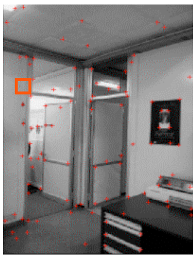
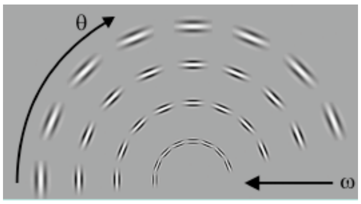
Patch-based Object Recognition

Basic Idea

- Determine interest points in image
- Determine local image properties around interest points
- Use local image properties for object classification



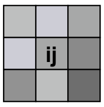
Example: Interest points determined by Haralick Operator



Example: Gabor-Filterbank for local image description

Interest Operators (1)

Moravec interest operator: $M(i,j) = \frac{1}{8}\sum_{m=i-1}^{i+1}\sum_{n=j-1}^{j+1}|g(m,n) - g(i,j)|$



Zuniga-Haralick operator:

- fit a cubic polynomial

  $f(i,j) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3$

  For a 5x5 neighbourhood the coefficients of the best-fitting polynomial can be directly determined from the 25 greyvalues

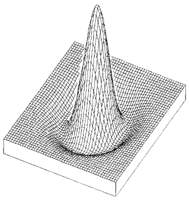
- compute interest value from polynomial coefficients

  $ZH(i,j) = \frac{-2(c_2^2c_6 - c_2c_3c_5 - c_3^2c_4)}{\left(c_2^2 + c_3^2\right)^{\frac{3}{2}}}$ measure of "cornerness" of the polynomial

Interest Operators (2)

Difference-of-Gaussians (DoG)

Locates edges at zero crossings of second derivative of smoothed image



"mexican-hat operator"

Harris interest operator:

Determine points with two strong principle curvatures

R = det(H) - k tr(H) = $\alpha\beta$ - k (α+β)

α and β are eigenvalues of Hessian matrix H and proportional to main curvatures

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

SIFT Features

SIFT = Scale Invariant Image Features

David G. Lowe: Distinctive Image Features from Scale-Invariant Keypoints
International Journal of Computer Vision, 2004

Computation Steps for SIFT Features

1. **Scale-space extrema detection:** The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2. **Keypoint localization:** At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

3. **Orientation assignment:** One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

4. **Keypoint descriptor:** The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

Scale Space

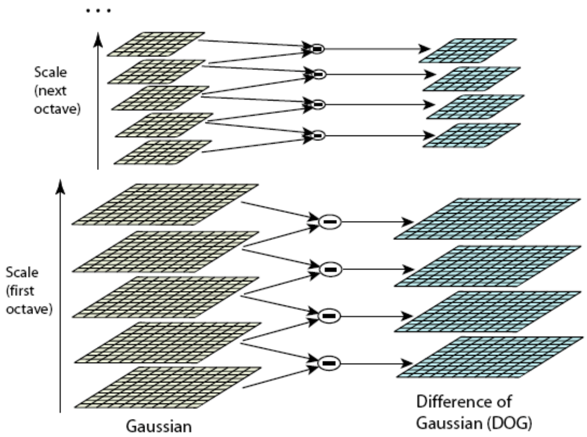
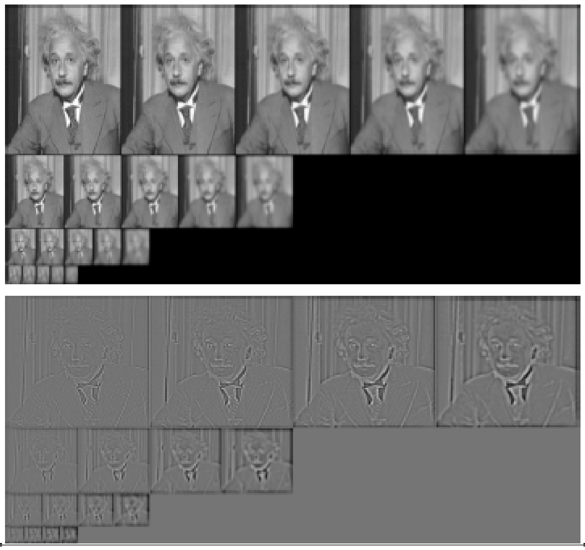


Figure 1: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated.

Example Image in Scale Space



Maxima Detection

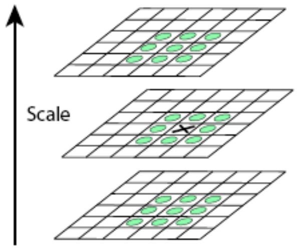


Figure 2: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

Keypoint Selection

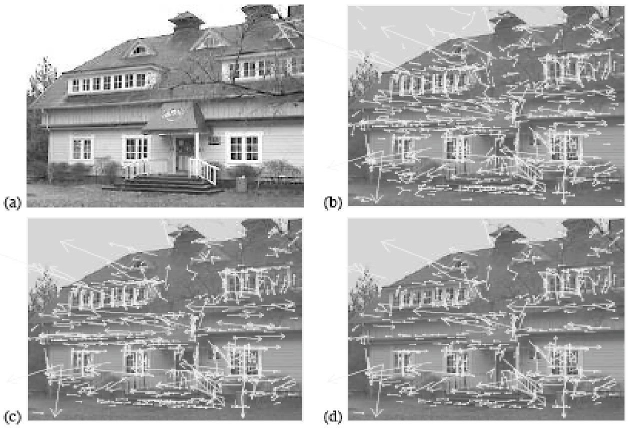


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

Eliminating Edge Responses

Compute Hessian at keypoint: $\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$

Eigenvalues α and β are proportional to principal curvatures.
Both principle curvatures must be significant for a keypoint to be stable.

Note that

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$
$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

Hence one can check the ratio r = α/β of the principle curvatures by evaluating

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$

Local Image Descriptor

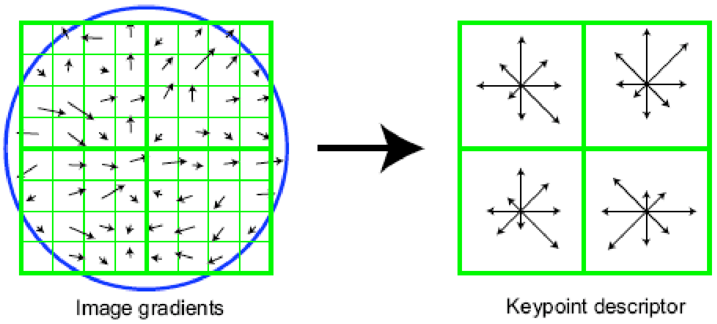
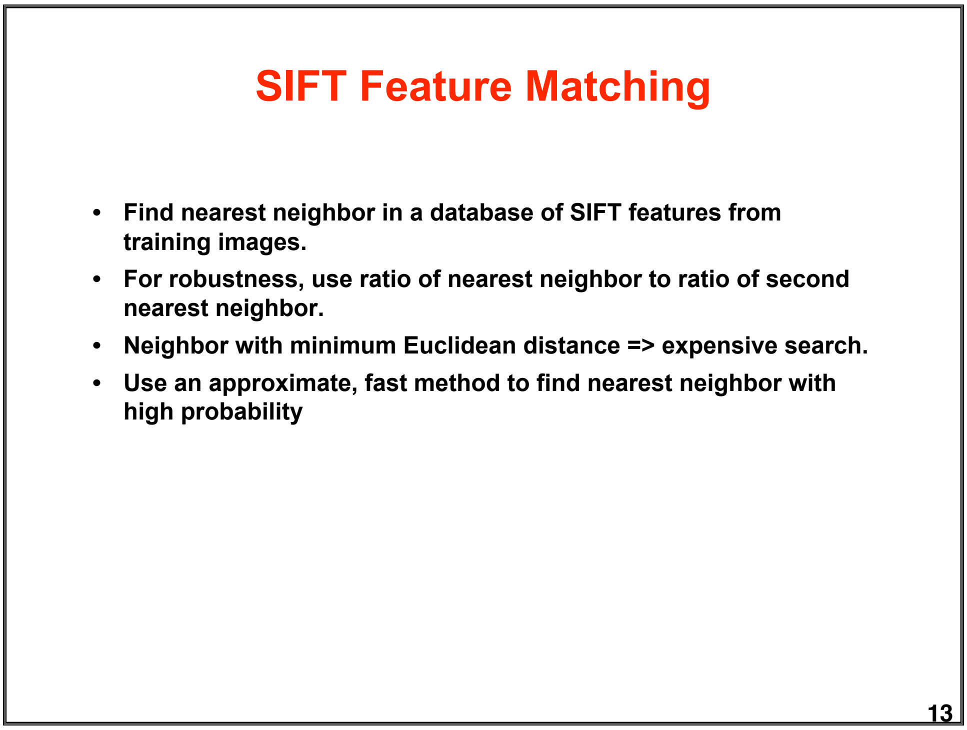
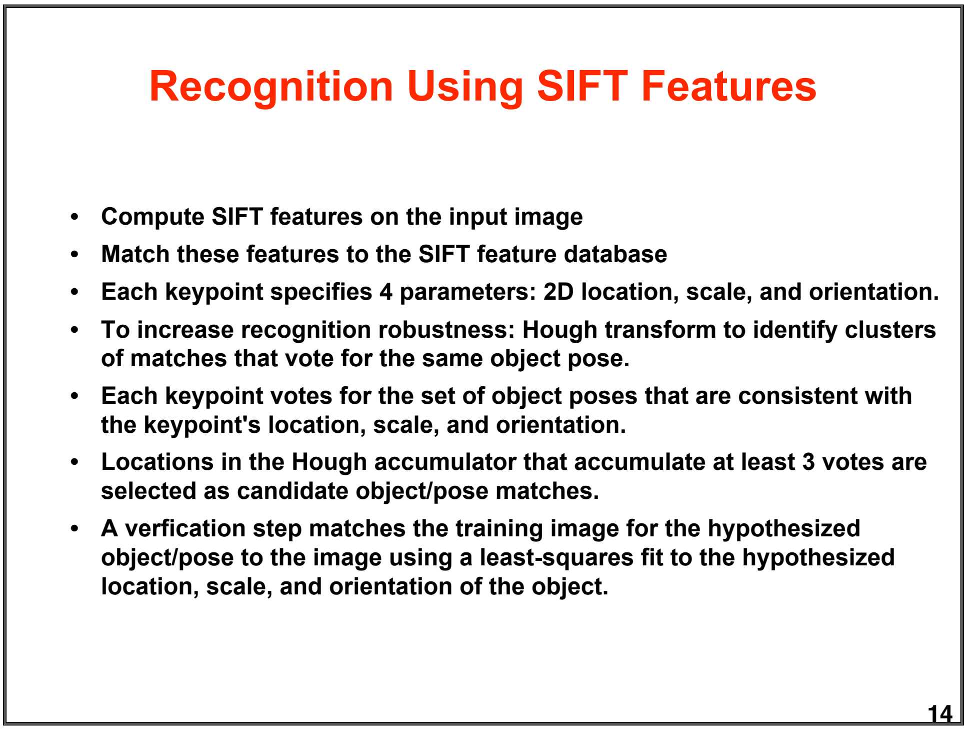


Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.



SIFT Feature Matching

- Find nearest neighbor in a database of SIFT features from training images.
- For robustness, use ratio of nearest neighbor to ratio of second nearest neighbor.
- Neighbor with minimum Euclidean distance => expensive search.
- Use an approximate, fast method to find nearest neighbor with high probability



Recognition Using SIFT Features

- Compute SIFT features on the input image
- Match these features to the SIFT feature database
- Each keypoint specifies 4 parameters: 2D location, scale, and 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.

Experiments (1)

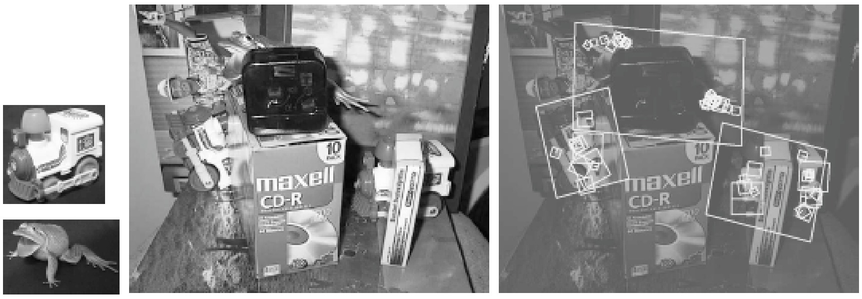


Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affine transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

Experiments (2)

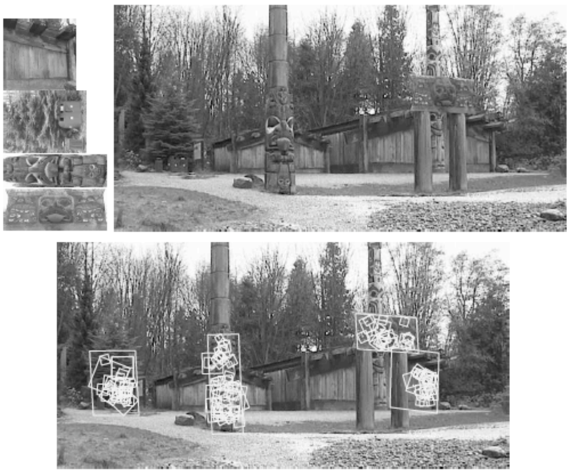
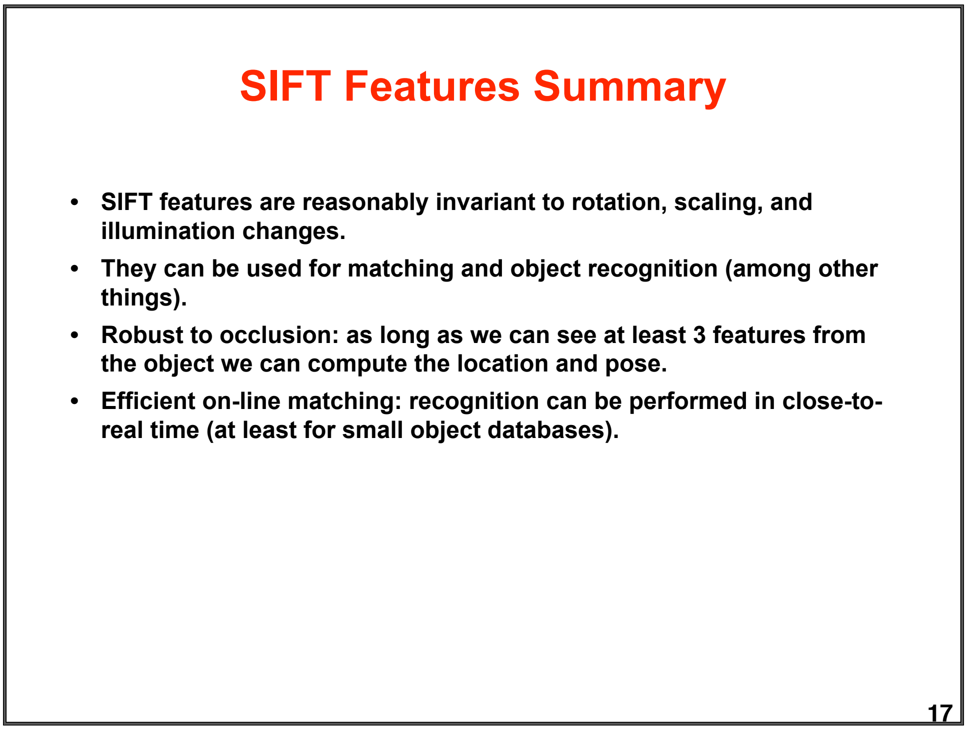
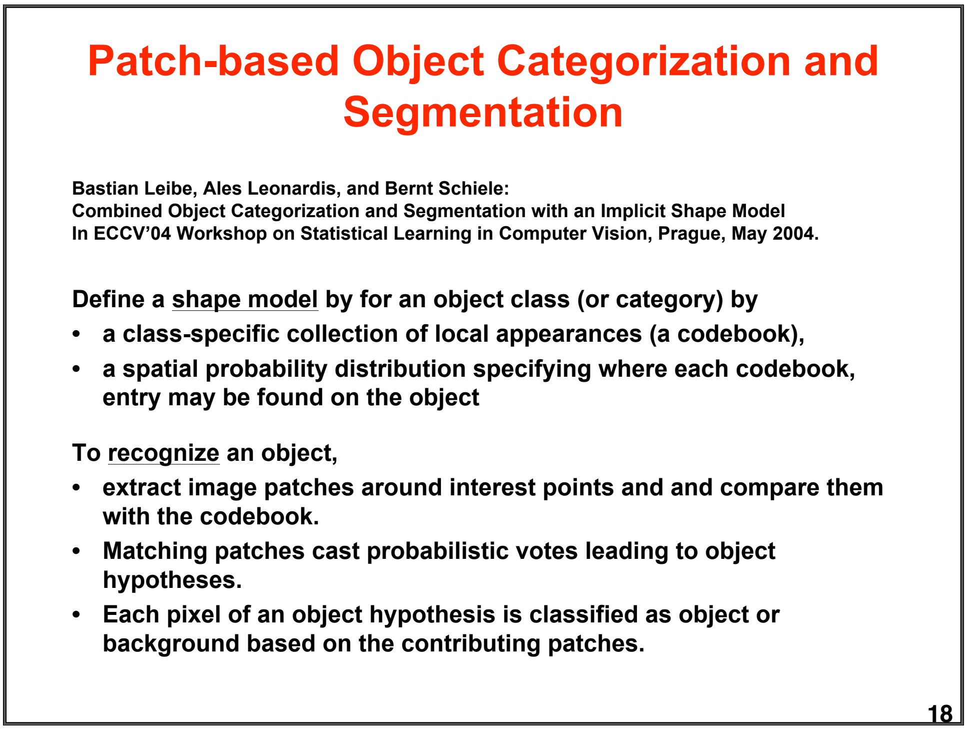


Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affine transform used for recognition.



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).



Patch-based Object Categorization and Segmentation

Bastian Leibe, Ales Leonardis, and Bernt Schiele:
Combined Object Categorization and Segmentation with an Implicit Shape Model
In ECCV'04 Workshop on Statistical Learning in Computer Vision, Prague, May 2004.

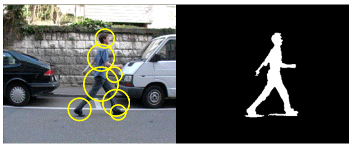
Define a shape model by for an object class (or category) by

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

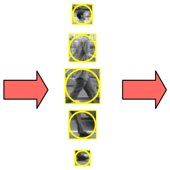
To recognize an object,

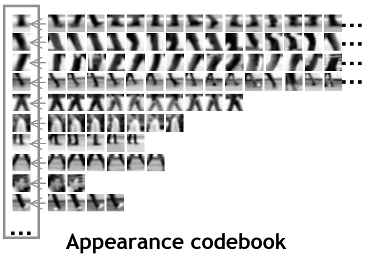
- extract image patches around interest points and and compare them with the codebook.
- Matching patches cast probabilistic votes leading to object hypotheses.
- Each pixel of an object hypothesis is classified as object or background based on the contributing patches.

Implicit Shape Model - Representation



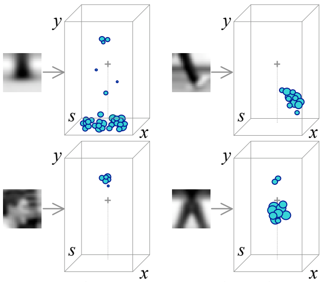
105 training images
(+ motion segmentation)





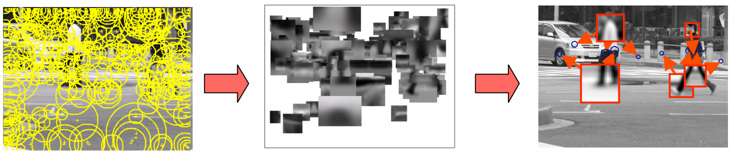
Appearance codebook

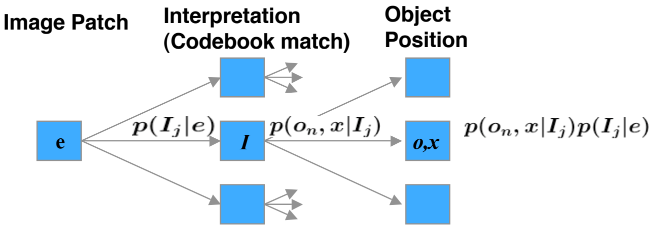
- Learn appearance codebook
  - Extract patches at DoG interest points
  - Agglomerative clustering ⇒ codebook
- Learn spatial distributions
  - Match codebook to training images
  - Record matching positions on object



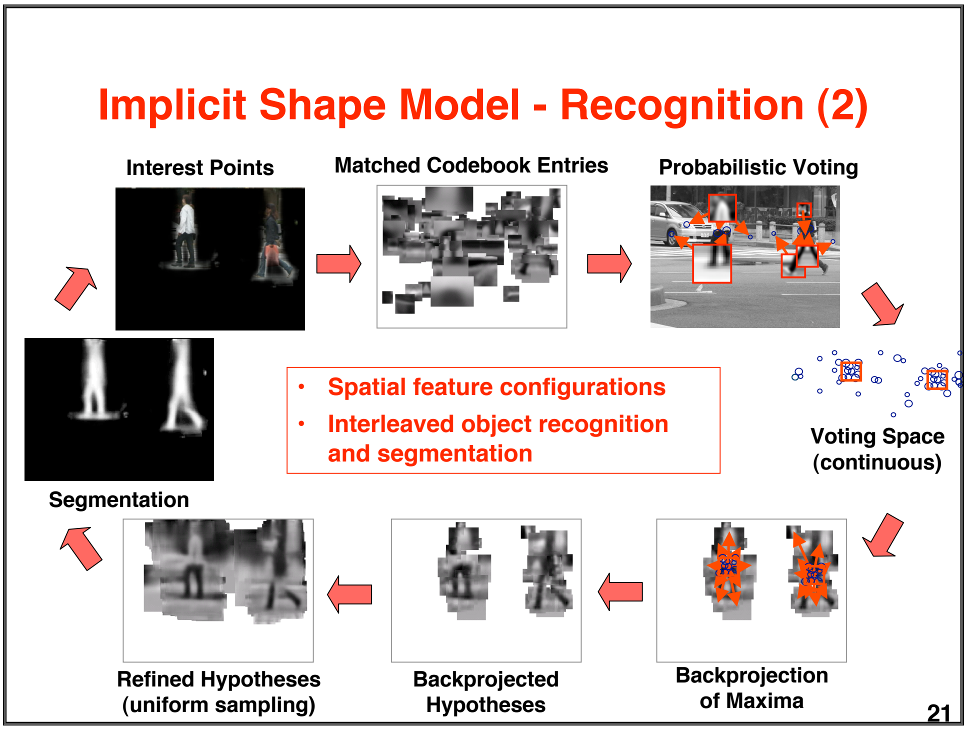
Spatial occurrence distributions

Implicit Shape Model - Recognition (1)





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



Implicit Shape Model - Recognition (2)

Interest Points

Matched Codebook Entries

Probabilistic Voting

Voting Space (continuous)

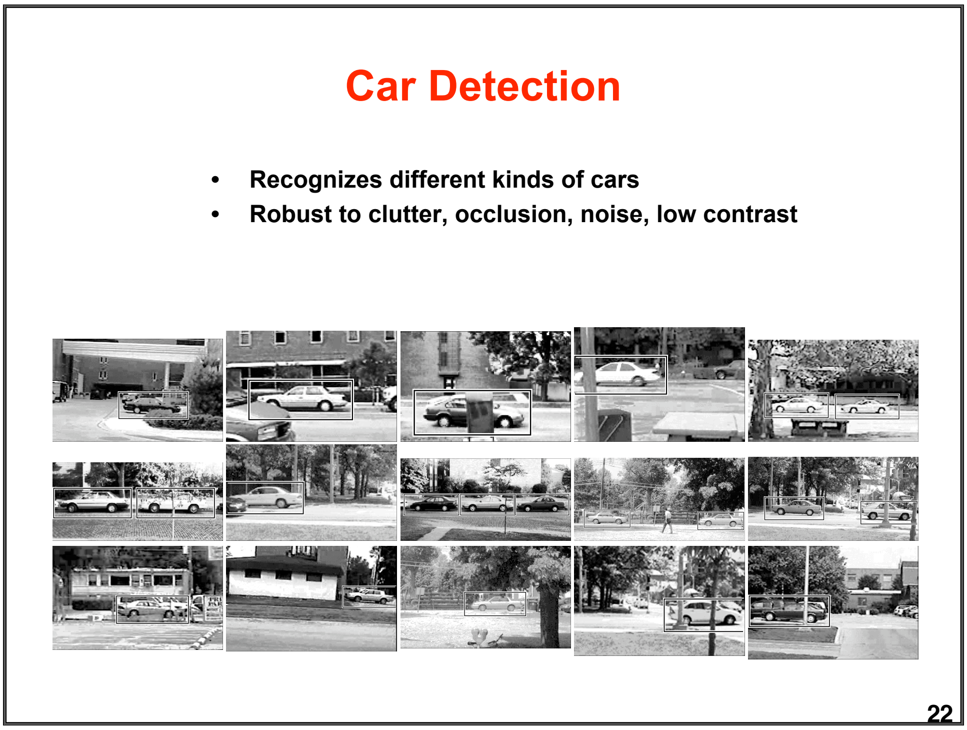
Backprojection of Maxima

Backprojected Hypotheses

Refined Hypotheses (uniform sampling)

Segmentation

- Spatial feature configurations
- Interleaved object recognition and segmentation



Car Detection

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

Cow Detection

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