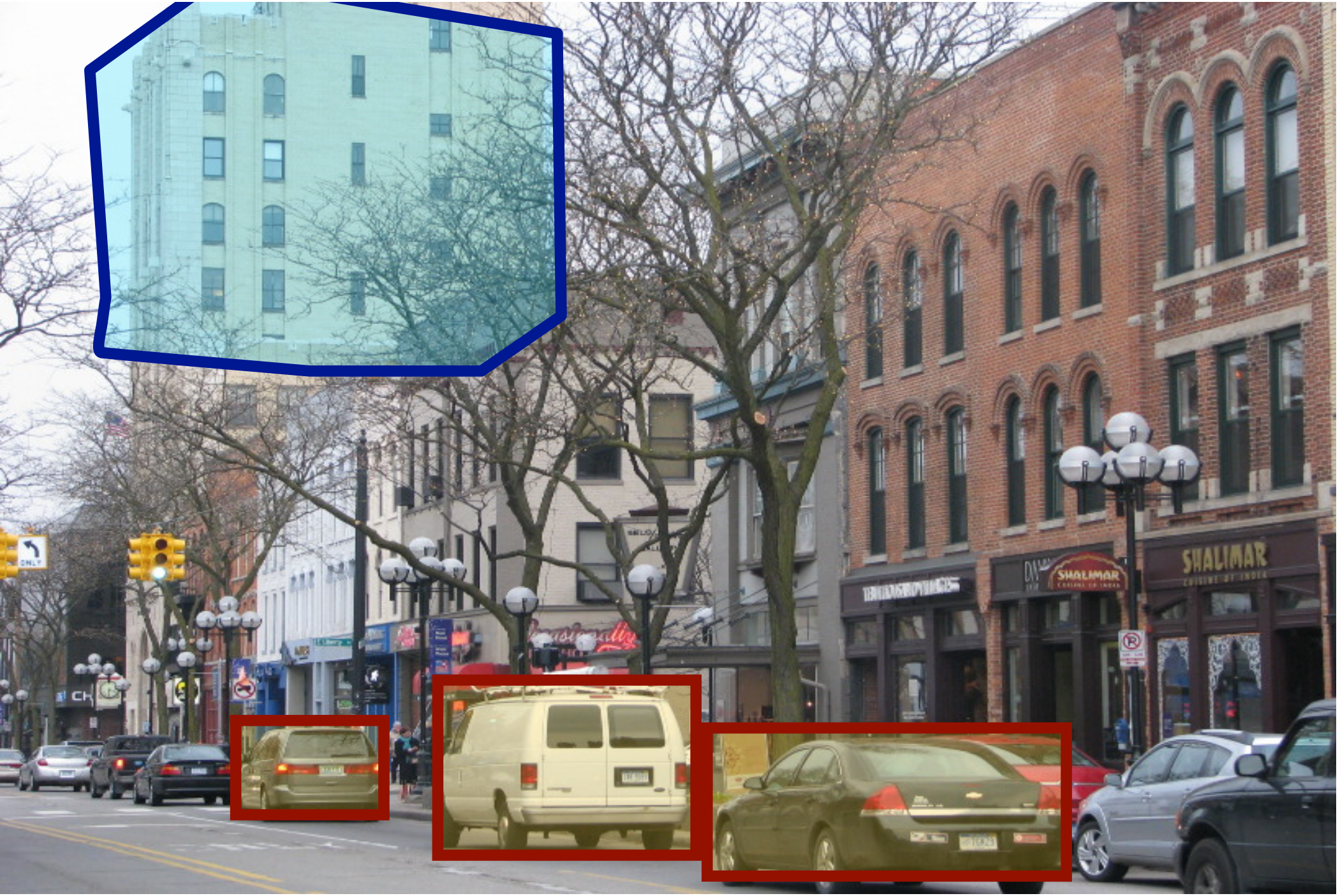
Lecture 14
Visual recognition

- 3D object detection
  - Introduction
  - Single instance 3D object detectors
  - Generic 3D object detectors

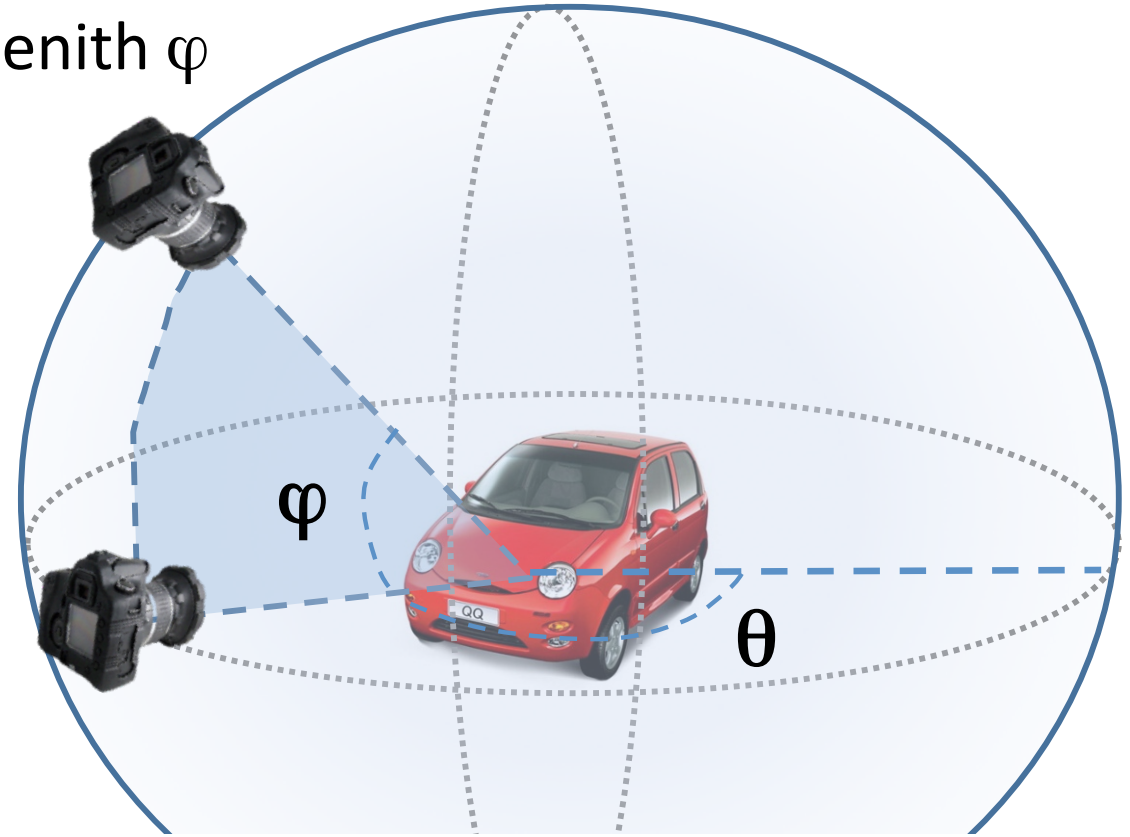
3D object detection



Properties of a 3D object detector

Viewing sphere

Azimuth θ, Zenith φ

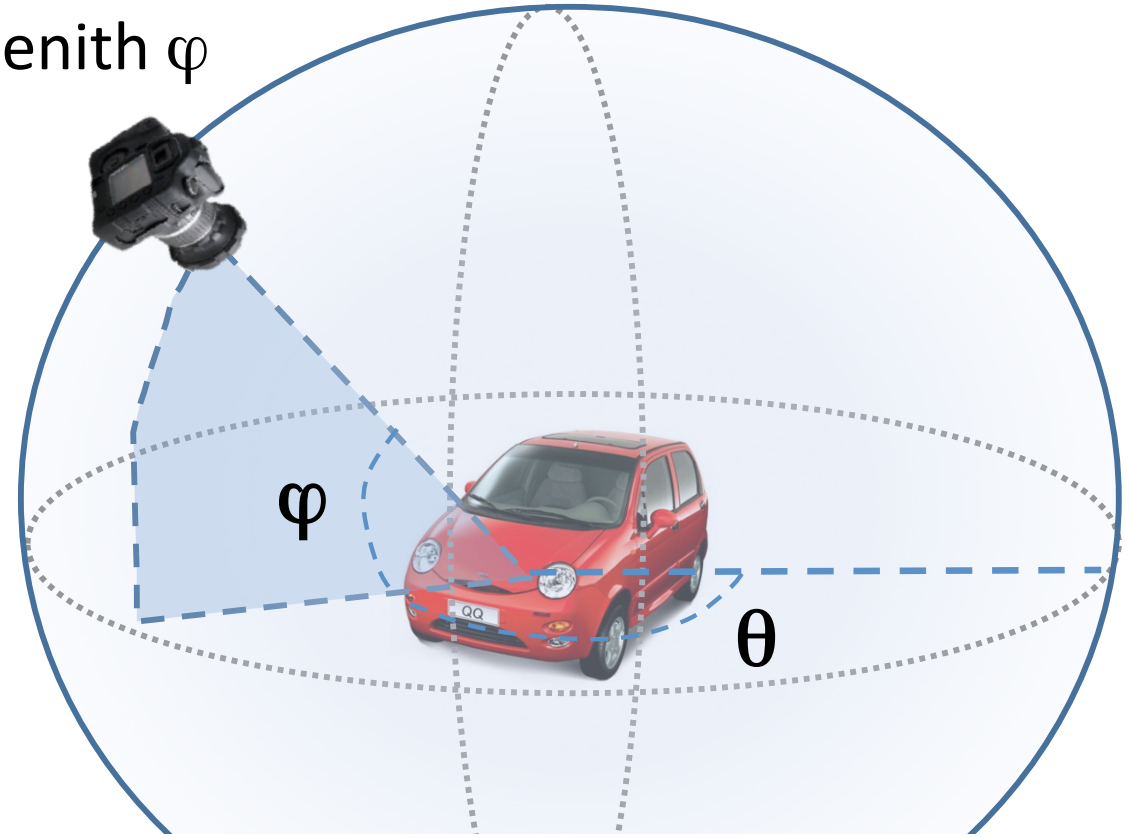


- Detect objects under generic view points
- Estimate object pose & 3D shape

Properties of a 3D object detector

Viewing sphere

Azimuth θ, Zenith φ

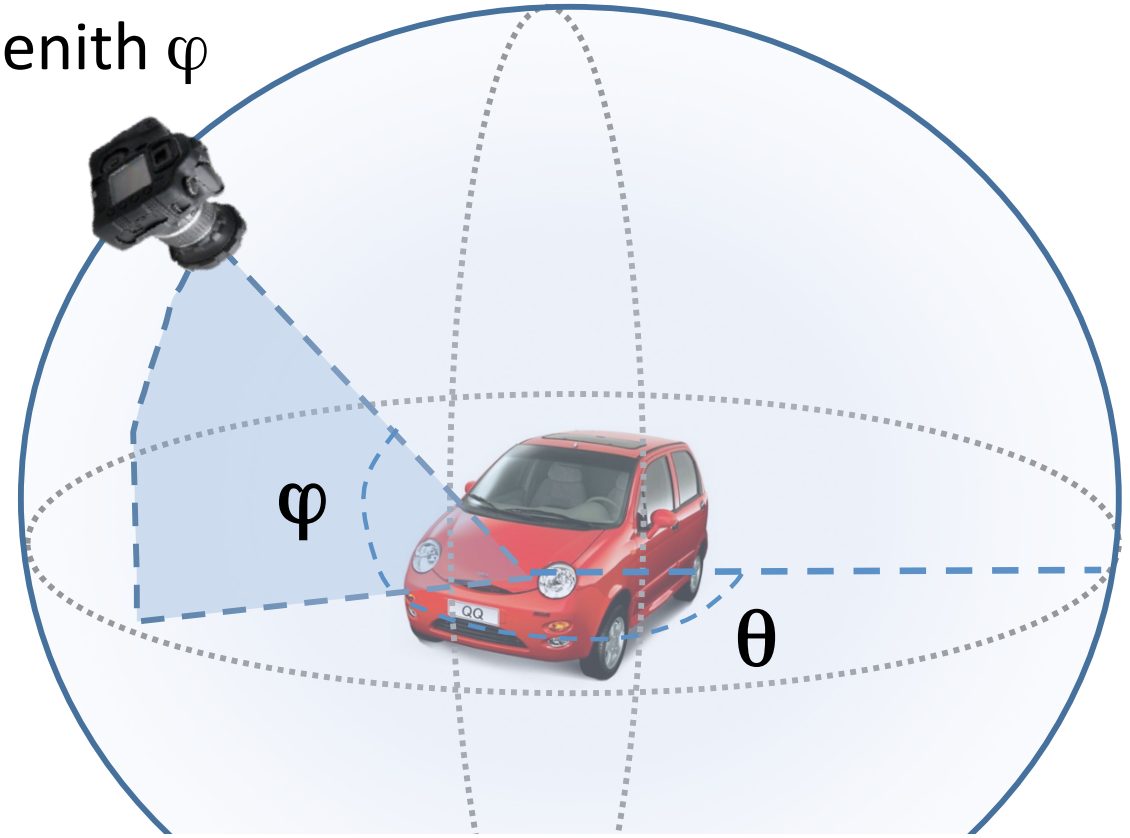


- Detect objects under generic view points
- Estimate object pose & 3D shape
- Work at different levels of specificity

Properties of a 3D object detector

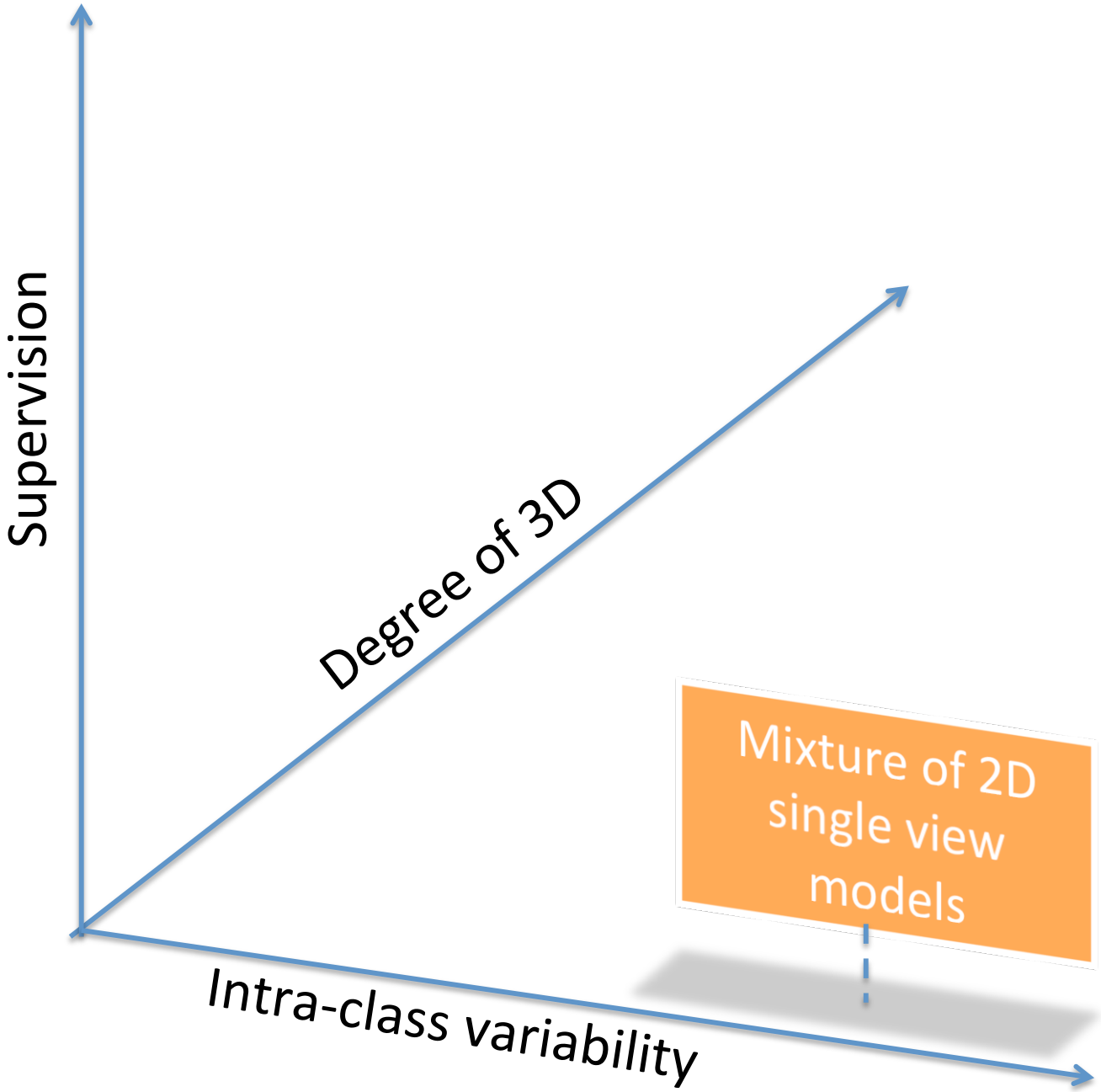
Viewing sphere

Azimuth θ, Zenith φ

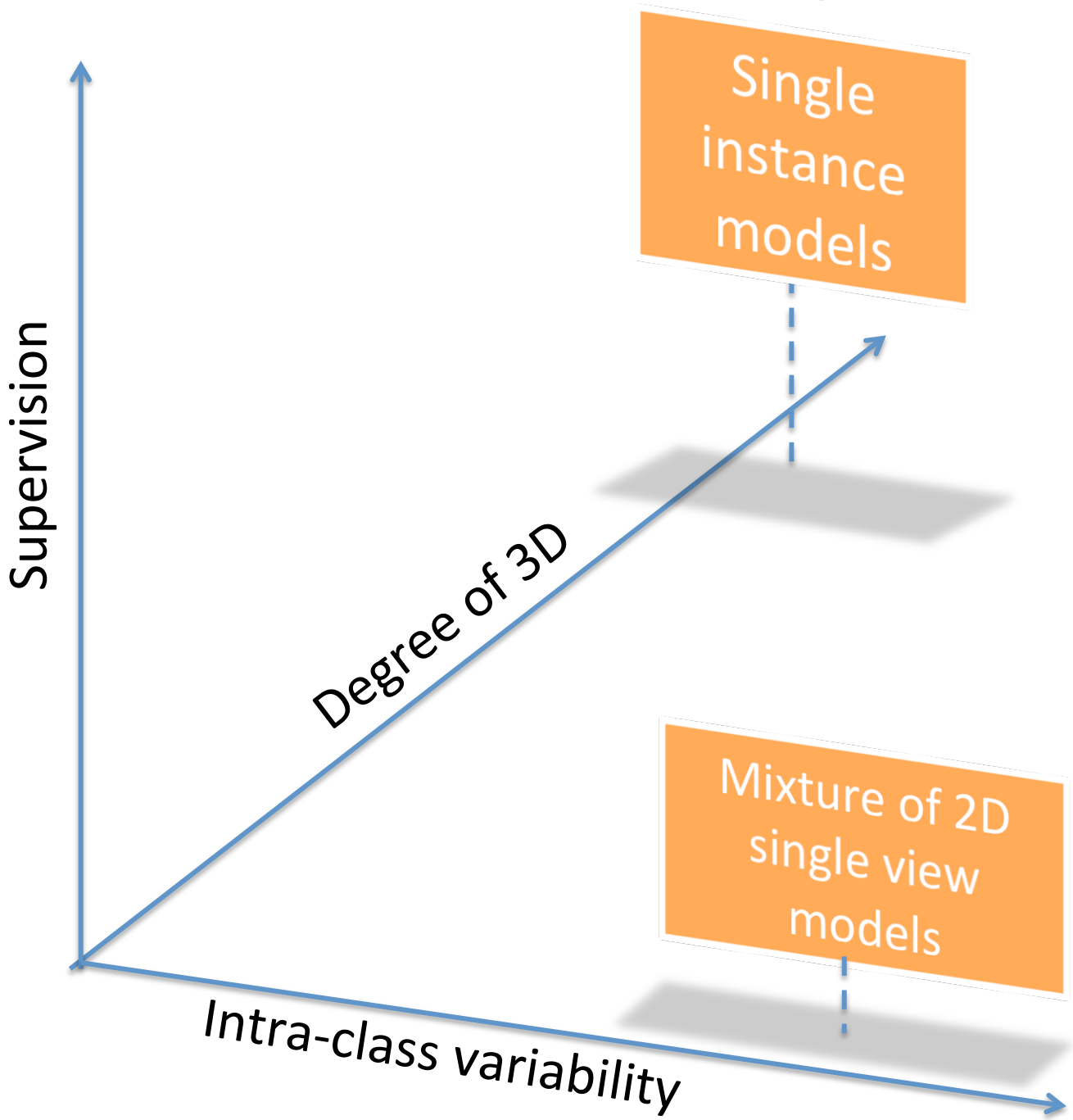


- Detect objects under generic view points
- Estimate object pose & 3D shape
- Work at different levels of specificity
- Limited amount of supervision

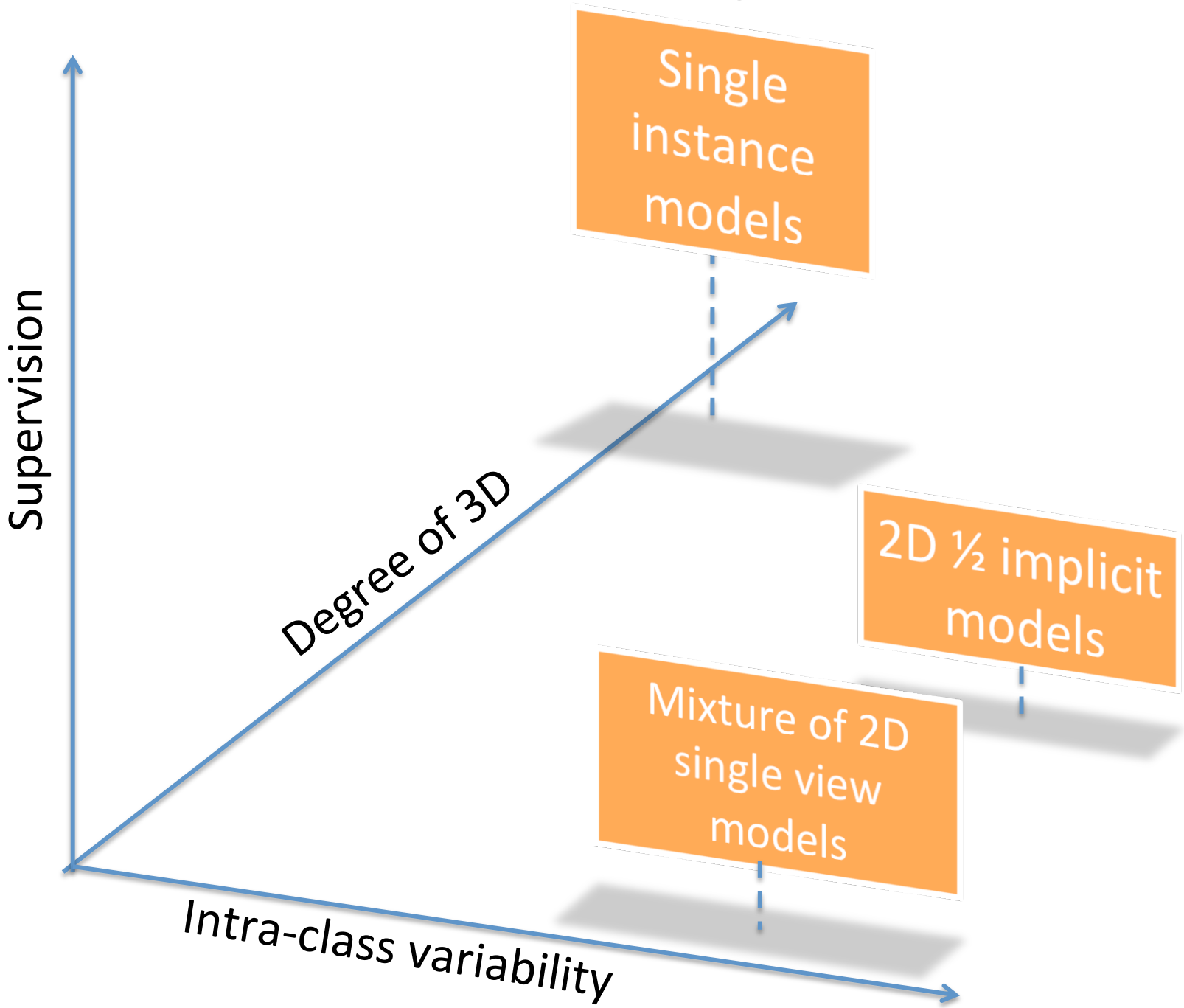
Models for 3d Object detection



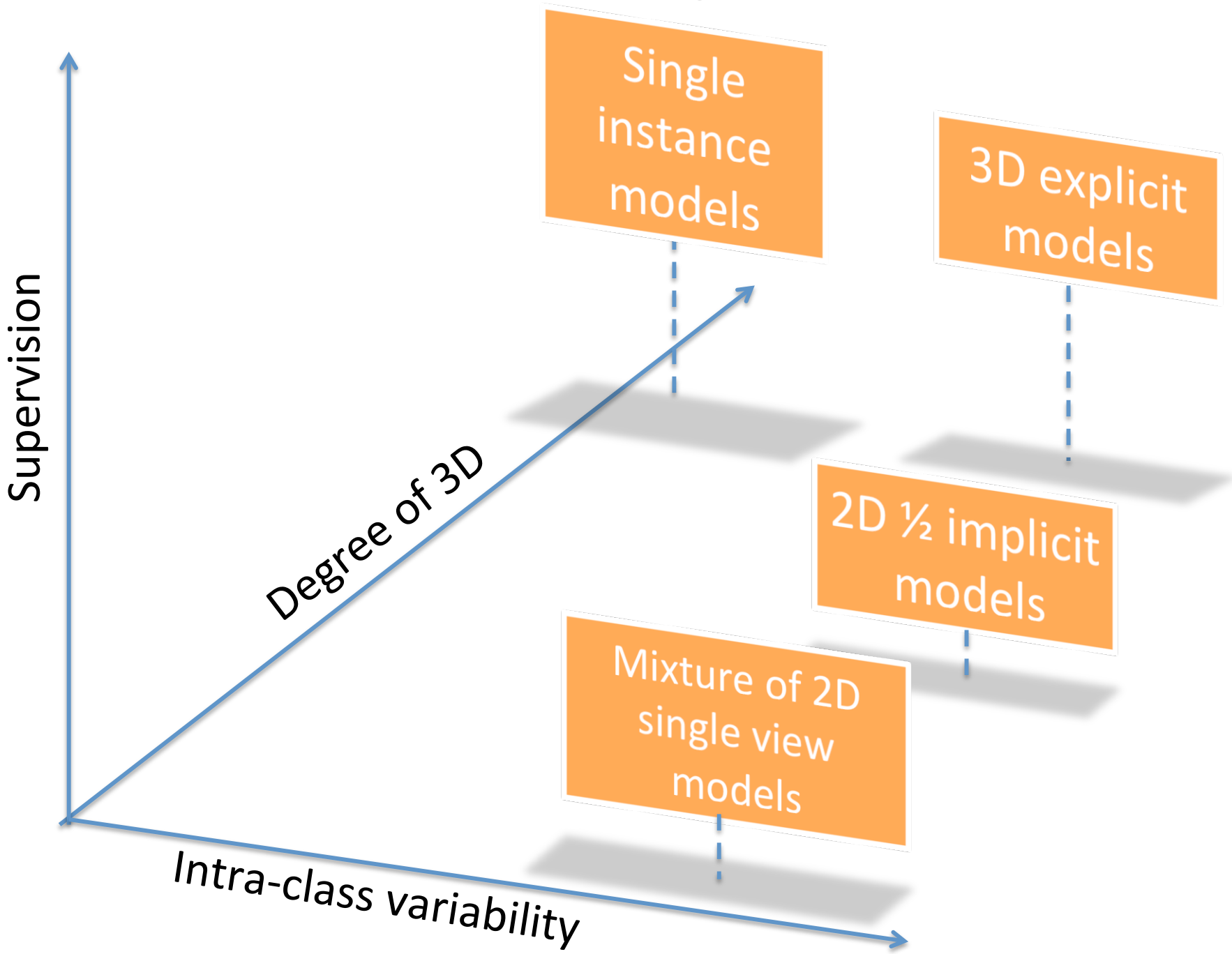
Models for 3d Object detection



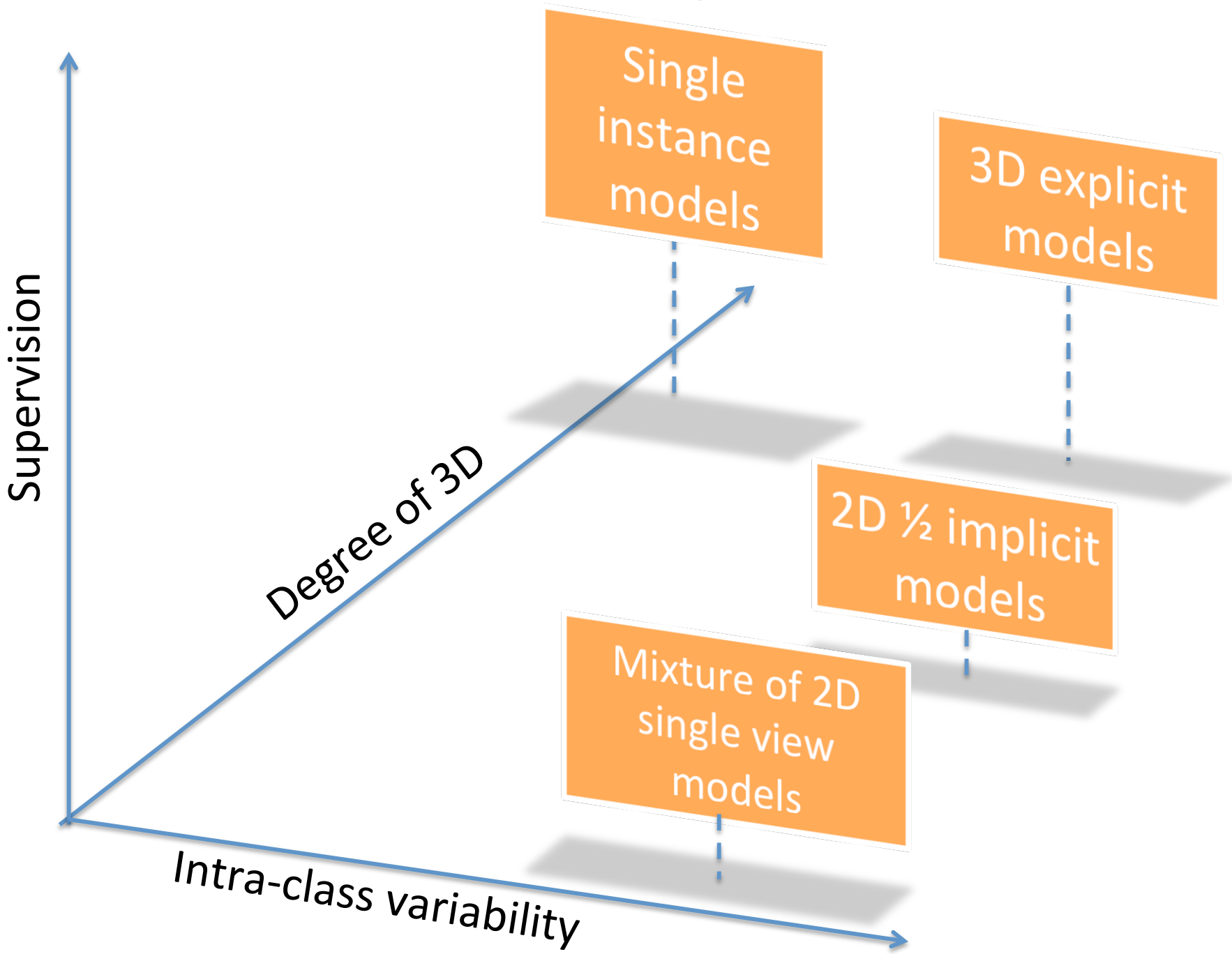
Models for 3d Object detection



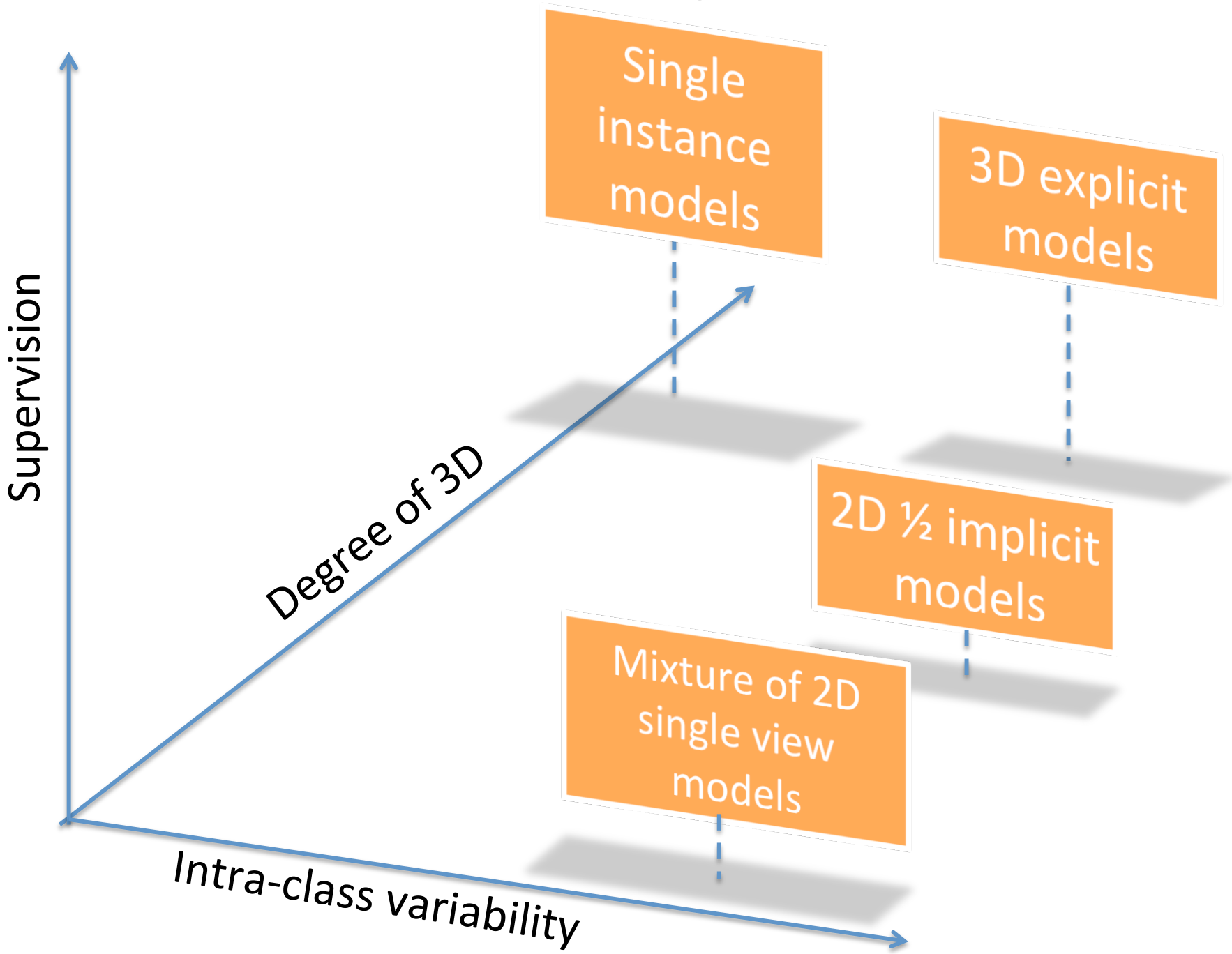
Models for 3d Object detection



Models for 3d Object detection



Models for 3d Object detection



Single 3D object recognition



















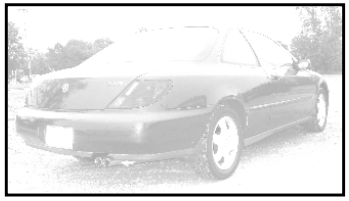
Single 3D object recognition









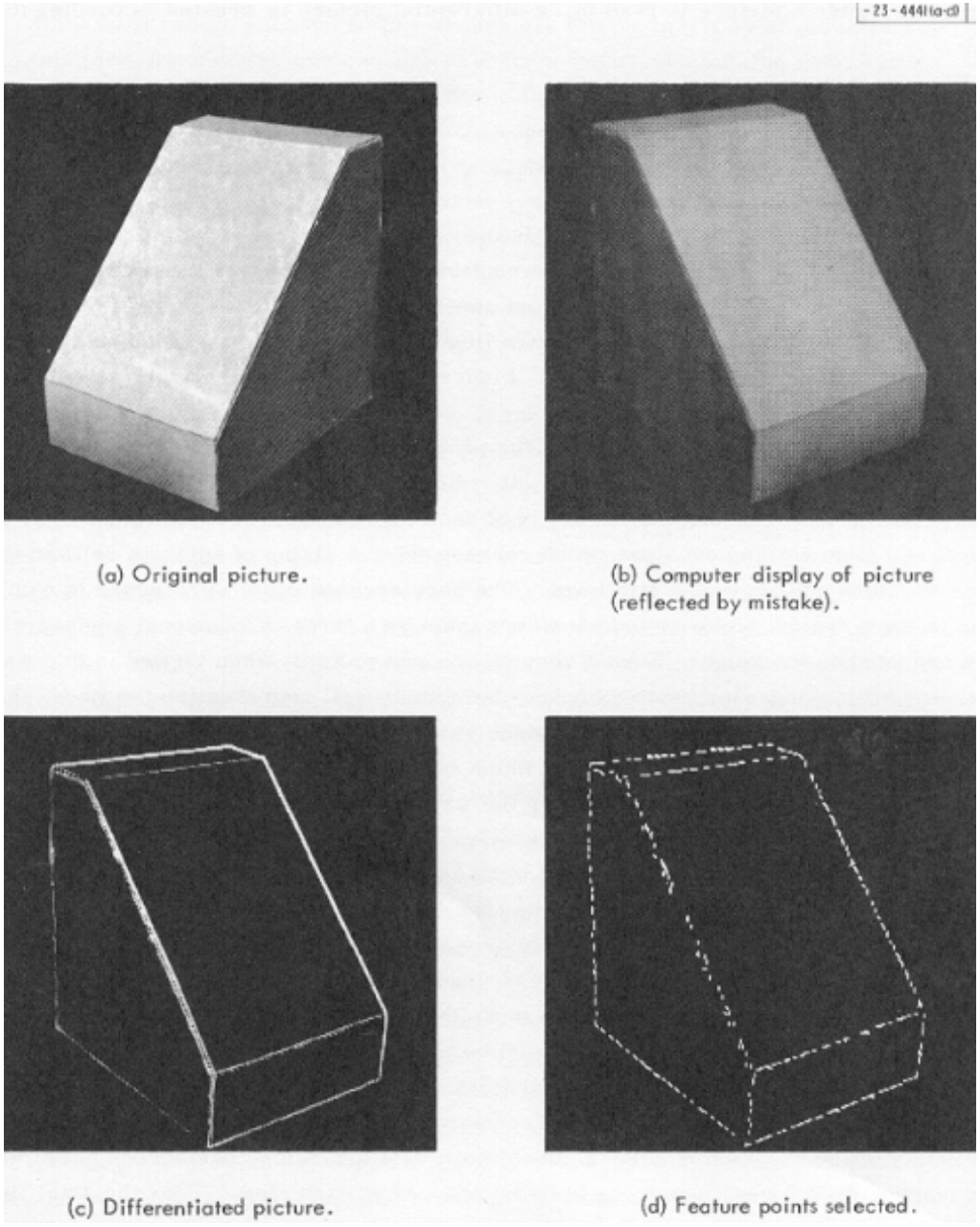







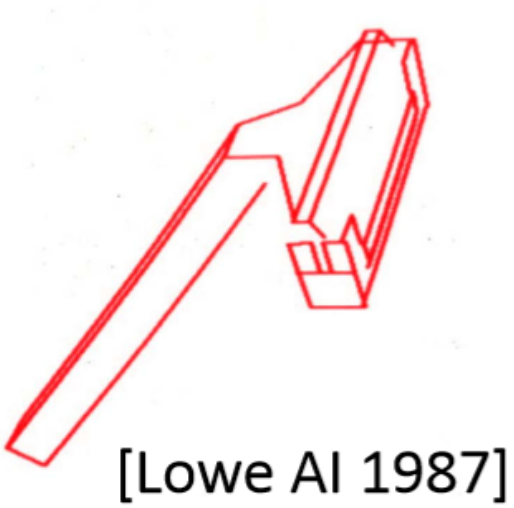


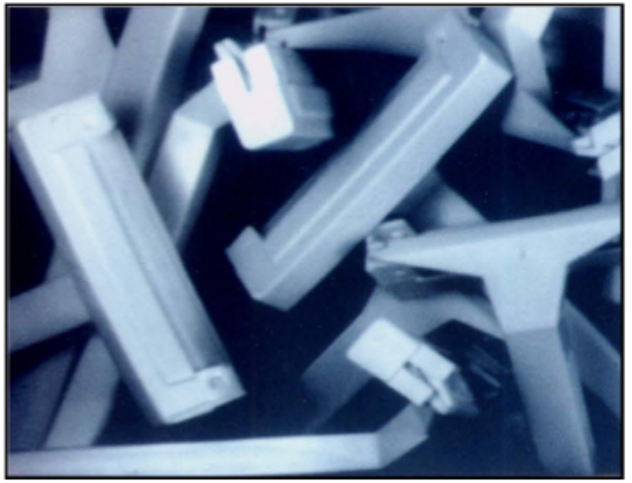

1963: Block world



Larry Roberts

80s: First 3D object detectors





- Marr ’78, ‘82
- Ballard, ‘81
- Grimson & L.-Perez, ‘87

- Lowe, ’87
- Forsyth et al. ‘91
- Edelman et al. ’91

- Ullman & Barsi, ’91
- Rothwell ‘92
- Linderberg, ’94
- Murase & Nayar ‘94

Key Challenges

Variability due to:

- View point
- Illumination
- Occlusions
- Arbitrary texture

NOTE: intra-class variability doesn't need to be modeled

Modern 3D object recognition

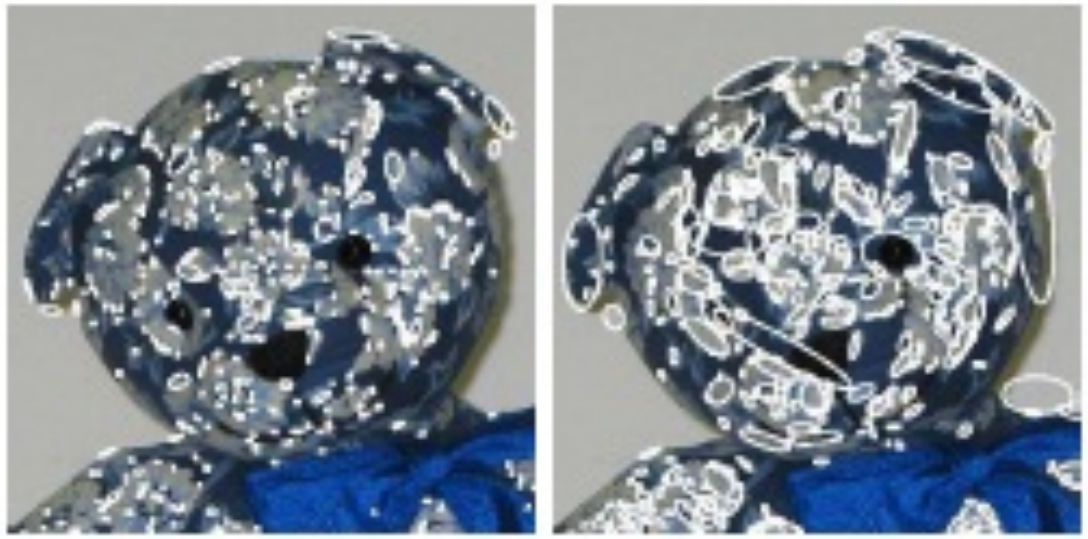
- Rothganger et al. ’04, ’06
- Brown et al, ‘05
- Lowe ‘99, ‘04
- Ferrari et al. ’04, ‘06
- Lazebnick et al ’04
- Hsiao et al., ’11-14
- Lim et al., 13

Recognition paradigm:

Hypothesis generation & validation

Object representation: 2D or 3D location of key points

Affine Harris-Laplace detector

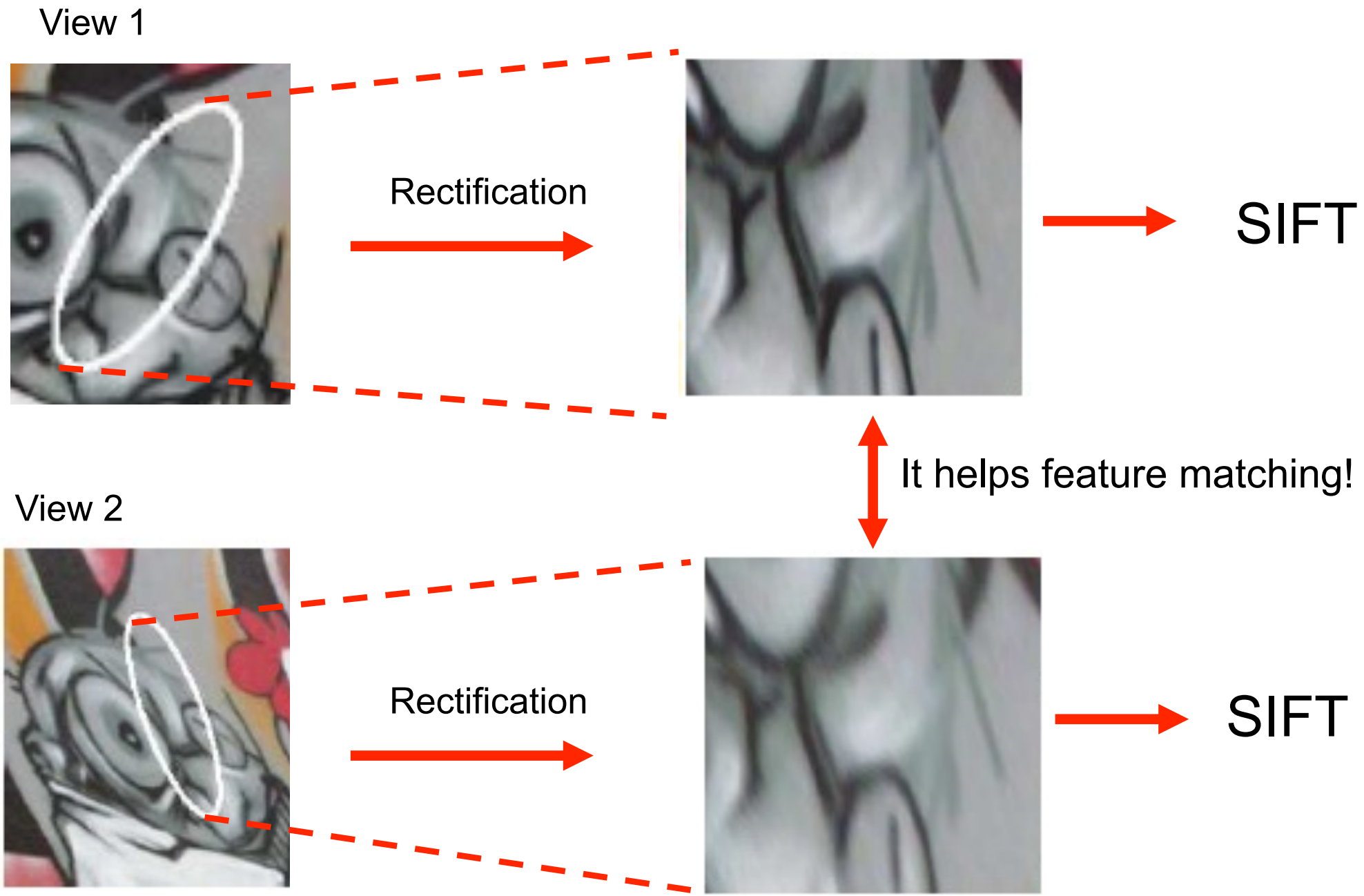


Courtesy of Rothganger et al.

- x,y
- Scale
- Orientation

Key idea: use scale and orientation to normalize descriptors

Why it is useful to normalize descriptors?



Basic scheme

-Representation

  -Features

  -2D/3D Geometrical constraints

-Model learning

-Recognition

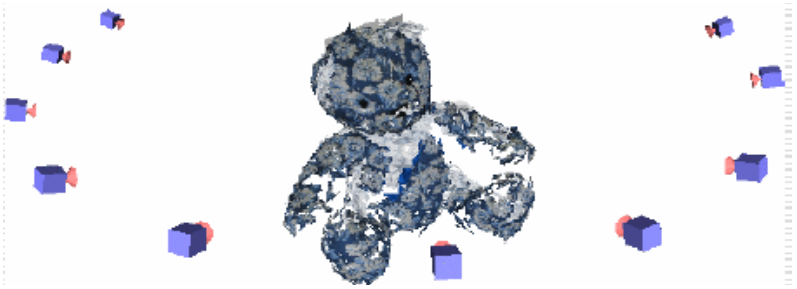
  -hypothesis generation

  -validation

Model learning

Build a 3D model:

- N images of object from N different views

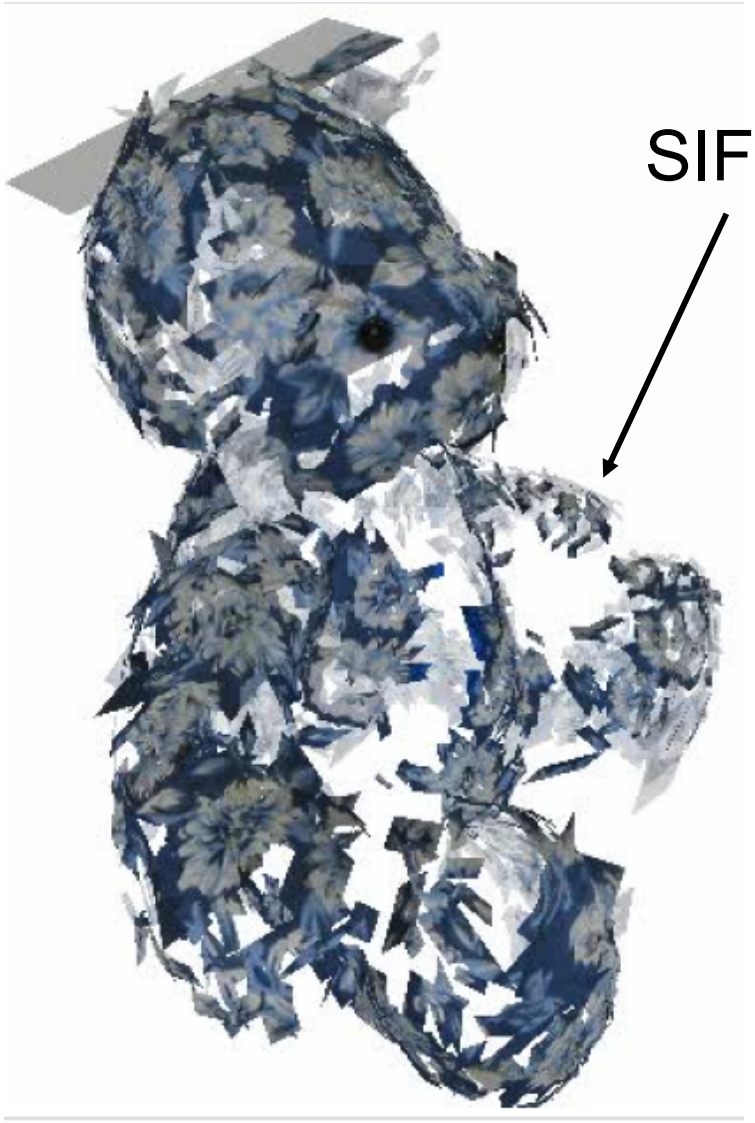


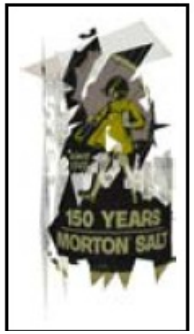
- Extract key points from each view
- Match key points between 2 views

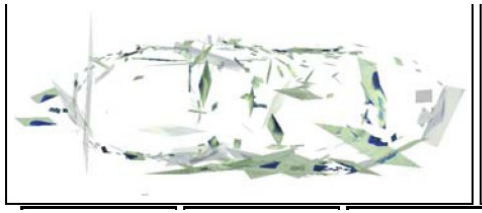
- Use affine structure from motion to compute:
  - Keypoints 3D location and orientation
  - Camera locations from 2 views
- Find connected components
- Use bundle adjustment to refine the model
- Upgrade model to Euclidean assuming zero skew and square pixels

Learnt models

x,y,z +
h,v +
SIFT descriptor







Courtesy of Rothganger et al

Basic scheme

-Representation

  -Features

  -2D/3D Geometrical constraints

-Model learning

-Recognition [object instance from object model]

  -hypothesis generation

  -Model verification

Recognition

Goal: given a query image I, detect object instance and estimate its pose

Equivalent to: from a collection of learnt object models, find object model that fits object in image

Equivalent to a fitting problem!

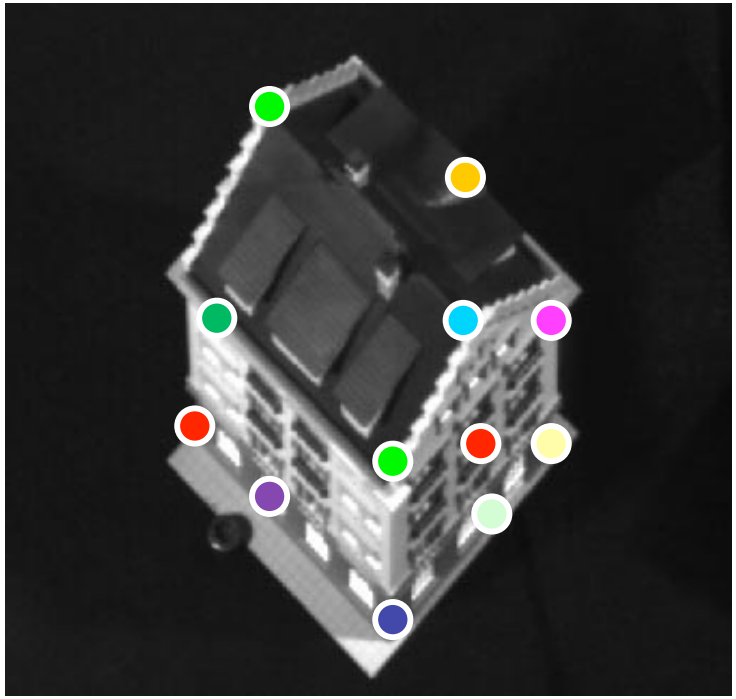
- Generate hypothesis
- Verify hypothesis
- Select hypothesis with lowest fitting error
- Generate recognition results

Recognition

Goal: given a query image I, find object model that matches with I

Model: collection of points on planar surface

query



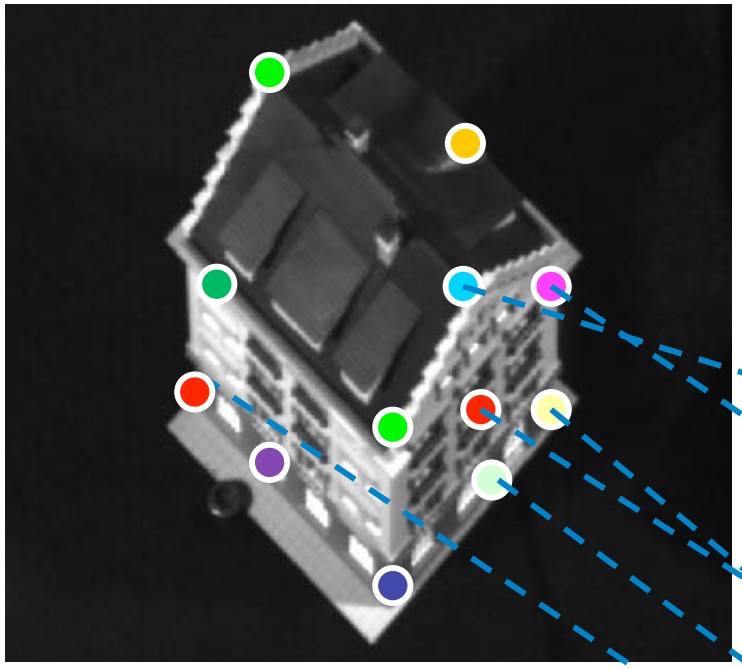
Model 1

Model 2

Recognition

- Find matches between “model” points and “query” points
- Using N matches to fit homographic transformation
- If matches and selected model are correct, the fitting error is small

query



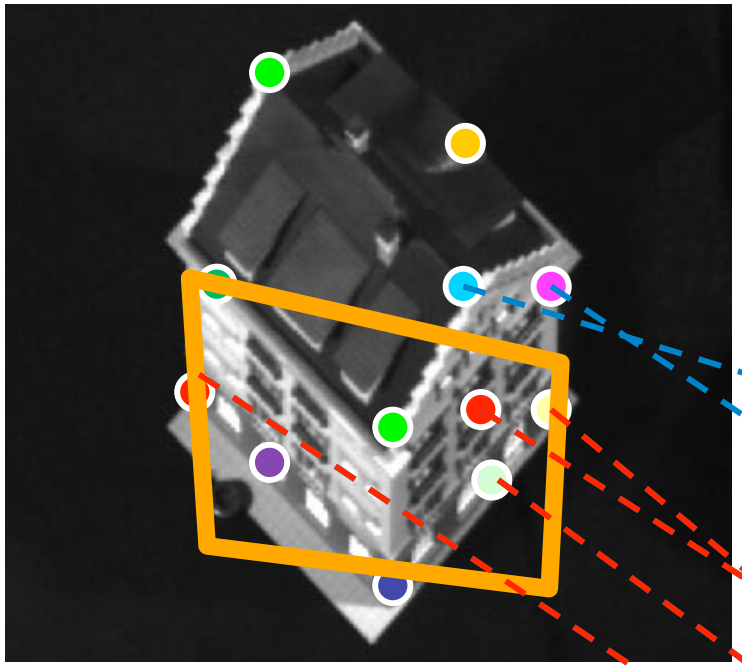
Model 1

Model 2

Recognition

- Find matches between “model” points and “query” points
- Using N matches to fit homographic transformation
- If matches and selected model are correct, the fitting error is small

query



- Generate hypothesis
- Verify hypothesis
- Select hypothesis with lowest fitting error
- Generate recognition results

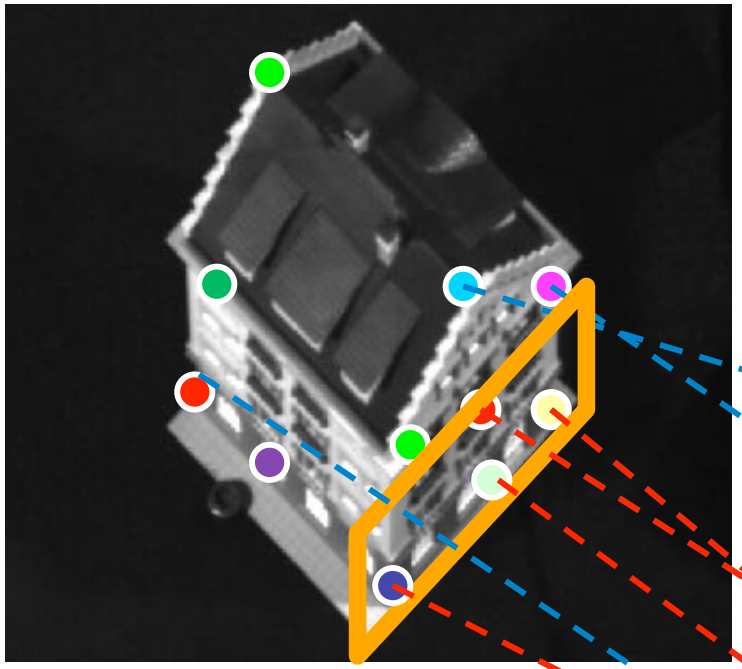
Model 2

Verification: The hypothesis generates ***high*** fitting error

Recognition

- Find matches between “model” points and “query” points
- Using N matches to fit homographic transformation
- If matches and selected model are correct, the fitting error is small

query



- Generate hypothesis
- Verify hypothesis
- Select hypothesis with lowest fitting error
- Generate recognition results

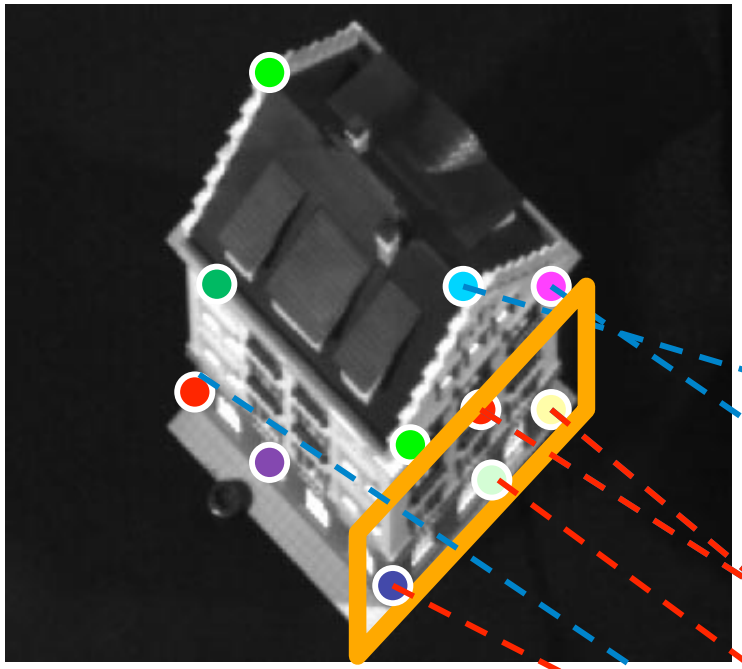
Model 2

Verification: The hypothesis generates ***low*** fitting error

Recognition

- Find matches between “model” points and “query” points
- Using N matches to fit homographic transformation
- If matches and selected model are correct, the fitting error is small

query

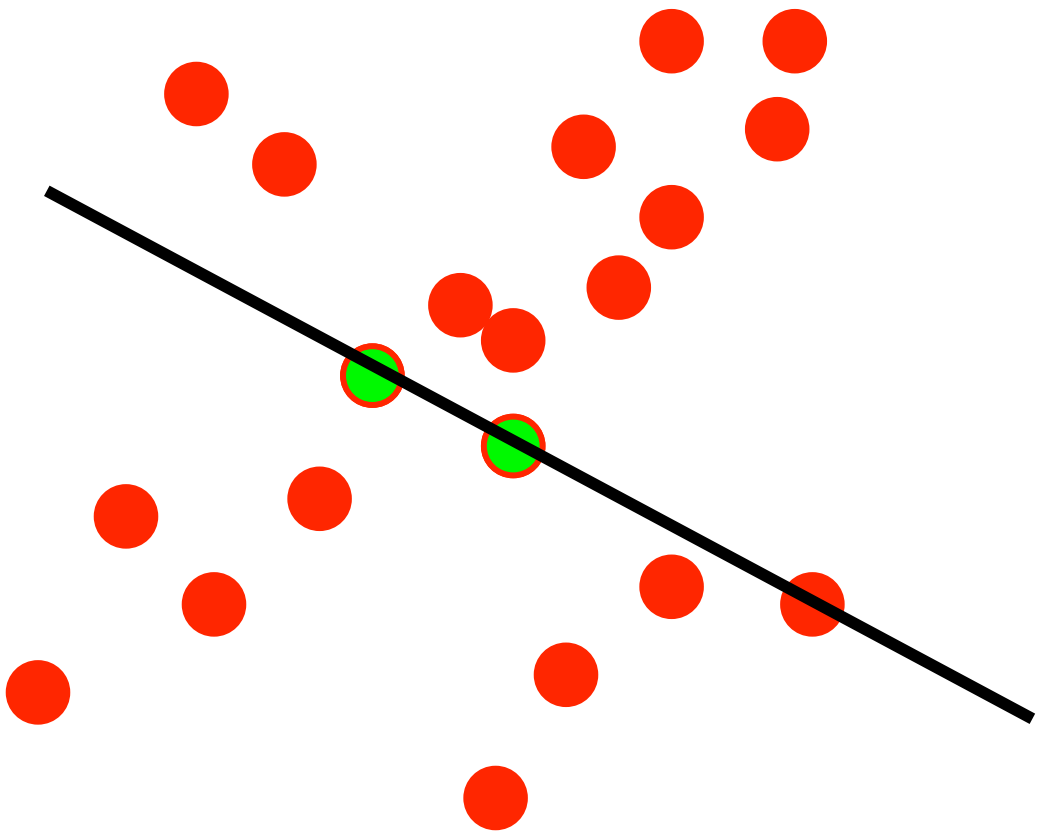


- Generate hypothesis
- Verify hypothesis
- Select hypothesis with lowest fitting error
- Generate recognition results

Model 2

Verification: The hypothesis generates ***low*** fitting error

RANSAC!



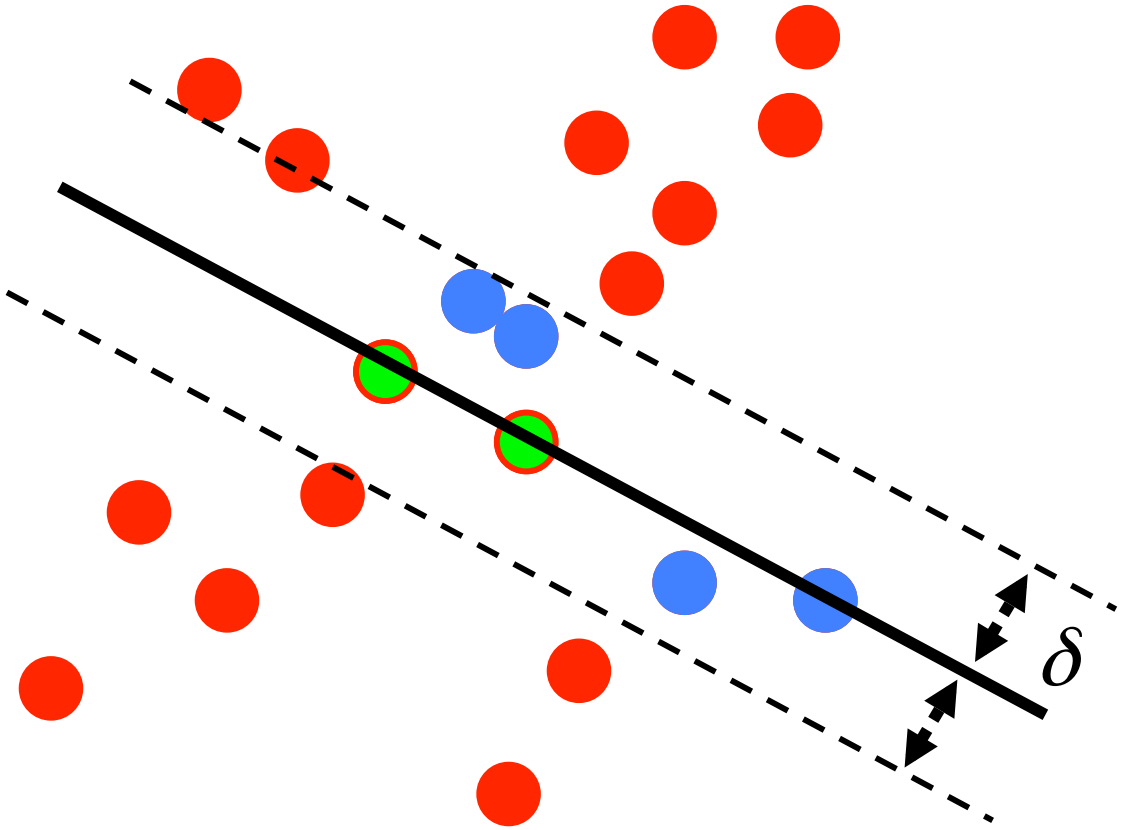
Sample set = set of points in 2D

Algorithm:

1. Select random sample of minimum required size to fit model [?] =[2]
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found

RANSAC!



Sample set = set of points in 2D

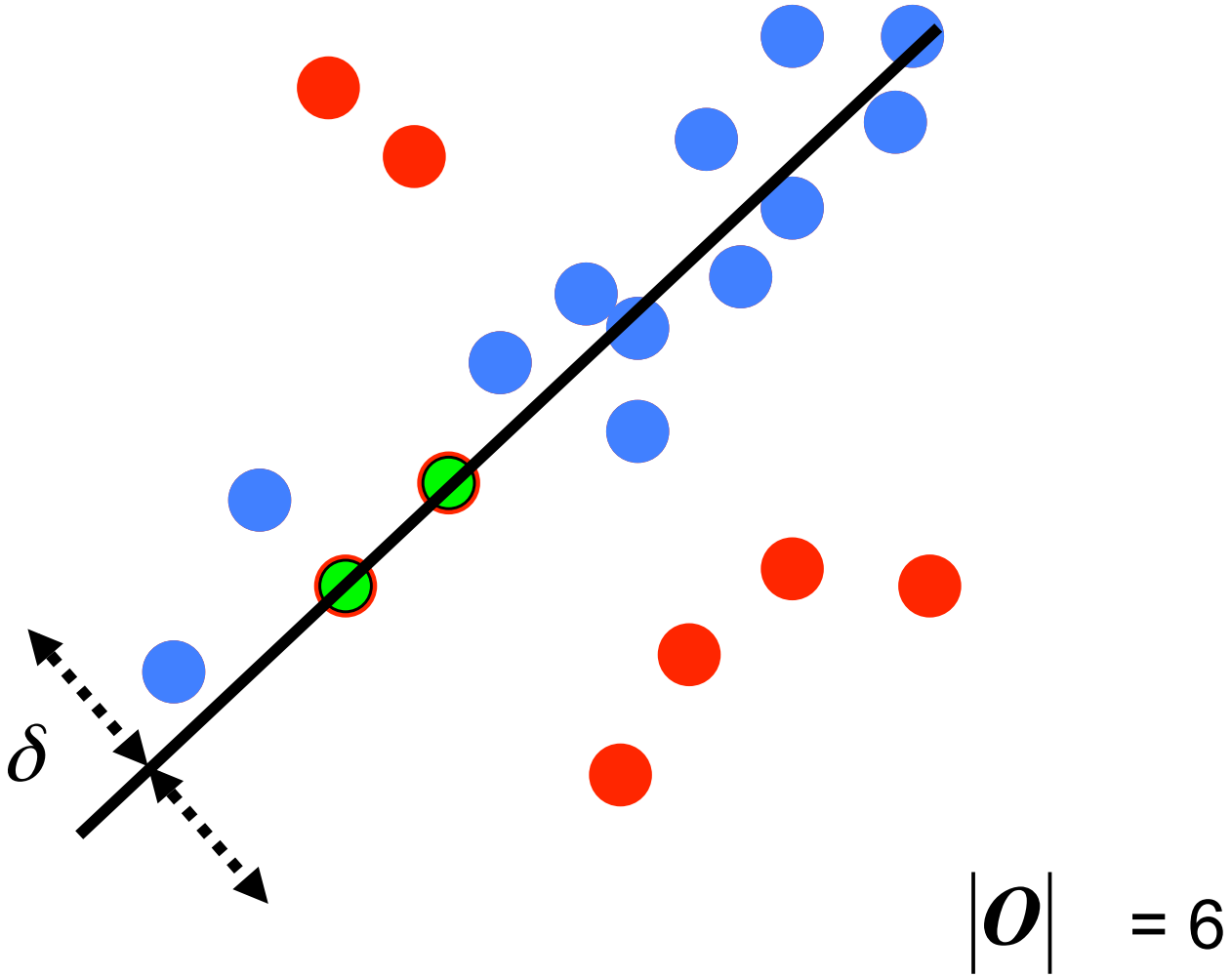
$|O| = 14$

Algorithm:

1. Select random sample of minimum required size to fit model [?] =[2]
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found

RANSAC!



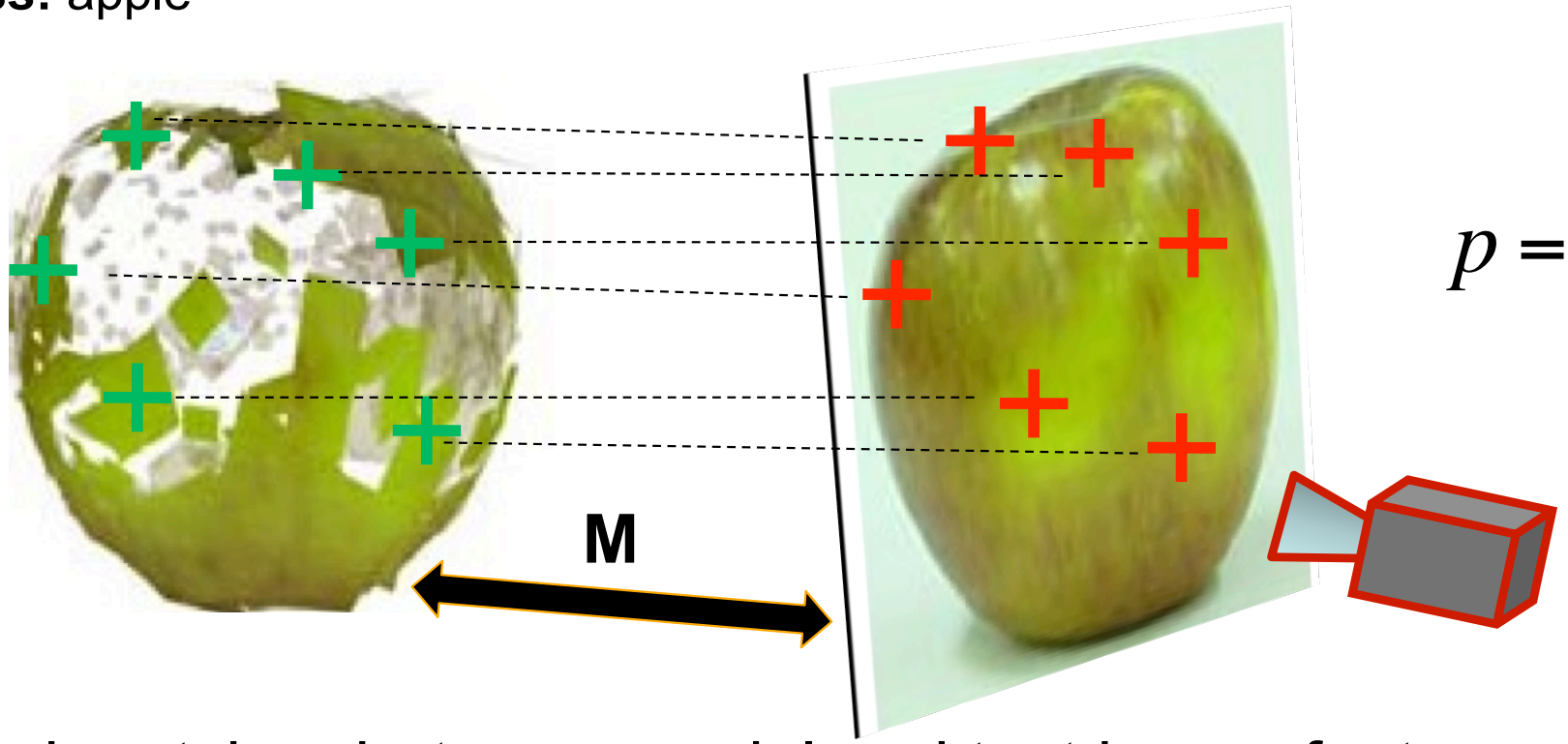
Algorithm:

1. Select random sample of minimum required size to fit model [?]
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found

Recognition

Class: apple

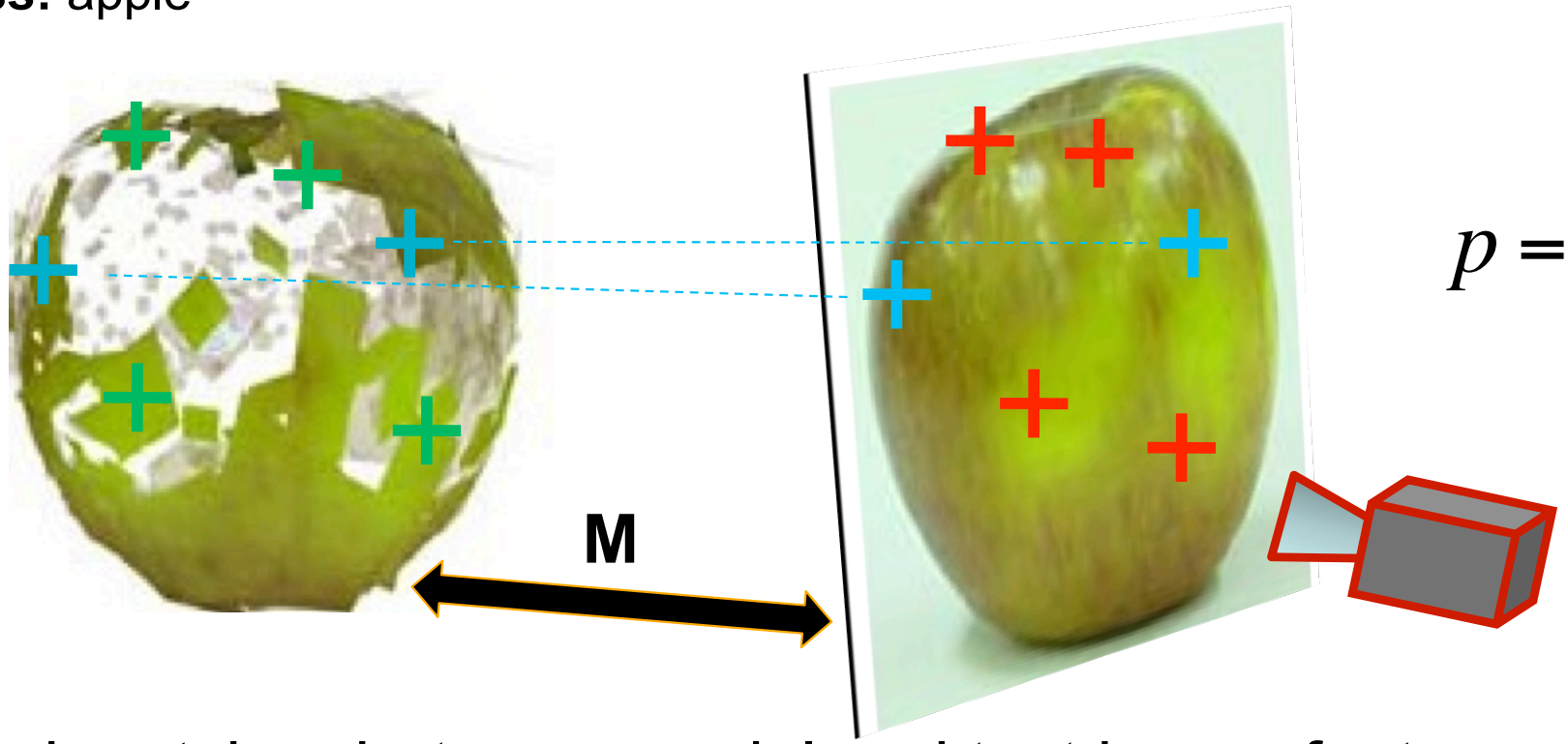


$$p = M\ P$$

1. Find matches between model and test image features
2. Generate hypothesis:
   - Compute transformation p = **M** P, from N matches

Recognition

Class: apple

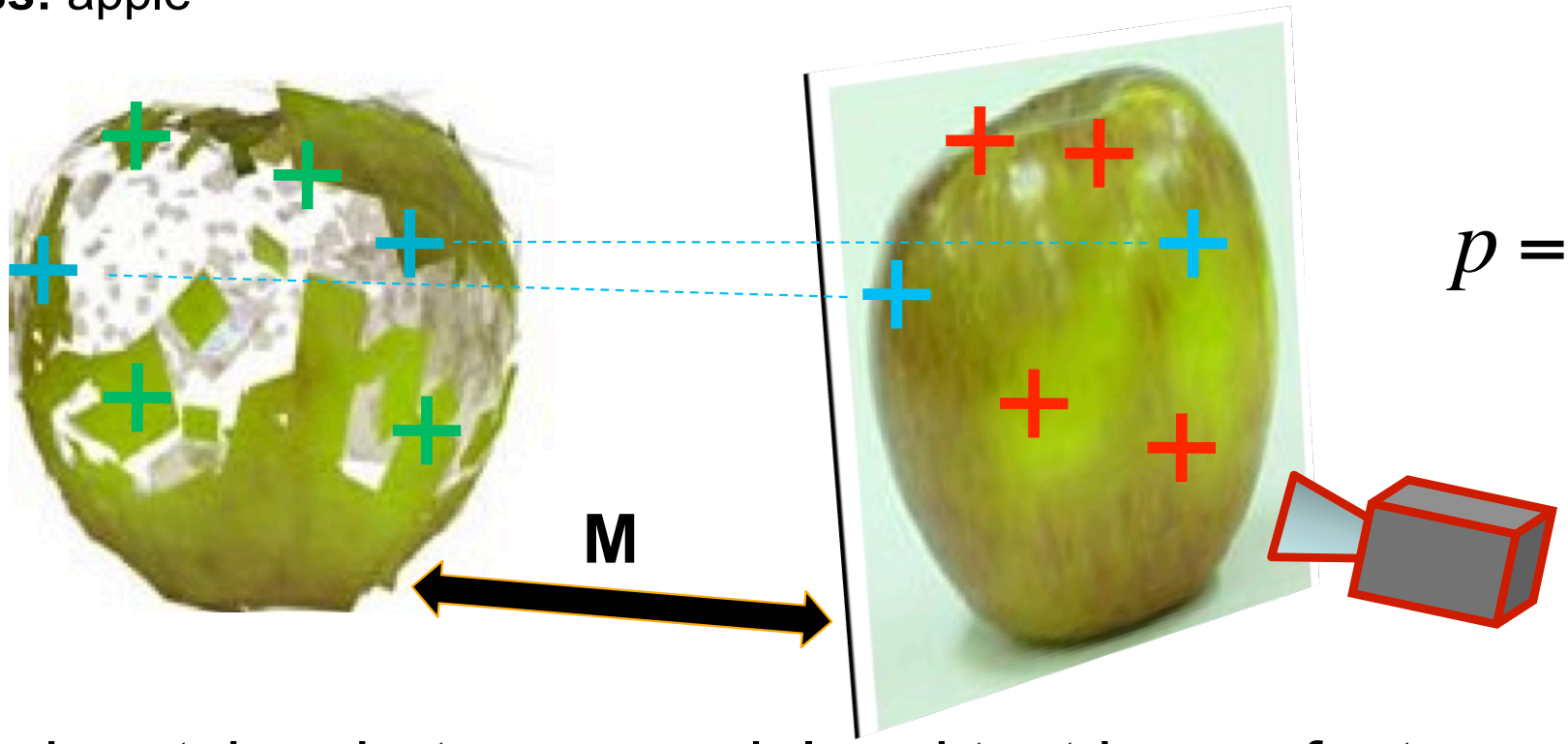


$$p = M\ P$$

1. Find matches between model and test image features
2. Generate hypothesis:
   - Compute transformation p = **M** P, from N matches (N=2, if affine camera & affine key points)

Recognition

Class: apple



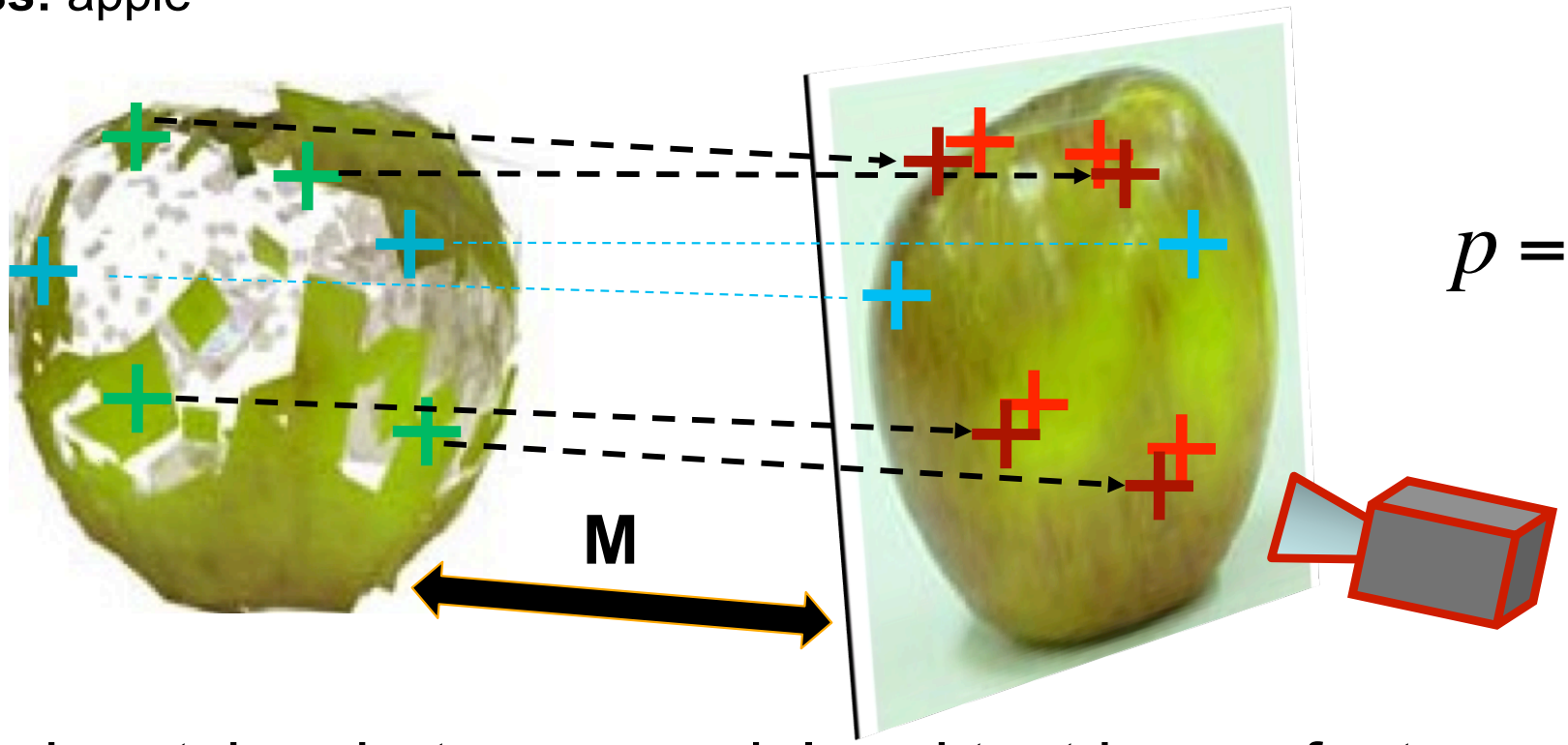
$$p = M\ P$$

1. Find matches between model and test image features
2. Generate hypothesis:
   - Compute transformation p = **M** P, from N matches (N=2, if affine camera & affine key points)

   → Generate hypothesis of object location and pose w.r.t. camera

Recognition

Class: apple



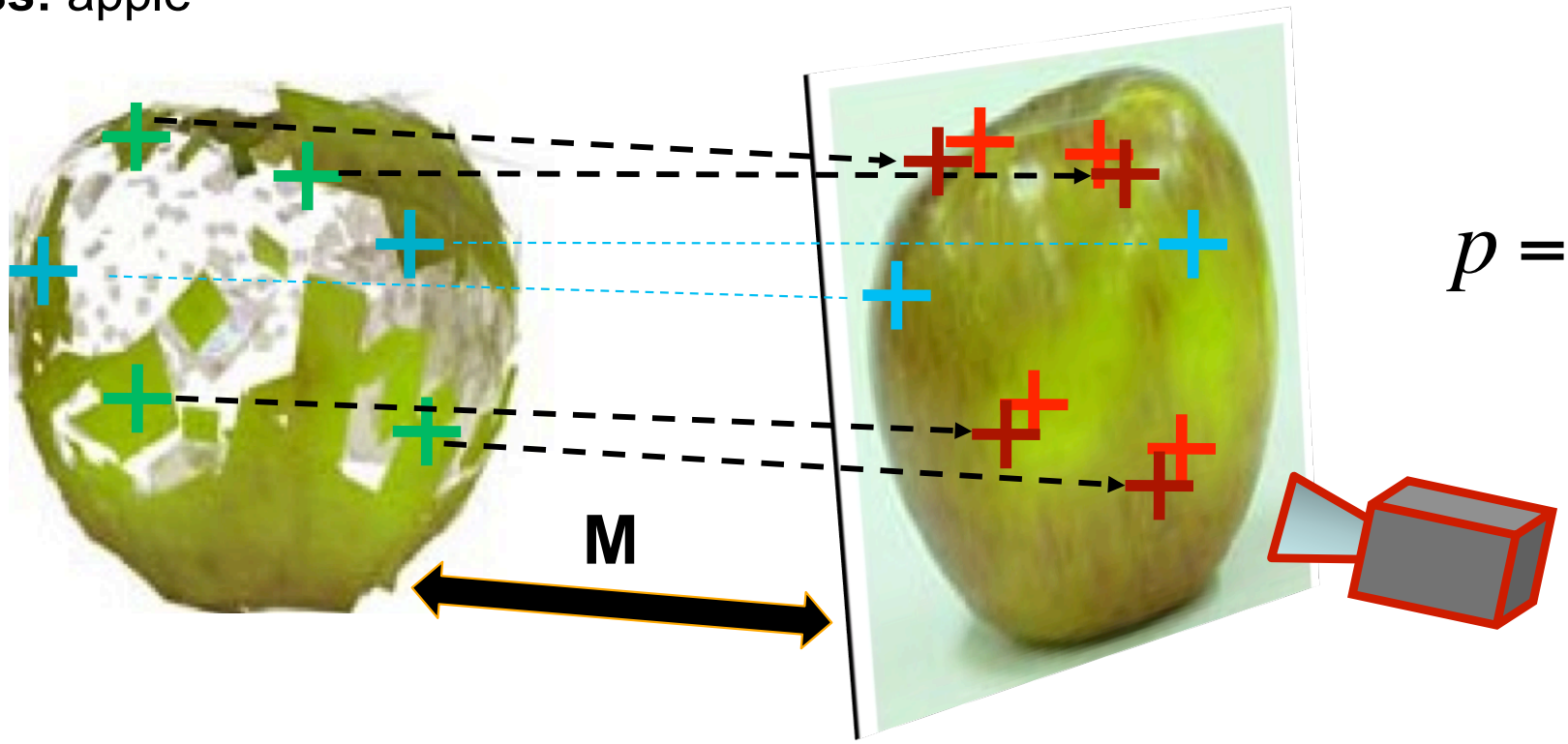
$$p = M\ P$$

1. Find matches between model and test image features
2. Generate hypothesis:
   - Compute transformation p = **M** P, from N matches (N=2, if affine camera & affine key points)

   → Generate hypothesis of object location and pose w.r.t. camera
3. Model verification
   - Use **M** to project other 3D model features into test image
   - Compute residual = D(projections, measurements)

Recognition

Class: apple



$$p = M\ P$$

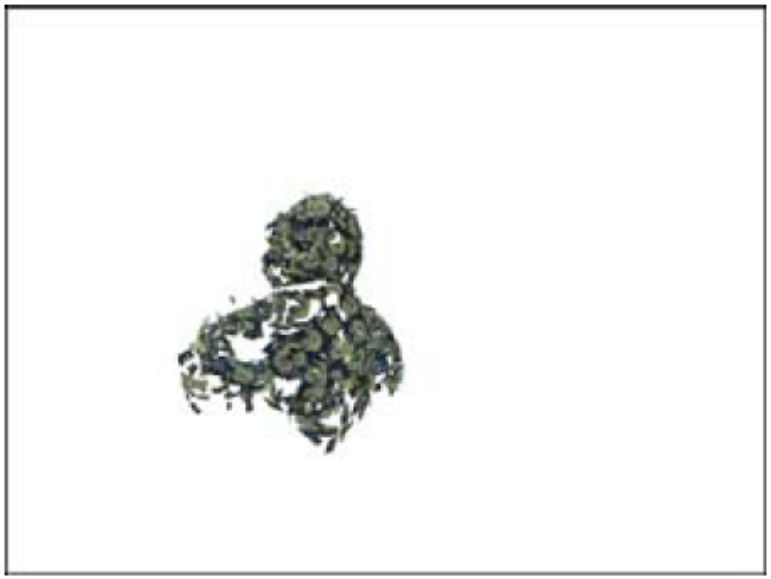
4. Repeat steps 2 and 3 until residual doesn't decrease anymore
5. Repeat steps 1-4 for different object instance C (apple, teddy bear, etc…)
6. M and C corresponding to min residual return the estimated object pose and object instance

Object to recognize

Initial matches based on appearance

Matches verified with geometrical constraints

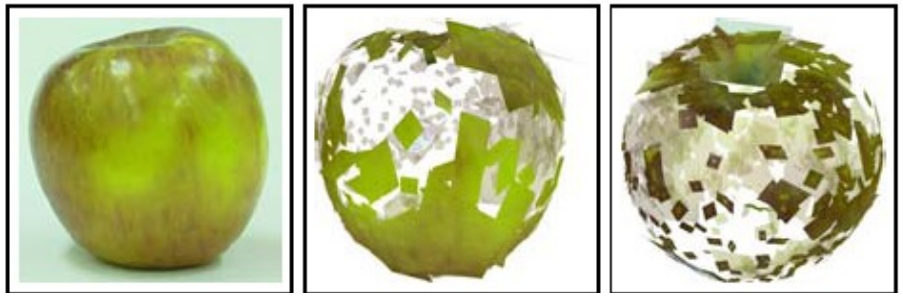
Recovered pose



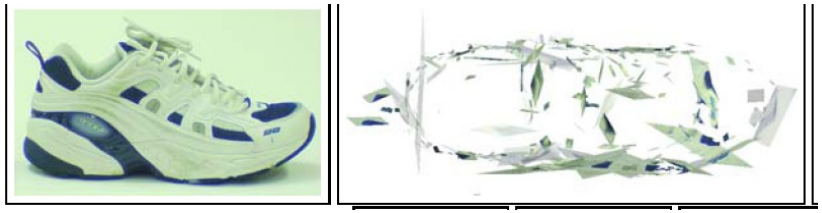
Courtesy of Rothganger et al

Detection and pose estimations results

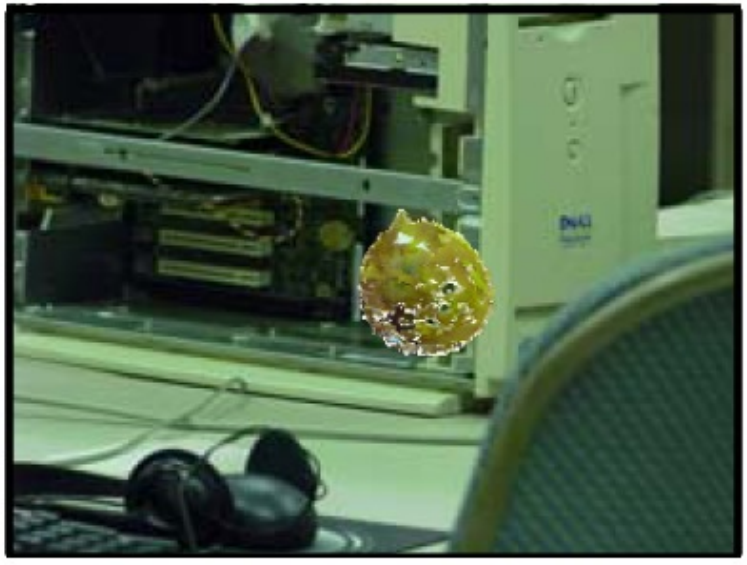
"apple" model



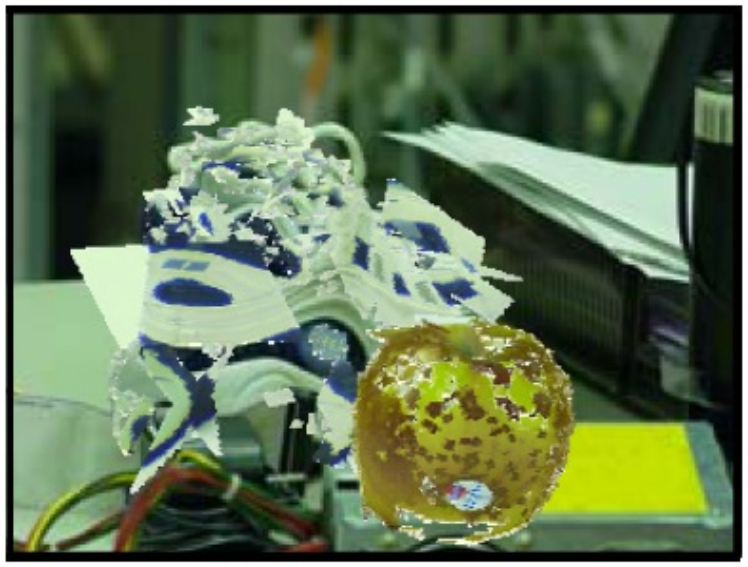
"shoe" model



Test image



Test image

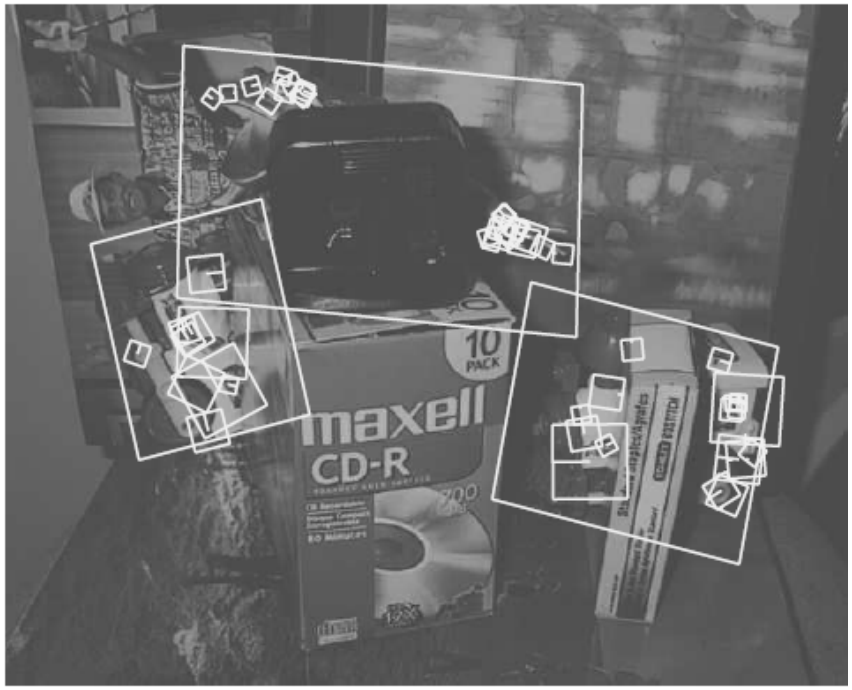


Courtesy of Rothganger et al

- Handle severe clutter

3D object detectors

Lowe. ’99, ’04



Courtesy of D. Lowe

- Handle severe occlusions
- Fast!

Detecting food in your kitchen!

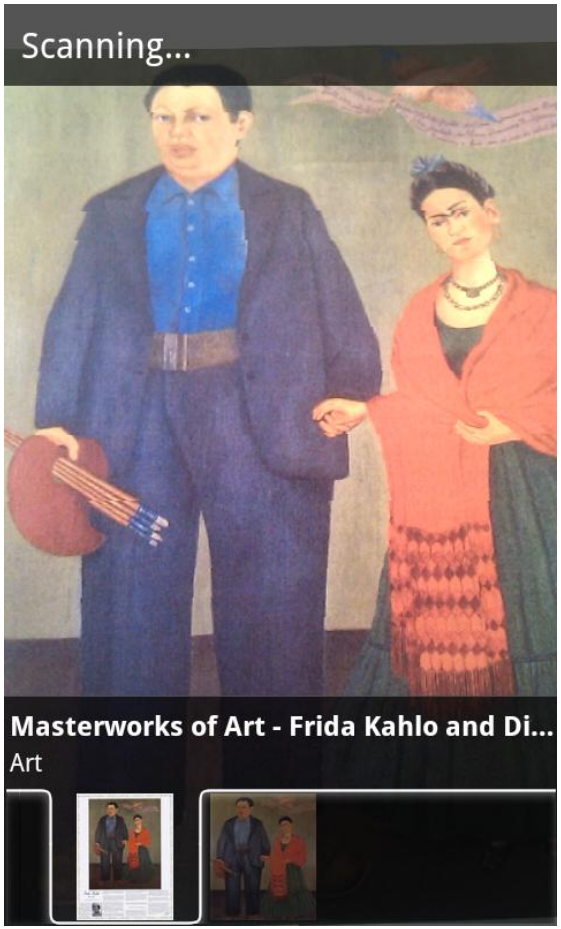
Edward Hsiao, Alvaro Collet and Martial Hebert. **Making specific features less discriminative to improve point-based 3D object recognition**. *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June, 2010.

Hsiao, Alvaro Collet and Martial Hebert, **Occlusion Reasoning for Object Detectionunder Arbitrary Viewpoint**, PAMI 2014

Detecting IKEA furniture!

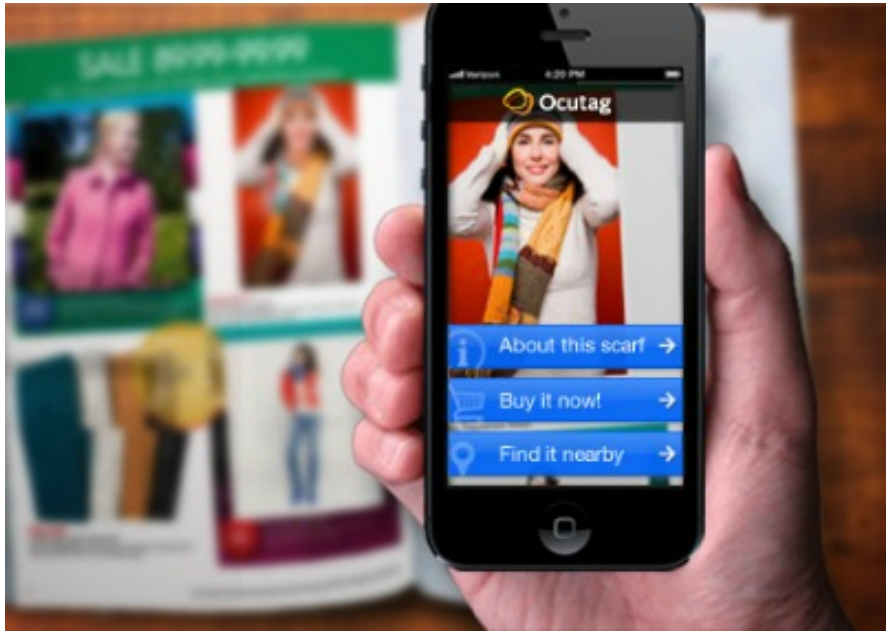
Parsing IKEA Objects: Fine Pose Estimation. Joseph Lim, Hamed Pirsiavash, and Antonio Torralba. International Conference on Computer Vision (ICCV), 2013.

Visual search and landmarks recognition



Google Goggles

Visual search and landmarks recognition



RICOH

A9

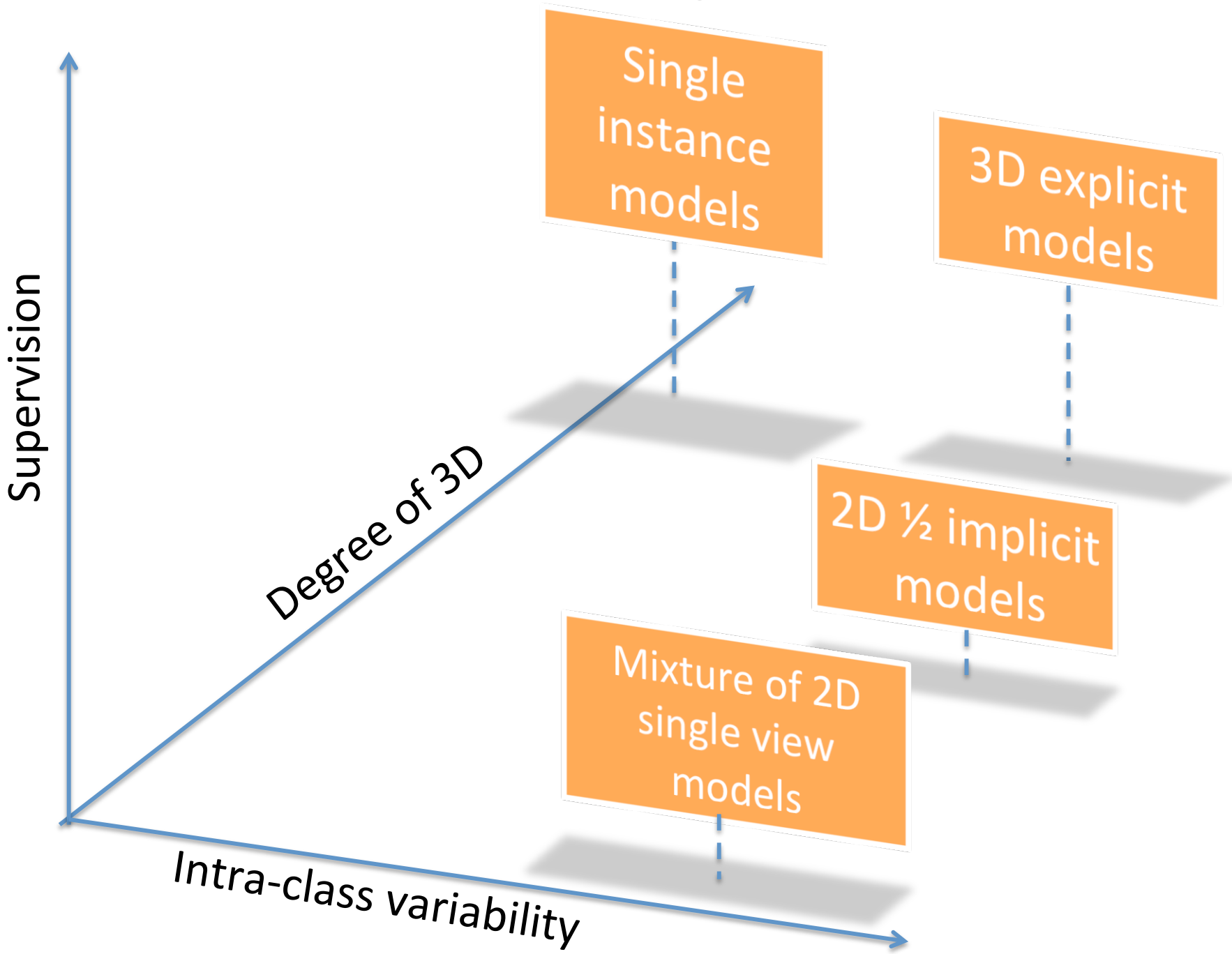
Limitations of single instance 3D object detectors

- Cannot handle intra-class variability.

Why?

- Models capture fine-grained details of the object instance which are not shared across instances in the same class
- Hypothesis-generation and verification scheme is not designed to maximize discrimination power

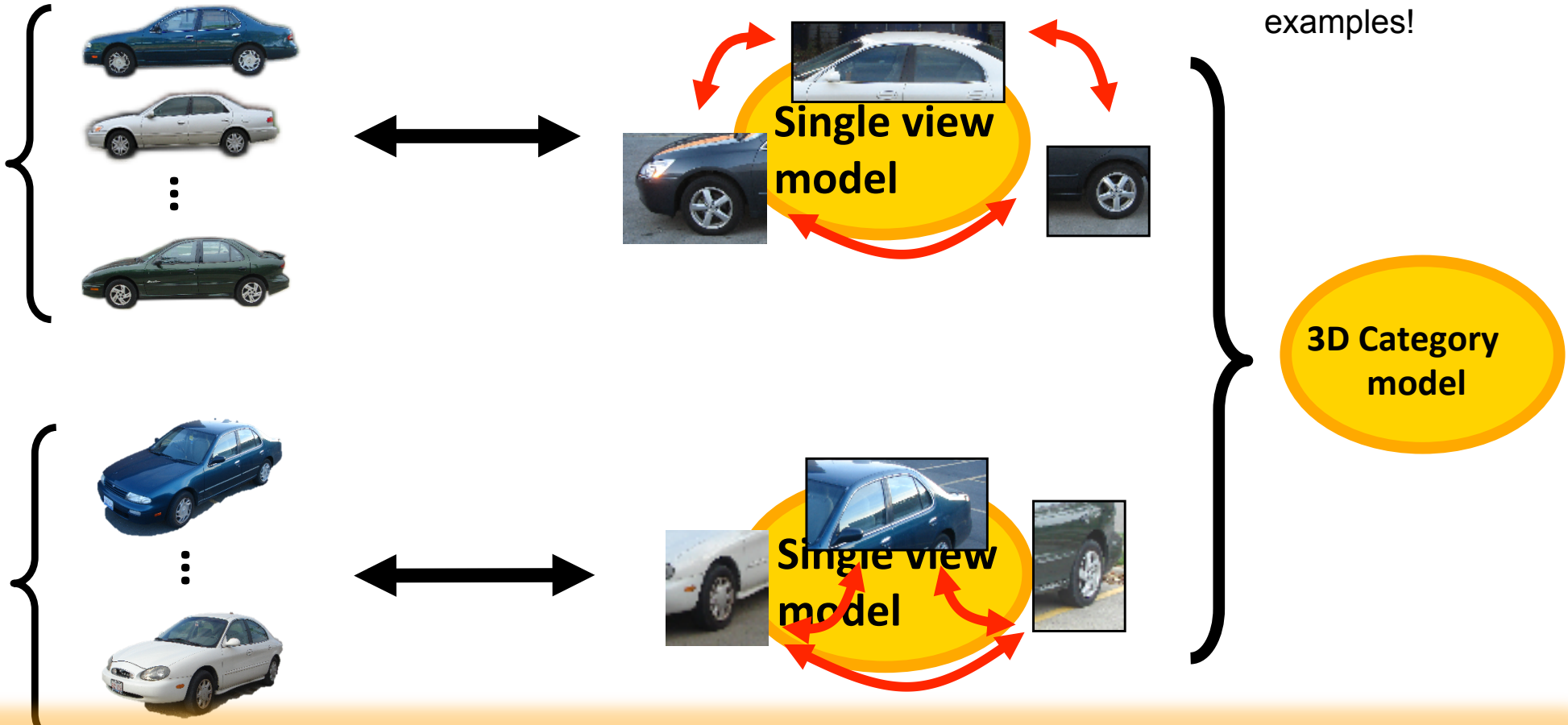
Models for 3d Object detection



Mixture of 2D models

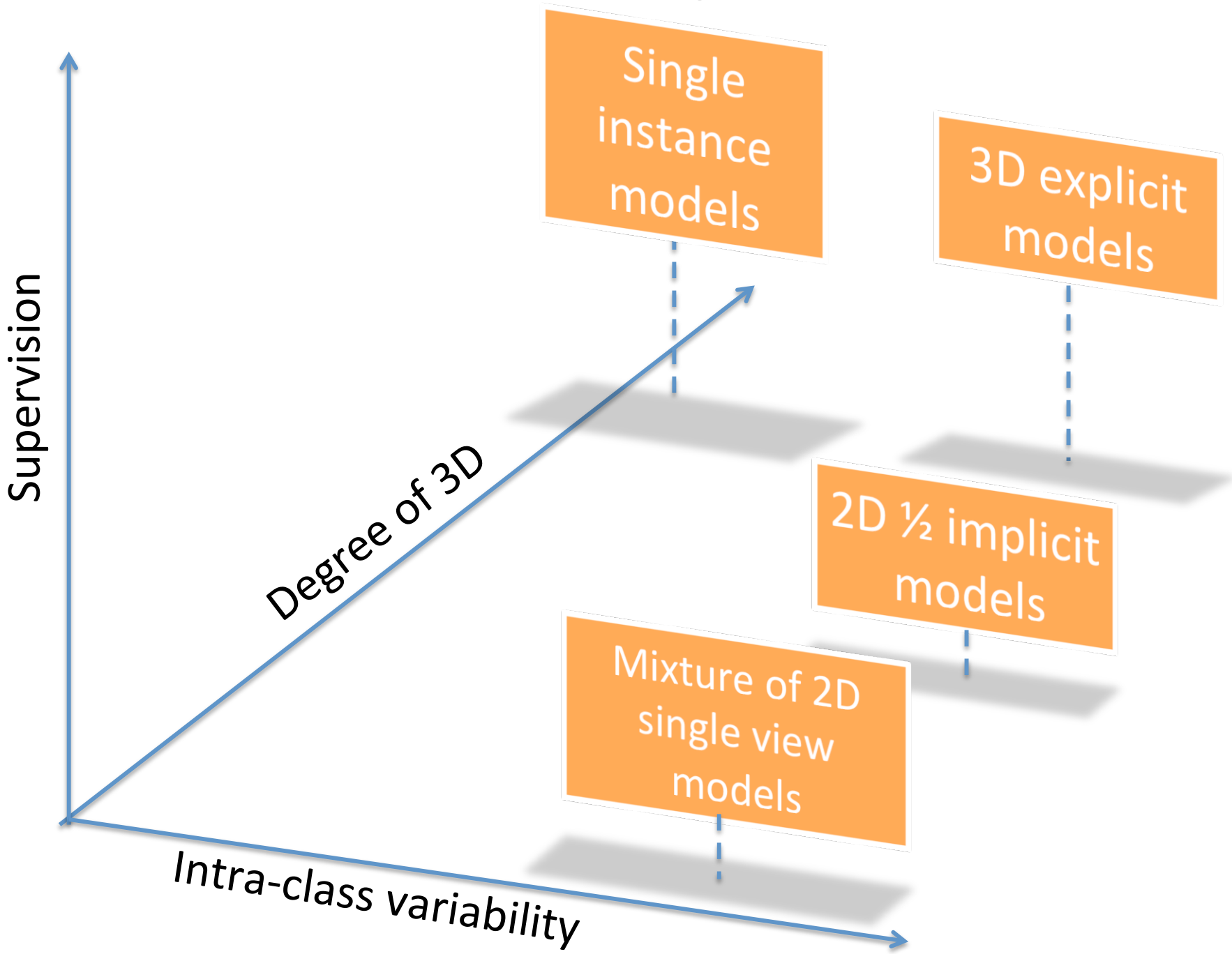
- Weber et al. ’00
- Schneiderman et al. ’01
- Ullman et al. 02
- Fergus et al. ’03
- Torralba et al. ’03
- Felzenszwalb & Huttenlocher ‘03
- Leibe et al. ’04
- Shotton et al. ‘05
- Grauman et al. ’05
- Savarese et al, ‘06
- Todorovic et al. ’06
- Vedaldi & Soatto ’08
- Zhu et al 08
- Gu & Ren, ‘10

DPM and ISM are examples!



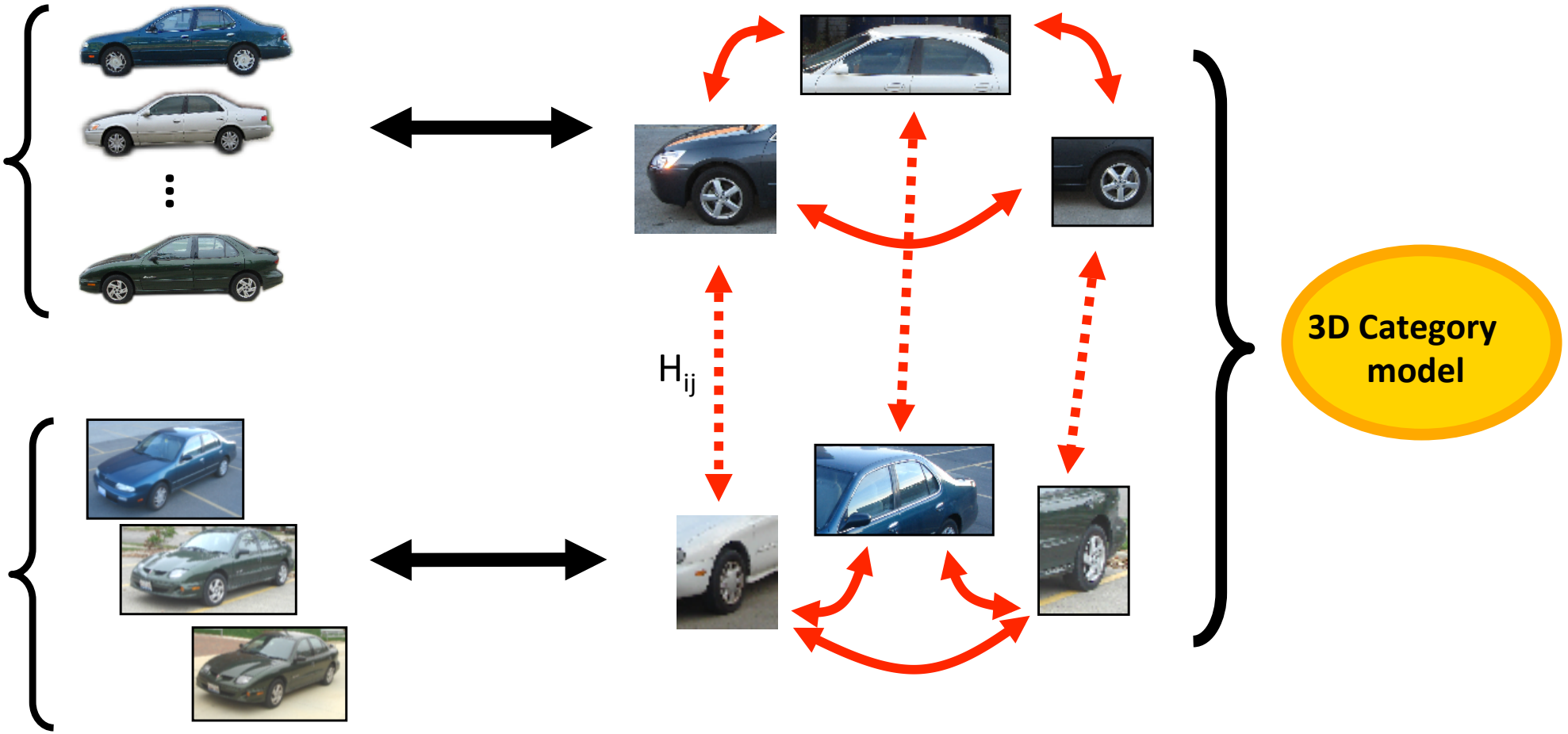
CONS: Single view models are independent • Non scalable to large number of categories/view-points • Just b. boxes • Cannot estimate 3D pose or 3D layout

Models for 3d Object detection



2D ½ implicit models

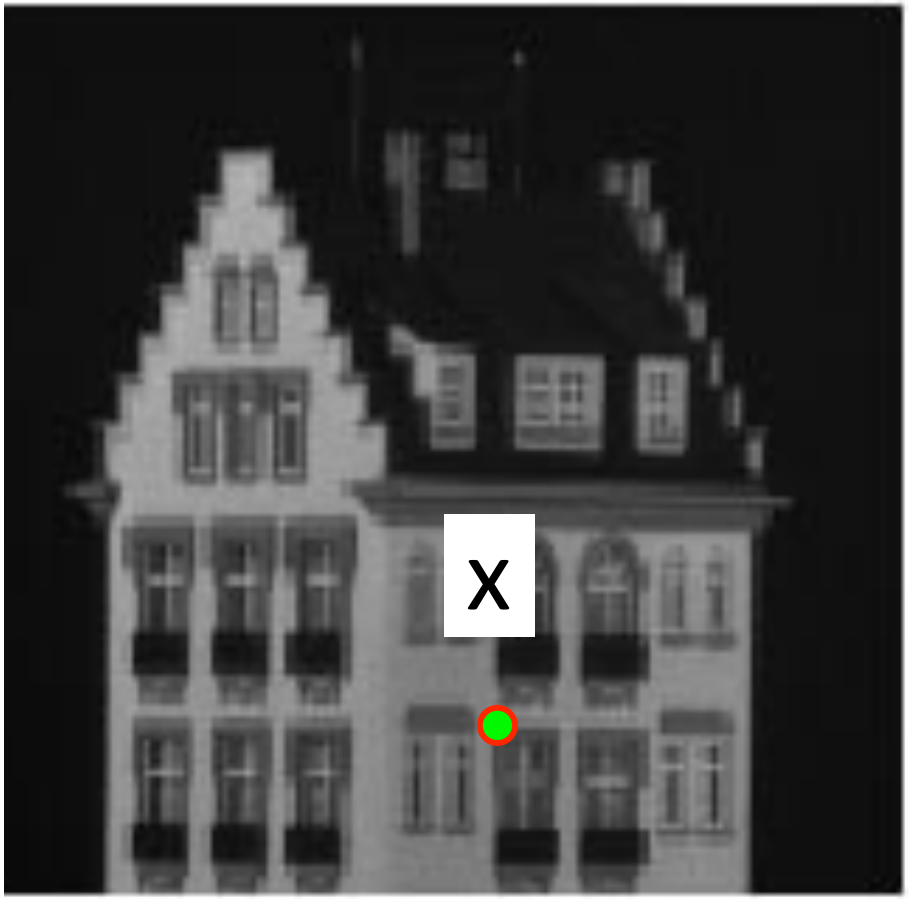
- Savarese & Fei-Fei, ICCV 07
- Savarese & Fei-Fei, ICCV 07
- Su, Sun, Fei-Fei, Savarese., CVPR 2009
- Sun, Su, Fei-Fei, Savarese, ICCV 2009
- Thomas et al. '06-09
- Kushal, et al., '07
- Farhadi '09
- Zhu et al. '09
- Ozuysal et al. '10
- Stark et al.'10
- Payet & Todorovic, 11
- Glasner et al., '11

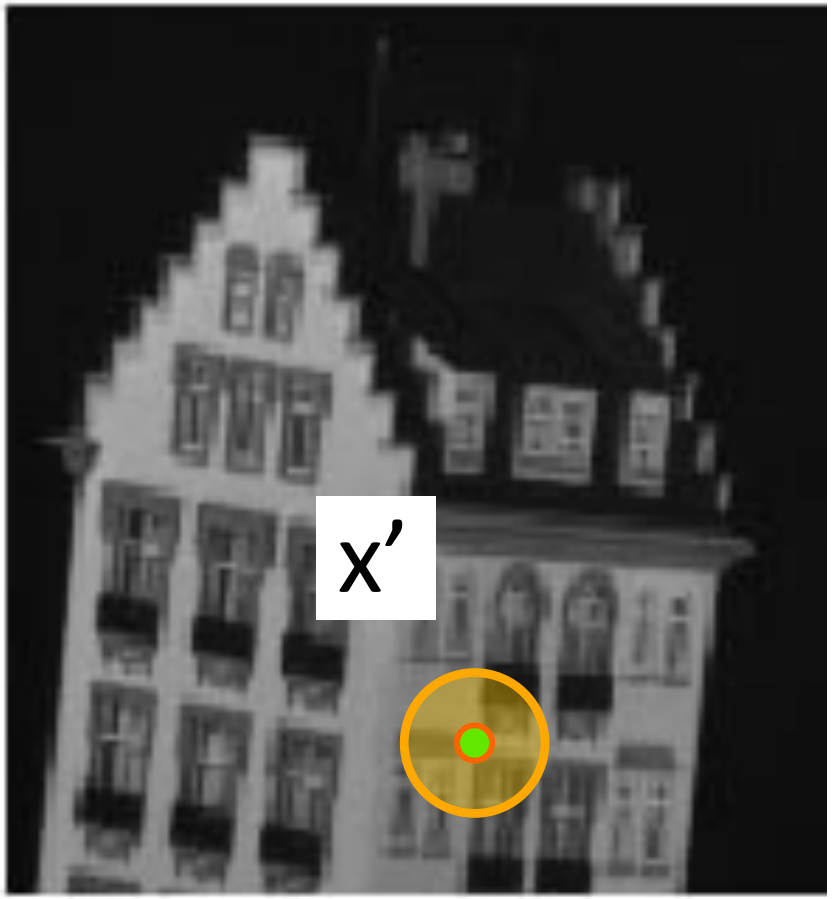


- Parts relationship can be probabilistic and learnt automatically

Linking features or parts across views:

Perspective or affine transformation constraints

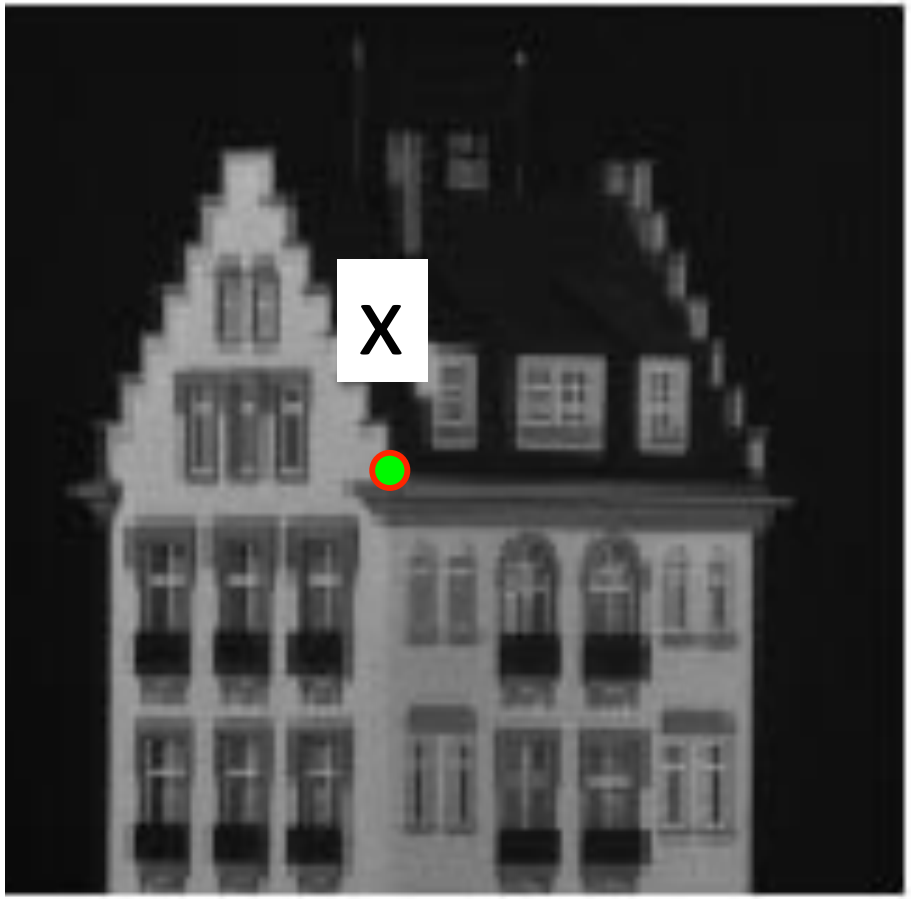


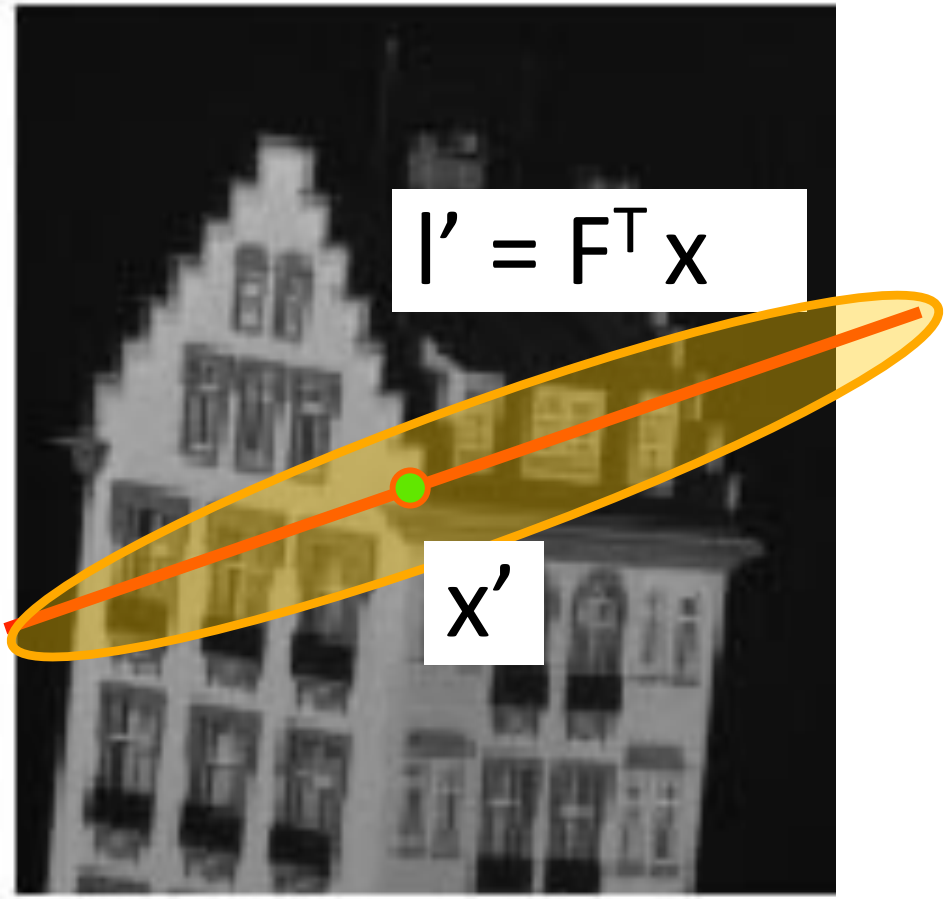


$$x' = H\,x$$

Linking features or parts across views:

Epipolar Transformation Constraints



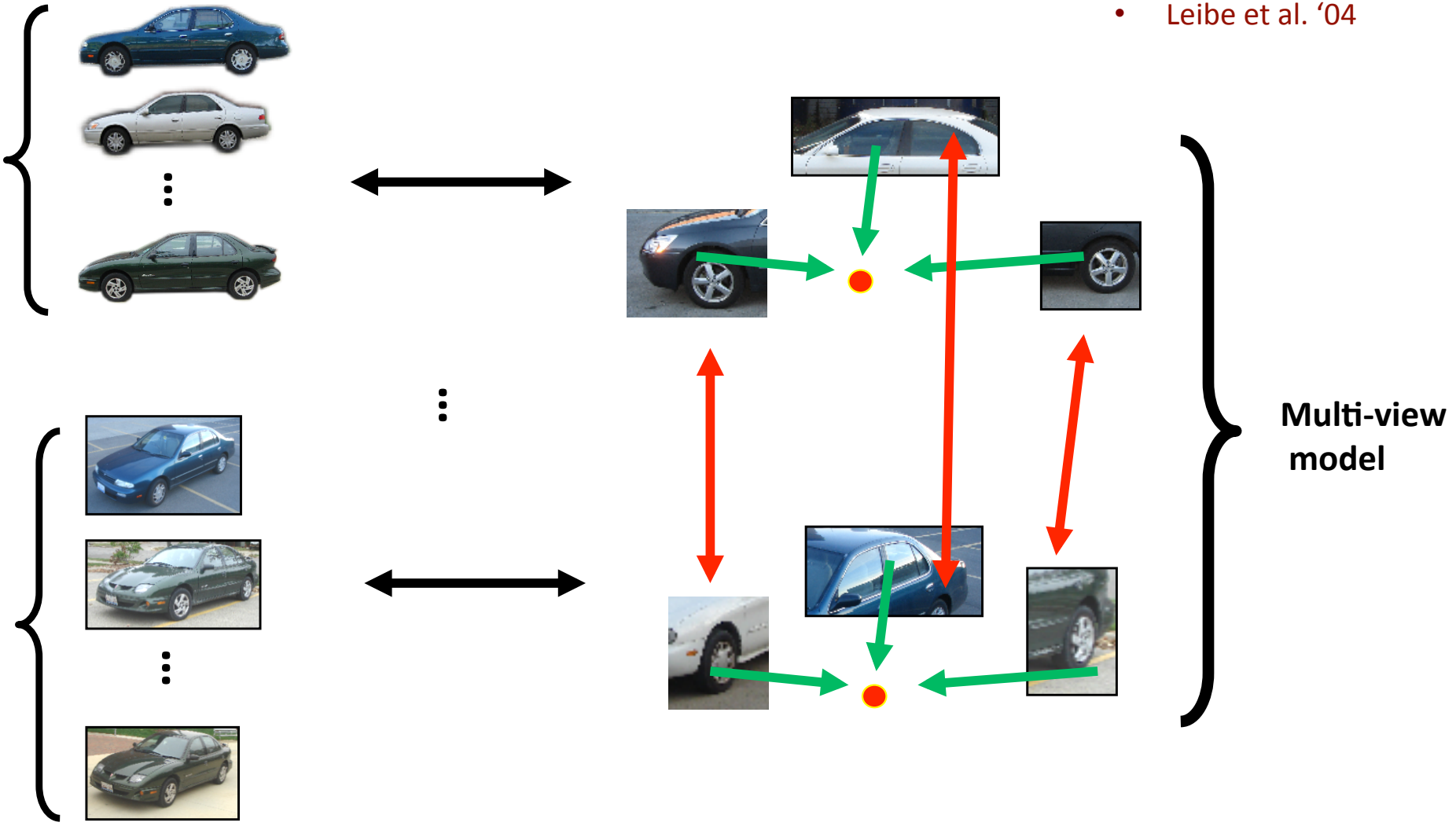


$$l' = F^T x$$

$$x' \in l'$$

Implicit 3D models – built upon the ISM

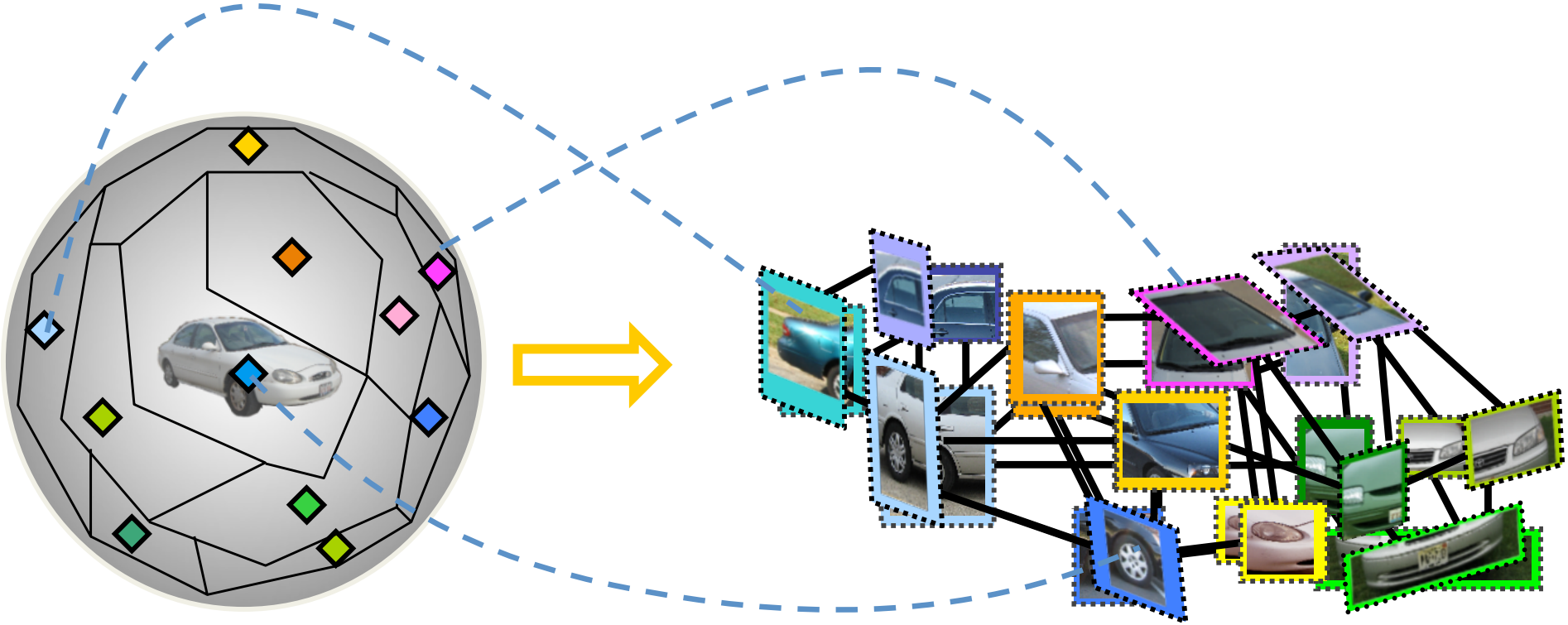
- Thomas et al. ’06
- Leibe et al. ‘04



- Sparse set of interest points or parts of the objects are linked across views.
- These links are used to transfer votes across views
- Each detected codeword votes for the object centroid within nearby views

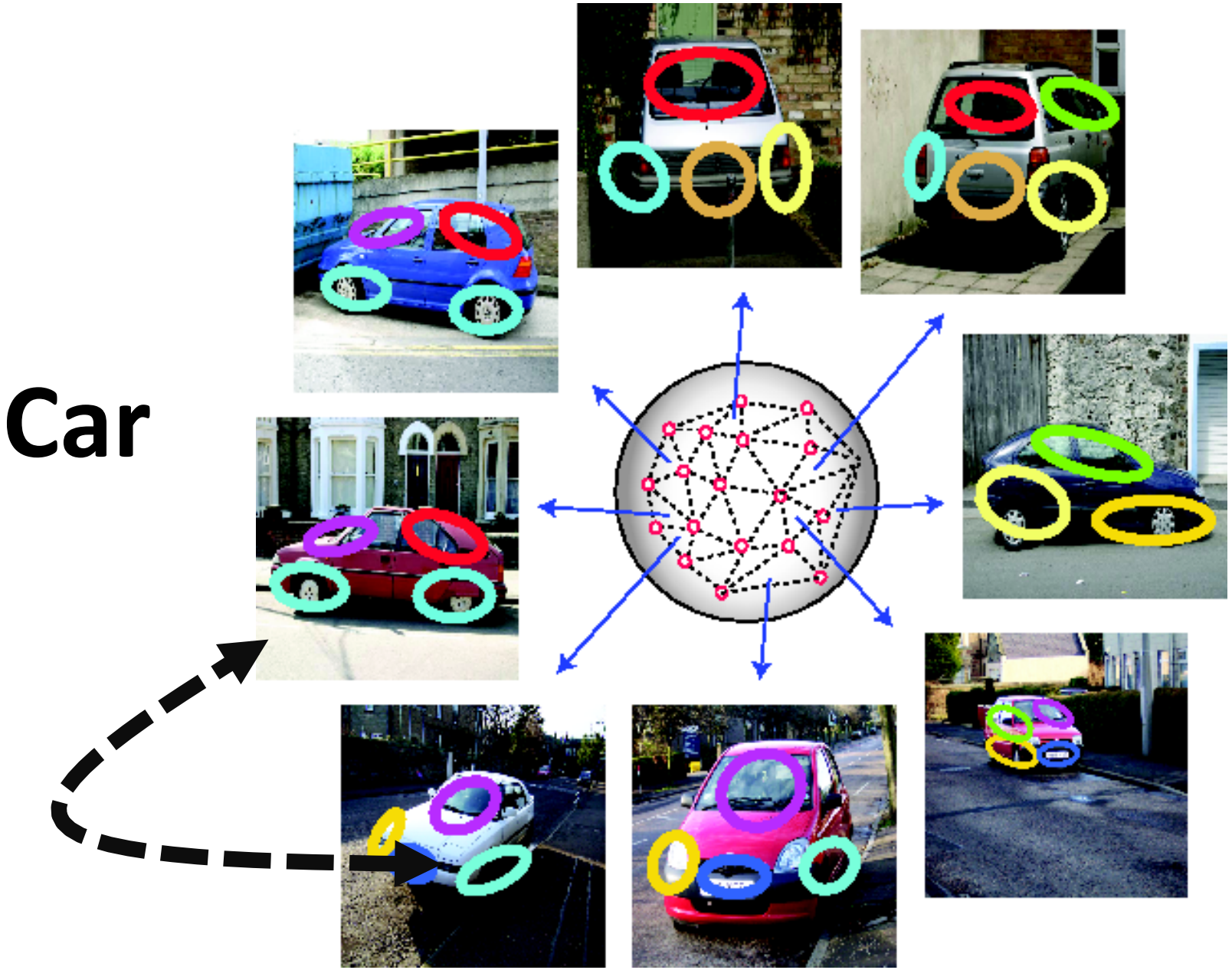
Implicit 3D models – graph-based representations

Savarese, Fei-Fei, ICCV 07
Sun, et al, CVPR 2009, ICCV 09



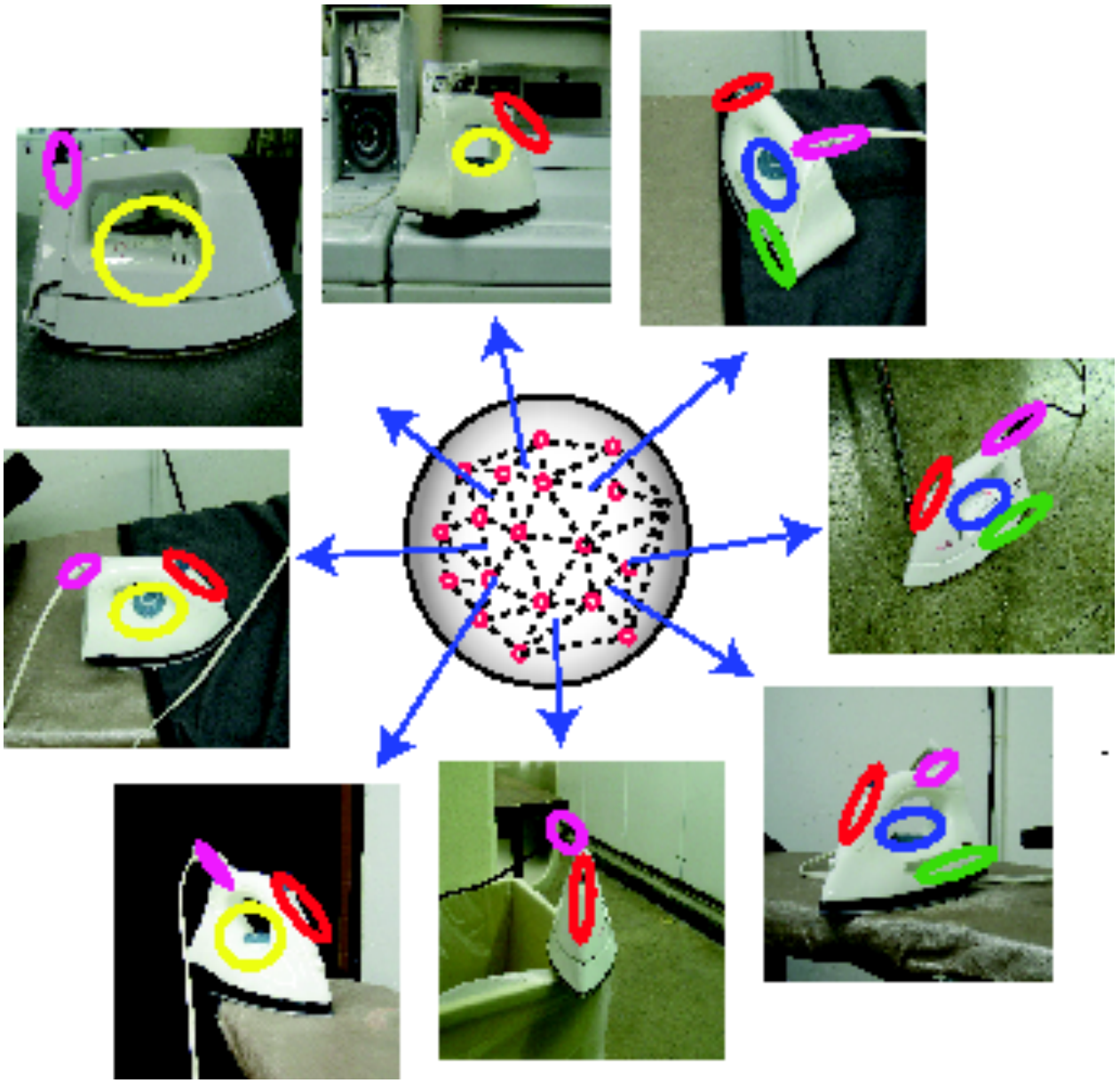
- Canonical parts captures view invariant diagnostic appearance information
- 2d ½ structure linking parts via weak geometry
- Parts and relationship are modeled in a probabilistic fashion
- Semi-supervised: only class labels, not view point or part annotations

Examples of learnt part-based models



Examples of learnt part-based models

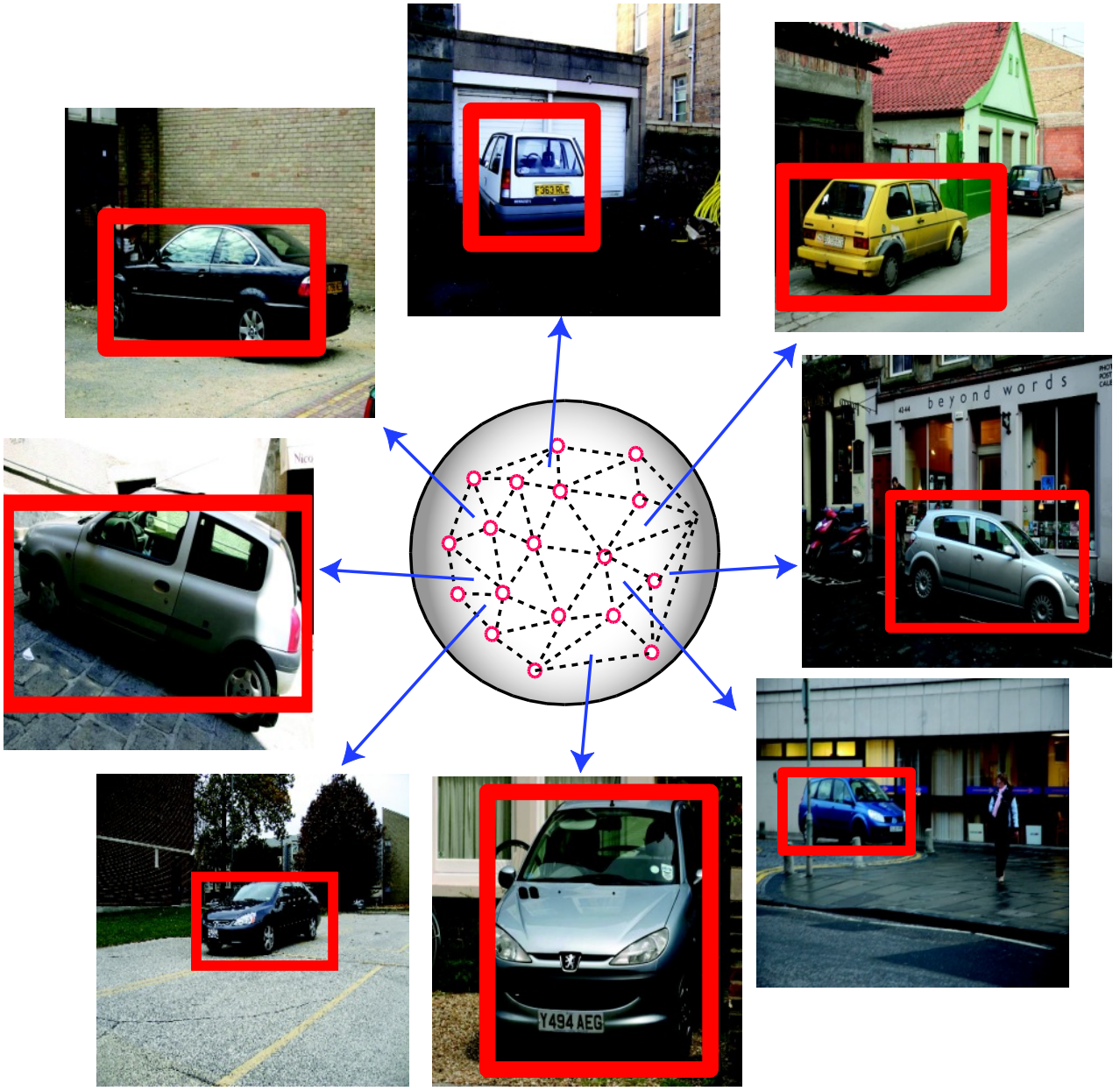
Travel iron



Experimental results

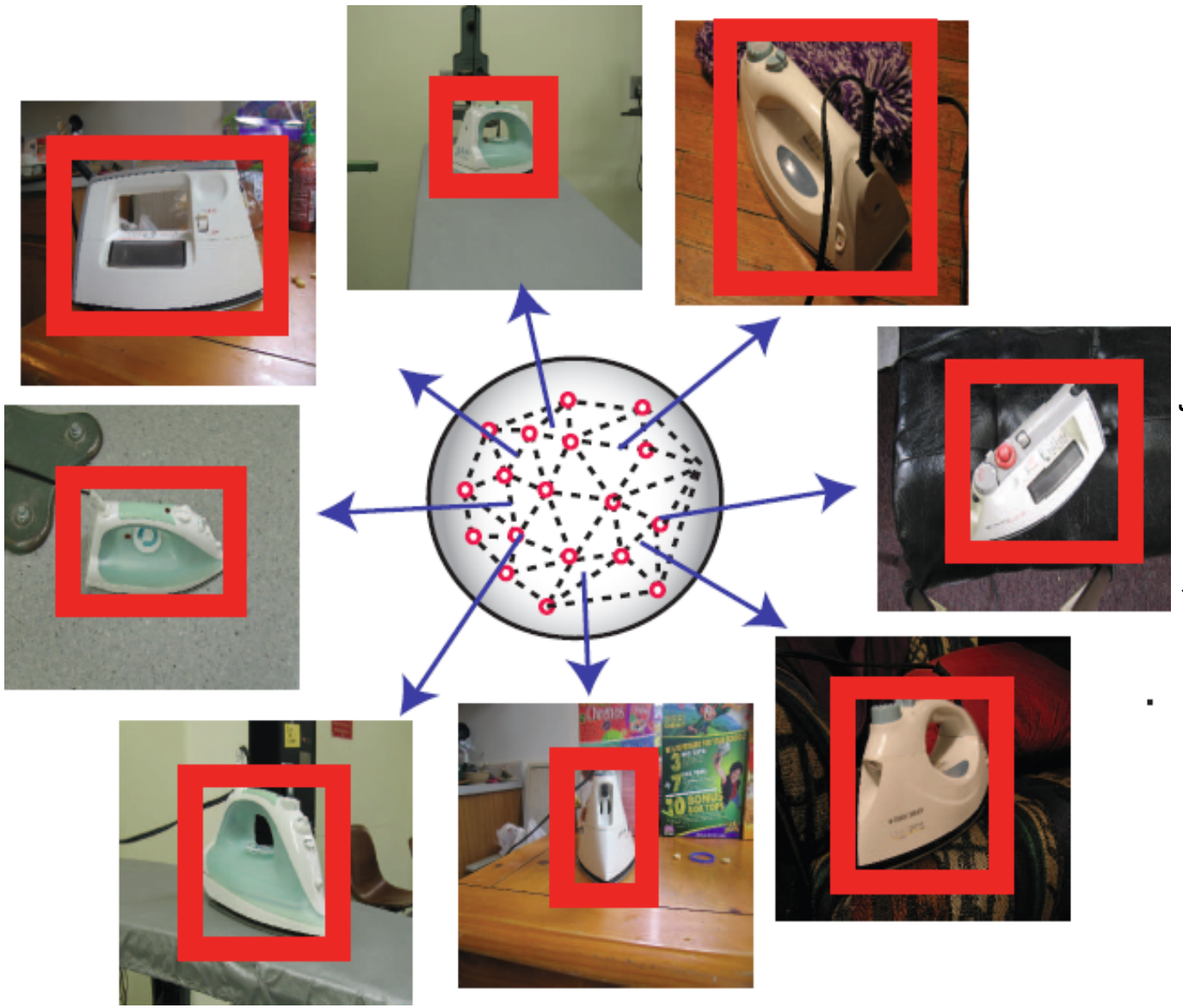
- Object detection from any viewing angles
- Accurate estimation of the object pose
- Synthesis of object appearance from unseen view points

Object detection and pose estimation



3D object dataset, 2007

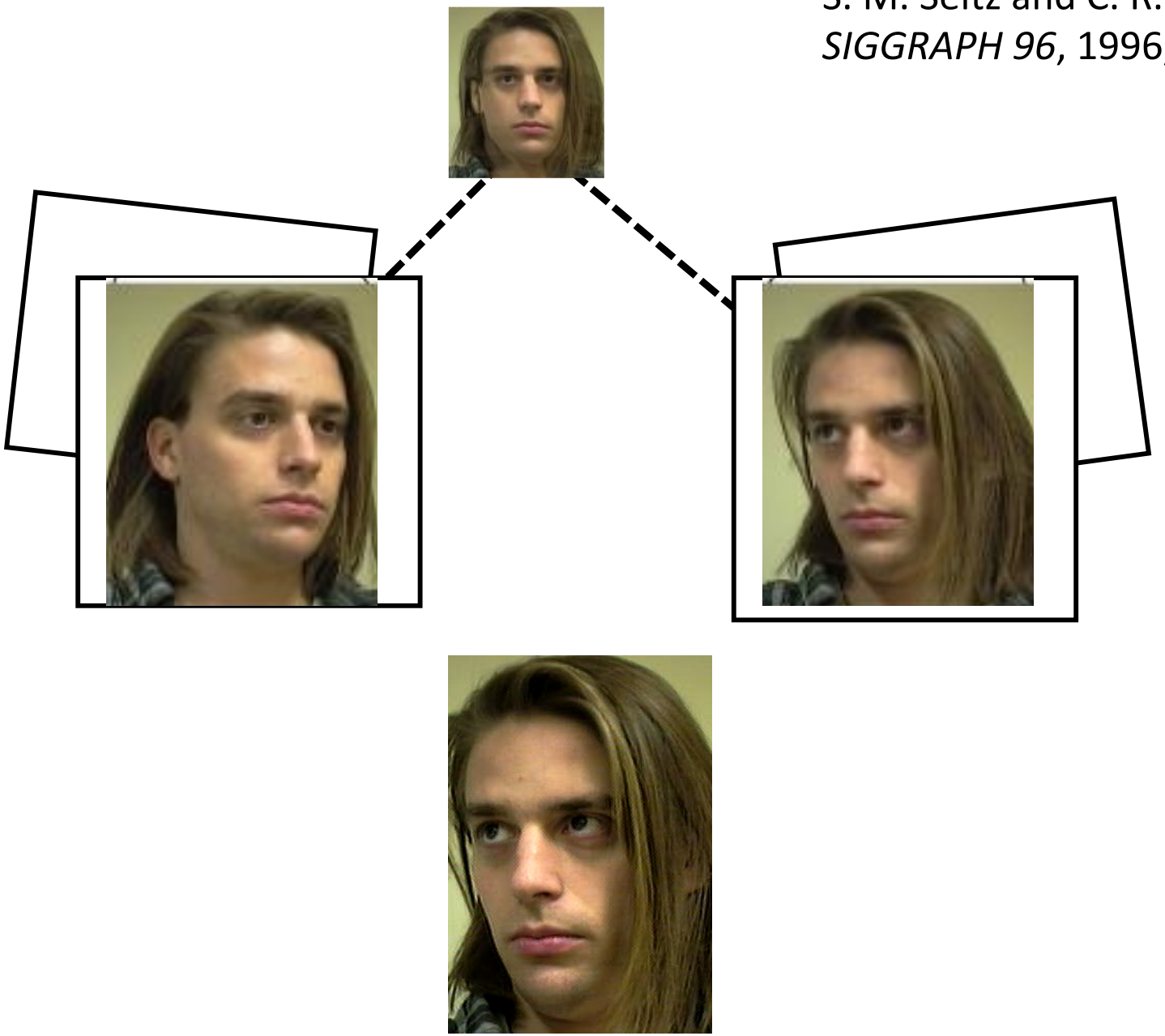
Object detection and pose estimation



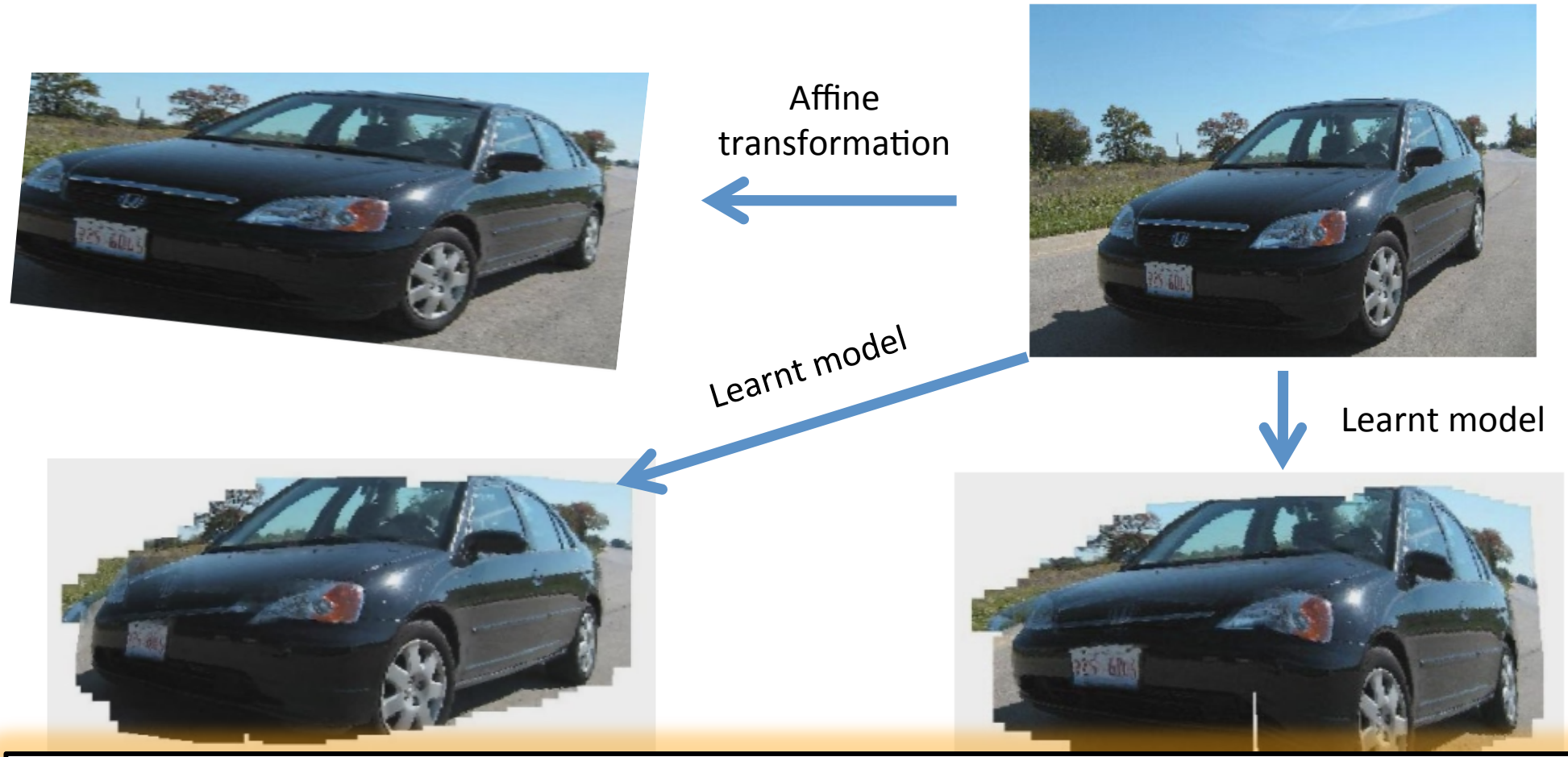
3D object dataset, 2007

Synthesizing novel views

S. M. Seitz and C. R. Dyer, *Proc. SIGGRAPH 96*, 1996, 21-30

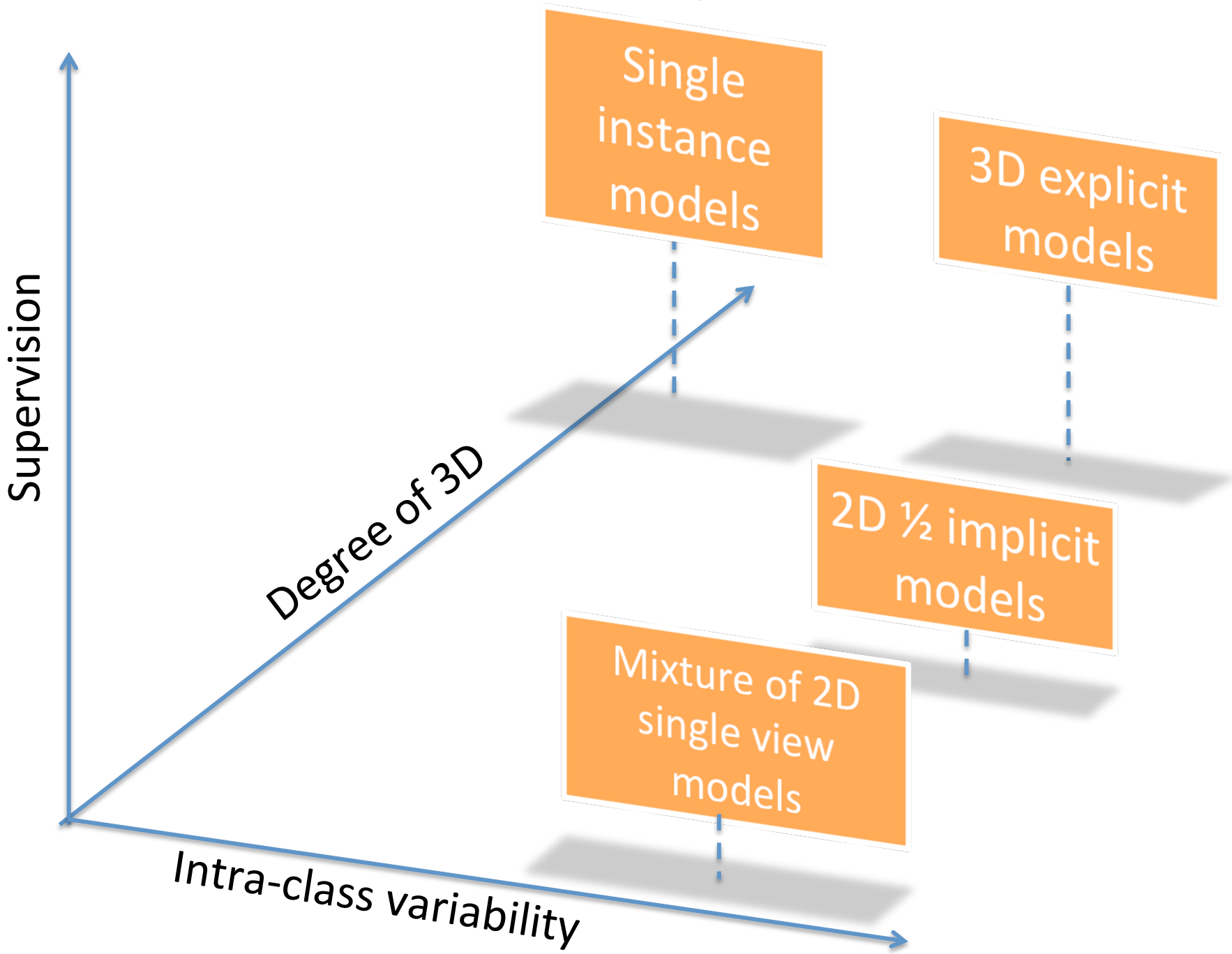


Predicting object appearance from novel views



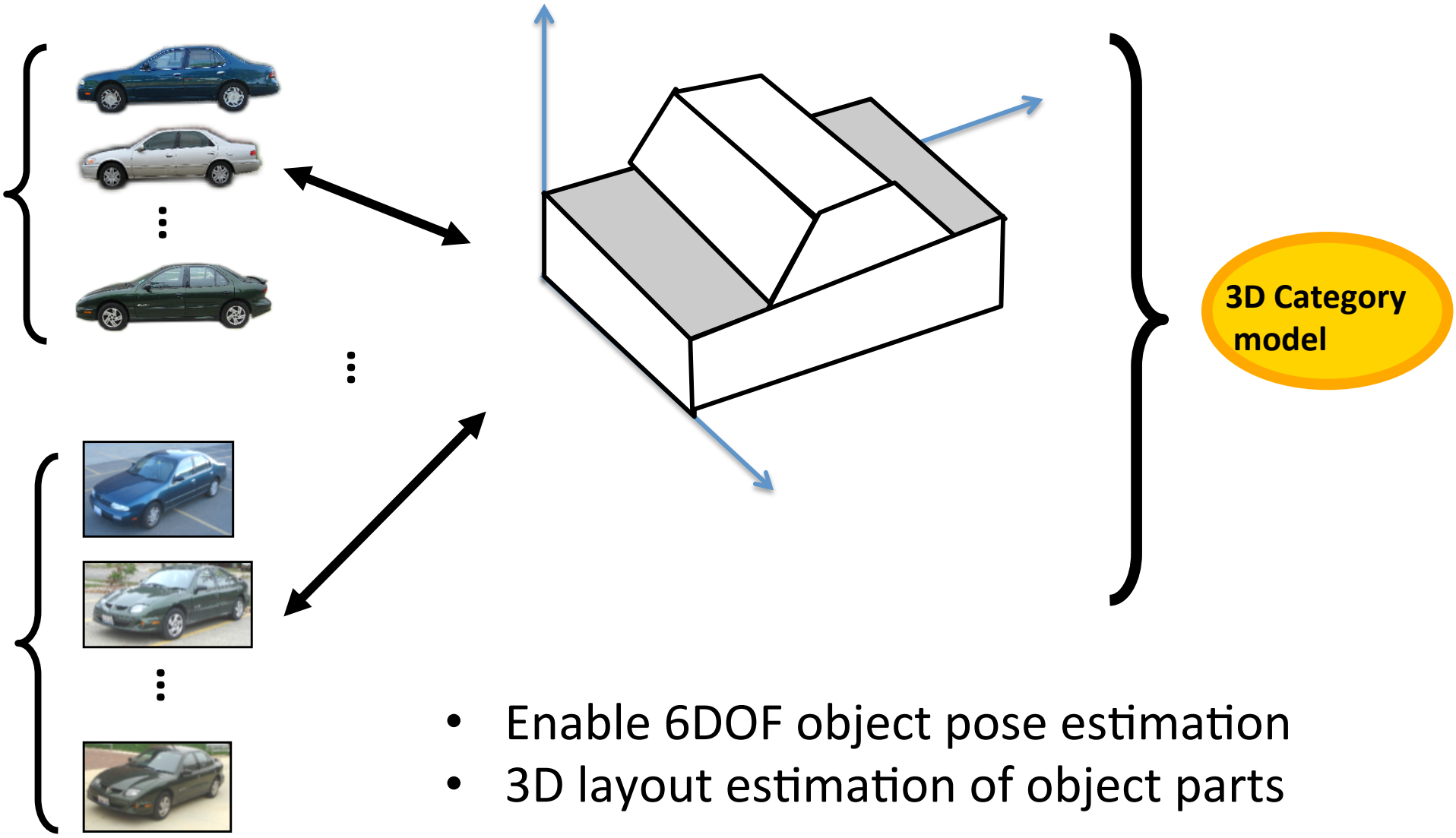
PROS: Flexible and easy to learn • Enable unsupervised discovery of parts
CONS: Limited accuracy • Unable to model part configurations in 3D

Models for 3d Object detection



3D explicit models

- Sun, Xu, Bradski, Savarese, ECCV 2010
- Sun, Kumar, Bradski, Savarese, 3DIM-PVT 2011
- Kumar, Sun, Savarese, CVPR 12
- Xiang & Savarese, CVPR 12
- Hoiem, et al. , '07
- Chiu et al . '07
- Liebelt et al. '08, 10
- Xiao et al . '08
- Yi et al. 09
- Arie-Nachimson & Barsi '09
- Sandhu et al . '09
- Hu & Zhu '10



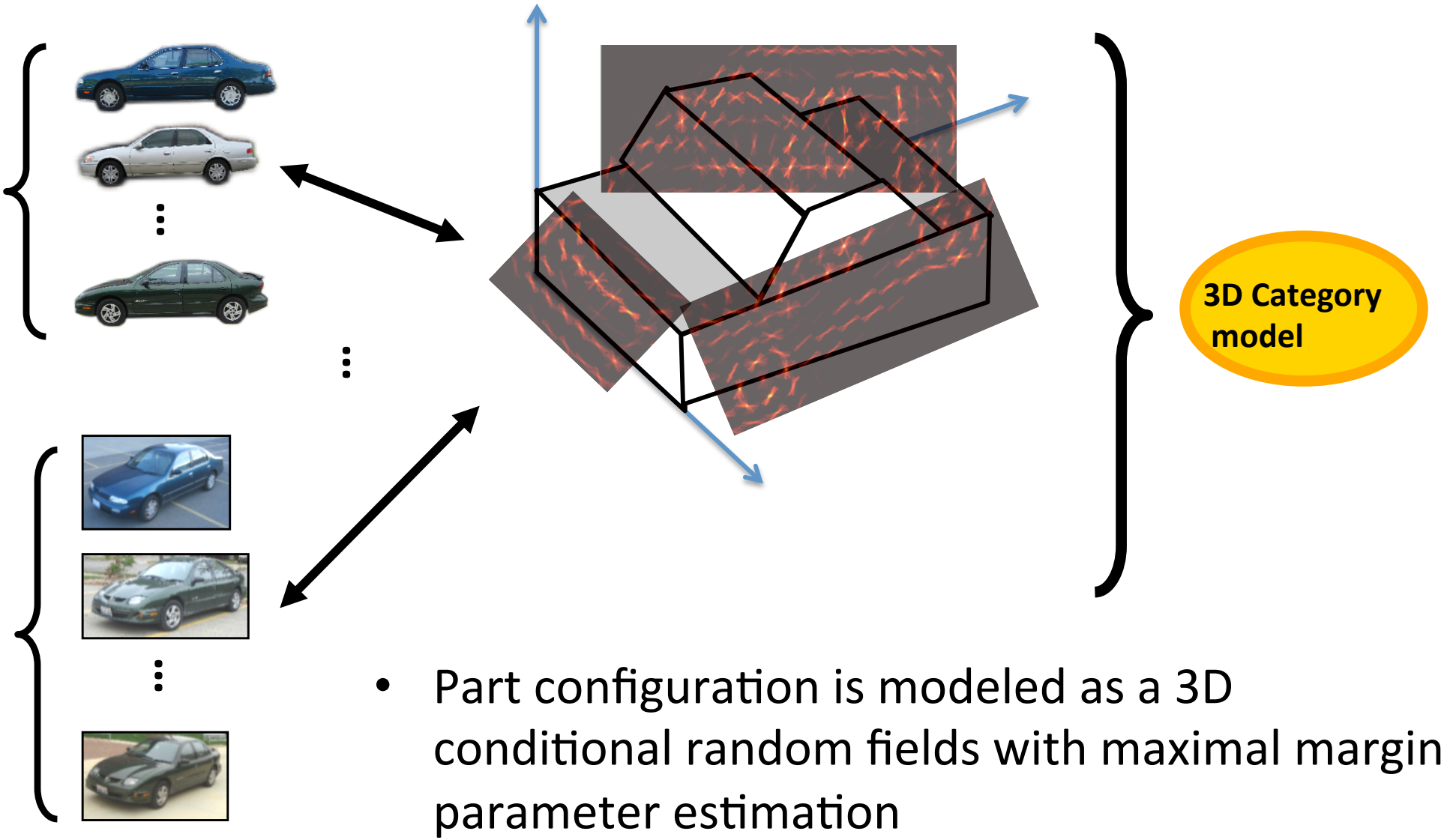
- Enable 6DOF object pose estimation
- 3D layout estimation of object parts

3D explicit models

Yan, et al. ’07

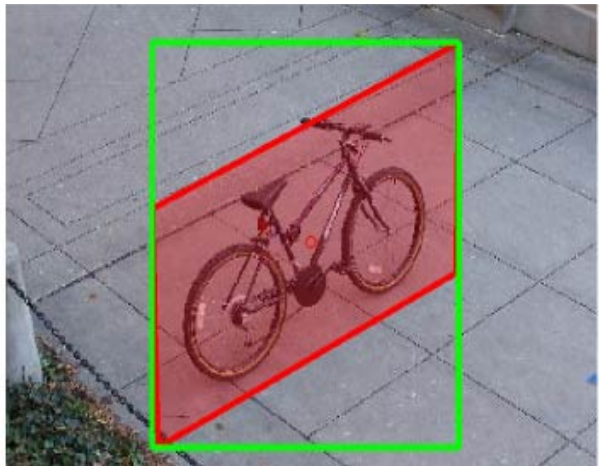
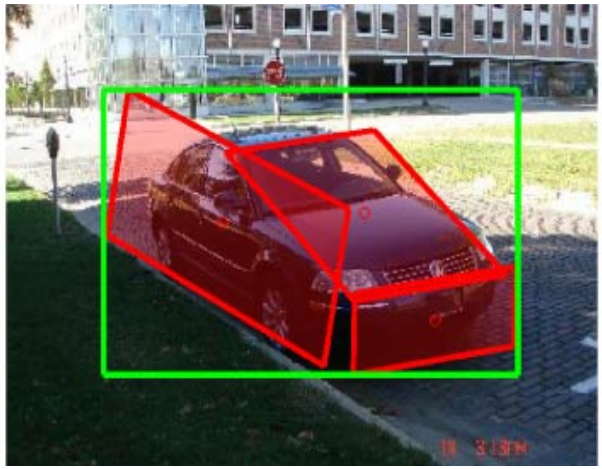
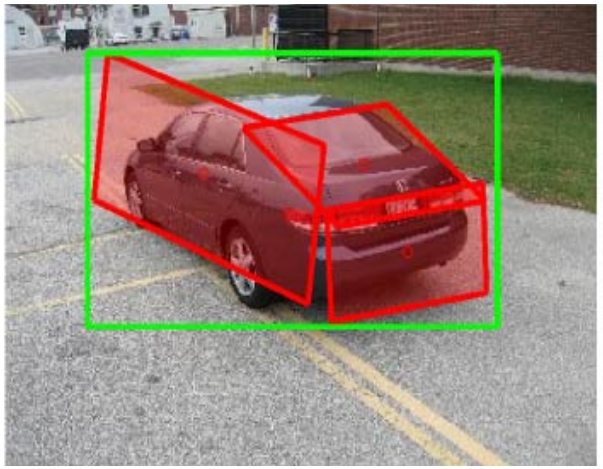
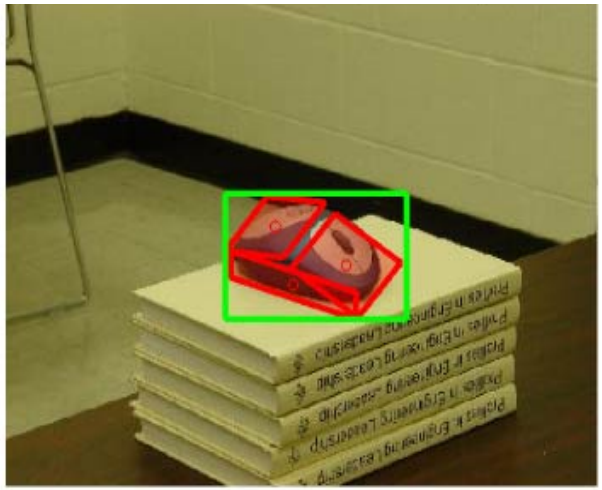
3D explicit models

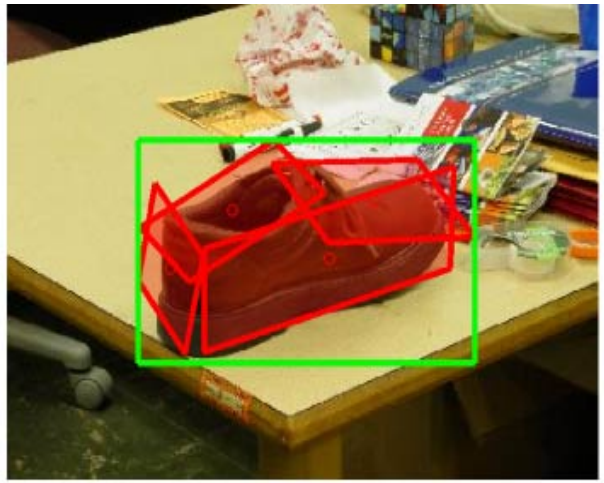
Xiang & Savarese, 2012
Pepik et al. 2013



- Part configuration is modeled as a 3D conditional random fields with maximal margin parameter estimation

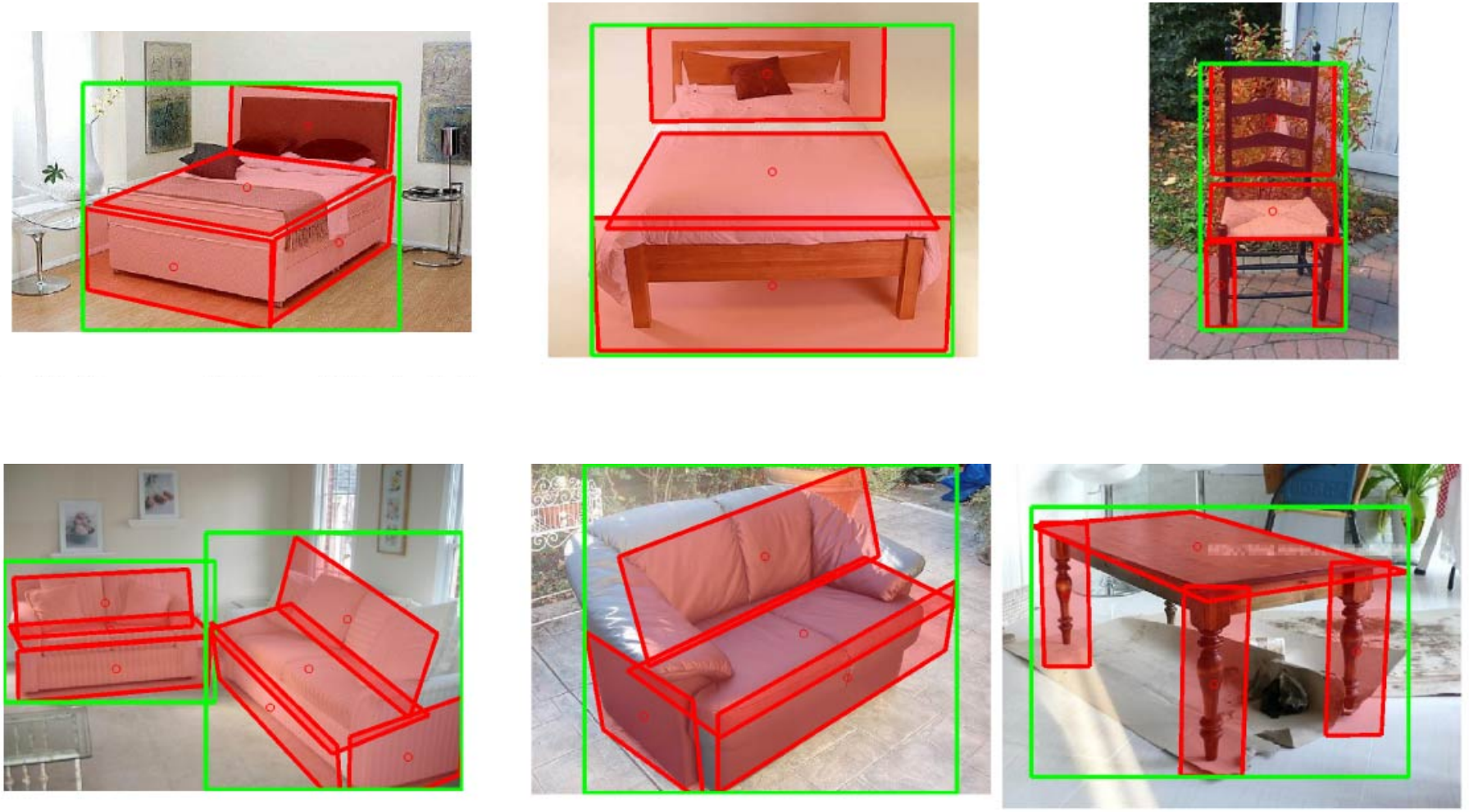
3D object detectors



3D object dataset [Savarese & Fei-Fei, ICCV 07]

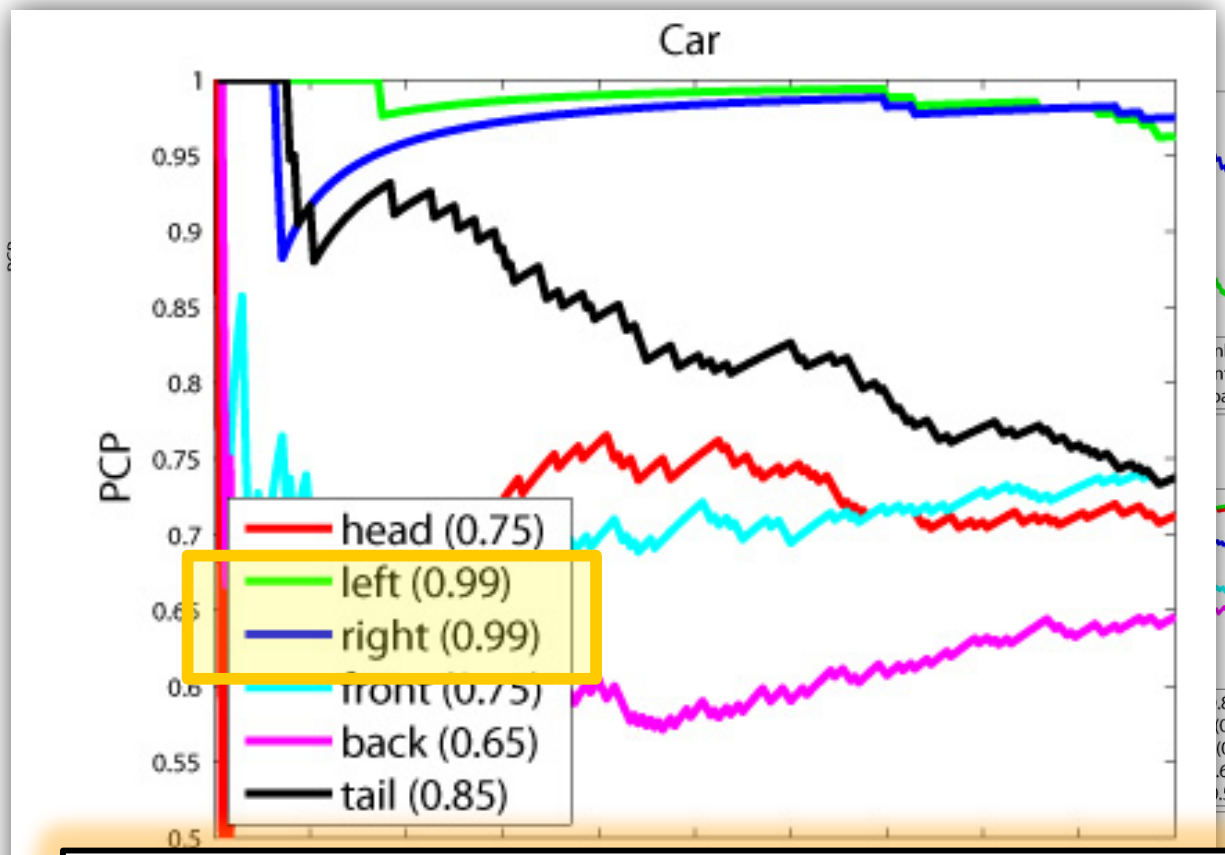
3D object detectors

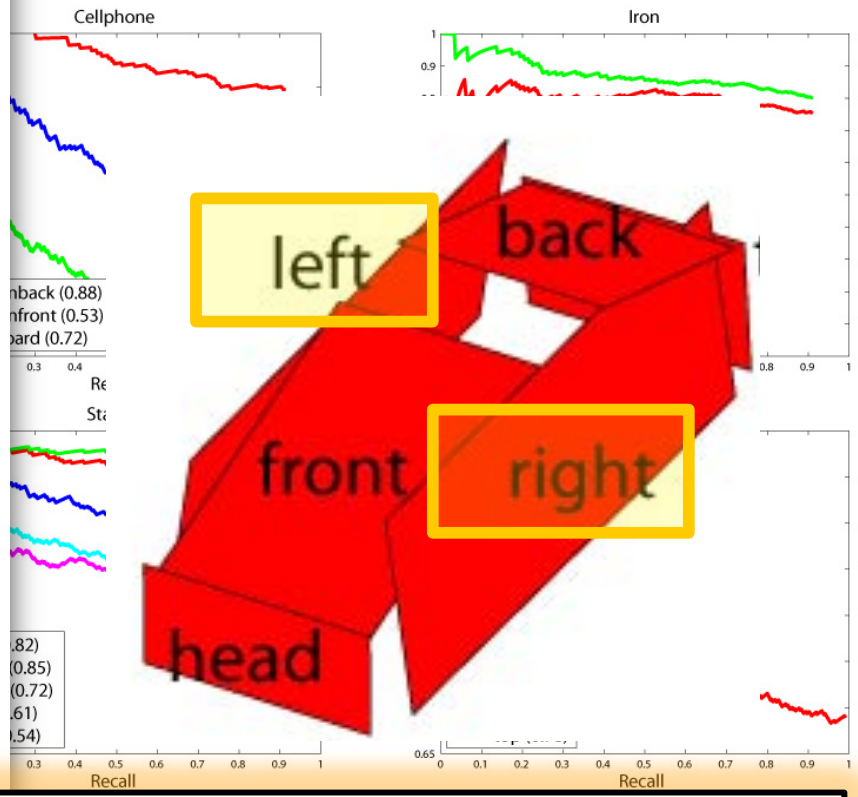


ImageNet dataset [Deng et al. 2010]

3D object detectors

- Part localization on the 3DObject dataset





PROS: Large discrimination power; Able to capture part configurations in 3D
CONS: Require more supervision; slow…

Next lecture

- Object classification, detection and segmentation by convolutional neural networks