Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN-Fakultät
Fachbereich Informatik
Arbeitsbereich SAV/BV (KOGS)

Image Processing 1 (IP1)
Bildverarbeitung 1

Lecture 22: Object Recognition 2

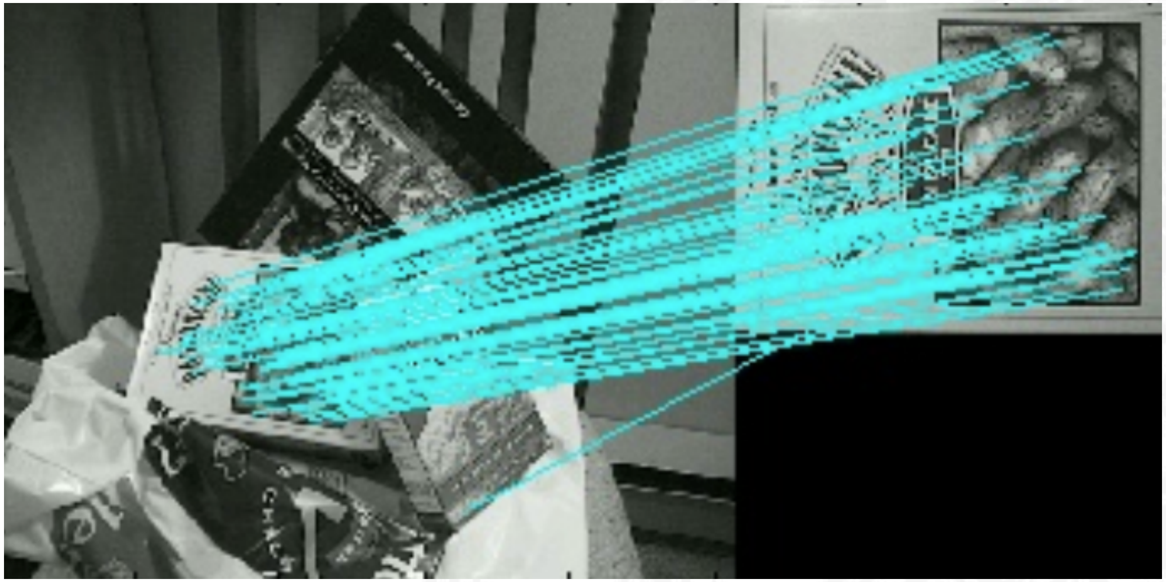
Winter Semester 2015/16

Slides: Prof. Bernd Neumann
Slightly revised by: Dr. Benjamin Seppke & Prof. Siegfried Stiehl

Object Recognition with Local Descriptors

Basic idea:

- Determine interest points in model images
- Determine invariant local image properties around interest points
- Use local image properties for finding matching objects



Matching images using SIFT features
(SIFT = Scale-Invariant Feature Transform)

SIFT Method

David G. Lowe: Distinctive Image Features from Scale-Invariant Keypoints
International Journal of Computer Vision, 2004 (Protected by US patent)

Lowe developed specific methods for:

1. Determining invariant local descriptors at interest points
    - finding stable interest points ("keypoints")
    - computing largely scale-invariant features at interest points
2. Extracting stable descriptors for object models
3. Finding and recognizing objects based on local descriptors

Determining SIFT Keypoints: Scale Space

Keypoints are local maxima and minima in the DoG of scaled images.

Recall:

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)$$

Convolution of image $I(x, y)$ with Gaussian $G(x, y, k\sigma)$

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma)$$

Difference of Gaussians (DoG)

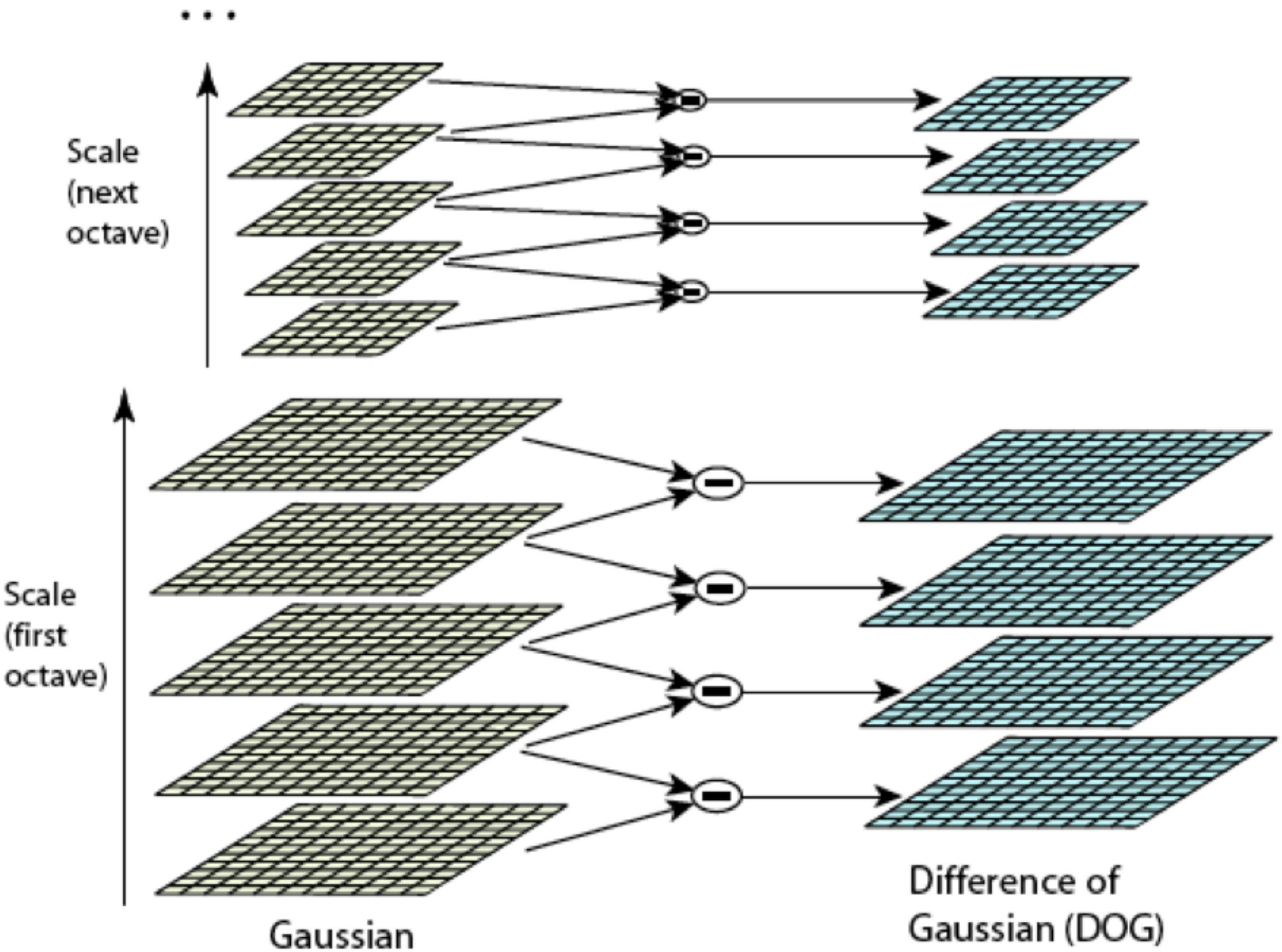
Procedure:

a) Initial image is repeatedly convolved with Gaussians of multiples of σ, forming a scale space.

b) Scaled images within an octave (σ ... 2σ) have same resolution. Adjacent scales are subtracted to produce DoGs.

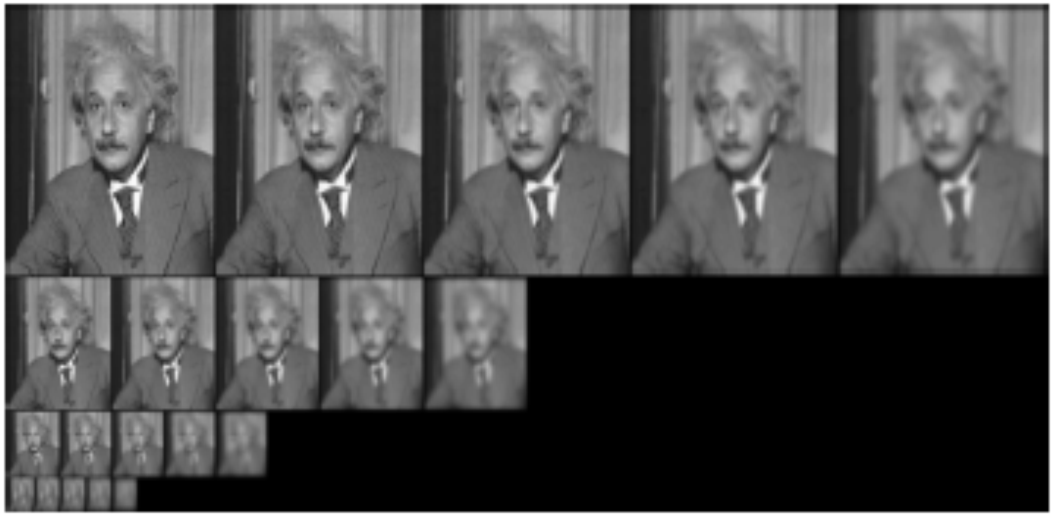
c) Scaled images are down-sampled from one octave to the next.

Illustration of SIFT Scale Space



Example Image in SIFT Scale Space

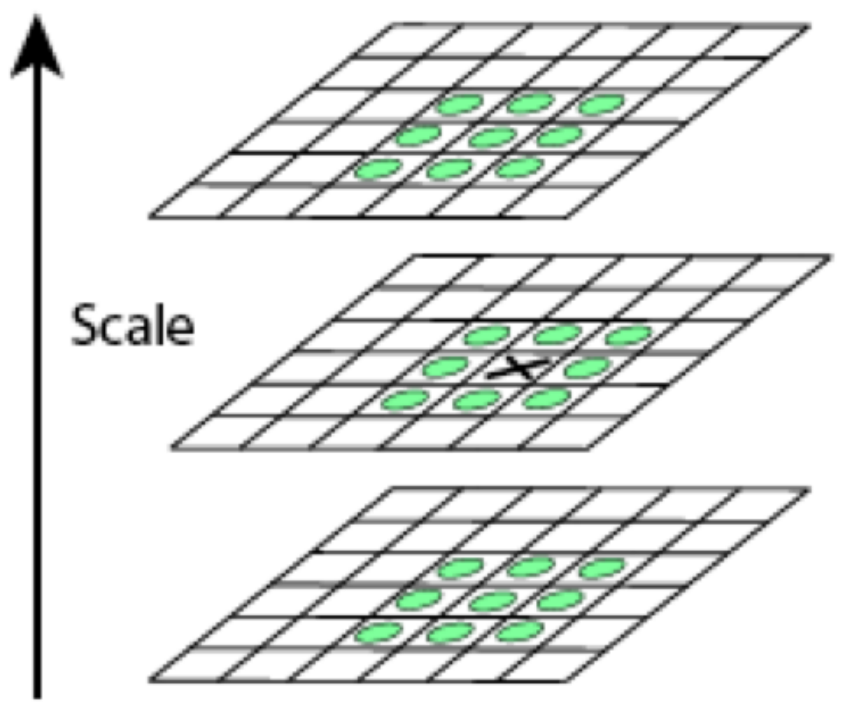
5 Gaussian filtered images per octace



Corresponding DoGs

Determining Extrema

Find local minima and maxima by comparing a DoG pixel to its 26 neighbours in 3x3 regions at the current and adjacent scales.



Sub-pixel Localization of Extrema

- Take extrema of previous step as keypoint candidates
- Determine Taylor expansion at candidate location
- Find subpixel extremum by setting derivatives to zero
- If location of subpixel extremum is within 0.5 of candidate location (in x- or y-direction), keep keypoint at subpixel location, otherwise discard keypoint candidate
- If value of expansion at subpixel location is less than 0.03, discard keypoint

Taylor expansion:

$$D(x,y) = D + x\frac{\partial D}{\partial x} + y\frac{\partial D}{\partial y} + \frac{1}{2}x^2\frac{\partial^2 D}{\partial x^2} + \frac{1}{2}y^2\frac{\partial^2 D}{\partial y^2} + xy\frac{\partial^2 D}{\partial x \partial y}$$

approximated from local neighbourhood

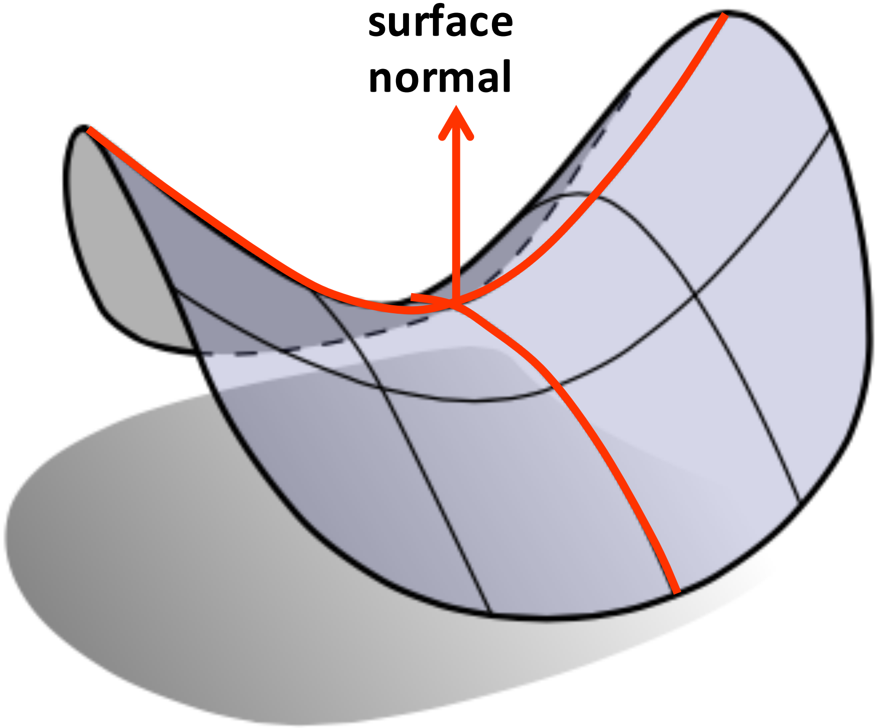
Extrema:

$$x_{ext} = \frac{D_y D_{xy} - D_x D_{yy}}{D_{xx}D_{yy} - D_{xy}^2} \qquad y_{ext} = \frac{D_x D_{xy} - D_y D_{xx}}{D_{xx}D_{yy} - D_{xy}^2} \qquad \text{with} \quad \mathrm{D}_x = \frac{\partial D}{\partial x} \text{ etc.}$$

Eliminating Edge Responses

- Keypoints at strong edges tend to be unstable. Principal curvatures at keypoint must be significant for keypoint to be stable.
- Compute Hessian at keypoint: $H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix}$
- Eigenvalues α and β of H are proportional to principal curvatures.
- Note that $R = \frac{Tr(H)^2}{Det(H)} = \frac{(r+1)^2}{r}$ with $\mathrm{r} = \frac{\alpha}{\beta}$, $\mathrm{tr}(H) = D_{xx} + D_{yy} = \alpha\alpha + \beta$, $\det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\alpha$
- The higher the absolute differences of principal curvatures of D, the higher the value of R.
- Hence if $R > \frac{(r_0+1)^2}{r_0}$ with r_0 as threshold, the keypoint is discarded.

Illustration of Principal Curvatures



Each point of a 3D surface has a maximum and minimum curvature.

Assigning Orientations

Each keypoint is marked by one or more dominant orientations based on image gradient directions computed in a neighbouring region.

Gradient magnitude:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

Gradient direction:

$$\theta(x,y) = \mathrm{atan2}\left[L(x,y+1) - L(x,y-1)\,,\, L(x+1,y) - L(x-1,y)\right]$$

Gradient magnitudes, weighted by a Gaussian of radius 1.5σ, are summed in 36 bins of an orientation histogram. The histogram peak and all other peaks within 80% of the absolute peak value are assigned as dominant keypoint orientations.

Dominant keypoint orientations are used to achieve orientation invariance for object recognition.

Illustration of Keypoint Selection I

233 x 189 greyvalue image

832 keypoint candidates at extrema of DoG images. Vectors show location, orientation and scale.

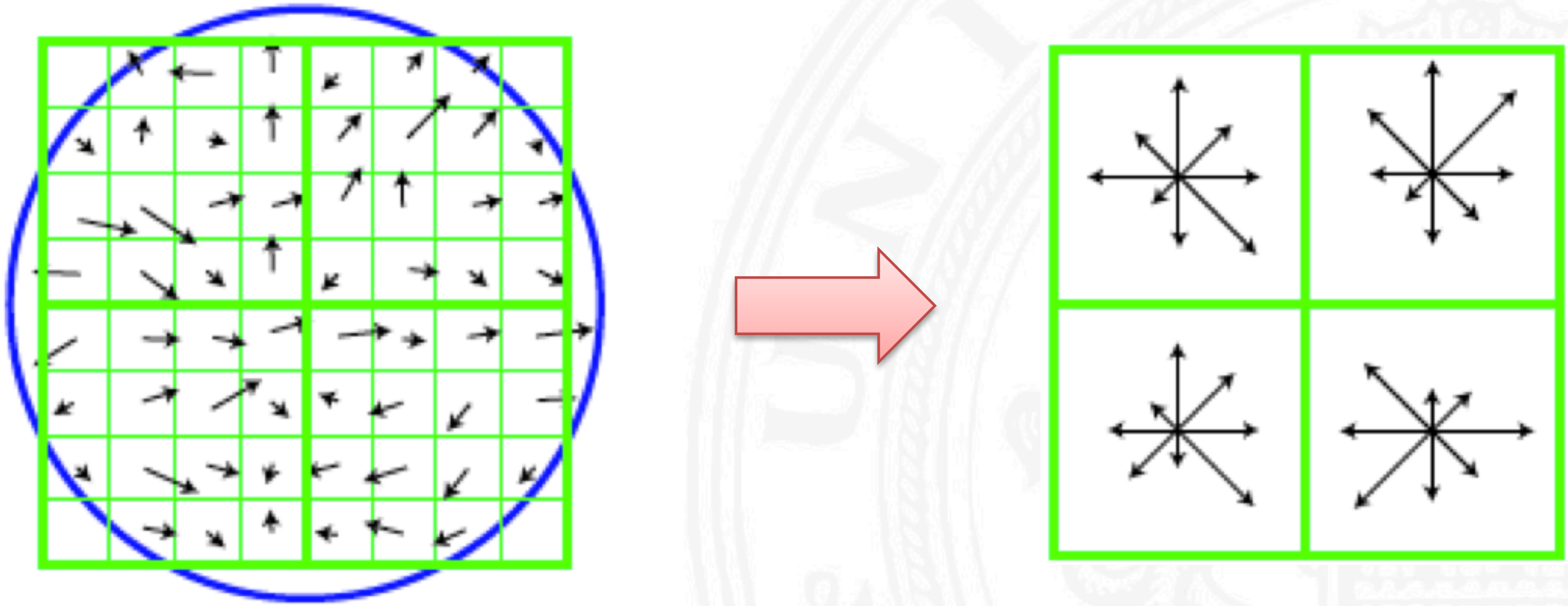
Illustration of Keypoint Selection II

729 keypoints remain after applying threshold on minimum contrast

536 keypoints remain after applying threshold on ratio of principal curvatures

Computing a Keypoint Descriptor

- 4 x 4 orientation histograms with 8 bins each are determined from a 16 x 16 neighbourhood of a keypoint. Each bin contains the sum of the gradient magnitudes of corresponding orientations, weighted by a Gaussian.
- Illustration shows 2 x 2 histograms for 8 x 8 neighbourhood, Gaussian indicated by circle.



Recognition Using SIFT Features

- Compute SIFT features on the input image
- Match these features to the SIFT feature database of an object model
- Each keypoint specifies 4 parameters: 2D location, scale, and dominant orientation.
- To increase recognition robustness: Hough transform to identify clusters of matches that vote for the same object pose.
- Each keypoint votes for the set of object poses that are consistent with the keypoint's location, scale, and orientation.
- Locations in the Hough accumulator that accumulate at least 3 votes are selected as candidate object/pose matches.
- A verfication step matches the training image for the hypothesized object/pose to the image using a least-squares fit to the hypothesized location, scale, and orientation of the object.

Experiment 1 I

Training images

Test image

Experiment 1 II

Test image with overlaid results.

Parallelograms show locations of recognized objects.

Small squares show keypoints used for recognition.

Experiment 2 I

Complex test image, 640 x 315 pixels

Experiment 2 II

Training images taken from independent viewpoints

Experiment 2 III

Results

SIFT Features Summary

- SIFT features are reasonably invariant to rotation, scaling, and illumination changes.
- They can be used for matching and object recognition (among other things).
- Robust to occlusion: as long as we can see at least 3 features from the object we can compute the location and pose.
- Efficient on-line matching: recognition can be performed in close-to-real time (at least for small object databases).

Combined Object Categorization and Segmentation

Bastian Leibe, Ales Leonardis, and Bernt Schiele: Combined Object Categorization and Segmentation with an Implicit Shape Model

ECCV'04 Workshop on Statistical Learning in Computer Vision, Prague, May 2004.

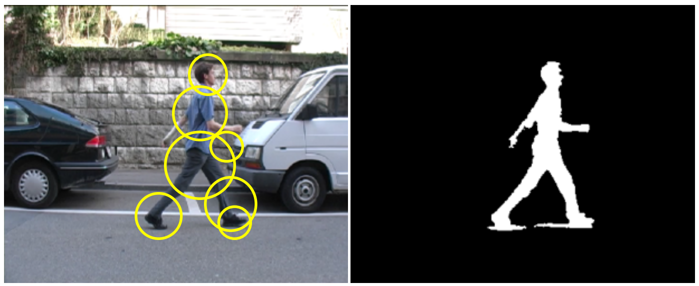
Define a shape model for an object class (or category) by

- a class-specific collection of local appearances (a "codebook"),
- a spatial probability distribution specifying where a codebook entry may be found on the object

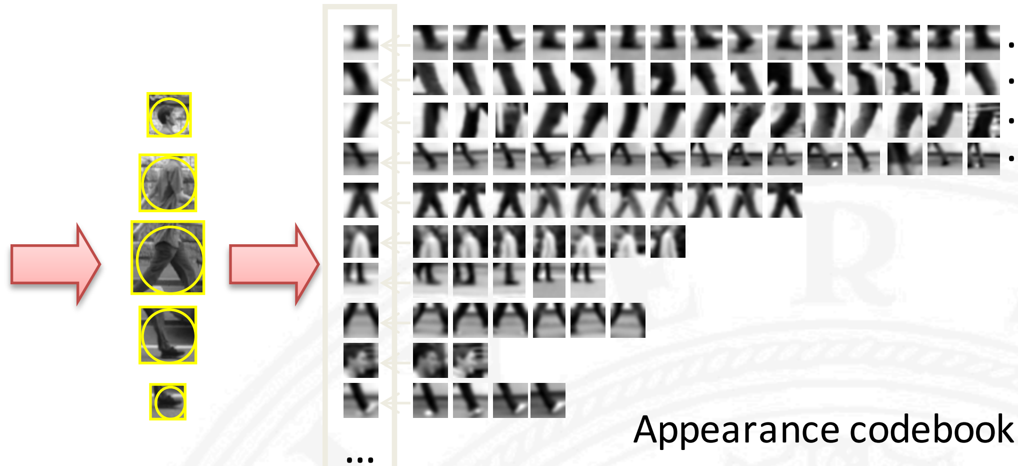
To recognize an object,

- extract image patches around interest points and and compare them with the codebook.
- Matching patches cast probabilistic votes leading to object hypotheses.
- Each pixel of an object hypothesis is classified as object or background based on the contributing patches.

Implicit Shape Model - Representation

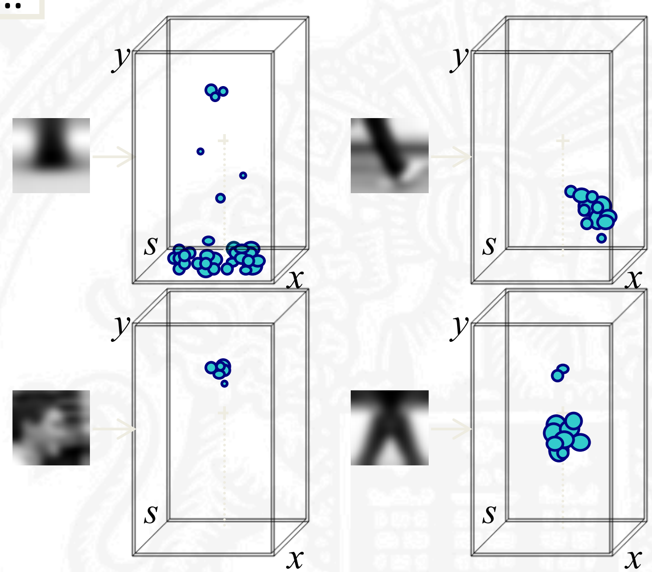


105 training images
(+ motion segmentation)



Appearance codebook

- Learn appearance codebook
  - Extract 25x25 patches at interest points
  - Agglomerative clustering ⇒ codebook
- Learn spatial distributions
  - Match codebook to training images
  - Record matching positions on object



Spatial occurrence distributions

Harris Corner Detector I

Large differences between a pixel and its surroundings:

$$S(x,y)=\sum_u\sum_v w(u,v)\,(I(u+x),v+y)-I(u,v))^2$$

Averaging over a circular window with Gaussian weights *w(u, v)*.

First-order Taylor Series approximation:

$$I(u+x,\ v+y)\approx I(u,\ v)+I_x(u,\ v)\,x+I_y(u,\ v)\,y$$

$$\Rightarrow\quad S(x,y)\approx\sum_u\sum_v w(u,v)\,(I_x(u,v)x+I_y(u,v)y)^2=[x\quad y]A\begin{bmatrix}x\\y\end{bmatrix}$$

with $A=\sum_u\sum_v w(u,v)\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix}$ **"Structure Tensor"**

Harris Corner Detector II

- Eigenvalues λ_1 and λ_2 of A indicate cornerness:
  - $\lambda_1 \approx 0$ *and* $\lambda_2 \approx 0$ basically flat greyvalues
  - $\lambda_1 \approx 0$ *and* $\lambda_2 \gg 0$ edge
  - $\lambda_1 \gg 0$ *and* $\lambda_2 \gg 0$ corner

- Instead of computing eigenvalues explicitly:
  - $M_c = \lambda_1\lambda_2 - \kappa(\lambda_1+\lambda_2)^2 = \boxed{det(A) - \kappa\, trace^2(A)}$
    measure of cornerness
  - $\kappa = 0.04 \ldots 0.15$ sensitivity parameter, must be tuned empirically

Agglomerative Clustering

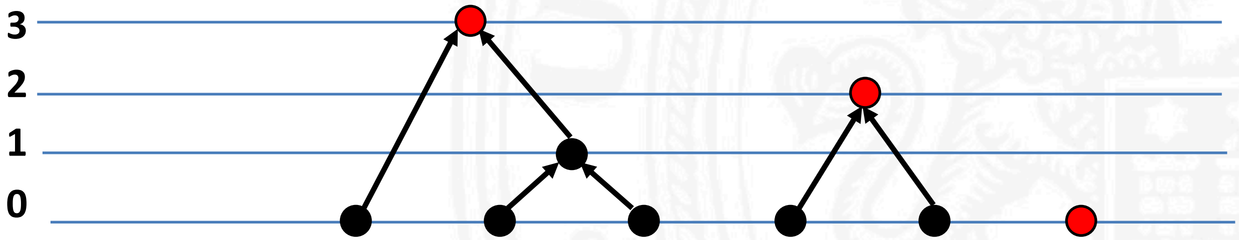
- Start with separate clusters for each single item
- Merge most similar clusters as long as average similarity within cluster stays above threshold

$$s(C) = \frac{\sum_{p \in C} NGC(p)}{|C|}$$

similarity s within cluster C

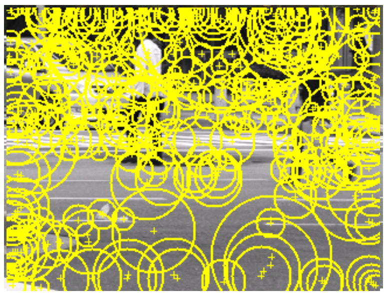
$$NGC(p,q) = \frac{\sum_i (p_i - \overline{p})(q_i - \overline{q})}{\sqrt{\sum_i (p_i - \overline{p})^2 \sum_i (q_i - q)^2}}$$

Normalized Greyscale Correlation



Implicit Shape Model - Recognition I

Interest Points



Matched Codebook Entries

Probabilistic Voting

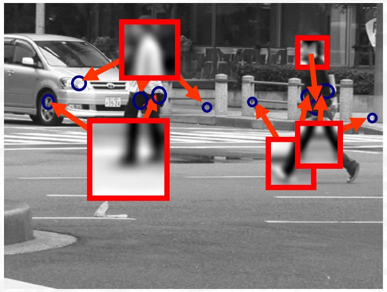


Image Patch (evidence e)

Interpretations I (Codebook match)

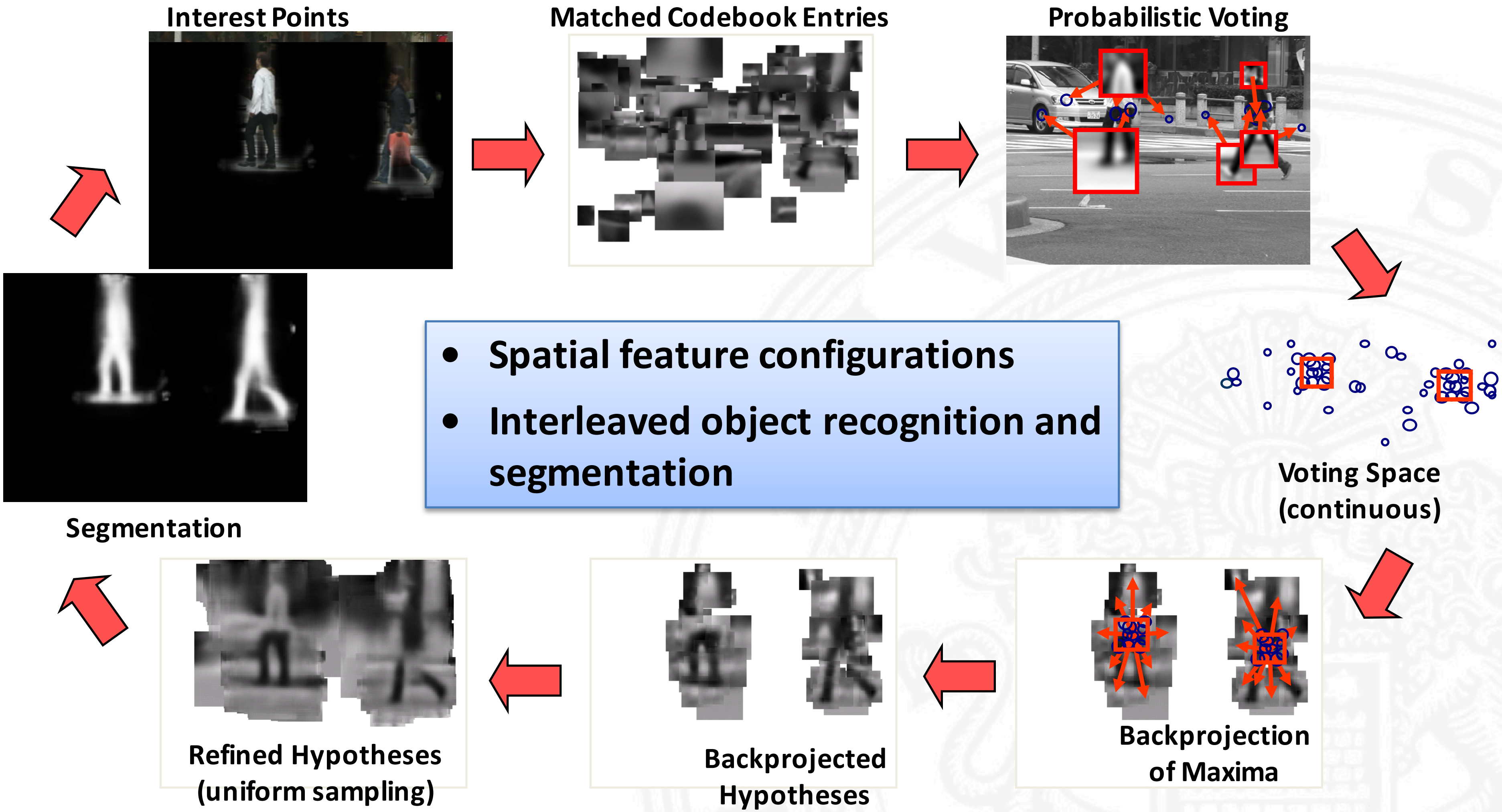
Objects o Positions x

e $\quad p(I_j|e) \quad$ I $\quad p(o_n, x|I_j) \quad$ o,x

$$p(o_n, x|I_j)p(I_j|e)$$

$$p(o_n, x|e) = \sum_j p(o_n, x|I_j)p(I_j|e)$$

Implicit Shape Model - Recognition II



- Spatial feature configurations
- Interleaved object recognition and segmentation

Car Detection

- Recognizes different kinds of cars
- Robust to clutter, occlusion, noise, low contrast

Cow Detection and Segmentation

- frame-by-frame detection
- no temporal continuity exploited