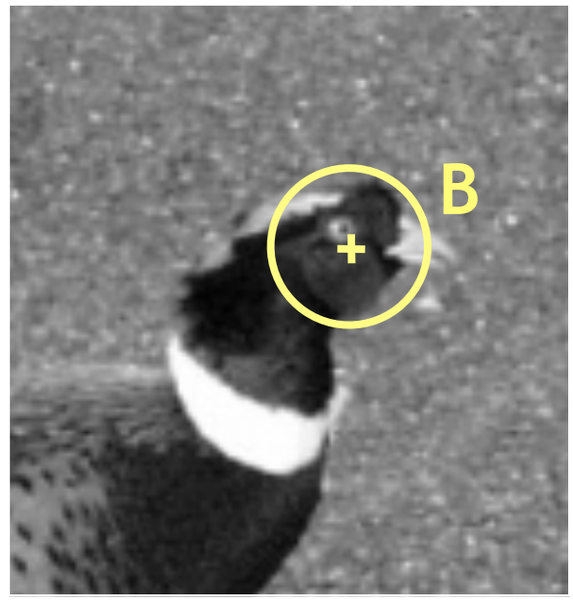
Evaluation criteria

TECHNISCHE UNIVERSITÄT DARMSTADT

- Repeatability rate : percentage of corresponding points



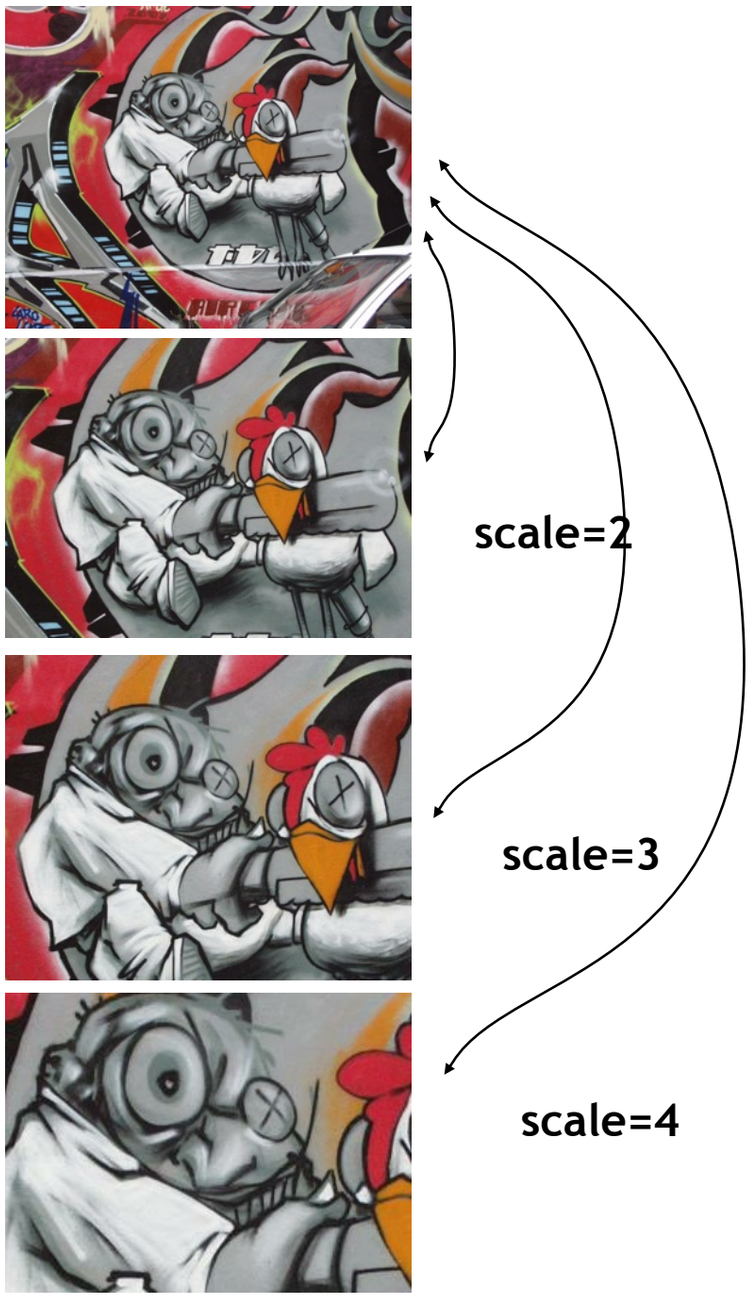
homography



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

T=60%

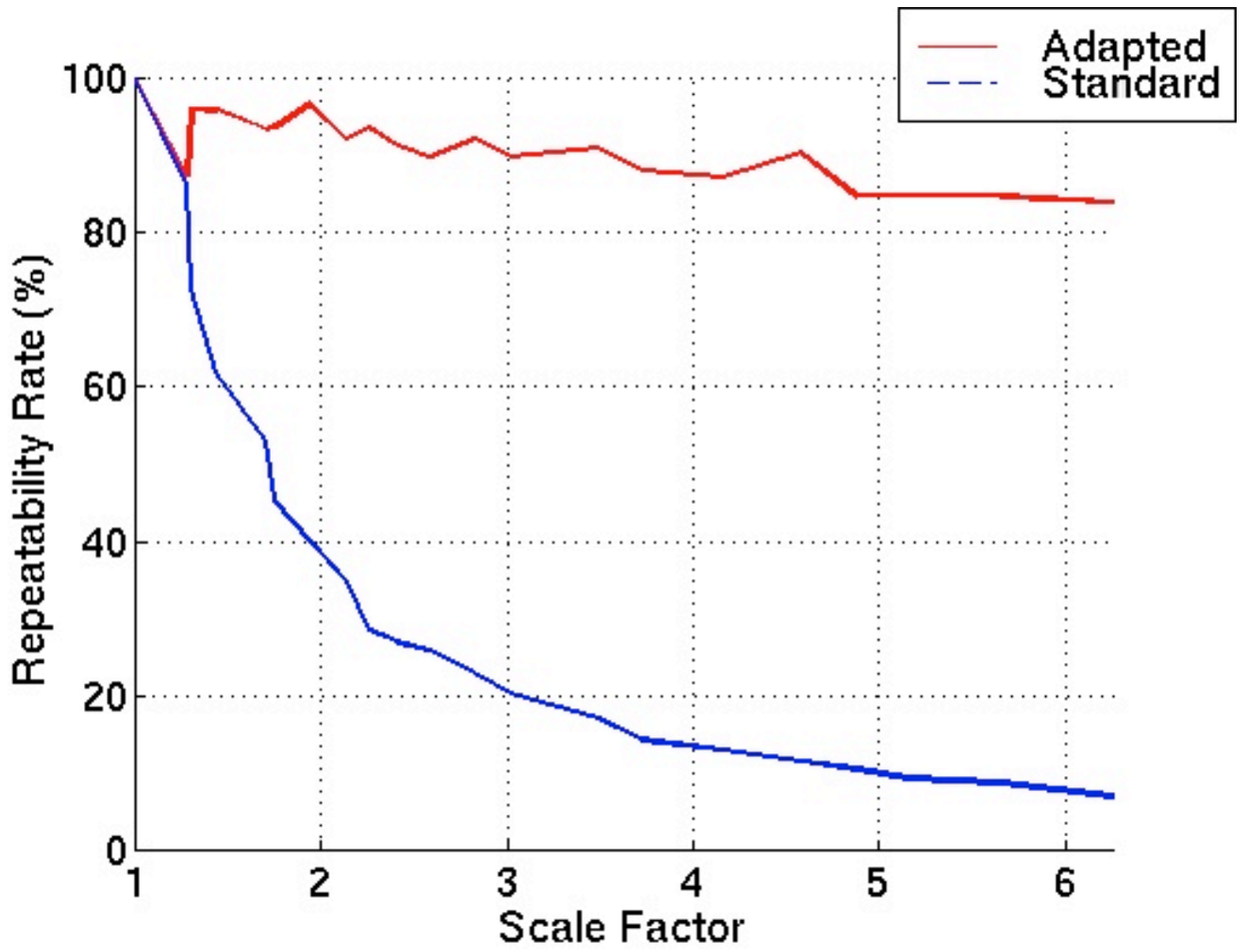
Evaluation criteria



Test set

Adapted – scale invariant

Standard – rotation invariant



Overview Today

TECHNISCHE UNIVERSITÄT DARMSTADT

- 1 - Local Interest Point Detection
  - finding discriminant points (Harris, Hessian)
  - scale invariant interest point detection

- **2 - Local Descriptors (= Features)**

- 3 - Bag-of-Words Model (BoW)
  - BoW for Object Categorization

2 - Local Descriptors / Features

TECHNISCHE UNIVERSITÄT DARMSTADT

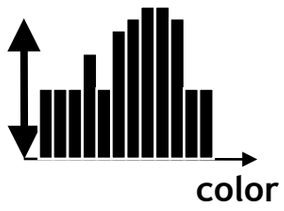
- Important properties of local descriptors
  - Distinctiveness, invariance, robustness, dimensionality, etc...
- Local descriptors
  - Image patches
  - SIFT
  - Shape context
- Evaluation criteria

Local descriptors

TECHNISCHE UNIVERSITÄT DARMSTADT

- Detector finds location, scale and shape of interest regions
- Local descriptors are computed for interest regions

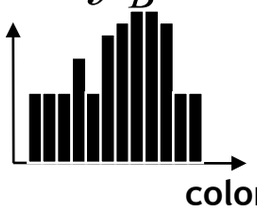
f_A



Similarity measure

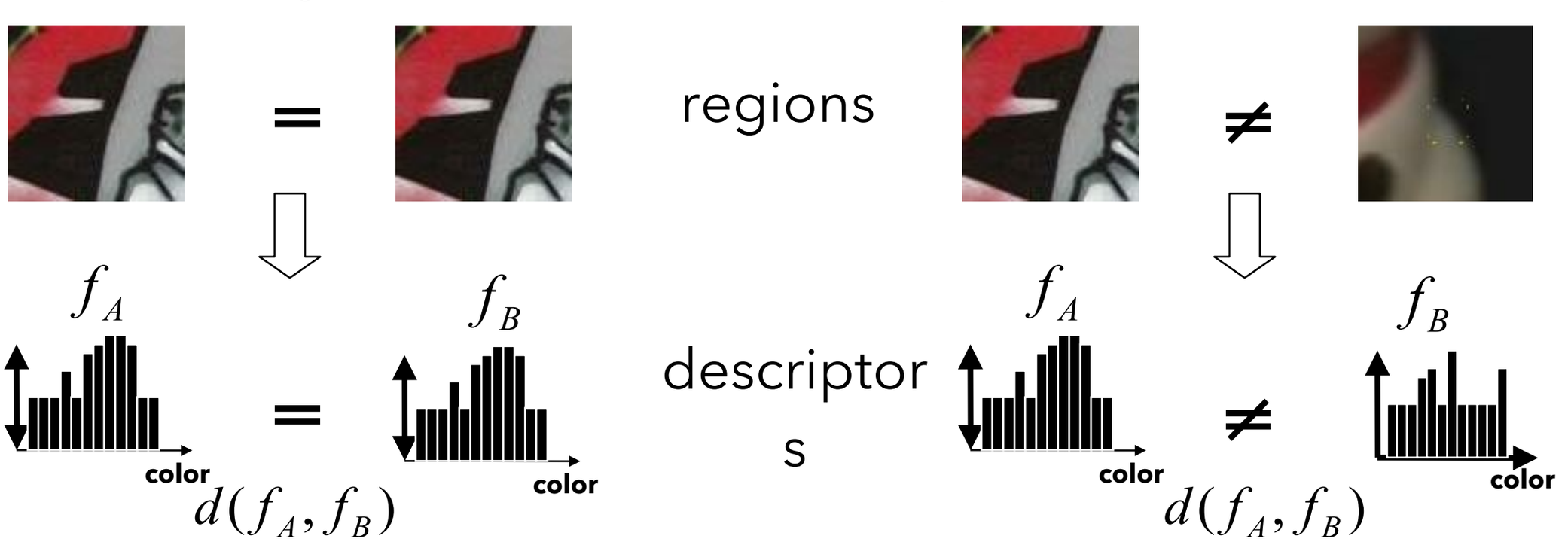
$$= d(f_A, f_B)$$

f_B



Local descriptors

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors



Local descriptors

TECHNISCHE UNIVERSITÄT DARMSTADT

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness

Two ways to obtain invariance

Local descriptors

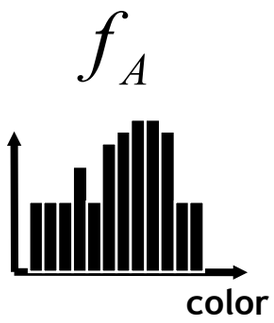
TECHNISCHE UNIVERSITÄT DARMSTADT

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness

Two ways to obtain invariance

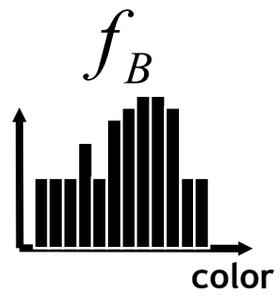
=

1. Computing invariant descriptor



⇩

=

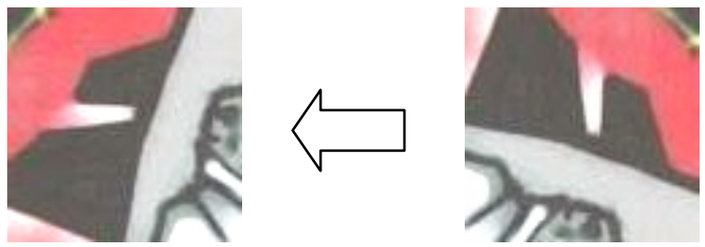


Local descriptors

TECHNISCHE UNIVERSITÄT DARMSTADT

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness

Two ways to obtain invariance



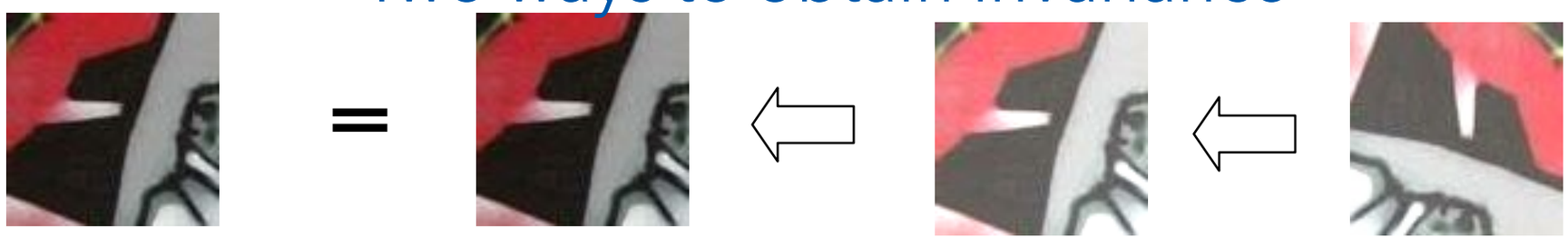
2. Geometric normalization

Local descriptors

TECHNISCHE UNIVERSITÄT DARMSTADT

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness

Two ways to obtain invariance



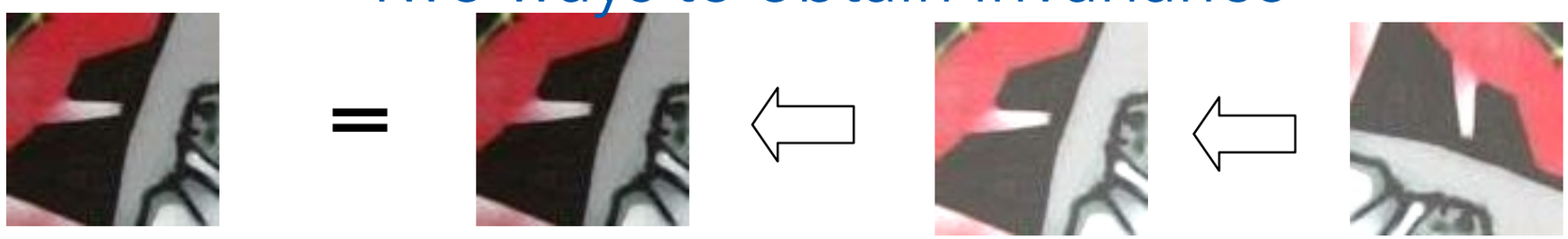
2. Geometric normalization
+ photometric normalization

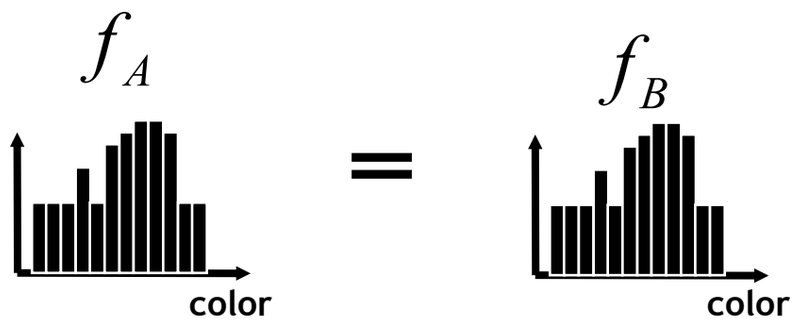
Local descriptors

TECHNISCHE UNIVERSITÄT DARMSTADT

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness

Two ways to obtain invariance



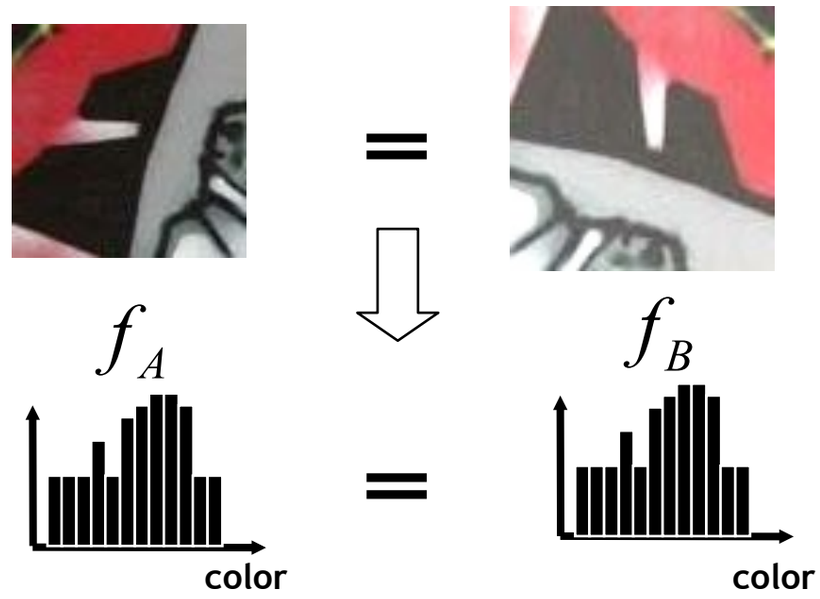


2. Geometric normalization
+ photometric normalization
+ computing descriptor

Local descriptors

TECHNISCHE UNIVERSITÄT DARMSTADT

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness
  - Robustness
    - Visually similar regions should have similar descriptors despite noise (geometric, photometric)



Local descriptors

TECHNISCHE UNIVERSITÄT DARMSTADT

- Important properties of local descriptors
  - Distinctiveness
    - Visually similar regions should have similar descriptors
    - Different regions should have different descriptors
  - Invariance
    - Visually similar regions should have similar descriptors despite the transformation (geometric, photometric) i.e., rotation, brightness
  - Robustness
    - Visually similar regions should have similar descriptors despite the noise (geometric, photometric)
  - Dimensionality
    - Descriptors should be low dimensional, i.e. small number of histogram bins.
      - Efficiency (large databases)
      - Generalization property

2 - Local Descriptors / Features

TECHNISCHE UNIVERSITÄT DARMSTADT

- Important properties of local descriptors
  - Distinctiveness, invariance, robustness, dimensionality, etc...

- Local descriptors
  - Image patches
  - SIFT
  - Shape context

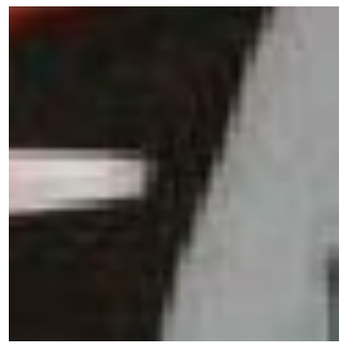
- Evaluation criteria

Local Descriptors

TECHNISCHE UNIVERSITÄT DARMSTADT

- Image patch (vector of pixels)
  - Invariant only when computed on normalized patches

$$f_1 = \begin{bmatrix} I_1(0,0) \\ I_1(0,1) \\ I_1(0,2) \\ \vdots \end{bmatrix}$$



$$f_2 = \begin{bmatrix} I_2(0,0) \\ I_2(0,1) \\ I_2(0,2) \\ \vdots \end{bmatrix}$$

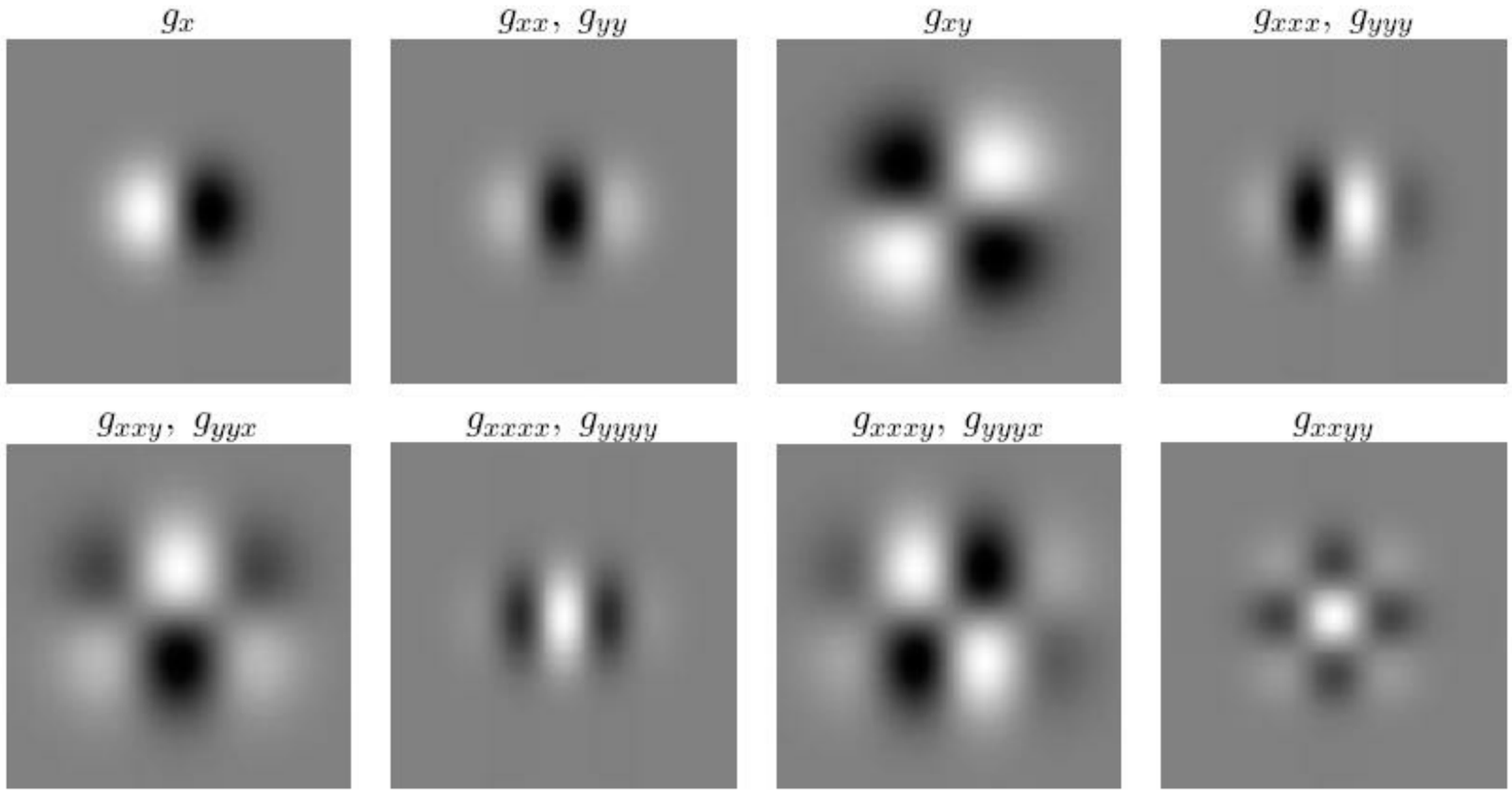
- Distinctive & easy to implement
- But high-dimensional and not very robust

- Match descriptors using:
  - Euclidean distance
  - Cross correlation

Local Descriptors

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bank of filter responses ("jet")
  - Invariant only when computed on normalized patch



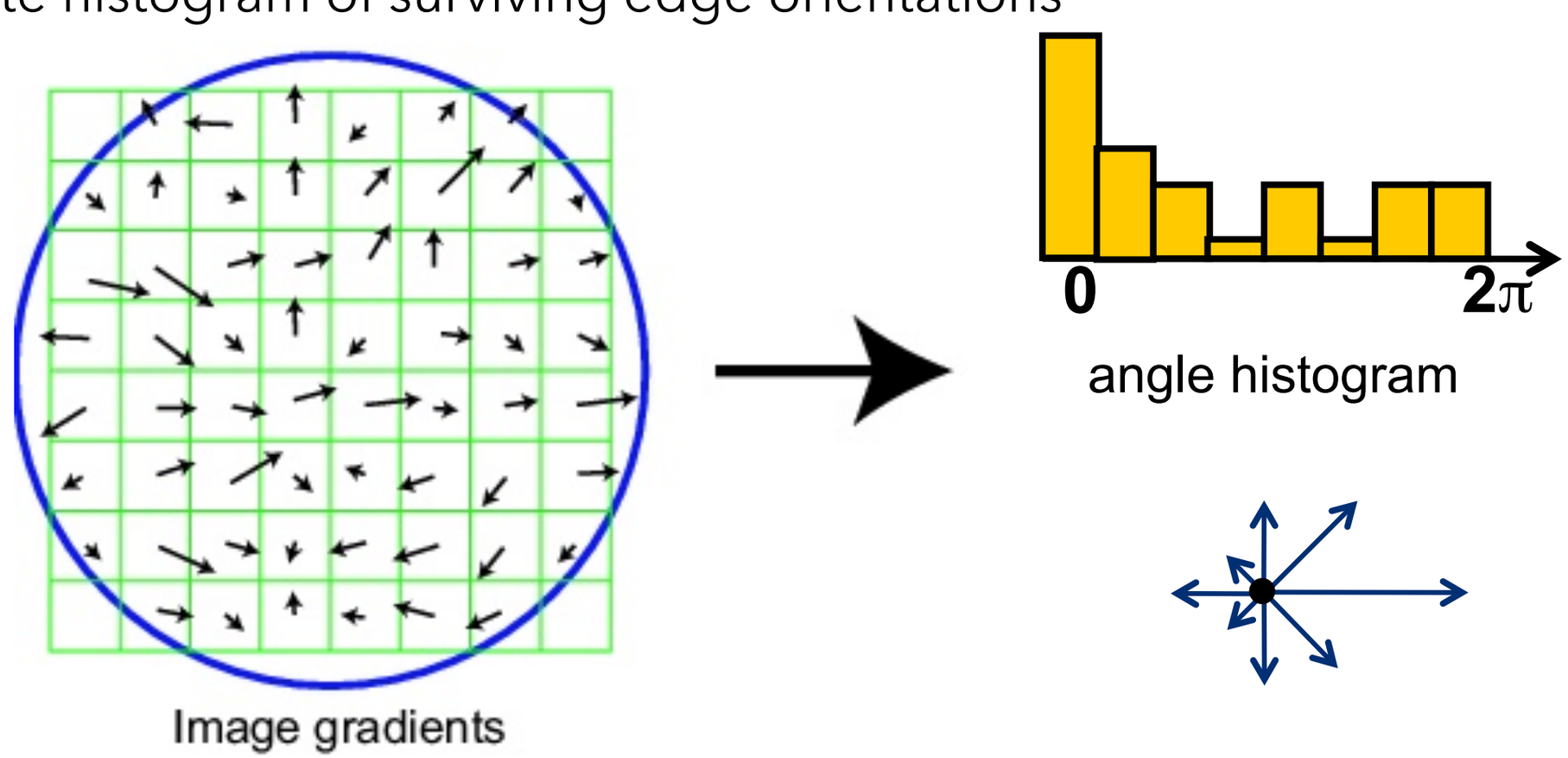
$$[g_x, g_y, g_{xx}, g_{xy}, g_{yy}, \ldots]$$

Responses of Gaussian derivatives

Scale Invariant Feature Transform

TECHNISCHE UNIVERSITÄT DARMSTADT

- Basic idea:
  - Take 16x16 square window around detected feature
  - Compute edge orientation (angle of the gradient - 90°) for each pixel
  - Throw out weak edges (threshold gradient magnitude)
  - Create histogram of surviving edge orientations

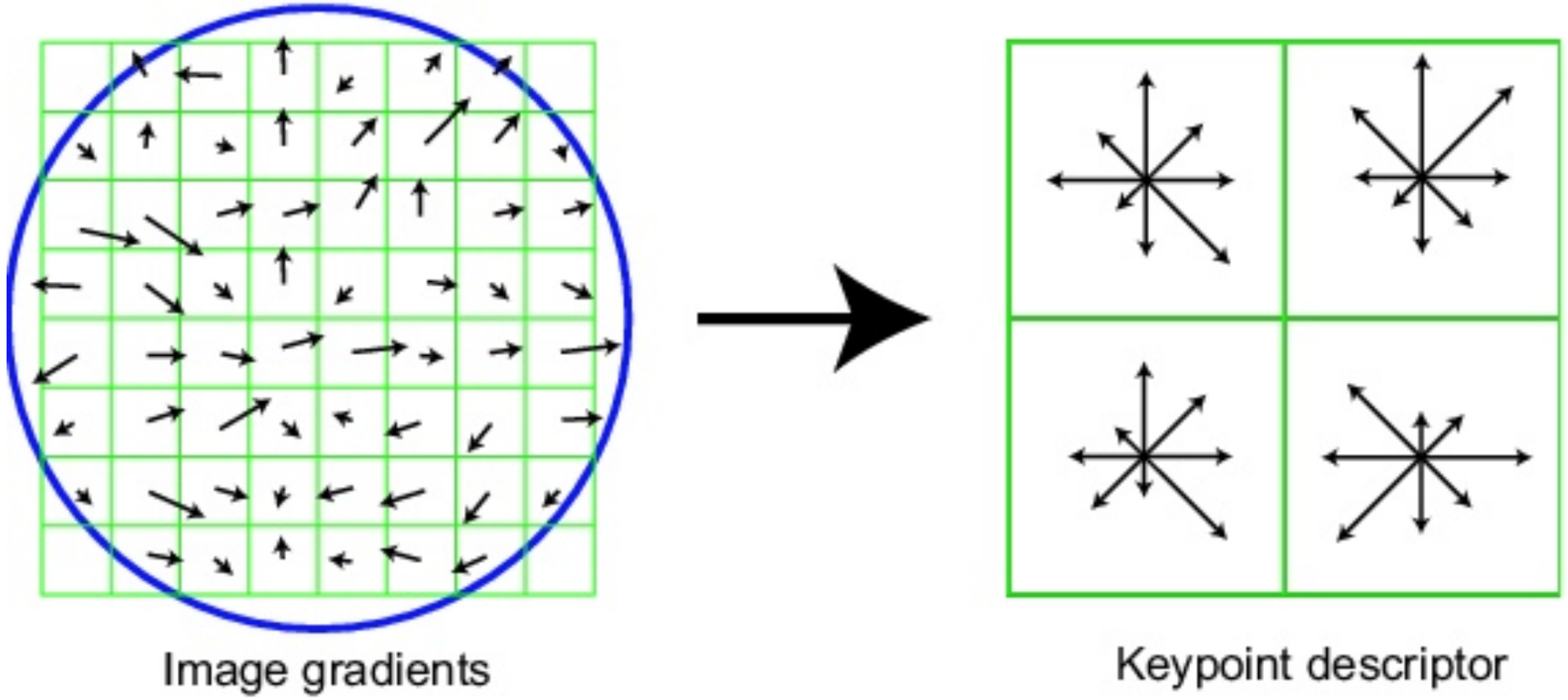


[Lowe]

SIFT Descriptor

TECHNISCHE UNIVERSITÄT DARMSTADT

- Full version:
  - Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
  - Compute an orientation histogram for each cell
  - 16 cells * 8 orientations = 128 dimensional descriptor

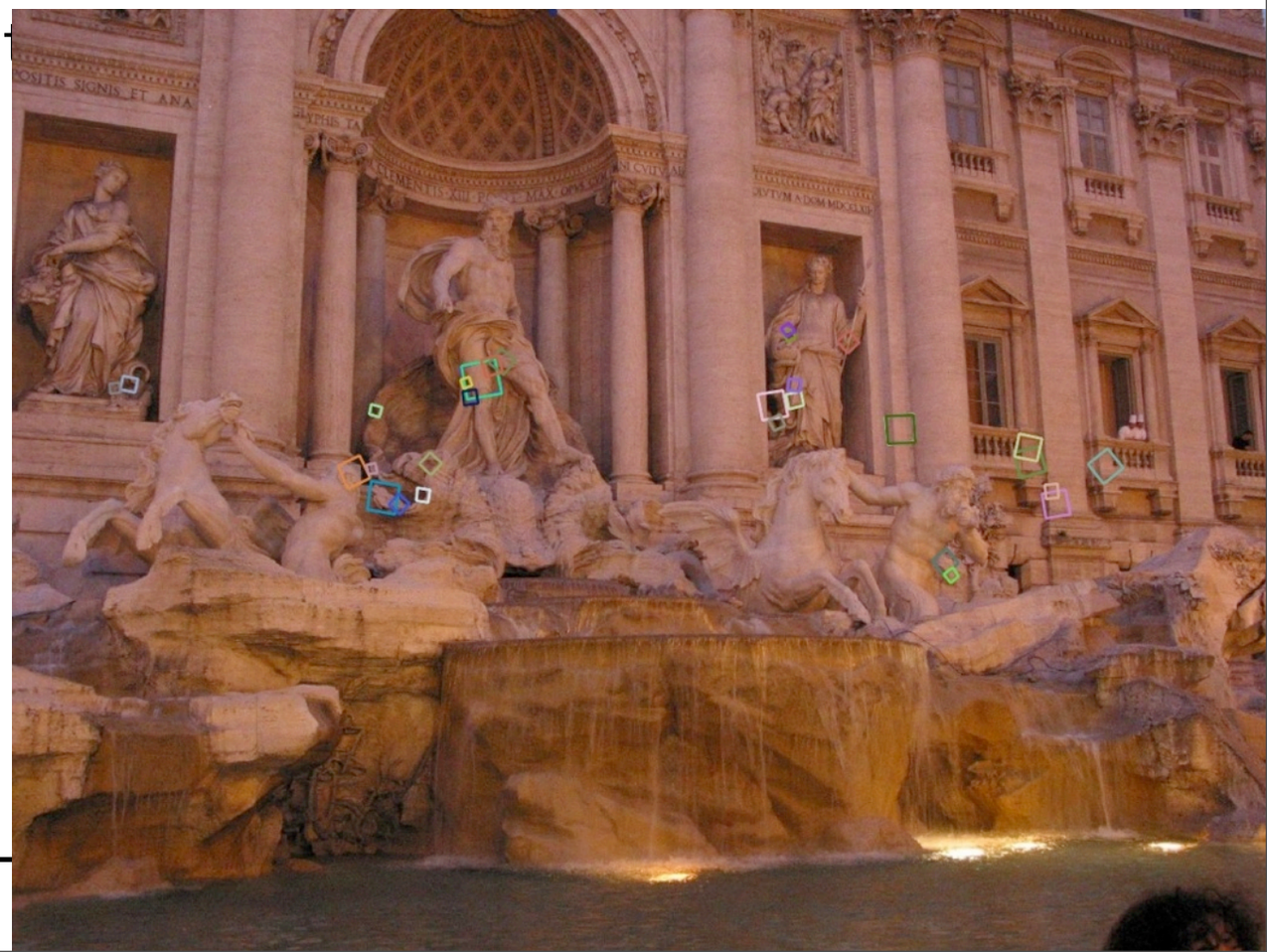


[Lowe]

Properties of SIFT

TECHNISCHE UNIVERSITÄT DARMSTADT

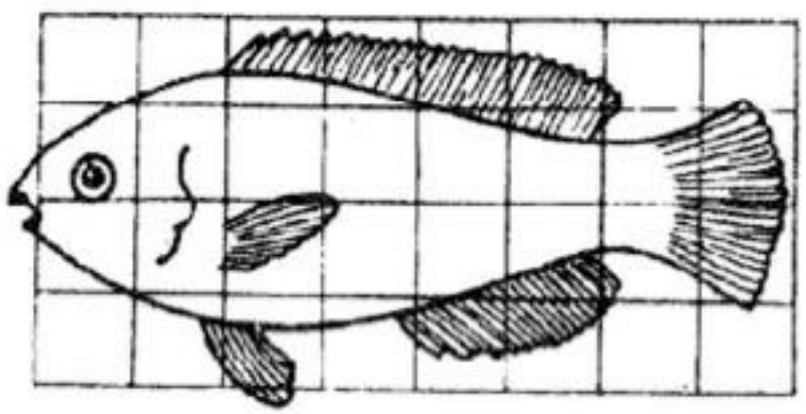
- Extraordinarily robust matching technique
  - Can handle changes in viewpoint
    - Up to about 60 degree out of plane rotation
  - Can handle significant changes in illumination
    - Sometimes even day vs. night (below)

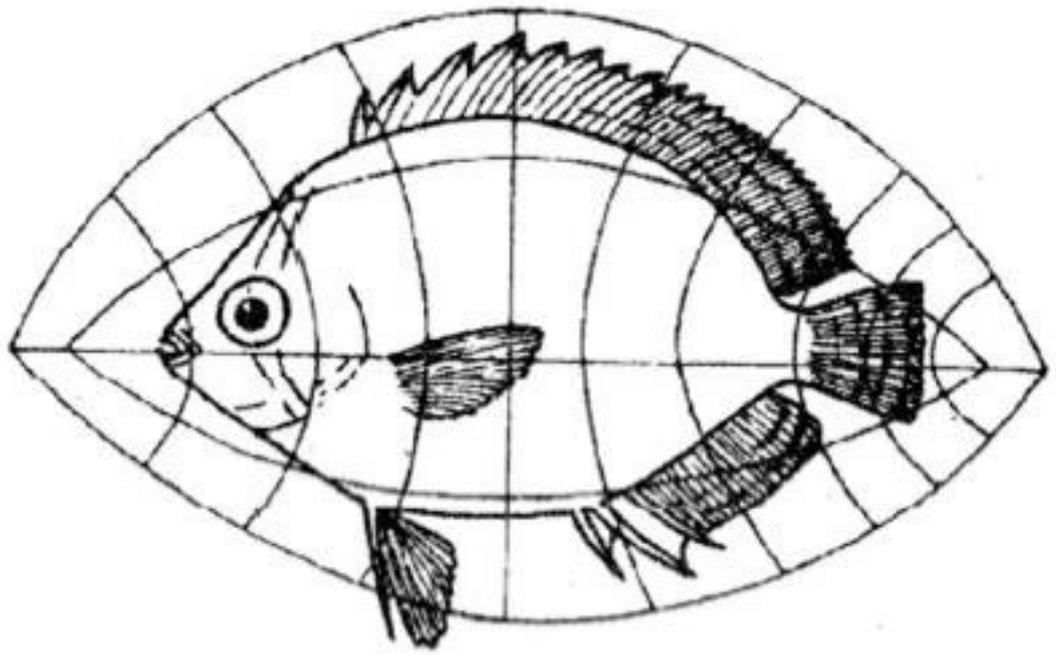


Shape context

TECHNISCHE UNIVERSITÄT DARMSTADT

- Motivation

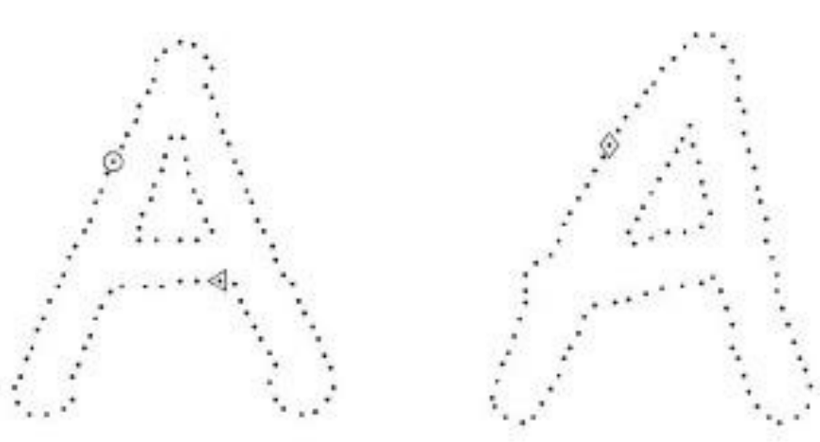


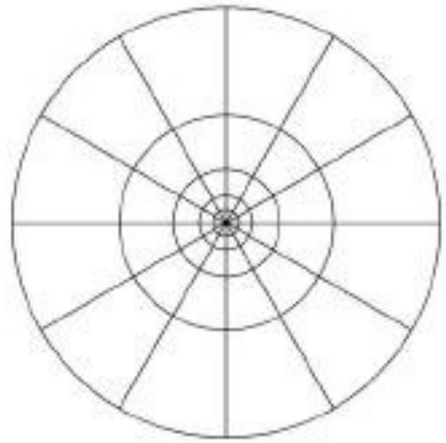


Shape context

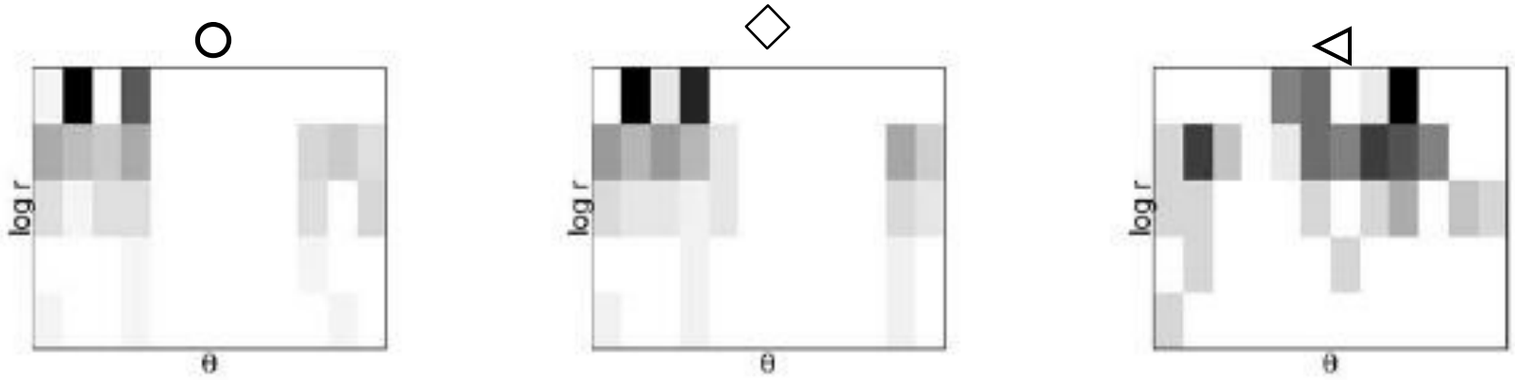
TECHNISCHE UNIVERSITÄT DARMSTADT

- Represent histogram of edge locations





Log polar coordinate system



Histogram of edge points in log-polar coordinate system

2 - Local Descriptors / Features

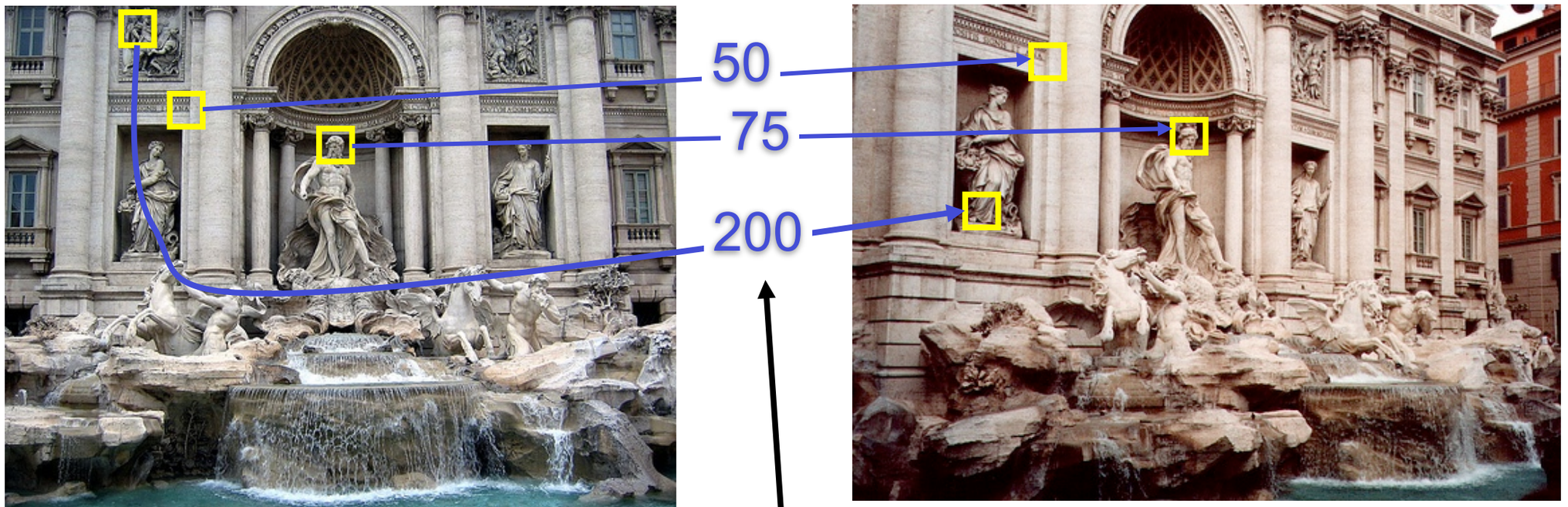
TECHNISCHE UNIVERSITÄT DARMSTADT

- Important properties of local descriptors
  - Distinctiveness, invariance, robustness, dimensionality, etc...
- Local descriptors
  - Image patches
  - SIFT
  - Shape context
- Evaluation criteria

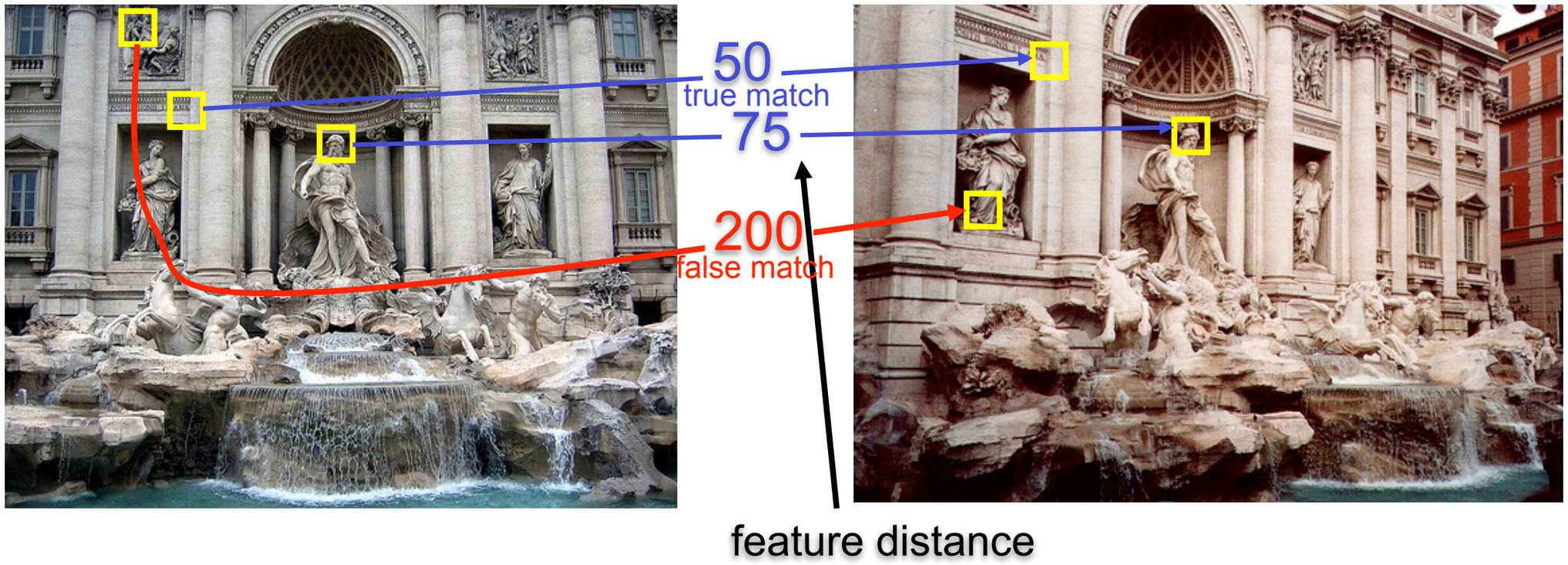
Evaluating the results

TECHNISCHE UNIVERSITÄT DARMSTADT

- How can we measure the performance of a feature matcher (descriptor + distance/classifier)?



True/false positives

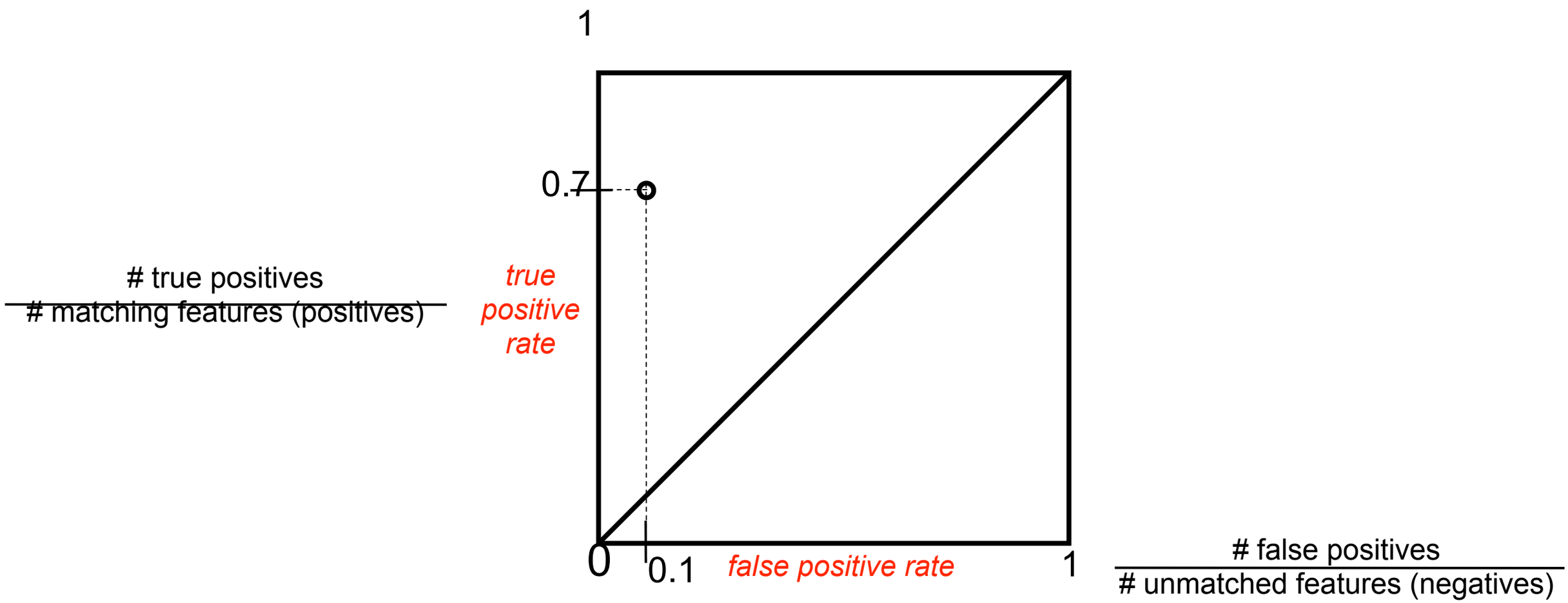


- The distance threshold affects performance
  - True positives = # of detected matches that are correct
    - Suppose we want to maximize these—how to choose threshold?
  - False positives = # of detected matches that are incorrect
    - Suppose we want to minimize these—how to choose threshold?

Evaluating the results

TECHNISCHE UNIVERSITÄT DARMSTADT

- How can we measure the performance of a feature matcher (descriptor + distance/classifier)?

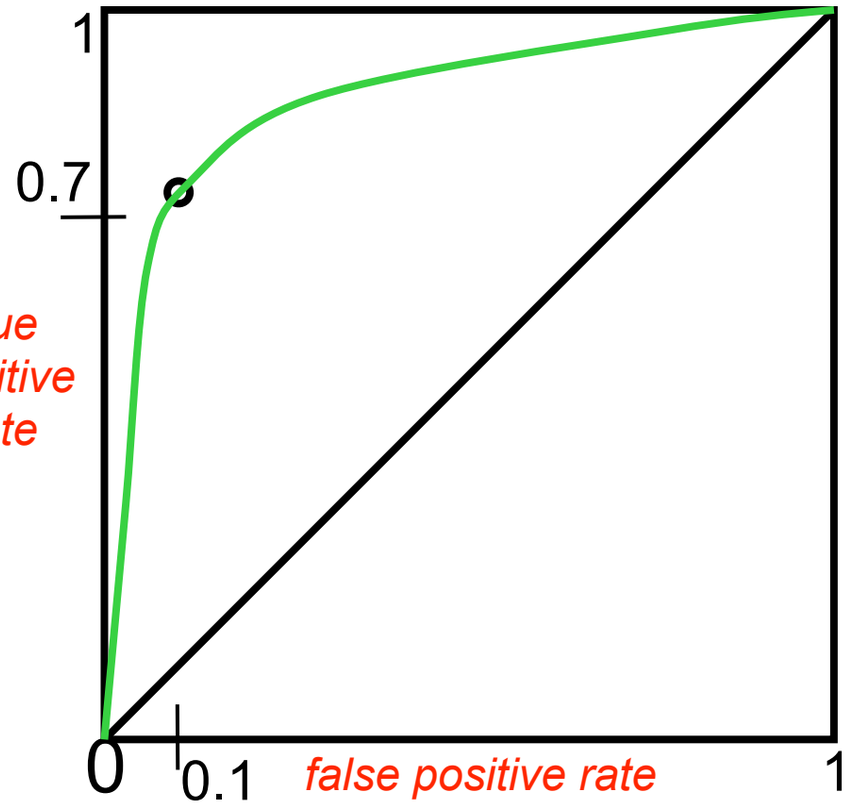


Evaluating the results

TECHNISCHE UNIVERSITÄT DARMSTADT

- How can we measure the performance of a feature matcher (descriptor + distance/classifier)?

ROC curve **(“Receiver Operator Characteristic”)**



true positives / # matching features (positives)

false positives / # unmatched features (negatives)

- ROC Curves
  - Generated by counting # current/incorrect matches, for different thresholds
  - Want to maximize area under the curve (AUC)

Overview Today

TECHNISCHE UNIVERSITÄT DARMSTADT

- 1 - Local Interest Point Detection
  - finding discriminant points (Harris, Hessian)
  - scale invariant interest point detection

- 2 - Local Descriptors (= Features)

- **3 - Bag-of-Words Model (BoW)**
  - **BoW for Object Categorization**

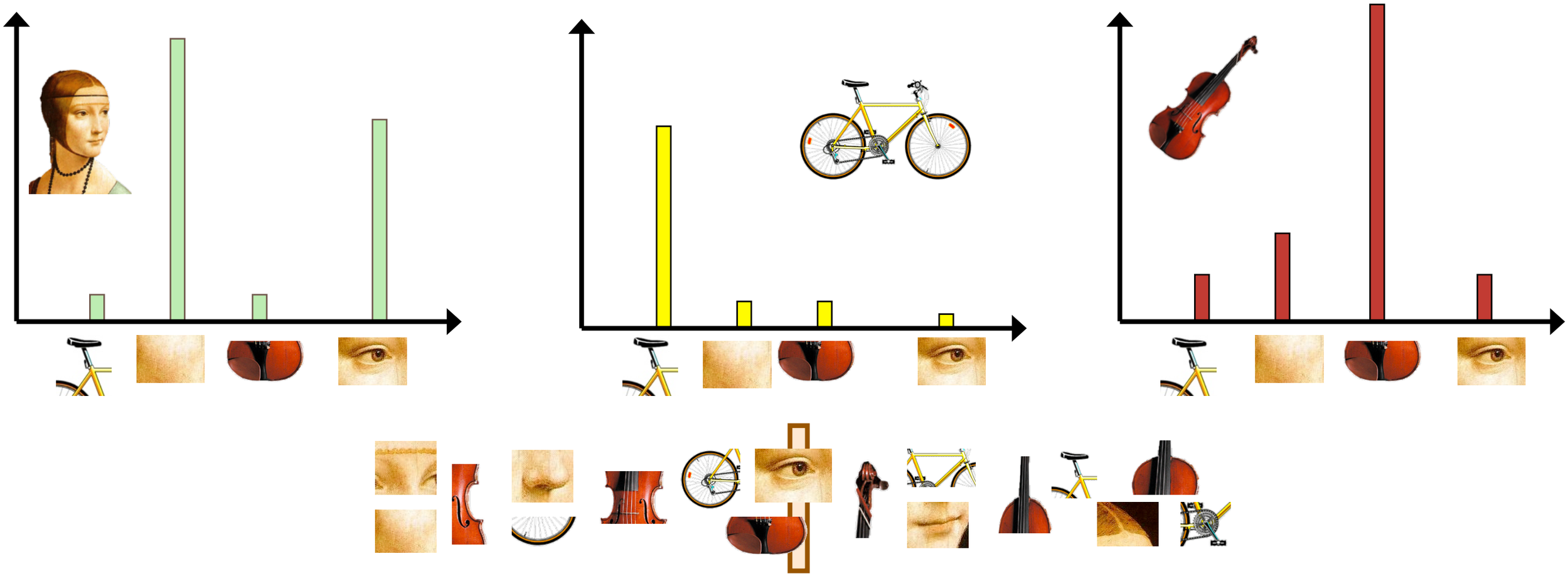
3 - Bag-of-Words Model (BoW) now for Object Categorization

Object → **Bag of 'words'**

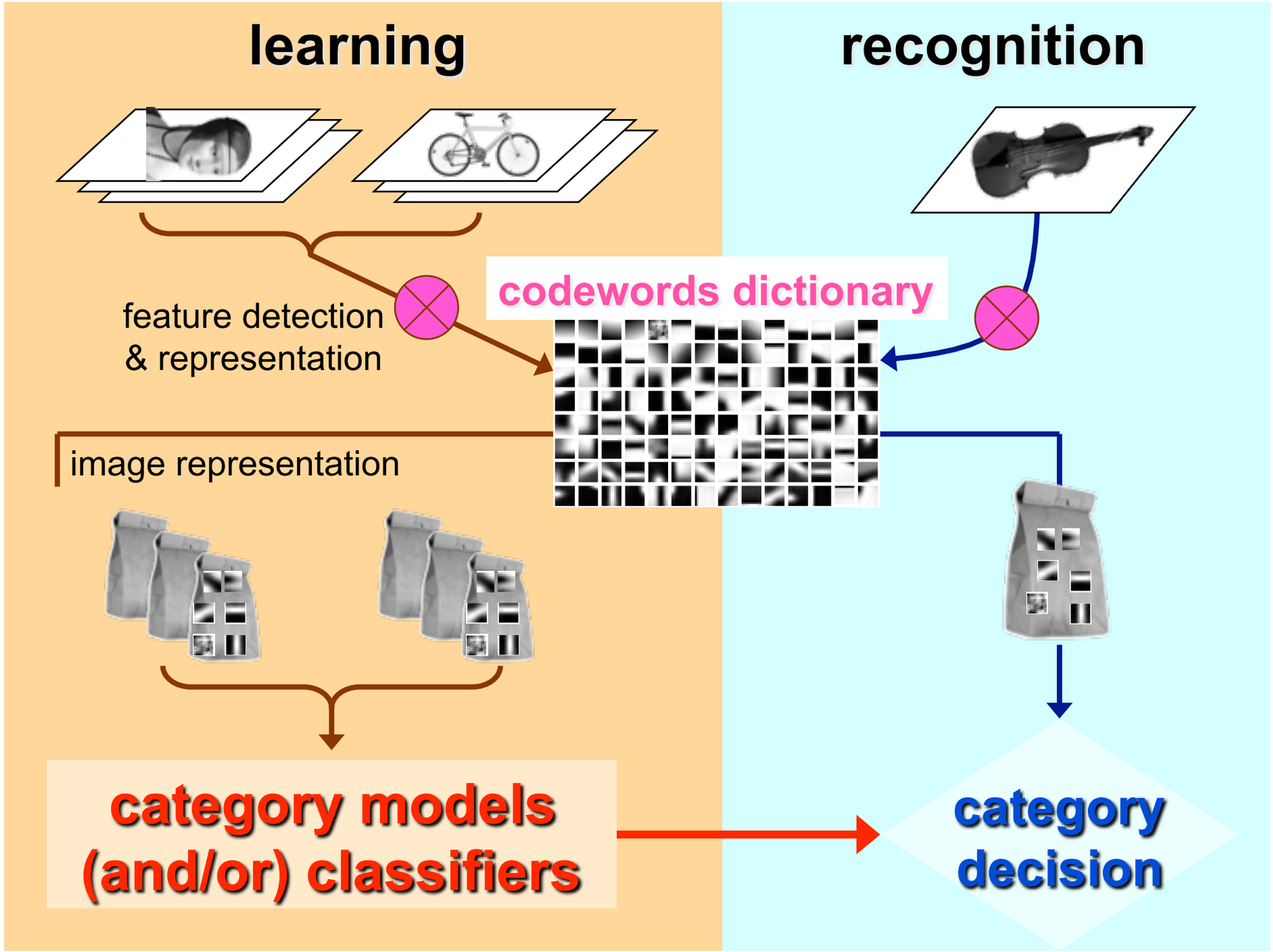
Bag of Words

TECHNISCHE UNIVERSITÄT DARMSTADT

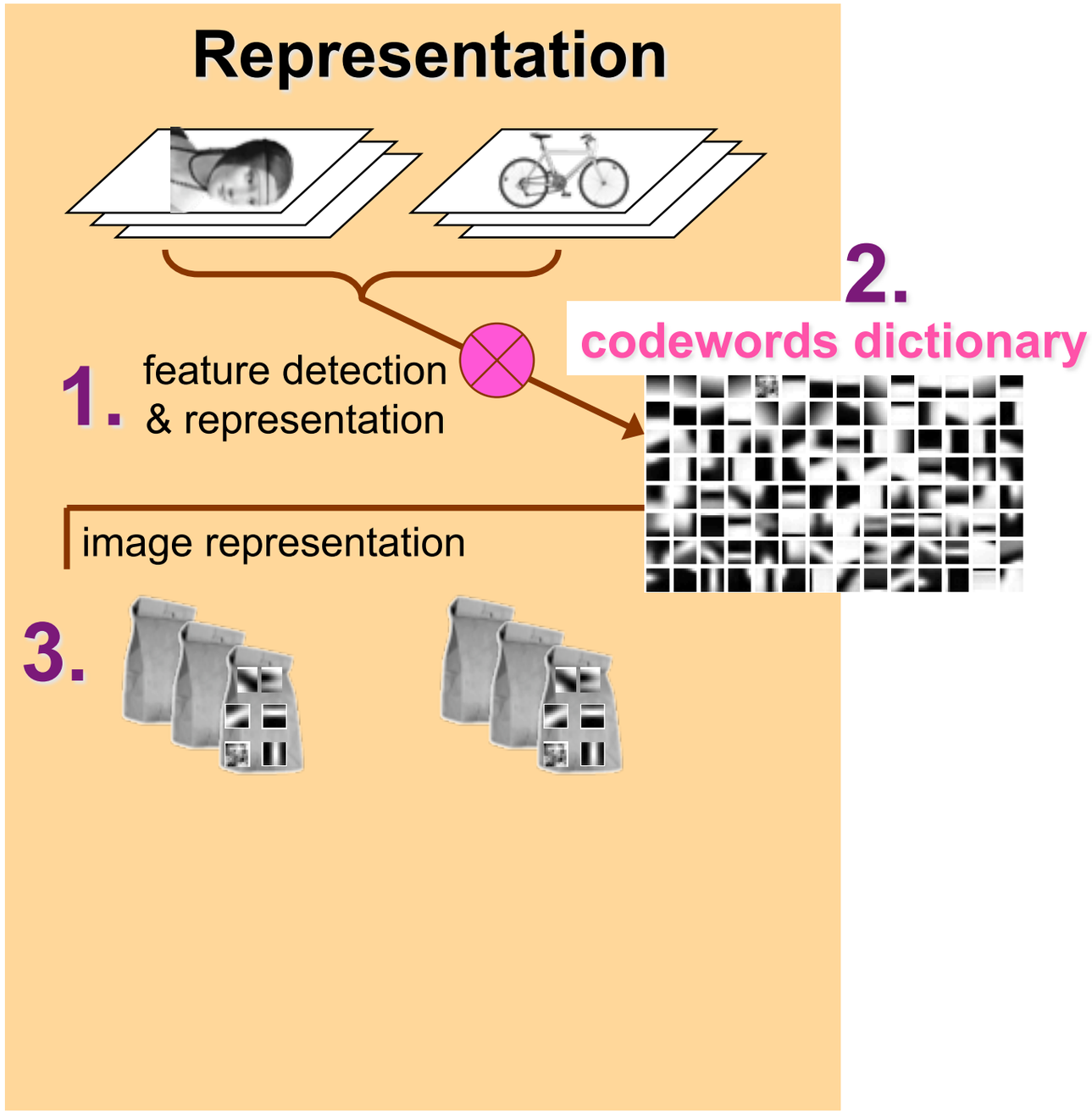
- Independent features
- Histogram representation



Bag-of-Words Model: Overview

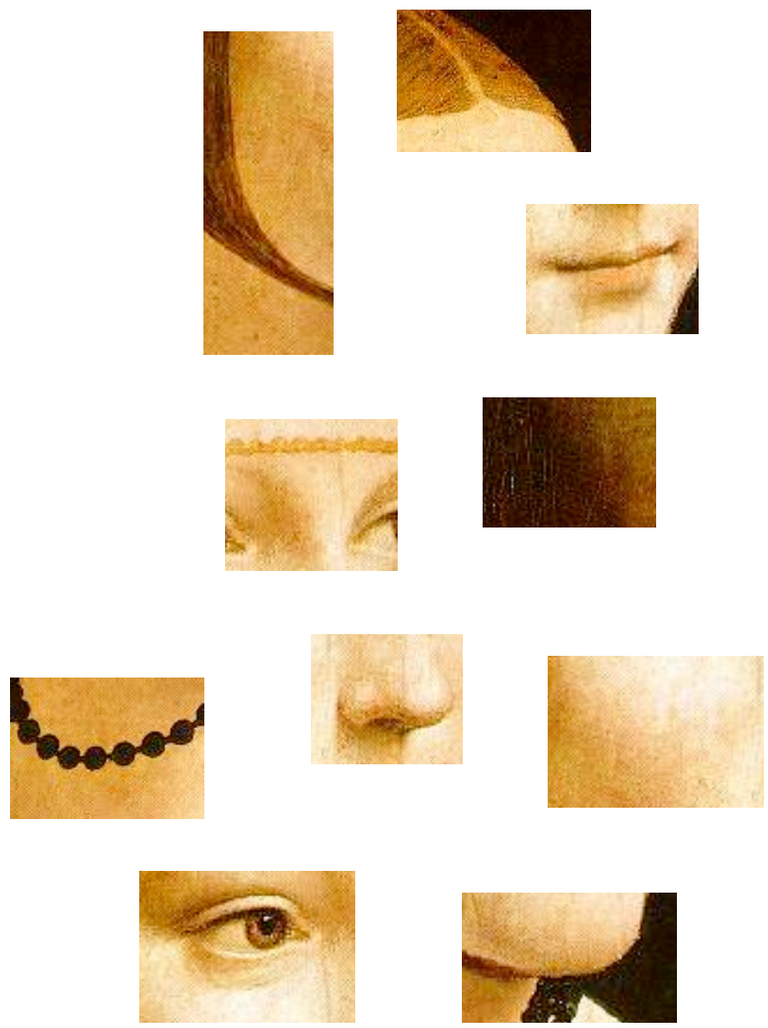


Bag-of-Words Model: Object Representation & Learning

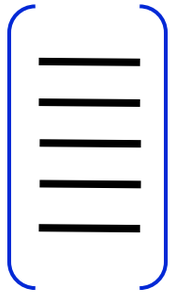


BoW-1. Feature detection and representation

TECHNISCHE UNIVERSITÄT DARMSTADT

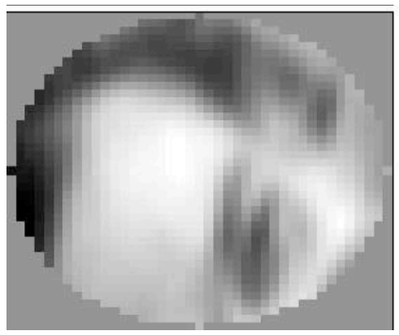


BoW-1. Feature detection and representation

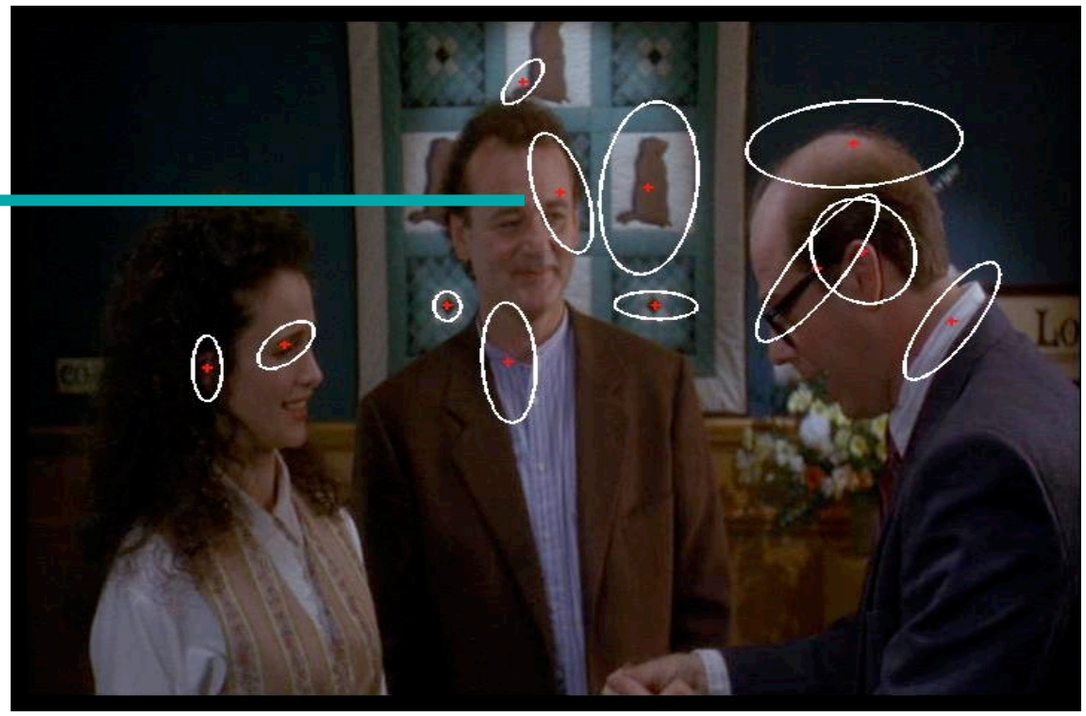


Compute descriptor

e.g. SIFT, shape context, etc.



Normalize patch

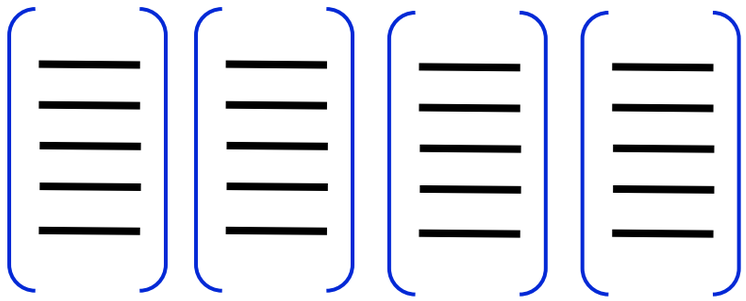


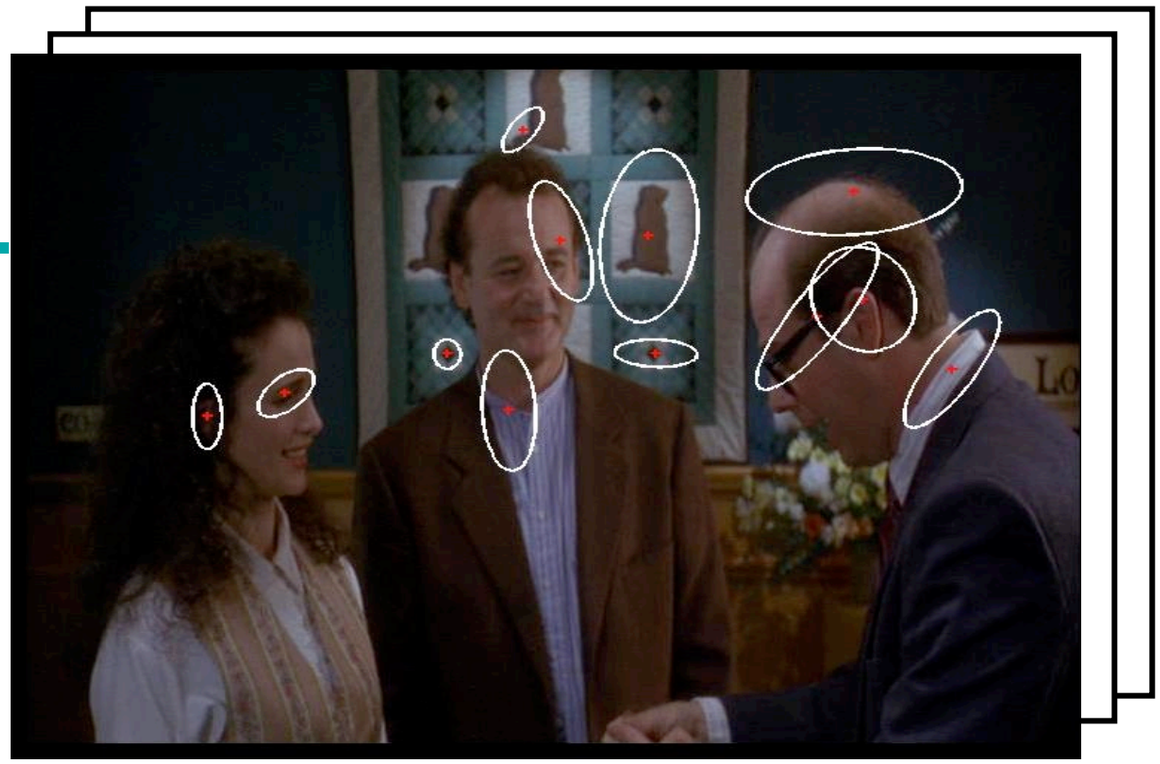
Detect patches

Local interest operator
(e.g. Harris-Laplace)
or regular grid

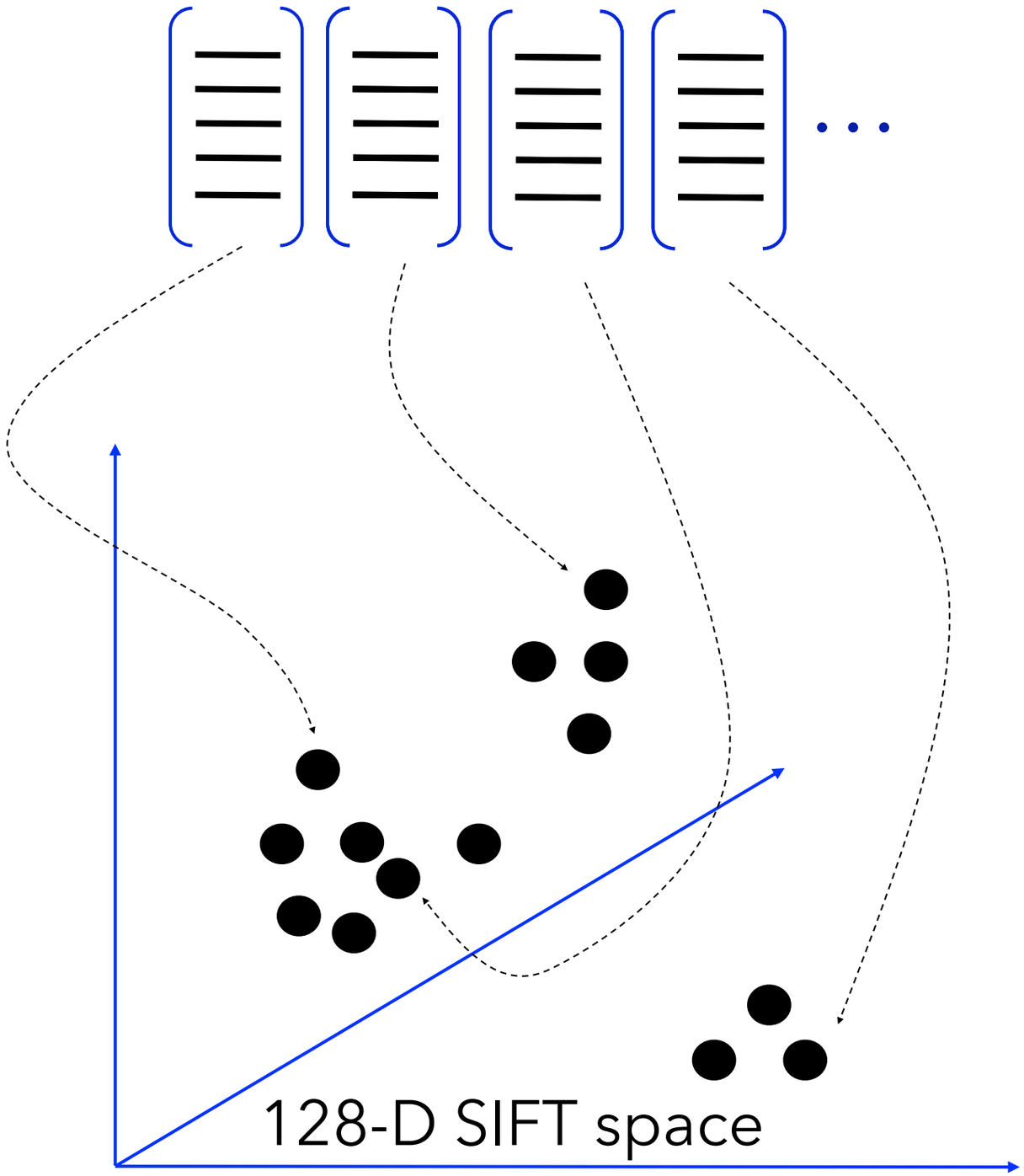
BoW-1. Feature detection and representation

TECHNISCHE UNIVERSITÄT DARMSTADT



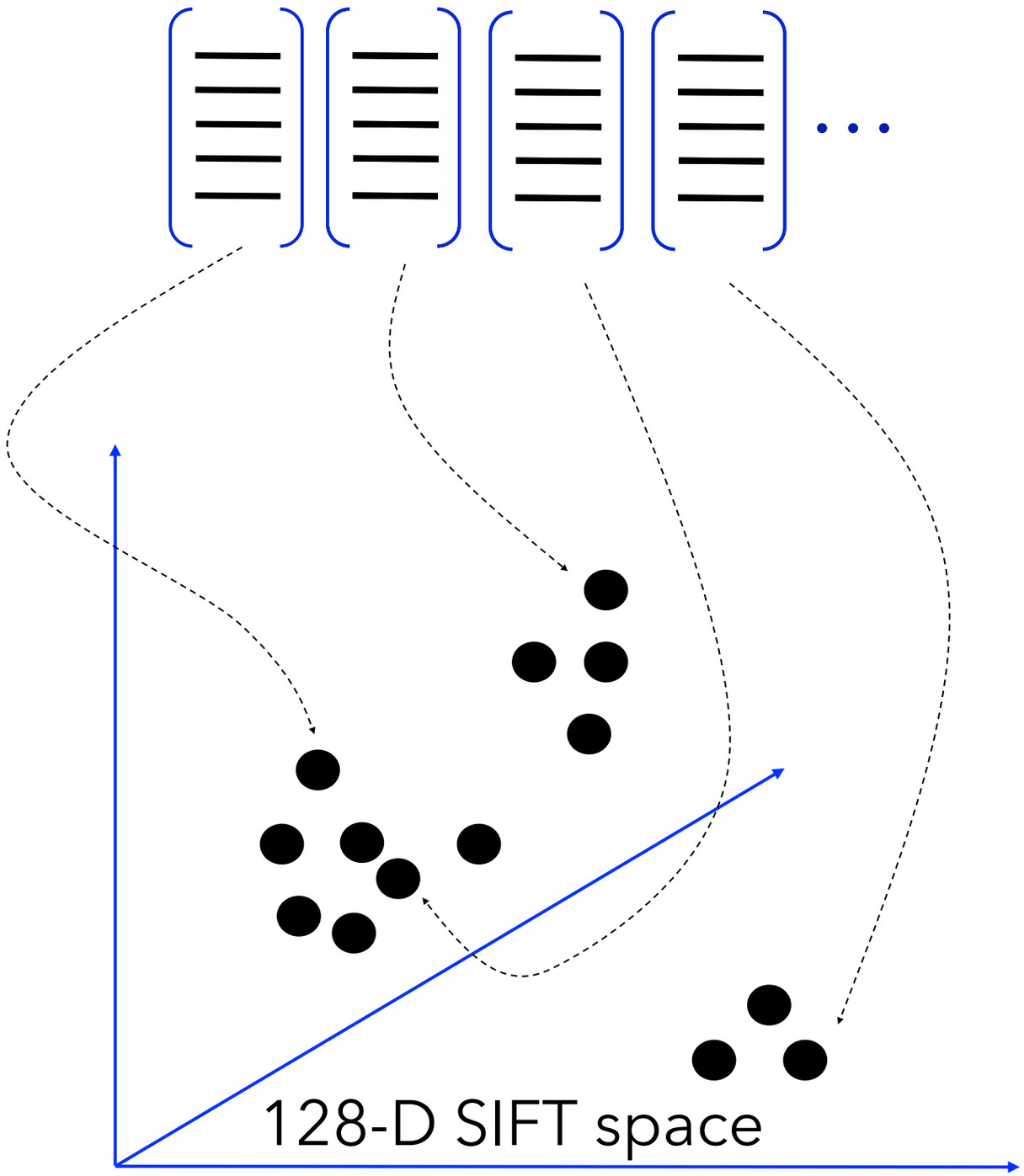


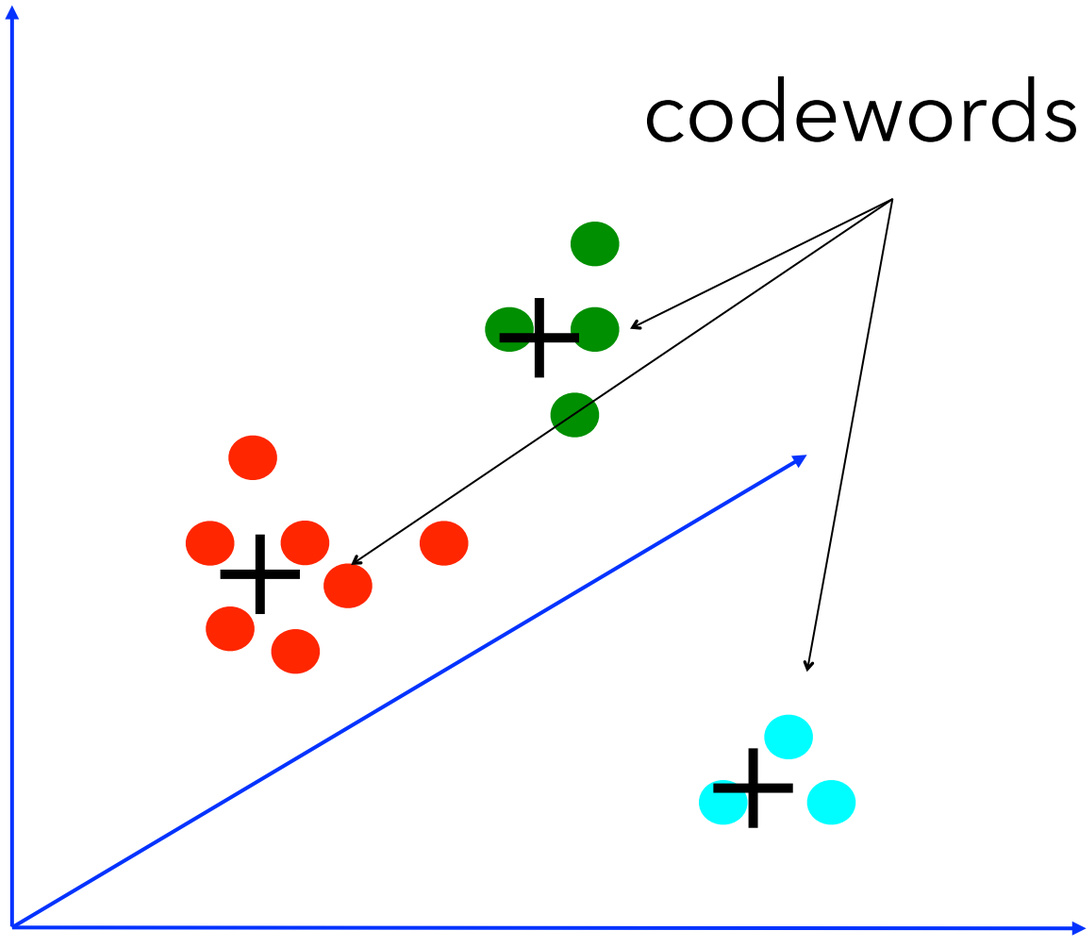
BoW-2. Codeword dictionary formation



BoW-2. Codeword dictionary formation

TECHNISCHE UNIVERSITÄT DARMSTADT





Vector quantization

e.g. k-means

Vector quantization

TECHNISCHE UNIVERSITÄT DARMSTADT

- Via clustering
- Simple, but good method: K-means clustering

```
Choose k data points to act as cluster centers

Until the cluster centers are unchanged

    Allocate each data point to cluster whose center is nearest

    Now ensure that every cluster has at least
    one data point; possible techniques for doing this include .
    supplying empty clusters with a point chosen at random from
    points far from their cluster center.

    Replace the cluster centers with the mean of the elements
    in their clusters.

end
```

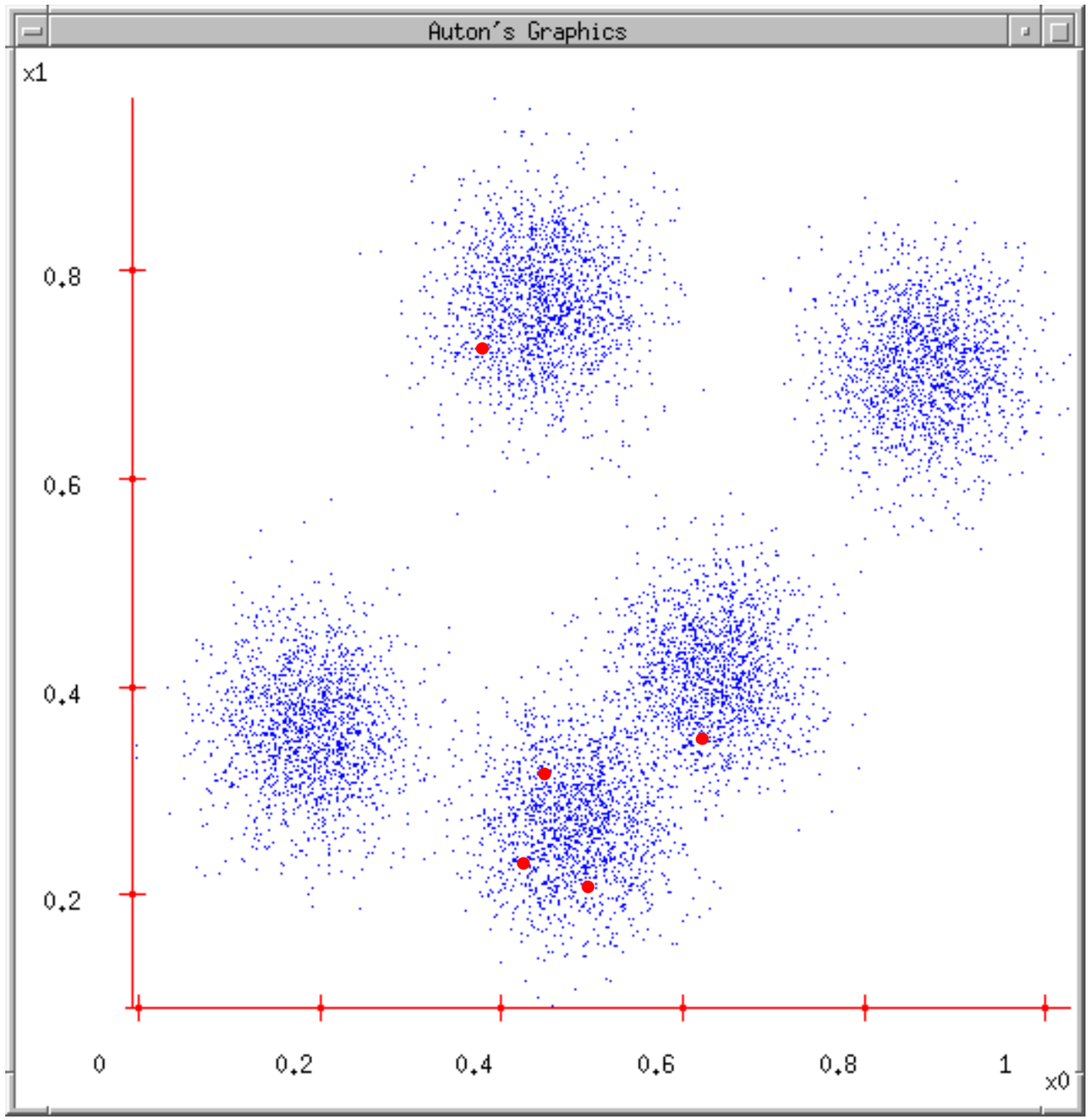
[FP]

K-Means

TECHNISCHE UNIVERSITÄT DARMSTADT

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations

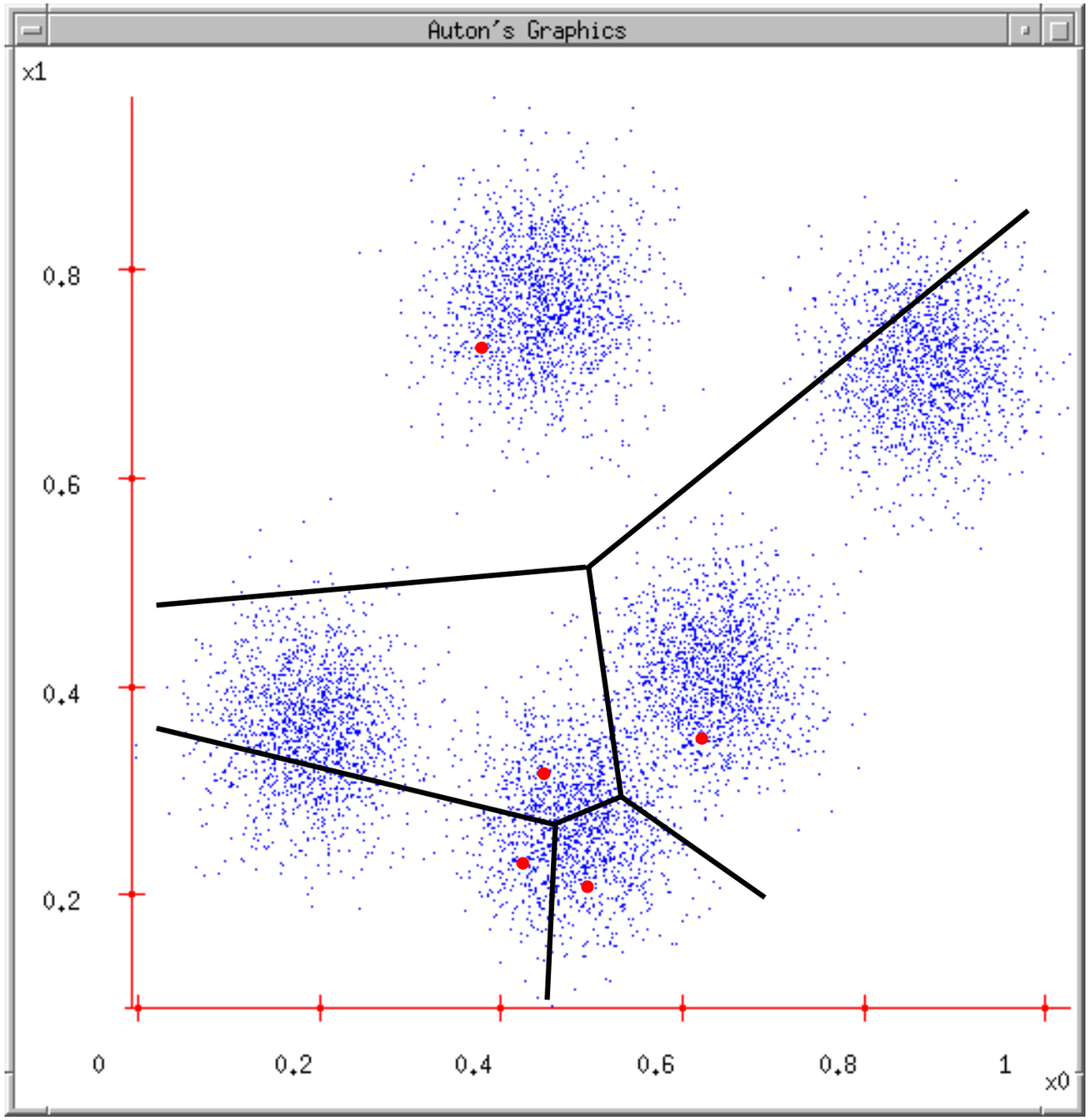


K-Means

TECHNISCHE UNIVERSITÄT DARMSTADT

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)

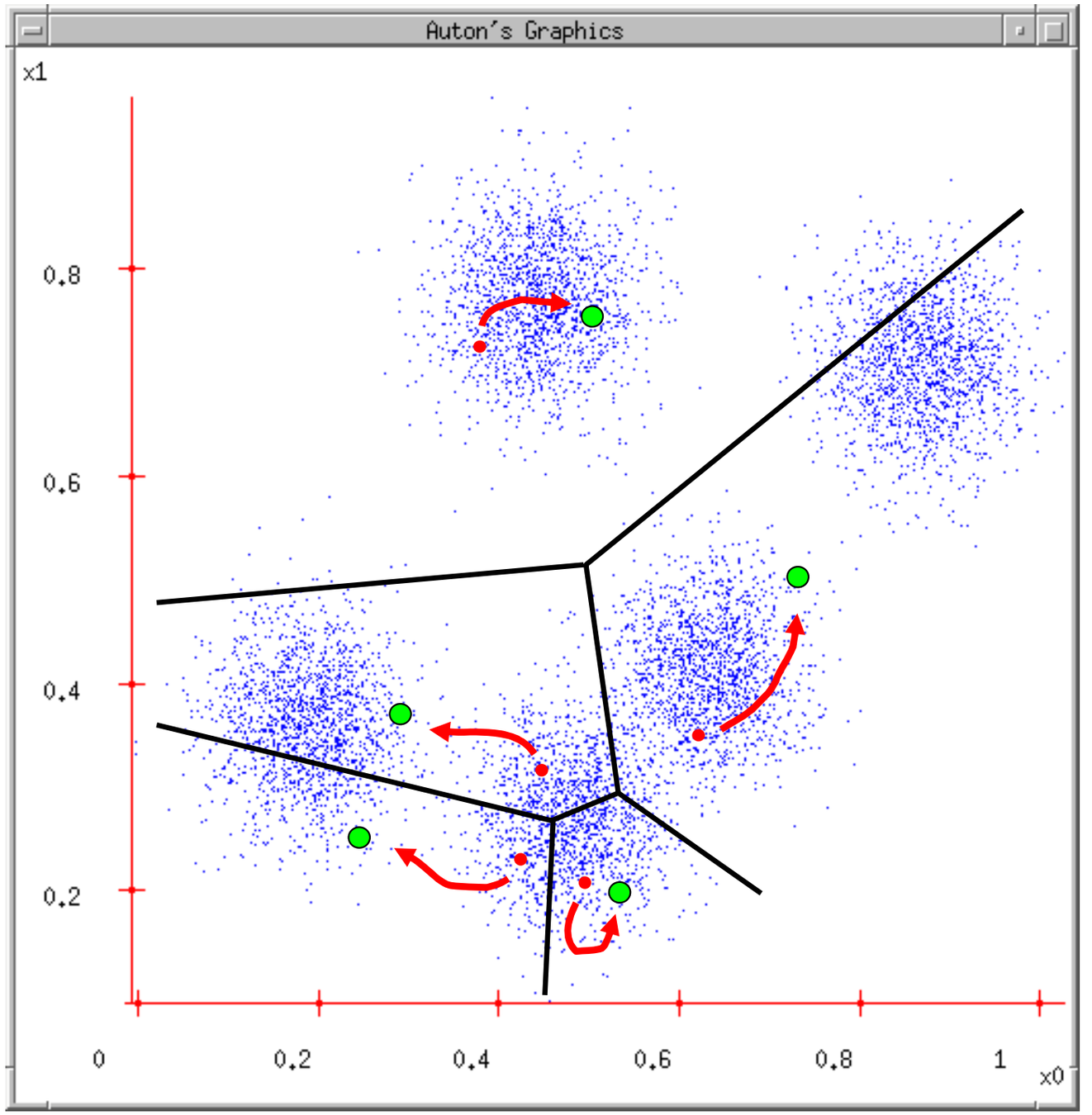


K-Means

TECHNISCHE UNIVERSITÄT DARMSTADT

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



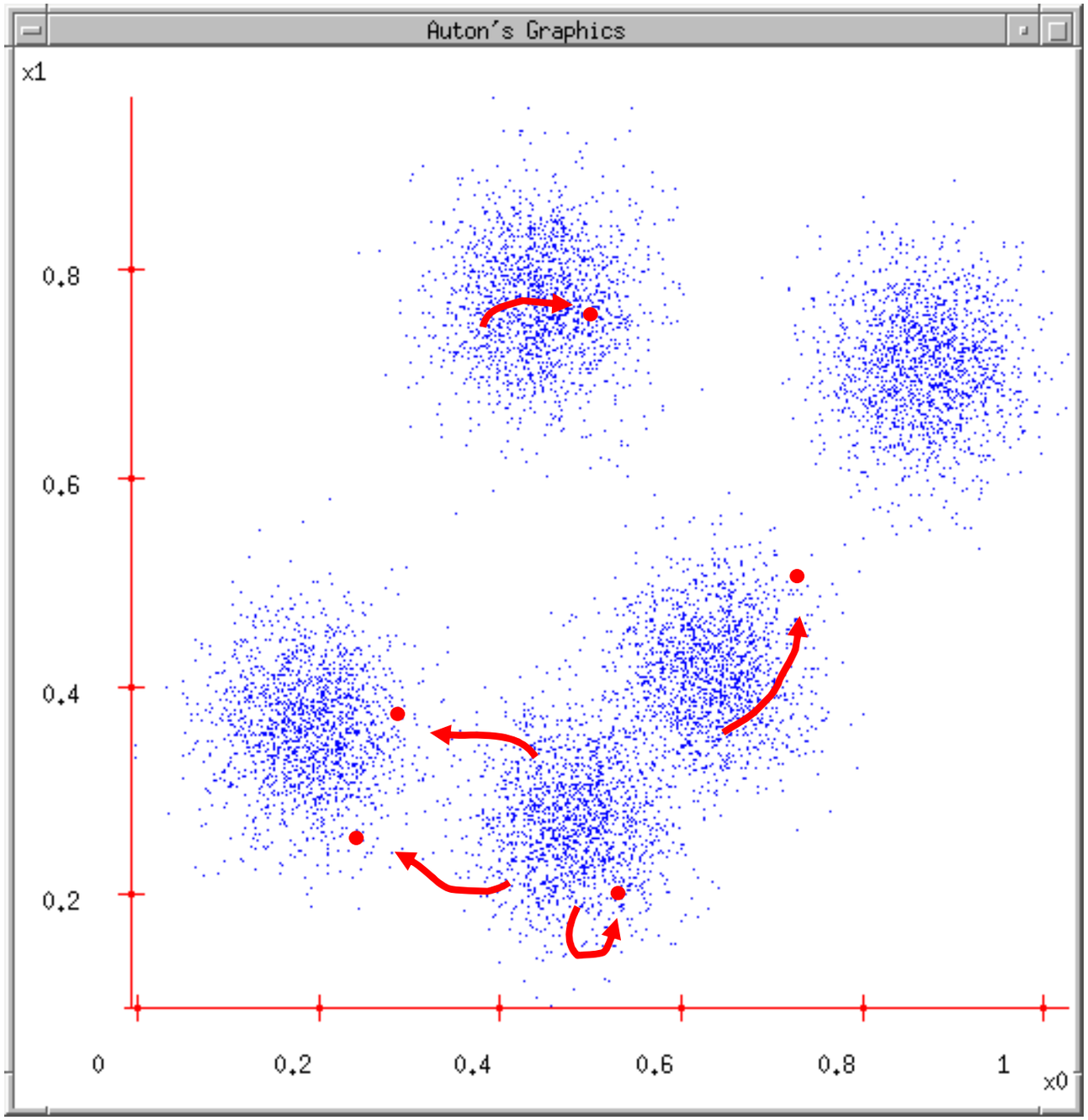
K-means and Hierarchical Clustering: Slide 9

K-Means

TECHNISCHE UNIVERSITÄT DARMSTADT

K-means

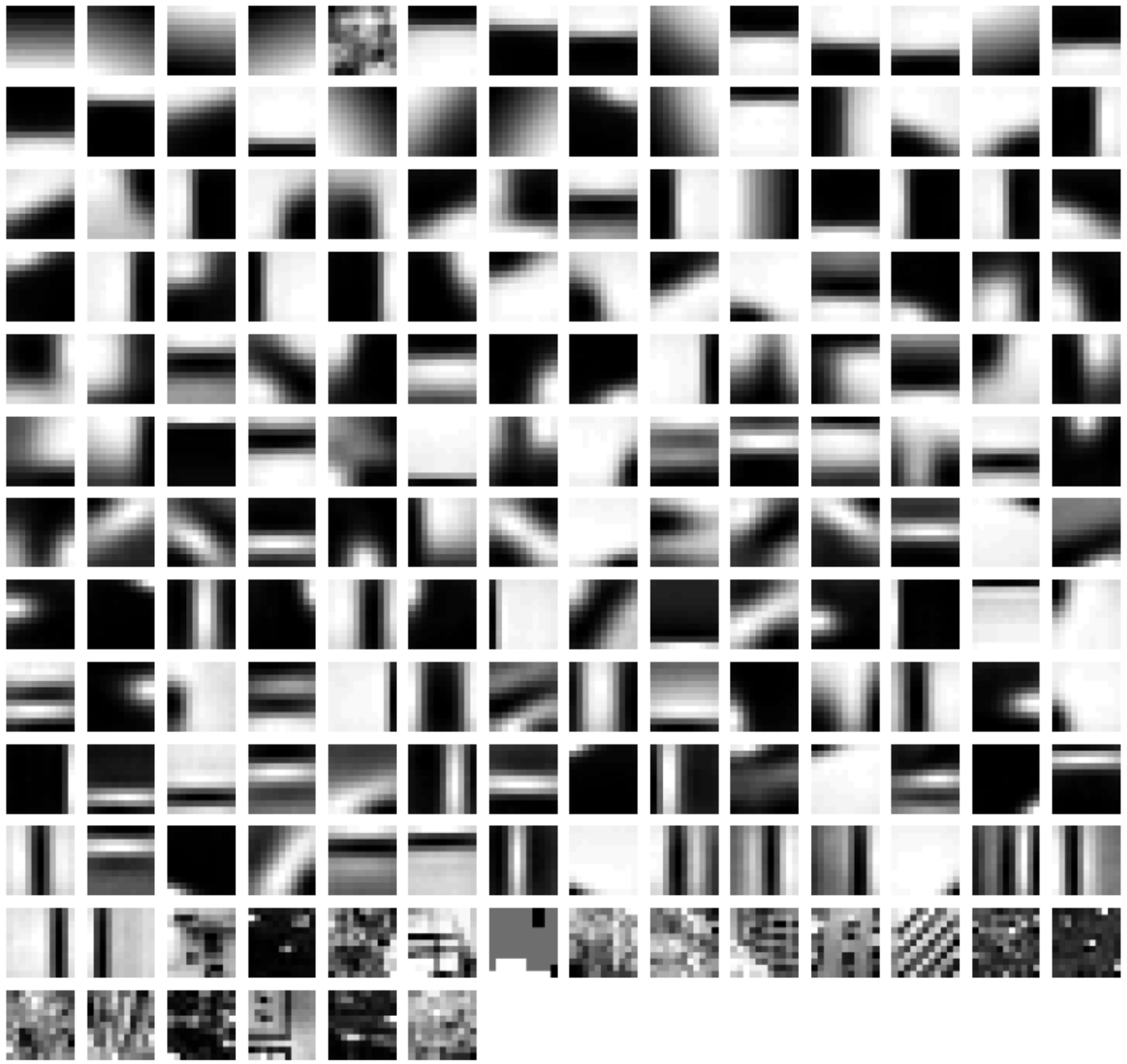
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



K-means and Hierarchical Clustering: Slide 10

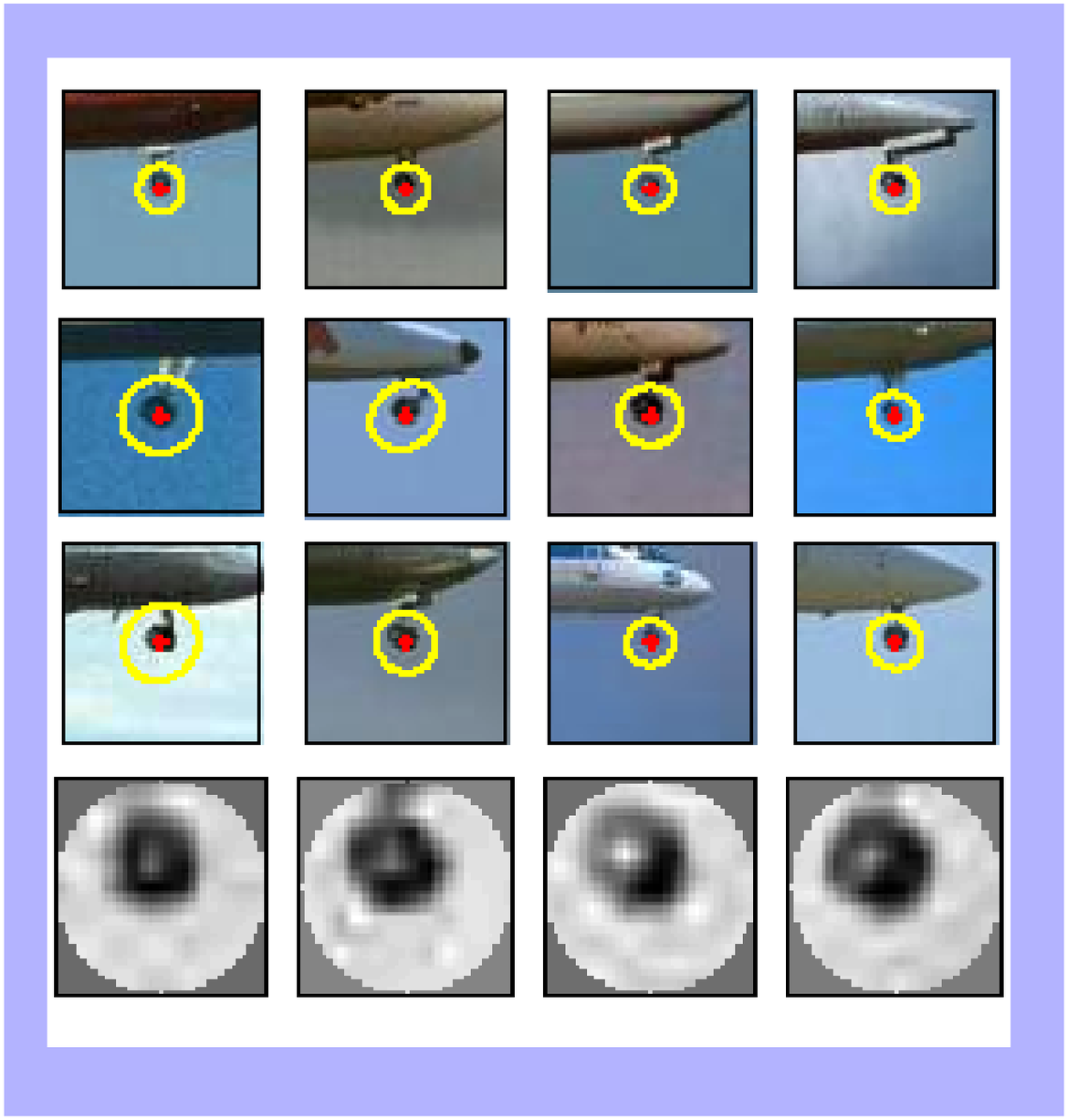
BoW-2. Codeword dictionary formation

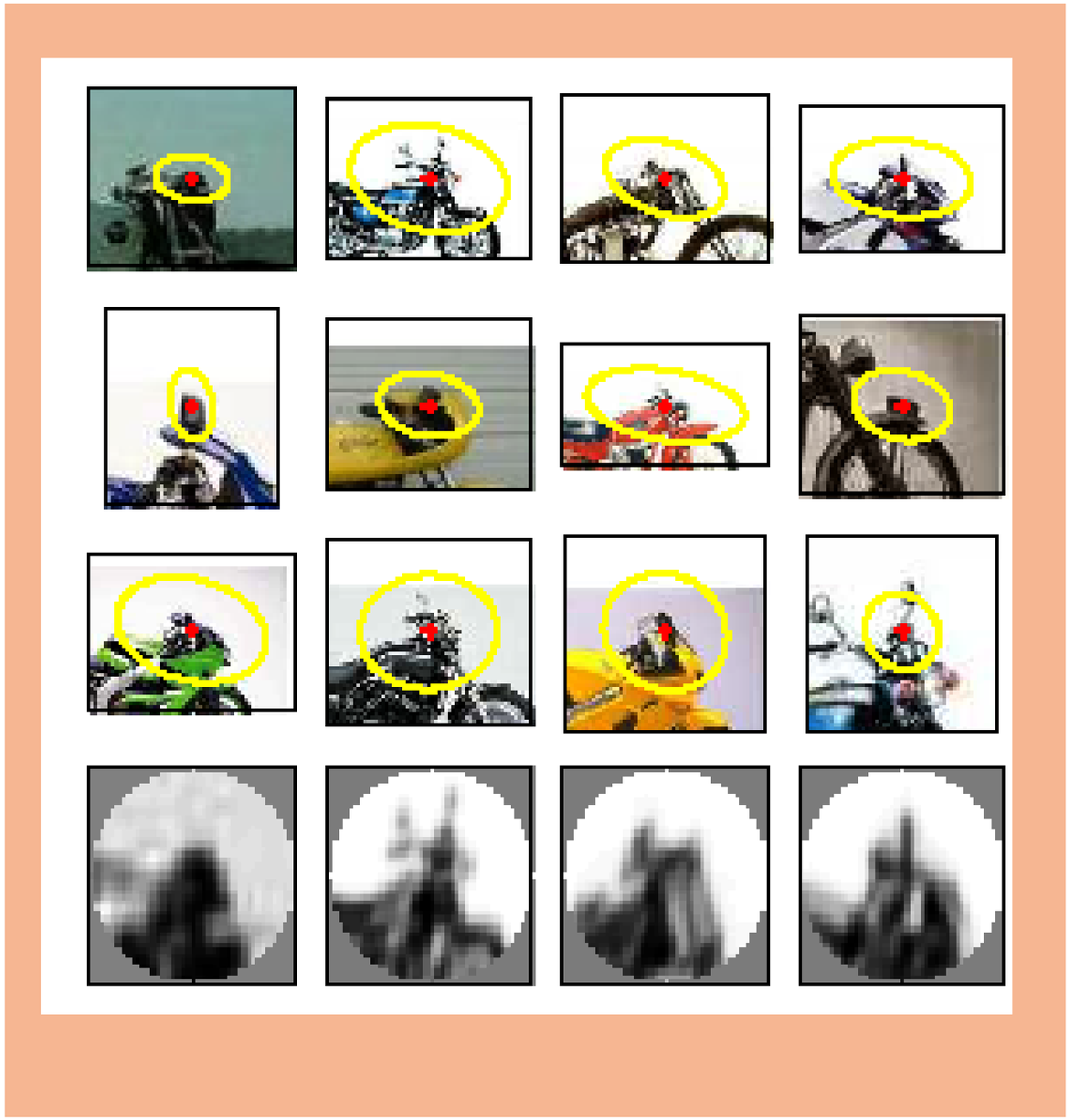
TECHNISCHE UNIVERSITÄT DARMSTADT



[Fei-Fei]

Image patch examples of codewords



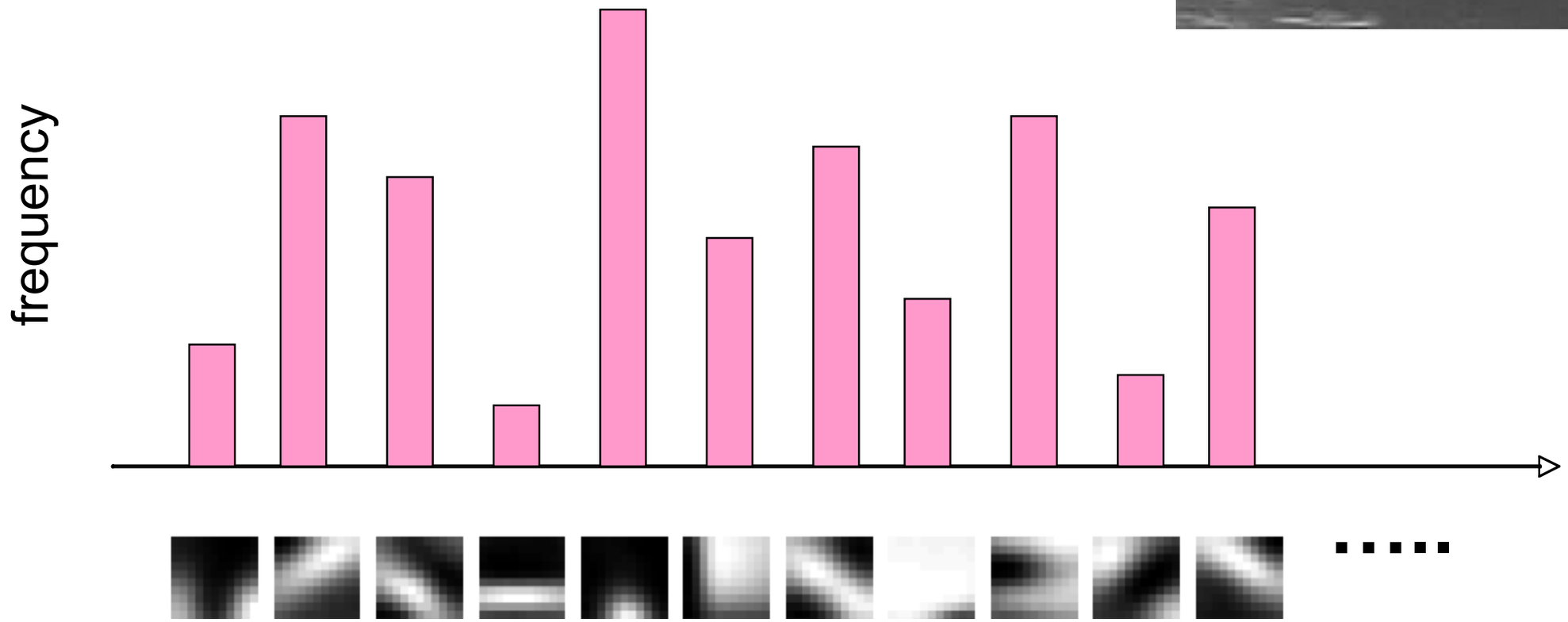


[Sivic]

BoW-3. Image representation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Histogram of features assigned to each cluster



Next: Actual Recognition