Visuelle Perzeption für Mensch-Maschine Schnittstellen

Vorlesung, WS 2013/14

Prof. Dr. Rainer Stiefelhagen
Arne Schumann

Institut für Anthropomatik
Universität Karlsruhe (TH)

http://cvhci.ira.uka.de
rainer.stiefelhagen@kit.edu
arne.schumann@kit.edu

Computer Vision:

People Detection II

WS 2013/14

Arne Schumann

arne.schumann@kit.edu

Review

- Person Detection I
  - video vs still image
  - discriminative vs. generative
  - part-based vs. holistic (or global)
  - HOG
    - global, discriminative model
    - 4000 dim. feature vector
    - SVM classifier
    - sliding window
    - image pyramid
  - Chamfer Matching
    - silhouettes
    - 2-3 distance

Today

- Part-based approaches
  - detecting / finding parts
  - learning part representations
  - learning spatial layout / part configurations
  - combining parts for person detection
  - optional: verification

Part-Based Models

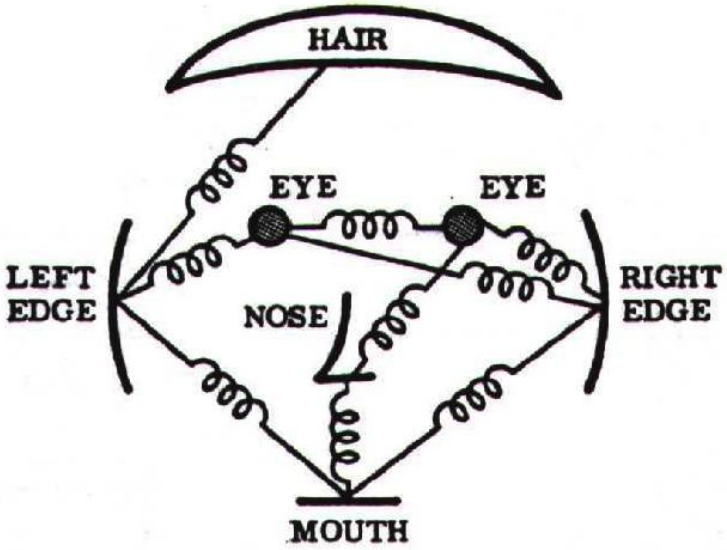
Motivation

- Break down an objects' overall variability into more manageable pieces
- Pieces can be classified by less complex classifiers
- Apply prior knowledge by (manually) splitting the global object into meaningful parts

Part-Based Models

- First proposed in:
  - Fischler & Elschlager, 1973: The representation and matching of pictorial structures

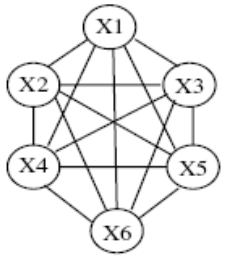
- Model has two components
  - parts (2D image fragments)
  - structure (configuration of parts)



Configuration of parts

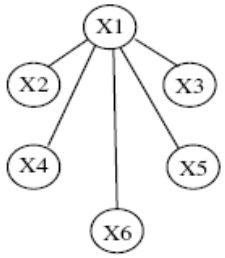
- Fixed Spatial Layout
  - local parts are modeled to have a mostly fixed position and orientation with respect to the object or detection window center
- Flexible Spatial Layout
  - local parts are allowed to shift in location and scale
  - can better handle deformations or articulation changes
  - well suited for non-rigid objects
  - spatial relations are often modeled probabilistically

Different Connectivity Structures



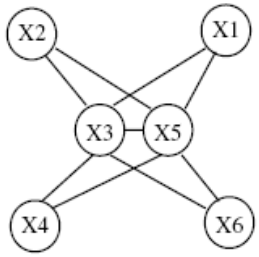
a) Constellation

Fergus et al. '03
Fei-Fei et al. '03



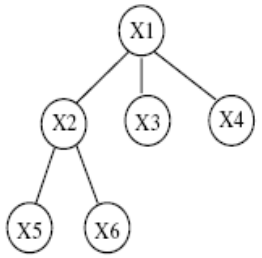
b) Star shape

Leibe et al. '04, '08
Crandall et al. '05
Fergus et al. '05



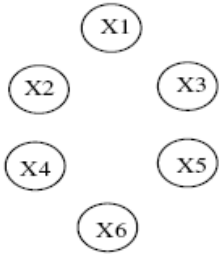
c) k-fan ($k = 2$)

Crandall et al. '05



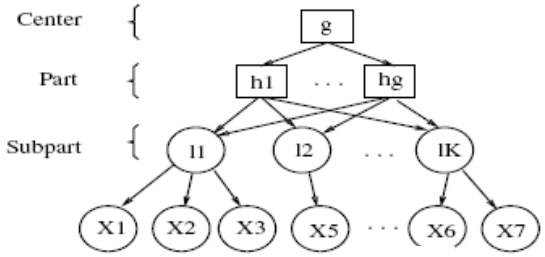
d) Tree

Felzenszwalb & Huttenlocher '05



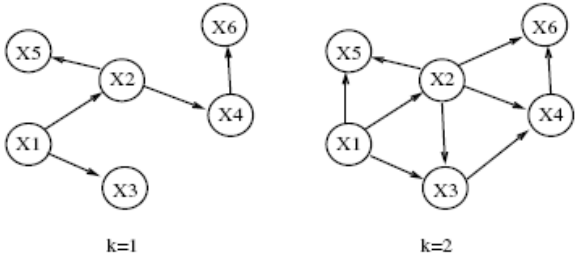
e) Bag of features

Csurka '04
Vasconcelos '00



f) Hierarchy

Bouchard & Triggs '05

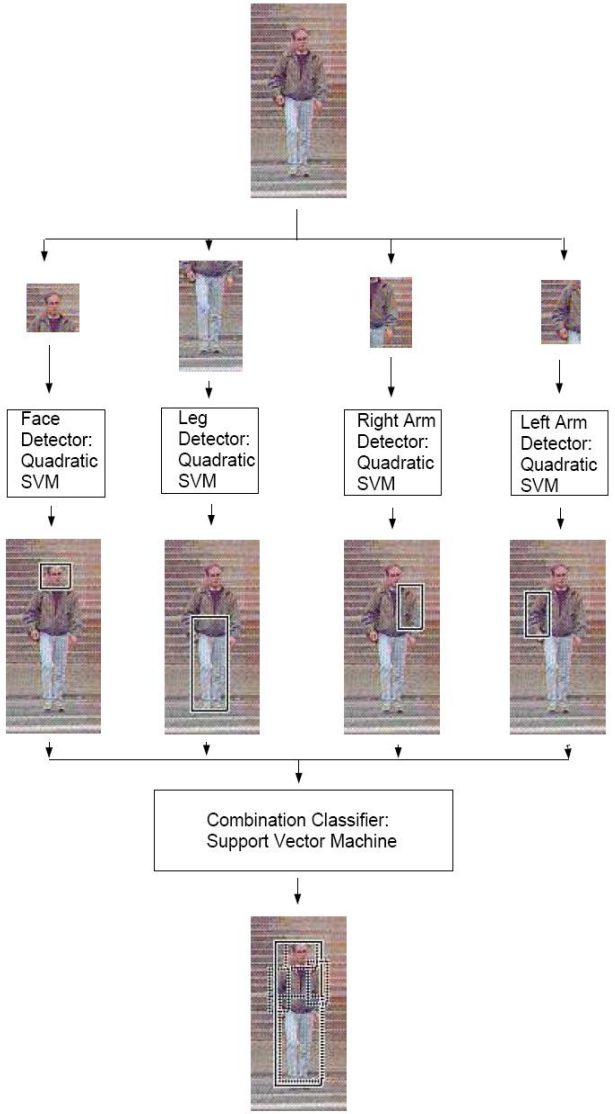


g) Sparse flexible model

Carneiro & Lowe '06

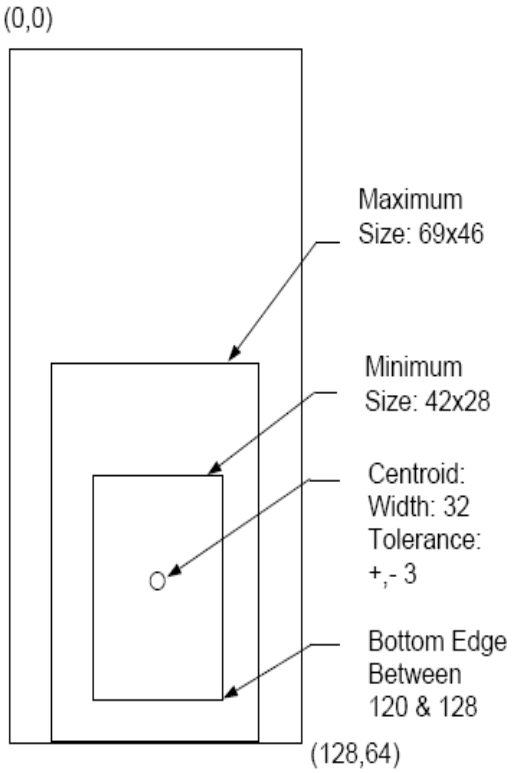
Example Fixed Spatial Layout – The Mohan Detector

- Body divided into 4 parts
  - face and shoulder
  - legs
  - right arm
  - left arm
- Detection
  - sliding window approach
  - 64x128 pixels
- Mohan 1999, Object Detection in Images by Components, MIT Technical Report



Geometric constraints

- Body parts are not always at the exact same position
- Allow local shifts
  - in position
  - and in scale
- Best location has to be found for each detection window

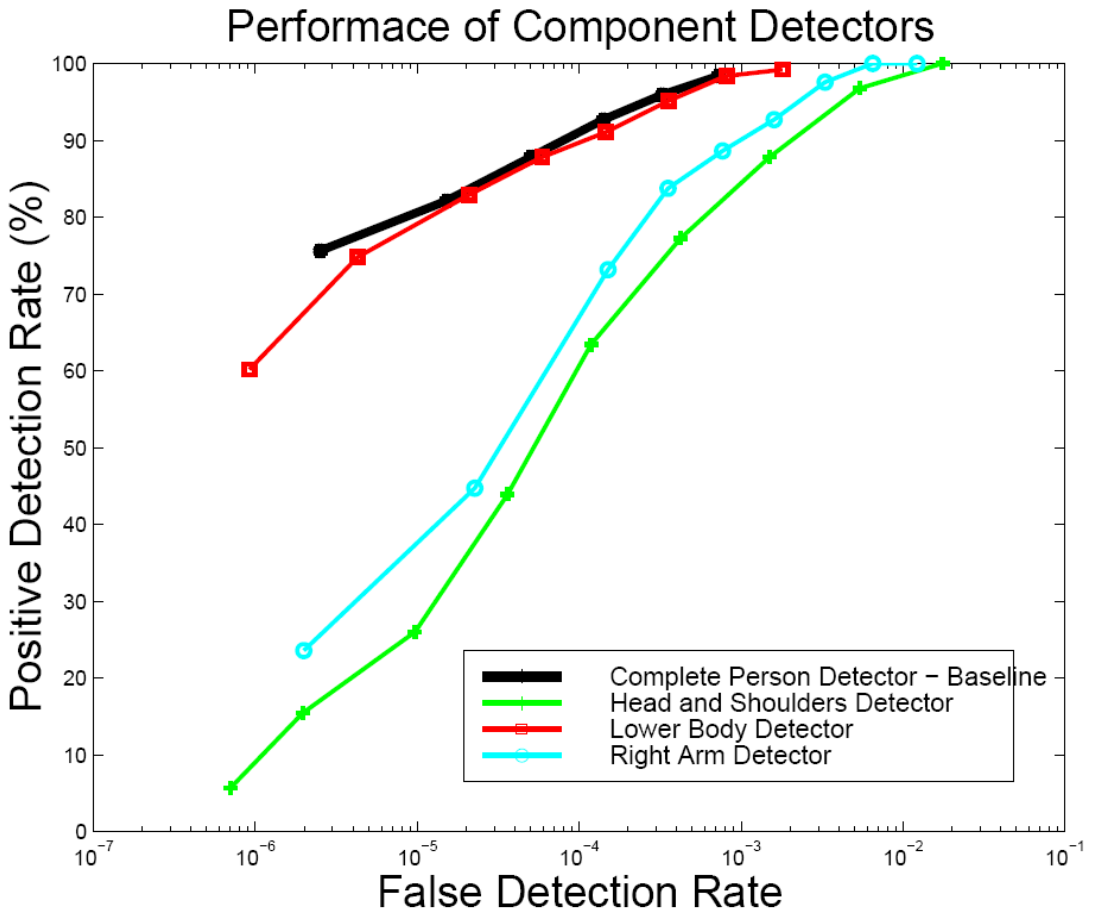


Allowed tolerances for lower body detections

Training Data

- MIT pedestrian database

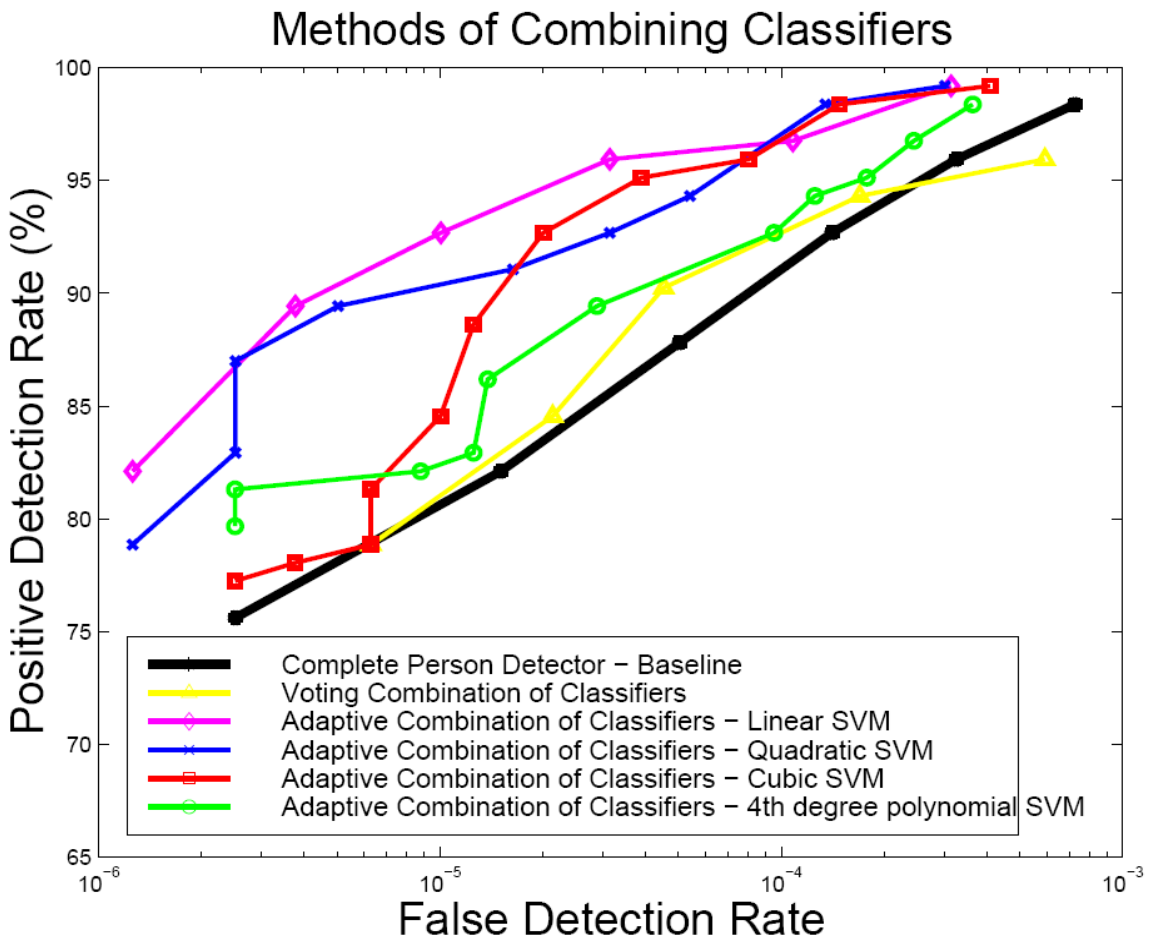
Performance of the Part Detectors



Part Classifier Combination

- Voting
  - e.g. majority of part detectors, classify detection window as person
- SVM combination
  - feed SVM scores of part detectors in a second stage SVM
  - referred to as Adaptive Classifier Combination

Combined Performance



Results of the Mohan Detector

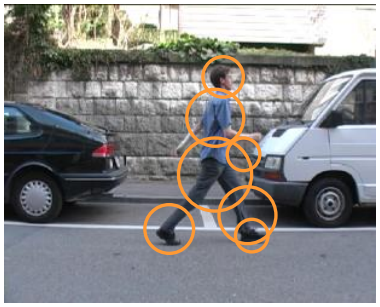
Robustness to Occlusions

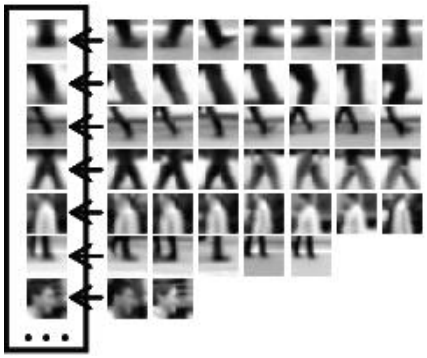
The Implicit Shape Model (ISM)

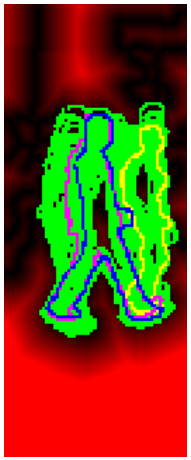
Flexible Spatial Layout

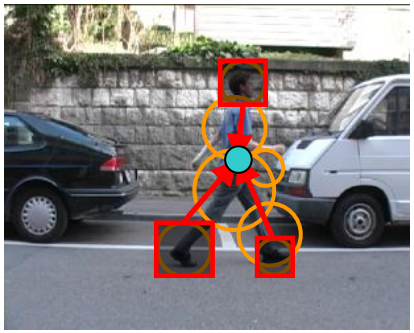
Implicit Shape Model (Leibe et al.)

1. Part Detection/Localization
2. Part Description
3. Learning Part Appearances
4. Learning the Spatial Layout of Parts
5. Combination of Part Detections
6. Verification
7. Extensions

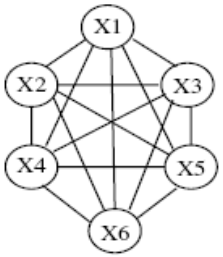






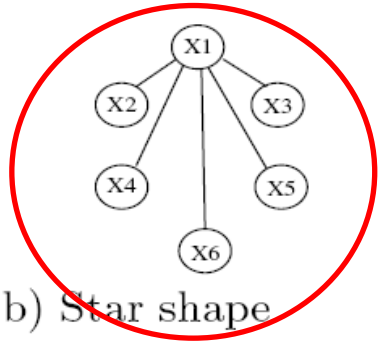


Different Connectivity Structures



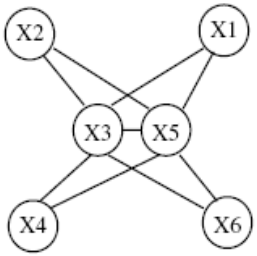
a) Constellation

Fergus et al. '03
Fei-Fei et al. '03



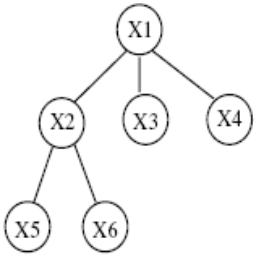
b) Star shape

Leibe et al. '04, '08
Crandall et al. '05
Fergus et al. '05



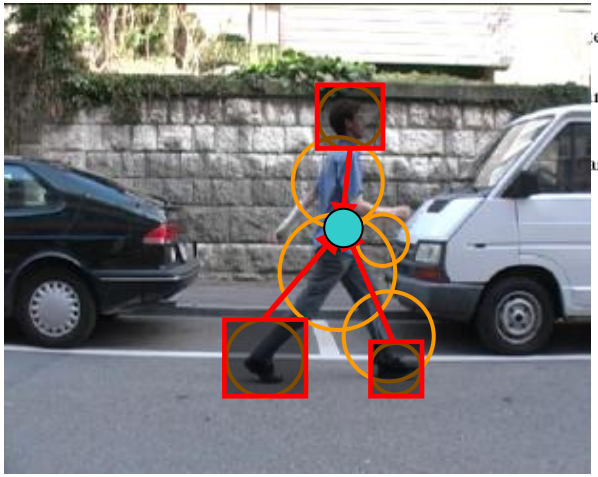
c) k-fan ($k = 2$)

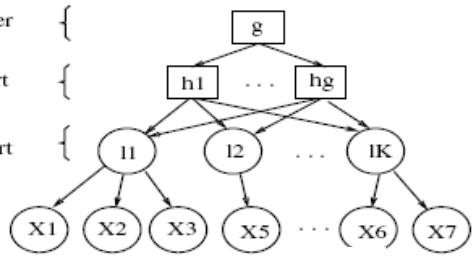
Crandall et al. '05



d) Tree

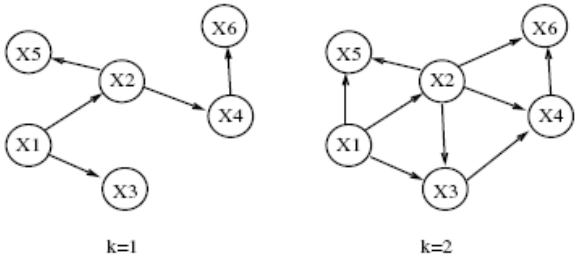
Felzenszwalb & Huttenlocher '05





f) Hierarchy

Bouchard & Triggs '05



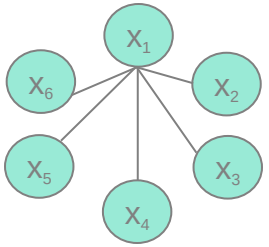
g) Sparse flexible model

Carneiro & Lowe '06

ISM – Basic Ideas

1. Automatically learn a large number of local parts that occur on the object
   - also referred to as visual vocabulary or appearance codebook

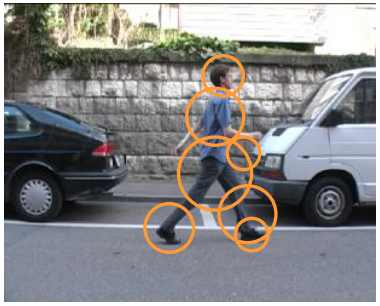
2. Learn a star-topology structural model
   - features are considered independent given the objects' center

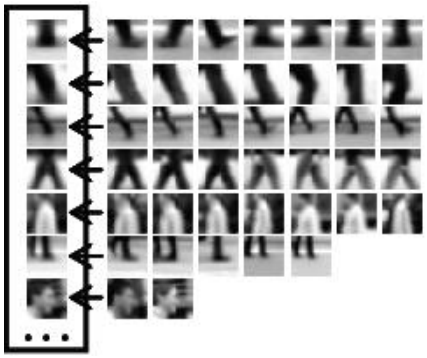


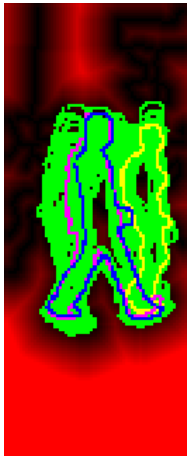
K. Grauman, B. Leibe

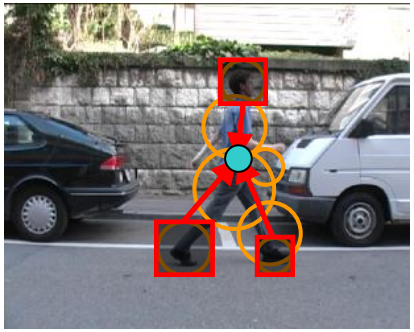
Implicit Shape Model

1. Part Detection/Localization
2. Part Description
3. Learning Part Appearances
4. Learning the Spatial Layout of Parts
5. Combination of Part Detections
6. Verification
7. Extensions









So far

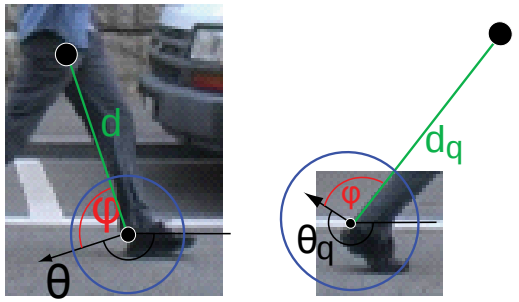
- Parts were defined manually
- Parts represented somewhat the semantic structure
  - i.e. face, leg etc.
- Questions:
  - Do these parts decompose the variability in an optimal way?
    - Must the parts have a semantic meaning?
    - Should we use smaller/larger parts?
  - Can we find parts automatically?

Requirements for a good part decomposition

- Repeatable
  - i.e. we should be able to find the part despite articulation or image transformations (e.g. rotation, perspective, lighting)
- Distinctive
  - a part should not be easily confused with other parts
  - the regions should contain an “interesting” structure
- Compact
  - typically no lengthy or strangely shaped parts
- Efficient
  - it should be computationally inexpensive to detect or represent part
- Cover
  - parts need to sufficiently cover the object

Going local

- Local Feature Approaches
  - use a large number of parts (typically 100-10000 parts)
  - parts are generated automatically
  - parts have mostly no direct semantic meaning
- Let the algorithm find its own parts
- Typically smaller parts

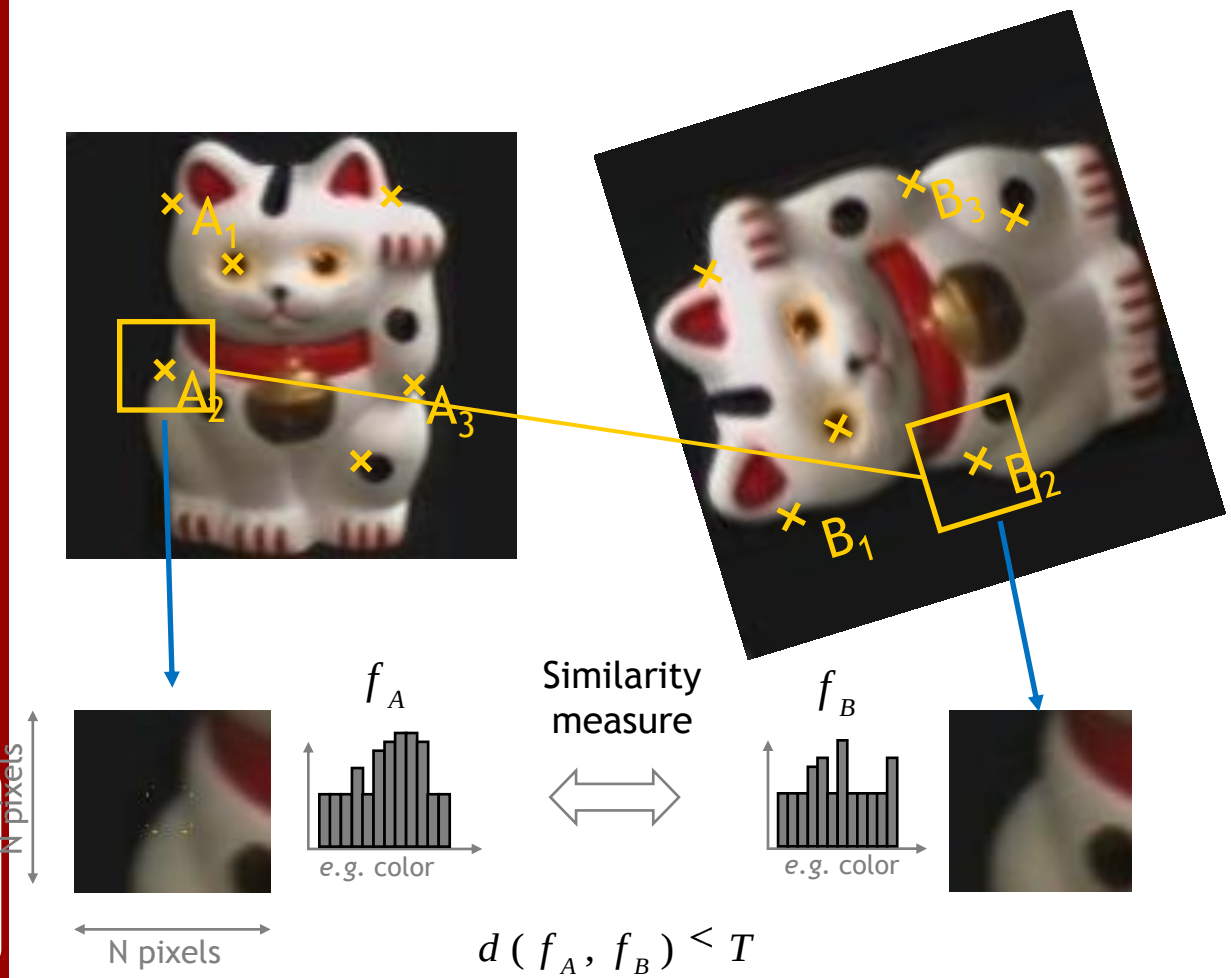


Local Features

Keypoints and descriptors

- We distinguish
  - key or interest points
  - local (key point) descriptors
- Interest Points → Where?
  - specify repeatable points on the object
  - consist of x-, y-position and scale
- Local Descriptors → How does it look like?
  - describe the area around an interest point
  - i.e. define the feature representation of an interest point

Approach



1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

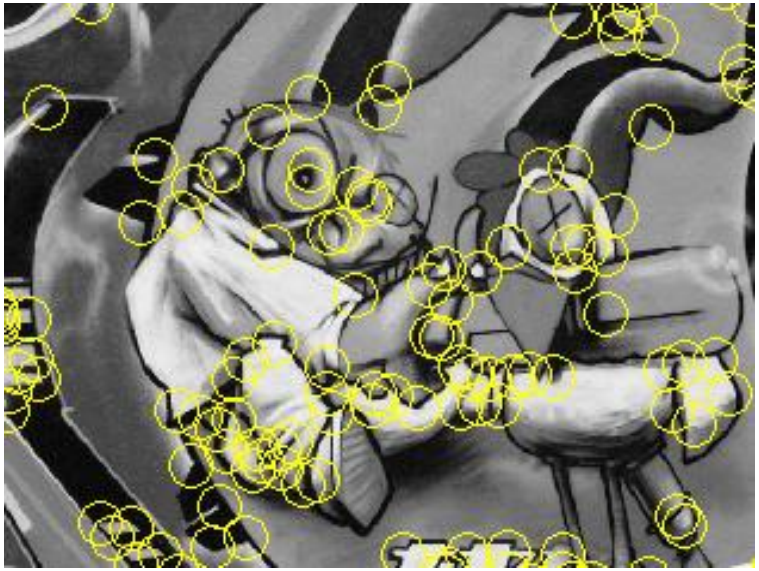
Local Features Part I

Key Point Detectors

Key Point Detectors

- Many Existing Detectors Available
  - Hessian & Harris [Beaudet ‘78], [Harris ‘88]
  - Laplacian, DoG [Lindeberg ‘98], [Lowe 1999]
  - Harris-/Hessian-Laplace [Mikolajczyk & Schmid ‘01]
  - Harris-/Hessian-Affine [Mikolajczyk & Schmid ‘04]
  - EBR and IBR [Tuytelaars & Van Gool ‘04]
  - MSER [Matas ‘02]
  - Salient Regions [Kadir & Brady ‘01]
  - Others…
- Reference site:
  - http://www.robots.ox.ac.uk/~vgg/research/affine/index.html

Keypoint Localization



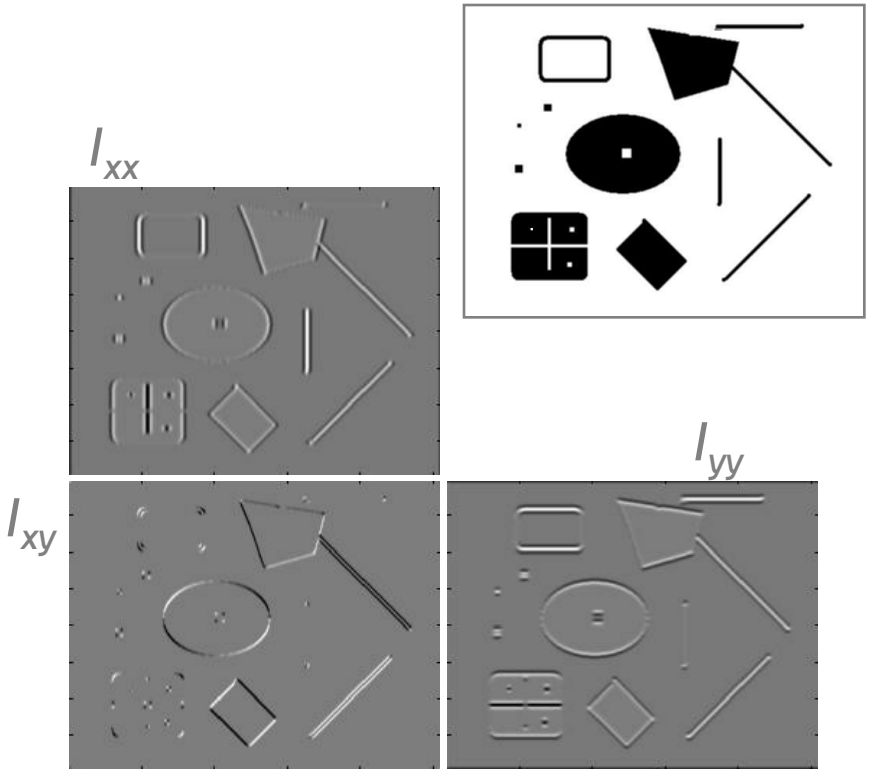
- Goals:
  - repeatable detection
  - precise localization
  - interesting content

→ *Look for two-dimensional signal changes*

Hessian Detector [Beaudet78]

- Hessian determinant

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



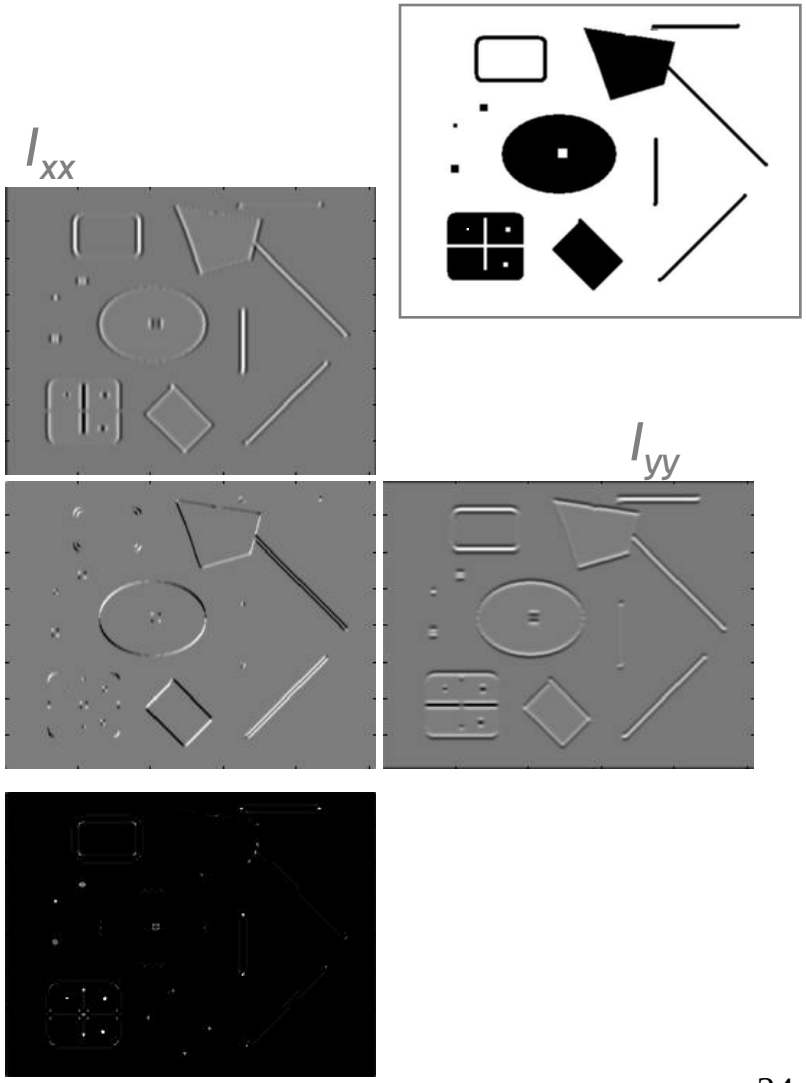
Intuition: **Search for strong derivatives in two orthogonal directions**

Hessian Detector [Beaudet78]

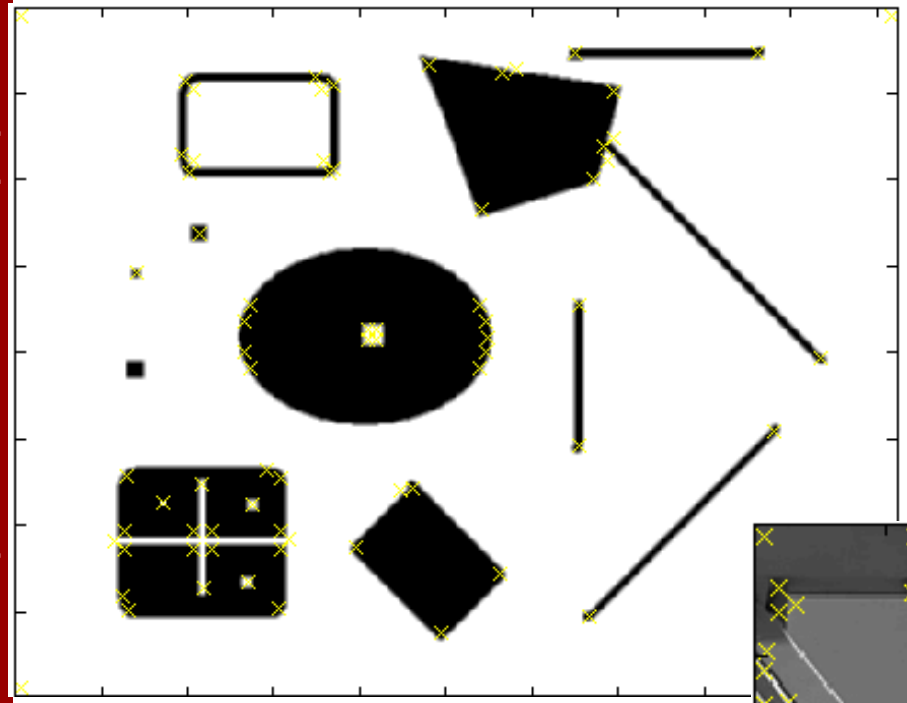
- Hessian determinant

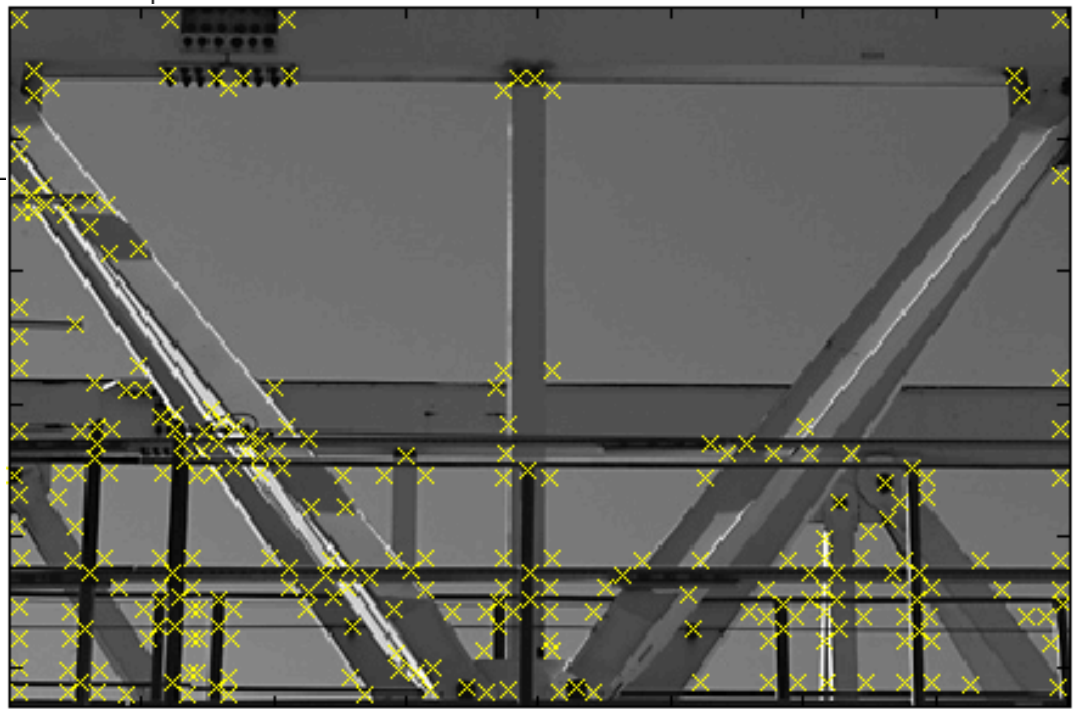
$$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

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



Hessian Detector – Responses [Beaudet78]





Effect: **Responses mainly on corners and strongly textured areas.**

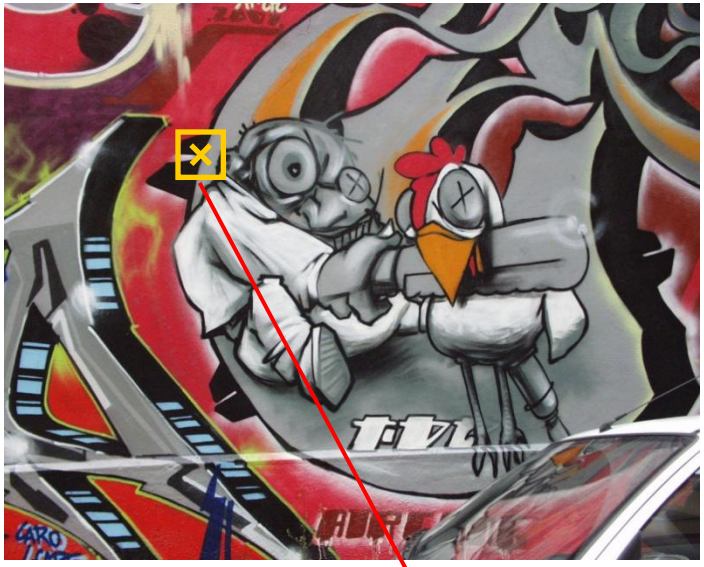
Computer Vision for Human-Computer Interaction
Research Group, Universität Karlsruhe (TH)
cv:hci

Hessian Detector – Responses [Beaudet78]

Scale Space

- So far, we can detect repeatable points in the image
- Now what about the image scale?
- Can we not only detect a distinctive position, but also a characteristic scale around an interest point?

Automatic Scale Selection

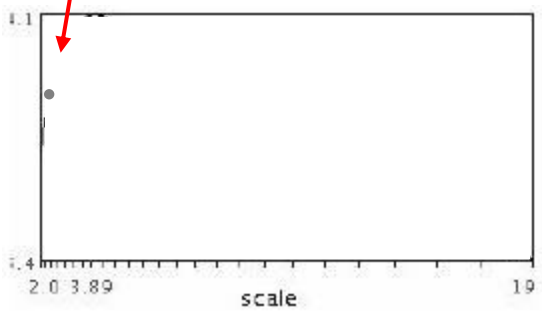


$$f(I_{i_1 \dots i_m}(x, \sigma)) = f(I_{i_1 \dots i_m}(x', \sigma'))$$

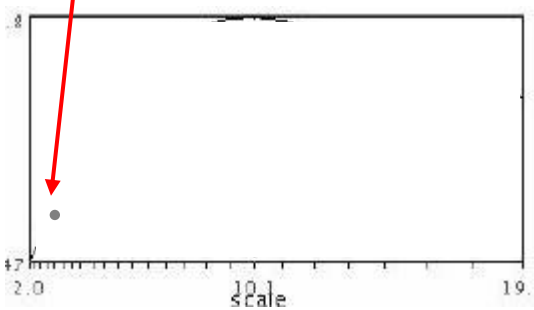
Same operator responses if the patch contains the same image up to scale factor

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



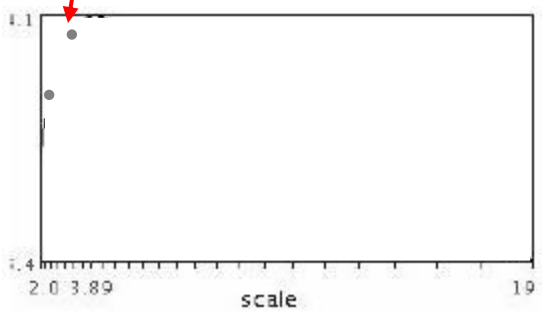
$f(I_{i_1 \dots i_m}(x, \sigma))$



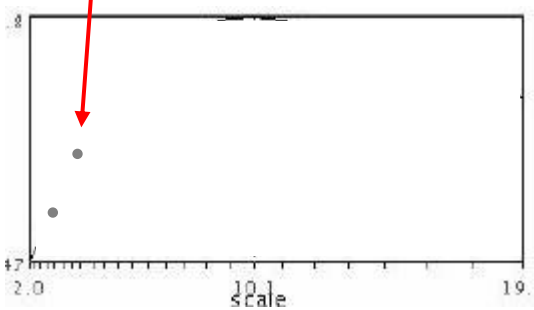
$f(I_{i_1 \dots i_m}(x', \sigma))$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



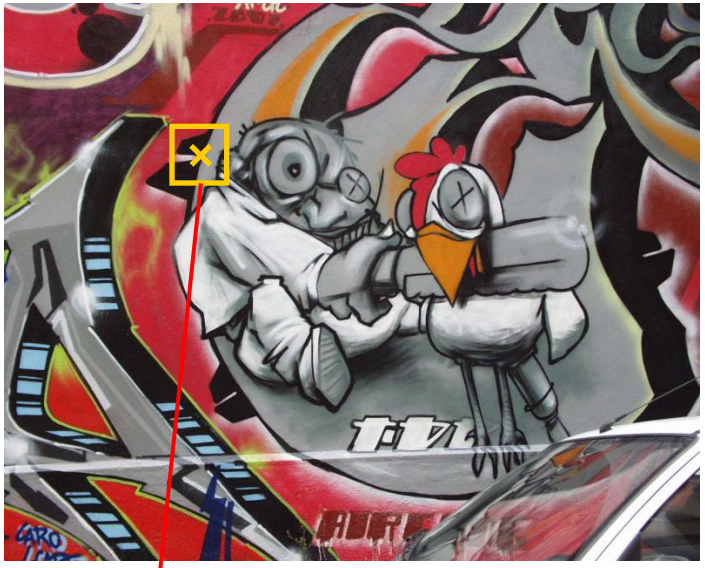
$f(I_{i_1 \dots i_m}(x, \sigma))$

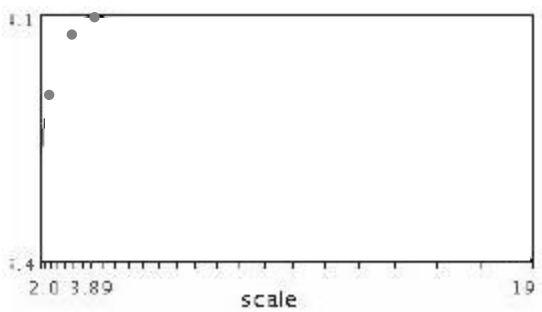


$f(I_{i_1 \dots i_m}(x', \sigma))$

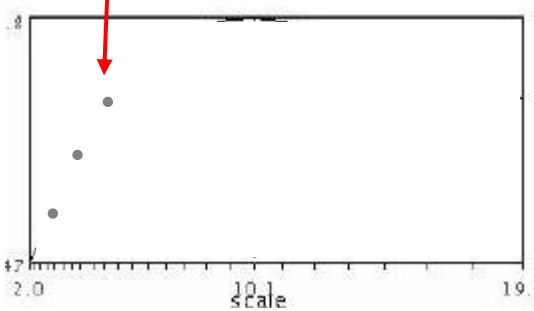
Automatic Scale Selection

- Function responses for increasing scale (scale signature)





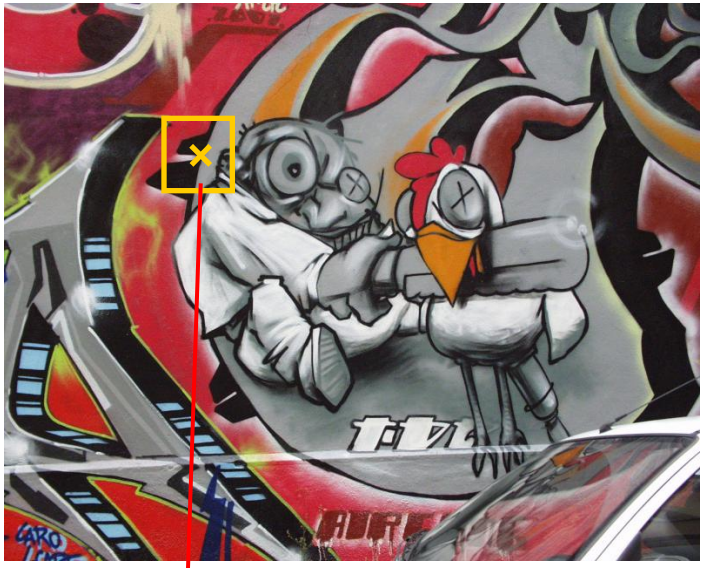
$f(I_{i_1 \dots i_m}(x, \sigma))$

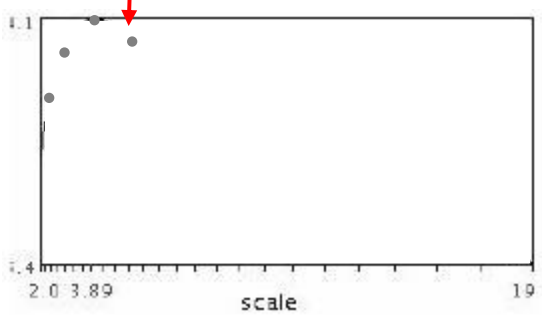


$f(I_{i_1 \dots i_m}(x', \sigma))$

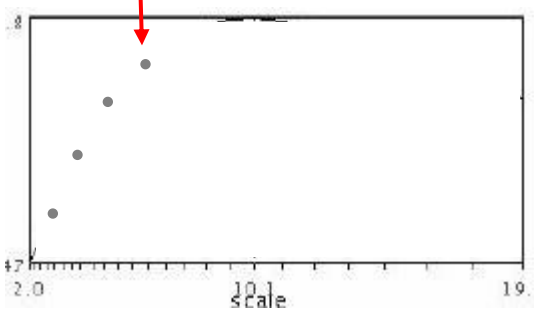
Automatic Scale Selection

- Function responses for increasing scale (scale signature)





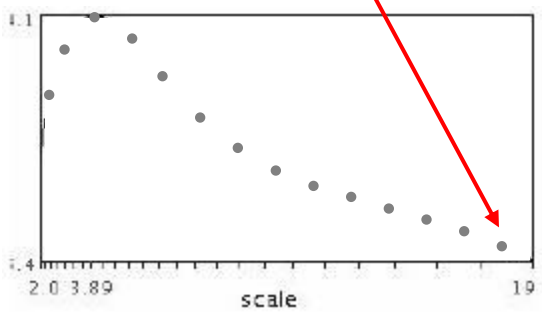
$f(I_{i_1 \dots i_m}(x, \sigma))$



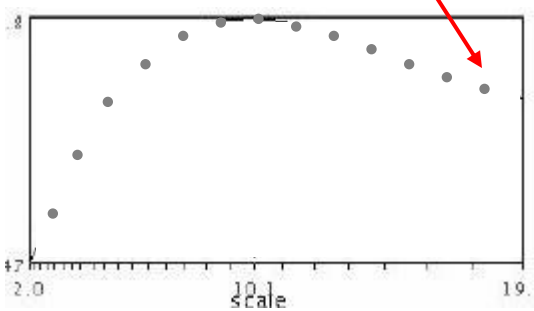
$f(I_{i_1 \dots i_m}(x', \sigma))$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



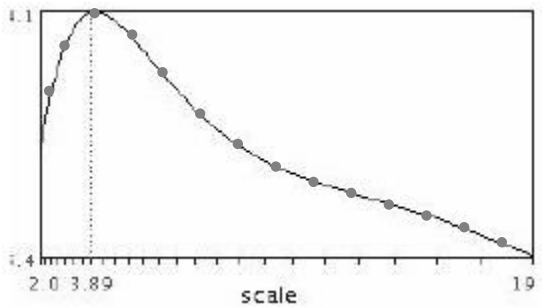
$f(I_{i_1 \dots i_m}(x, \sigma))$



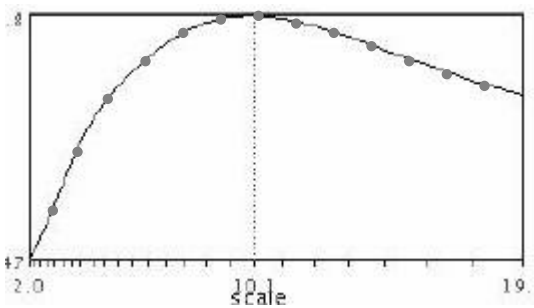
$f(I_{i_1 \dots i_m}(x', \sigma))$

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



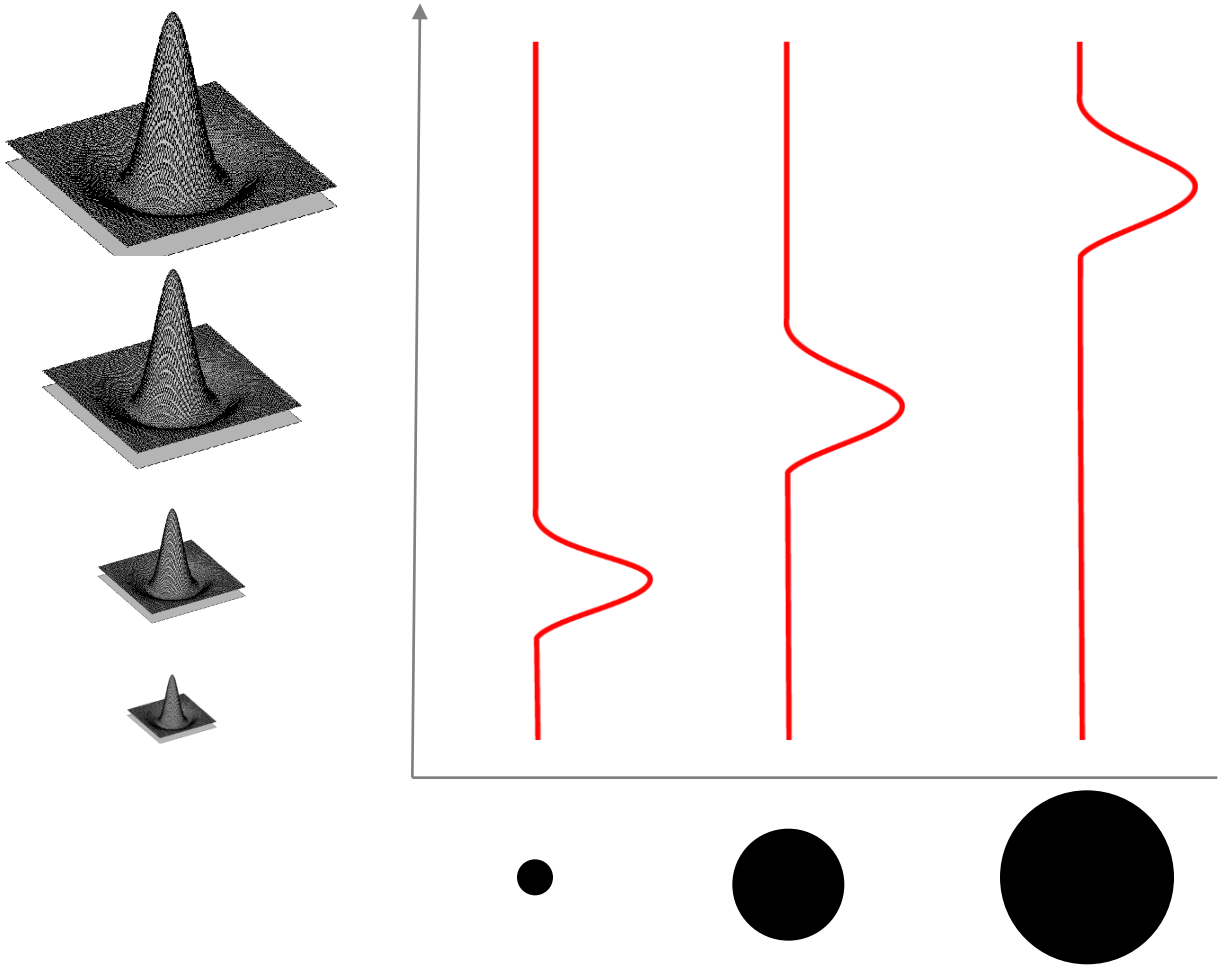
$f(I_{i_1 \dots i_m}(x, \sigma))$



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

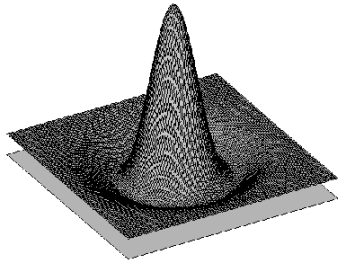
What Is A Useful Signature Function?

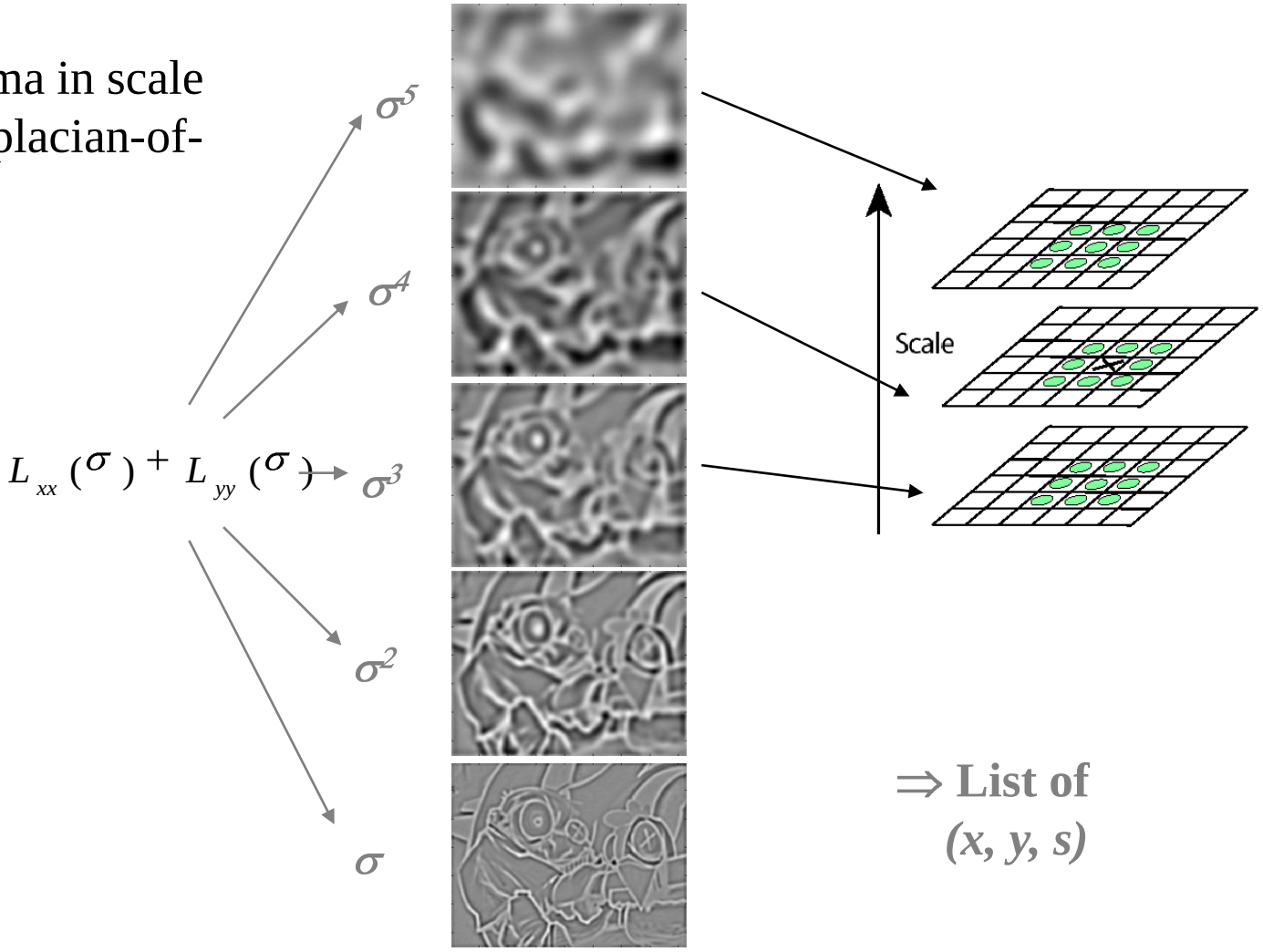
- Laplacian-of-Gaussian = “blob” detector



Laplacian-of-Gaussian (LoG)

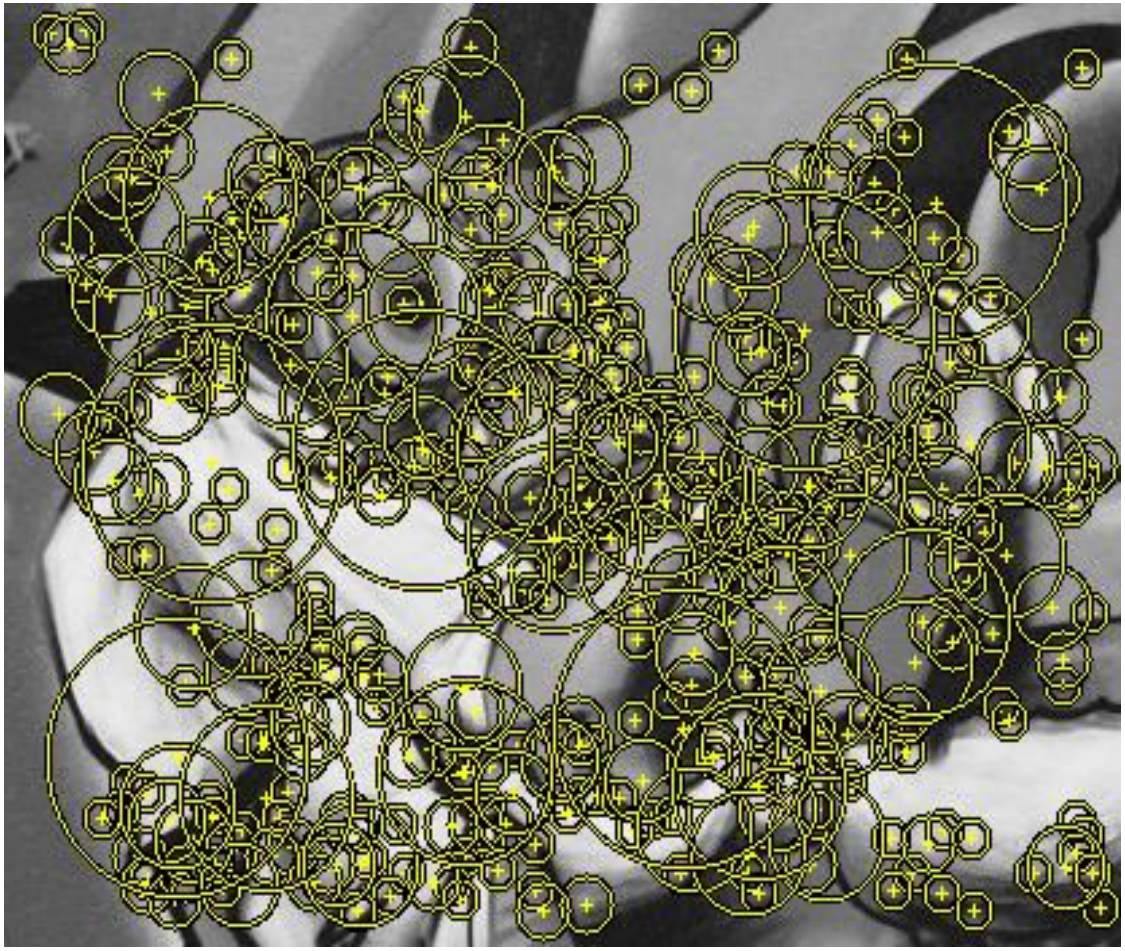
- Local maxima in scale space of Laplacian-of-Gaussian





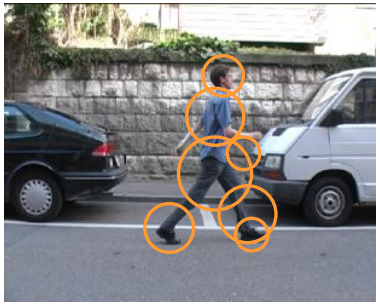
⇒ List of
(x, y, s)

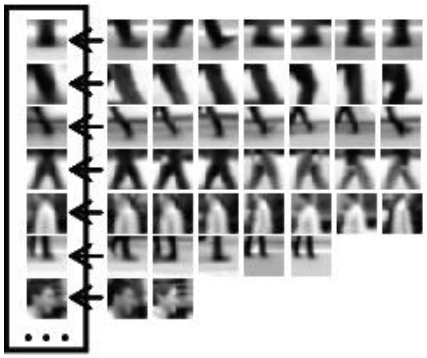
Results: Laplacian-of-Gaussian

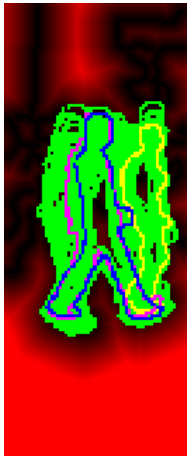


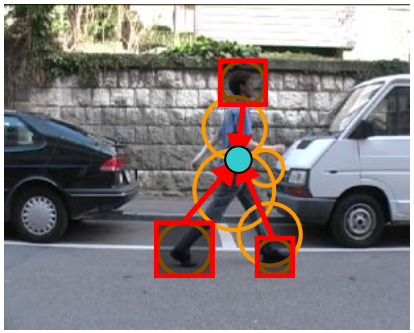
Implicit Shape Model

1. Part Detection/Localization
2. Part Description
3. Learning Part Appearances
4. Learning the Spatial Layout of Parts
5. Combination of Part Detections
6. Verification
7. Extensions









Local Features Part II

Local Descriptors

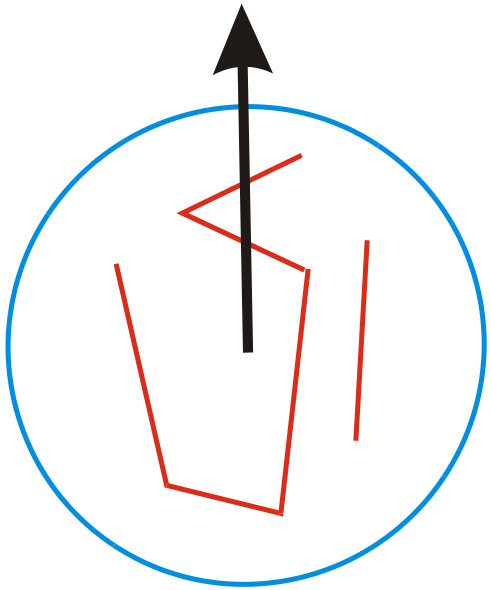
Local Descriptors

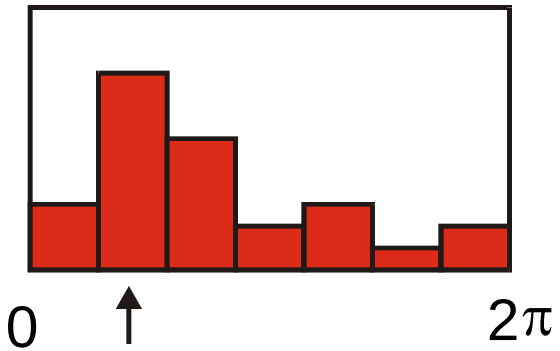
- Goal:
  - Describe (local) region around a keypoint
- Most available descriptors focus on edge/gradient information
  - Capture boundary and texture information
  - Color still used relatively seldom (more suitable for homogenous regions)

Orientation Normalization

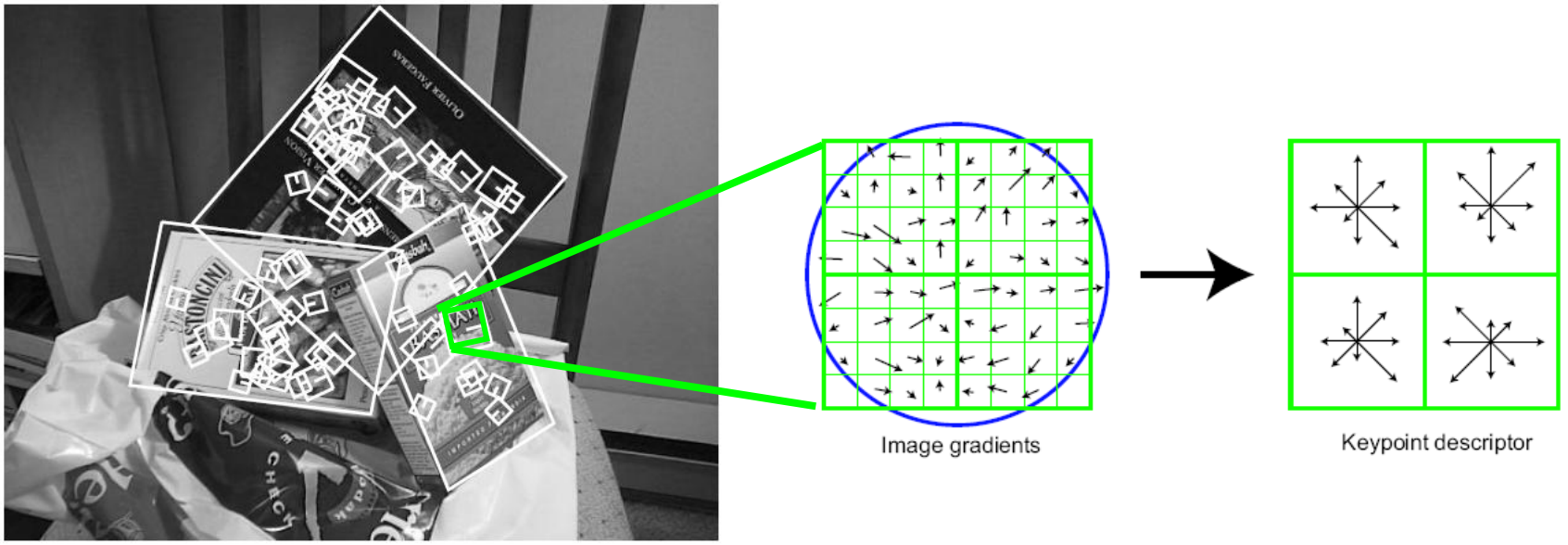
[Lowe, SIFT, 1999]

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





Local Descriptors: SIFT Descriptor

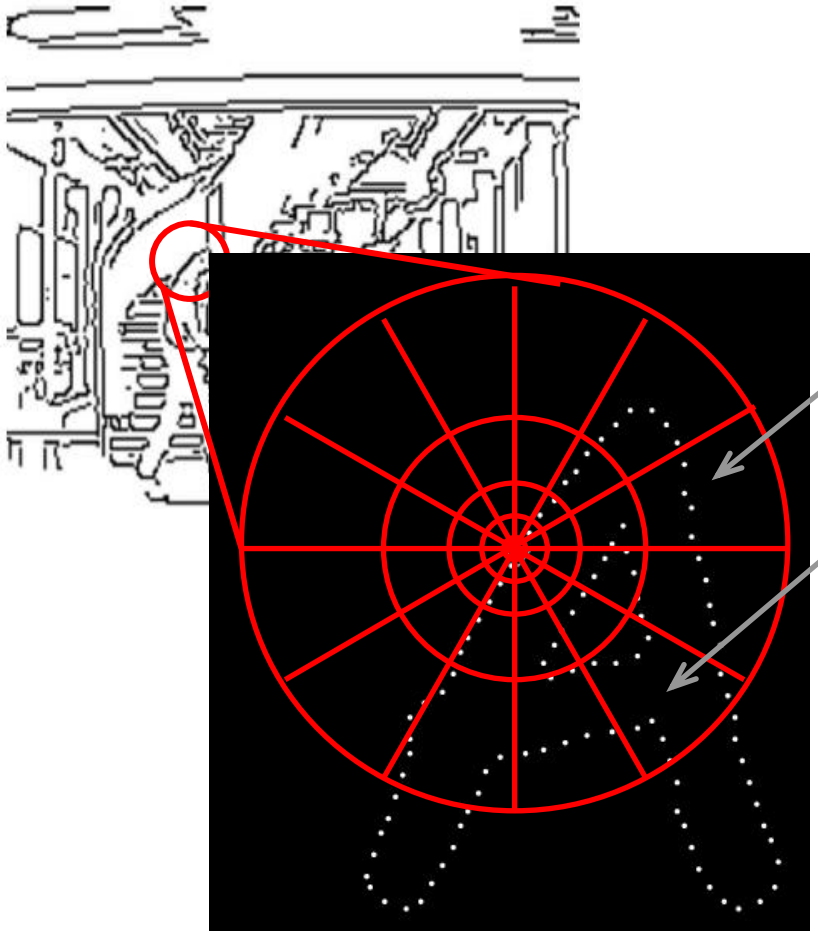


Histogram of oriented gradients

- captures important texture information
- robust to small translations / affine deformations

[Lowe, ICCV 1999]

Local Descriptors: Shape Context



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

Count = 4

⋮

Count = 10

Log-polar binning: more precision for nearby points, more flexibility for farther points.

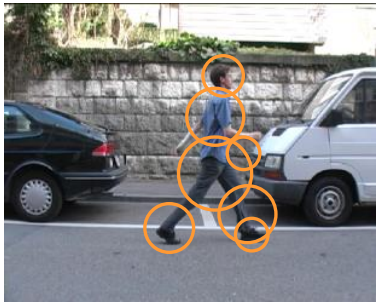
Belongie & Malik, ICCV 2001

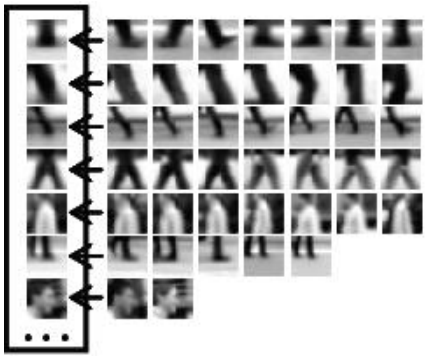
So, What Local Features Should I Use?

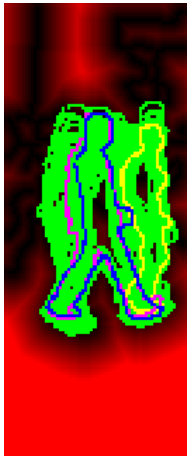
- There have been extensive evaluations/comparisons
  - [Mikolajczyk et al., IJCV'05, PAMI'05]
  - all detectors/descriptors shown there work well
- Best choice often application dependent
  - Harris-/Hessian-Laplace/DoG work well for many natural categories
- More features are better
  - combining several detectors often helps

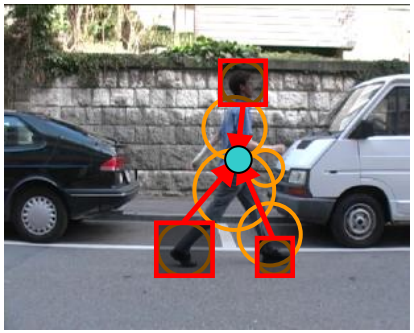
Implicit Shape Model

1. Part Detection/Localization
2. Part Description
3. Learning Part Appearances
4. Learning the Spatial Layout of Parts
5. Combination of Part Detections
6. Verification
7. Extensions



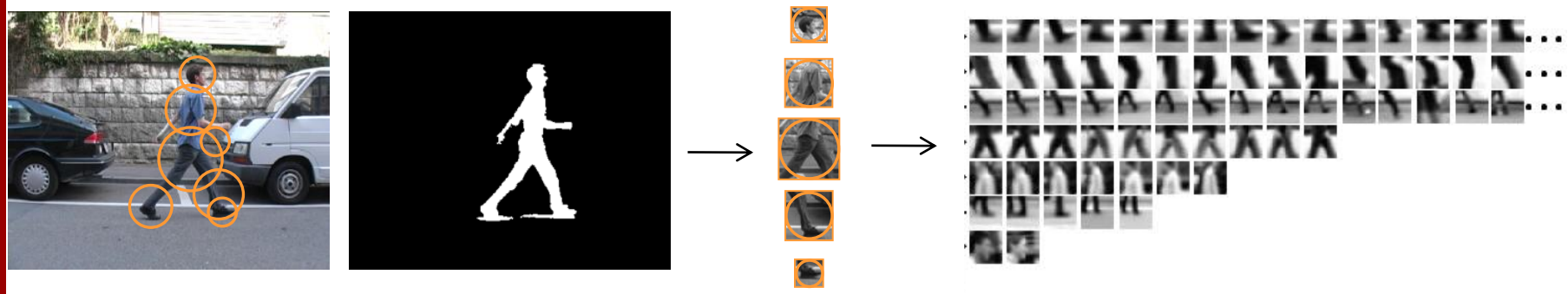






Visual Vocabulary / Appearance Codebook

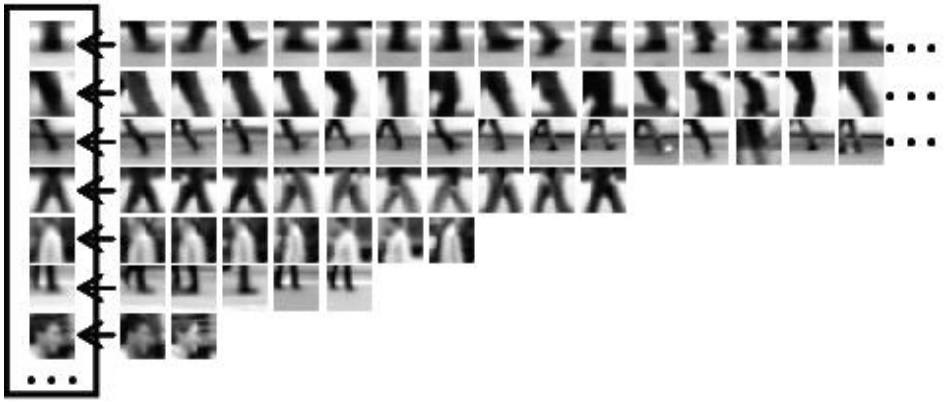
Visual Vocabulary



1. Detect keypoints on all training examples
2. Extract feature descriptions around keypoints

- Result: A large set of local image descriptors occurring on people

Visual Vocabulary



- Group visually similar local descriptors
  - i.e. parts that are reoccurring
  - parts, that occur only once are discarded (they could result from noise or background structures)

Side Note: Grouping Algorithms

- Partitional Clustering
  - K-Means
  - Gaussian Mixture Clustering (EM)
- Hierarchical or Agglomerative Clustering
  - Single-Link (minimum)
  - Group Average
  - Ward's method (minimum variance)

Agglomerative Clustering

- Algorithm (Average-Link)
  1. Start with each patch as a cluster of its own
  2. Merge the two most similar clusters P and Q, where the similarity between two clusters is defined as the average similarity between their members

$$dist\ (P, Q) = \frac{1}{|P||Q|} \sum_i \sum_j dist\ (p_i, q_j)$$

  3. Go to 2, unless $dist\ (P, Q) > \theta$

- Commonly used similarity measures
  - normalized correlation
  - euclidean distances

Complexity

- Time complexity
  - standard approach: $O(n^2 \log n)$
    - compute distance matrix
    - consecutively merge the two most similar clusters
- Space complexity:
  - $O(n^2)$
  - note, that space complexity is quite important for clustering large data sets
  - Example: 100 000 data points
  - Standard distance matrix contains: $10^5 * 10^5 = 10^{10}$ entries
    => ~40 GB if one entry has 32 bit

Reciprocal Nearest Neighbor (RNN)

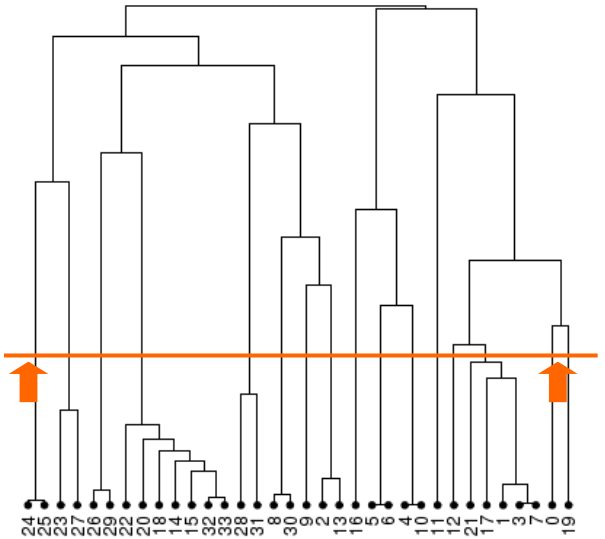
- RNN Algorithm [de Rham'80, Benzecri'82]
  - time complexity: $O(n^2)$
  - space complexity: $O(n)$
  - requirement: "reducibility property" [Bruynooghe'77]

$$d(c_i, c_j) \leq \inf(d(c_i, c_k), d(c_j, c_k)) \Rightarrow \inf(d(c_i, c_k), d(c_j, c_k)) \leq d(c_i \cup c_j, c_k)$$

The reducibility property effectively states that the agglomeration of a reciprocal nearest-neighbor pair does not alter the nearest-neighbor relations of any other cluster. It is easy to see that this property is fulfilled, among others, for the Group Average criterion (regardless of the employed similarity measure) and the Centroid criterion based on correlation (however, it is not fulfilled for the Centroid criterion based on Euclidean distances).

Clustering Hierarchy

- Agglomerative clustering produces a hierarchy

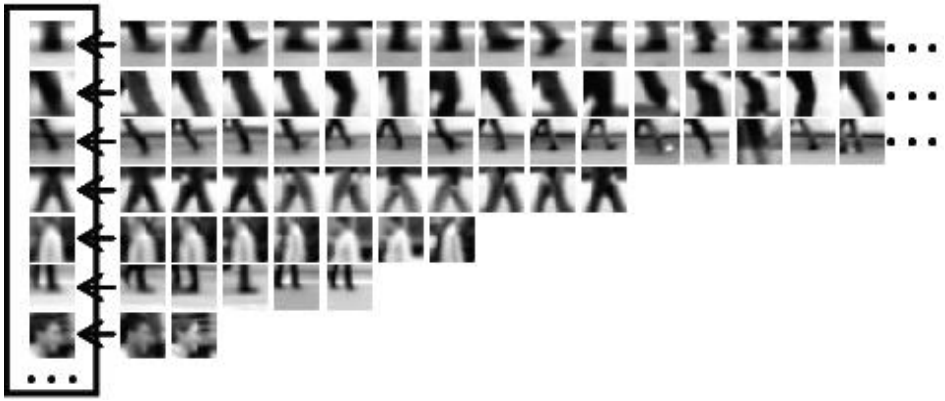


- When/where to stop clustering?
  - ideally, clusters should be visually compact

- But
  - distance value depends on feature dimensionality
  - appropriate ratio #features / #clusters depends on data set and interest point detector

→ Needs to be selected for each detector/descriptor combination!

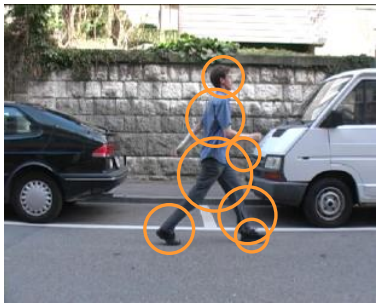
Visual Vocabulary

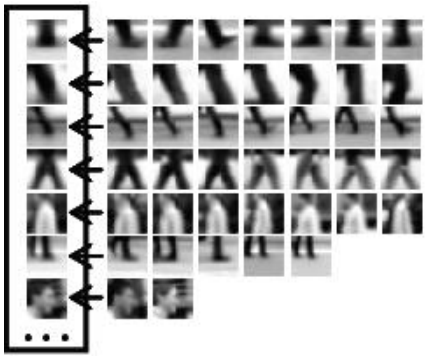


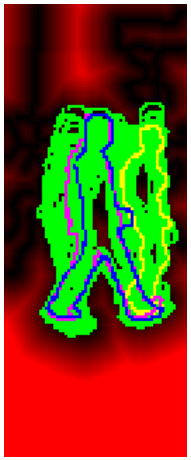
- Vocabulary size ~10000 clusters
  - probabilistic votes decide, whether part is important or not

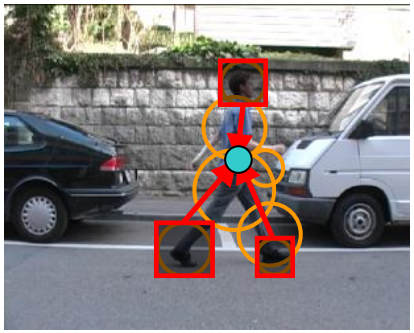
Implicit Shape Model

1. Part Detection/Localization
2. Part Description
3. Learning Part Appearances
4. Learning the Spatial Layout of Parts
5. Combination of Part Detections
6. Verification
7. Extensions



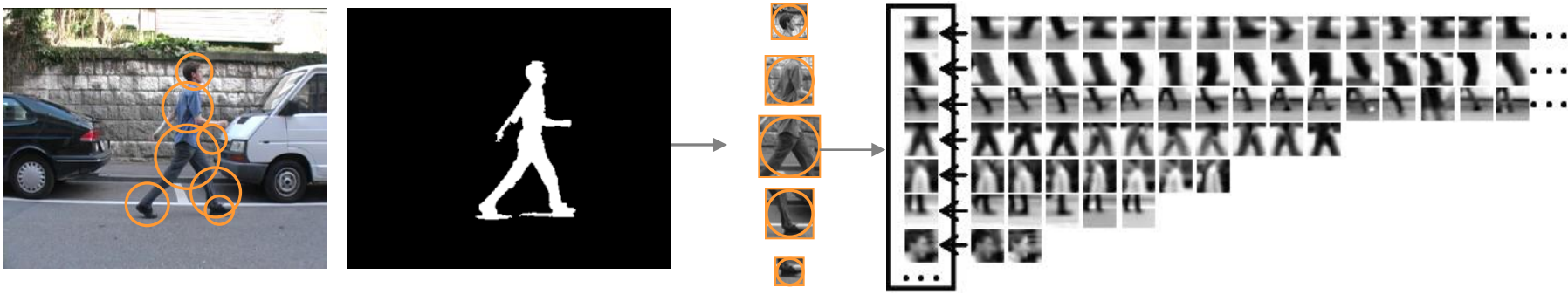






Learning Spatial Structure: "Star"-Model

Implicit Shape Model - Representation

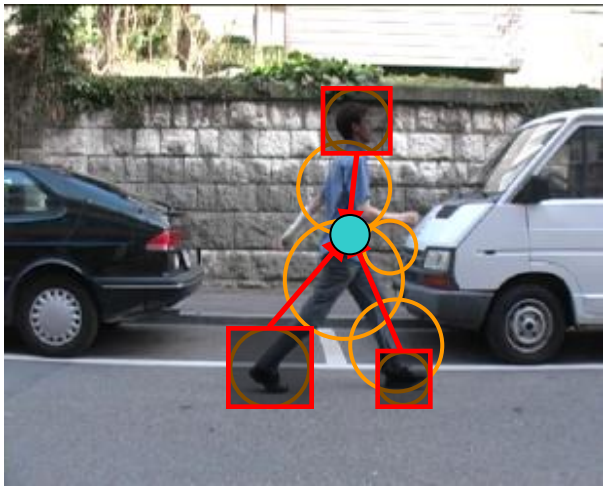


1. Learn appearance codebook
   - extract local features at interest points
   - agglomerative clustering ⇒ codebook
2. Learn spatial distributions
   - match codebook to training images
   - record matching positions on object

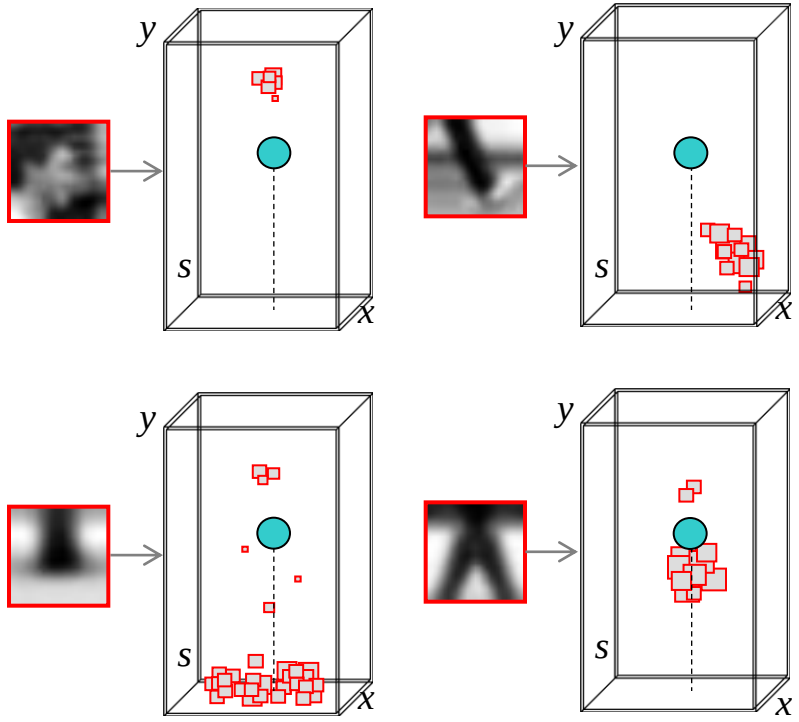
- Sparse representation of the object appearance

Training: Spatial Occurrence (Star-Model)

1. Record spatial occurrence
   - match codebook to training images
   - record occurrence distributions with respect to object center
     - location (x, y) and scale



Star-Model



Spatial occurrence distributions

Occurrence Distribution

- For each codebook entry, we obtain a non-parametric probability distribution of its position relative to the object center

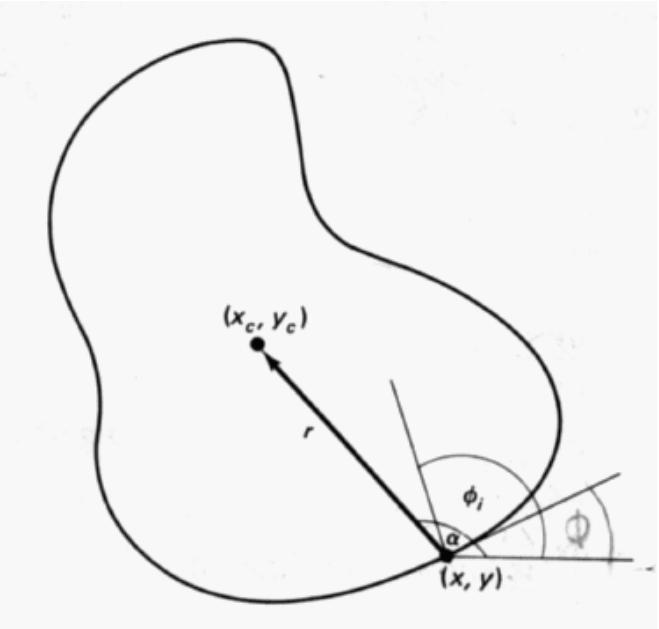
$$P(o_n, \lambda | c_i)$$

- With
  - o_n the object (person)
  - c_i a codebook entry
  - $\lambda = (\lambda_x, \lambda_y, \lambda_s)$ the relative position and scale

Remember: Generalized Hough Transform

[Ballard81]

- Choose reference point for the contour (e.g. center)
- For each point on the contour remember where it is located w.r.t. to the reference point
- Remember radius ***r*** and angle ϕ relative to the contour tangent
- Recognition: whenever you find a contour point, calculate the tangent angle and 'vote' for all possible reference points

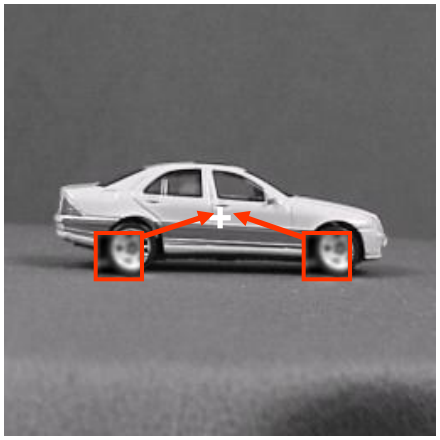


- Instead of reference point, can also vote for transformation

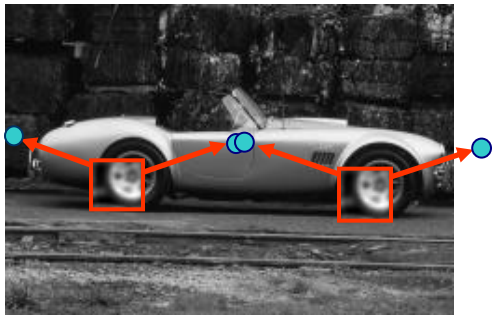
$\Rightarrow$ The same idea can be used with local features!

Generalized Hough Transform

- For every feature, store possible "occurrences"



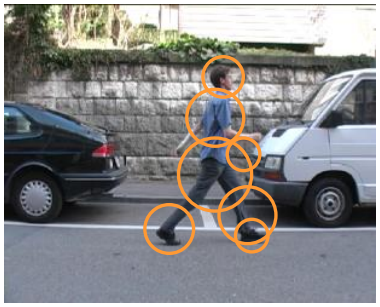
For new image, let the matched features vote for possible object positions

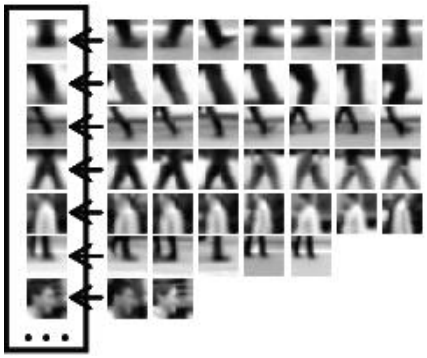


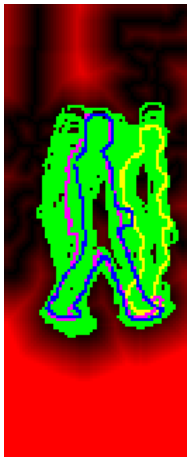
- Object identity
- Pose
- Relative position

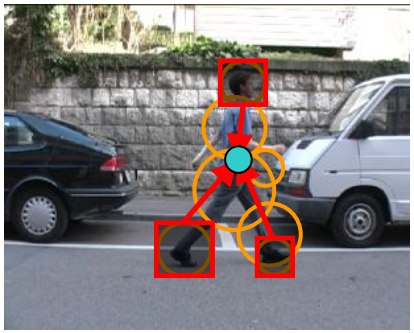
Implicit Shape Model

1. Part Detection/Localization
2. Part Description
3. Learning Part Appearances
4. Learning the Spatial Layout of Parts
5. Combination of Part Detections
6. Verification
7. Extensions





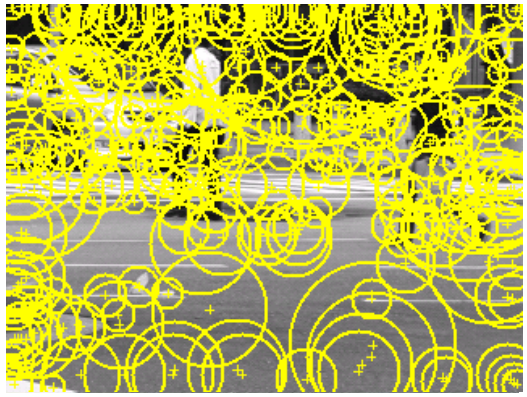




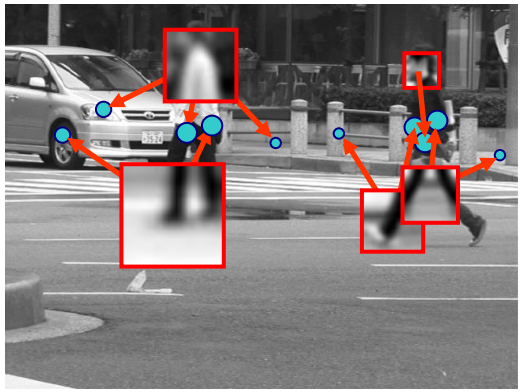
Detection Procedure

Recognition: ISM Detection Procedure

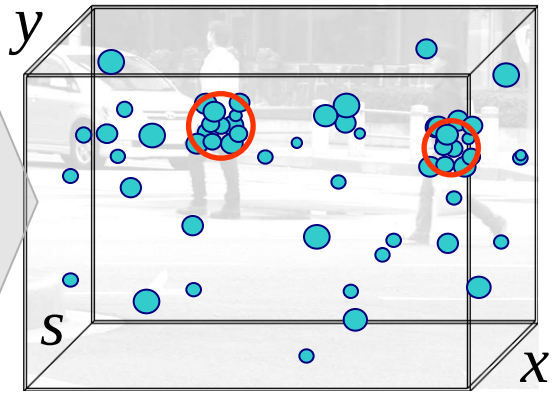
Image

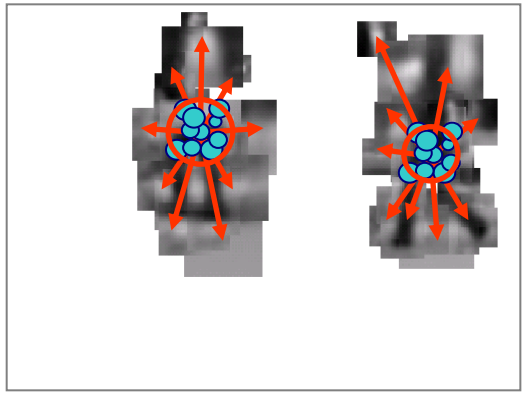


Probabilistic Voting

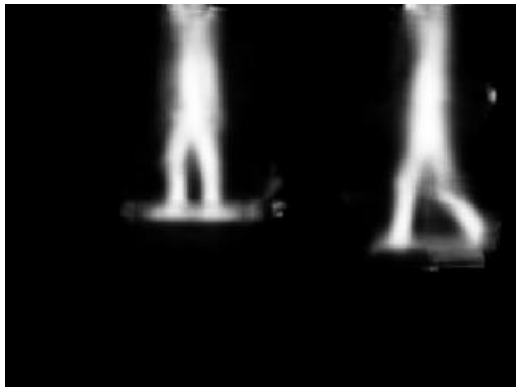


3D Voting Space

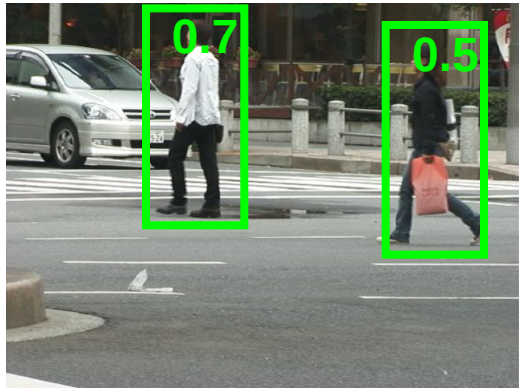




Back-Projection



Segmentation



Detection Confidences

Probabilistic Formulation

- Descriptor contribution:

$$p(o_n, \lambda | e, \ell) = \sum_i p(o_n, \lambda | c_i, \ell) p(c_i | e)$$

- With
  - e an extracted image descriptor
  - l the position of the descriptor in the image

- Marginalization over all found descriptors gives the probability or detection confidence of an object at any location λ

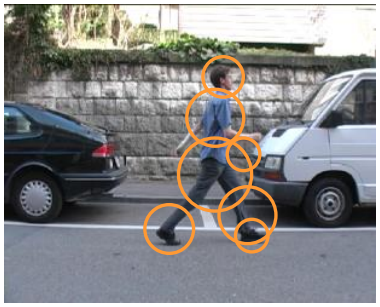
$$p(o_n, \lambda) = \sum_k p(o_n, \lambda | e_k, \ell_k)$$

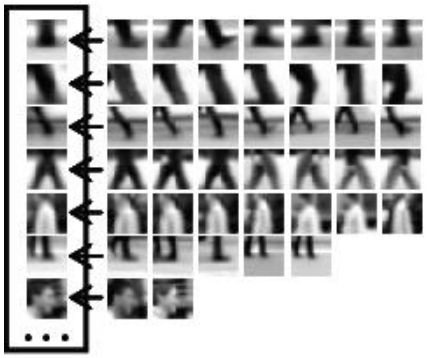
Implicit Shape Model – Next Time

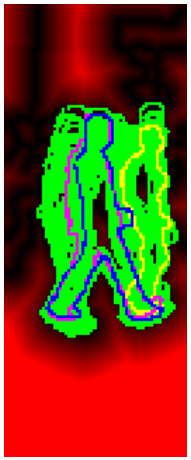
1. Part Detection/Localization
2. Part Description
3. Learning Part Appearances
4. Learning the Spatial Layout of Parts
5. Combination of Part Detections

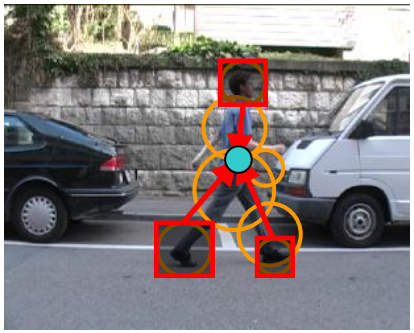
---

6. Verification
7. Extensions









End of lecture