Applied Computer Vision

David Vernon
Carnegie Mellon University Africa

vernon@cmu.edu
www.vernon.eu

Lecture 17

Object Recognition

Generalized Hough transform
Extension to code-word features

The Generalised Hough Transform

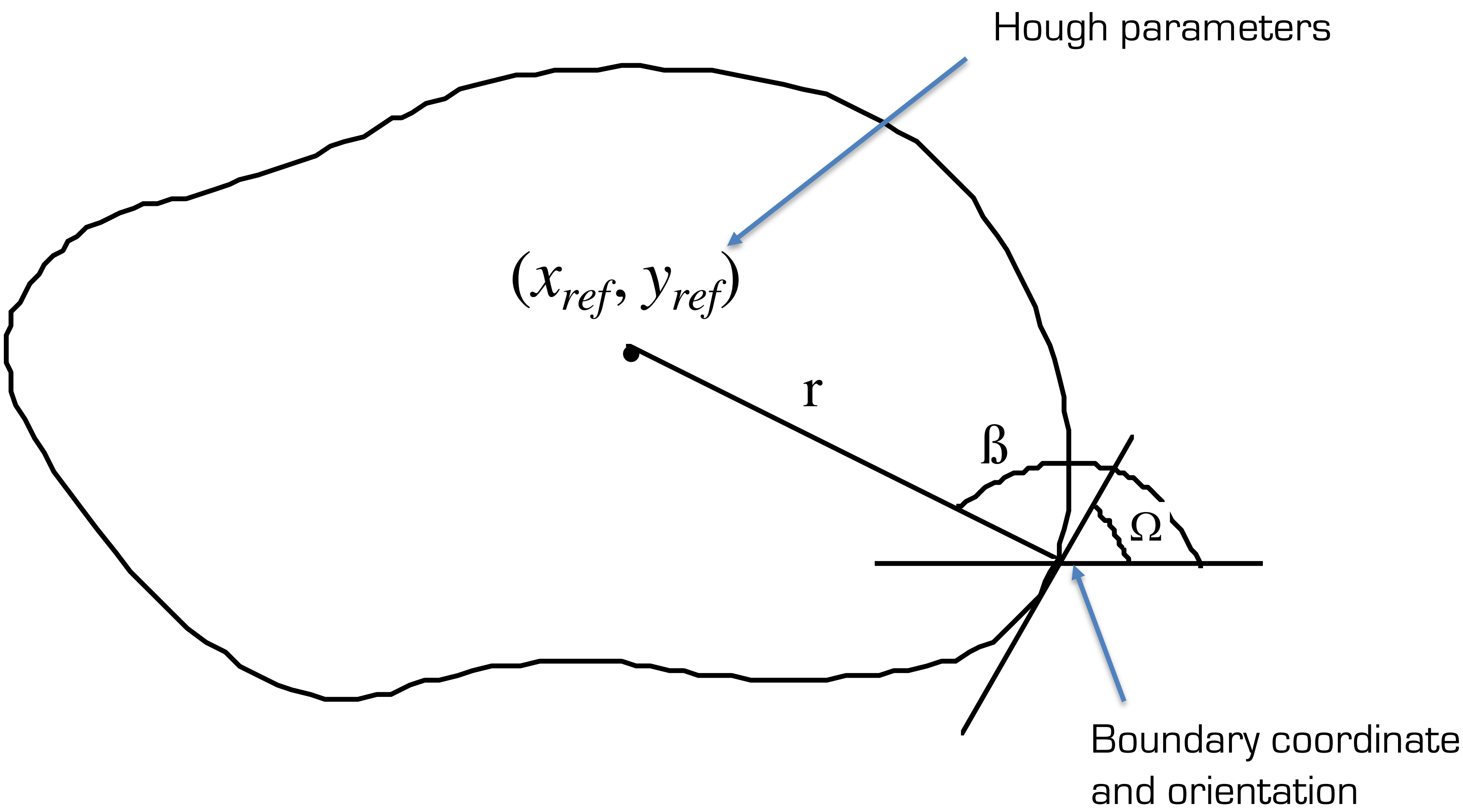
- In the previous formulation of the classical Hough Transform, we used the parametric equation of the shape to map from image space to transform space

- If the shape we wish to detect does not have a simple analytic equation describing its boundary, we can still use a generalised form of the Hough transform

  – we use a look-up table to define the mapping between the feature point (e.g. boundary edge) and the Hough parameters (e.g. coordinates of the centre of the shape)

  – the look-up table values must be computed during a training phase using a prototype shape

The Generalised Hough Transform



The Generalised Hough Transform

Training Phase

- Given the shape and orientation of the required object, select an arbitrary reference point (x_{ref}, y_{ref})
- Define the shape in terms of the distance and angle β_i of a line from the boundary point to this reference point
- For all points of the boundary, we draw a line to the reference point
- We then compute the orientation of the boundary, Ω_i (possibly using the gradient direction)

The Generalised Hough Transform

Training Phase

- Add an entry in the look-up table with
  - distance
  - direction

  from the boundary point to the reference point

- at look-up table location given by the boundary orientation Ω_i

The Generalised Hough Transform

- Since it is likely that there will be more than one occurrence of a particular orientation Ω_i as we travel around the boundary, we have to make provision for more than one pair of distance and angle values (i.e. multiple entries in the LUT)
- This look-up table is called an **R-Table**
- The Hough transform space is now defined in terms of the possible positions of the shape in the image, *i.e.*,
  the possible values of x_{ref} and y_{ref}
  (instead of r and ϕ in the case of the Hough transform for line detection)

The Generalised Hough Transform

- To perform the transform on an image we compute the point (x_{ref}, y_{ref}) from the co-ordinates of the boundary point, the distance r and the angle β

$$x_{ref} = x + r \cos \beta$$
$$y_{ref} = y + r \sin \beta$$

The Generalised Hough Transform

- What values of r and β do we use?
- These are given by the R-Table
  - computing the boundary orientation Ω at that point
  - using it as an index into the R-table
  - reading off all the (r, β) pairs
- The accumulator array cell (x_{ref}, y_{ref}) is then incremented
- We reiterate this process for all edge points in the image
- The location of the shape is given by local maxima in the accumulator array

The Generalised Hough Transform

- Problem: we have assumed that we know the orientation of the shape

- If this is not the case, we have to extend the accumulator by incorporating an extra parameter ϕ to take change in orientation into consideration

- Thus, we now have a 3D accumulator indexed by (x_{ref}, y_{ref}, ϕ) and we compute

$$x_{ref} = x + r\cos(\beta + \phi)$$
$$y_{ref} = y + r\sin(\beta + \phi)$$

Generalised Hough Transform

```
/* Pseudo-code for Generalized Hough Transform */

Train the Shape by building the R-Table
For all points on the boundary
   Compute orientation Ω (gradient direction+90°)

   Compute r and β

   Add an (r, β) entry into the R-table
   at a location indexed by Ω
```

Generalised Hough Transform

Quantise the Hough transform

identify maximum and minimum values of x_{ref}, y_{ref} and ϕ

identify the total number of x_{ref}, y_{ref} and ϕ values.

Generate an accumulator array $A(x_{ref}, y_{ref}, \phi)$

set all values to 0

Generalised Hough Transform

For all edge points (x_i, y_i) **in the image**

Do

Compute the orientation Ω
(gradient diction + 90°)

Compute possible reference points x_{ref}, y_{ref}

For each table entry, indexed by Ω

For each possible shape orientation ϕ

Compute $x_{ref} = x_i + r \cos (\beta + \phi)$

$y_{ref} = y_i + r \cos (\beta + \phi)$

Increment $A(x_{ref}, y_{ref}, \phi)$

Generalised Hough Transform

For all cells in the accumulator array

Do

Search for maximum values

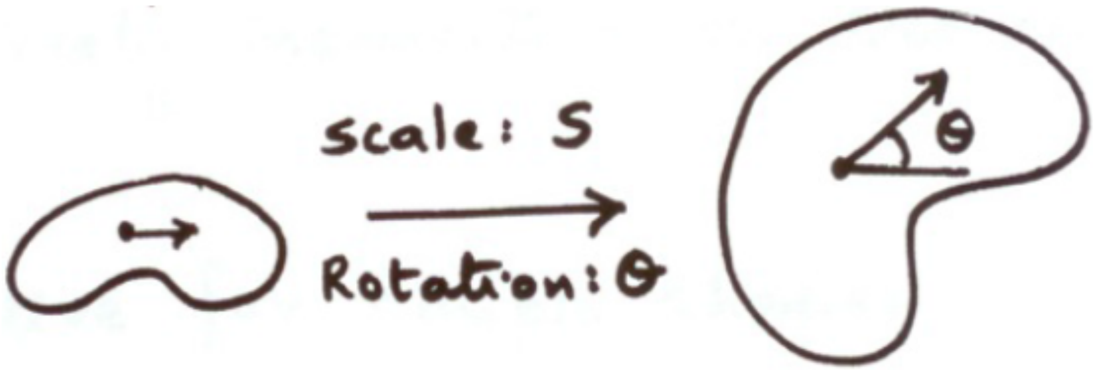
the co-ordinates x_{ref}, y_{ref} and ϕ give the position and orientation of the shape in the image

Generalised Hough Transform

- Detecting arbitrary shapes
- Complete specification of the exact shape of the target object is required
- Model of the shape is stored in the R-table
- Information that can be extracted are
    1. Location
    2. Size
    3. Orientation
    4. Number of occurrences of that particular shape

Generalised Hough Transform

To add scale use accumulator array $\mathrm{A}(x_{ref}, y_{ref}, s, \phi)$

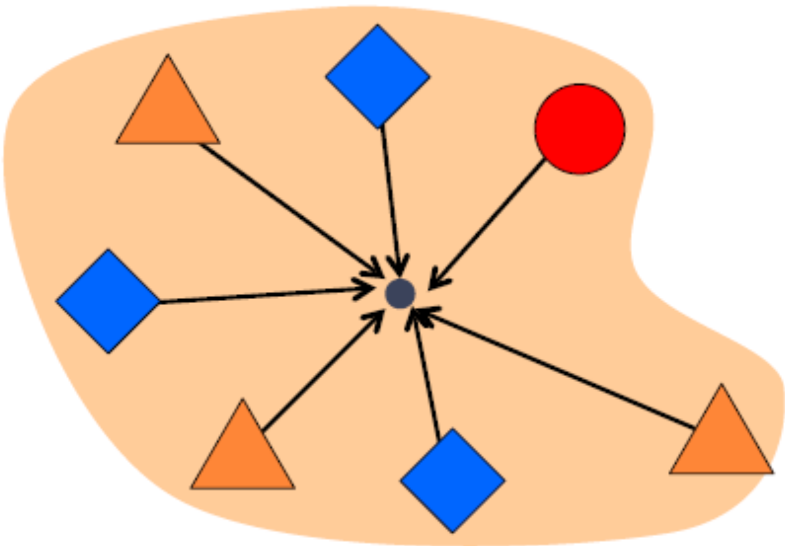


[Kris Kitani]

Generalised Hough Transform with Features

Instead of gradient points, features can be used

Two phases: train and test phase

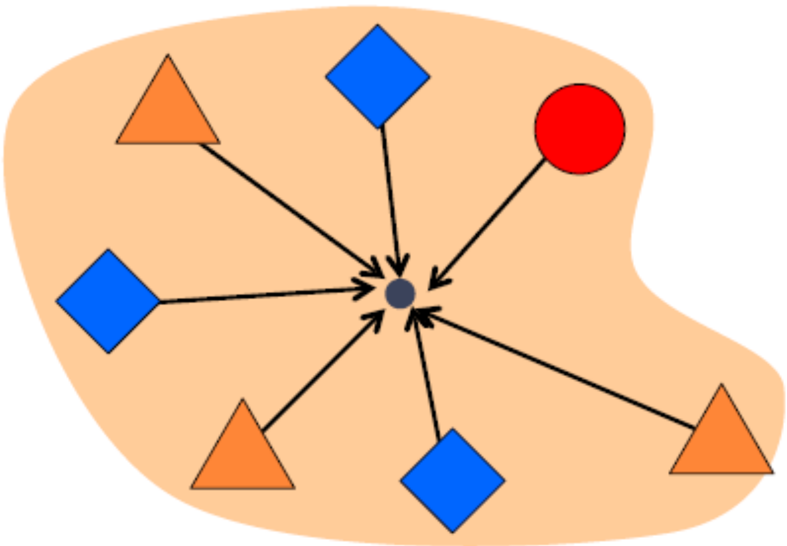


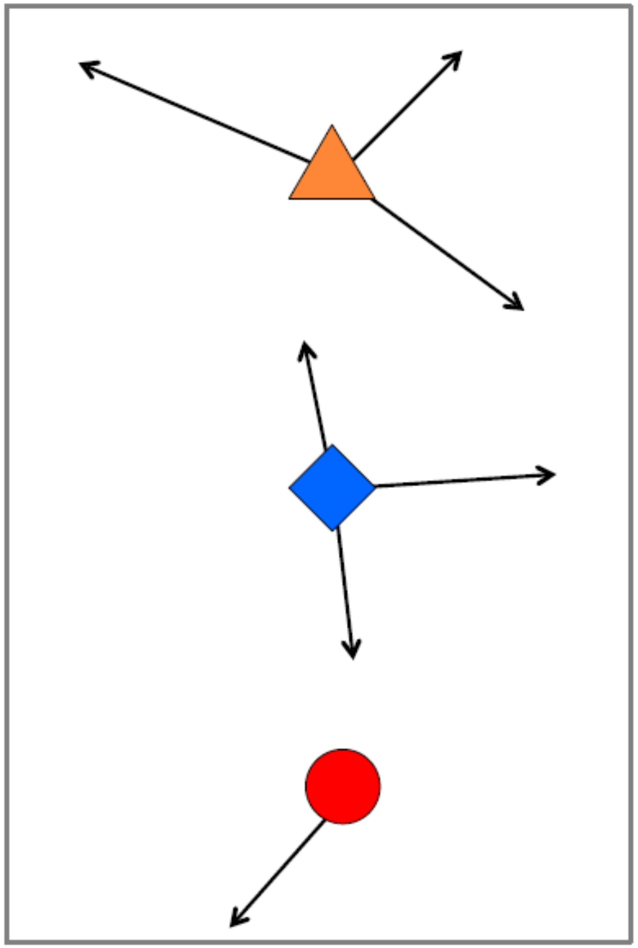
[Kris Kitani, CMU]

Generalised Hough Transform with Features

Training Phase

- Get features
- Store all displacements of features from centre



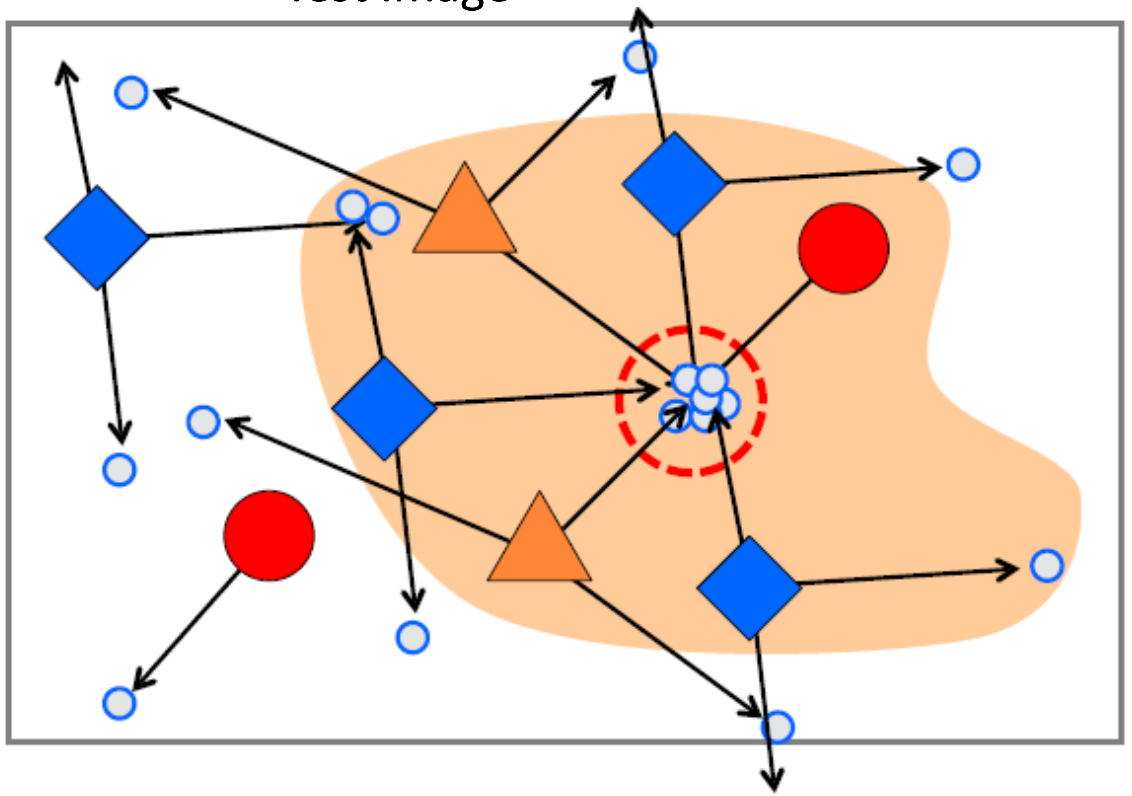


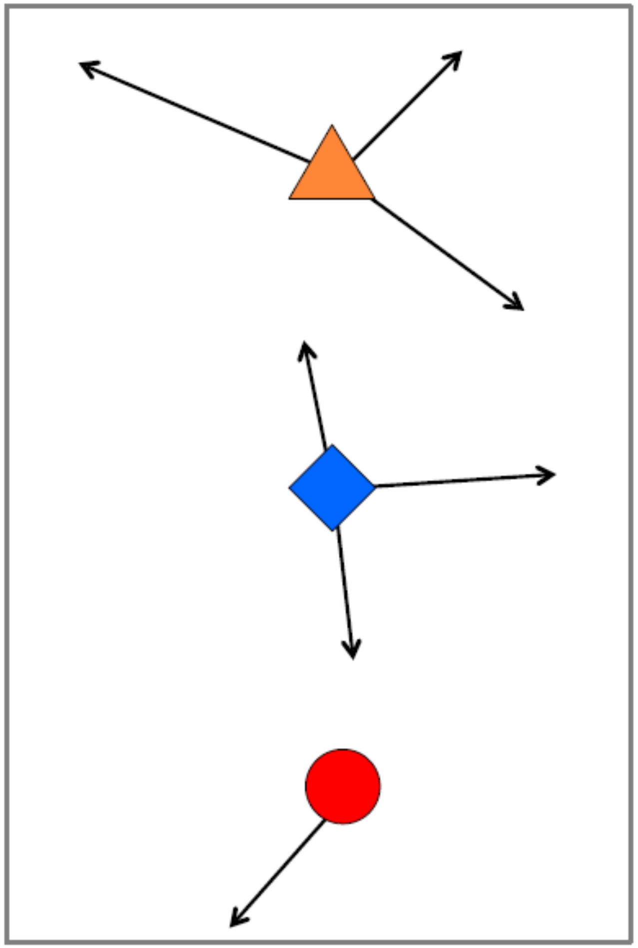
Generalised Hough Transform with Features

Test Phase

- Get features & look up displacements!
- Vote for center location

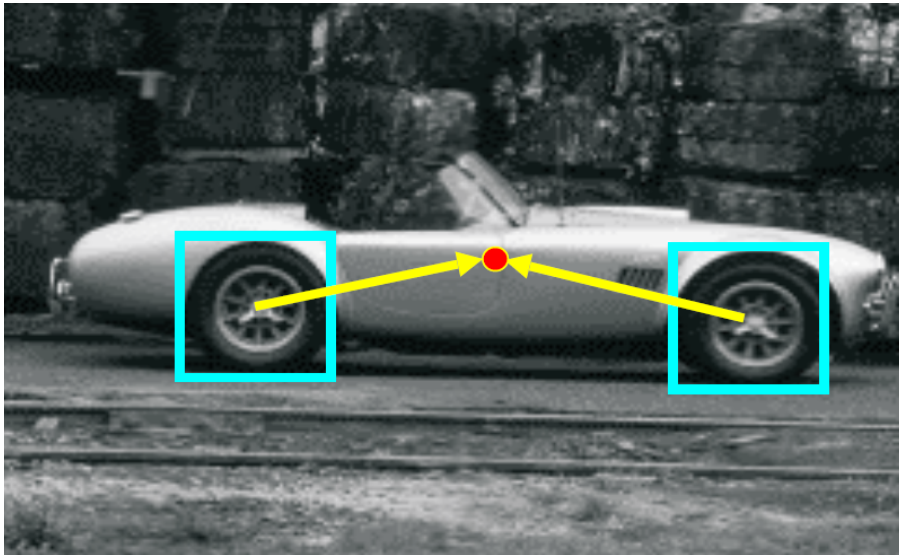
Test Image



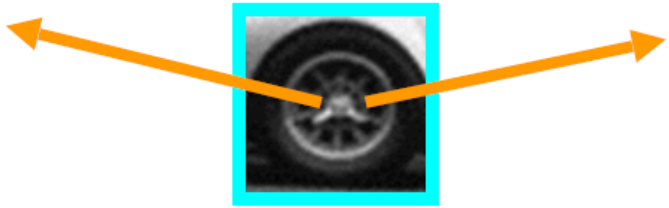


Generalised Hough Transform with Features

Index displacements by “visual codewords”



Training image



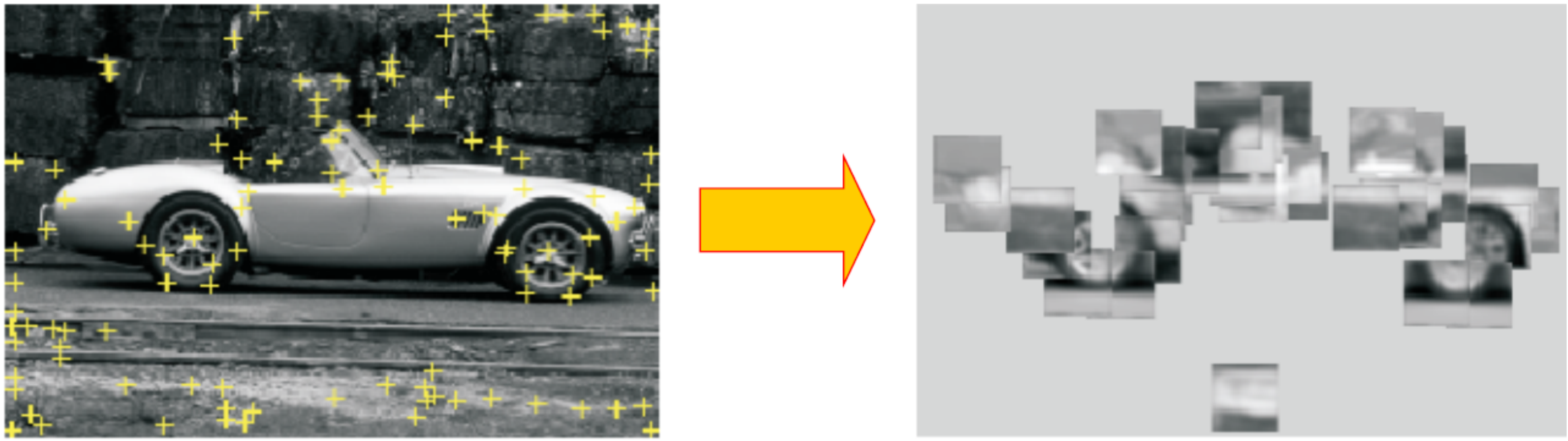
Visual codeword with displacement vectors

[B. Leibe, A. Leonardis, B. Schiele: Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004]

Generalised Hough Transform with Features

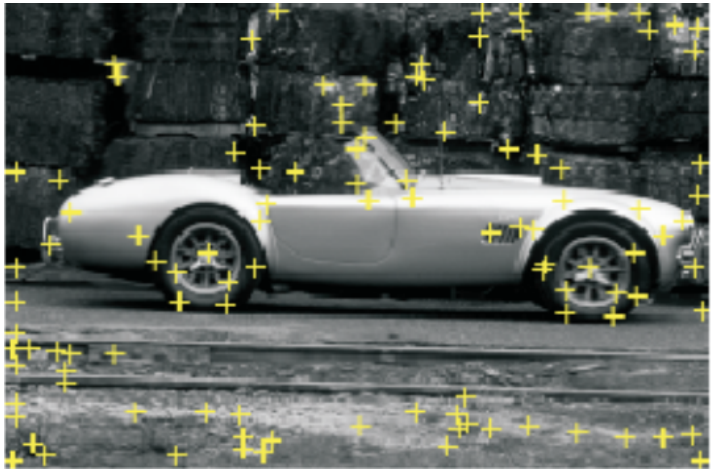
Generalised Hough Transform with Features

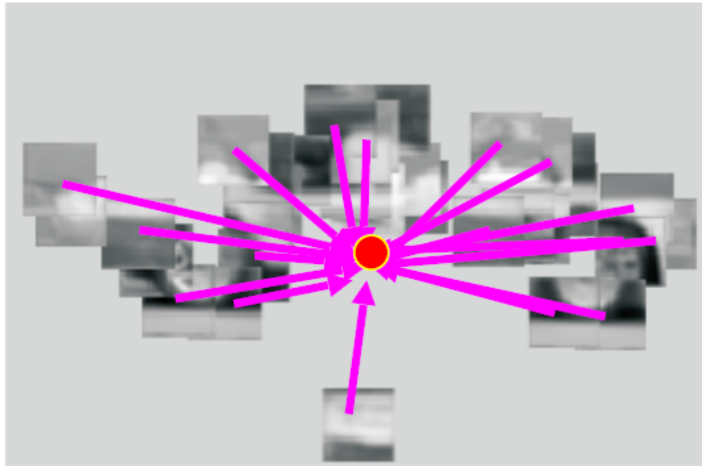
Training phase: get features



Generalised Hough Transform with Features

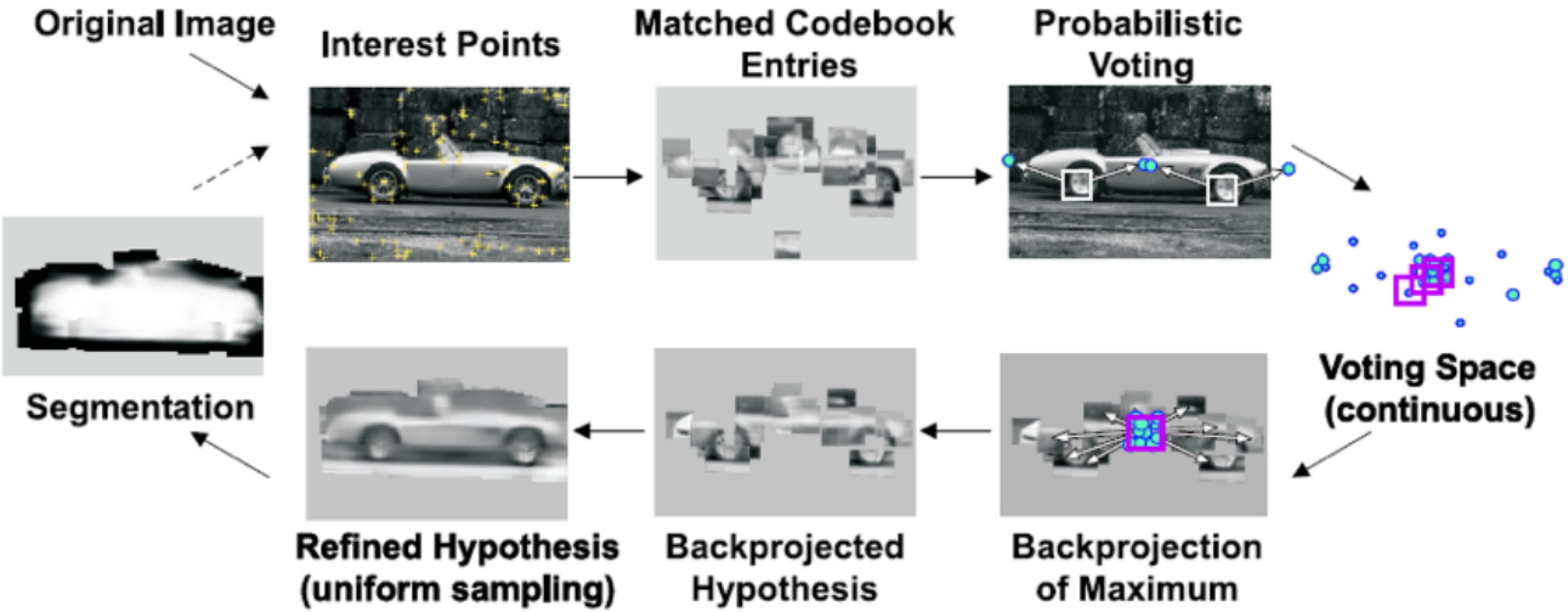
Training phase: store displacements





Generalised Hough Transform with Features

Test phase



Scale-Invariant Feature Transform - SIFT

Recognition of Planar Objects

- Planar surfaces recognised robustly with up to 60° rotation away from camera
- Affine transformation estimates the perspective projection
- 3 points sufficient to obtain full object pose

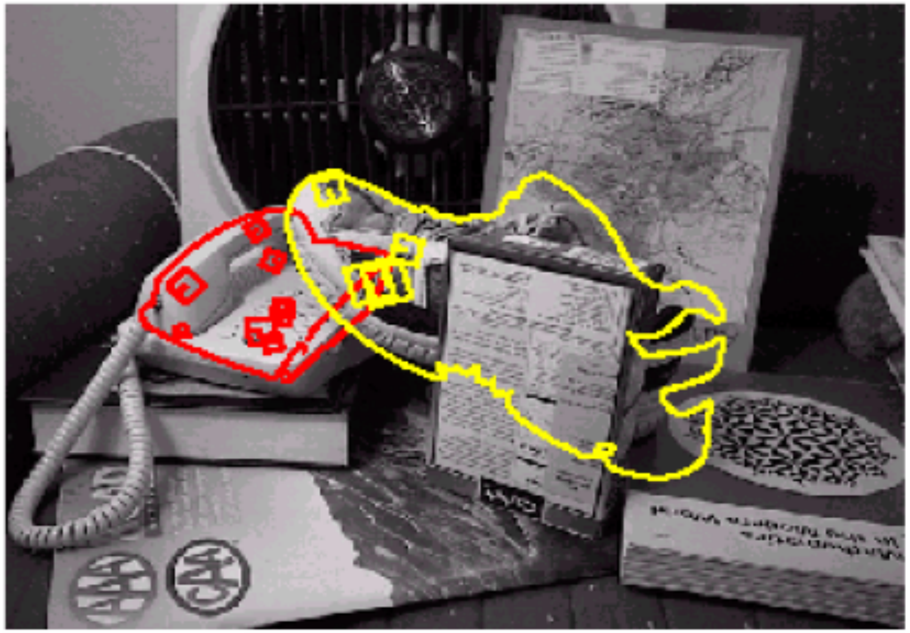
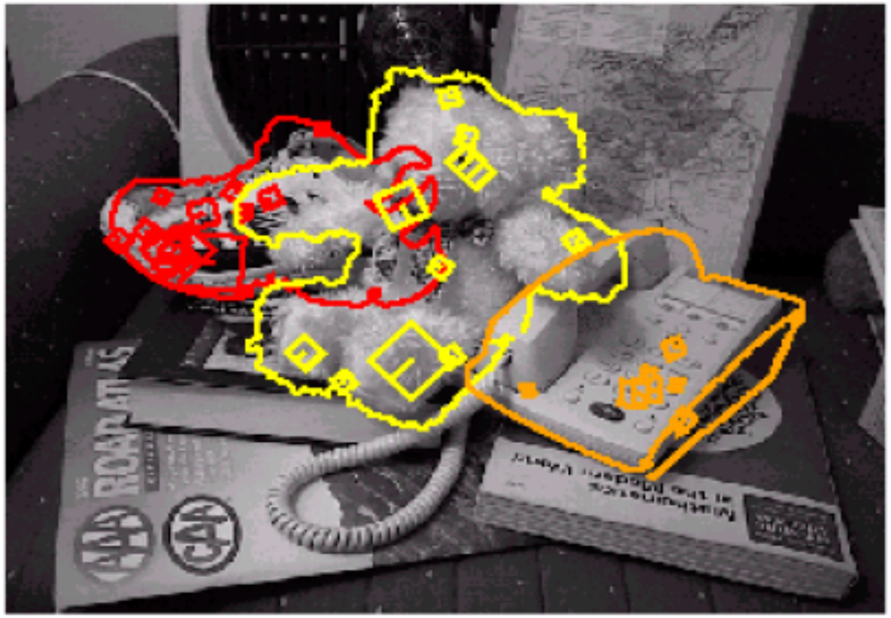
→ good if partially occluded

[Lowe 2004]

Credit: Markus Vincze, Technische Universität Wien

Scale-Invariant Feature Transform - SIFT

Recognition with occlusion

[Lowe 2004]

Credit: Markus Vincze, Technische Universität Wien

Scale-Invariant Feature Transform - SIFT

Object Recognition

First match each keypoint independently to the database of keypoints extracted from training images

How? ....

Scale-Invariant Feature Transform - SIFT

Matching Keypoints

Nearest neighbour matching

- Euclidean distance between keypoint vectors (128-D vector)
- Discard keypoints that have poor match
  - A global distance threshold does not work well
  - Instead, compare distance to closest neighbour to distance to 2nd closest neighbour (from a different object)
  - Reject matches where distance ratio threshold > 0.8 … eliminates
    - 90% false matches
    - 5% correct matches

Scale-Invariant Feature Transform - SIFT

Object Recognition

- Many of these initial matches will be incorrect due to ambiguous features or features that arise from background clutter

- Therefore, clusters of at least 3 features are first identified that agree on an object and its pose

  - These clusters have a much higher probability of being correct than individual feature matches
  - Each cluster is checked by performing a detailed geometric fit to the model

Scale-Invariant Feature Transform - SIFT

Object Recognition

- Identify cluster of at least 3 features matches using the Hough transform
- Each keypoint specifies 4 parameters
  - Location (2D)
  - Scale
  - Orientation
- Each matched keypoint in the database has a record of the keypoint's parameters relative to the training image in which it was found
- Create a Hough transform entry predicting the model location, orientation, and scale from the match hypothesis

Scale-Invariant Feature Transform - SIFT

Object Recognition

– Really a 6 degree-of-freedom pose estimation problem

– And there may be non-rigid deformation

– Consequently, there are large error bounds

– So, to compensate

Use broad bins

- 30° for orientation
- Factor of 2 for scale
- 0.25 times training image dimension for location

Scale-Invariant Feature Transform - SIFT

Object Recognition

- Each keypoint match votes for the 2 closest bins in each dimension
  - giving a total of 16 entries for each hypothesis
- Consider all bins in the Hough transform with at least 3 entries
- Each such cluster is then subject to geometric verification
  - A least-squares solution for the best affine projection parameters relating the training image to the test image is computed

Scale-Invariant Feature Transform - SIFT

Object Recognition

Credit: [Lowe 2004]

Reading

R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010.

Section 4.3 Lines

Section 4.3.2 Hough Transforms