Recognition with Hough Transform

Computer Vision

Carnegie Mellon University (Kris Kitani)

Do you have to use edge detectors to vote in Hough Space?

A. Train phase:

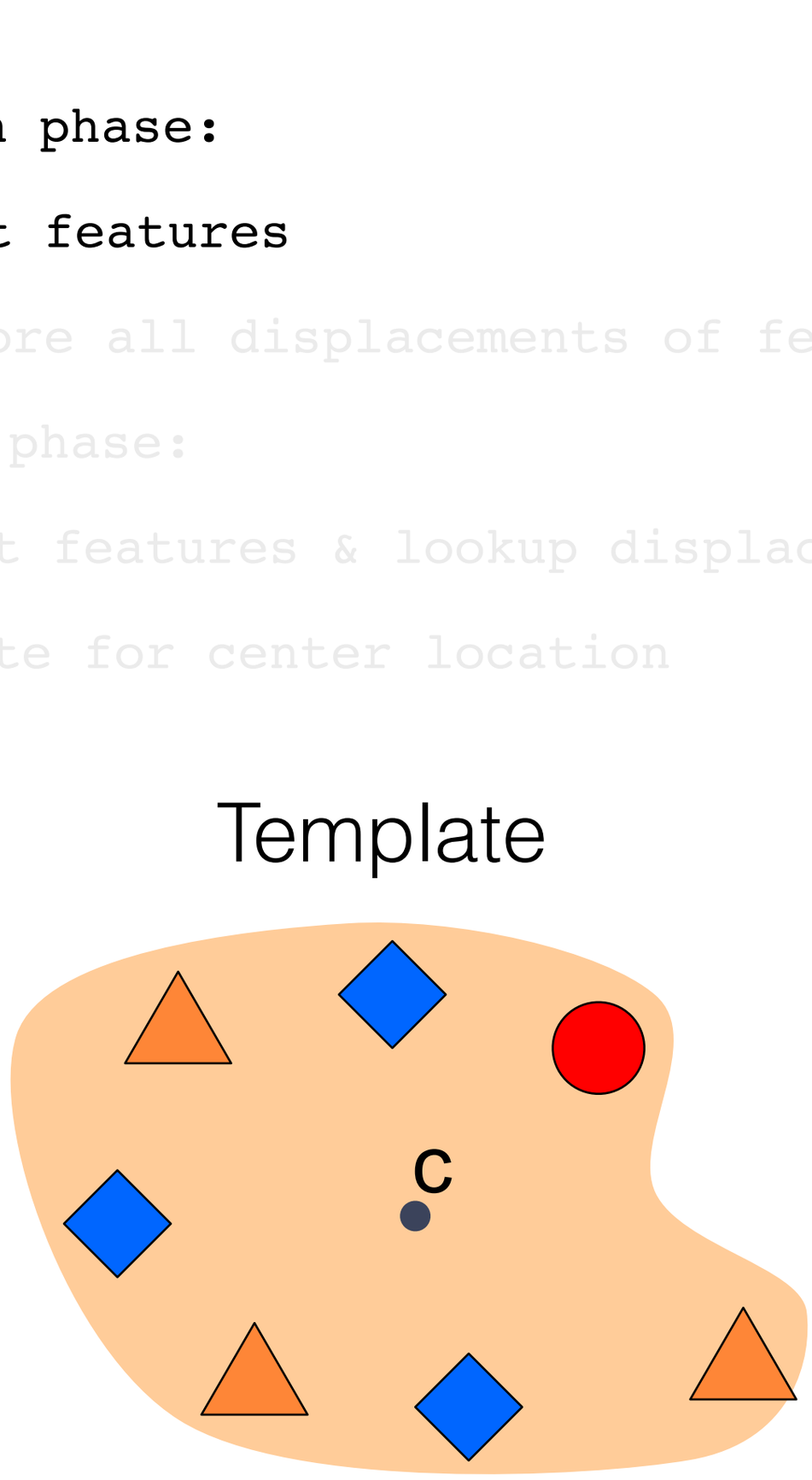
   1. Get features

   2. Store all displacements of feature from center

B. Test phase:

   1. Get features & lookup displacements

   2. Vote for center location



A. Train phase:

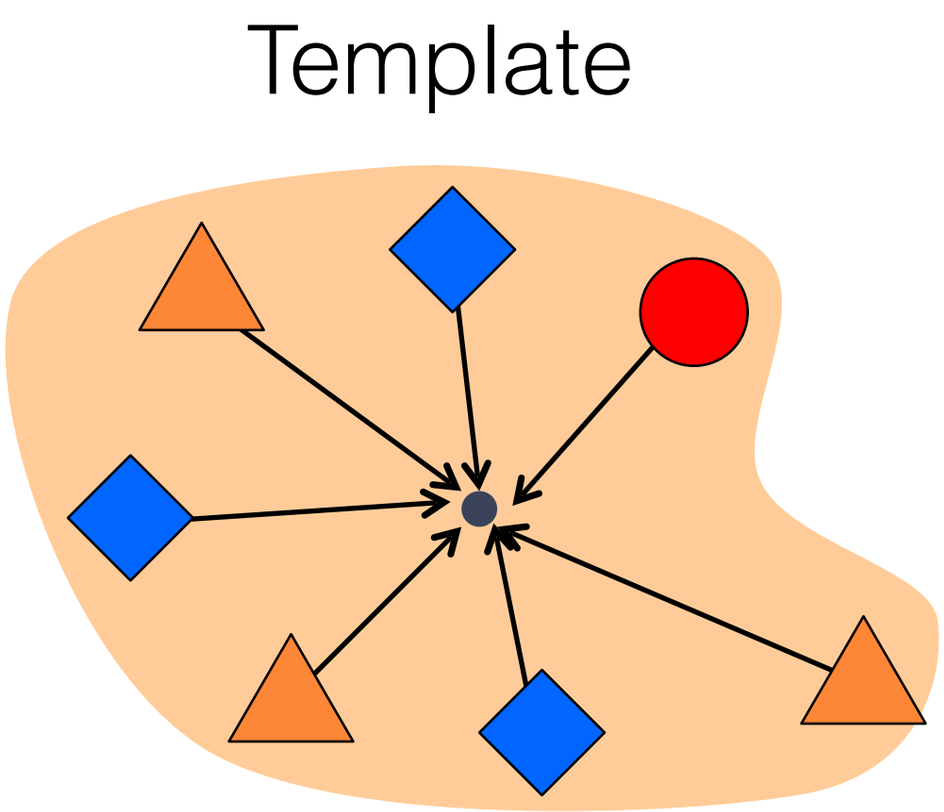
   1. Get features

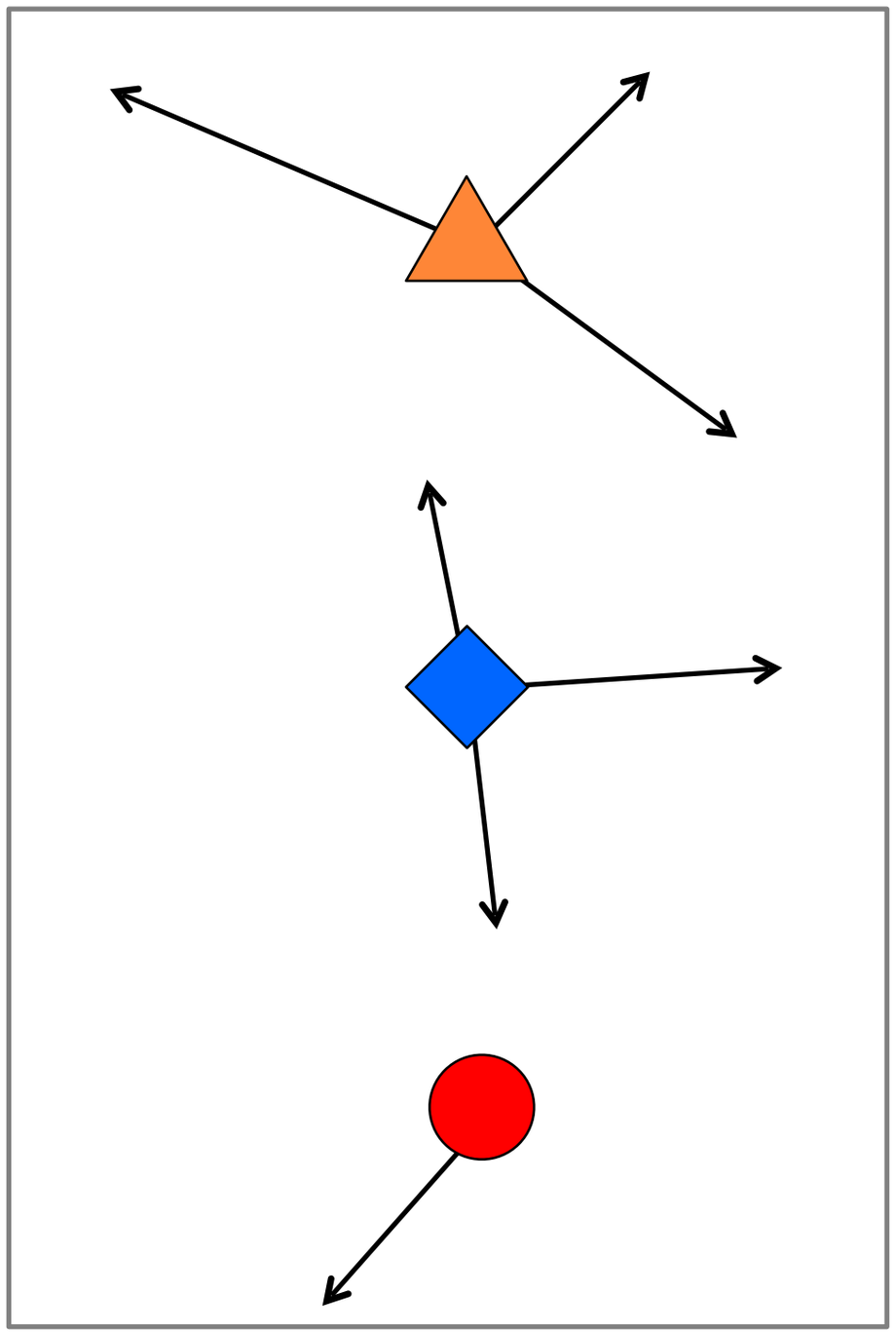
   2. Store all displacements of feature from center

B. Test phase:

   1. Get features & lookup displacements

   2. Vote for center location





A. Train phase:

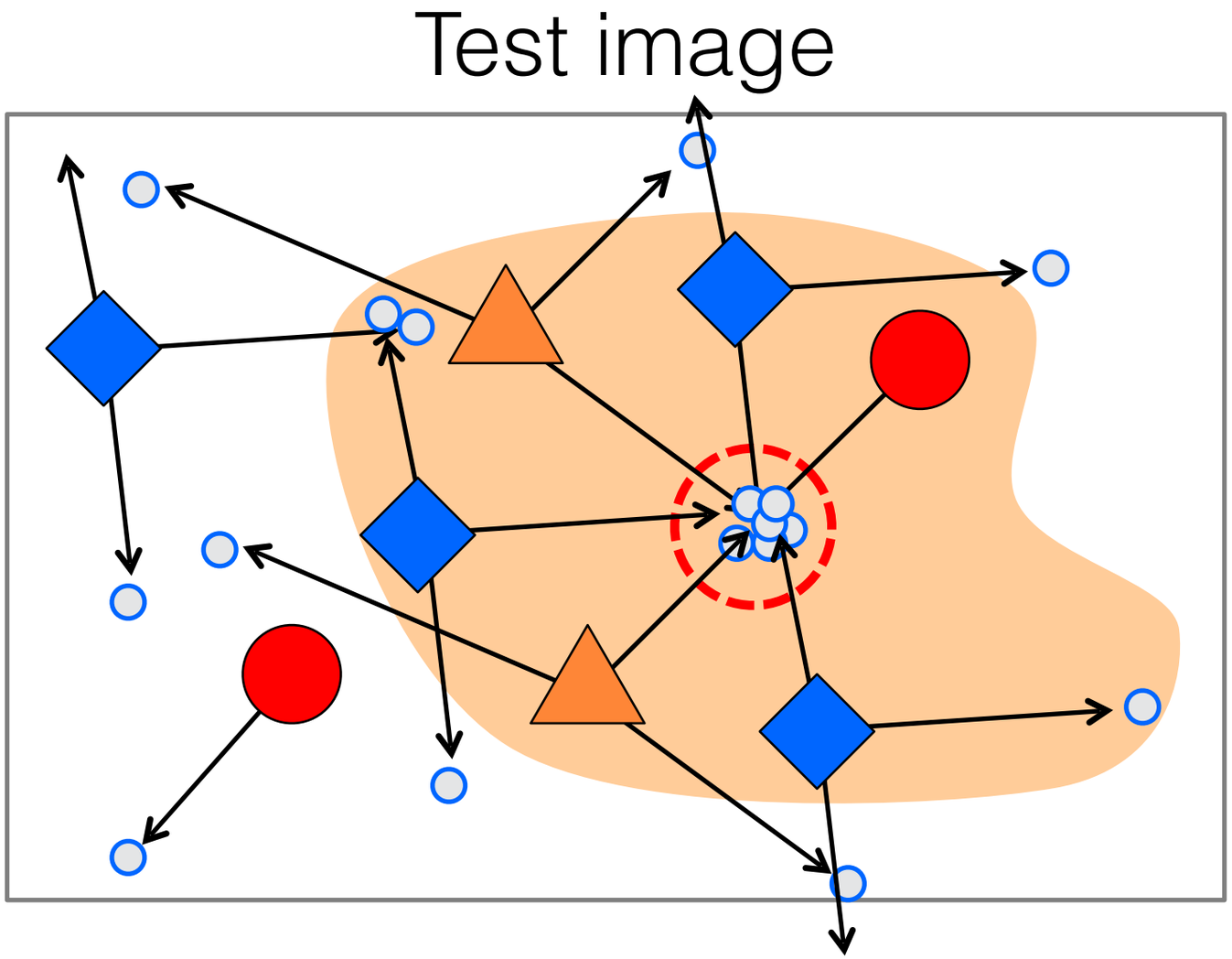
   1. Get features

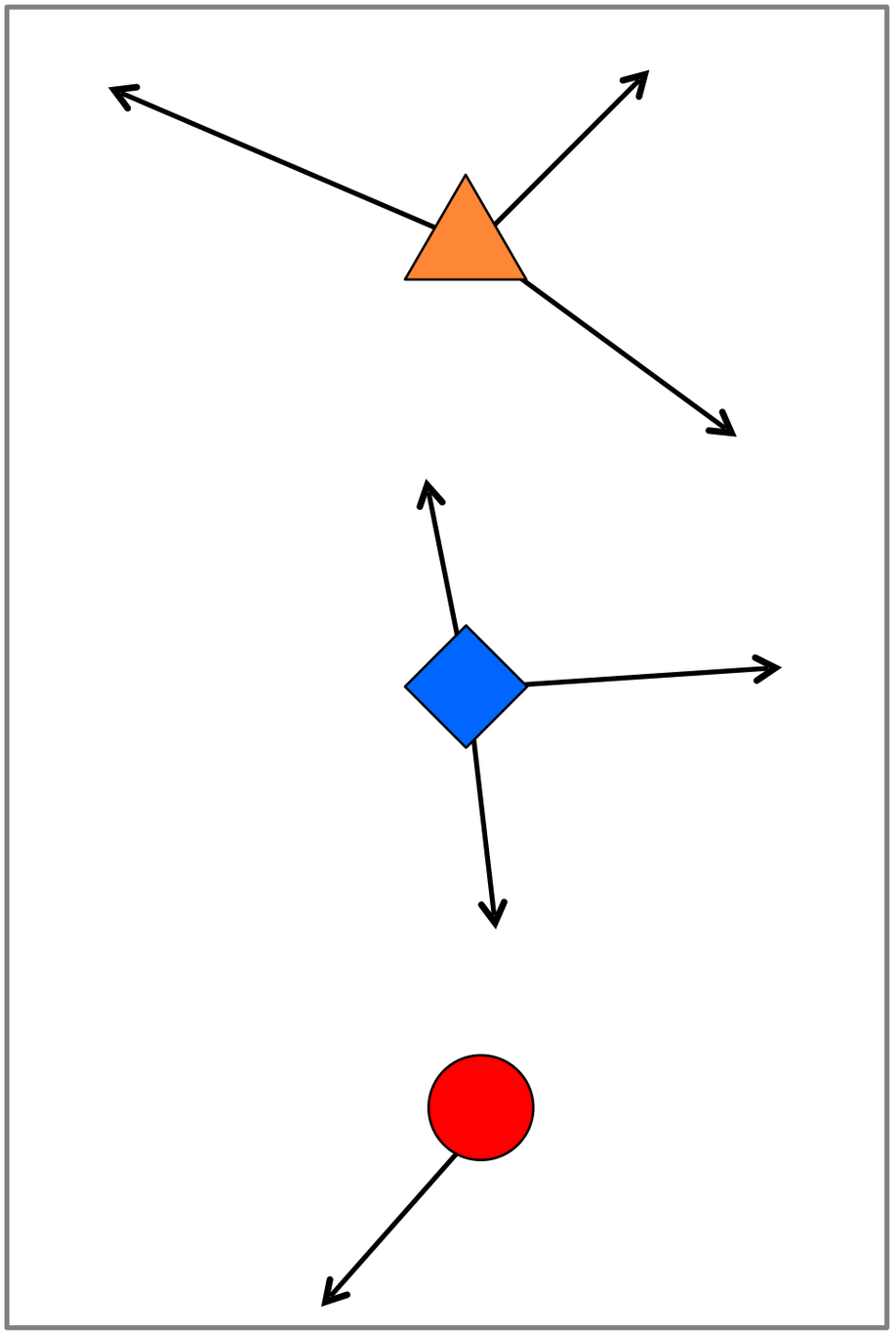
   2. Store all displacements of feature from center

B. Test phase:

   1. Get features & lookup displacements

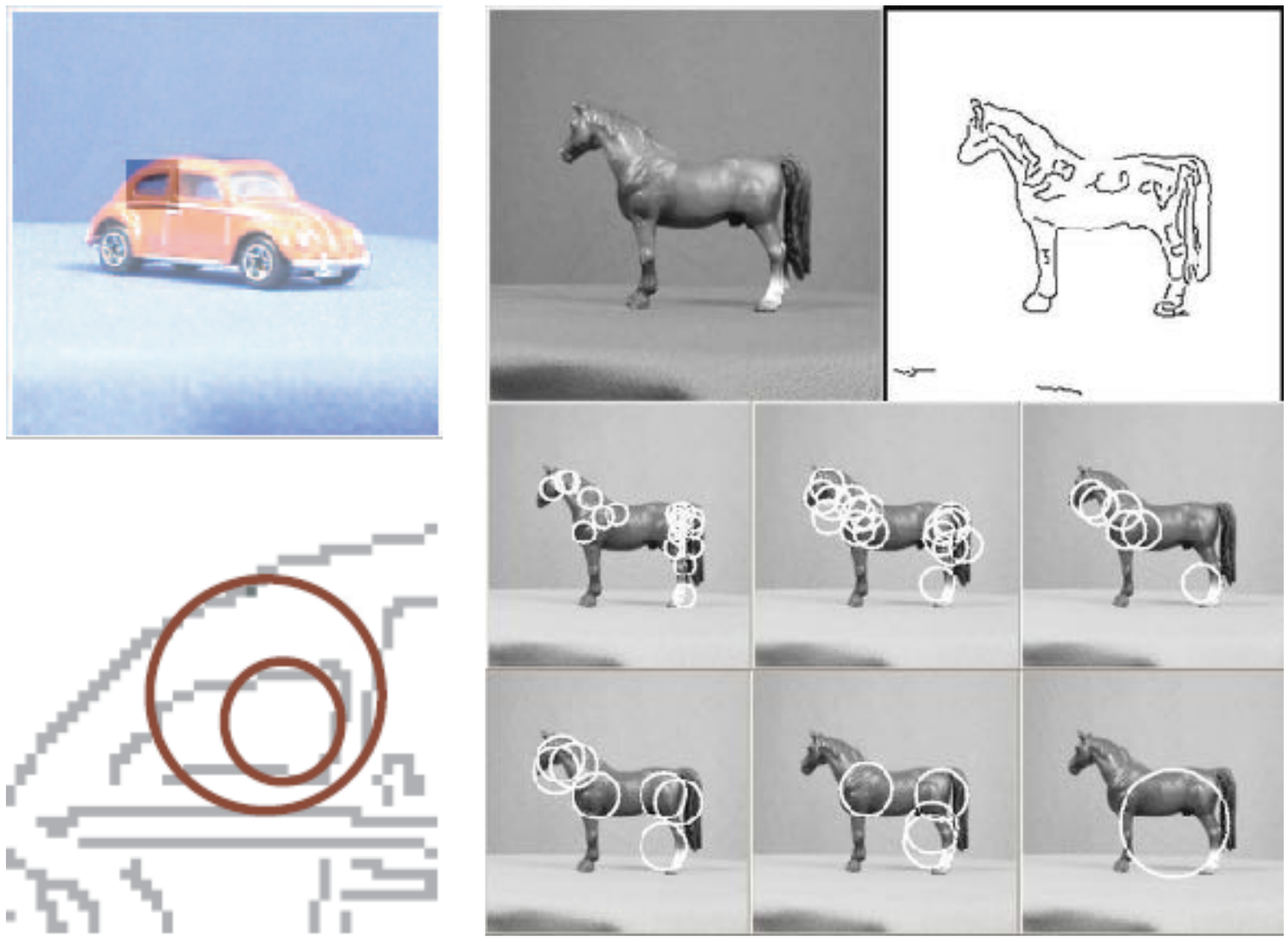
   2. Vote for center location





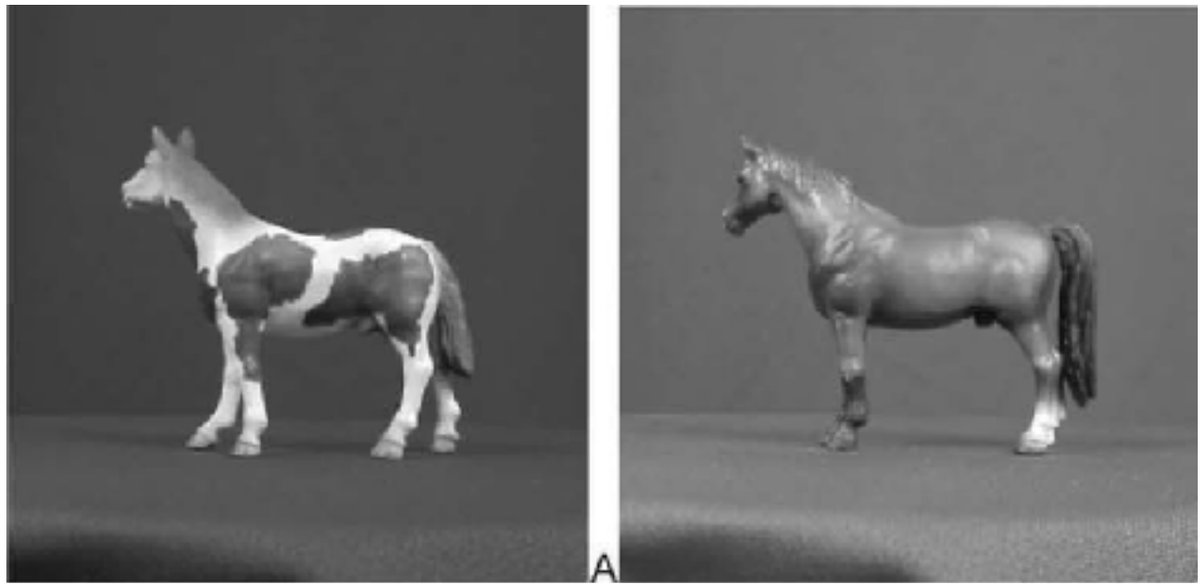
Application of Hough Transforms

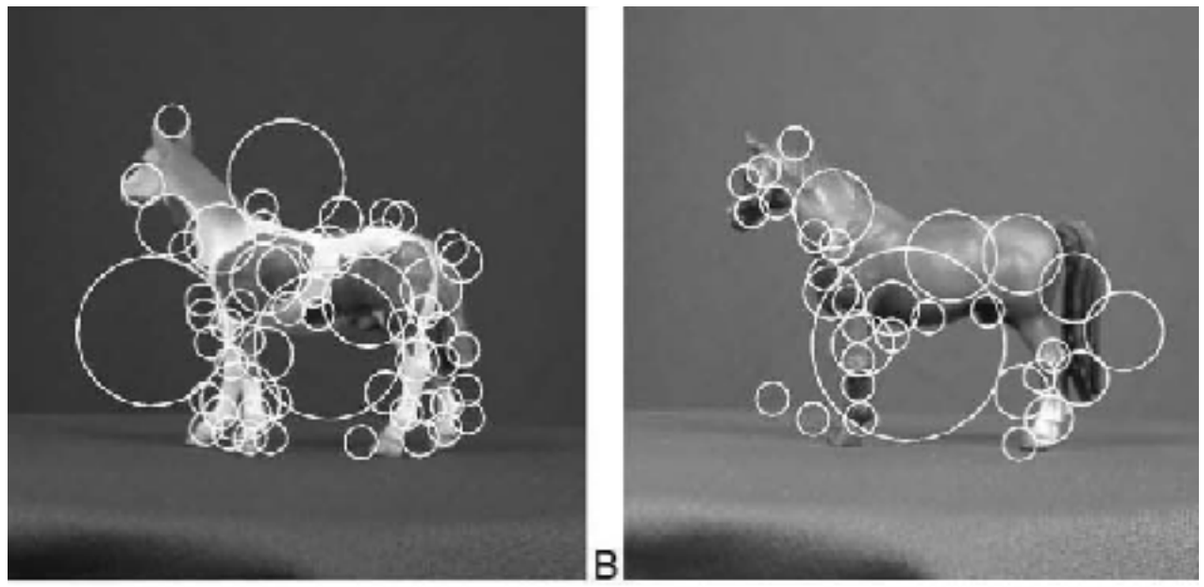
Detecting shape features



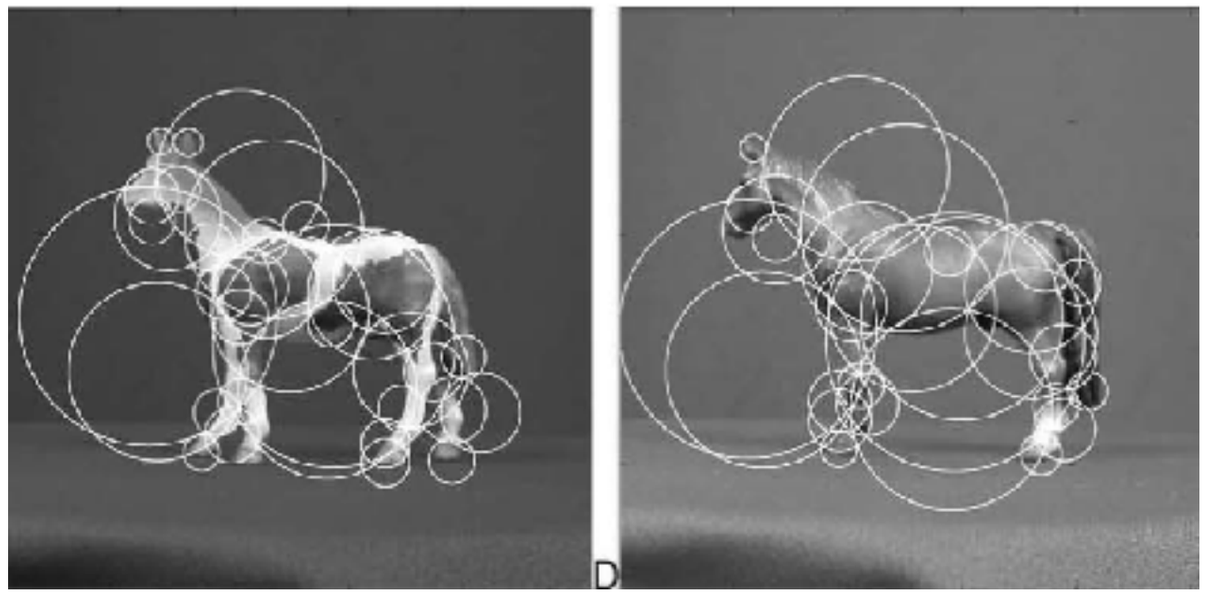
F. Jurie and C. Schmid, Scale-invariant shape features for recognition of object categories, CVPR 2004

Original images



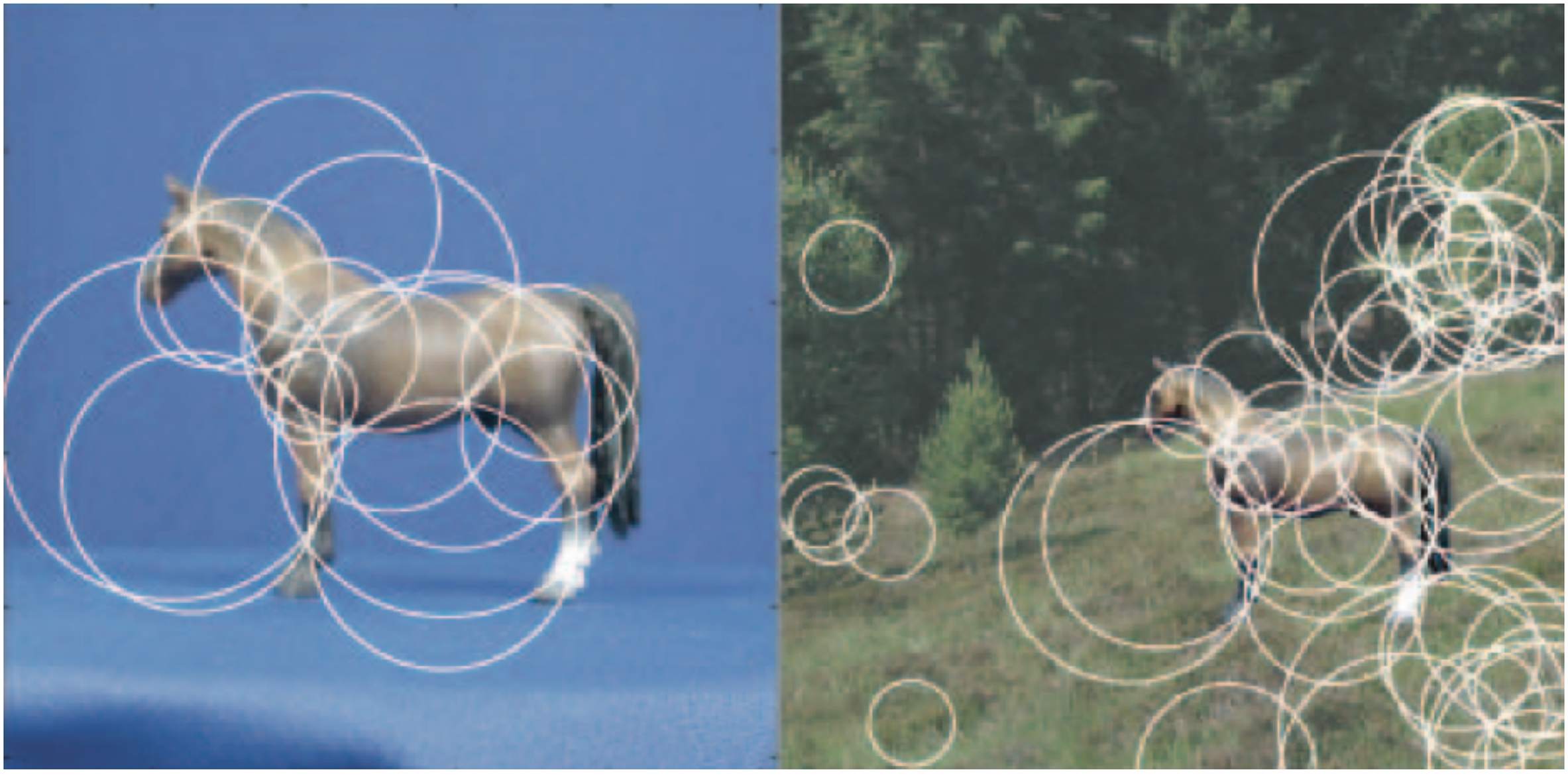


Laplacian circles



Hough-like circles

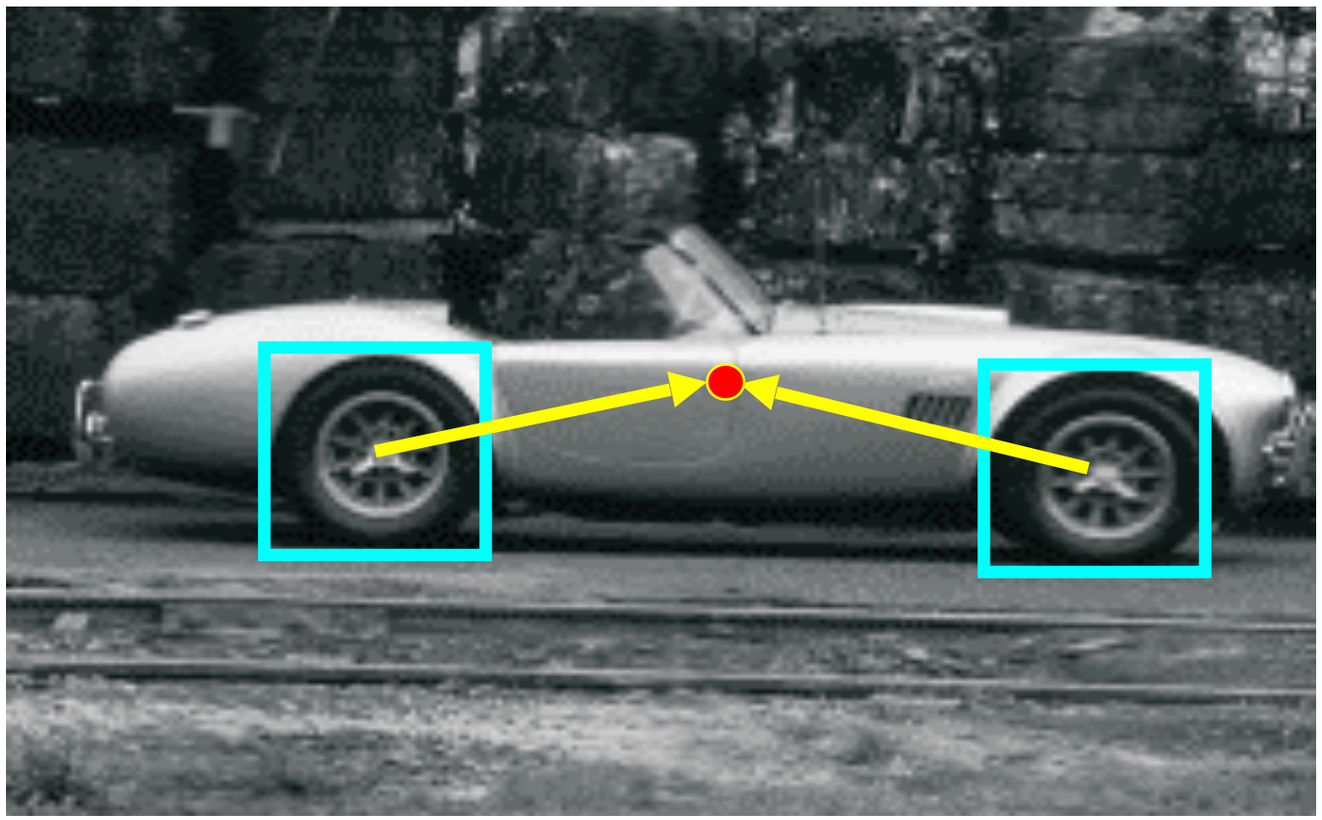
Which feature detector is more consistent?



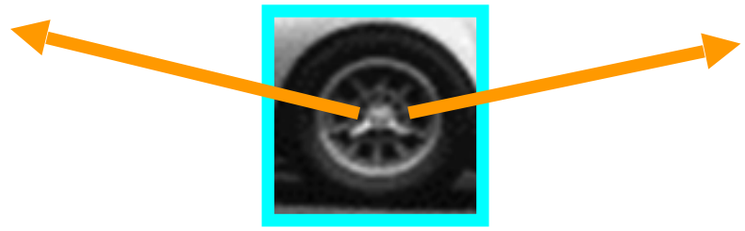
Robustness to scale and clutter

Object detection

Index displacements by “visual codeword”



training image

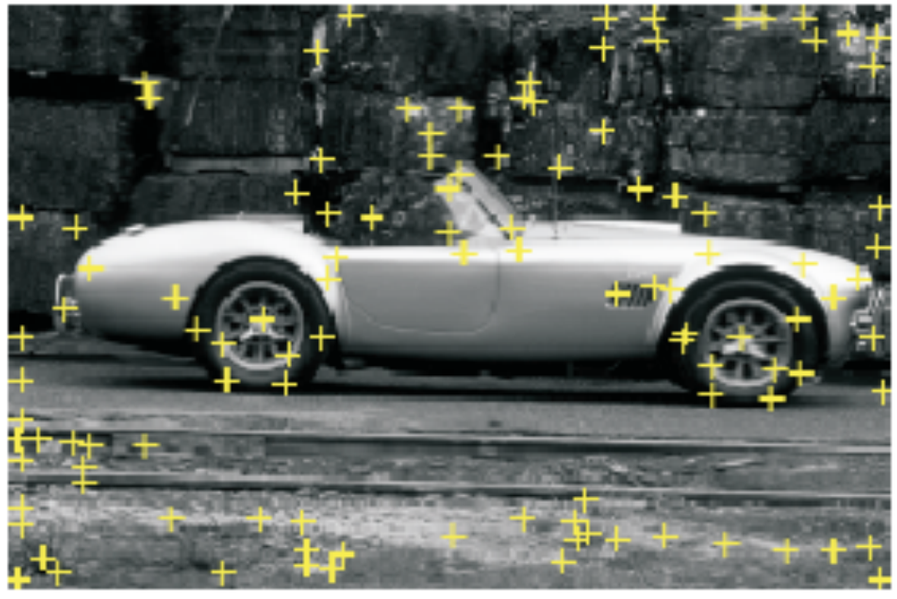


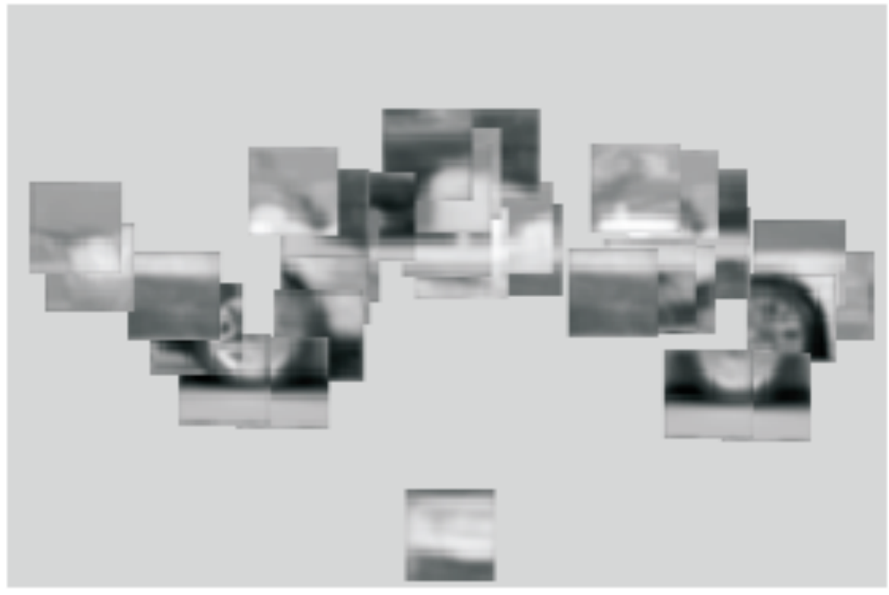
visual codeword with
displacement vectors

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

Train phase

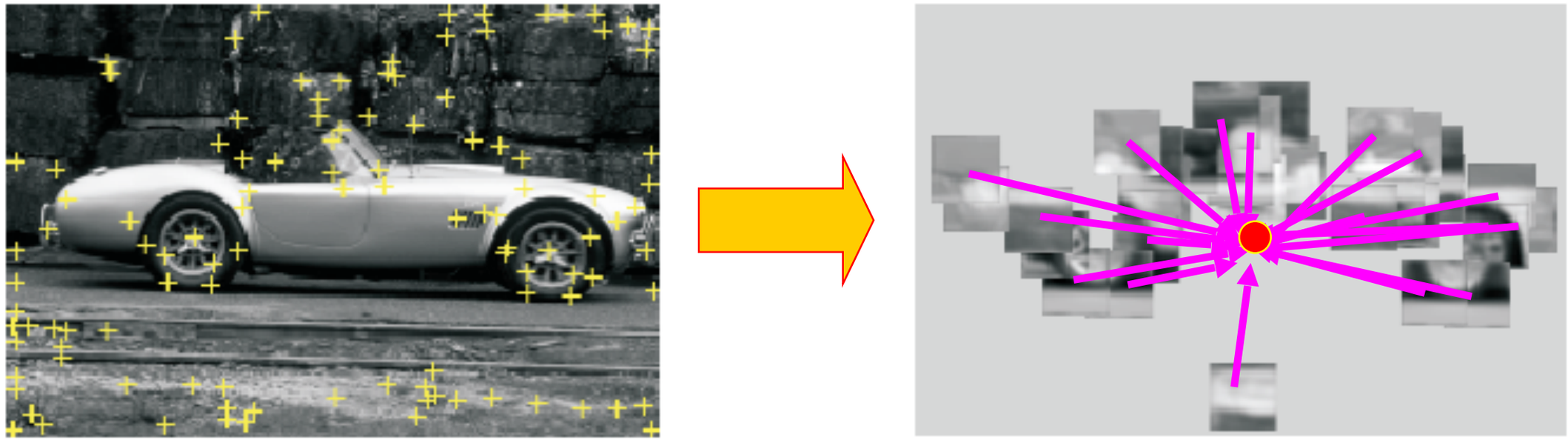
1. get features



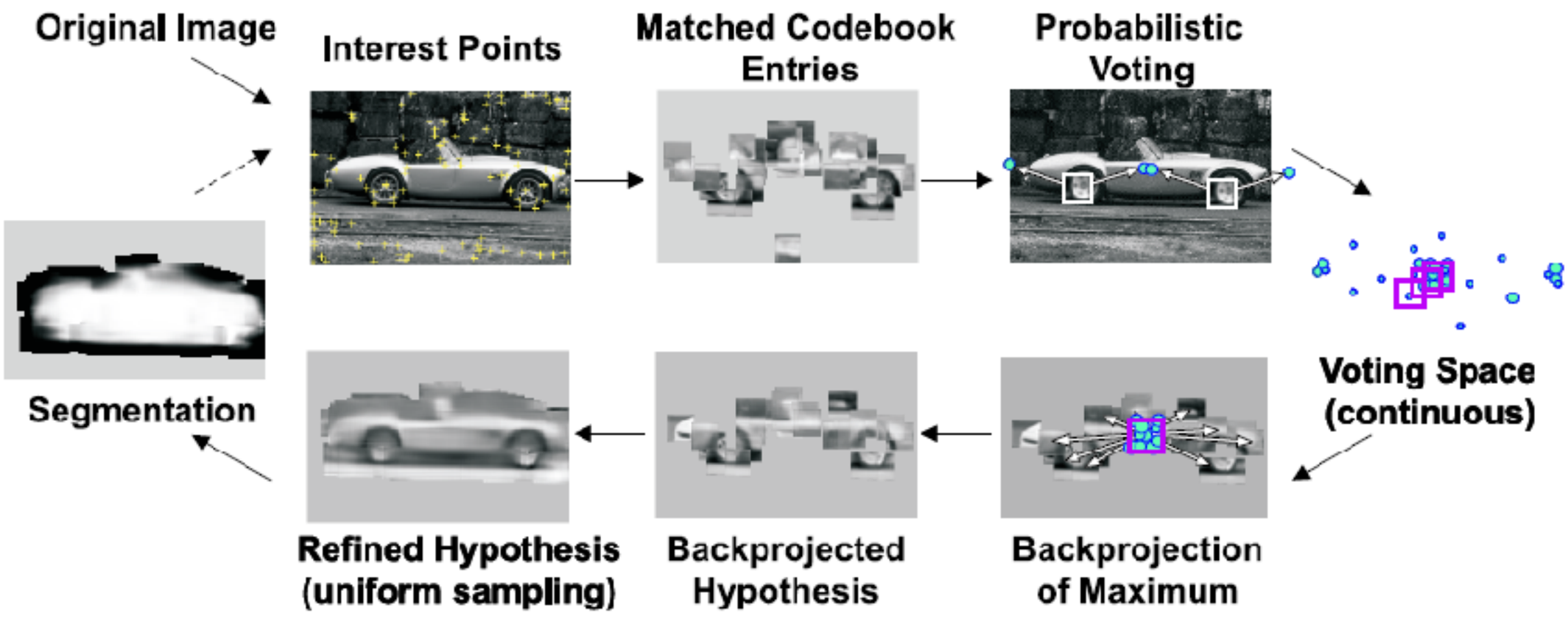


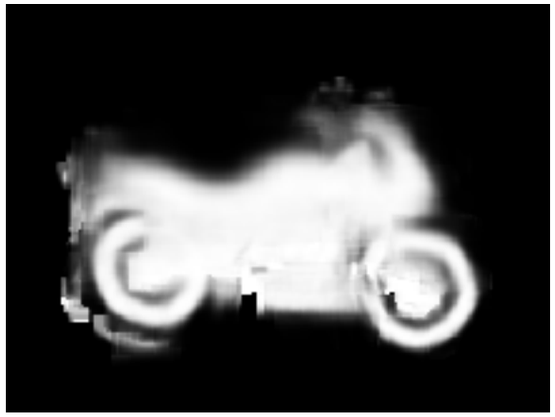
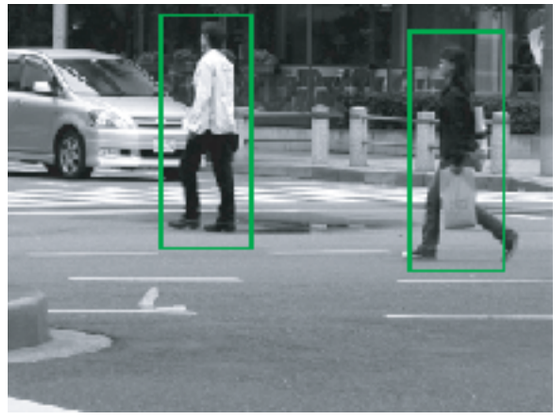
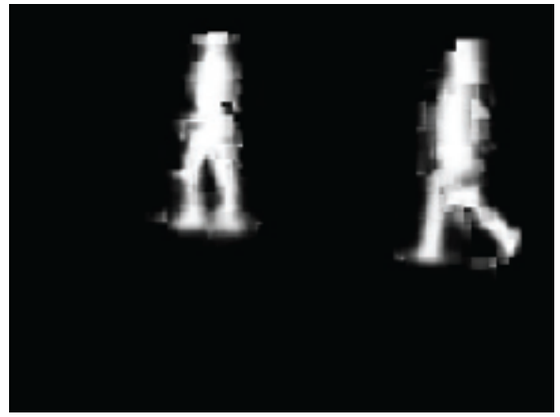
Train phase

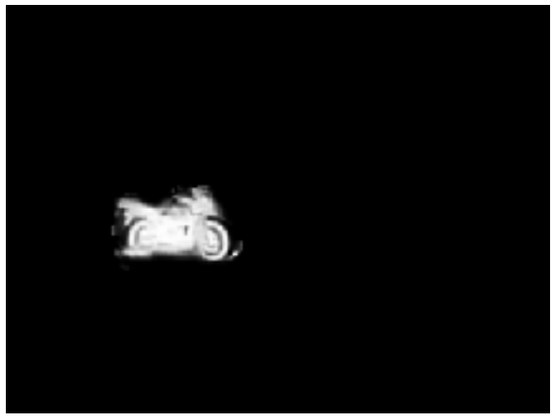
2. store displacements

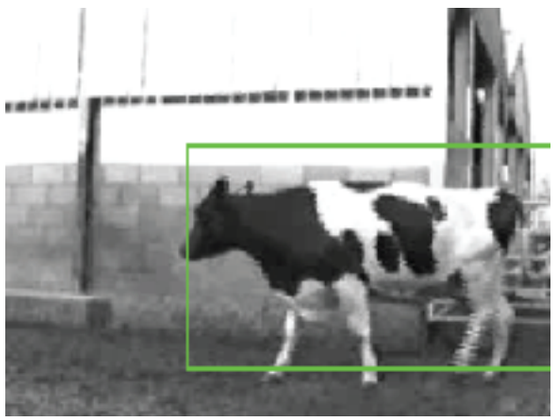


Test phase


Original Image
Interest Points
Matched Codebook Entries
Probabilistic Voting
Segmentation
Voting Space (continuous)
Refined Hypothesis (uniform sampling)
Backprojected Hypothesis
Backprojection of Maximum




The Hough transform …

Deals with occlusion well? 

Detects multiple instances? 

Robust to noise? 

Good computational complexity? 

Easy to set parameters?