Computer Vision I

Features for Recognition – 29.05.2013

TECHNISCHE UNIVERSITÄT DARMSTADT

visual inference

Announcements

- **Exam**
  - 21. August 2013, 10:00 - 12:00
  - Please double-check that you are registered!

Today

- Recap
  - Appearance-based representations
- Object identification / instance recognition
  - Global representations
  - Bag of Visual Words
  - Color histograms
  - Receptive field histograms
- Toward local features
  - Harris interest point detector

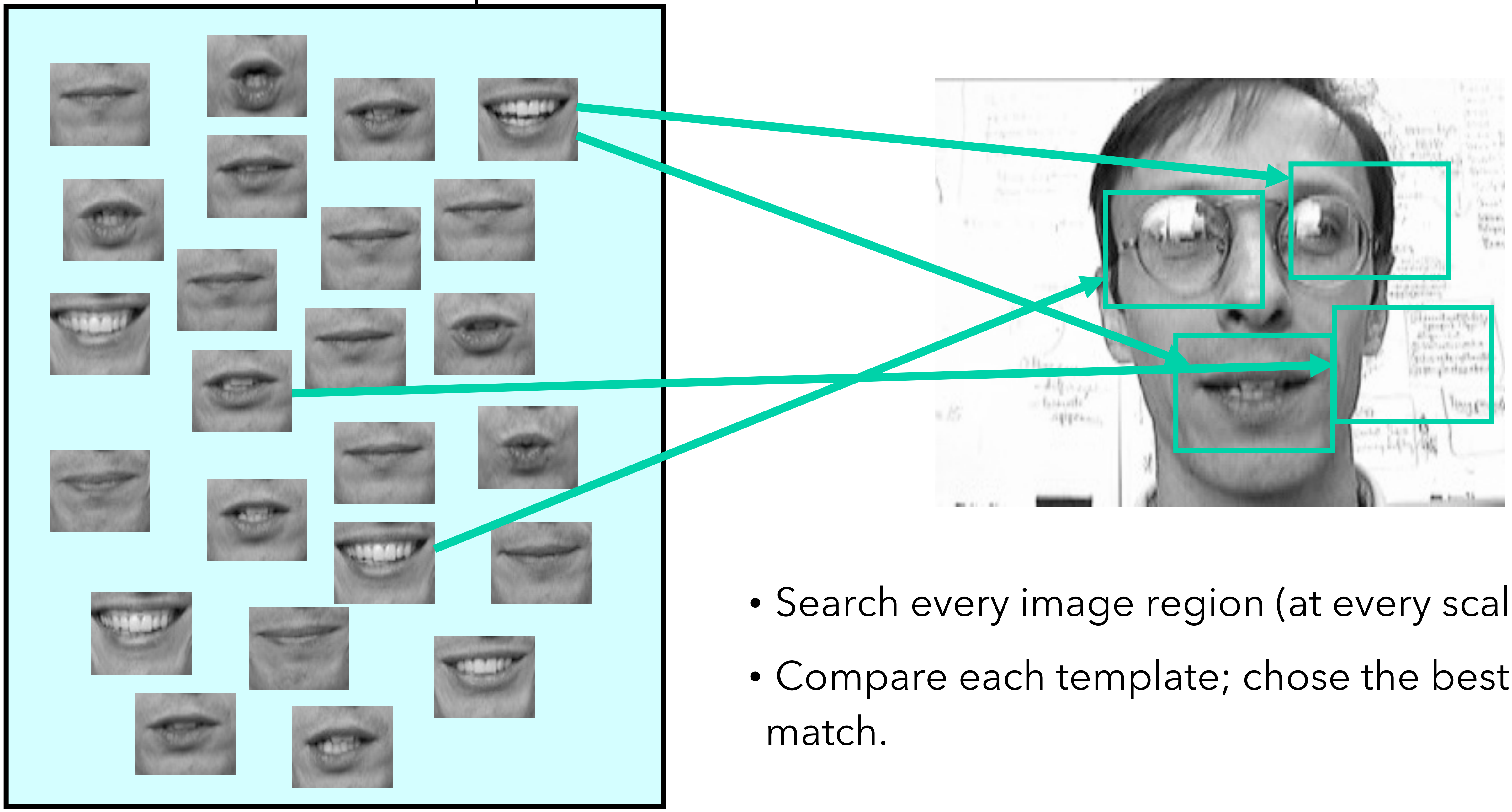
Search and Recognition

- What if we do not have a line-based object model?
  - Edges may not help us.

  - How can we find the mouth?
  - How can we recognize the “expression”?

Naïve View-Based Approach

Database of mouth "templates"



- Search every image region (at every scale).
- Compare each template; chose the best match.

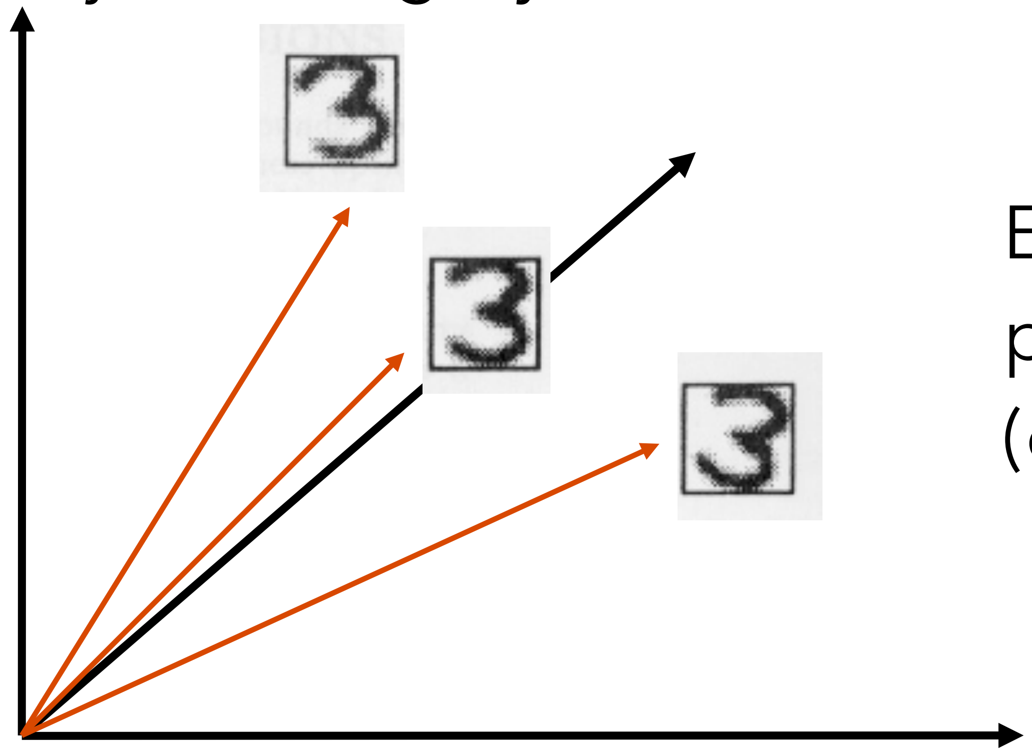
View-Based Methods

TECHNISCHE UNIVERSITÄT DARMSTADT

- Idea: Represent objects by their appearance in an ensemble of images, including different poses, illuminations, configurations of shape, …
- Approach covered here:
  - Subspace methods (also called "Eigen methods")

Subspace Methods

- How can we find more efficient representations for the ensemble of views, and more efficient methods for matching?
  - Idea: images are not random… especially images of the same object (category) have similar appearance



E.g., let images be represented as points in a high-dimensional space (e.g., one dimension per pixel)

Fleet & Szeliski

Goal

Data point n:

$$\mathbf{x}^n \in \mathbb{R}^M$$

Low-dimensional representation:

$$\mathbf{a}^n \in \mathbb{R}^D \qquad D << M$$

Mapping: $\mathbf{x}^n \longrightarrow \mathbf{a}^n$

- Restrict this mapping to be a linear function:

$$\boldsymbol{a}^n = \mathbf{B}\mathbf{x}^n, \qquad \mathbf{B} \in \mathbb{R}^{D \times M}$$

- **Note:**
  - From now on, I will use boldface symbols to denote vectors, e.g.: $\mathbf{x}, \mathbf{y}, \ldots$

Principal Component Analysis

TECHNISCHE UNIVERSITÄT DARMSTADT

- We know how we can represent our data in a lower dimensional space in a principled fashion.
  - We compute the mean of the data, and subtract it.
  - We compute the covariance matrix, decompose it, and choose a the first D eigenvalues.
  - This gives us an (eigen)basis for representing the data:

$$\mathbf{a}^n = \mathbf{B}^{\mathrm{T}}(\mathbf{x}^n - \bar{\mathbf{x}}) \qquad \text{where} \qquad \mathbf{B} = [\mathbf{u}_1, \ldots, \mathbf{u}_D]$$
$$\tilde{\mathbf{x}}^n = \bar{\mathbf{x}} + \mathbf{B}\mathbf{a}^n$$

  - It is common to also normalize the variance of each dimension.

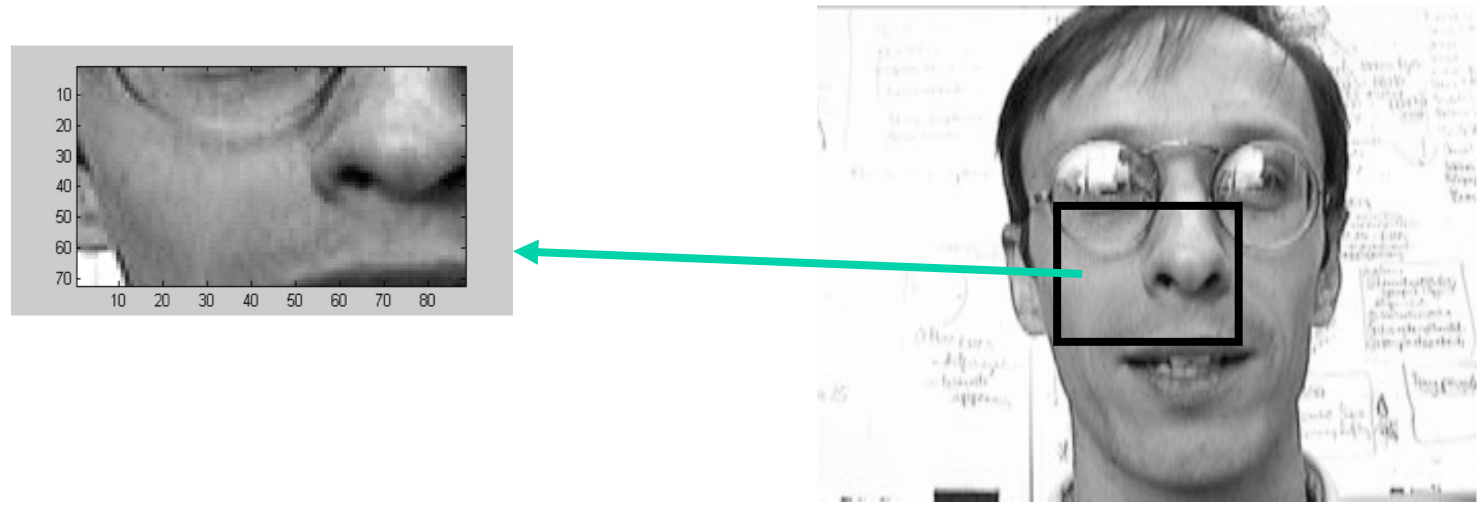
How to use SVD for PCA

- If we perform SVD on the data matrix $\mathbf{X} = [\mathbf{x}^1, \ldots, \mathbf{x}^N]$ (after subtracting the mean) then:
  - The left-singular vectors give us the eigenvectors of the covariance matrix.
  - The singular values let us easily compute the eigenvalues of the covariance matrix:

$$\lambda_i = \frac{1}{N} s_i^2$$

- **Advantage:** We never have to explicitly build and store the covariance matrix!

Simple Search Strategy



- Project each training image onto the low-dimensional subspace. Store the vectors of coefficients.
- For each image region:
  - Project it onto the low-dimensional subspace.
  - Compare this to each stored coefficient vector (cheap).
  - If the smallest distance is less than some threshold, then it is a mouth.

Recognition problems

TECHNISCHE UNIVERSITÄT DARMSTADT

- What is it?
  - Object and scene recognition
- Who is it?
  - Identity recognition
- Where is it?
  - Object detection
- What are they doing?
  - Activity recognition
- All of these are **classification** problems
  - Choose one class from a list of possible candidates

[Szeliski & Seitz]

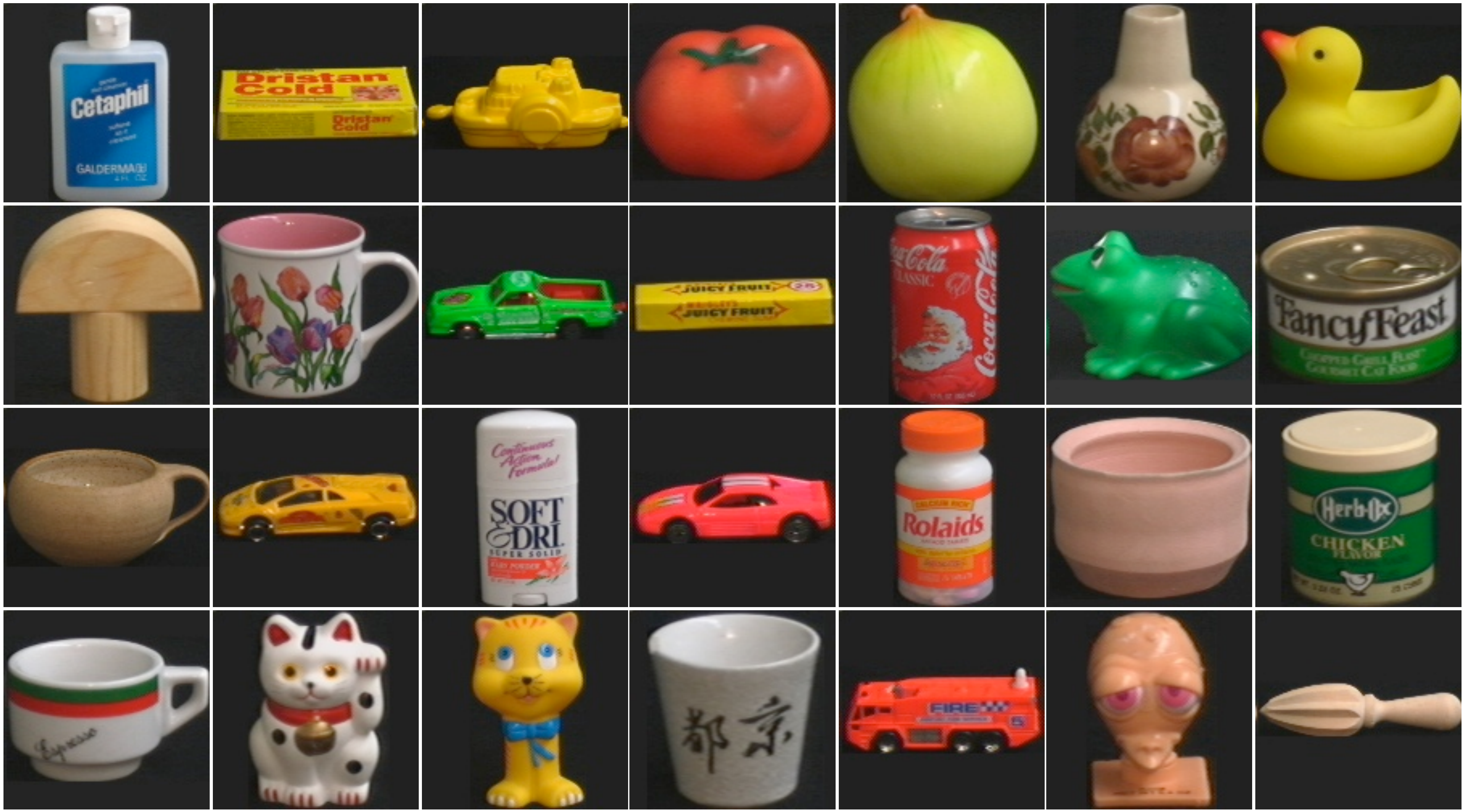
What is recognition?

TECHNISCHE UNIVERSITÄT DARMSTADT

- A different taxonomy from [Csurka et al. 2006]:
- Recognition (focus for now)
  - Where is this particular object?
- Categorization (next time)
  - What kind of object(s) is(are) present?
- Content-based image retrieval
  - Find me something that looks similar
- Detection (next time)
  - Locate all instances of a given class

[Szeliski & Seitz]

From faces and mouths to objects

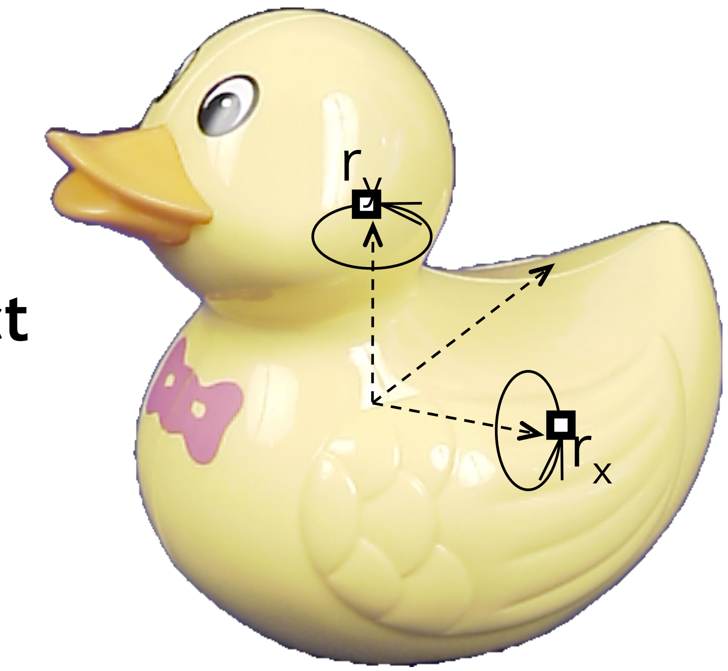


Appearance-Based Instance Recognition

TECHNISCHE UNIVERSITÄT DARMSTADT

- Basic assumption:
  - Objects can be represented by a set of images ("appearances").
  - For recognition, it is sufficient to just compare the 2D appearances.
  - No 3D model is needed.

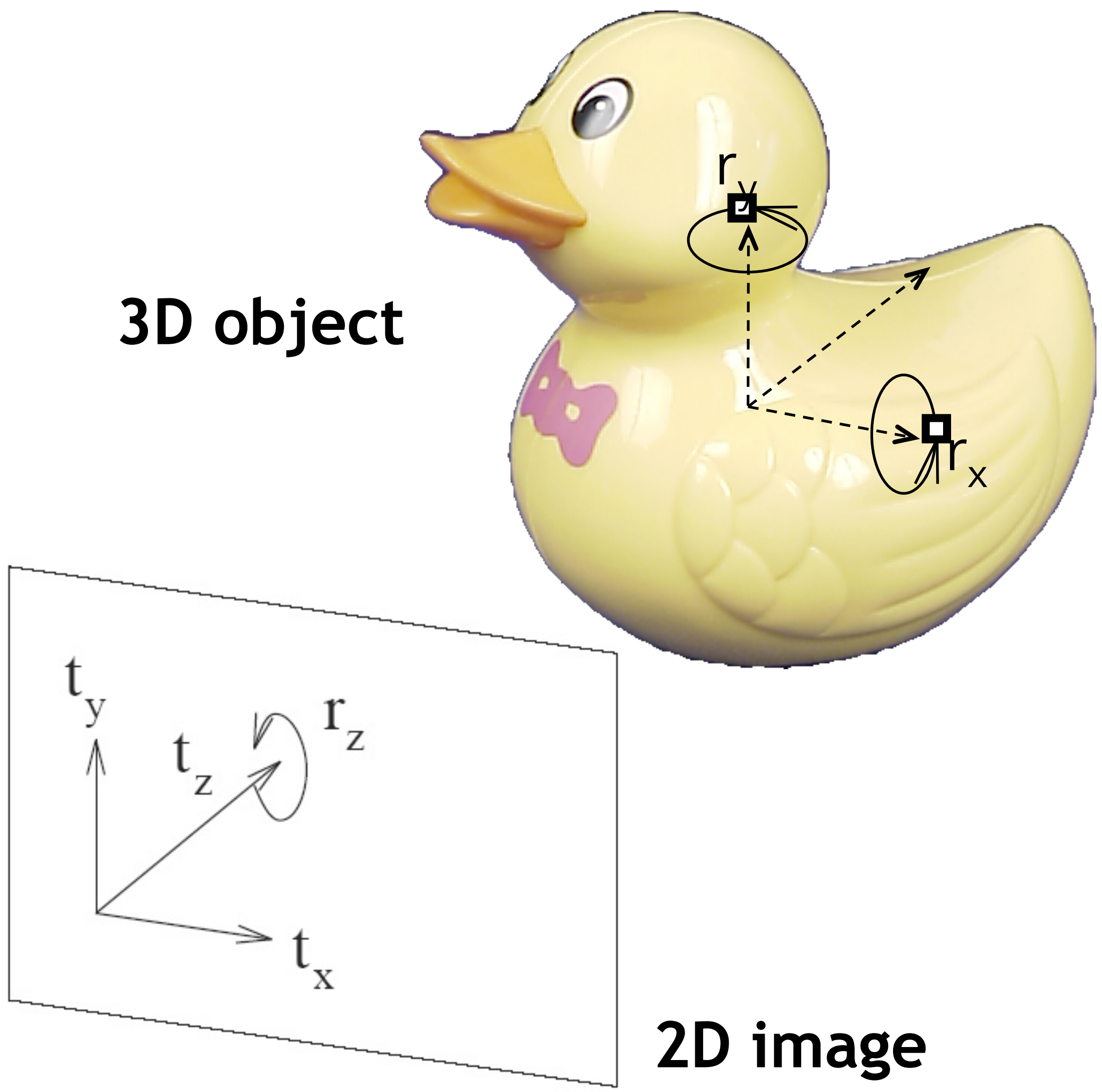
3D object



⇒ Fundamental paradigm shift in the 90's

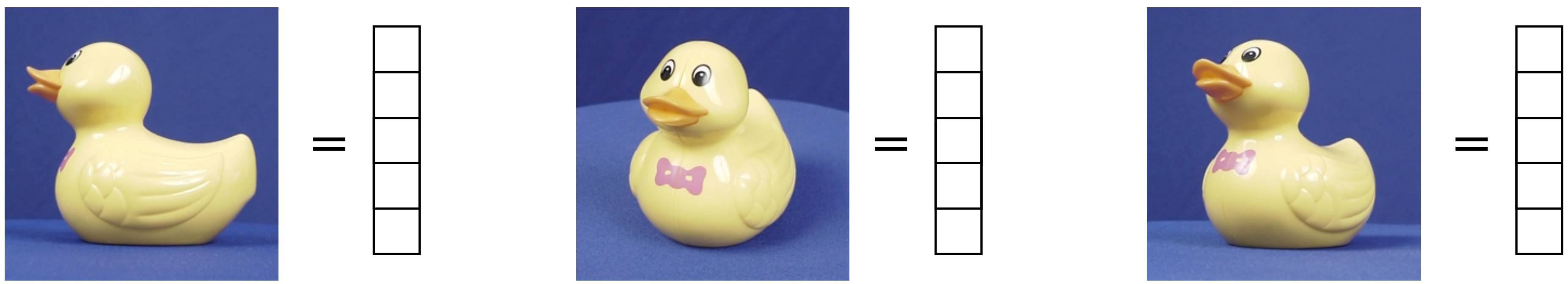
Challenges

- Viewpoint changes
  - Translation
  - Image-plane rotation
  - Scale changes
  - Out-of-plane rotation
- Illumination
- Clutter
- Occlusion
- Noise



Global Representation

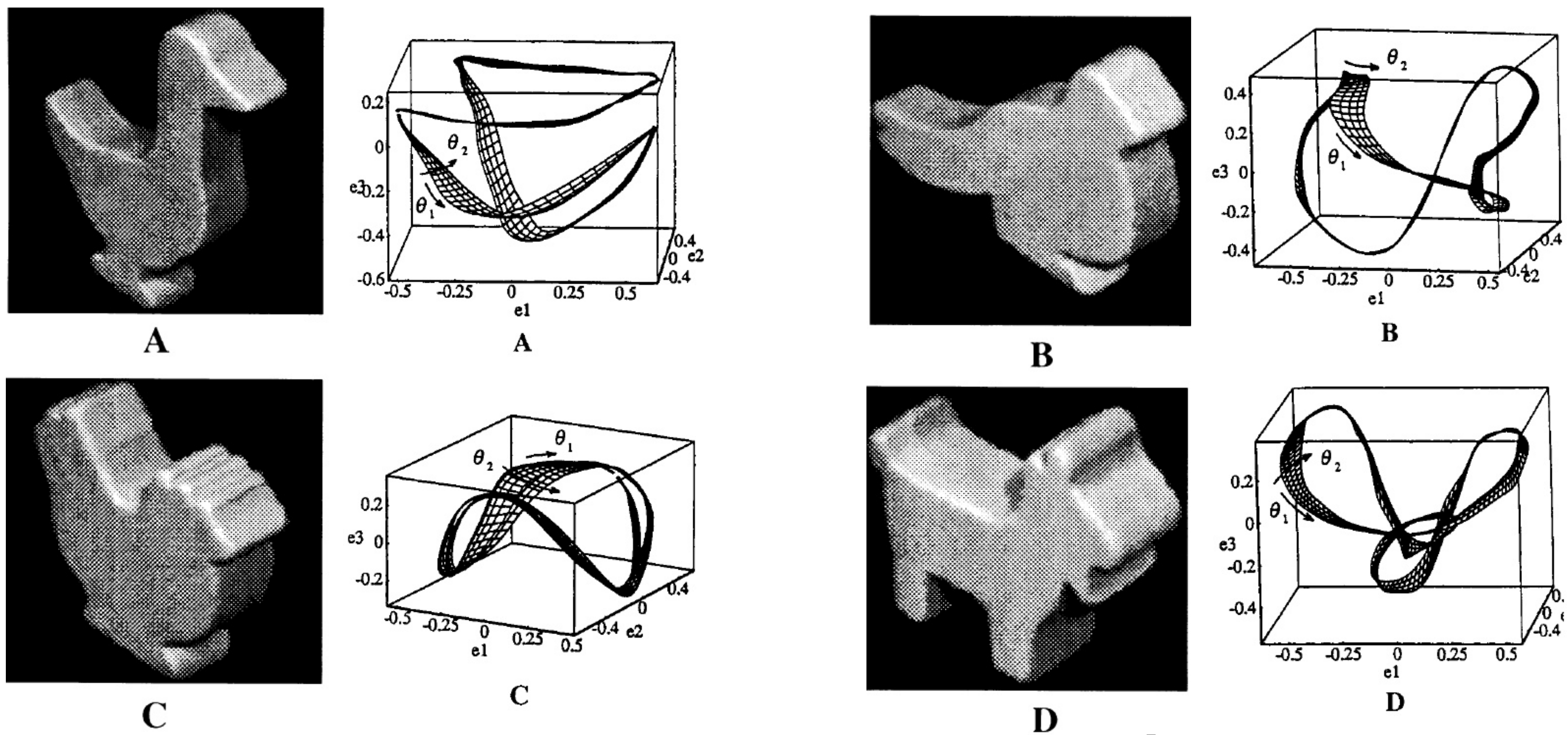
- Idea:
  - Represent each object (view) by a global feature descriptor.



  - For recognizing objects, just match the (global) descriptors.
  - Some modes of variation are built into the descriptor, others have to be incorporated in the training data or the recognition process.
  - View-based representations:
    - Pixels (or projections onto global basis vectors) are the descriptor.
    - Very limited amount of invariance!

Appearance Manifolds

- Limitations of linear representations:



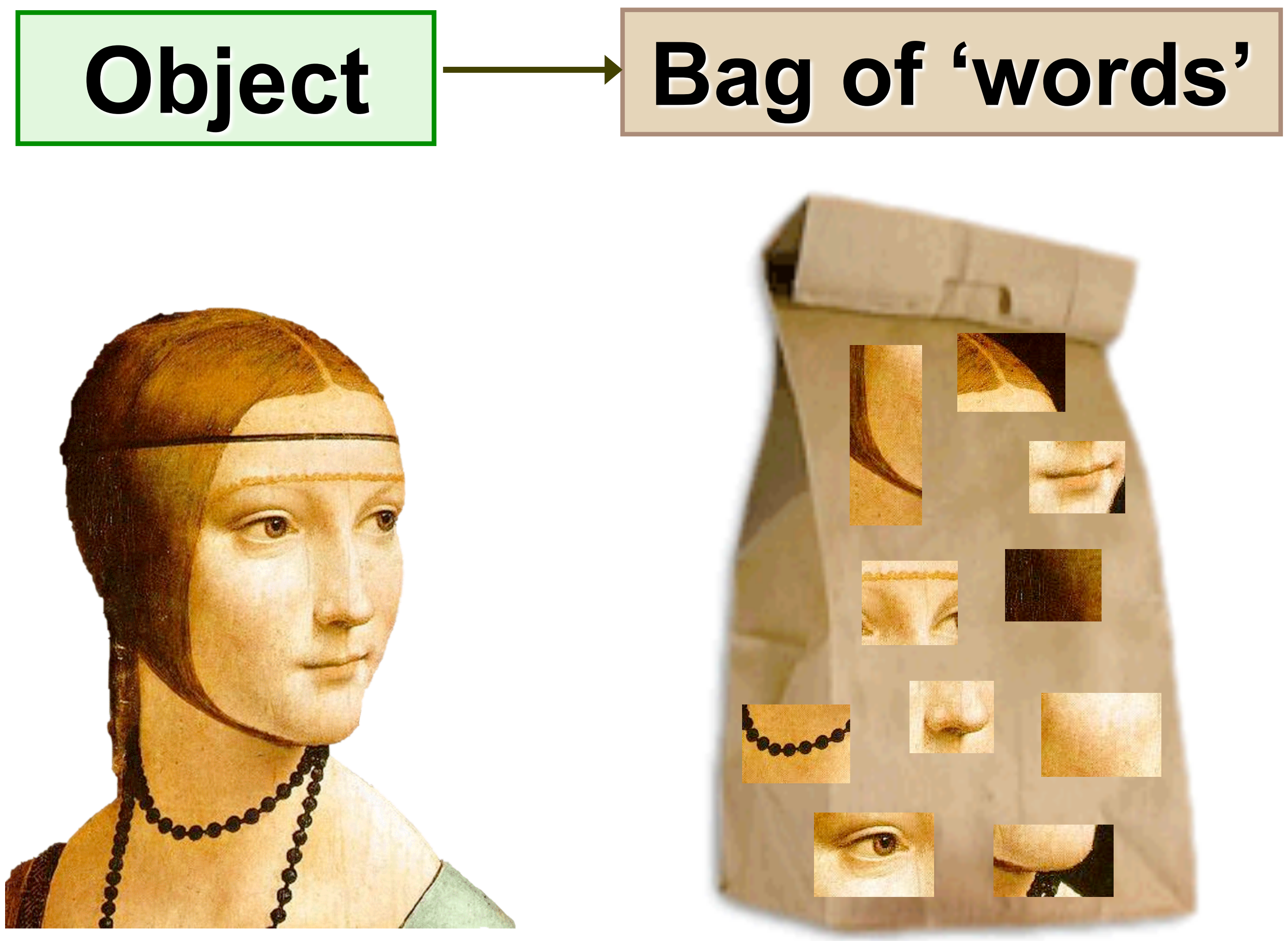
[Murase & Nayar, 1996]

- PCA will not work!

View-Based Approaches

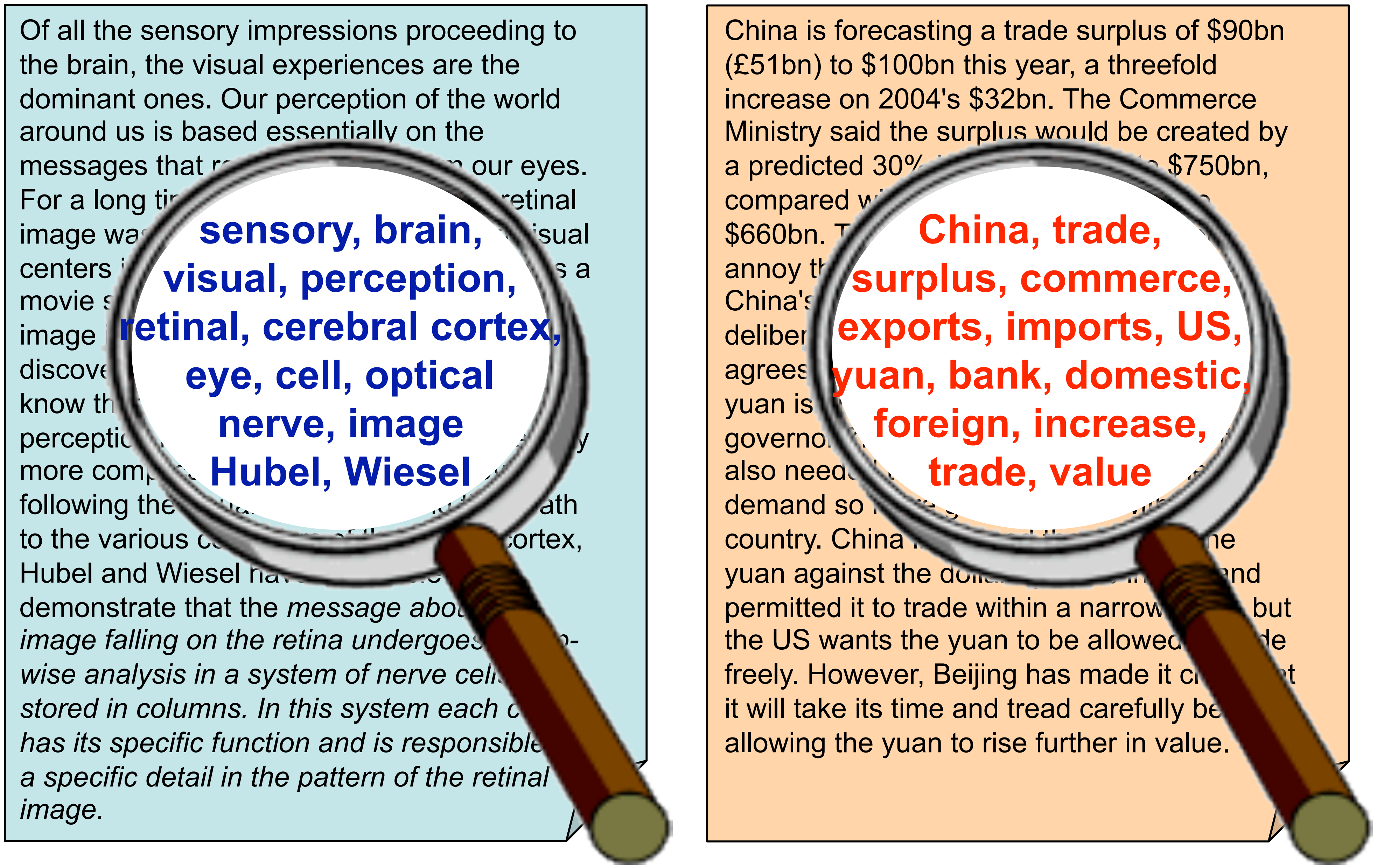
- ... are severely challenged by these common variations
  - To make them work, we would need an unmanageable amount of examples (training data).
- ... do not generalize well
  - Almost any variation that hasn't been captured in the training data will not be handled gracefully.
- Training data is expensive:
  - Humans have to gather and label it.
- What else can we do?
  - Move away from representing the object simply by its pixels.
  - Images as representation are **too rigid**.

"Bag of words" Model



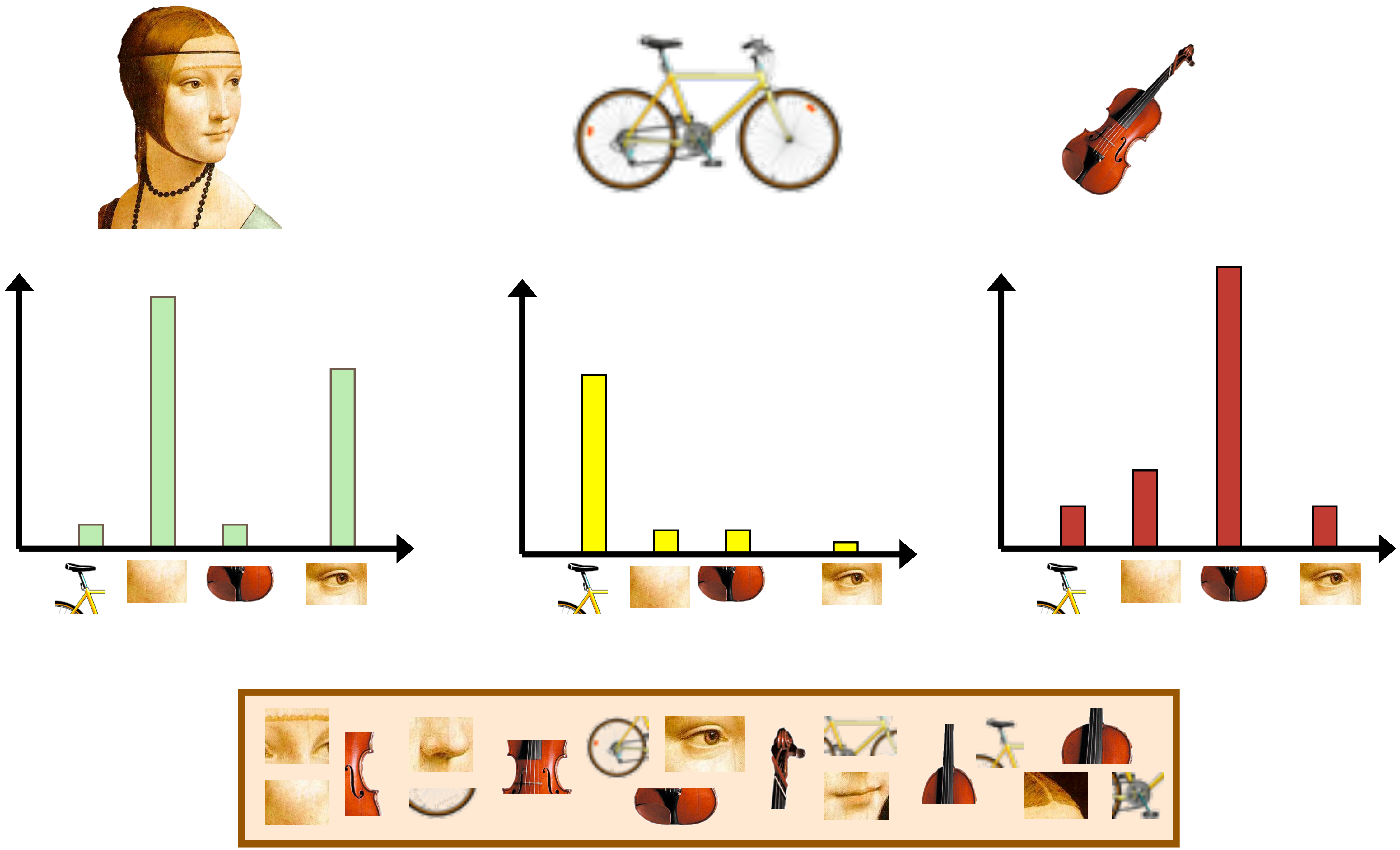
[Fei-Fei]

Analogy to Documents



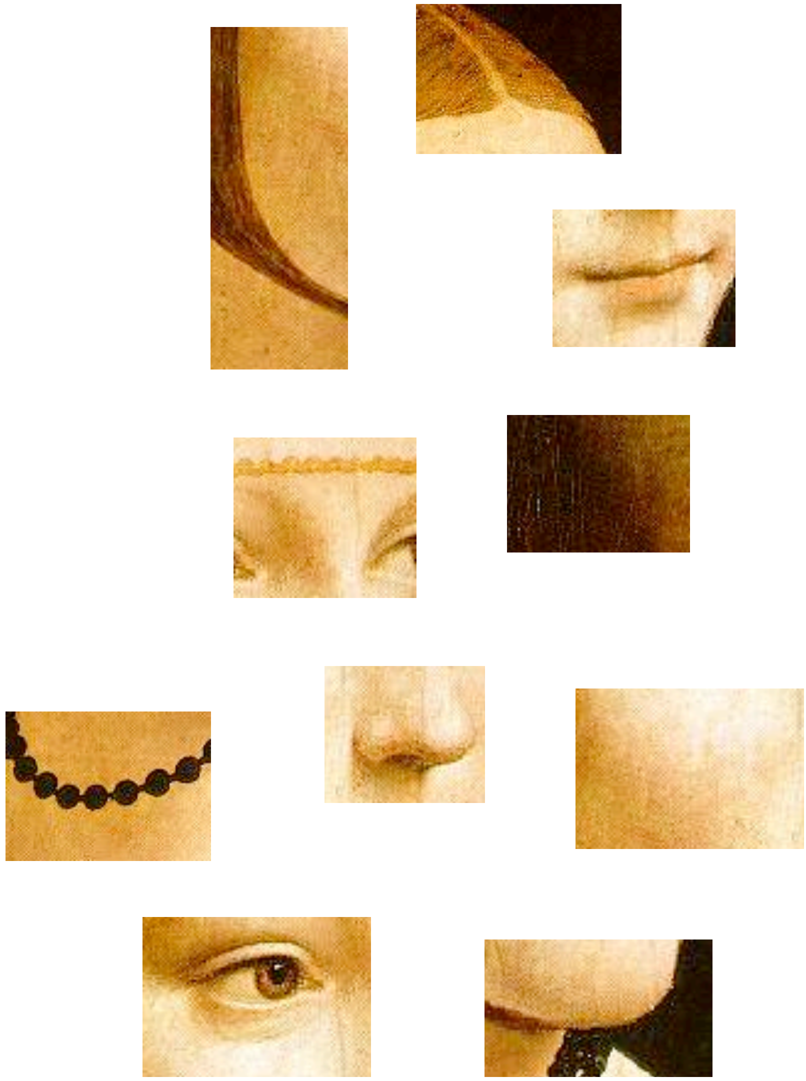
[Fei-Fei]

Visual word distributions



[Fei-Fei]

Which feature representation?



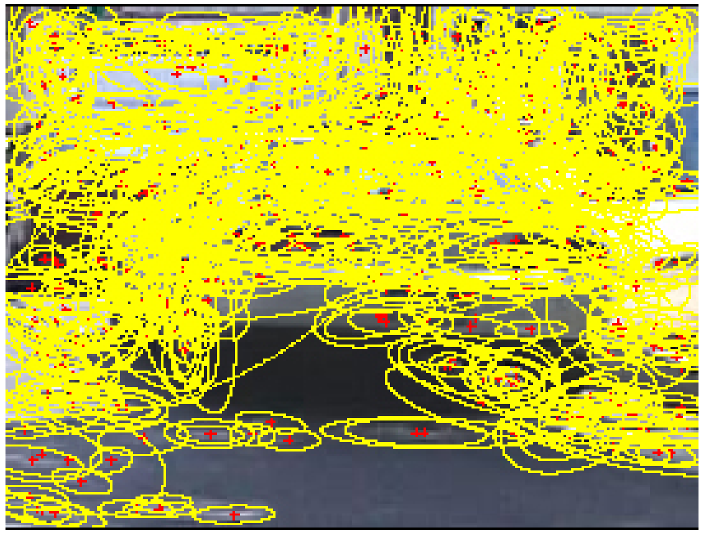
[Fei-Fei]

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



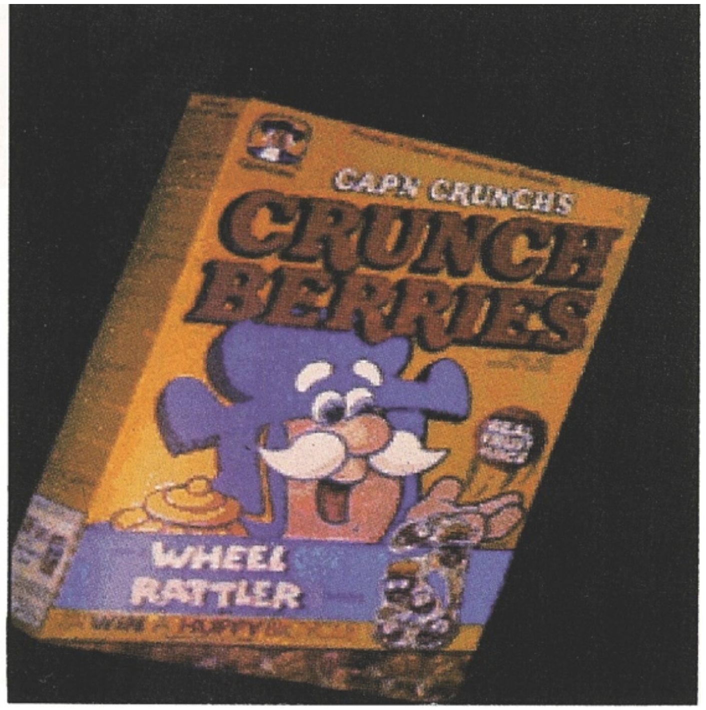
[Fei-Fei]

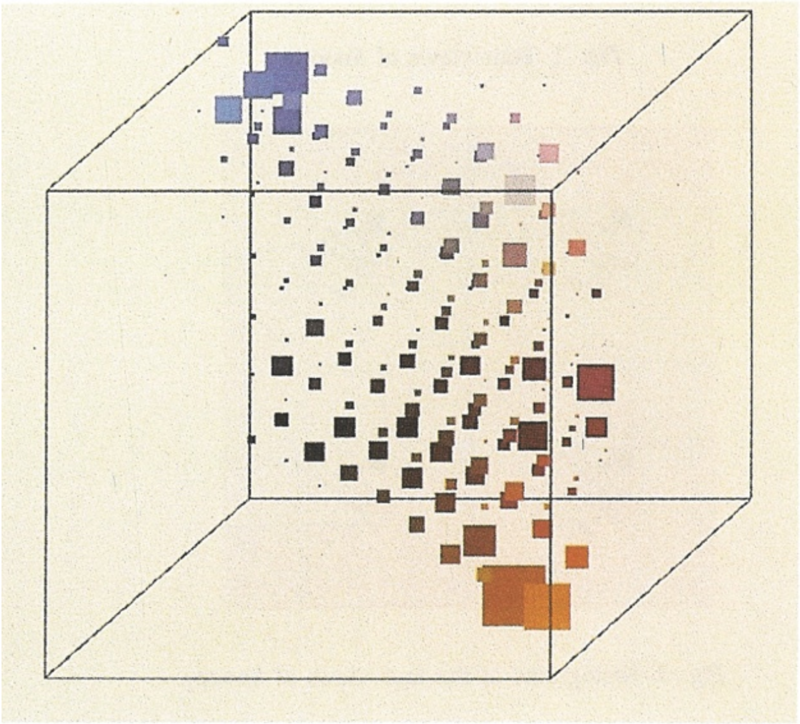
Simplest "visual word": Color

- From rigid to very non-rigid representations:
  - Forget any spatial information!
- Color:
  - Color stays constant under geometric transformations
  - Local feature
    - Color is defined for each pixel
    - Robust to partial occlusion
- Idea
  - Directly use object colors for (instance) recognition
  - Better: use statistics of object colors

Color Histograms

- Color statistics
  - Given: tristimulus R,G,B for each pixel
  - Compute 3D histogram
    - H(R,G,B) = #(pixels with color (R,G,B))

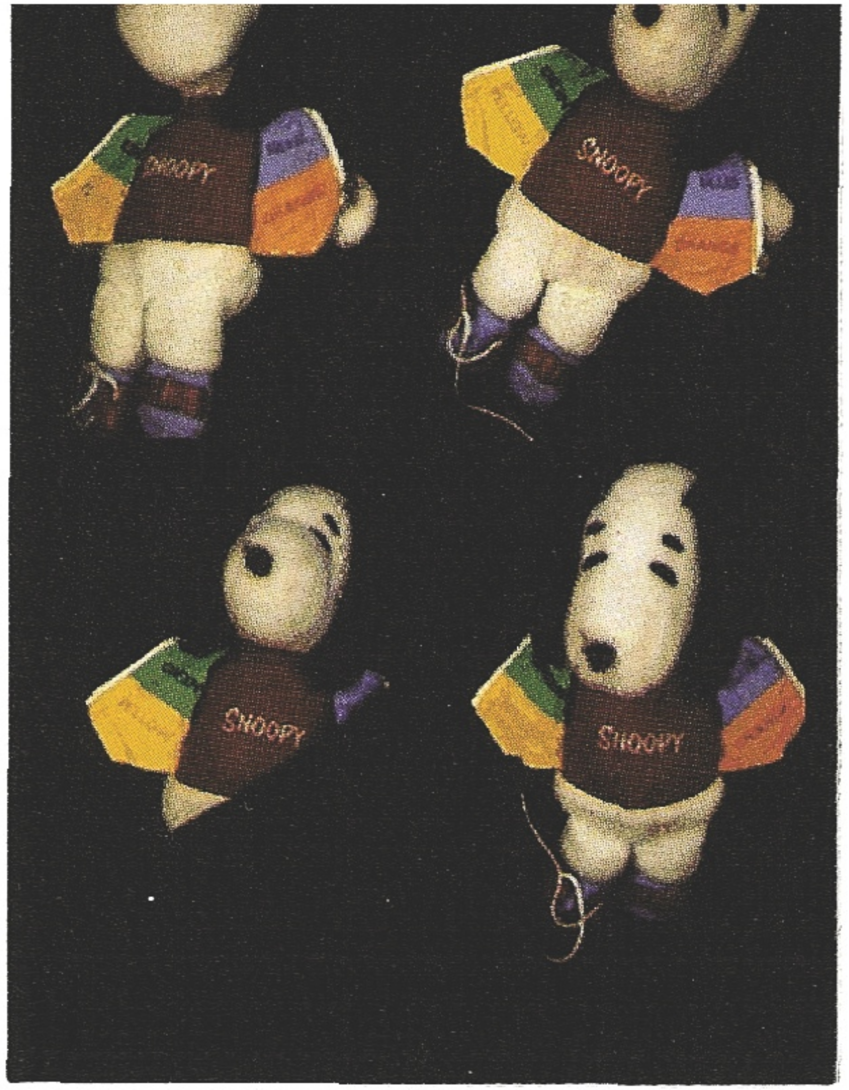


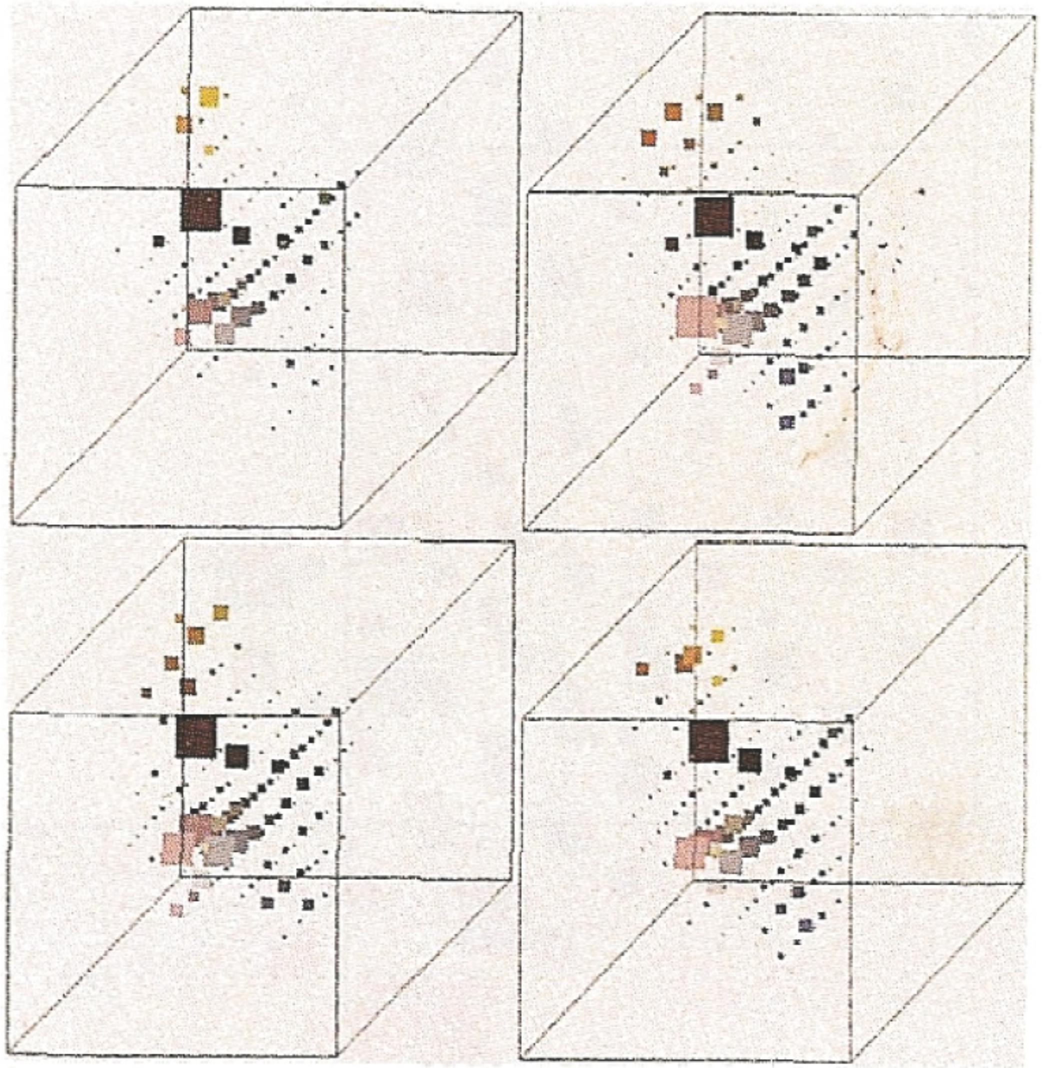


[Swain & Ballard, 1991]

Color Histograms

- Relatively robust representation:
  - Presence of occlusion, rotation





[Swain & Ballard, 1991]

Intensity normalization

TECHNISCHE UNIVERSITÄT DARMSTADT

- One component of the 3D color space is the intensity
  - If a color vector is multiplied by a scalar, the intensity changes, but not the color itself.
  - This means colors can be normalized by the intensity.
    - Intensity is given by: I = R + G + B:
  - „Chromatic representation“
  - Invariant to changes of the light intensity.

$$r = \frac{R}{R+G+B}$$

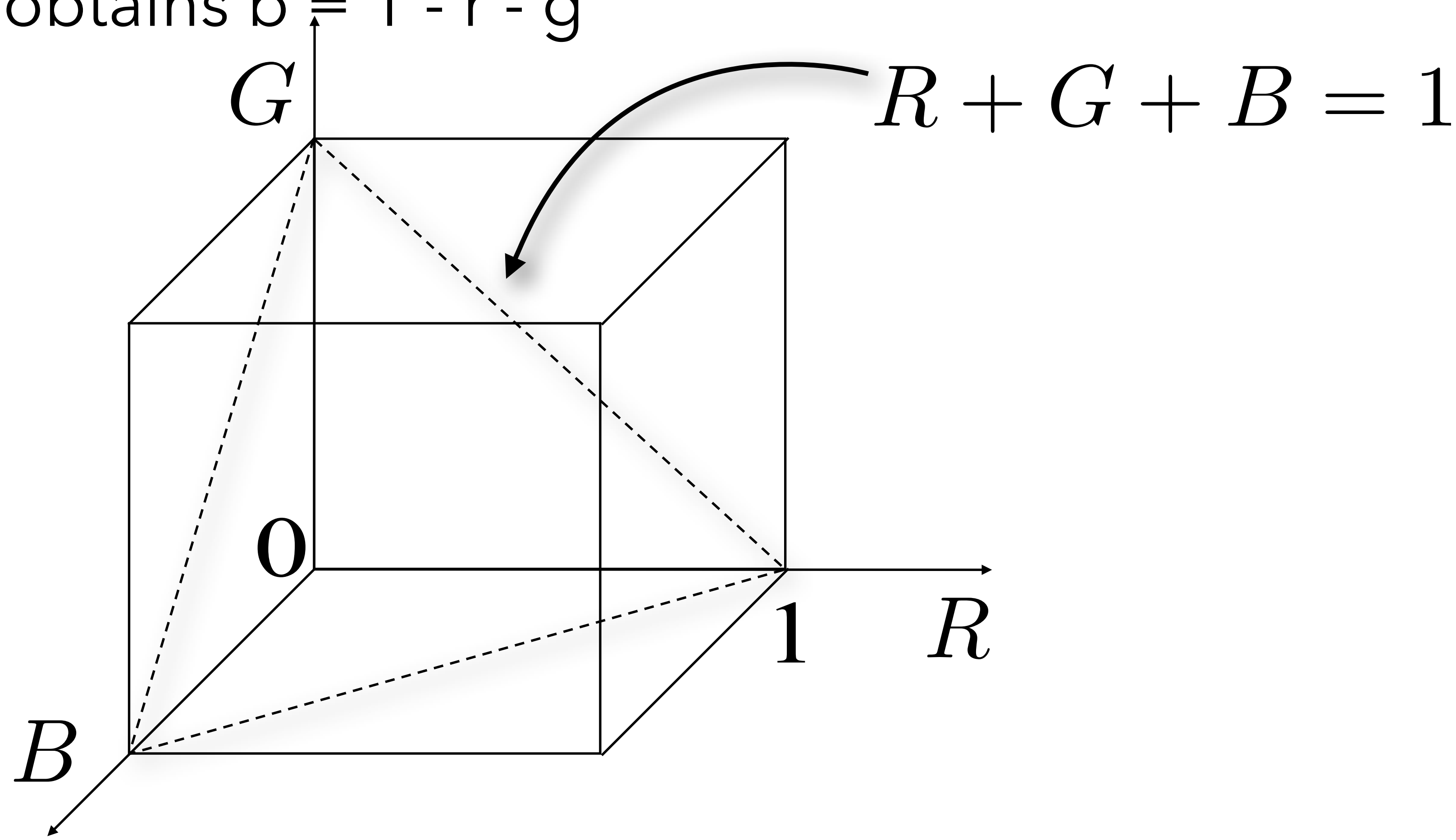
$$g = \frac{G}{R+G+B}$$

$$b = \frac{B}{R+G+B}$$

Color

- Observation:
  - Since r + g + b = 1, only 2 parameters are necessary
  - E.g. one can use r and g
  - and obtains b = 1 - r - g

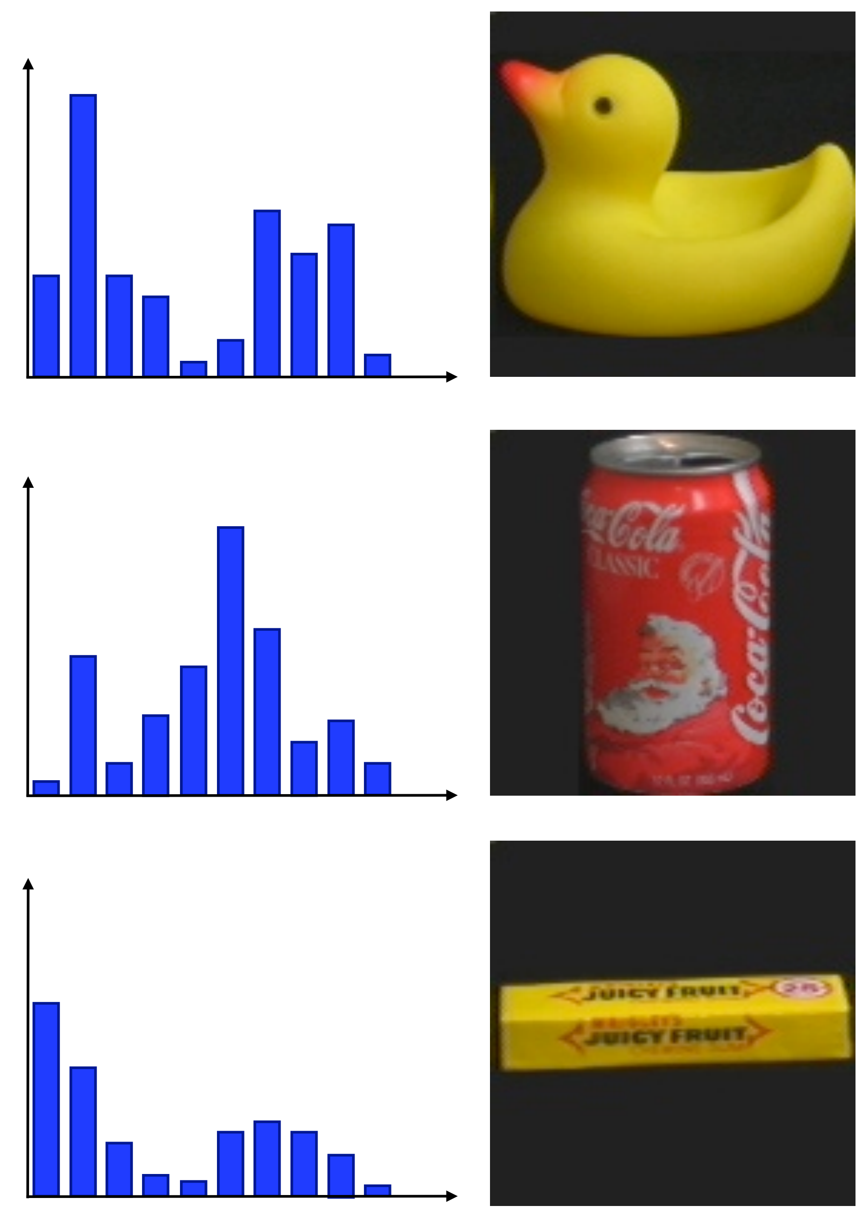
$$r + g + b = 1$$
$$\Rightarrow b = 1 - r - g$$

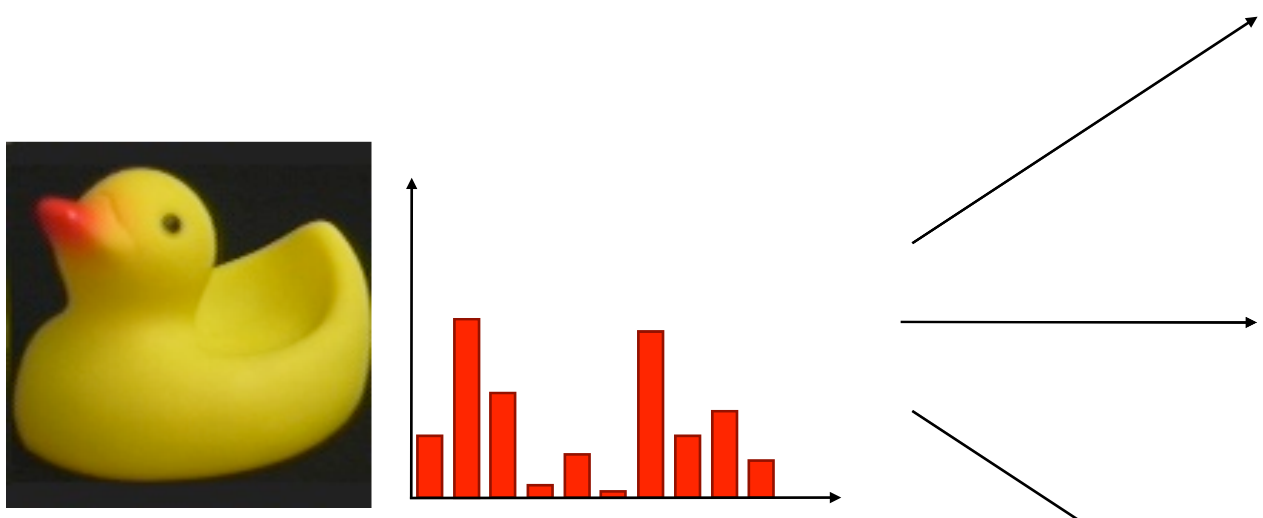


Recognition using Histograms

- Histogram comparison
  - Database of known objects
  - Test image of unknown object

known objects

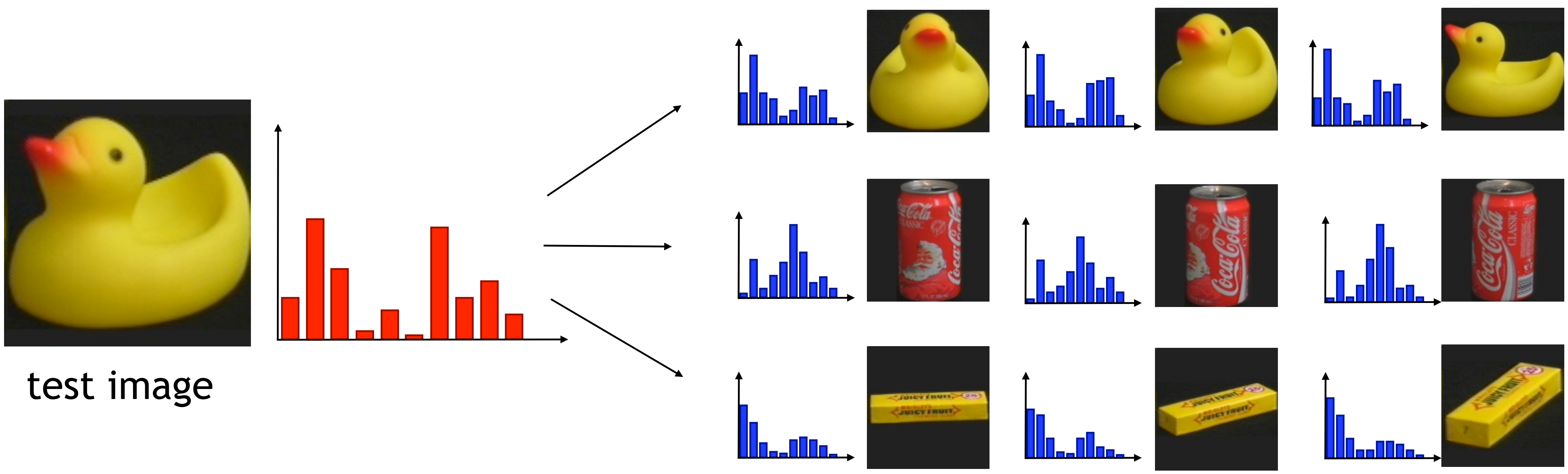




test image

Recognition using Histograms

- Database with multiple training views per object

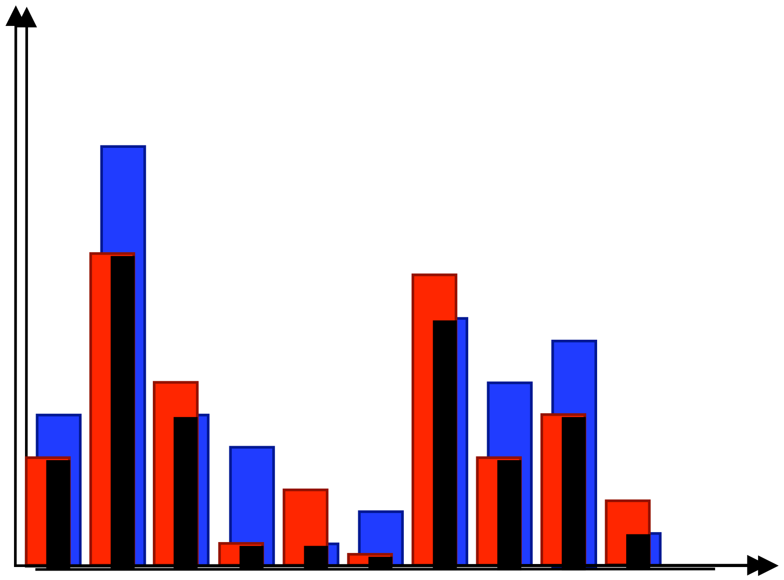


Histogram Comparison

TECHNISCHE UNIVERSITÄT DARMSTADT

- Comparison measures
  - Histogram intersection

$$\cap(Q,V) = \sum_i \min(q_i, v_i)$$



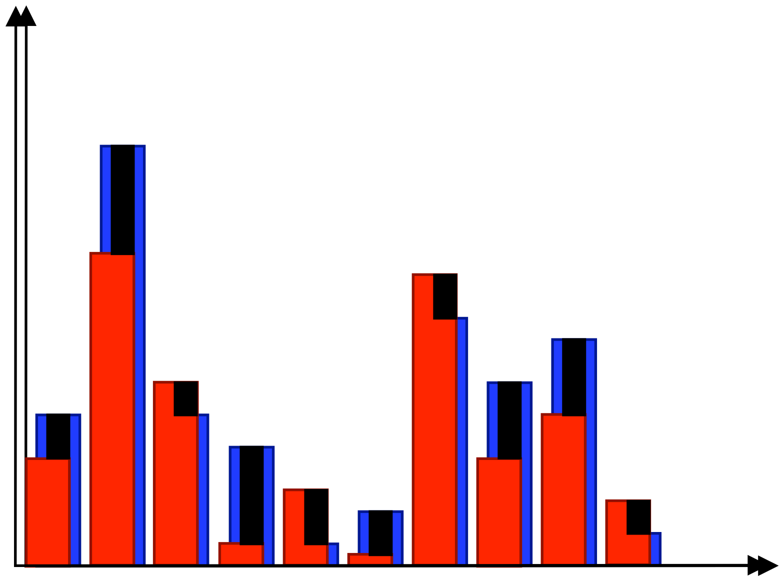
- Motivation:
  - Measures the common part of both histograms.
  - Range: [0,1]
  - For unnormalized histograms, use

$$\cap(Q,V) = \frac{1}{2}\left(\frac{\sum_i \min(q_i, v_i)}{\sum_i q_i} + \frac{\sum_i \min(q_i, v_i)}{\sum_i v_i}\right)$$

Histogram Comparison

- Comparison measures
  - Euclidean distance

$$d(Q,V) = \sum_i (q_i - v_i)^2$$



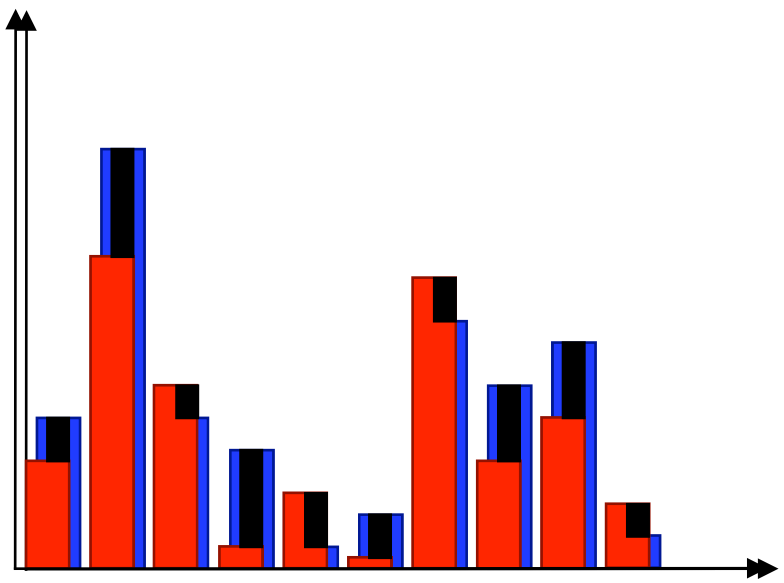
- Motivation:
  - Focuses on the differences between the histograms.
  - Range: [0,∞]
  - All cells are weighted equally.
  - Not very discriminative.

Histogram Comparison

TECHNISCHE UNIVERSITÄT DARMSTADT

- Comparison measures
  - Chi-square

$$\chi^2(Q,V) = \sum_i \frac{(q_i - v_i)^2}{q_i + v_i}$$



- Motivation:
  - Statistical background:
    - Test if two distributions are different.
    - Possible to compute a significance score.
  - Range: $[0,\infty]$
  - Cells are not weighted equally!
    - therefore more discriminative
    - may have problems with outliers (therefore assume that each cell contains at least a minimum of samples)

Histogram Comparison

TECHNISCHE UNIVERSITÄT DARMSTADT

- Which measure is best?
  - Depends on the application…
  - Both intersection and χ^2 give good performance.
    - Intersection is a bit more robust.
    - χ^2 is a bit more discriminative.
  - Euclidean distance is not robust enough.
- There exist many other measures:
  - Bhattacharyya distance
  - Information theoretic: Kullback-Leiber divergence, ...

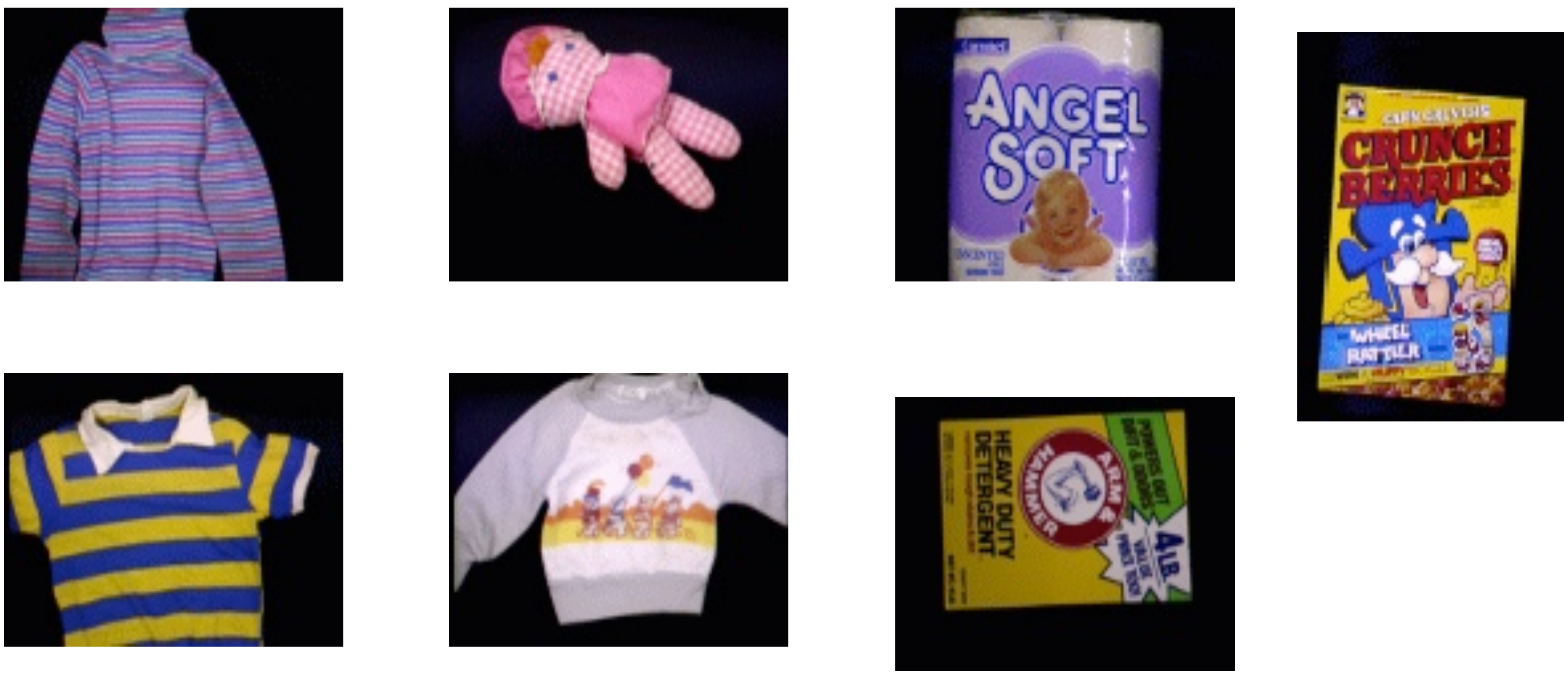
Recognition using Histograms

- Simple algorithm:
  1. Build a set of histograms $H = \{M_1, M_2, M_3, \ldots\}$ for each known object
     - More precisely, for each view of each object.
  2. Build a histogram T for the test image.
  3. Compare T to each $M_k \in H$
     - Using a suitable comparison measure.
  4. Select the object with the best matching score
     - Or reject the test image if no object is similar enough.

“Nearest-Neighbor” strategy

Color Histograms

- Recognition
  - Works surprisingly well
  - In the first paper (1991), 66 objects could be recognized almost without errors



[Swain & Ballard, 1991]

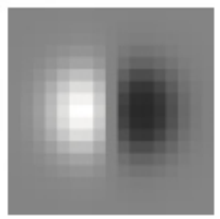
Discussion: Color Histograms

- Advantages:
  - Invariant to object translations
  - Invariant to image rotations
  - Slowly changing for out-of-plane rotations
  - No perfect segmentation necessary
  - Histograms change gradually when part of the object is occluded
  - Possible to recognize deformable objects
- Problems:
  - The pixel colors change with the illumination („color constancy problem“)
    - Intensity
    - Spectral composition (illumination color)
  - Not all objects can be identified by their color distribution.

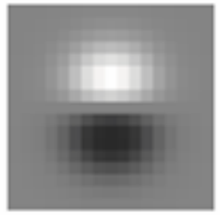
Generalization of the Idea

- Histograms of derivatives

D_x

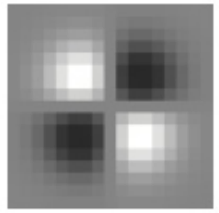


D_y

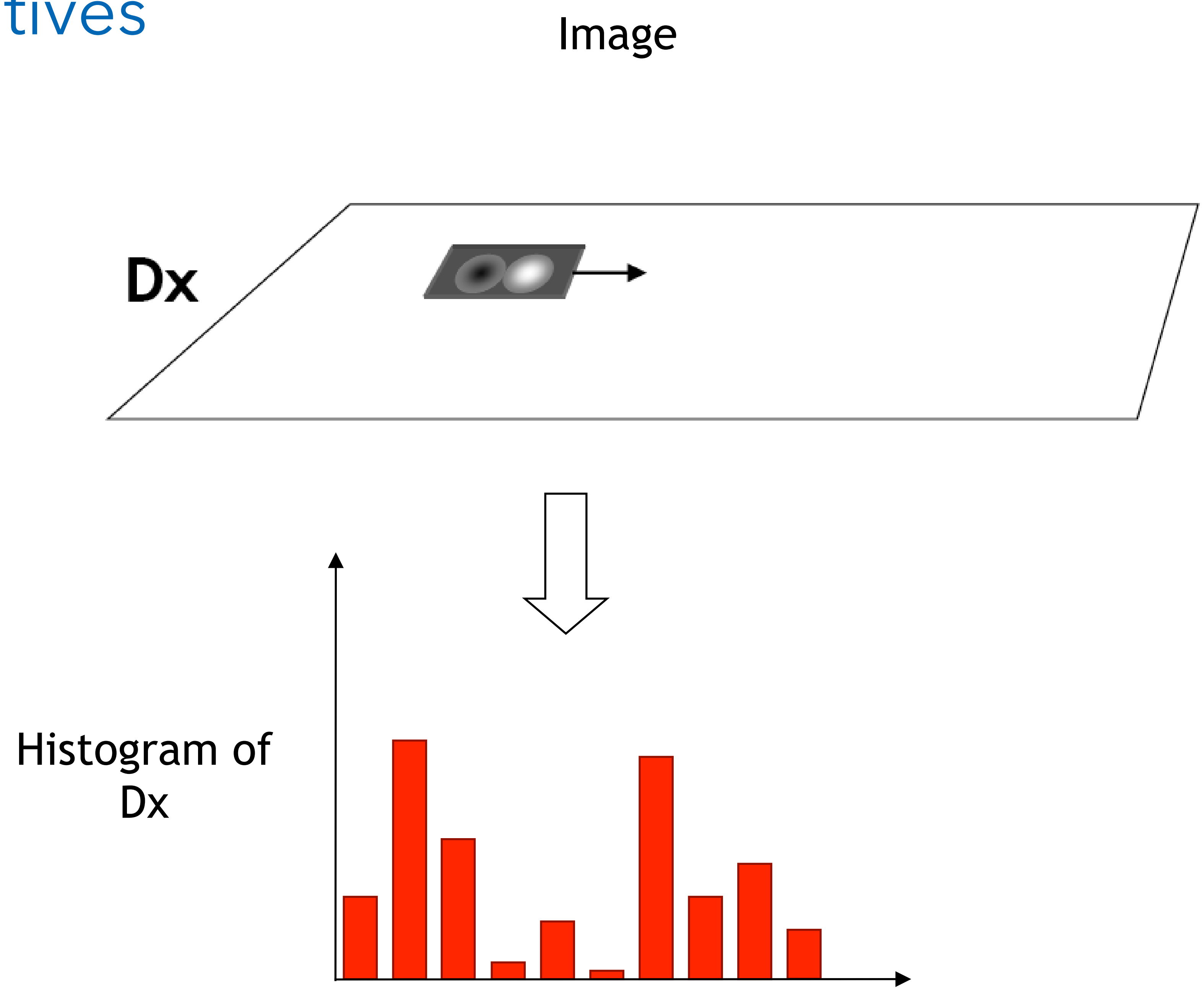


D_{xx}

D_{xy}



D_{yy}

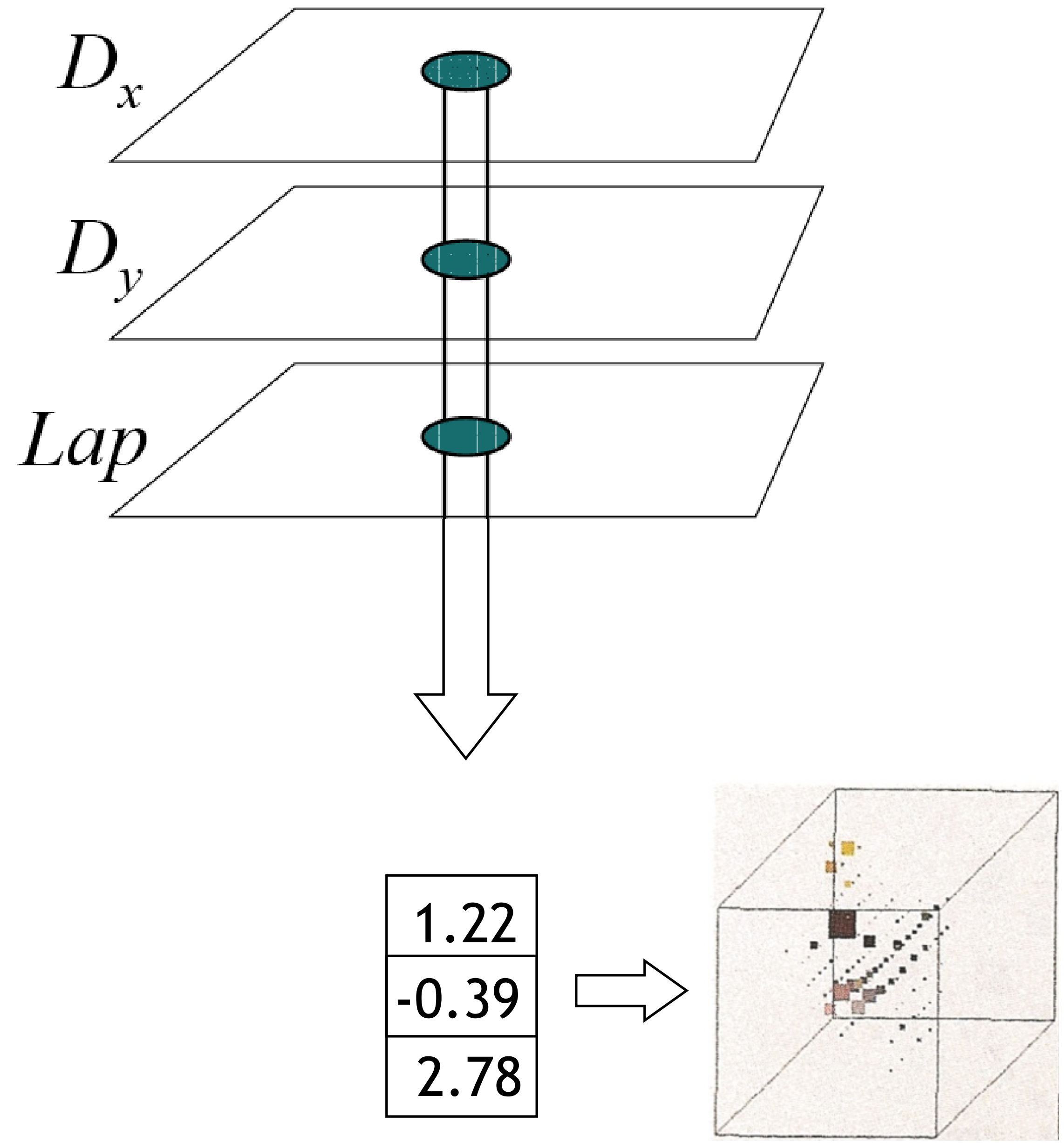


Generalization of the Idea

- General receptive field histograms
  - Any local descriptor (e.g. filter, filter combination) can be used to build a histogram.
  - Examples:
    - Gradient magnitude $$\mathrm{Mag} = \sqrt{D_x^2 + D_y^2}$$
    - Gradient direction $$\mathrm{Dir} = \arctan \frac{D_y}{D_x}$$
    - Laplacian $$\mathrm{Lap} = D_{xx} + D_{yy}$$

Multidimensional Histograms

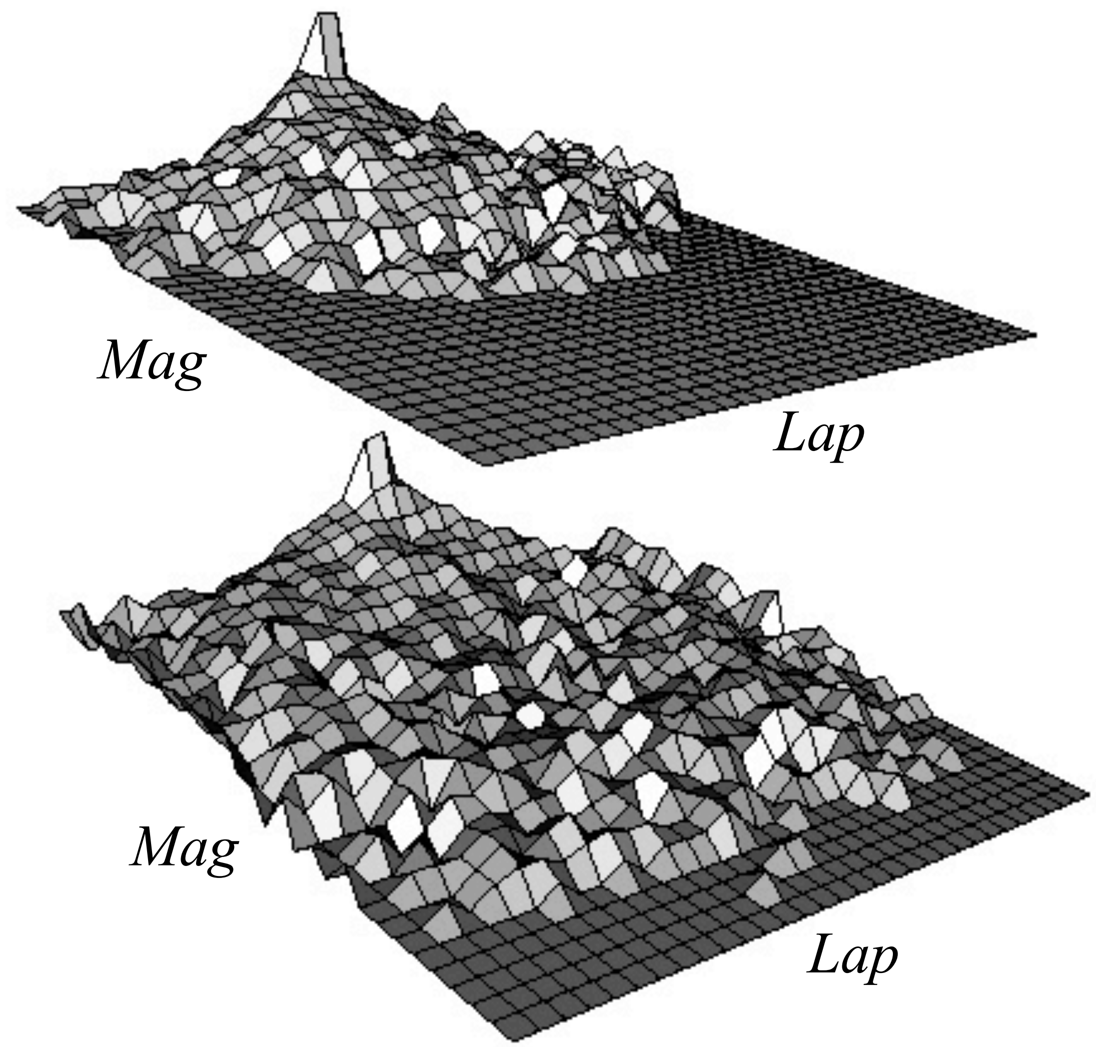
- Combination of several descriptors
  - Each descriptor is applied to the whole image.
  - Corresponding pixel values are combined into one feature vector.
  - Feature vectors are collected in a multidimensional histogram.



Multidimensional Histograms

- Examples:

[Schiele & Crowley, 2000]



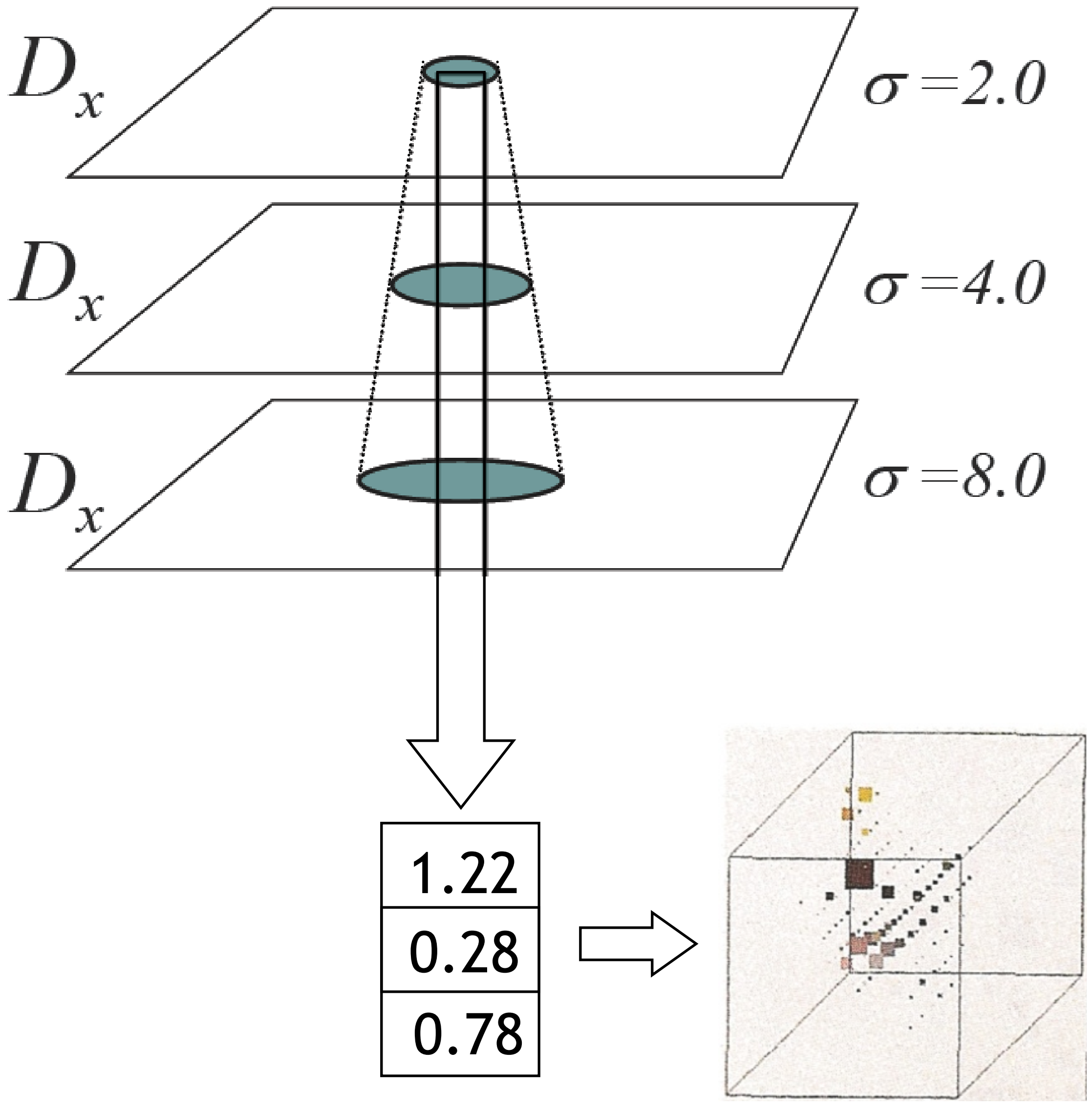
Multidimensional Histograms

- Useful combinations:
- D_x-D_y : Rotation-variant
  - Descriptor changes when image is rotated.
  - Useful for recognizing oriented structures (e.g. vertical lines)
- Mag-Lap : Rotation-invariant
  - Descriptor does not change when image is rotated.
  - Can be used to recognize rotated objects.
  - Less discriminative than rotation-variant descriptor.

Multidimensional Histograms

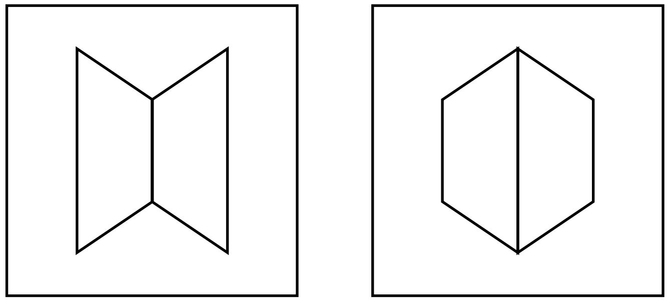
- Combination of several (smoothing) scales
  - Descriptors are computed at different scales.
  - Each scale captures different information about the object.
  - Size of the support region grows with increasing **σ**.

  - Feature vectors capture both local details and larger-scale structures.

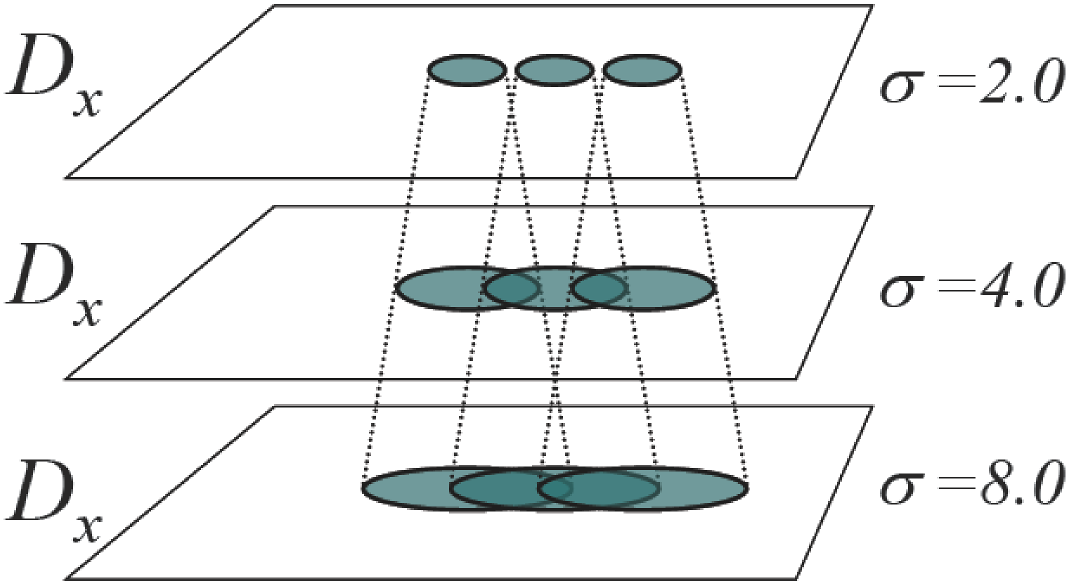


Why multiple scales?

- Histogram representation
  - ... contains no structural description.
  - Many different objects should result in the same histograms.



- But
  - Support regions of neighboring descriptors overlap.
  - Neighborhood relations are captured implicitly.



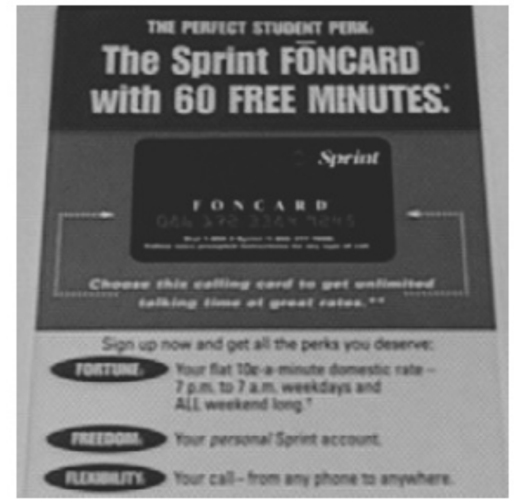
Multidimensional Histograms

- Experiences:
  - Work well for simple recognition problems.
  - Usually, simple combinations are sufficient (e.g. D_x-D_y, Mag-Lap)
  - Multiple scales are important!
- Problems:
  - High-dimensional histograms ⇒ lots of storage, need many examples
  - Global representation ⇒ not robust to occlusion

Recognition Results

Test image 1

First Match



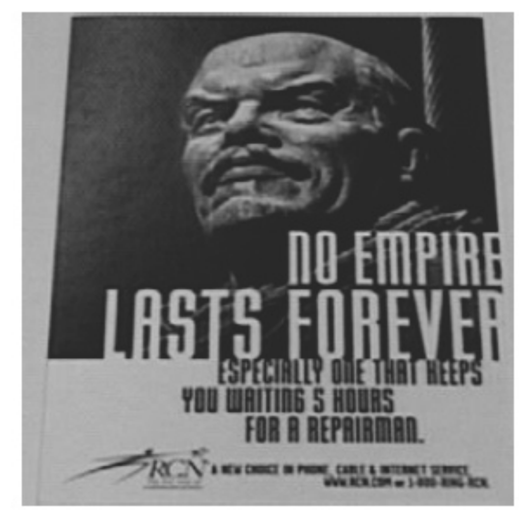
Second Match

Third Match

Test image 2

First Match

Second Match

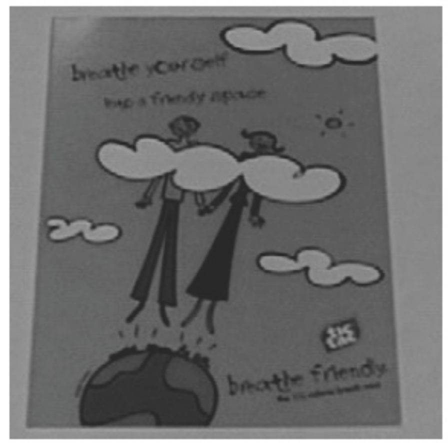


Third Match

[Schiele & Crowley, 2000]

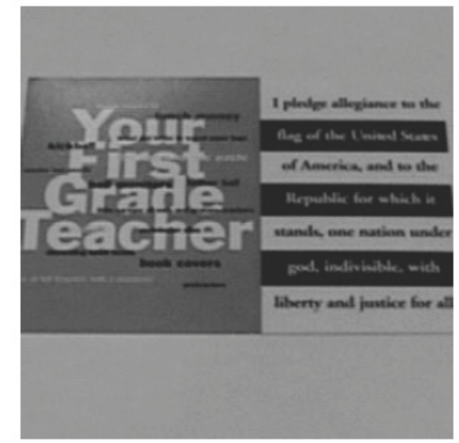
Recognition Results

Test image 3



First Match

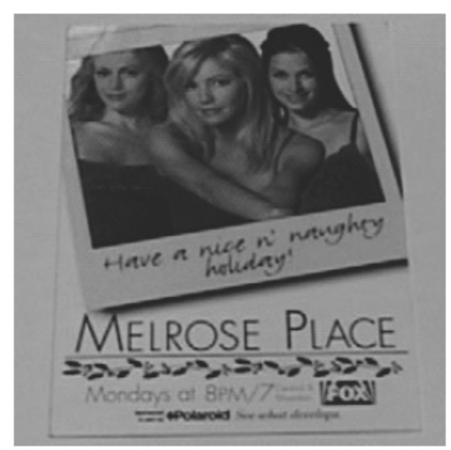
Second Match



Third Match

Test image 4

First Match



Second Match

Third Match

Fourth Match

[Schiele & Crowley, 2000]

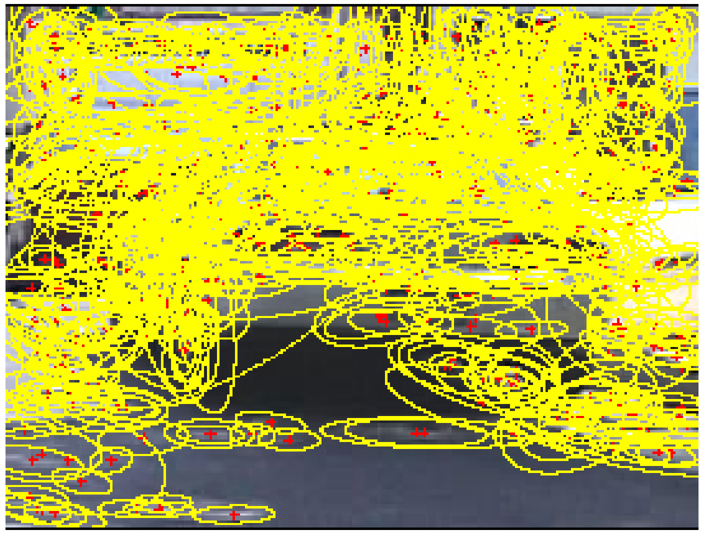
Summary

- Appearance-based object recognition
  - Using global representations
  - With full and without any spatial information
- Histograms:
  - Color histograms
  - Histogram comparison measures
  - Multidimensional histograms
- So far only object identification / instance recognition!
  - The histogram idea can be also used for object categorization... stay tuned.

Feature representation

- 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

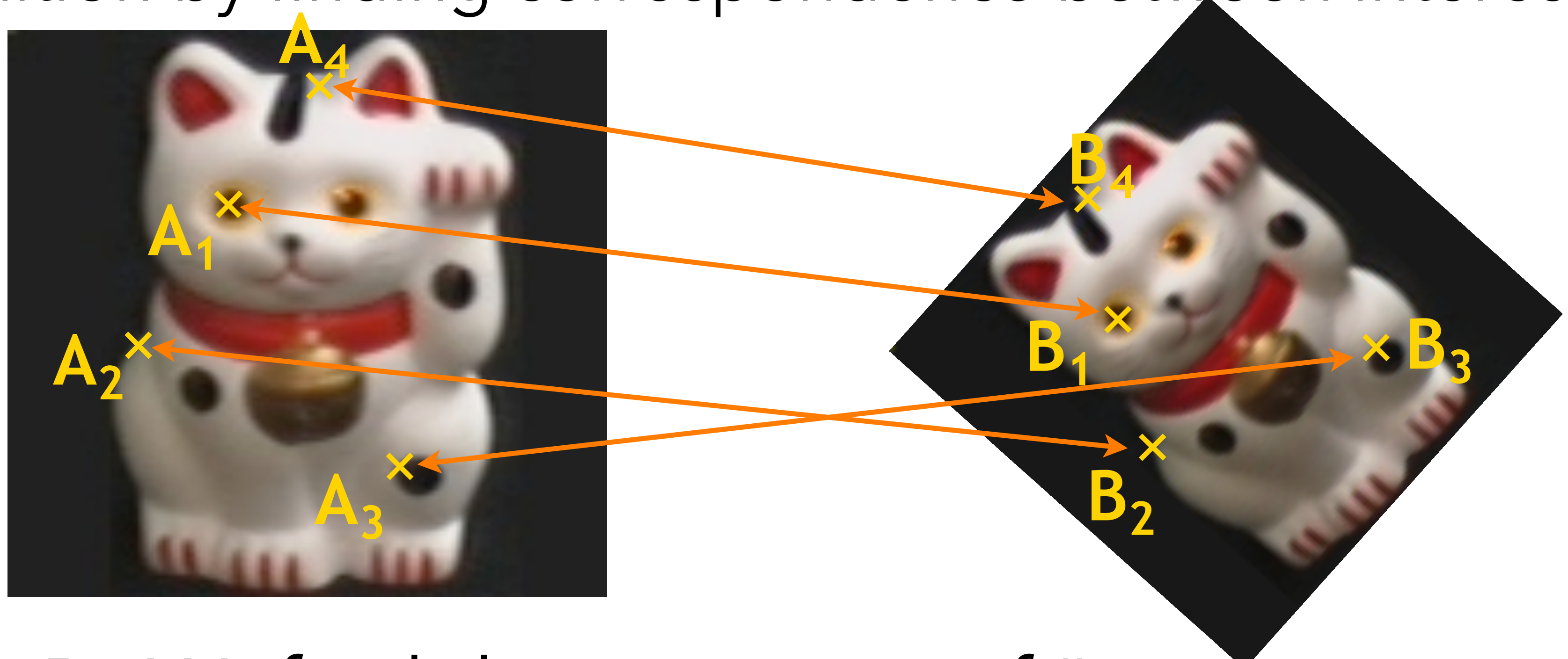


Overview

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

Motivation for Local Interest Point Detection

- 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

Image matching

- Different application
- Same goal!

by Diva Sian

by swashford

[Szeliski & Seitz]

Image matching - Harder case

TECHNISCHE UNIVERSITÄT DARMSTADT

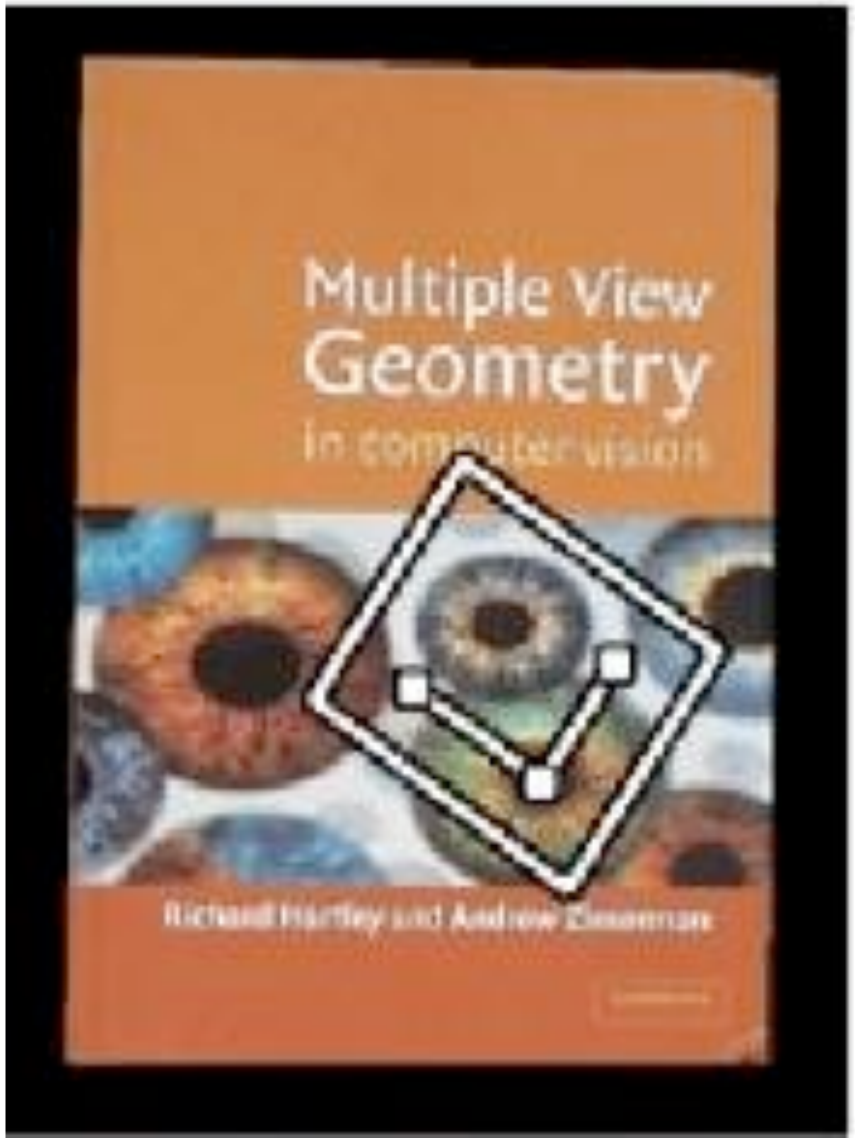
- Different application
- Same goal!

by Diva Sian

by scgbt

[Szeliski & Seitz]

Image matching



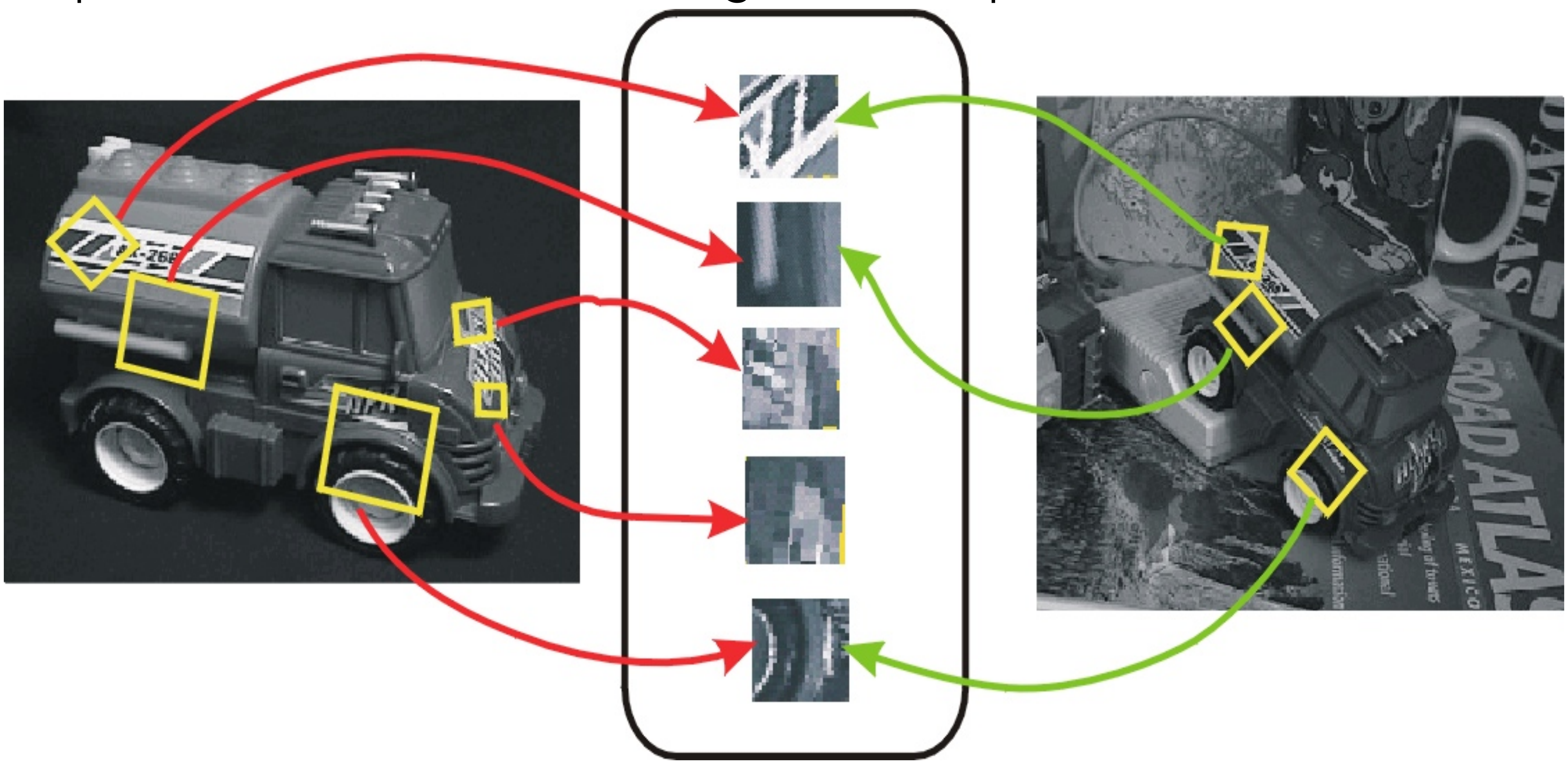
[Szeliski & Seitz]

Image matching

[Szeliski & Seitz]

Invariant local features

- Find features that are invariant to transformations
  - geometric invariance: translation, rotation, scale
  - photometric invariance: brightness, exposure, …



[Szeliski & Seitz]

Advantages of local features

- Locality:
  - features are local, so robust to occlusion and clutter
- Distinctiveness:
  - can differentiate a large database of objects
- Quantity:
  - hundreds or thousands in a single image
- Efficiency:
  - real-time performance achievable
- Generality:
  - exploit different types of features in different situations

[Szeliski & Seitz]

More motivation...

- Feature points are used for:
  - Object recognition
  - Image alignment (e.g., mosaics)
  - 3D reconstruction
  - Motion tracking
  - Indexing and database retrieval
  - Robot navigation
  - … other

[Szeliski & Seitz]

What makes a good feature?

[Szeliski & Seitz]

Want uniqueness

TECHNISCHE UNIVERSITÄT DARMSTADT

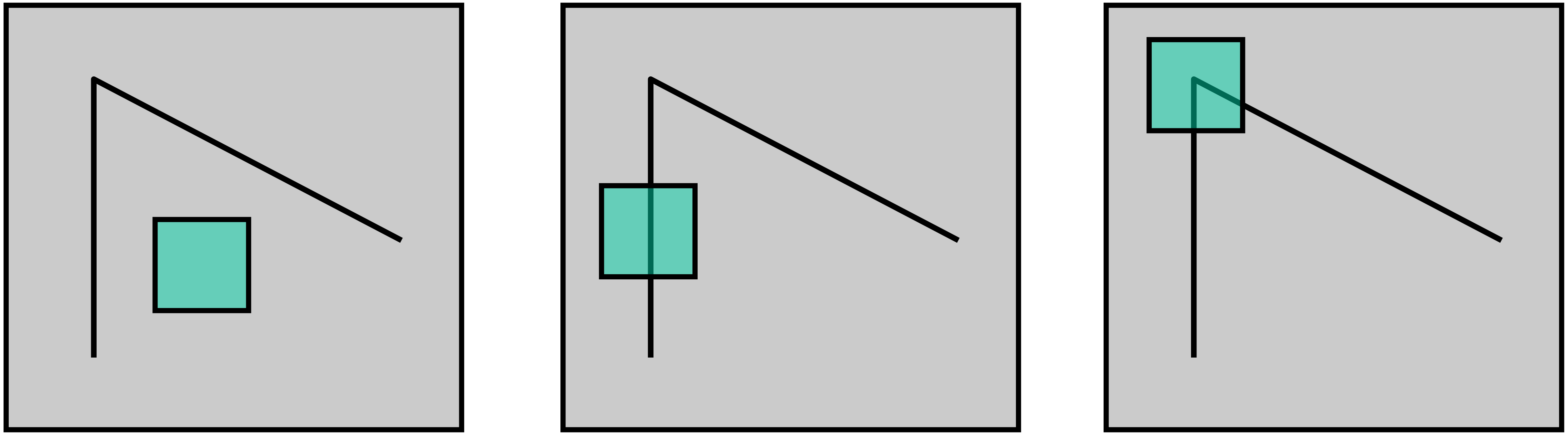
- Look for image regions that are unusual.
  - Lead to unambiguous matches in other images

- How to define “unusual”?

[Szeliski & Seitz]

Local measures of uniqueness

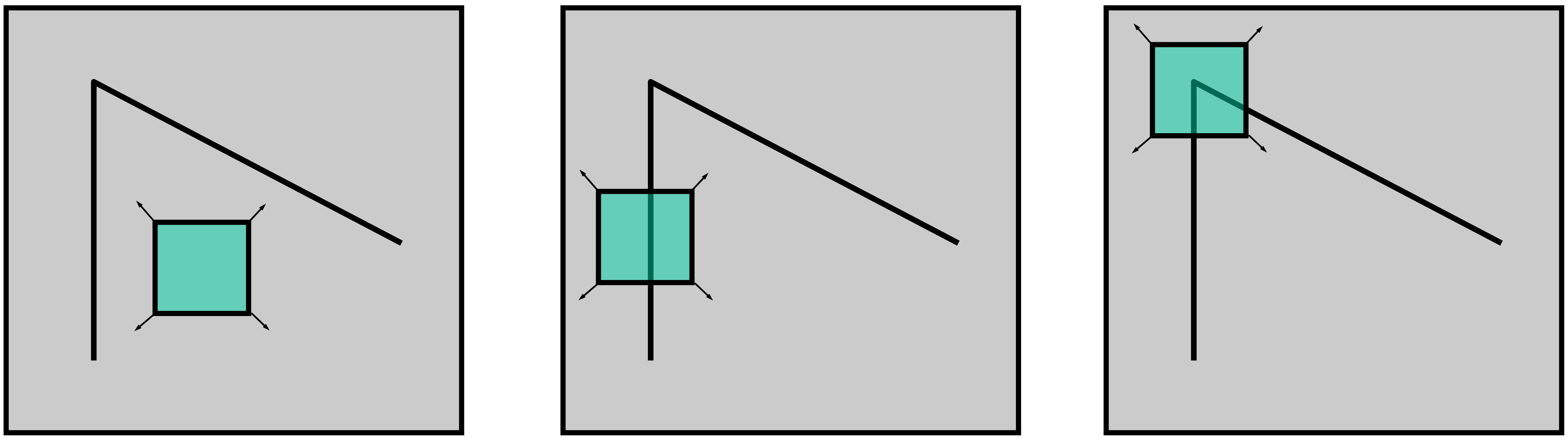
- Suppose we only consider a small window of pixels
  - What defines whether a feature is a good or bad candidate?



[Szeliski & Seitz]

Feature detection

- 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

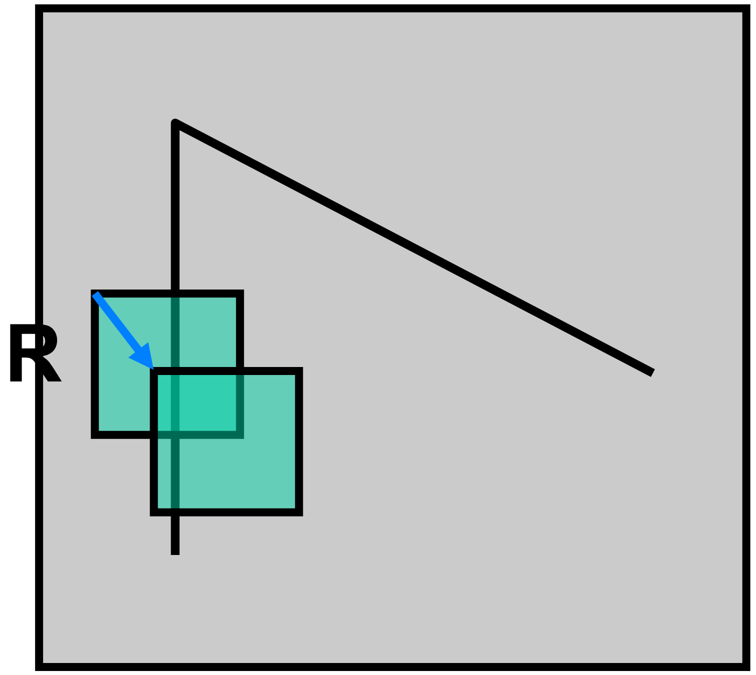
"edge": no change along the edge direction

"corner": significant change in all directions

[Szeliski & Seitz]

Feature detection: Derivation

- Consider shifting the window R by (u,v)
  - how do the pixels in R change?
  - compare each pixel before and after by summing up the squared differences (SSD)
  - this defines an SSD "error" of E(u,v):



$$E(u,v) = \sum_{(x,y)\in R} \big(I(x+u, y+v) - I(x,y)\big)^2$$

[Szeliski & Seitz]

Can we approximate this?

$$E(u,v) = \sum_{(x,y)\in R} \left(I(x+u, y+v) - I(x,y)\right)^2$$

Taylor series approximation:

$$I(x,y) + u\frac{\partial}{\partial x}I(x,y) + v\frac{\partial}{\partial y}I(x,y) + \epsilon(u^2, v^2)$$

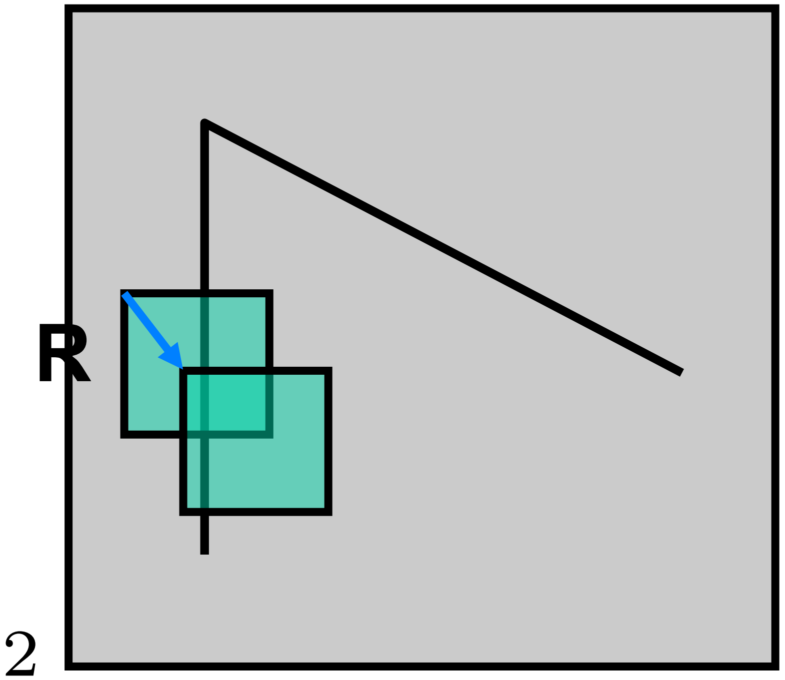
Approximation error

$$E(u,v) \approx \sum_{(x,y)\in R} \left(u\frac{\partial}{\partial x}I(x,y) + v\frac{\partial}{\partial y}I(x,y)\right)^2$$

Feature detection: Derivation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Consider shifting the window R by (u,v)
  - how do the pixels in R change?
  - compare each pixel before and after by summing up the squared differences (SSD)
  - this defines an SSD "error" of E(u,v):



$$E(u,v) = \sum_{(x,y)\in R} \big(I(x+u,y+v) - I(x,y)\big)^2$$

$$\approx \sum_{(x,y)\in R} \big(u \cdot I_x(x,y) + v \cdot I_y(x,y)\big)^2$$

$$= \sum_{(x,y)\in R} (u,v)\nabla I(x,y)\nabla I(x,y)^T \begin{pmatrix} u \\ v \end{pmatrix}$$

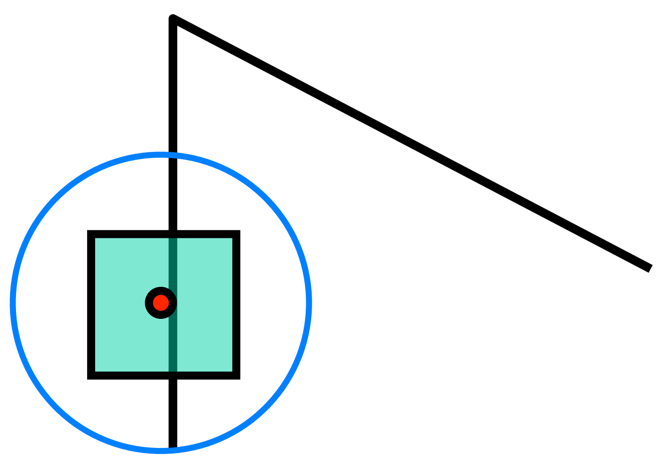
$$\nabla I = \begin{pmatrix} I_x \\ I_y \end{pmatrix}$$

[Szeliski & Seitz]

Feature detection: Derivation

$$E(u,v) \approx (u,v) \underbrace{\left[\sum_{(x,y)\in R} \nabla I(x,y) \nabla I(x,y)^T \right]}_{H} \begin{pmatrix} u \\ v \end{pmatrix}$$

structure tensor



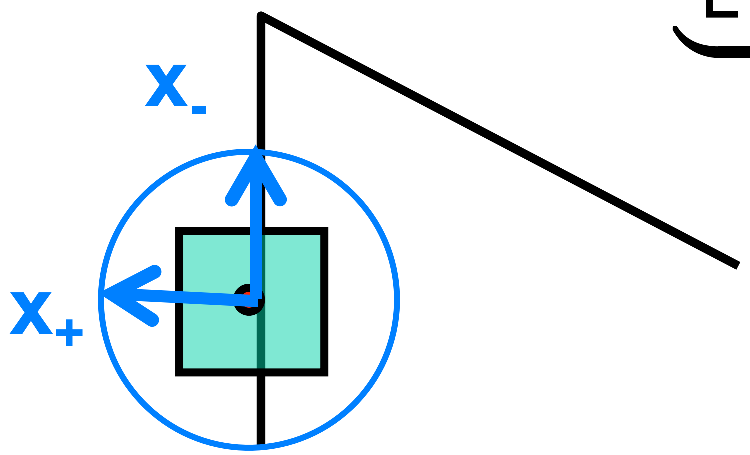
- For the example above
  - Suppose you can move the center of the green window to anywhere on the blue unit circle
  - Which directions will result in the largest and smallest E values?
  - We can find these directions by looking at the eigenvectors of H

[Szeliski & Seitz]

Feature detection: Derivation

$$E(u,v) \approx (u,v) \underbrace{\left[\sum_{(x,y)\in R} \nabla I(x,y) \nabla I(x,y)^T \right]}_{H} \begin{pmatrix} u \\ v \end{pmatrix}$$

structure tensor



- Eigenvalues and eigenvectors of H
  - Define shifts with the smallest and largest change (E value)
  - x_+ = direction of largest increase in E.
  - λ_+ = amount of increase in direction x_+
  - x_- = direction of smallest increase in E.
  - λ_- = amount of increase in direction x_-

$$\begin{aligned} Hx_+ &= \lambda_+ x_+ \\ Hx_- &= \lambda_- x_- \end{aligned}$$

[Szeliski & Seitz]

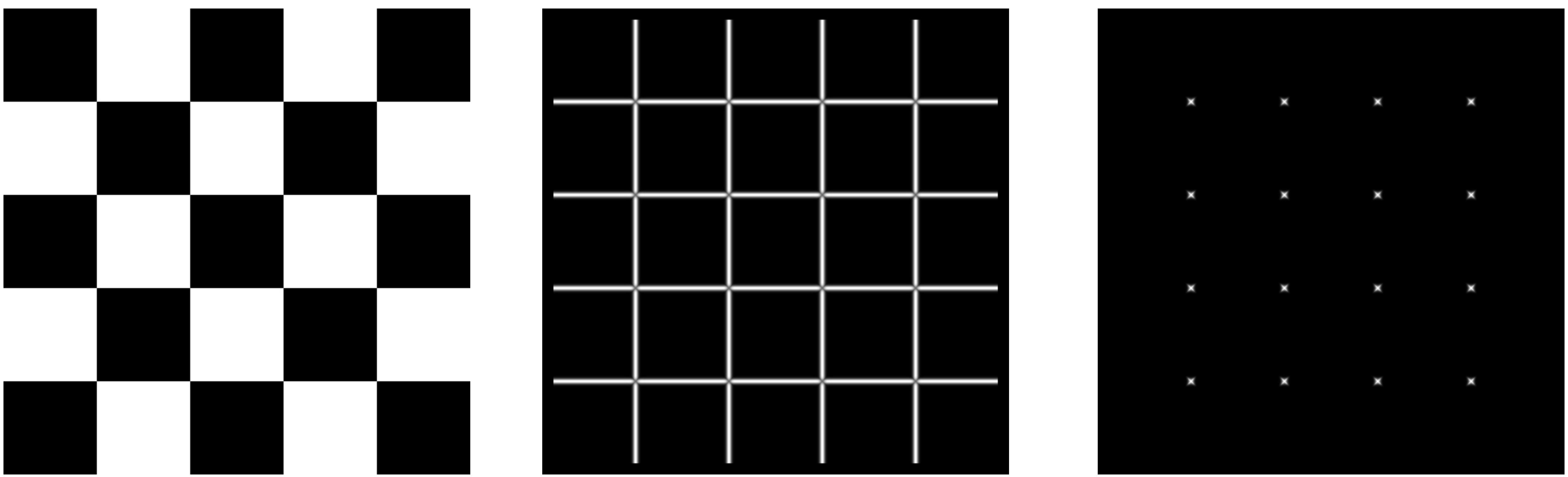
Feature detection

- How are λ_+, x_+, λ_-, and x_- relevant for feature detection
  - What's our feature scoring function?

[Szeliski & Seitz]

Feature detection

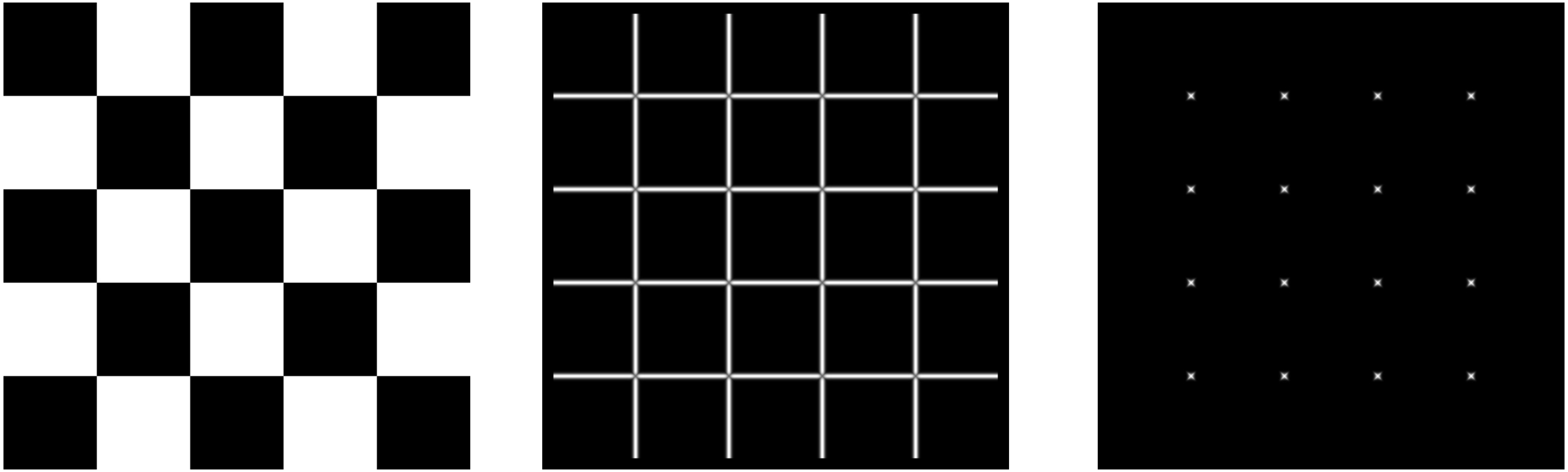
- How are λ_+, x_+, λ_- and x_- relevant for feature detection
  - What's our feature scoring function?
- Want $E(u,v)$ to be ***large*** for small shifts in ***all*** directions
  - the *minimum* of $E(u,v)$ should be large, over all unit vectors [u v]
  - this minimum is given by the smaller eigenvalue (λ_-) of H



I λ_+ λ_-

Feature detection

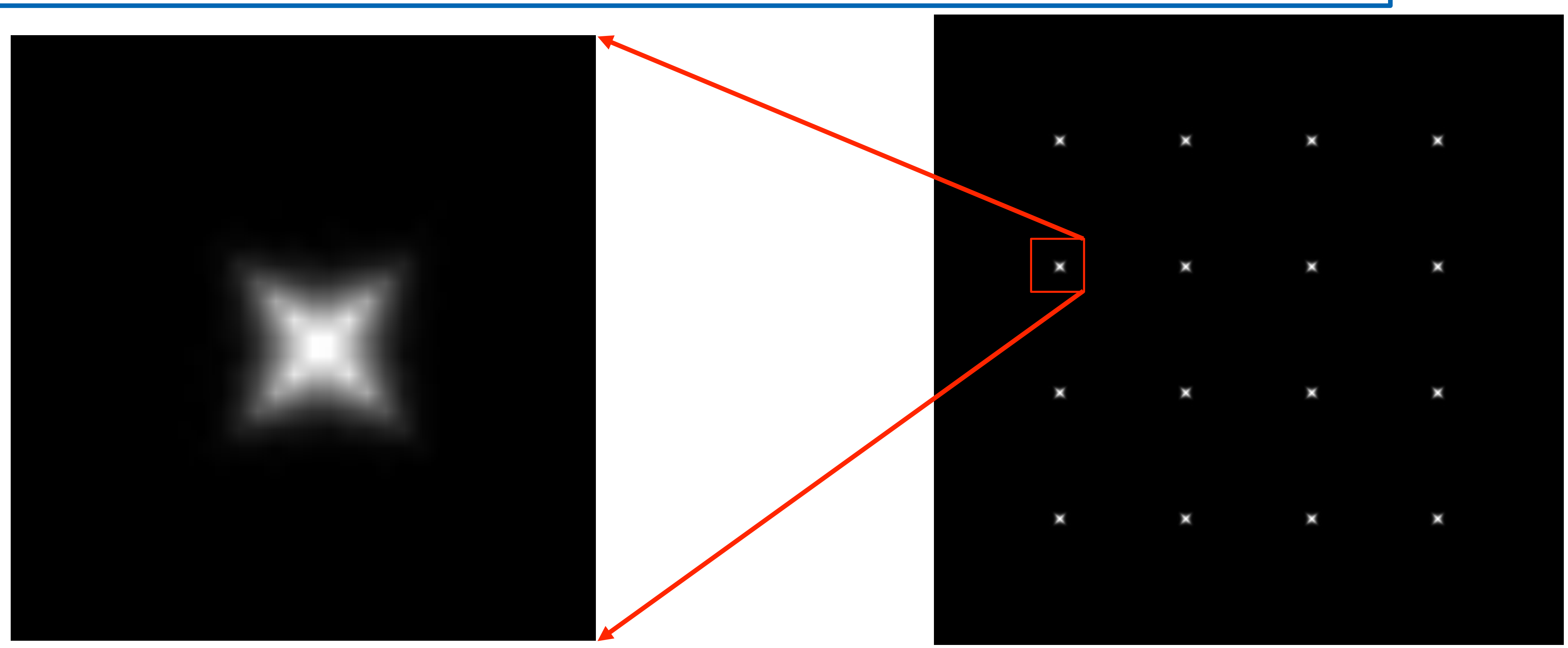
- Here's what you do
  - Compute the gradient at each point in the image
  - Create the H matrix from the entries in the gradient
  - Compute the eigenvalues.
  - Find points with large response ($\lambda_{-} >$ threshold)
  - Choose those points where λ_{-} is a local maximum as features



I λ_{+} λ_{-}

Feature detection summary

- 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

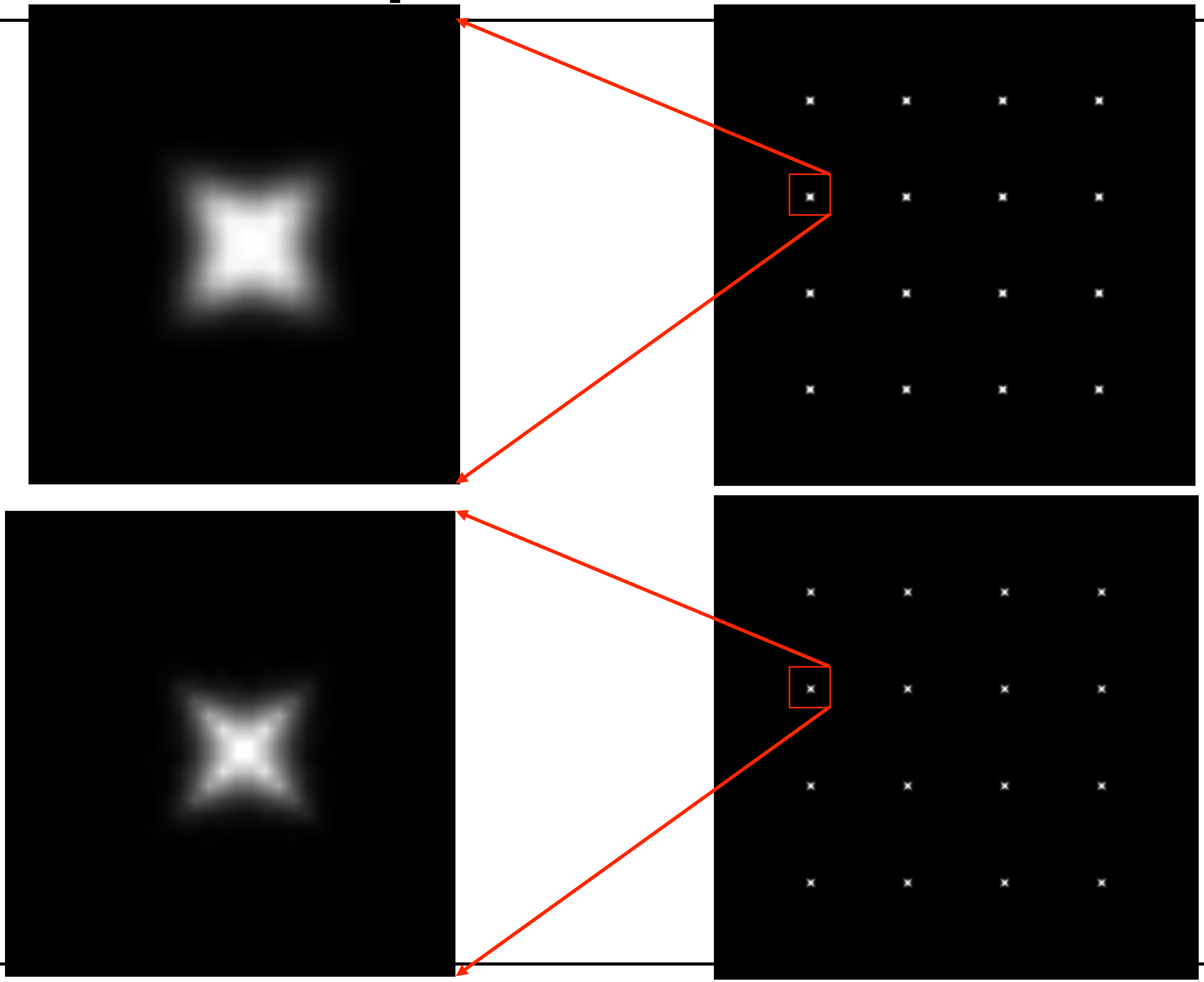
- λ_- is a variant of the "Harris operator" for feature detection

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

  - 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]

The Harris operator



Harris operator

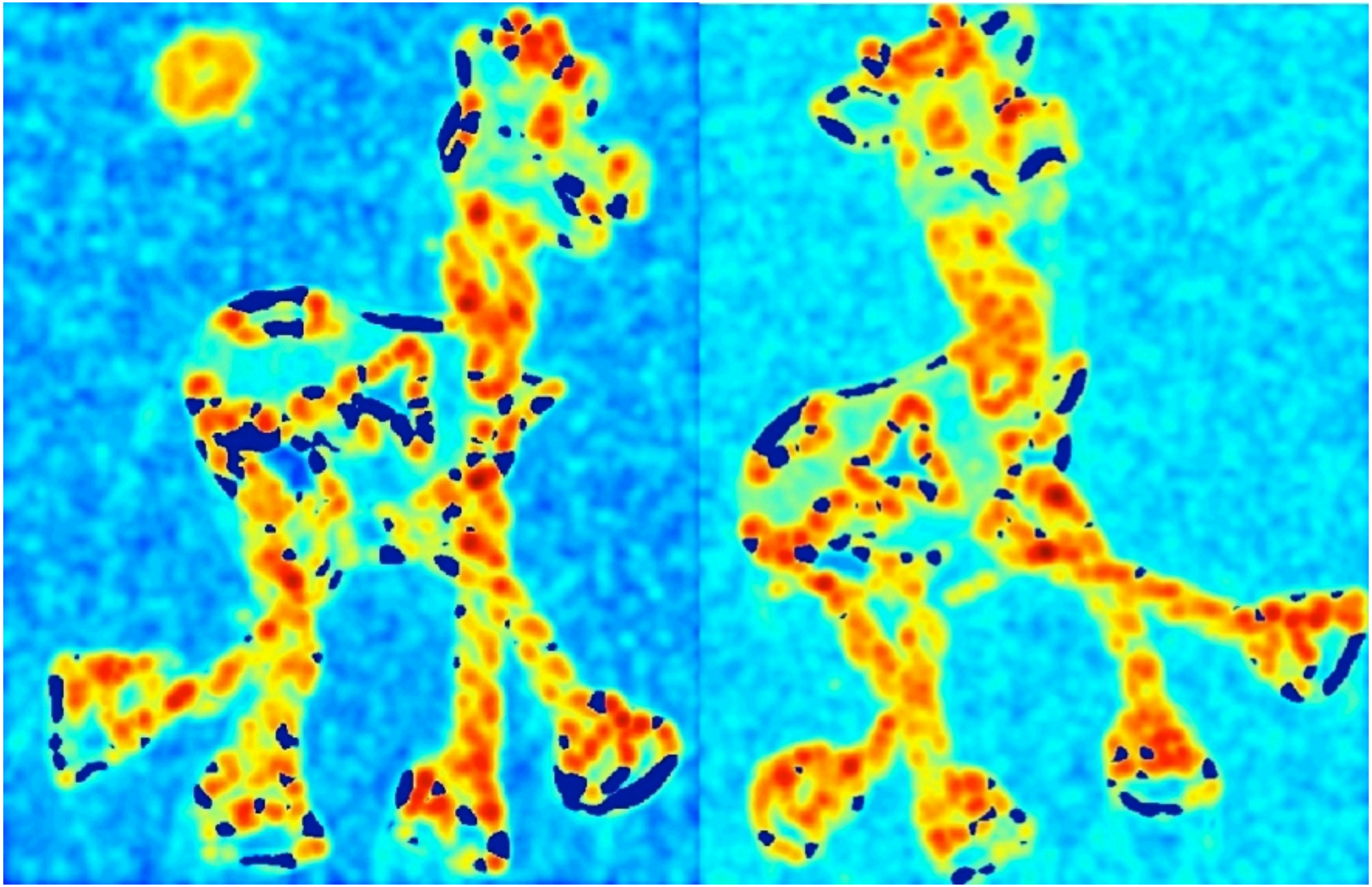
λ_-

[Szeliski & Seitz]

Harris detector example

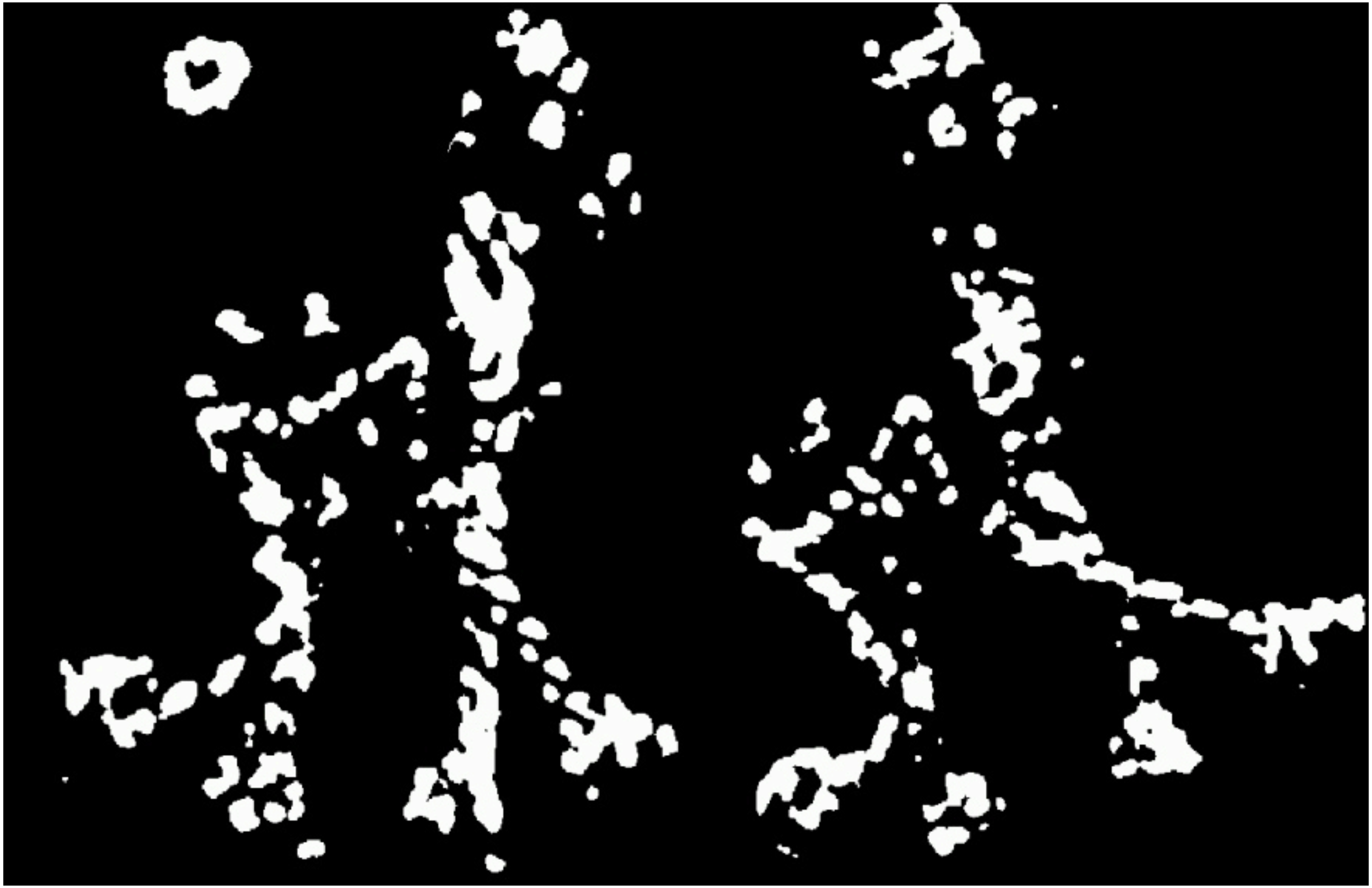
[Szeliski & Seitz]

f value (red high, blue low)



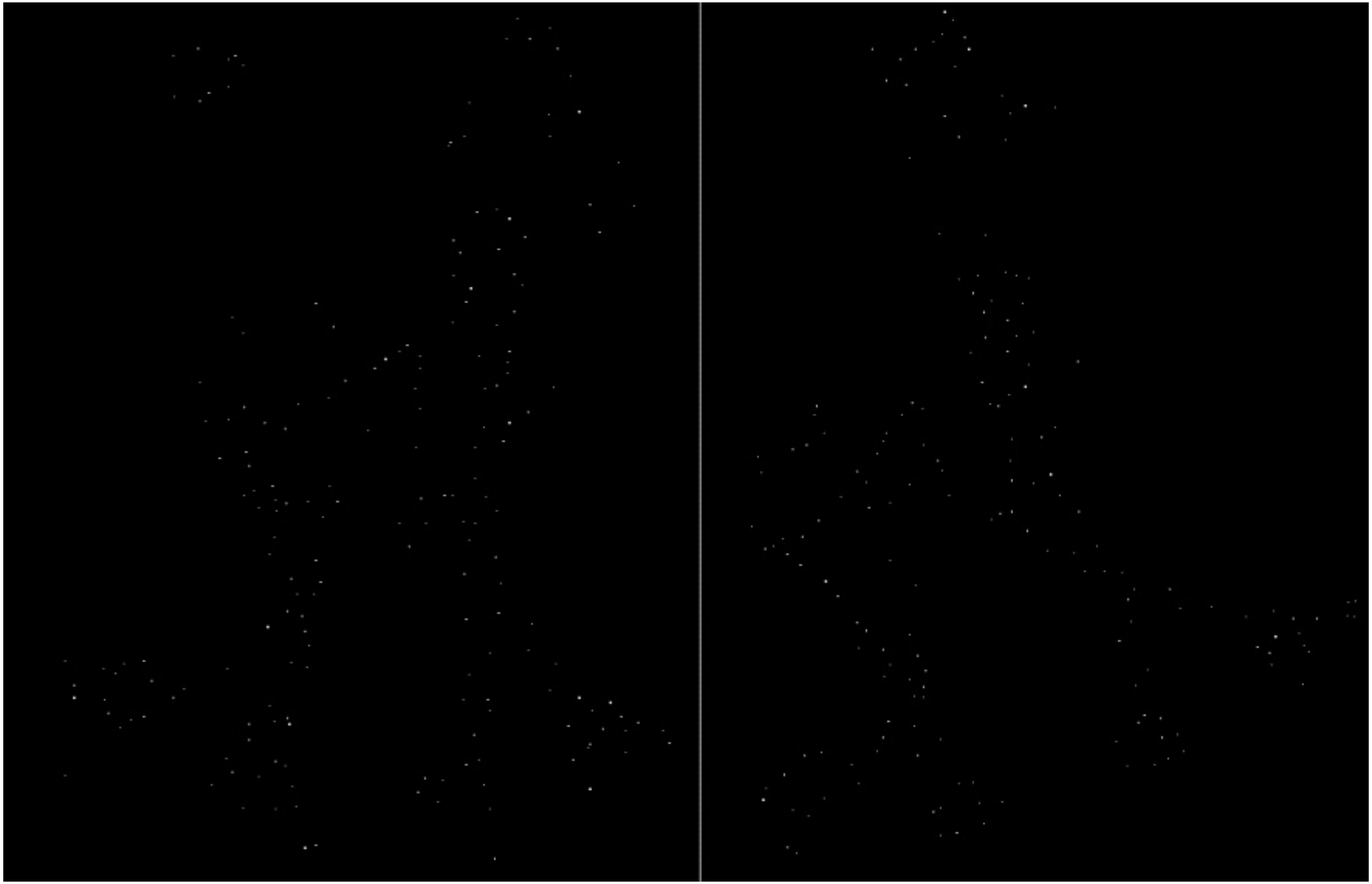
[Szeliski & Seitz]

Threshold (f > value)



[Szeliski & Seitz]

Find local maxima of f

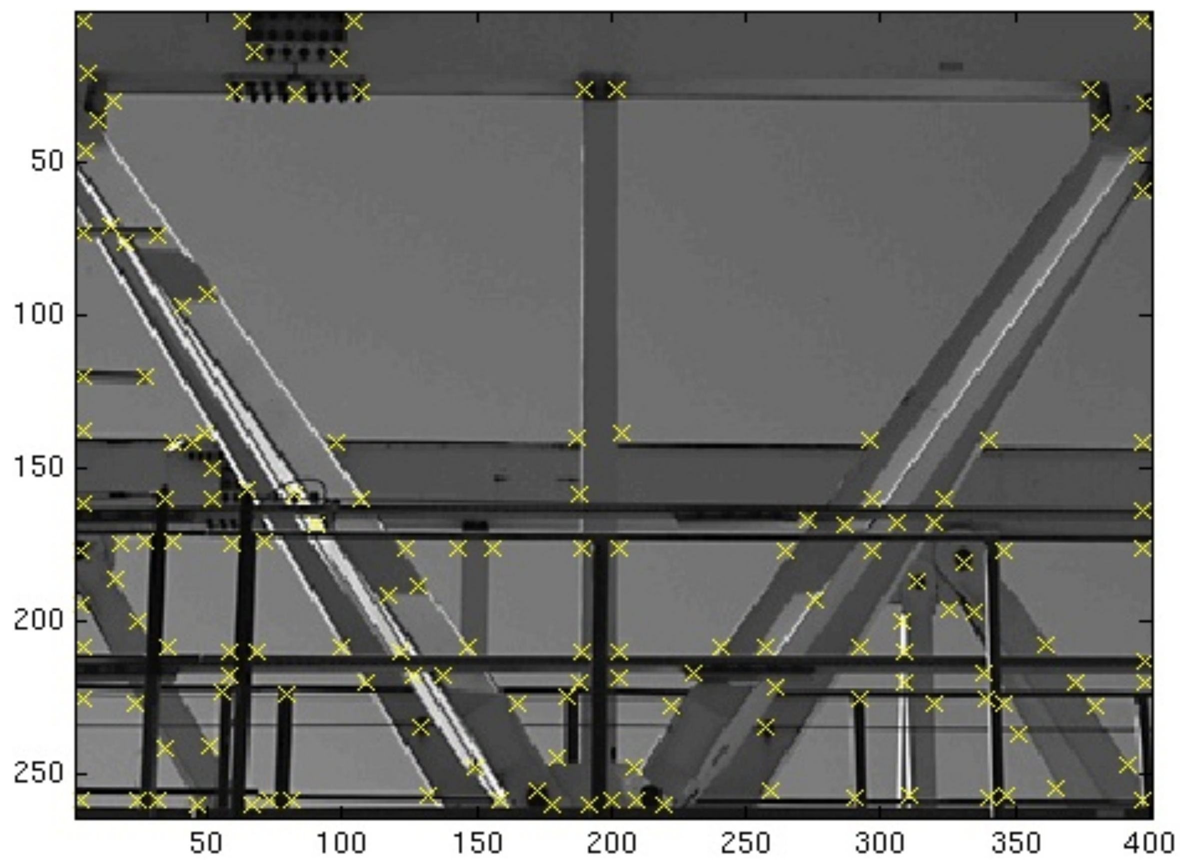


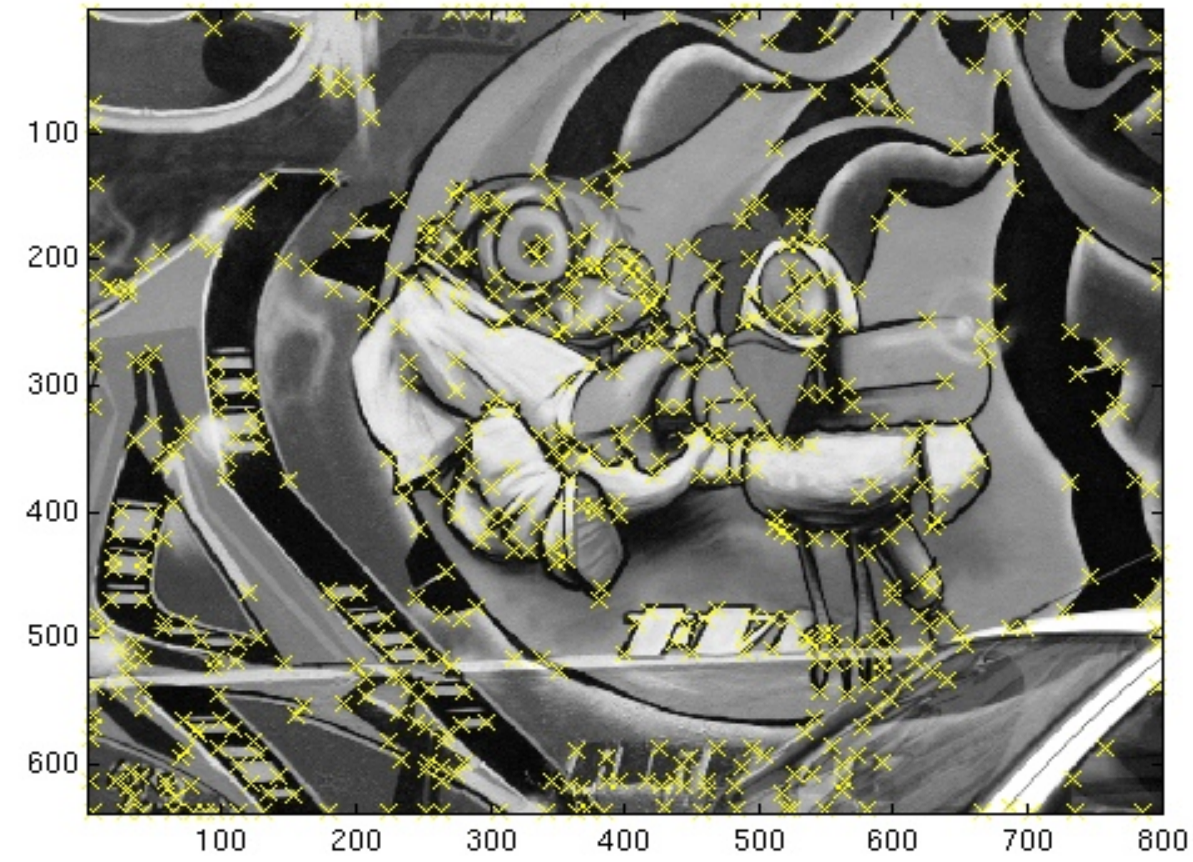
[Szeliski & Seitz]

Harris features (in red)

[Szeliski & Seitz]

Harris interest points



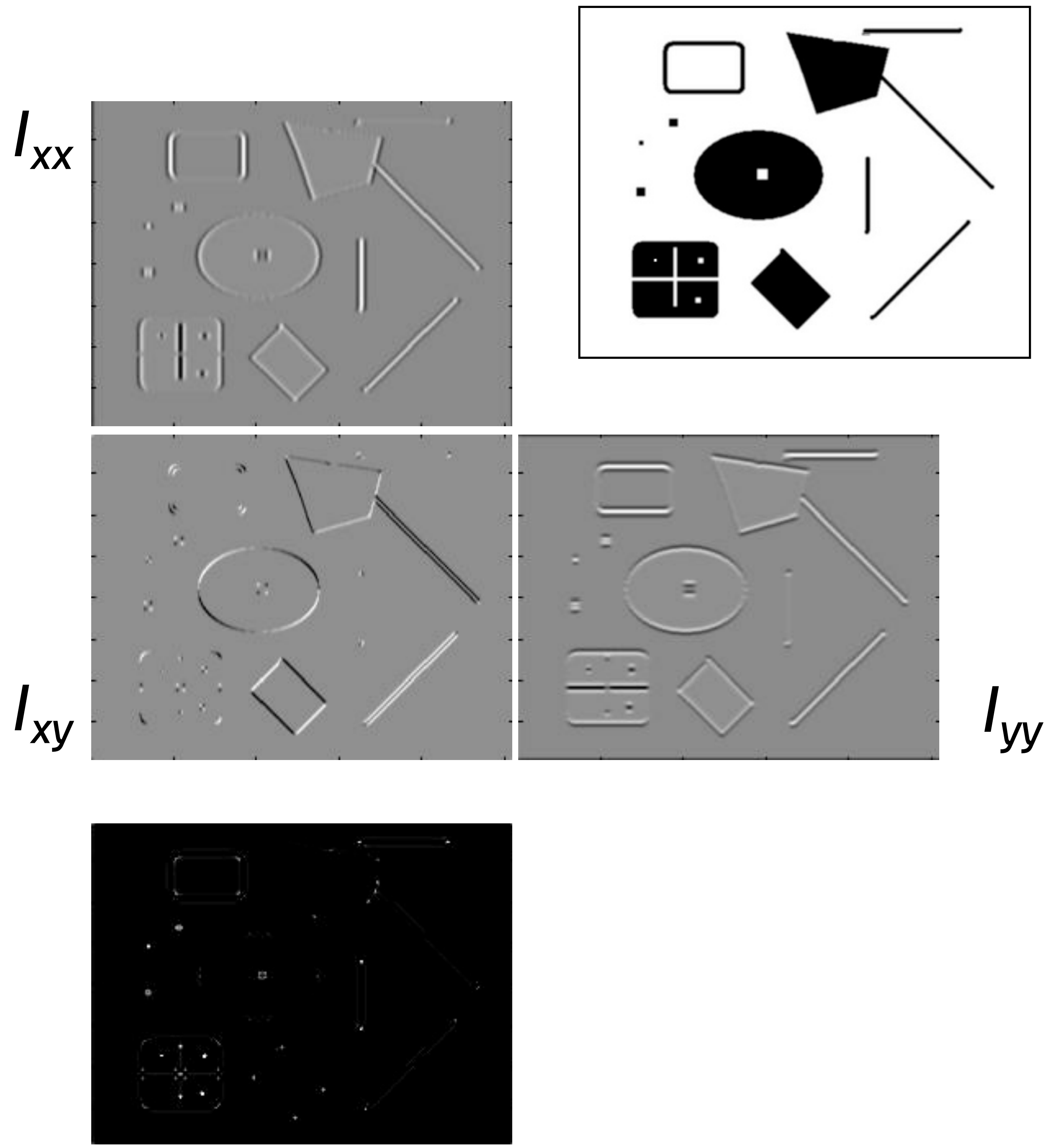


Other interest point detectors

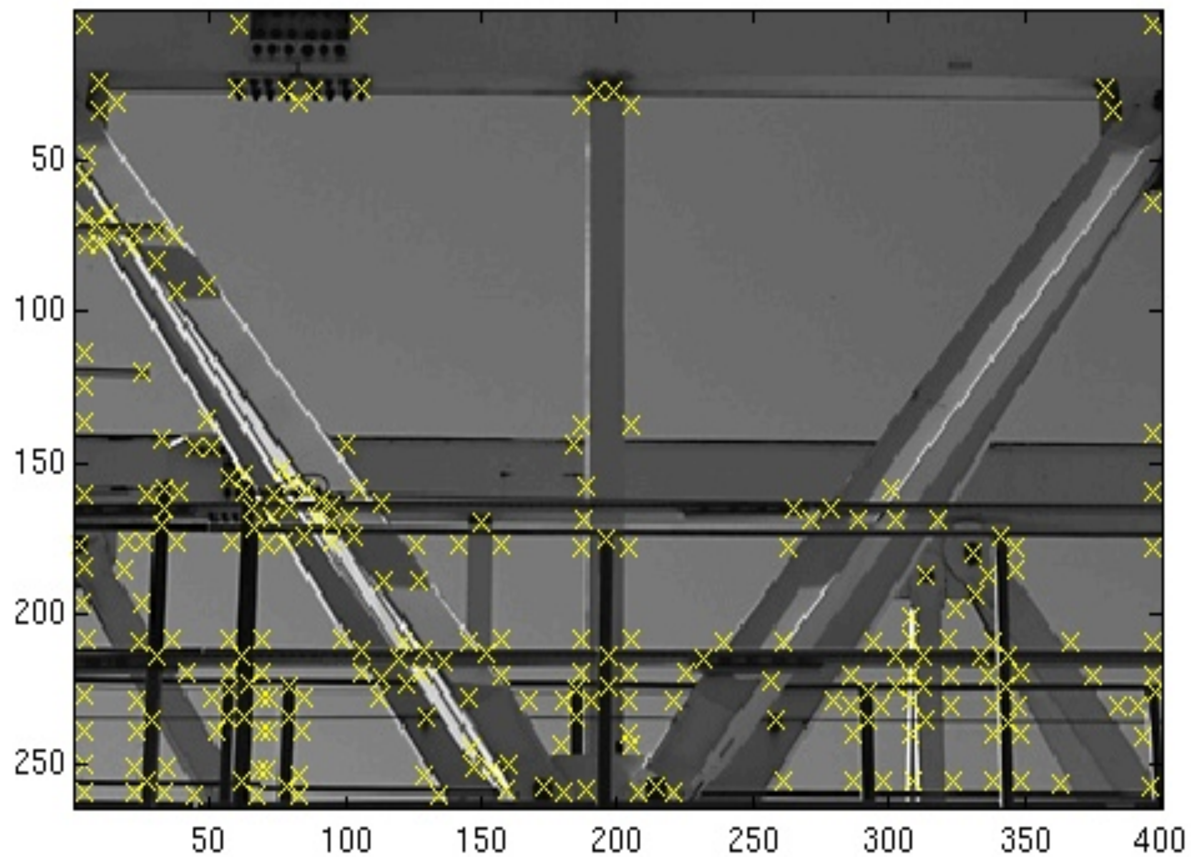
- Hessian determinant

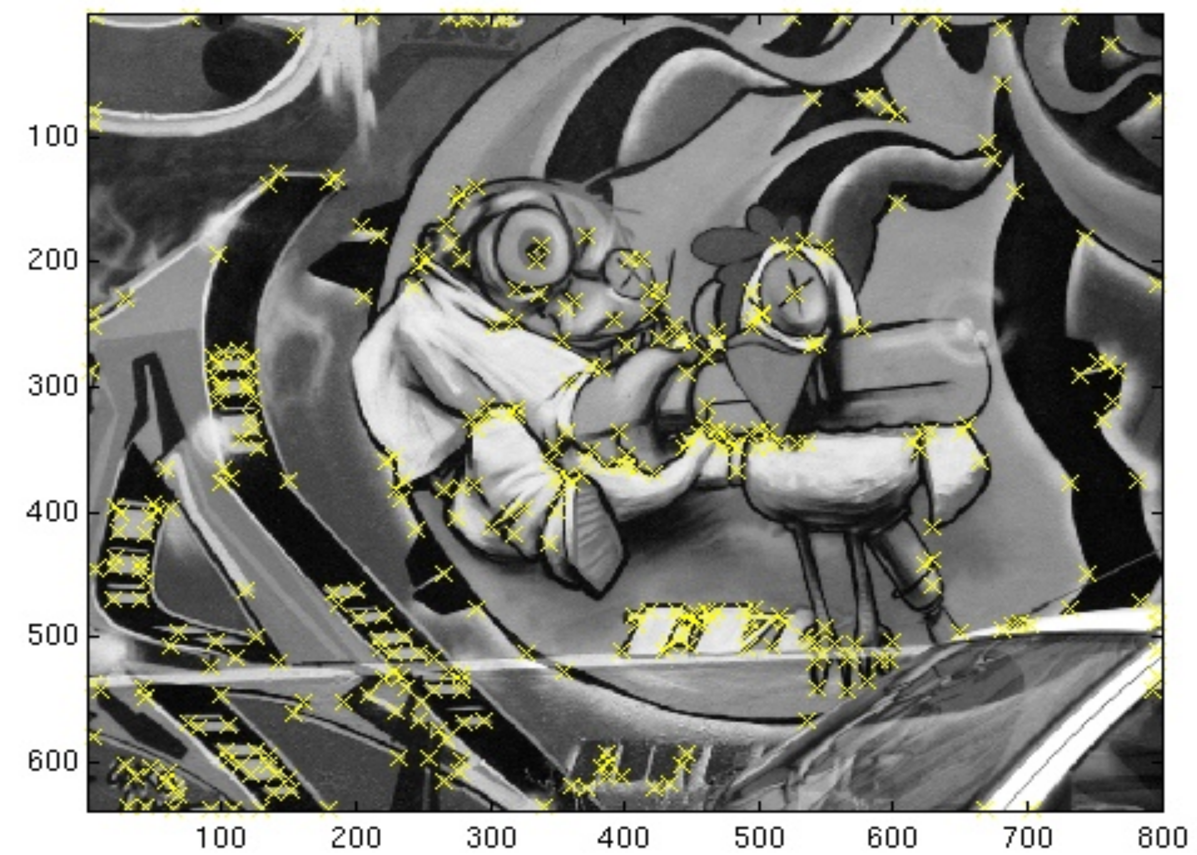
$$H(I) = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}$$

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



Hessian interest points





Discussion

- 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