3D Indoor Scene Understanding

CVPR'15 Tutorial

Sanja Fidler and Raquel Urtasun

June 7, 2015

UNIVERSITY OF TORONTO

Logistics

- The tutorial is online:

  http://www.cs.toronto.edu/~fidler/3DsceneTutorialCVPR15.html

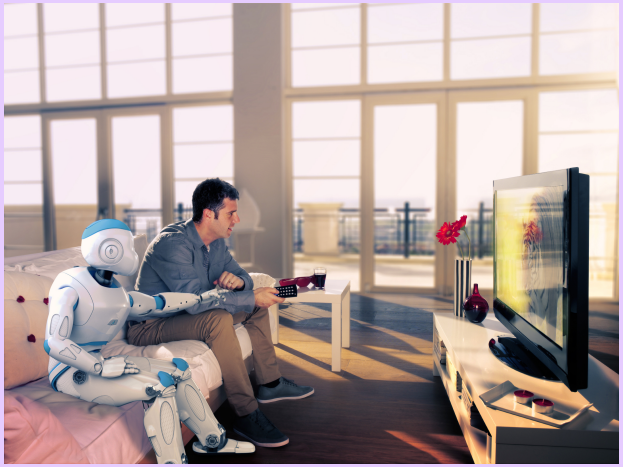
  with:

  - Slides
  - References
  - Links to datasets and code
  - Links to other similar tutorials

- Today: break 3.45-4.15pm

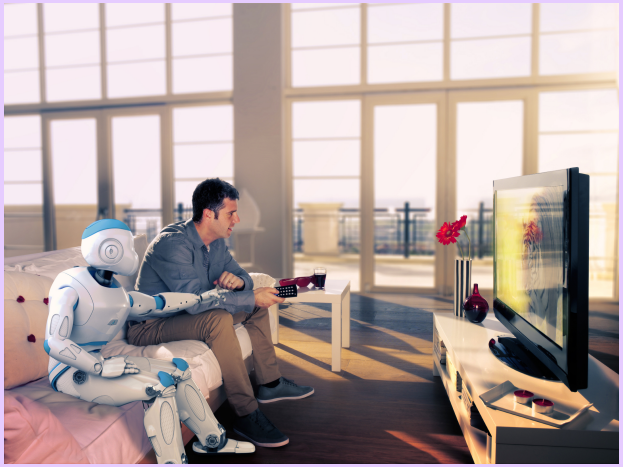
Why Indoors?

Robotics

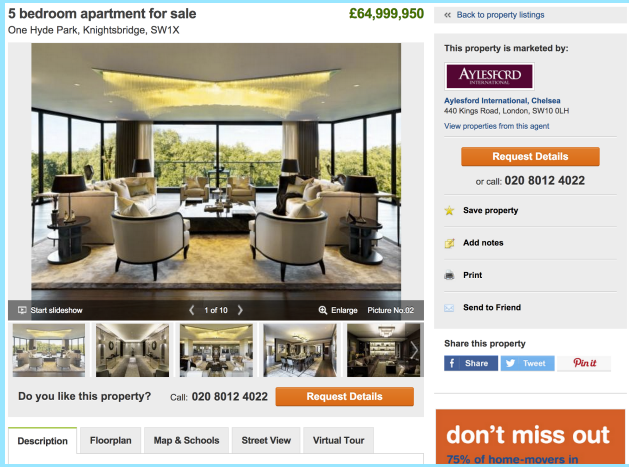


Why Indoors?

Robotics

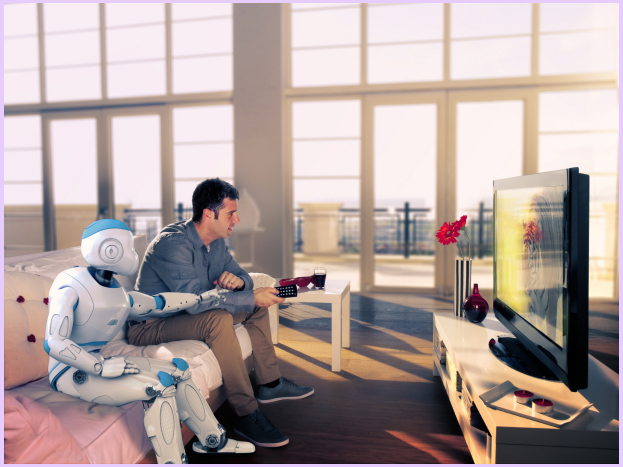


Real-estate

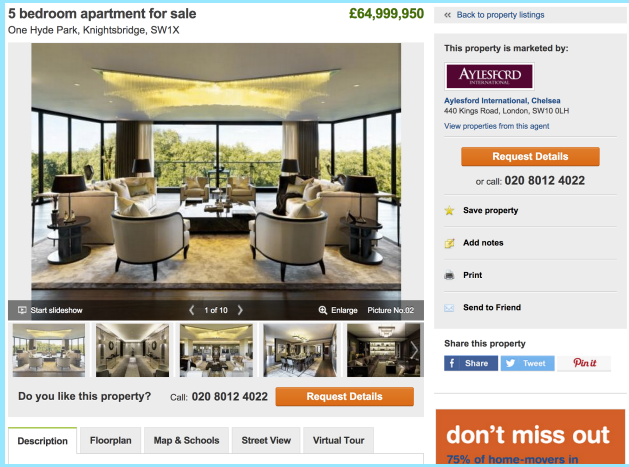


Why Indoors?

Robotics



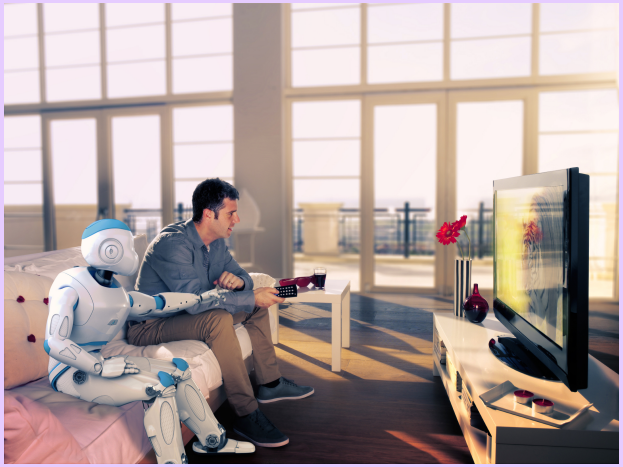
Real-estate



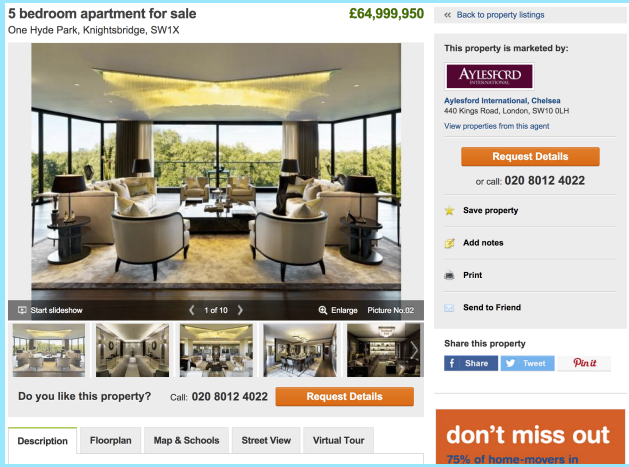
Gaming

Why Indoors?

Robotics



Real-estate



Gaming

Virtual tours

“Full” Scene Understanding?

- Full understanding of a scene?

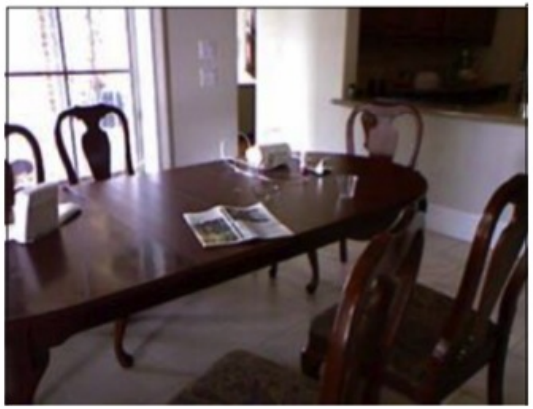
"Full" Scene Understanding?

- Full understanding of a scene? **You can answer any question about it**

[M. Malinowski, M. Fritz, A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input, *NIPS*, 2014]

"Full" Scene Understanding?

- Full understanding of a scene? **You can answer any question about it**



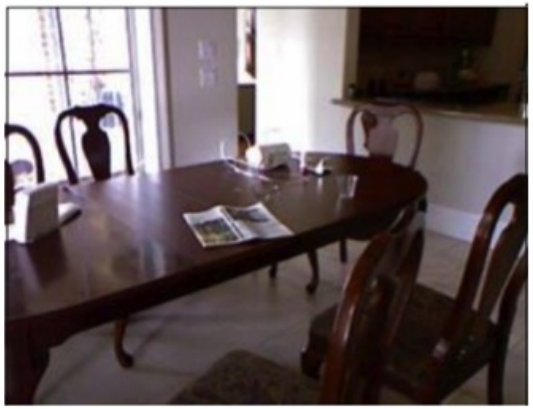
Q: What is behind the table?
A: window

Q: What is in front of the toilet?
A: door

Q: What is on the counter in the corner? A: microwave

"Full" Scene Understanding?

- Full understanding of a scene? **You can answer any question about it**



Q: What is behind the table?
A: window

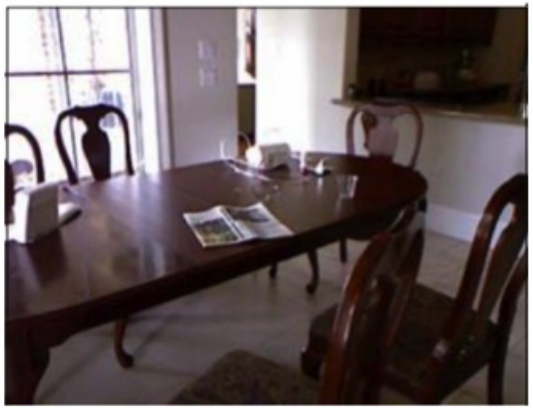
Q: What is in front of the toilet?
A: door

Q: What is on the counter in the corner? A: microwave

Q: What is the shape of the green chair? A: horse shaped

"Full" Scene Understanding?

- Full understanding of a scene? **You can answer any question about it**



Q: What is behind the table?
A: window

Q: What is in front of the toilet?
A: door

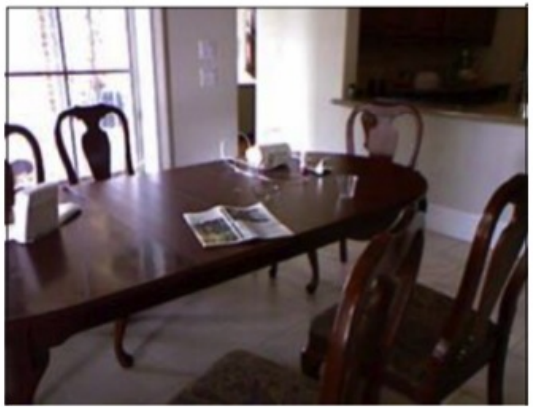
Q: What is on the counter in the corner? A: microwave

Q: What is the shape of the green chair? A: horse shaped

Q: Where is the oven?
A: on the right side of the fridge

"Full" Scene Understanding?

- Full understanding of a scene? **You can answer any question about it**



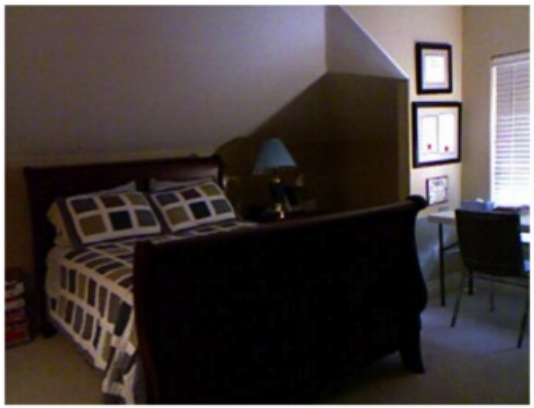
Q: What is behind the table?
A: window

Q: What is in front of the toilet?
A: door

Q: What is on the counter in the corner? A: microwave

Q: What is the shape of the green chair? A: horse shaped

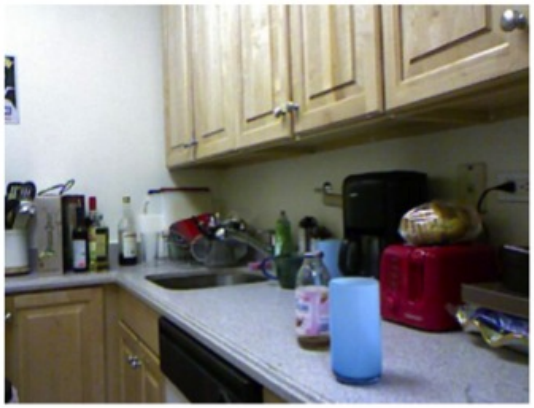
Q: Where is the oven?
A: on the right side of the fridge



Q: What is the largest object?
A: bed

“Full” Scene Understanding?

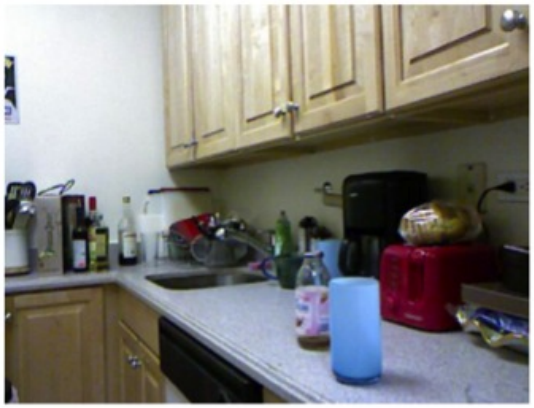
- Full understanding of a scene? **You can answer any question about it**



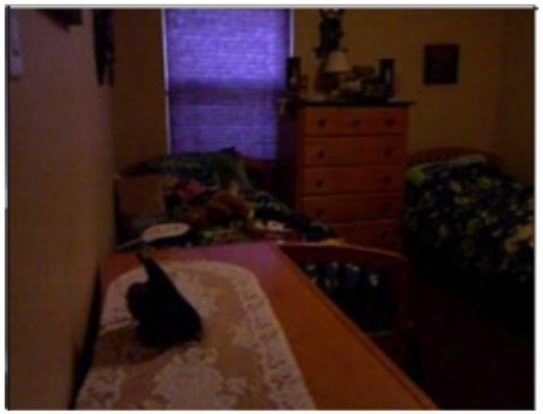
Q: Which object is red?
A: toaster

"Full" Scene Understanding?

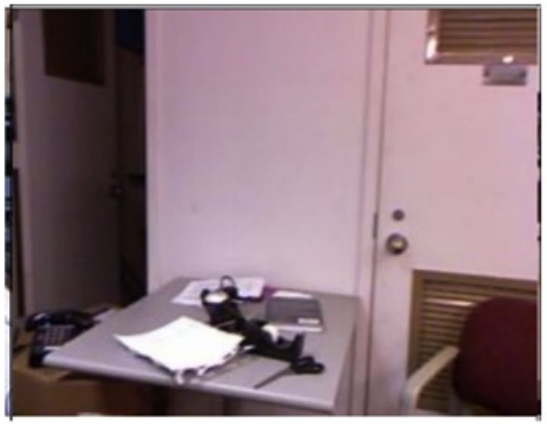
- Full understanding of a scene? **You can answer any question about it**



Q: Which object is red?
A: toaster



Q: How many drawers are there?
A: 6

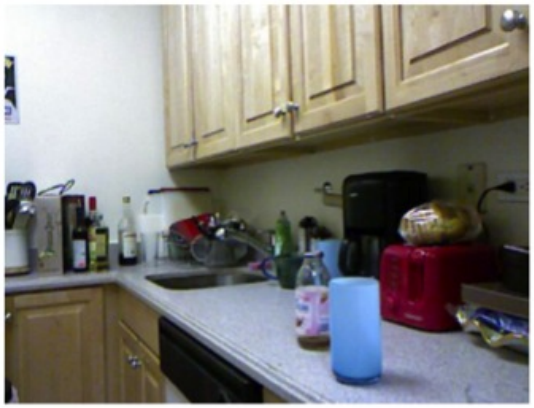


Q: How many doors are open
A: 1

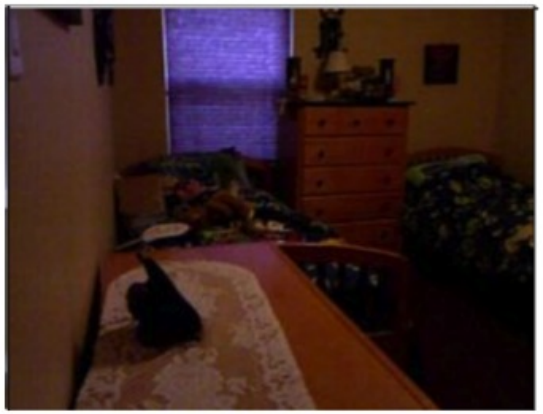
Q: How many lights are on?
A: 6

"Full" Scene Understanding?

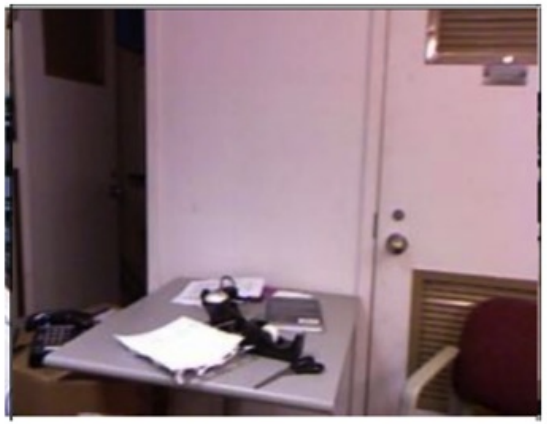
- Full understanding of a scene? **You can answer any question about it**



Q: Which object is red?
A: toaster

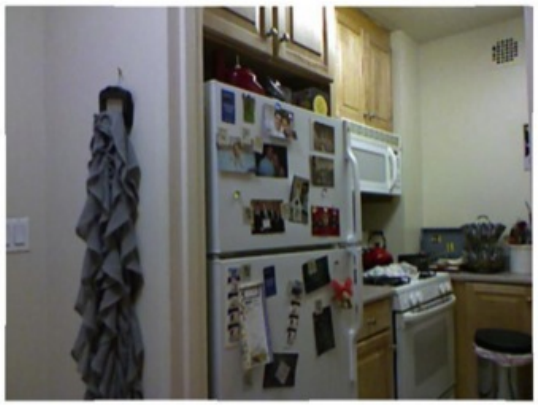


Q: How many drawers are there?
A: 6



Q: How many doors are open
A: 1

Q: How many lights are on?
A: 6



Q: Can you make pizza in this room?
A: yes

Q: Where can you sit?
A: chairs, table, floor

Overview

- Monocular 3D Object Detection
- Room Layout Estimation
  - Monocular
  - Holistic Models
- Reconstruction and Localization
- Inferring Semantics in RGB-D

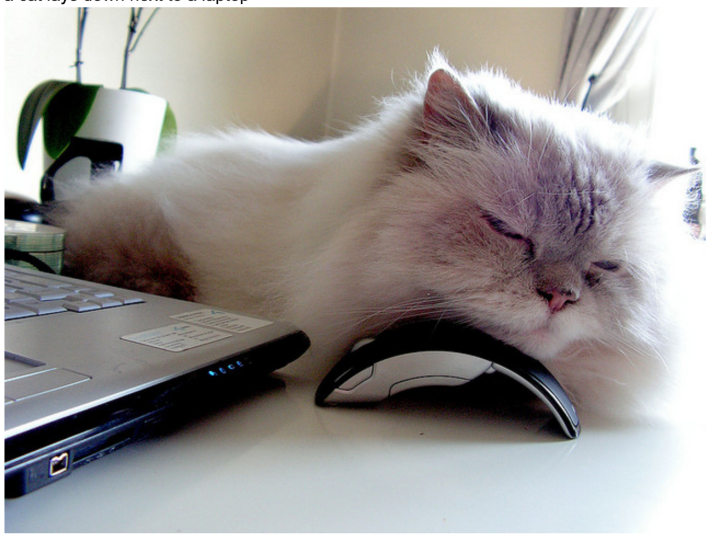
Indoor vs Outdoor vs Generic Scenes

In what way are indoor scenes “special”?

Generic Scenes

Examples from Microsoft Coco

a cat taking a nap next to a laptop resting its head on the mouse.
a cat sleeping on the mouse of a computer next to the computer.
a cat lays down next to a laptop



some very big commercial planes over the water.
two airplanes flying over water and passing each other
a couple of large airplanes out in the open.

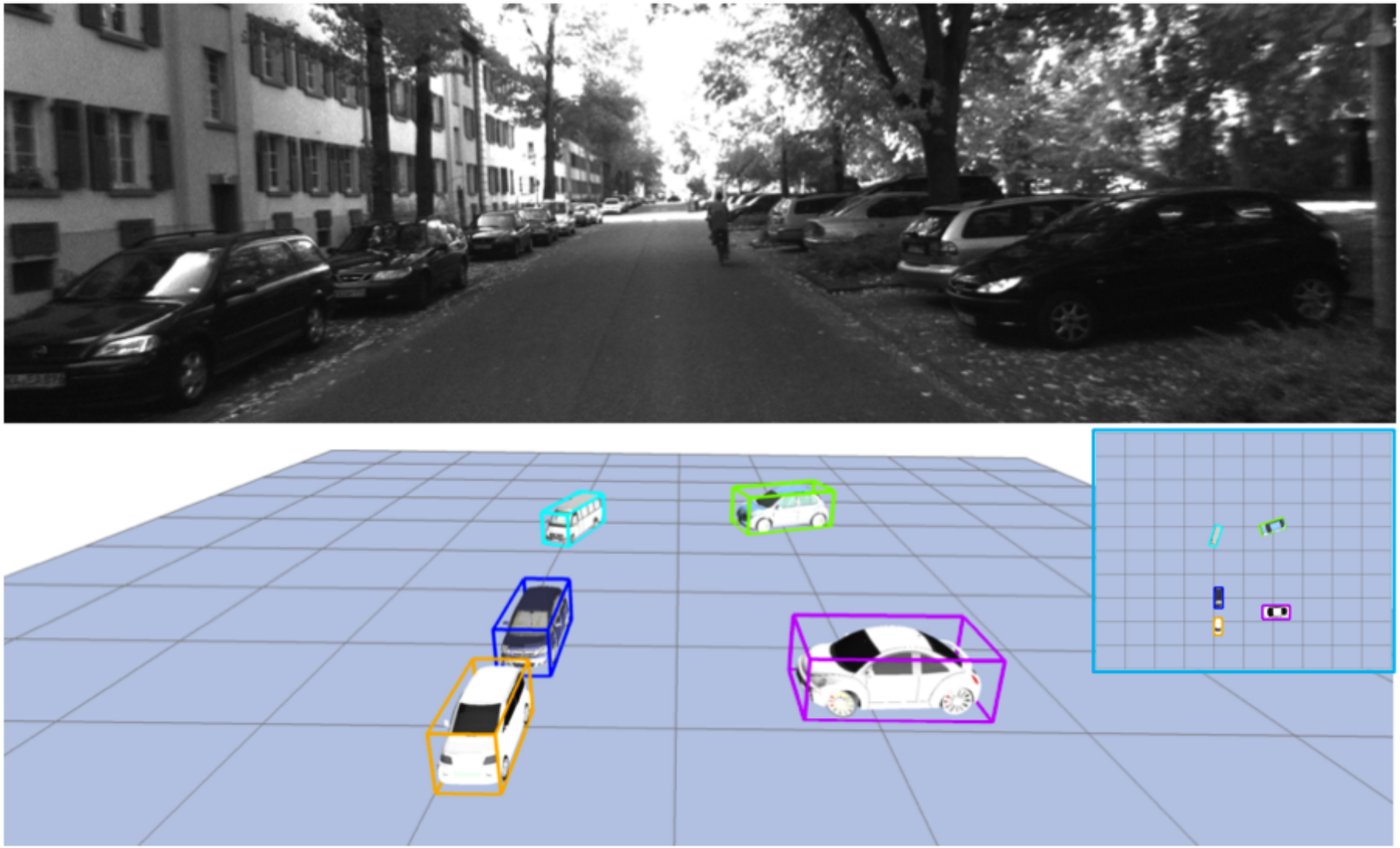
a little girl wearing a jacket and a backpack with a face on it.
a child with a backpack looking at a polar bear.
a little girl in a purple coat watches the polar bears

two white teddy bears one has pink feet the other blue.
a pair of white, boy and girl teddy bears
there are two stuffed animals sitting next to each other

Outdoor Scenes

Objects typically on the ground. Biased viewpoint.

Outdoor Scenes



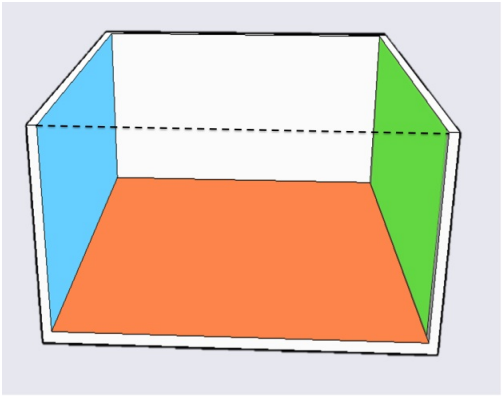
Objects typically on the ground. Biased viewpoint.

Indoor Scenes

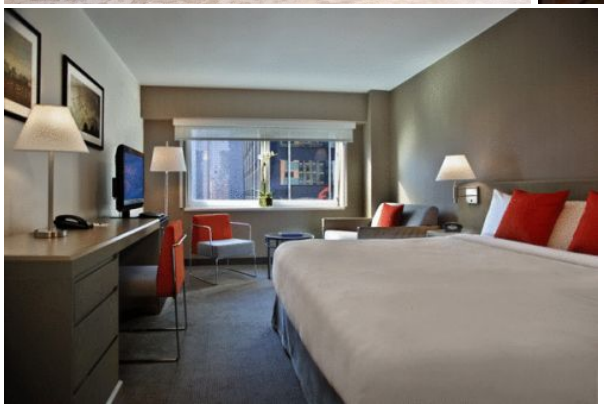
[C. Kong, D. Lin, M. Bansal, R. Urtasun, S. Fidler, What are you talking about? Text-to-Image Coreference, CVPR'14]

Description: This room is filled with different types of furniture and home goods. The lights on the ceiling are strung across the room, they are circular and bright. At the back of the room, there are shelves filled with an assortment of pillows and blankets. There are a few couches facing away from those shelves. The couches have many pillows on top of them. On the second couch, which is dark green, sits a man in a plaid shirt. Another black couch faces the second couch. In front of the black couch is a shelf containing large brown bowls on the bottom shelf, towels on the second shelf, and vases on the top shelf. In front of the shelf is a dining table with brown wooden chairs, pink placemats, white dinnerware, and a brown glass bottle.

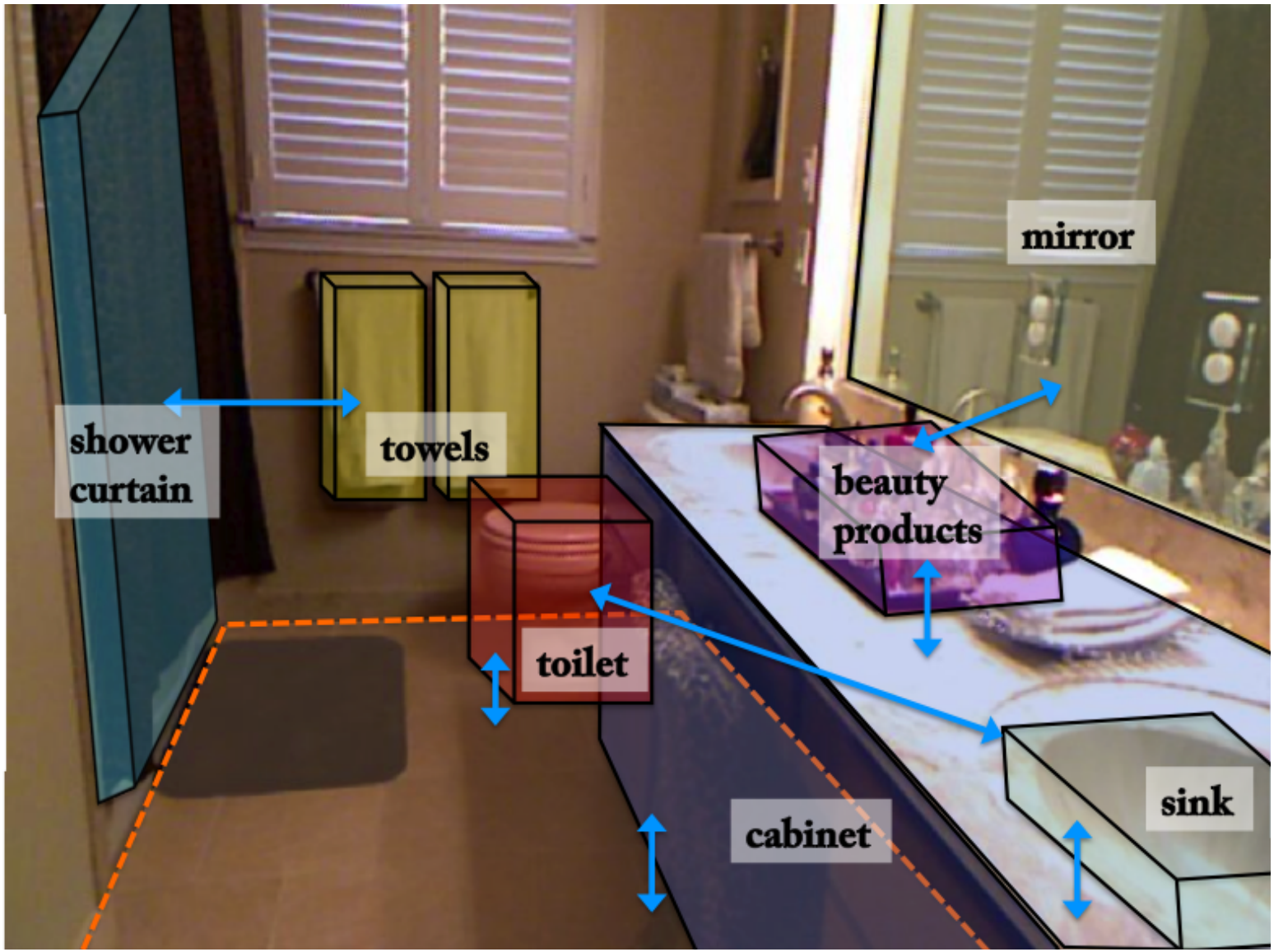
Indoor Scenes – Manhattan World





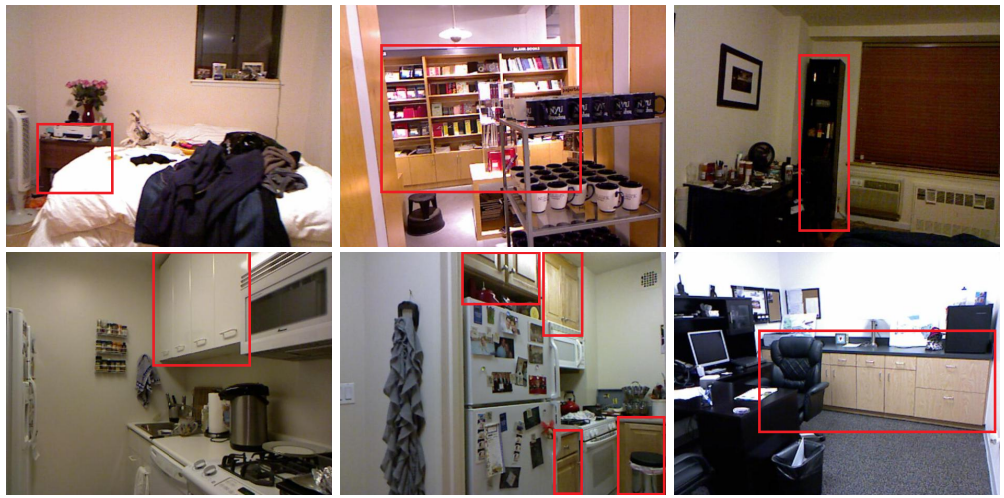
Indoor Scenes – Lots of Structure



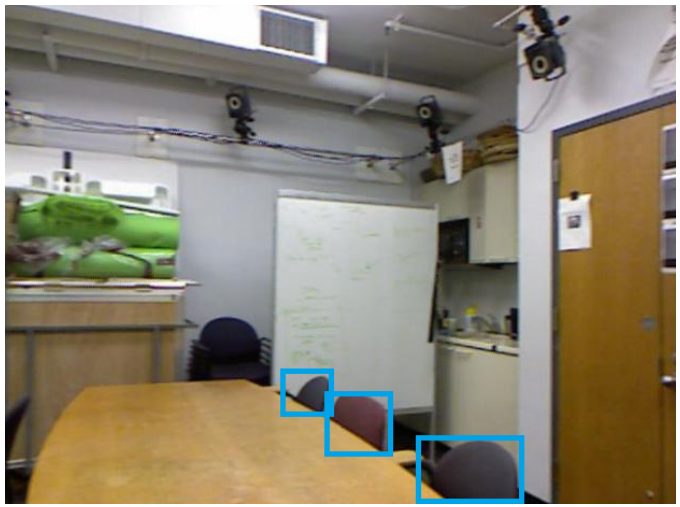
Difficult problem?

Lots of instances

Viewpoint, aspect-ratio variation



Occlusion

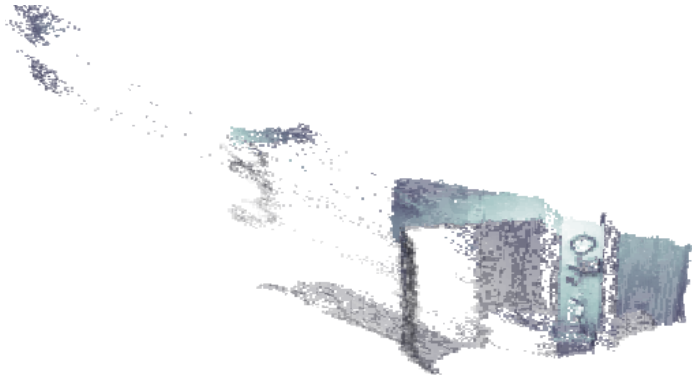


Beyond the Visible Scene

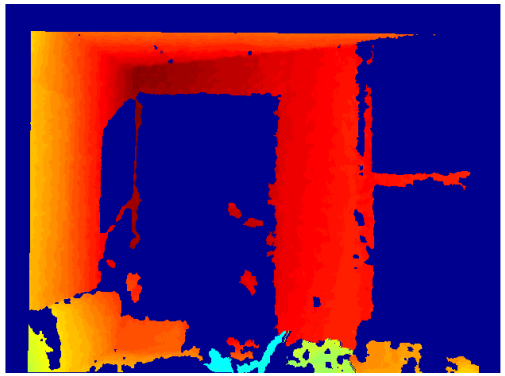
Figure by Derek Hoiem

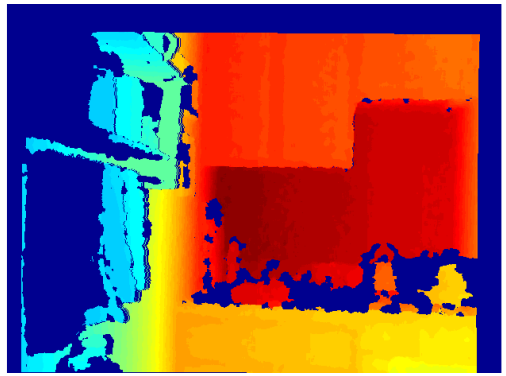
Difficult problem?

Noisy depth



Missing depth

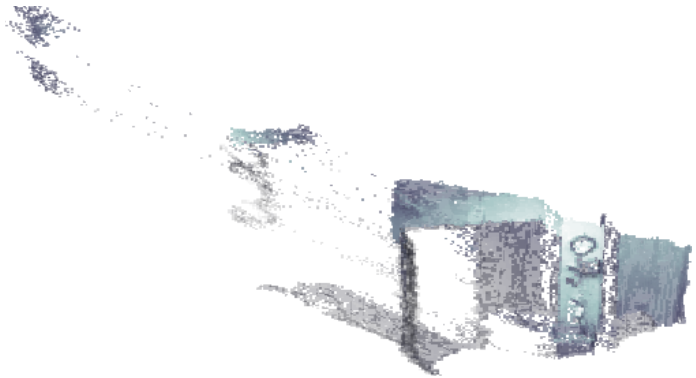




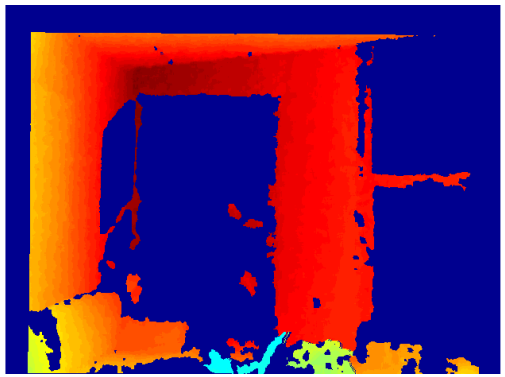
- For example, 30% of chairs have more than 50% missing depth pixels [Gupta et al., CVPR'15]

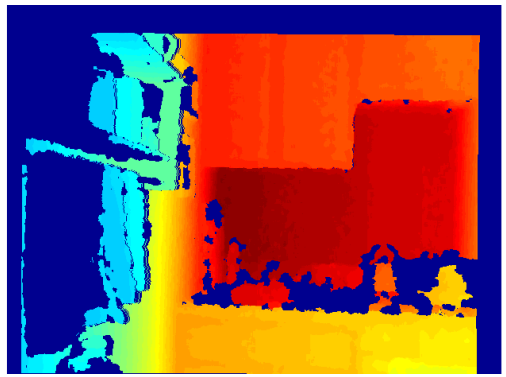
Difficult problem?

Noisy depth



Missing depth





	mean	aero plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motor bike	person	potted plant	sheep	sofa	train	tv/ monitor
segDeepM [?]	67.2	82.3	75.2	67.1	50.7	49.8	71.1	69.6	88.2	42.5	71.2	50.0	85.7	76.6	81.8	69.3	41.5	71.9	62.2	73.2	64.6
BabyLearning [?]	63.8	77.7	73.8	62.3	48.8	45.4	67.3	67.0	80.3	41.3	70.8	49.7	79.5	74.7	78.6	64.5	36.0	69.9	55.7	70.4	61.7
R-CNN (bbox reg) [?]	62.9	79.3	72.4	63.1	44.0	44.4	64.6	66.3	84.9	38.8	67.3	48.4	82.3	75.0	76.7	65.7	35.8	66.2	54.8	69.1	58.8
R-CNN [?]	59.8	76.5	70.4	58.0	40.2	39.6	61.8	63.7	81.0	36.2	64.5	45.7	80.5	71.9	74.3	60.6	31.5	64.7	52.5	64.6	57.2
Feature Edit [?]	56.4	74.8	69.2	55.7	41.9	36.1	64.7	62.3	69.5	31.3	53.3	43.7	69.9	64.0	71.8	60.5	32.7	63.0	44.1	63.6	56.6
R-CNN (bbox reg) [?]	53.7	71.8	65.8	53.0	36.8	35.9	59.7	60.0	69.9	27.9	50.6	41.4	70.0	62.0	69.0	58.1	29.5	59.4	39.3	61.2	52.4
R-CNN [?]	50.2	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2

PASCAL

	mean	bath tub	bed	book shelf	box	chair	count- -er	desk	door	dress- -er	garba- -ge bin	lamp	monit- -or	night stand	pillow	sink	sofa	table	tele vision	toilet
RGB DPM	**9.0**	0.9	27.6	9.0	0.1	7.8	7.3	0.7	2.5	1.4	6.6	22.2	10.0	9.2	4.3	5.9	9.4	5.5	5.8	34.4
RGBD-DPM	**23.9**	19.3	56.0	17.5	0.6	23.5	24.0	6.2	9.5	16.4	26.7	26.7	34.9	32.6	20.7	22.8	34.2	17.2	19.5	45.1
RGB R-CNN	**22.5**	16.9	45.3	28.5	0.7	25.9	30.4	9.7	16.3	18.9	15.7	27.9	32.5	17.0	11.1	16.6	29.4	12.7	27.4	44.1
Our	**37.3**	**44.4**	**71.0**	**32.9**	**1.4**	**43.3**	**44.0**	**15.1**	**24.5**	**30.4**	**39.4**	**36.5**	**52.6**	**40.0**	**34.8**	**36.1**	**53.9**	**24.4**	**37.5**	**46.8**

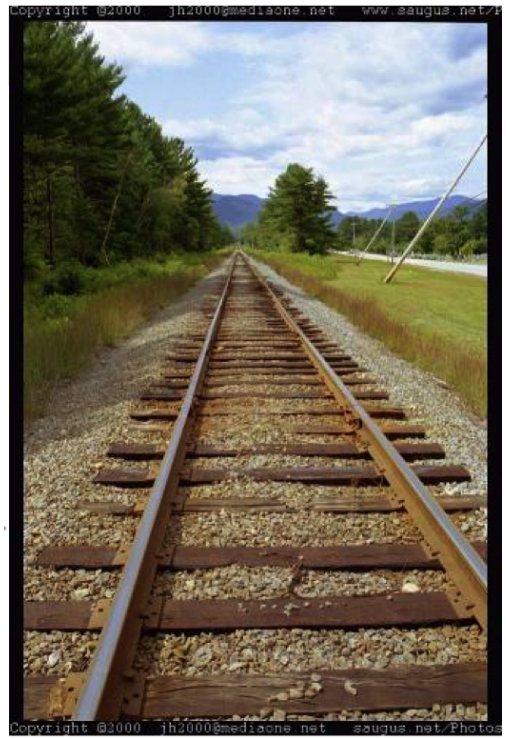
[Gupta'14]

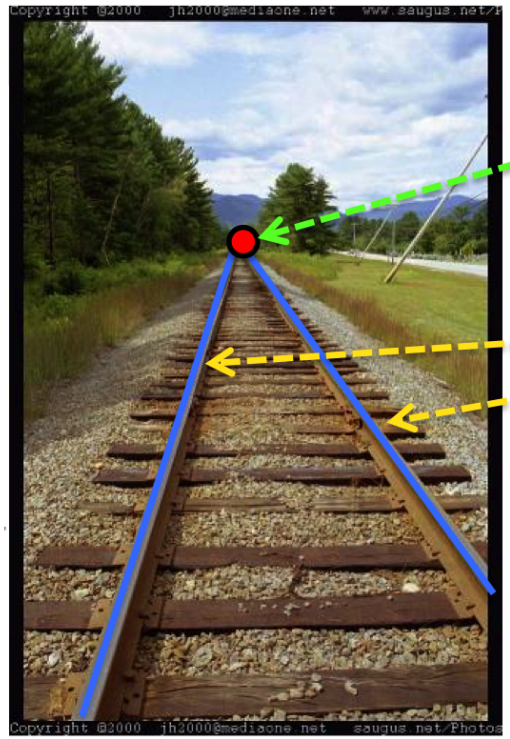
Basic Geometry

Basic Geometry

Parallel lines converge at a **vanishing point**

- Each different direction in the world **has its own vanishing point**



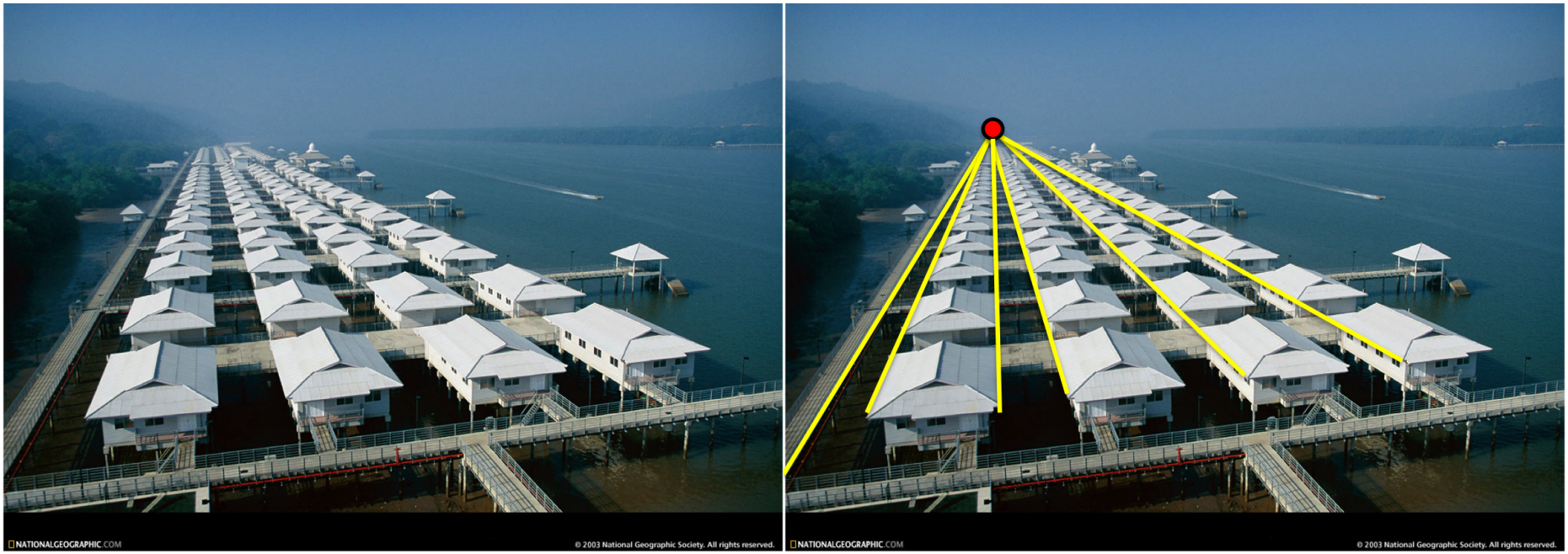


[Adopted from: N. Snavely, R. Urtasun]

Basic Geometry

Parallel lines converge at a **vanishing point**

- Each different direction in the world **has its own vanishing point**
- All lines with the same 3D direction intersect at the **same vanishing point**

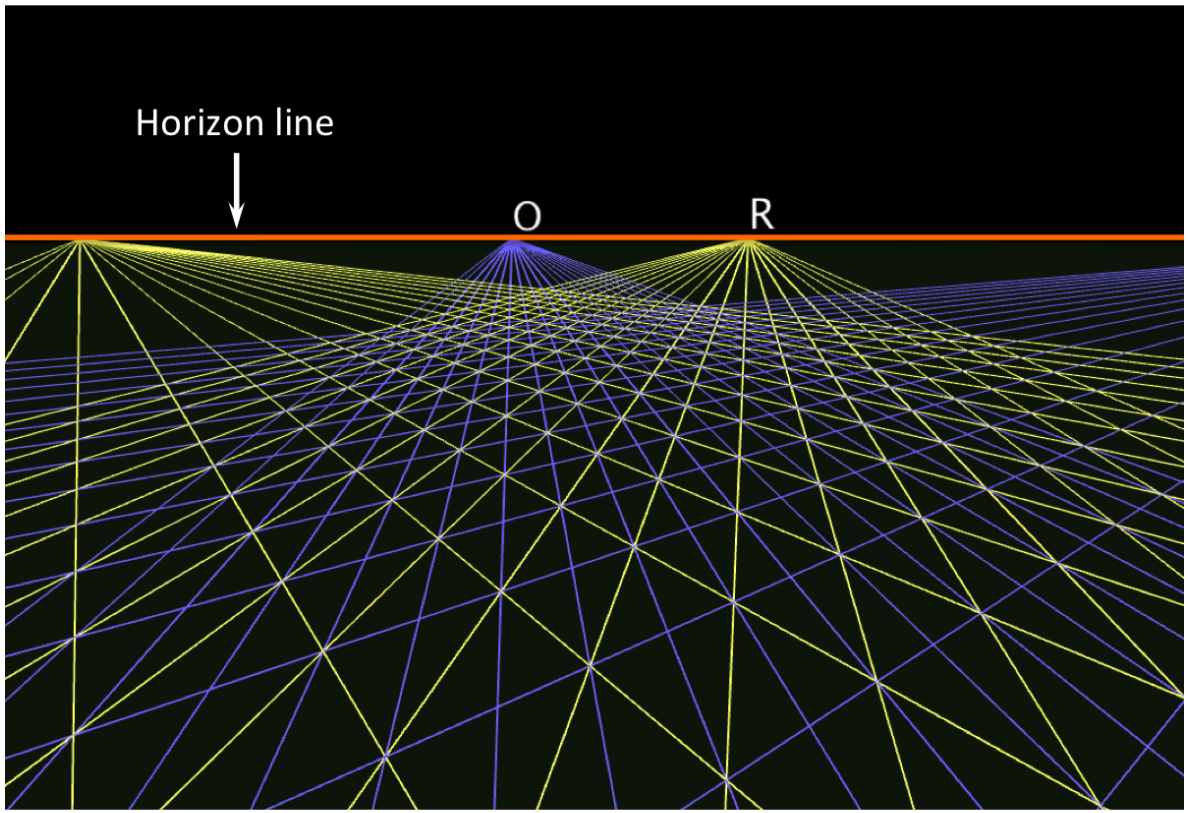


[Pic: R. Szeliski]

Basic Geometry

Parallel lines converge at a **vanishing point**

- Each different direction in the world **has its own vanishing point**
- For lines on the same 3D plane, the vanishing points lie on a **line**. We call it a **vanishing line**. Vanishing line for the ground plane is a **horizon line**.



http://4.bp.blogspot.com/-0Jm9d9j35Tc/T5ESbVpKI7I/AAAAAAAACEk/nVAiTxBuiyc/s1600/perspectiveGrid-01.png

Basic Geometry

Parallel lines converge at a **vanishing point**

- For lines on the same 3D plane, the vanishing points lie on a **line**. We call it a **vanishing line** or a **horizon line**.
- Parallel planes in 3D have the **same horizon line** in the image.

Example

- Can I tell how much above ground this picture was taken?

Example

- Can I tell how much above ground this picture was taken?

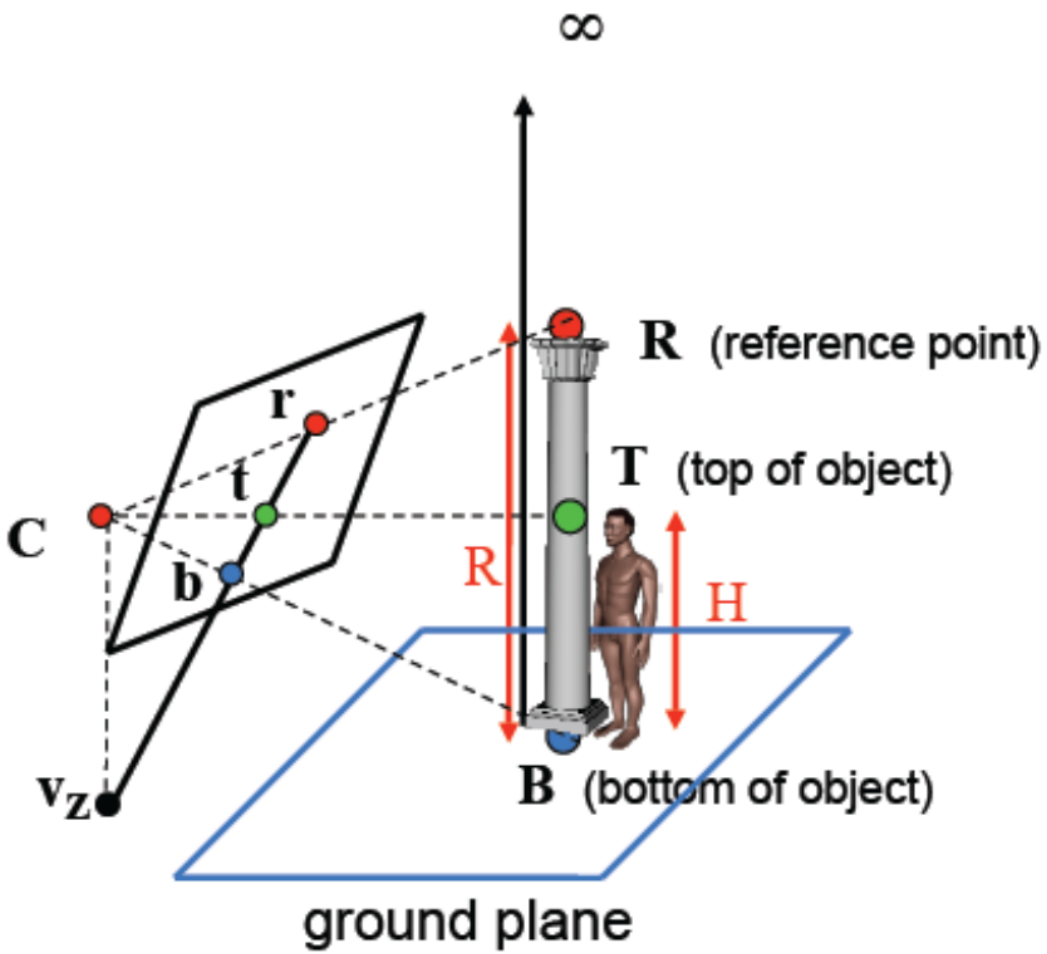
Example

- Same distance as where the horizon intersects a building

Example

- Same distance as where the horizon intersects a building: 50 floors up

Cross-ratio



$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|}=\frac{H}{R}$$

scene cross ratio

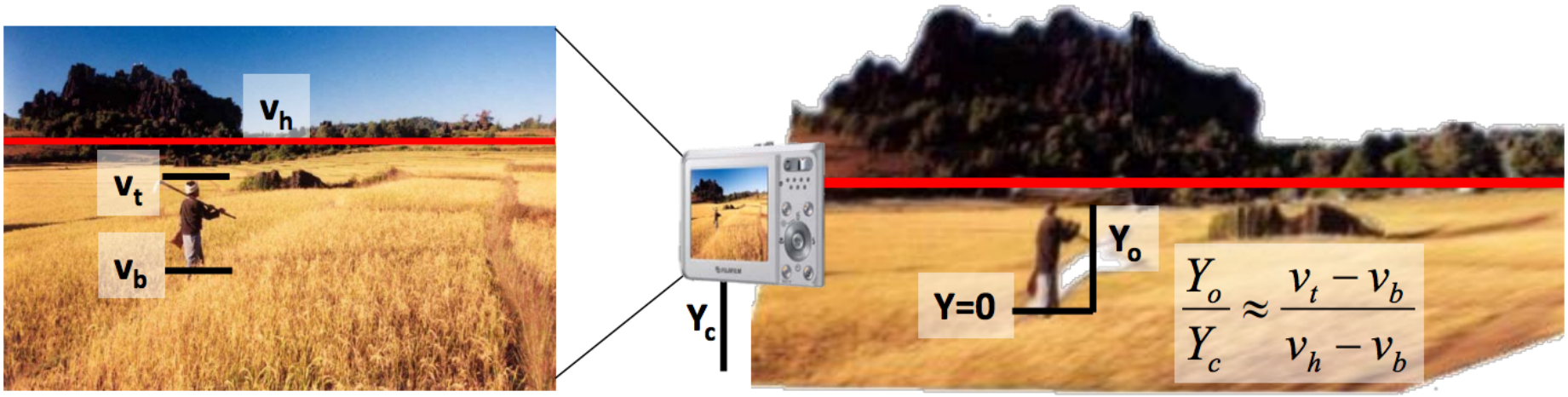
$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}=\frac{H}{R}$$

image cross ratio

[Figure by Steve Seitz]

Cross-ratio

- When the camera is upright and not slanted:



[Figure by Derek Hoiem]

Camera Estimation for a Manhattan World

- For images where you see lines corresponding to 3 orthogonal directions you can compute the camera matrix K as well as rotation matrix R

- Reference: Zisserman & Hartley book.

Single Image Reconstruction

- One can reconstruct the scene in 3D from a **single image**, under certain assumptions.

link to video

Single Image Reconstruction

- One can reconstruct the scene in 3D from a **single image**, under certain assumptions.

A. Criminisi, I. Reid, and A. Zisserman

Single View Metrology

International Journal of Computer Vision, vol 40, num 2, 2000

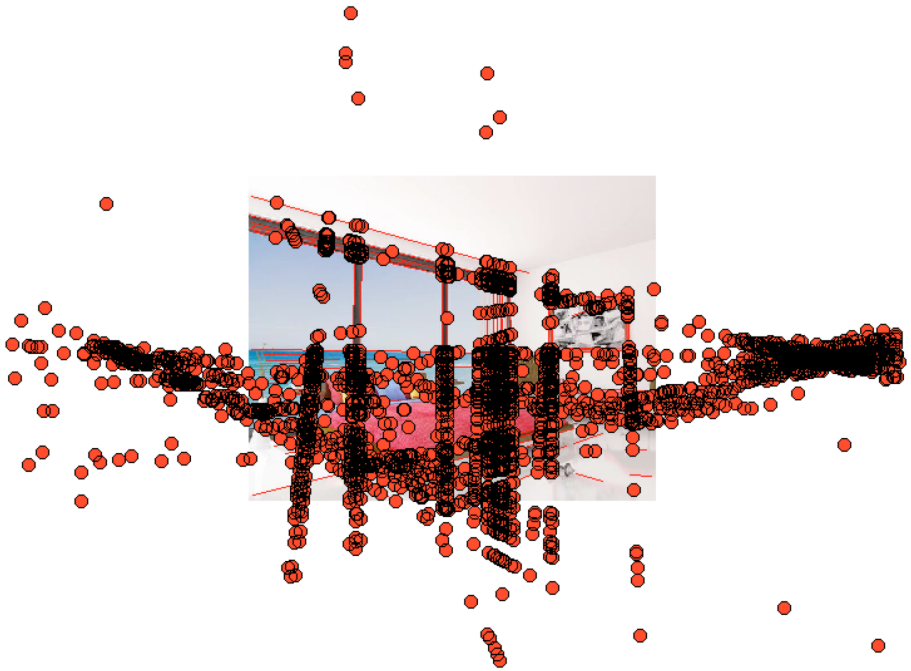
http://www.cs.cmu.edu/ ph/869/papers/Criminisi99.pdf

Estimating Vanishing Points

- Detect lines in an image

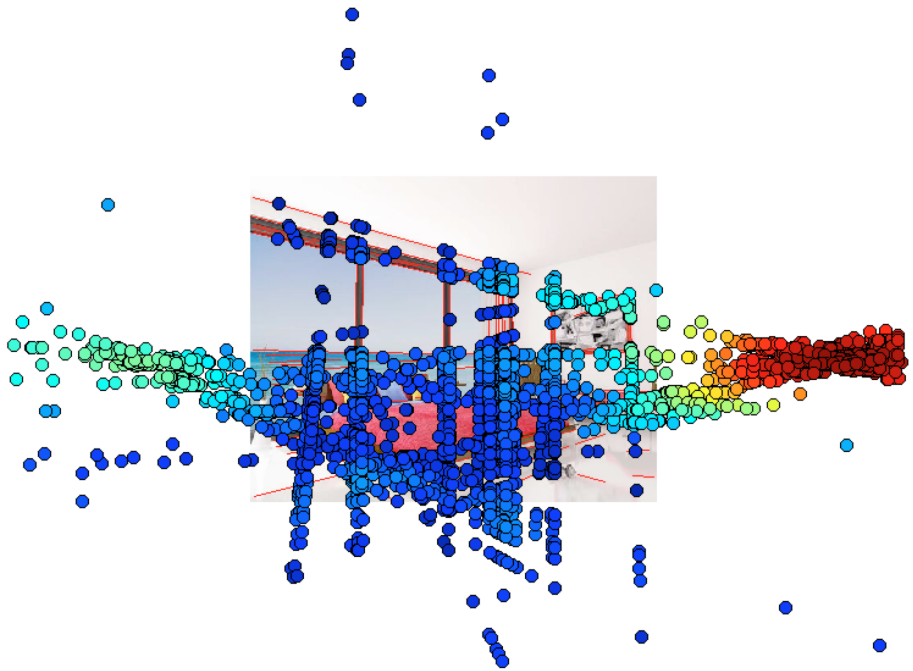
Estimating Vanishing Points

- Detect lines in an image
- Find all intersections of lines



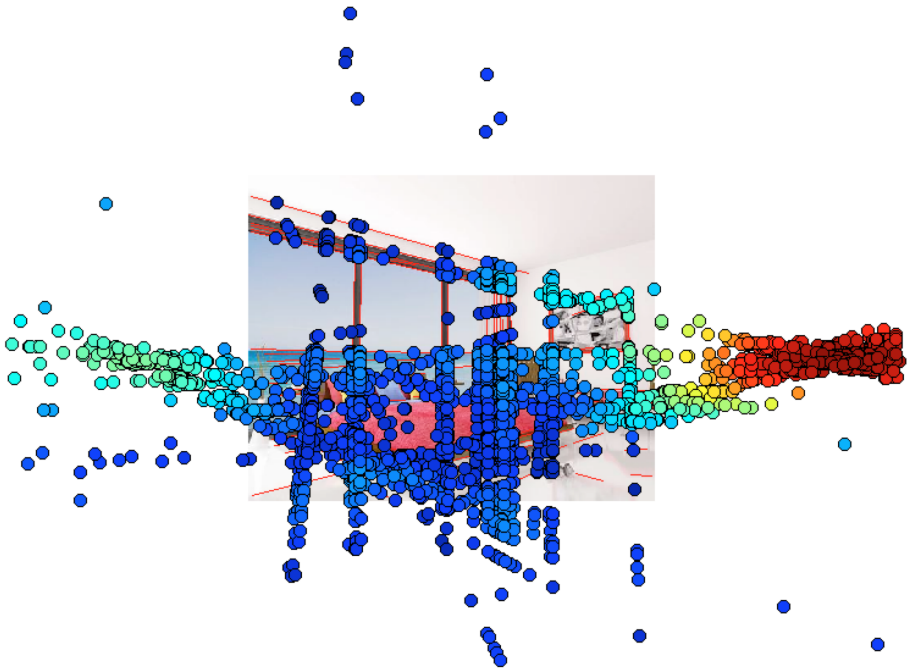
Estimating Vanishing Points

- Detect lines in an image
- Find all intersections of lines
- Vote for each intersection



Estimating Vanishing Points

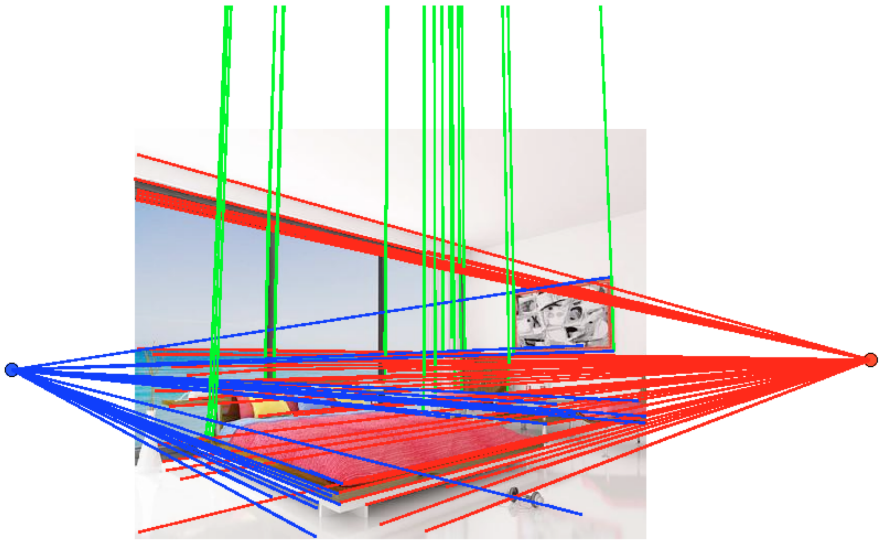
- Detect lines in an image
- Find all intersections of lines
- Vote for each intersection
- Solve: $\mathrm{vp}_1, \mathrm{vp}_2, \mathrm{vp}_3 = \underset{\mathrm{ortho}(p,q,r)}{\mathrm{argmax}} \Big(\mathrm{vote}(p) + \mathrm{vote}(q) + \mathrm{vote}(r)\Big)$



Estimating Vanishing Points

- Detect lines in an image
- Find all intersections of lines
- Vote for each intersection
- Solve: $\mathrm{vp}_1, \mathrm{vp}_2, \mathrm{vp}_3 = \underset{\mathrm{ortho}(p,q,r)}{\mathrm{argmax}} \Big(\mathrm{vote}(p) + \mathrm{vote}(q) + \mathrm{vote}(r)\Big)$

- **Greedy**: Lee et al., NIPS'10, Hedau et al., ICCV'09 (code)
- **Exact** (when K known): Bazin et al., CVPR'12

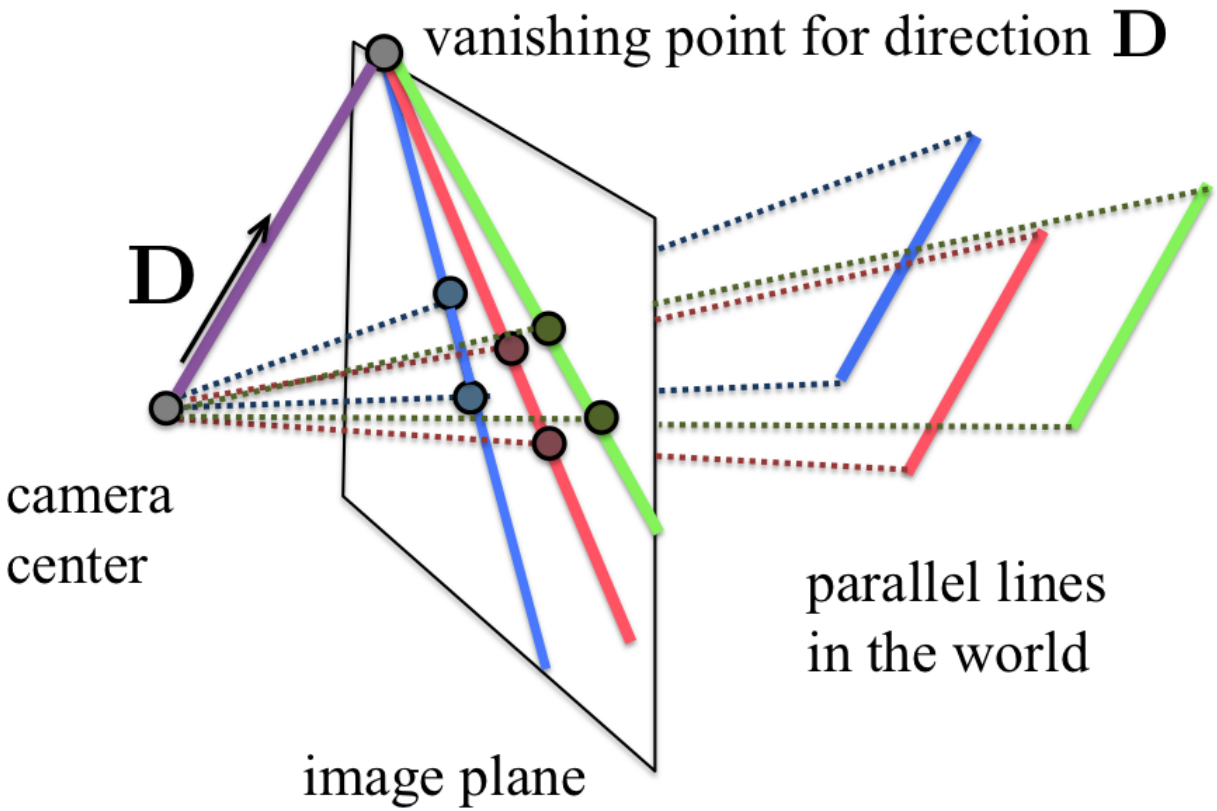


Estimate K and R

- Direction in 3D:

$$\mathbf{d} = \frac{K^{-1}\overline{\mathrm{vp}}}{||K^{-1}\overline{\mathrm{vp}}||}$$

where $\overline{\mathrm{p}}$ denotes a point in homogeneous coordinates



Estimate K and R

- Direction in 3D:

$$\mathbf{d} = \frac{K^{-1}\overline{\mathrm{vp}}}{||K^{-1}\overline{\mathrm{vp}}||}$$

  where $\overline{\mathrm{p}}$ denotes a point in homogeneous coordinates

- The three directions are orthogonal:

$$(K^{-1}\overline{\mathrm{vp}_1})^T \cdot K^{-1}\overline{\mathrm{vp}_2} = 0$$
$$(K^{-1}\overline{\mathrm{vp}_1})^T \cdot K^{-1}\overline{\mathrm{vp}_3} = 0$$
$$(K^{-1}\overline{\mathrm{vp}_2})^T \cdot K^{-1}\overline{\mathrm{vp}_3} = 0$$

Estimate K and R

- Direction in 3D:

$$\mathbf{d} = \frac{K^{-1}\overline{\mathrm{vp}}}{||K^{-1}\overline{\mathrm{vp}}||}$$

  where $\overline{\mathrm{p}}$ denotes a point in homogeneous coordinates

- The three directions are orthogonal:

$$(K^{-1}\overline{\mathrm{vp}_1})^T \cdot K^{-1}\overline{\mathrm{vp}_2} = 0$$
$$(K^{-1}\overline{\mathrm{vp}_1})^T \cdot K^{-1}\overline{\mathrm{vp}_3} = 0$$
$$(K^{-1}\overline{\mathrm{vp}_2})^T \cdot K^{-1}\overline{\mathrm{vp}_3} = 0$$

- Compute K

Estimate K and R

- Direction in 3D:

$$\mathbf{d} = \frac{K^{-1}\overline{\mathrm{vp}}}{||K^{-1}\overline{\mathrm{vp}}||}$$

  where $\overline{\mathrm{p}}$ denotes a point in homogeneous coordinates

- The three directions are orthogonal:

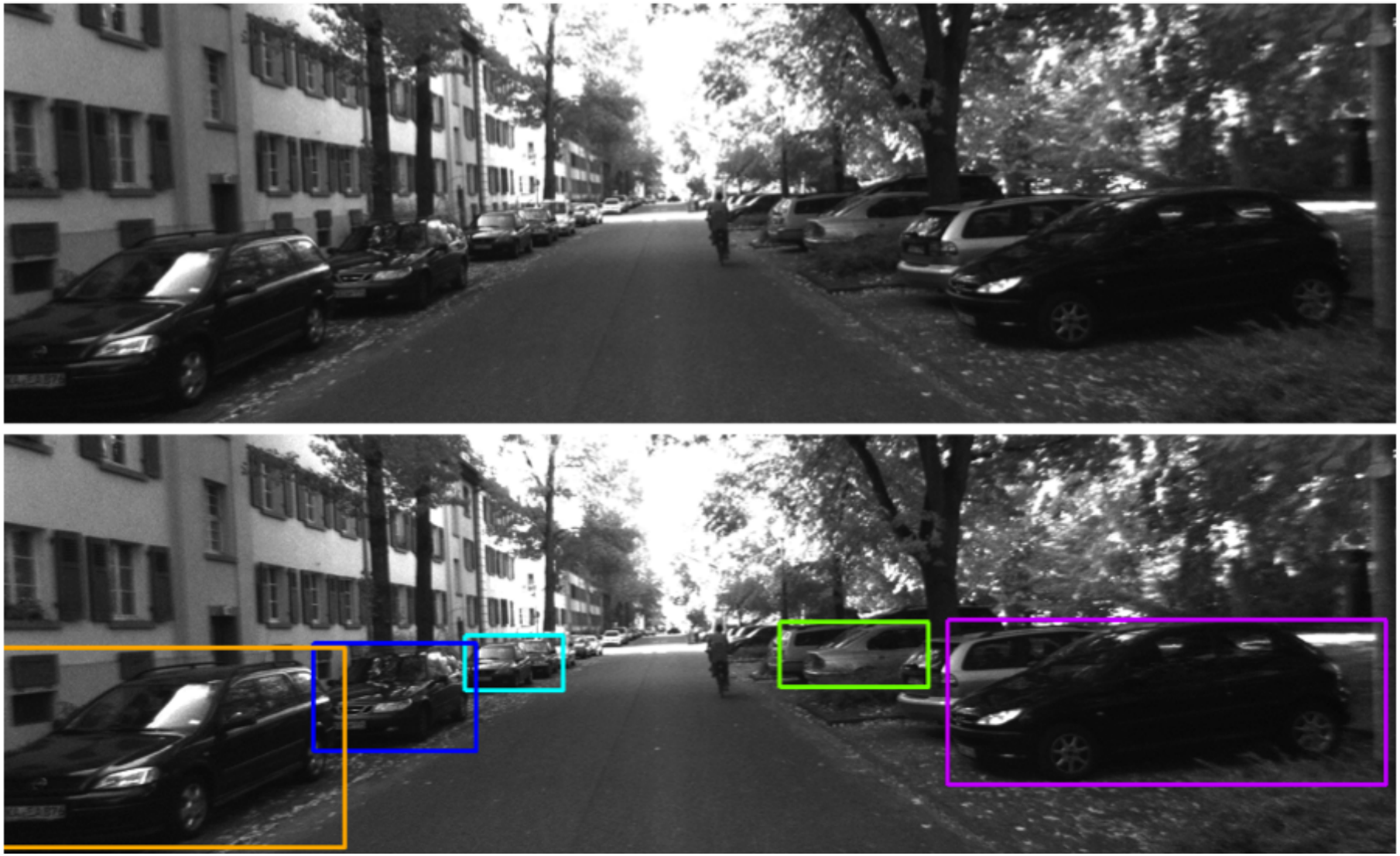
$$(K^{-1}\overline{\mathrm{vp}_1})^T \cdot K^{-1}\overline{\mathrm{vp}_2} = 0$$
$$(K^{-1}\overline{\mathrm{vp}_1})^T \cdot K^{-1}\overline{\mathrm{vp}_3} = 0$$
$$(K^{-1}\overline{\mathrm{vp}_2})^T \cdot K^{-1}\overline{\mathrm{vp}_3} = 0$$

- Compute K
- Compute $R = \left[\mathbf{d_1}, \mathbf{d_2}, \mathbf{d_3}\right]$, where $\mathbf{d_i}$ is a direction corresponding to the vanishing point vp_i

3D Object Detection
Monocular Case

Object detection

Object detection

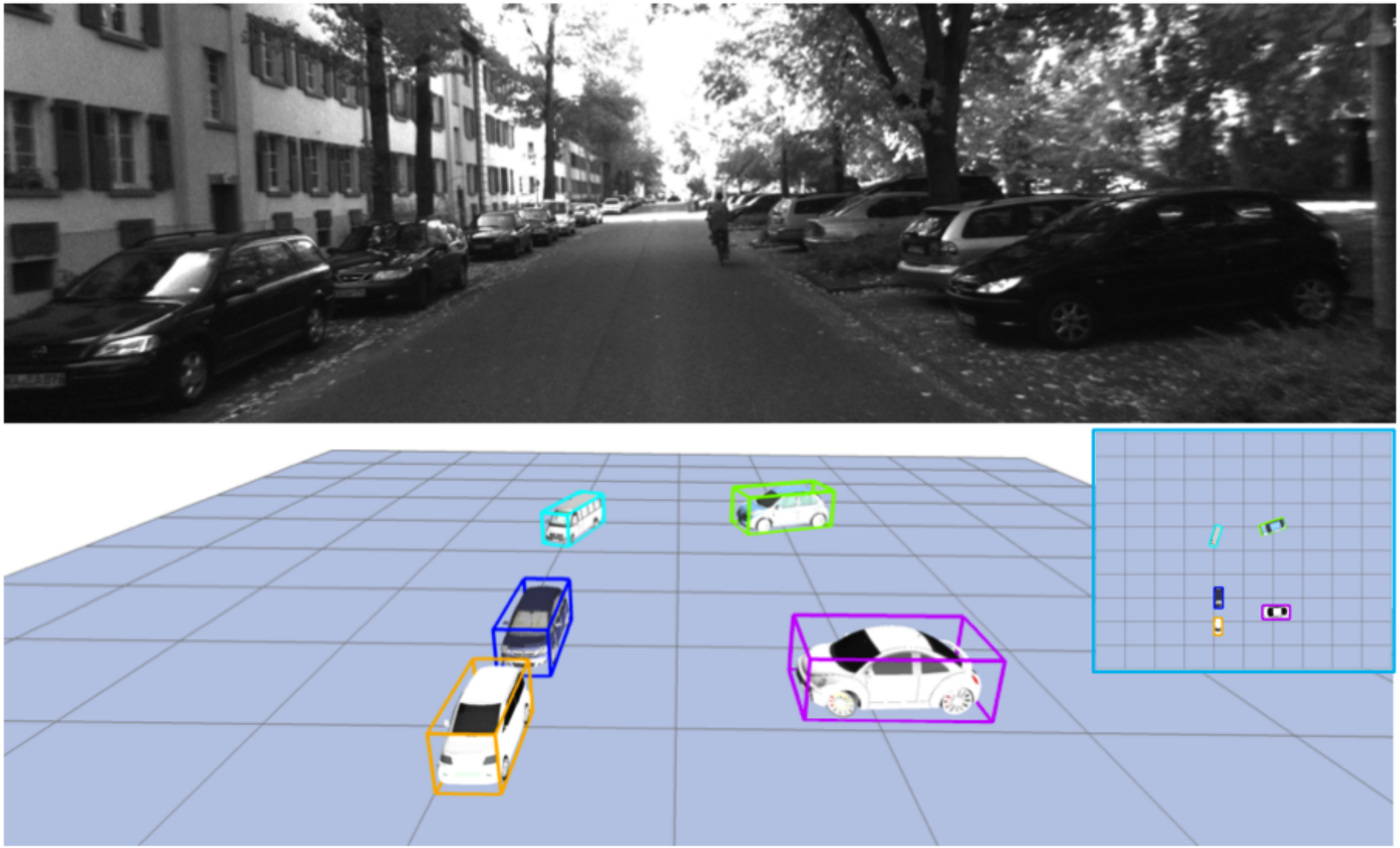


Usually detectors output 2D boxes around the objects.

3D Object detection

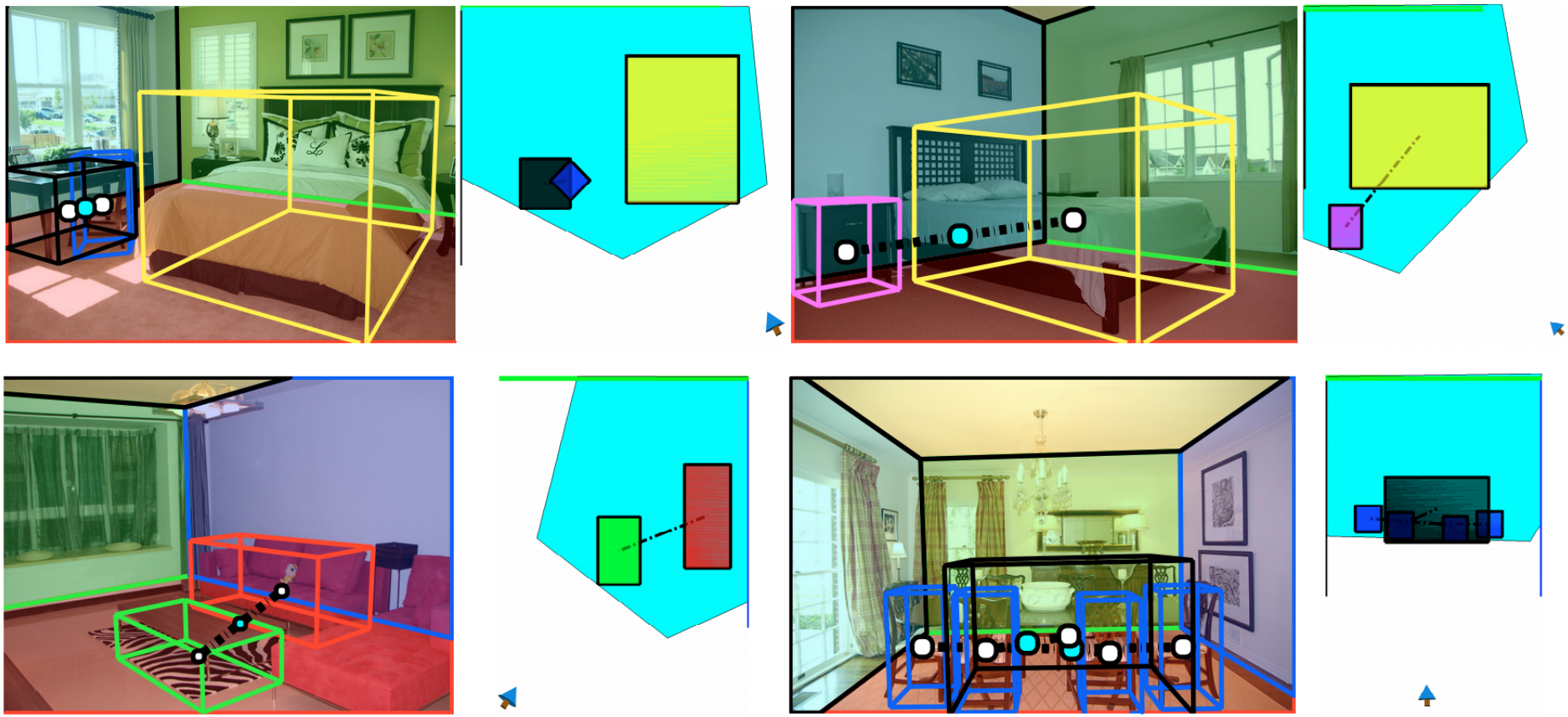
Important to also infer accurate object pose.

3D Object detection



as well as location and extent of objects in 3D.

3D Object Detection Indoors



Important for **free space** estimation.

Figure from: Choi et al., CVPR 2013

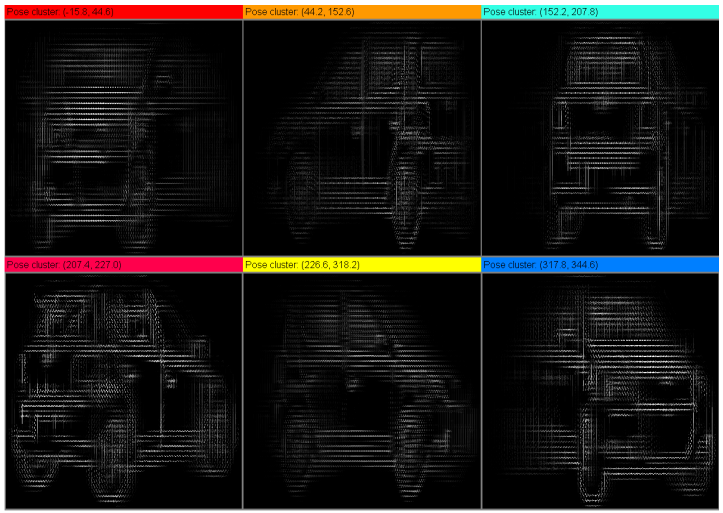
3D Object Detection Indoors

Accurate prediction is important.

Literature – 3D Object Detection

Essentially two types of approaches:

- **Viewer-centered**: object is modeled by a collection of 2D appearance models [Torralba07, Felzenswalb10, Pepik12, etc], one for each viewpoint
- **Object-centered**: represent object classes with a 3D model typically equipped with view-invariant geometry and appearance [Leibelt08, Savarese07, Glasner11, Yan07]



Types of Approaches

Object is a:

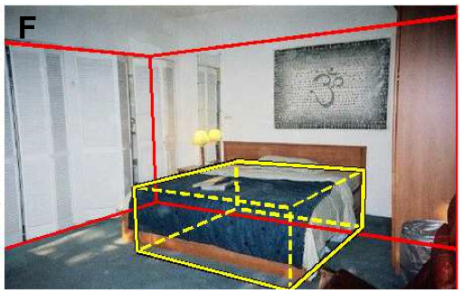
box

Object is:

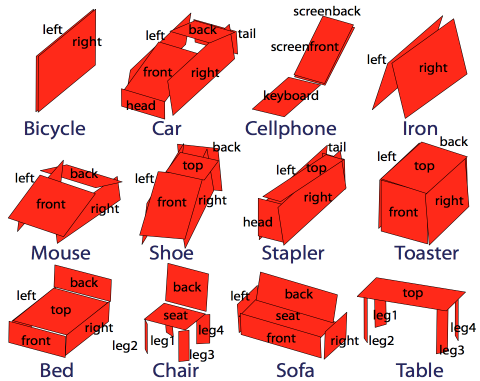
polygonal

Object is a detailed

CAD model



Hedau et al., ECCV'10
Fidler et al., NIPS'12
Hedau et al., CVPR'12



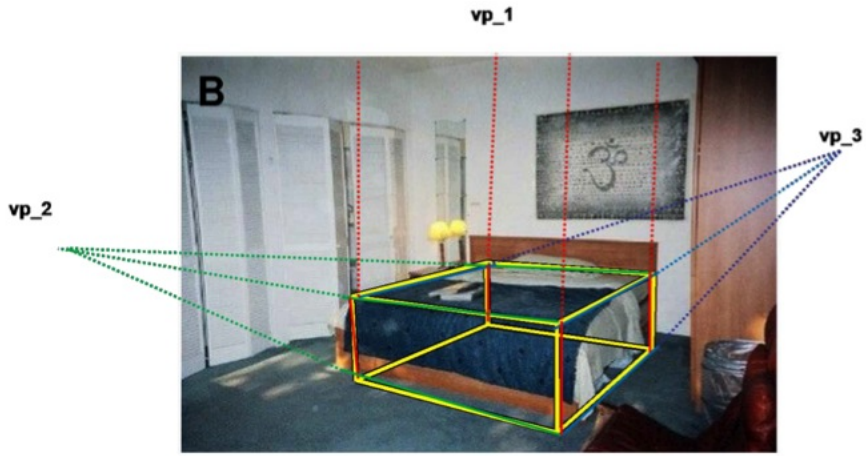
Xiang et al., CVPR'12

Lim et al., ICCV'13
Aubry et al., CVPR'14

Thinking Inside the Box [Hedau et al., 2010]

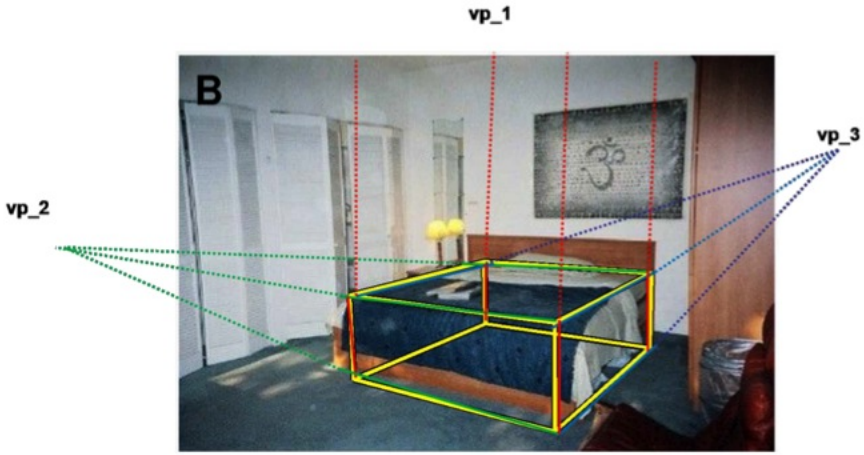
V. Hedau, D. Hoiem, D. Forsyth, Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry, *ECCV* 2010

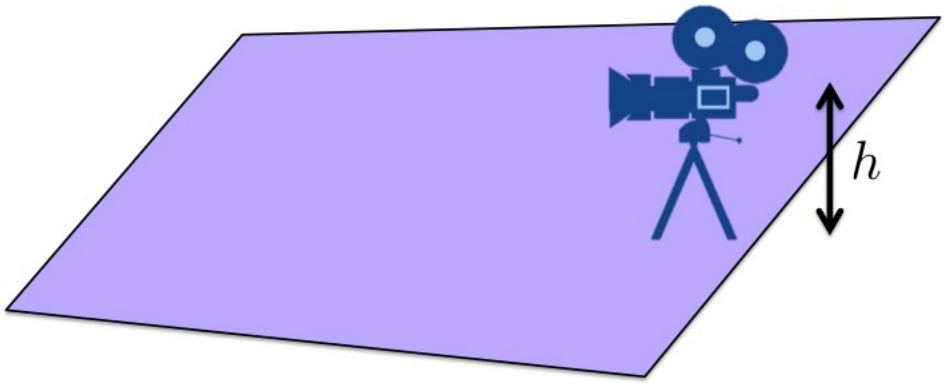
- Object is a box, aligned with the (Manhattan) room



Thinking Inside the Box [Hedau et al., 2010]

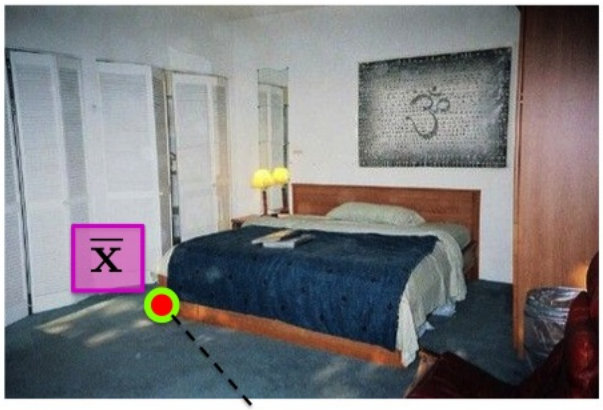
- Object is a box, aligned with the (Manhattan) room
- Assume the camera is distance h above ground

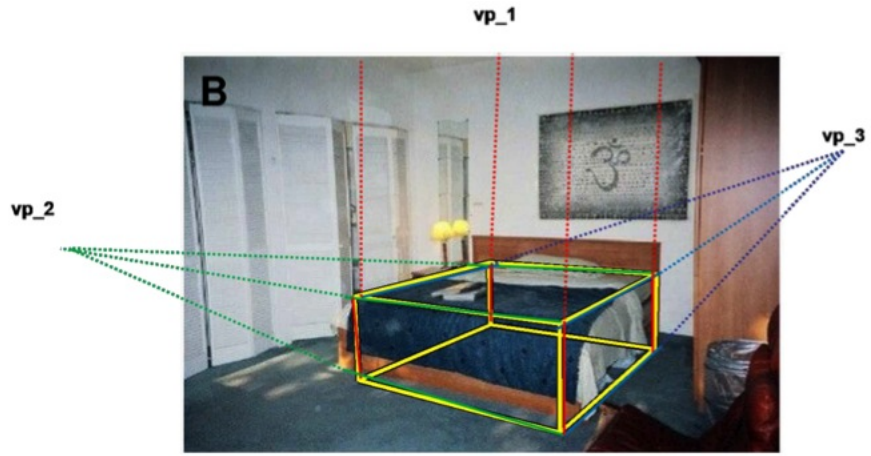




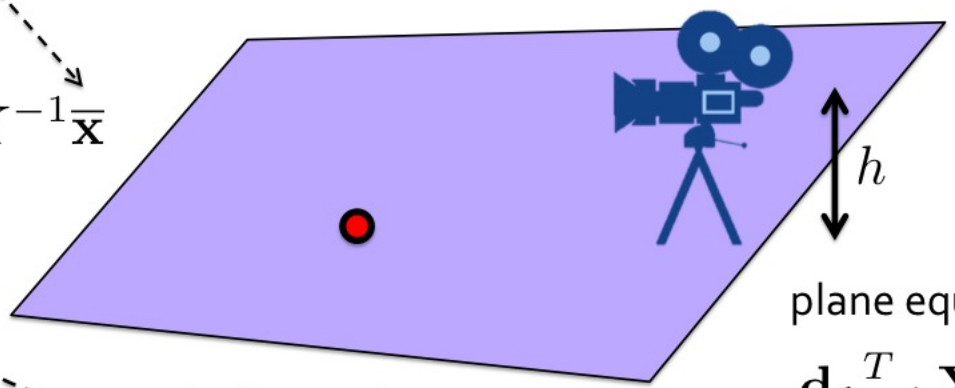
Thinking Inside the Box [Hedau et al., 2010]

- Object is a box, aligned with the (Manhattan) room
- Assume the camera is distance h above ground
- Place a point on the floor





$$\mathbf{X} = \lambda \cdot K^{-1}\bar{\mathbf{x}}$$



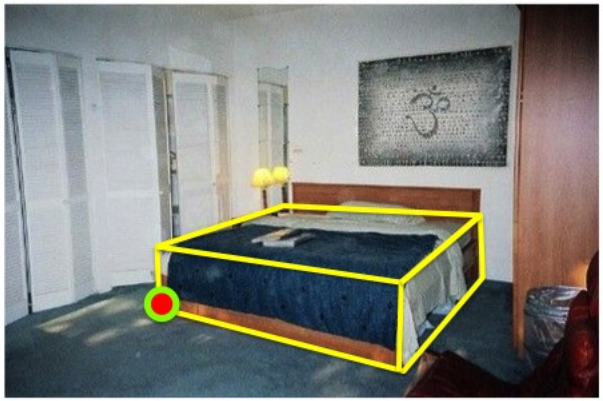
plane equation:

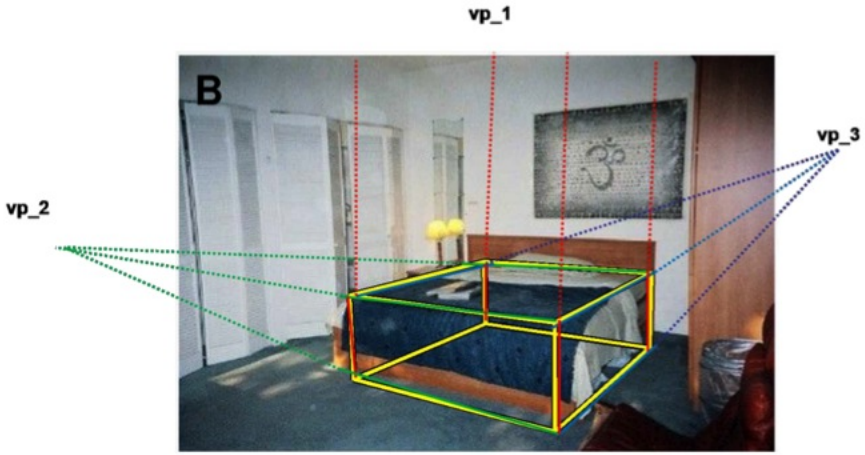
$$\mathbf{d_1}^T \cdot \mathbf{X} + h = 0$$

Plug here and get point in 3D

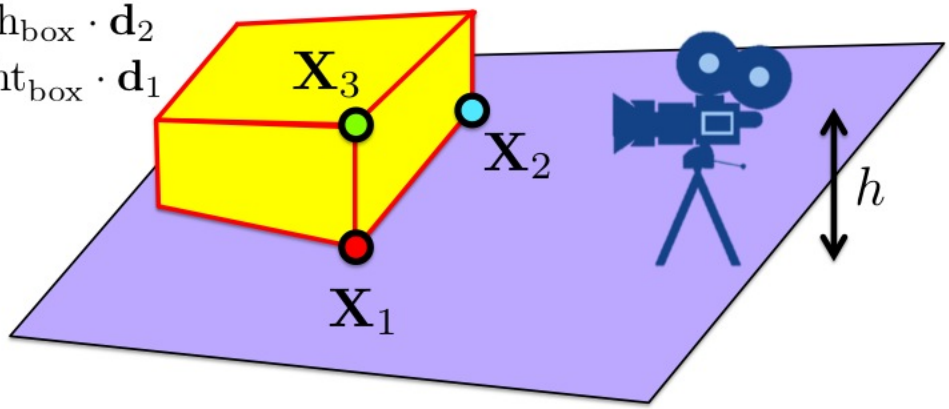
Thinking Inside the Box [Hedau et al., 2010]

- Object is a box, aligned with the (Manhattan) room
- Assume the camera is distance h above ground
- Place a point on the floor, assume box of known physical height



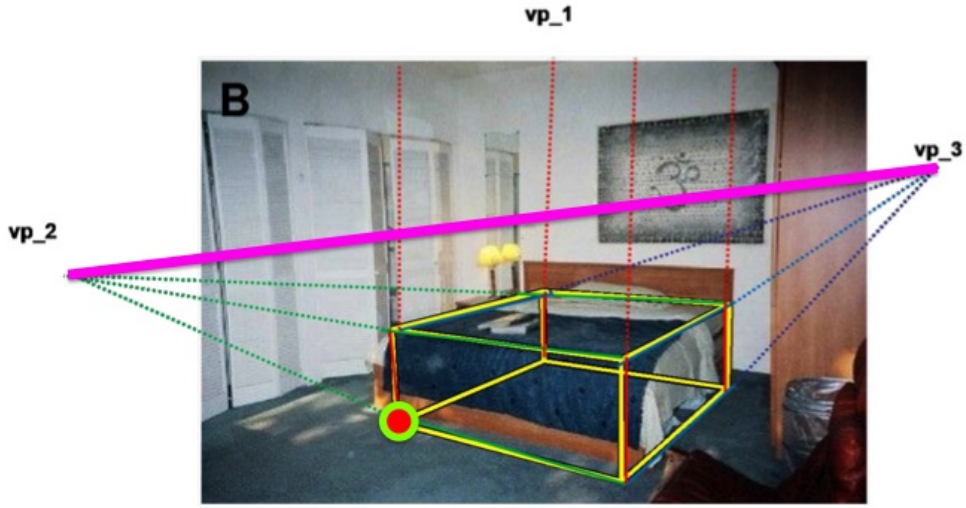


$$\mathbf{X}_2 = \mathbf{X}_1 + \text{width}_{\text{box}} \cdot \mathbf{d}_2$$
$$\mathbf{X}_3 = \mathbf{X}_1 + \text{height}_{\text{box}} \cdot \mathbf{d}_1$$
$$\vdots$$



Thinking Inside the Box [Hedau et al., 2010]

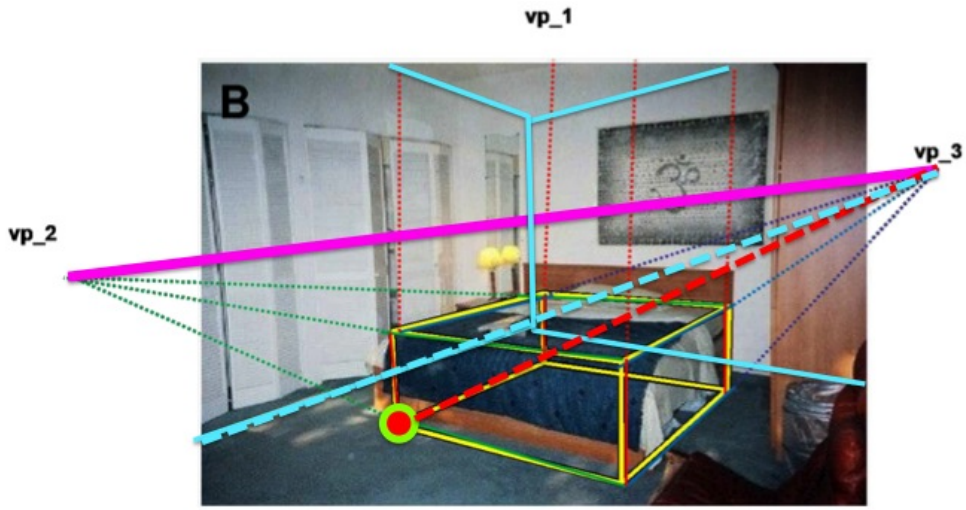
- Object is a box, aligned with the (Manhattan) room
- Assume the camera is distance h above ground
- Place a point on the floor, assume box of known physical height



Place the points only below the horizon

Thinking Inside the Box [Hedau et al., 2010]

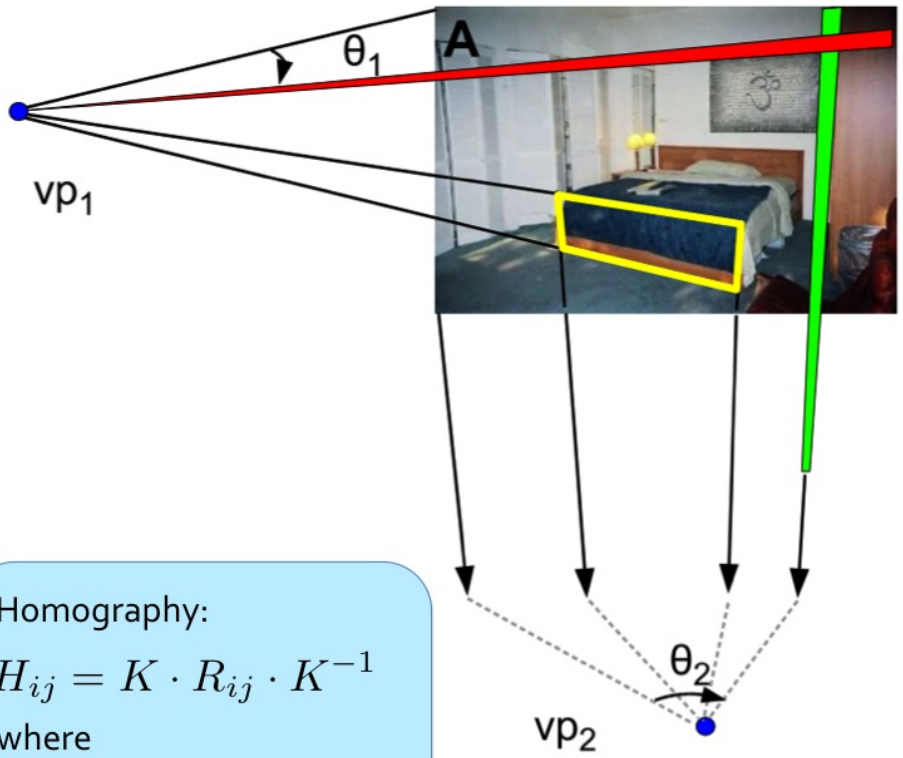
- Object is a box, aligned with the (Manhattan) room
- Assume the camera is distance h above ground
- Place a point on the floor, assume box of known physical height

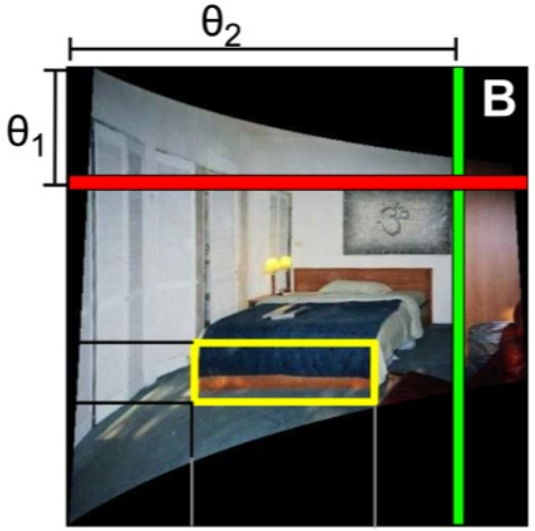


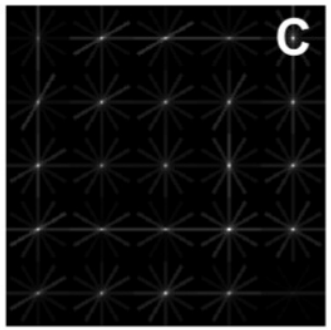
Additional constraints for placing the point when layout is known
(object cannot penetrate the walls)

Thinking Inside the Box [Hedau et al., 2010]

- Object is a box, aligned with the (Manhattan) room
- Assume the camera is distance *h* above ground
- Place a point on the floor, assume box of known physical height
- Score each face in fronto-parallel coordinates







Homography:

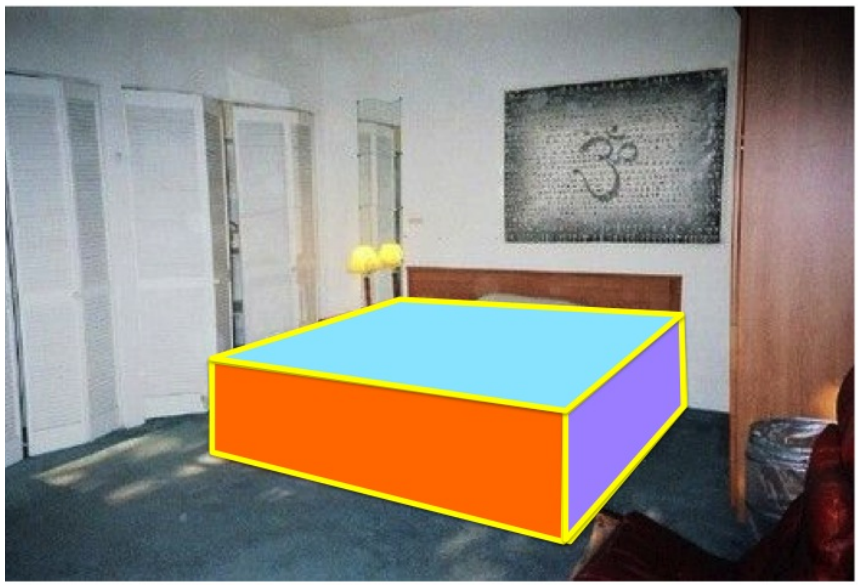
$$H_{ij} = K \cdot R_{ij} \cdot K^{-1}$$

where

$$R_{ij} = [\mathbf{d}_i, \mathbf{d}_j, \mathbf{d}_i \times \mathbf{d}_j]$$

Thinking Inside the Box [Hedau et al., 2010]

- Object is a box, aligned with the (Manhattan) room
- Assume the camera is distance h above ground
- Place a point on the floor, assume box of known physical height
- Score each face in fronto-parallel coordinates
- Score a box by summing the scores of the visible faces



$$\text{score}(\text{box}) = \frac{\sum_i v_i \cdot \max_{f \in \mathcal{N}(f_i)} \text{sc}(f_i)}{\sum_i v_i}$$

Thinking Inside the Box [Hedau et al., 2010]

Inference:

- Object is a box, aligned with the (Manhattan) room
- Assume the camera is distance h above ground
- Place a point on the floor, assume box of known physical height
- Score each face in fronto-parallel coordinates
- Score a box by summing the scores of the visible faces

Training:

- Train each face independently using SVM

Thinking Inside the Box [Hedau et al., 2010]

Bedroom dataset:

- Dataset contains 181 train and 128 test images with annotated beds.

Indoor dataset (Hedau et al., CVPR'12):

- 592 indoor images (containing bedroom dataset as subset)
- Annotated: sofas, chairs, tables, and dressers

Thinking Inside the Box

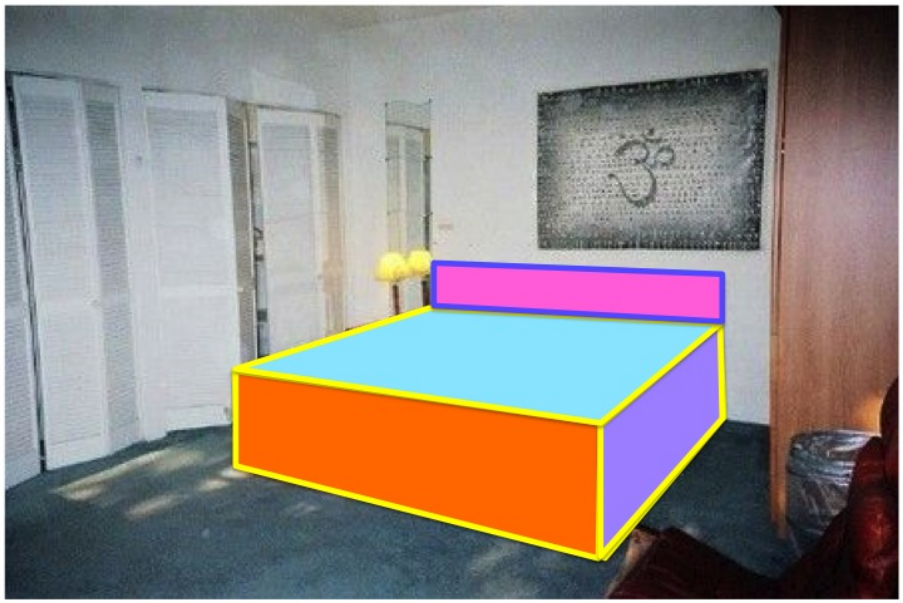
[Hedau et al., 2010]

Method	1.Cuboid detector	2. Felzenszwalb et al.	1+2	1+2+scene layout
Average Precision	0.513	0.542	0.596	0.628

Extension [Hedau et al., 2012]

V. Hedau, D. Hoiem, D. Forsyth, Recovering Free Space of Indoor Scenes from a Single Image, *CVPR* 2012

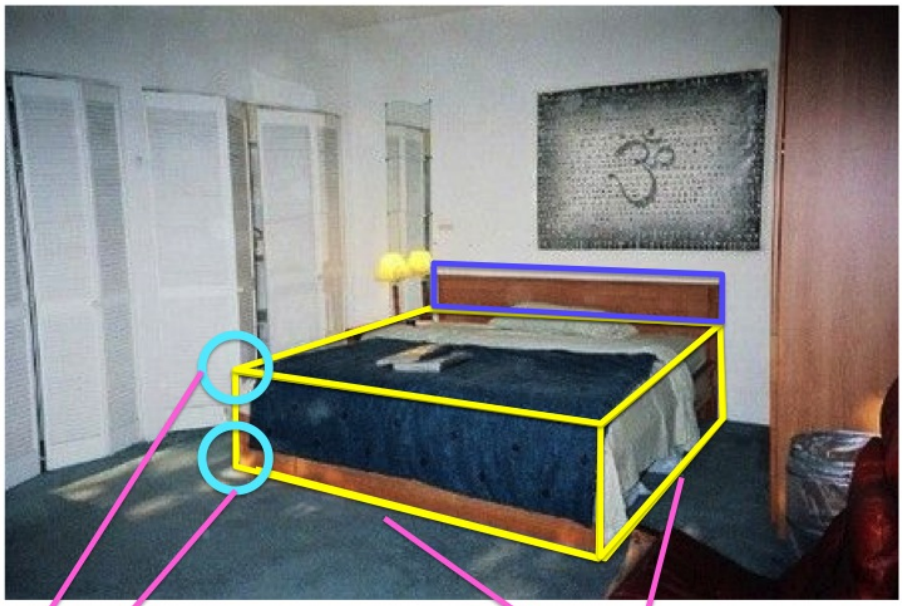
- Adds headrest as a latent variable (scores it only if the overall score increases)



Extension

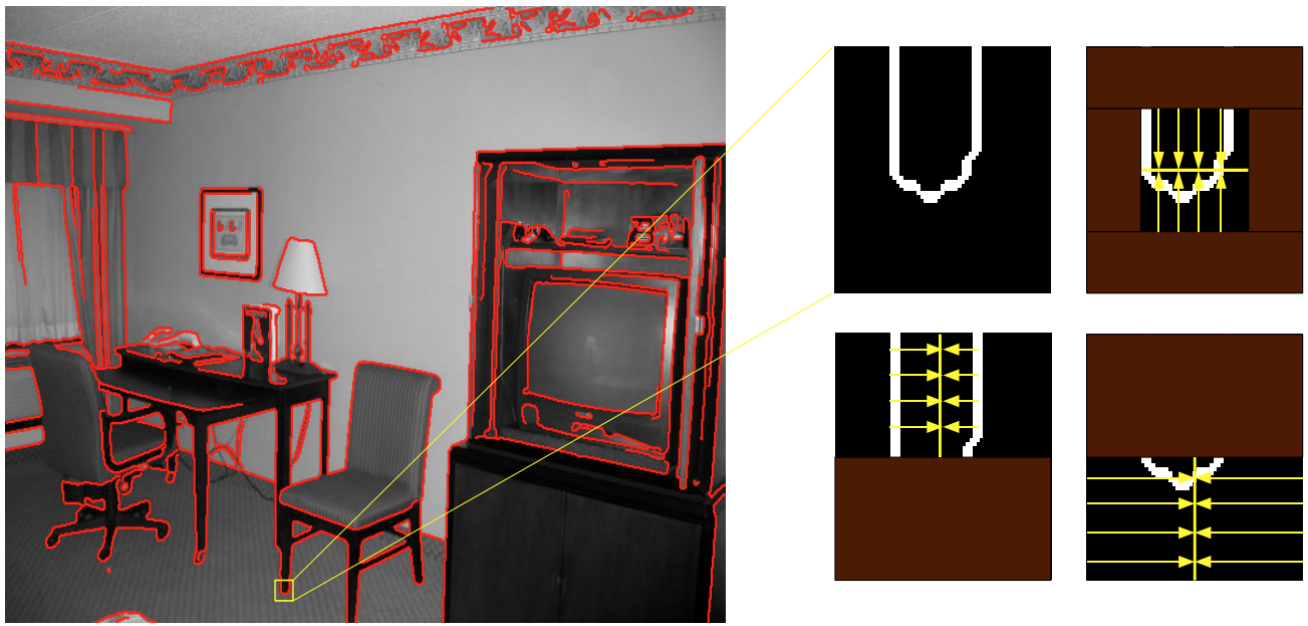
[Hedau et al., 2012]

- Adds headrest as a latent variable (scores it only if the overall score increases)
- Relocalizes the box more precisely via several cues:
  - Edge-based features (line segments) on the cuboid edges
  - Corner-based features (Harris cornerness measure) on cuboid corners



Extension [Hedau et al., 2012]

- Adds headrest as a latent variable (scores it only if the overall score increases)
- Relocalizes the box more precisely via several cues:
  - Edge-based features (line segments) on the cuboid edges
  - Corner-based features (Harris cornerness measure) on cuboid corners
  - "Peg" detector

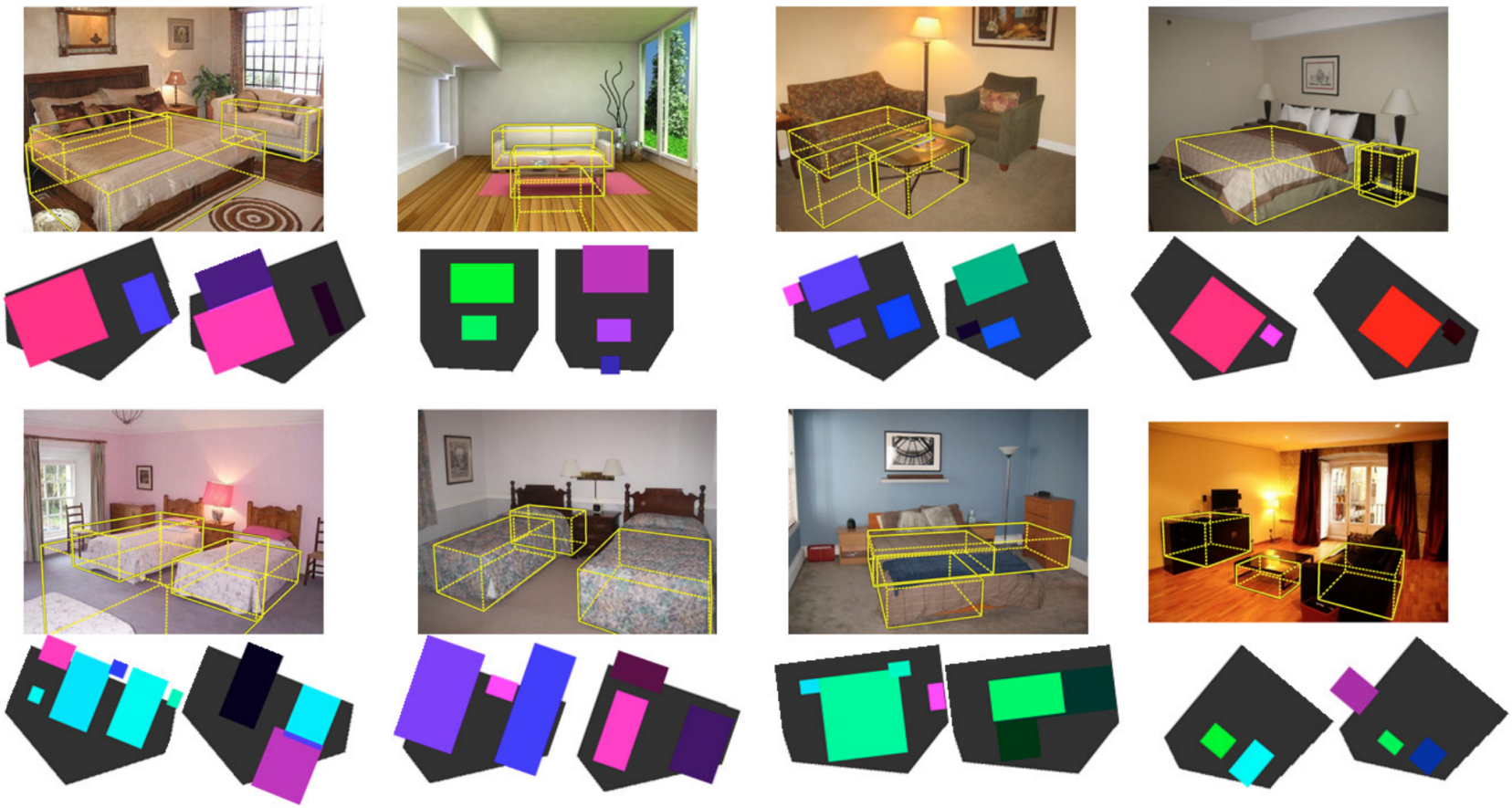


Results

[Hedau et al., 2012]

Results

[Hedau et al., 2012]



Precision (at recall)	Floor occupancy	3D voxels
Gupta et al. [6]	0.48 (0.48)	0.08 (0.25)
Ours	0.74 (0.48)	0.49 (0.25)

Deformable 3D Cuboid Model [Fidler et al., 2012]

S. Fidler, S. Dickinson, R. Urtasun, 3D Object Detection and Viewpoint Estimation with a Deformable 3D Cuboid Model, *NIPS* 2012

Represent objects with a **deformable 3D cuboid model**:

- that score parts and spatially relates them to the cuboid faces
- scores visible faces and spatially relates them to the **stitching point**, the intersection point of the visible faces
- reasons about the faces and parts in rectified coordinates
- explicitly reasons about face visibility patterns called **aspects**
- shares appearance models for the faces and parts across aspects

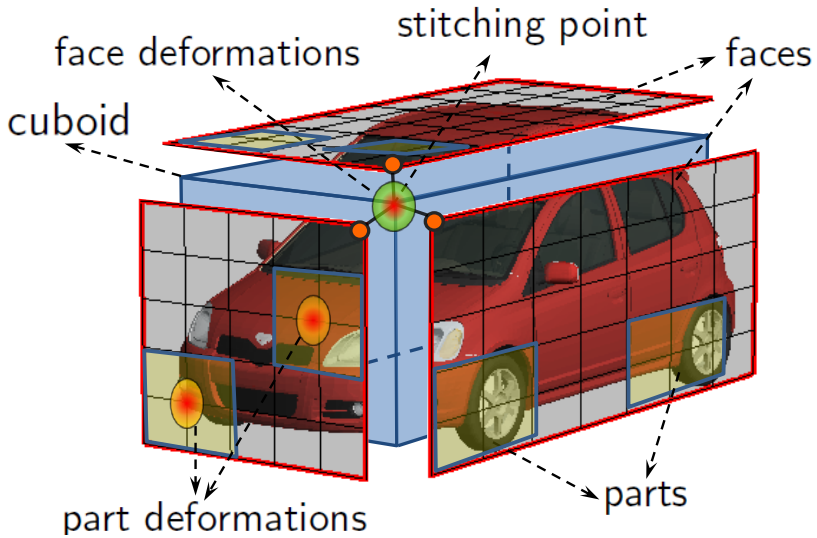


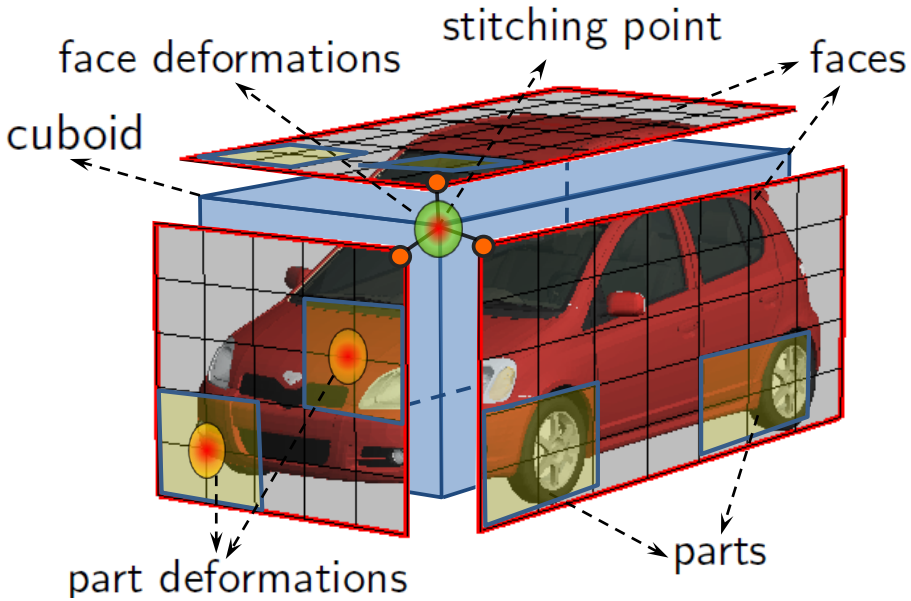
Figure: Aspects: topologically different visibility patterns

Deformable 3D Cuboid Model [Fidler et al., 2012]

- Following Felzenswalb et al, the model is scored as:

$$f_{\mathbf{w}}(x) = \max_{(y,z)} \mathbf{w} \cdot \phi(x, y, z)$$

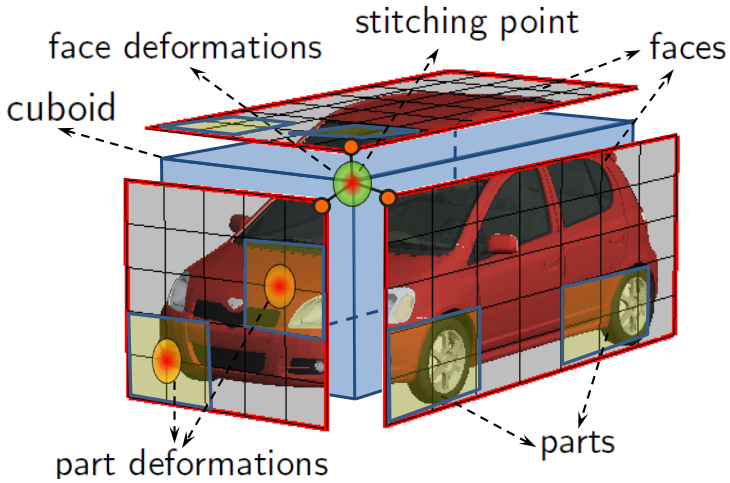
- x ... image features
- $y = \pm 1$
- z ... hypothesis representing angle θ, positions and scales of stitching point, faces and parts
- Reasoning about face visibility via θ, position, scale of 3D bbox

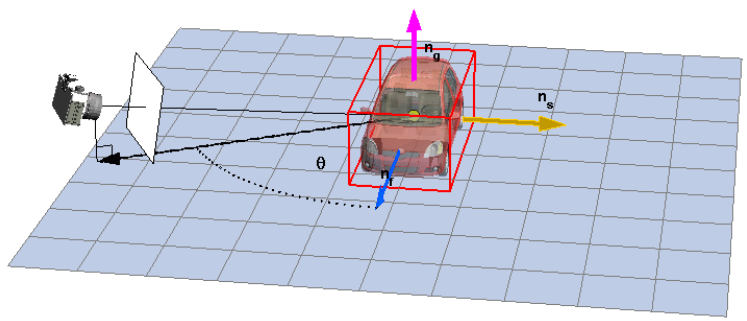


Deformable 3D Cuboid Model [Fidler et al., 2012]

$$score(x, \theta, \mathbf{s}, \mathbf{f}) = \sum_{i=1}^{6} V(i,a) \cdot score_{parts}(\mathbf{f}_i, \theta) +$$
$$+ \sum_{i=1}^{6} V(i,a) \left(score(f_i, \theta) - d_{a,i}^{stitch} \cdot \phi_d^{stich}(f_i, \mathbf{s}, \theta) \right) +$$
$$- \sum_{i>ref}^{6} V(i,a) \cdot d_{i,ref}^{face} \phi_d^{face}(f_i, f_{ref}, \theta) + b_a$$

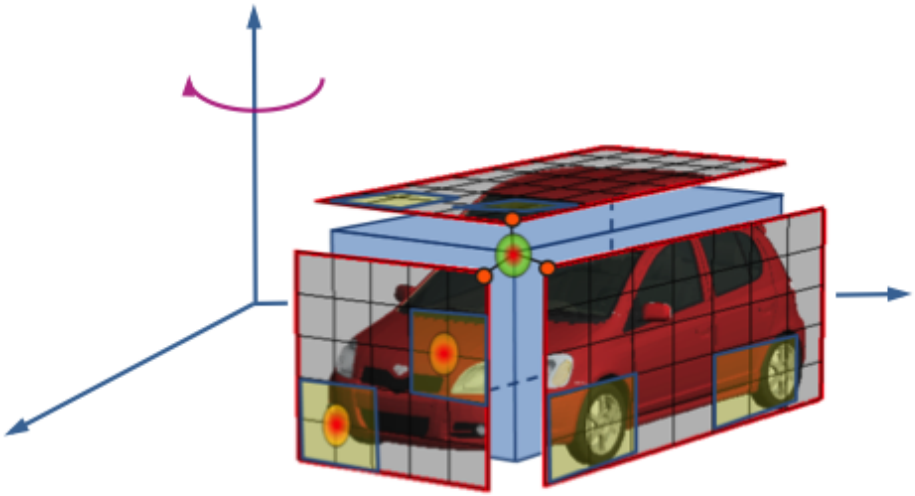
where $V(i,a)$ a binary variable encoding visibility of face i under aspect a.





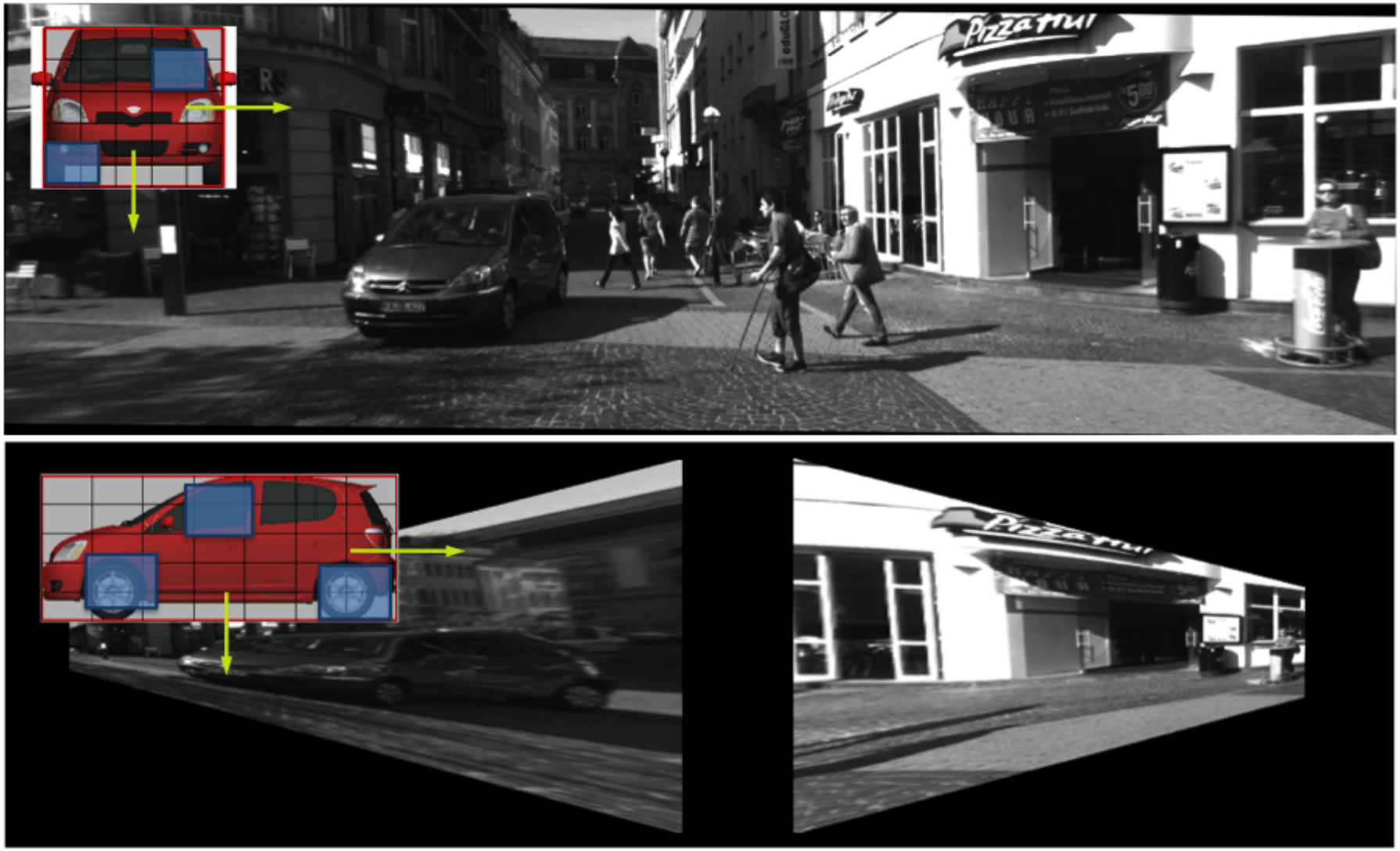
Deformable 3D Cuboid Model [Fidler et al., 2012]

- In inference, the model slides and rotates in 3D



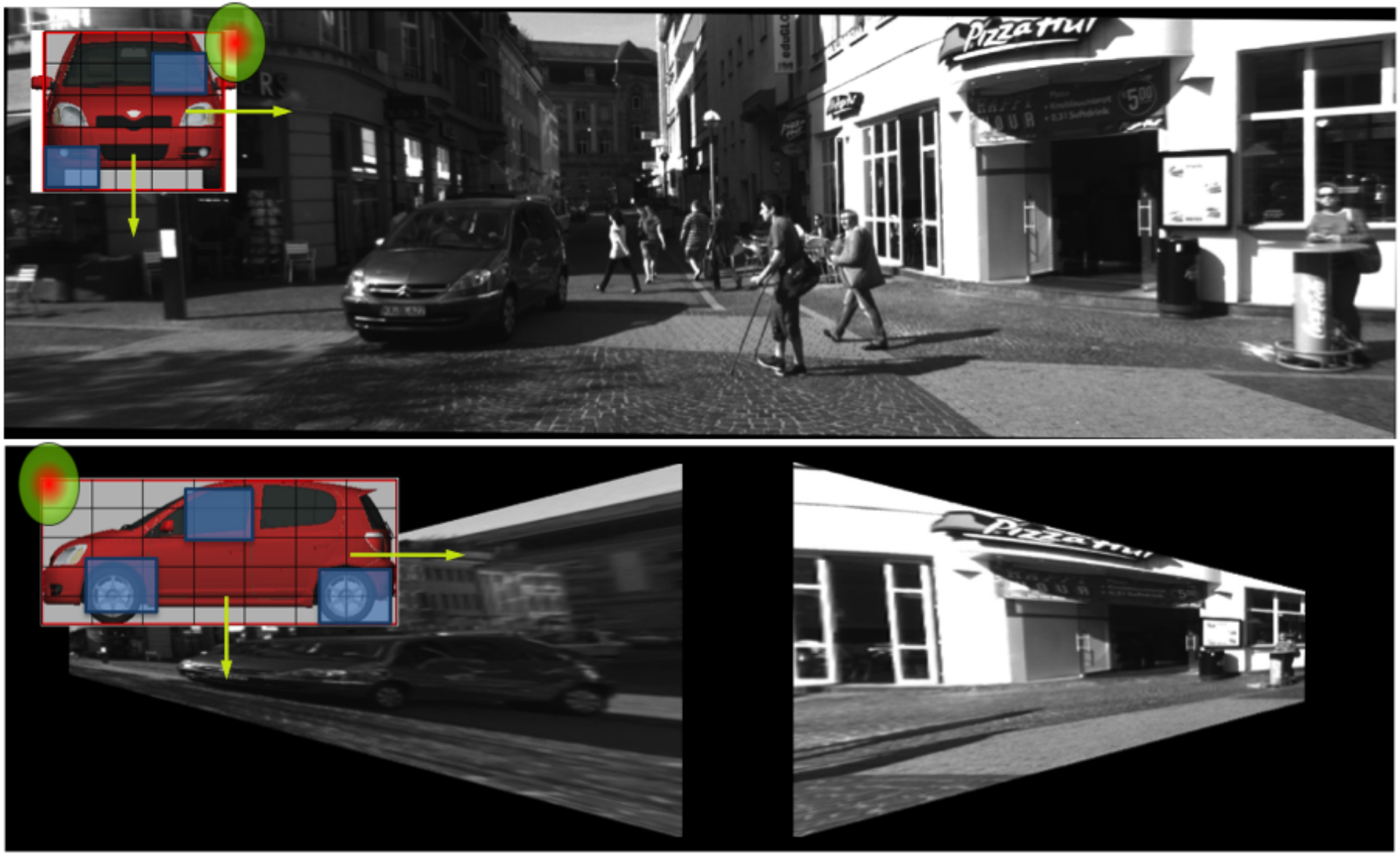
Deformable 3D Cuboid Model [Fidler et al., 2012]

- For each viewpoint, the faces are scored in frontal coords.



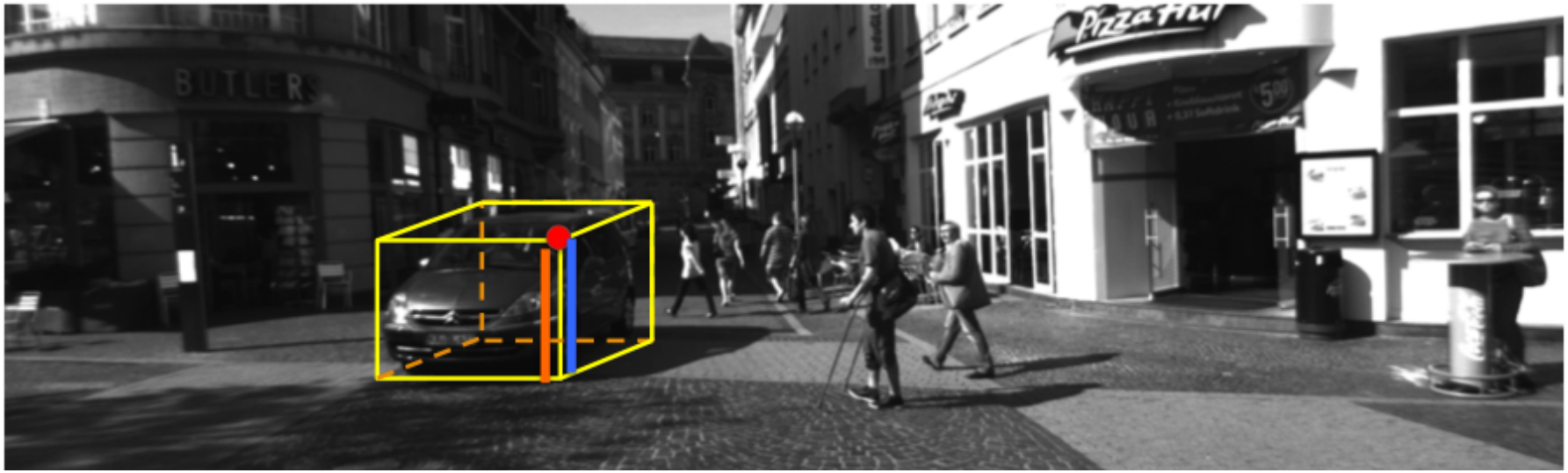
Deformable 3D Cuboid Model [Fidler et al., 2012]

- Compute deformation with respect to stitching point.



Deformable 3D Cuboid Model [Fidler et al., 2012]

- Compute deformations between face sides
- And stitch the hypotheses into a proper deformable cuboid.



- For training the model latent SVM [Felzenswalb et al] is used

Deformable 3D Cuboid Model [Fidler et al., 2012]

- Evaluation on Hedau's bedroom dataset
- Bed model was trained with 5 aspects, 4 faces and two parts per face
- Faces + parts were shared between different aspects

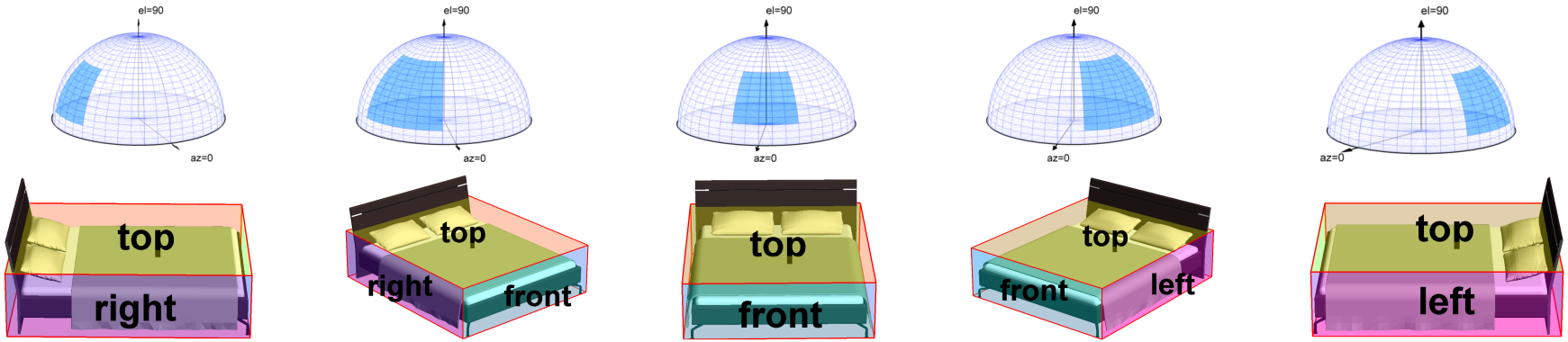


Figure: Aspects, together with the range of θ that they cover.

Deformable 3D Cuboid Model [Fidler et al., 2012]

	Detectors' performance			**Layout rescoring**		
	DPM	3D det.	combined	DPM	3D det.	combined
Hedau et al.	54.2%	51.3%	59.6%	-	-	62.8%
ours	55.6%	59.4%	60.5%	60.0%	64.6%	63.8%

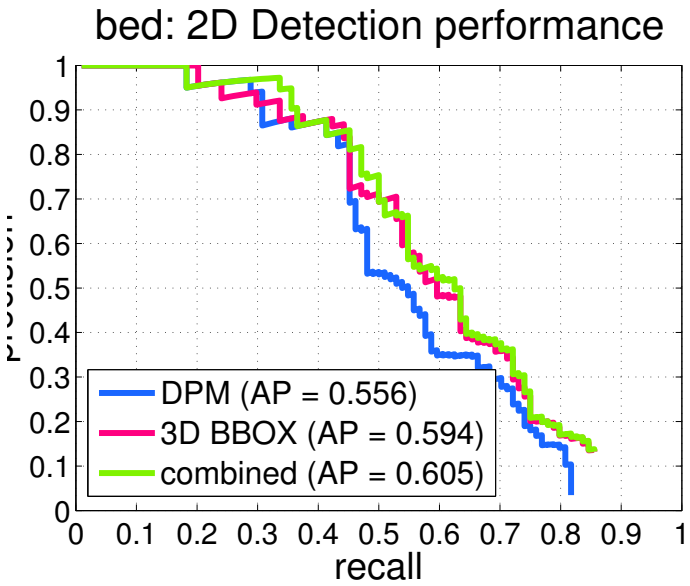
Table: Detection performance (measured in AP at 0.5 IOU overlap) for the bedroom dataset.

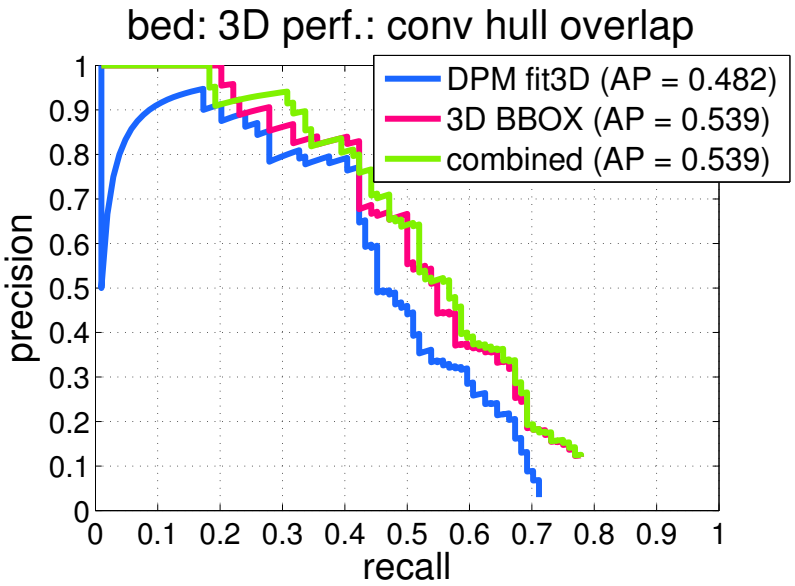
3D measure	DPM fit3D	3D det	comb.	3D det+layout	comb.+layout
convex hull	48.2%	53.9%	53.9%	57.8%	57.1%
face overlap	16.3%	33.0%	34.4%	33.5%	33.6%

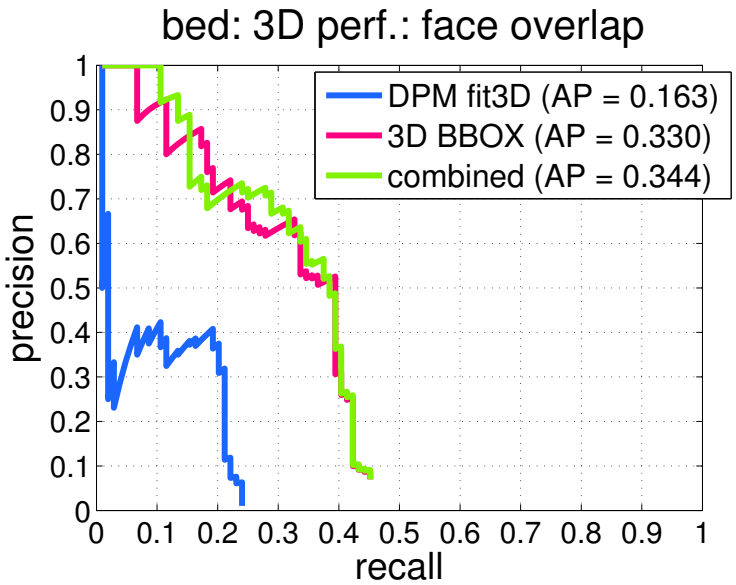
Table: 3D detection performance in AP of predicted and GT boxes)

- **convex hull measure**: convex hulls of our 3D box hypotheses projected to the image plane and groundtruth annotations overlap at least 50% IOU
- **face overlap measure**: average of the overlaps between top faces and vertical faces exceeds 50% IOU

Deformable 3D Cuboid Model [Fidler et al., 2012]



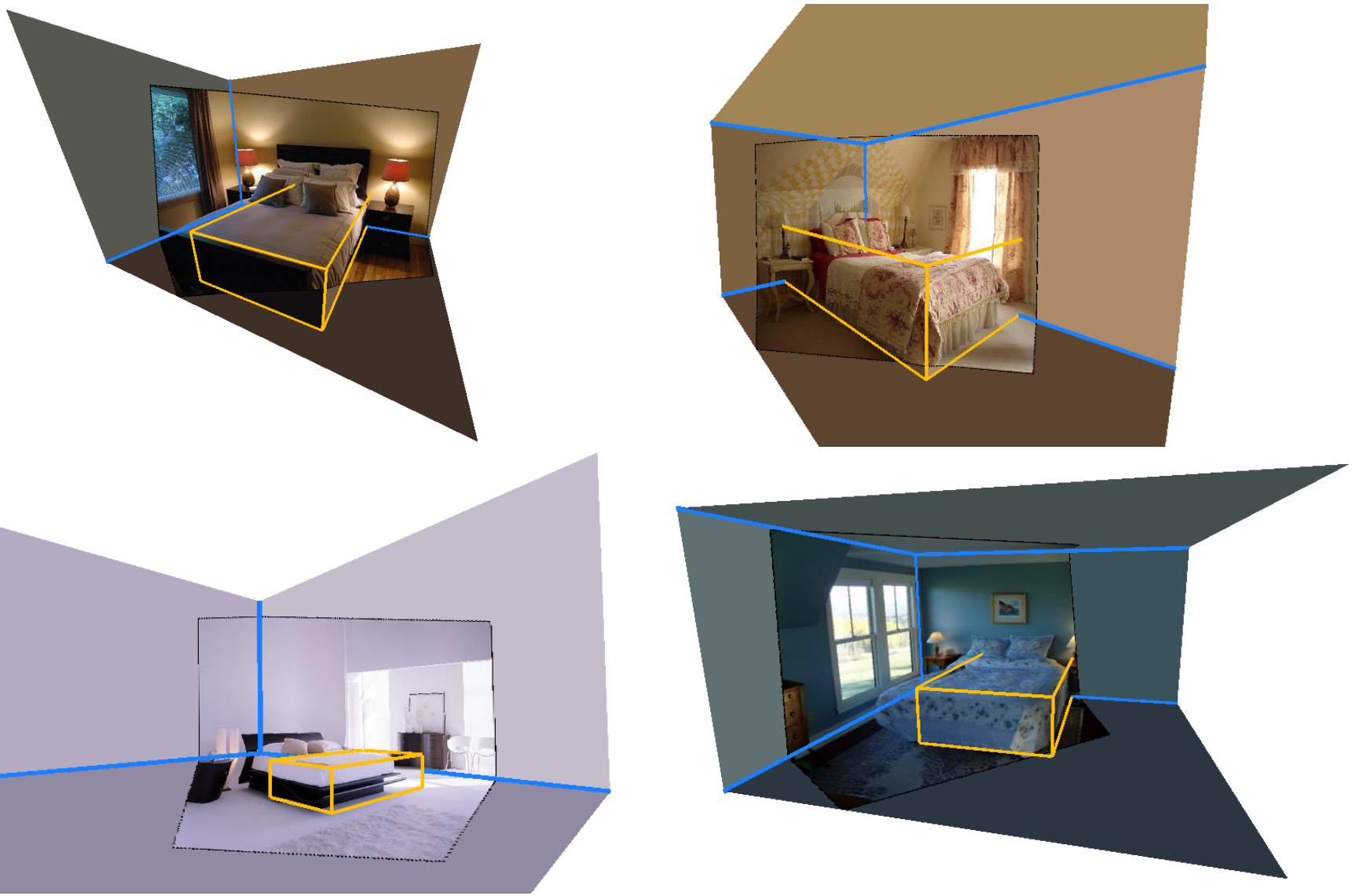




Deformable 3D Cuboid Model

[Fidler et al., 2012]

Deformable 3D Cuboid Model [Fidler et al., 2012]



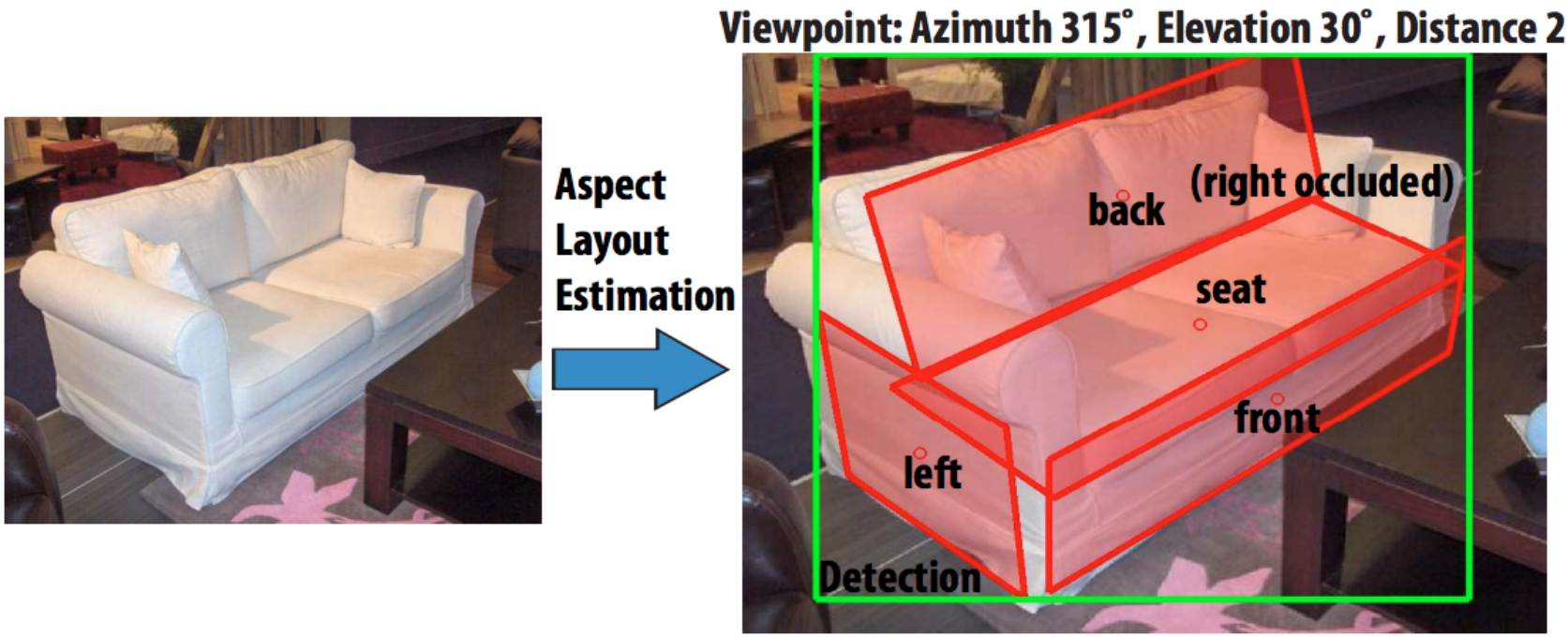
Used room layout estimation from [Schwing and Urtasun, ECCV 2012]

Objects as Compositions of Aspect Parts [Xiang et al., 2012]

Y. Xiang and S. Savarese, Estimating the Aspect Layout of Object Categories, *CVPR* 2012

Code, data: http://wwweb.eecs.umich.edu/vision/projects/ALM/ALMproj.html

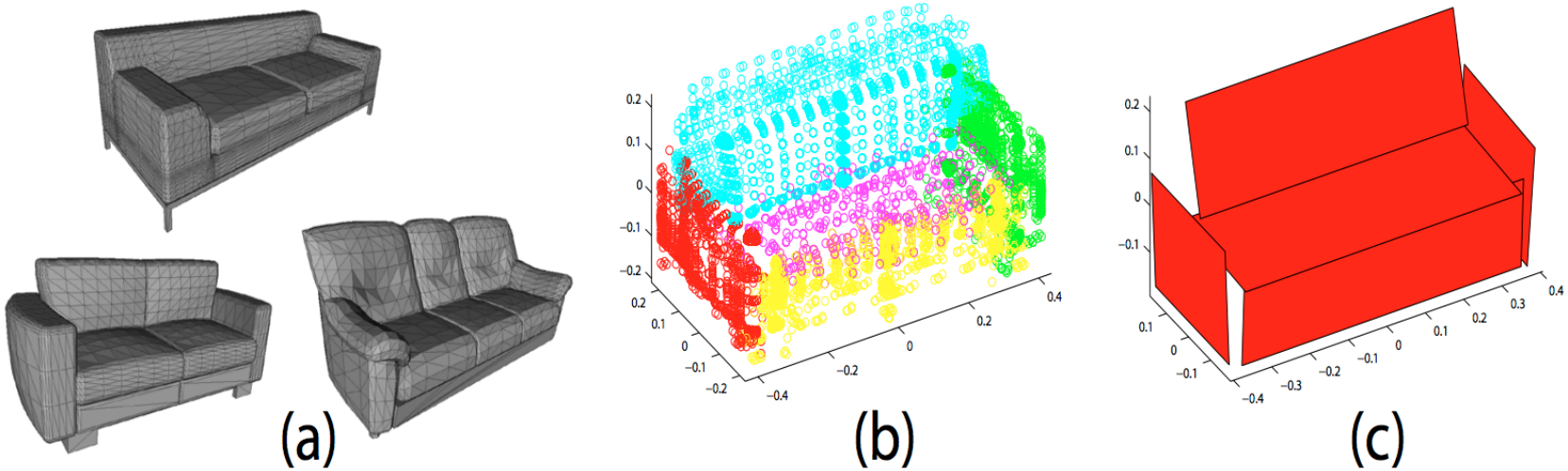
- Objects represented as deformable *aspect parts* (not necessarily orthogonal)
- Aspect parts: surfaces either fully visibly or invisible (e.g. a plane)



Objects as Compositions of Aspect Parts [Xiang et al., 2012]

Obtaining the aspect parts:

- Align the poses and scales of CAD models for a class
- Aggregate the point cloud and manually mark the parts
- Fit planar surfaces (with bounding boxes) to the point cloud of each part



Objects as Compositions of Aspect Parts [Xiang et al., 2012]

- Model the object in each section of a viewpoint sphere as a Conditional Random Field:

$$p(\text{object}, \text{view}) \sim \exp\left(E(\sum_i \mathbf{w}_{u,i}\phi_u(x, \text{part}_i) + \sum_{i,j} \mathbf{w}_{p,i}\phi_p(\text{part}_i, \text{part}_j)\right)$$

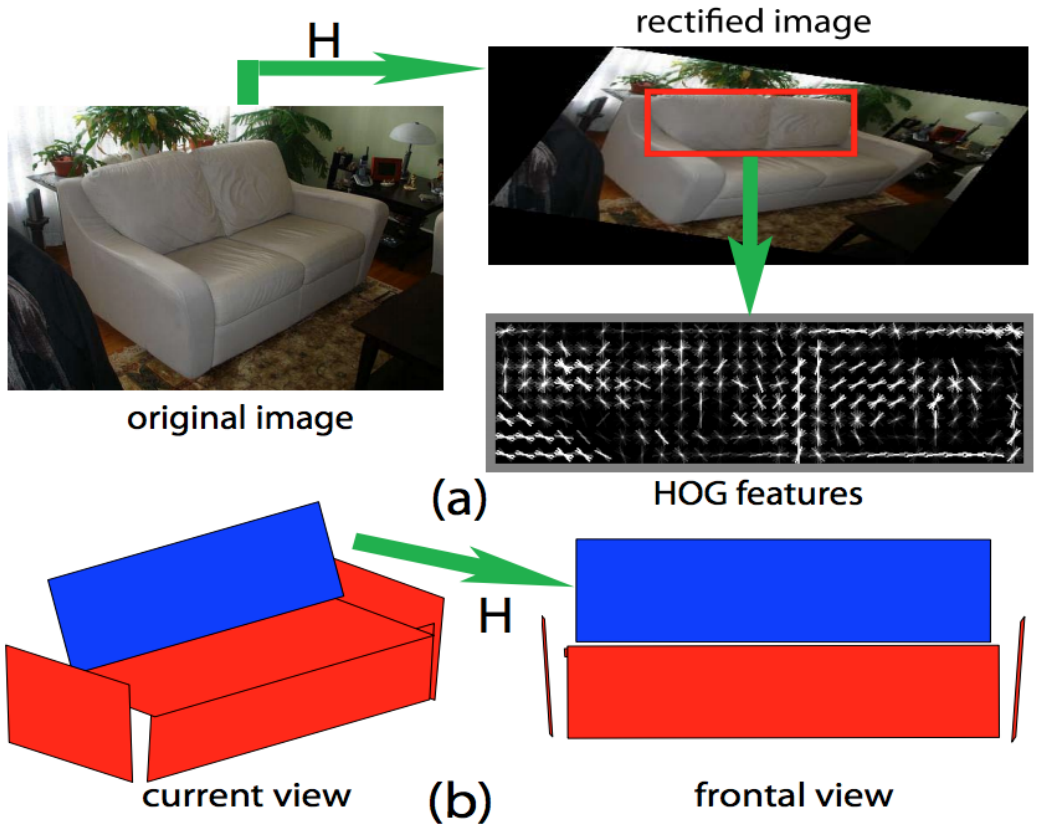
Objects as Compositions of Aspect Parts [Xiang et al., 2012]

- Model the object in each section of a viewpoint sphere as a Conditional Random Field:

$$p(\text{object}, \text{view}) \sim \exp\left(E(\sum_i \mathbf{w}_{u,i}\phi_u(x, \text{part}_i) + \sum_{i,j} \mathbf{w}_{p,i}\phi_p(\text{part}_i, \text{part}_j)\right)$$

Unary potential:

- score each aspect part in frontal view



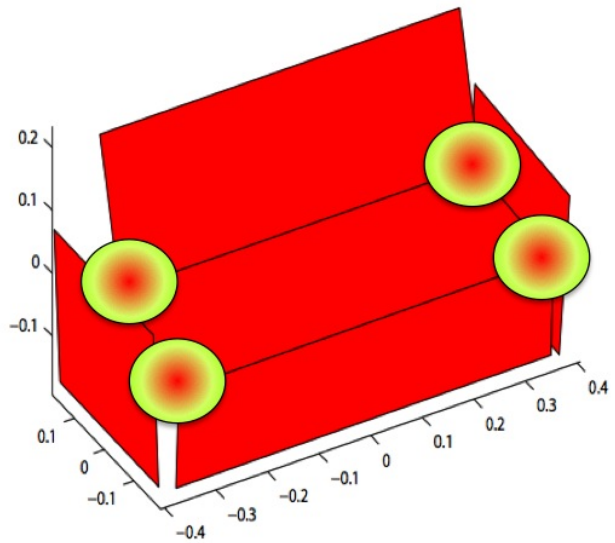
Objects as Compositions of Aspect Parts [Xiang et al., 2012]

- Model the object in each section of a viewpoint sphere as a Conditional Random Field:

$$p(\text{object}, \text{view}) \sim \exp\left(E(\sum_i \mathbf{w}_{u,i}\phi_u(x, \text{part}_i) + \sum_{i,j} \mathbf{w}_{p,i}\phi_p(\text{part}_i, \text{part}_j)\right)$$

Pairwise potentials:

- score deformations between pairs of parts
- part dependency forms a *tree*



Objects as Compositions of Aspect Parts [Xiang et al., 2012]

- Model the object in each section of a viewpoint sphere as a Conditional Random Field:

$$p(\text{object}, \text{view}) \sim \exp\left(E(\sum_i \mathbf{w}_{u,i}\phi_u(x, \text{part}_i) + \sum_{i,j} \mathbf{w}_{p,i}\phi_p(\text{part}_i, \text{part}_j)\right)$$

- **Inference:** Dynamic programming
- **Learning:** Structure SVM

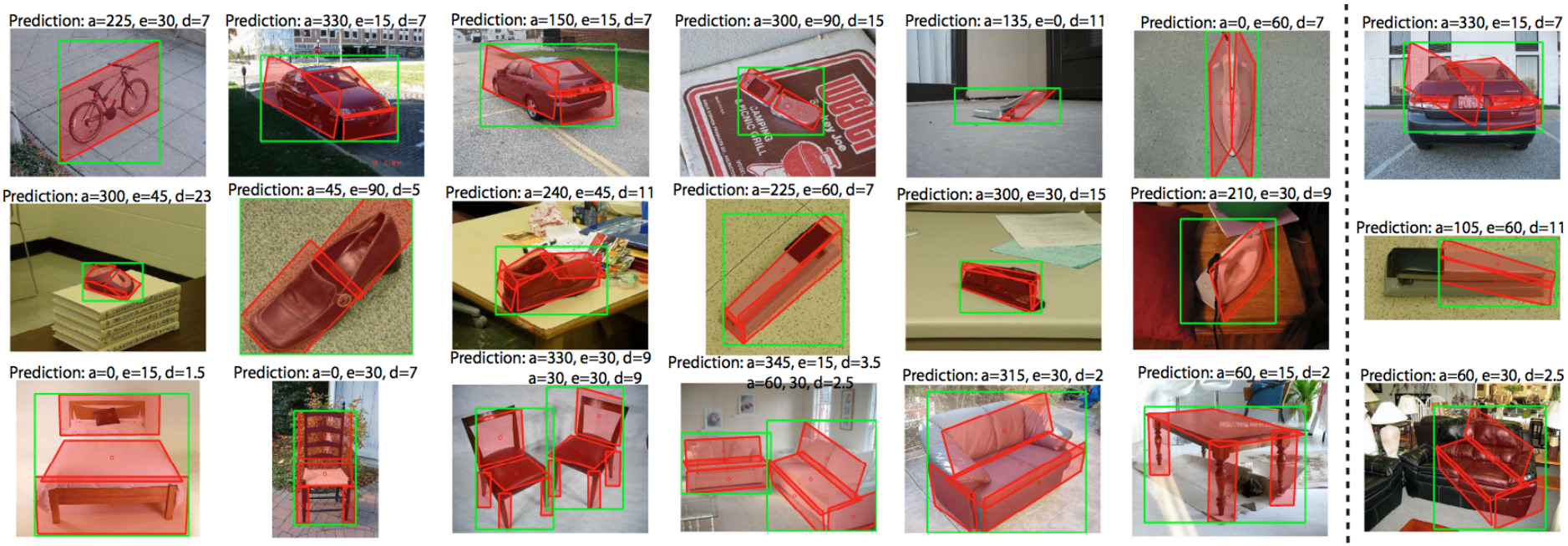
Objects as Compositions of Aspect Parts [Xiang et al., 2012]

Table 1. Results on the 3DObject dataset and the VOC2006 Car dataset.

Dataset	3DObject (8 views)			VOC2006 Car (4 views)		
Method	ALM	[17]	[29]	ALM	[17]	[32]
Viewpoint	**80.7**	74.2	57.2	**85.9**	85.7	73.0
Detection	81.8	n/a	n/a	48.7	**51**	35

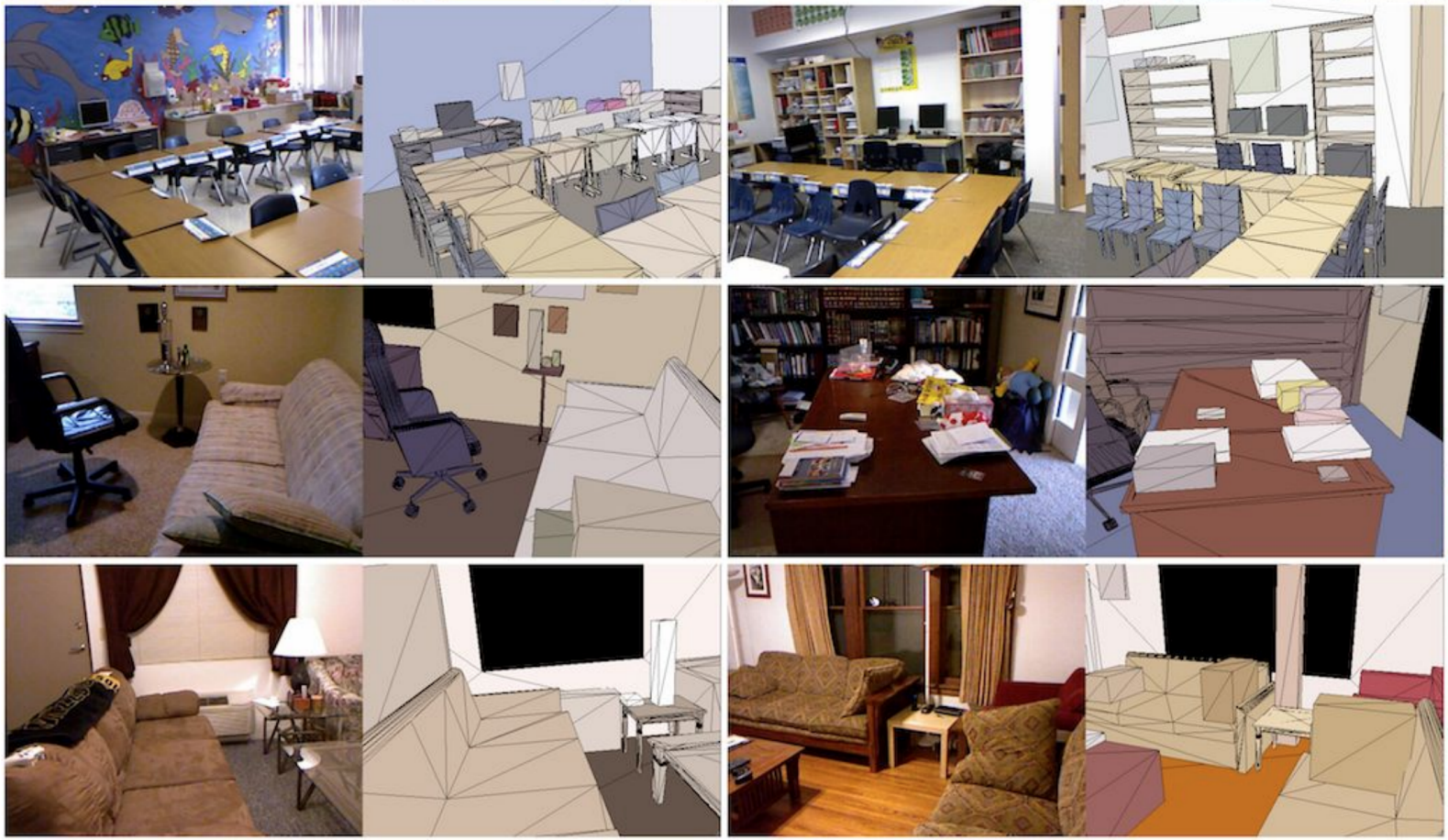
Table 3. Average viewpoint accuracy on the 3DObject dataset.

Category	Bicycle	Car	Cellphone	Iron	Mouse	Shoe	Stapler	Toaster	*Mean*
DPM [13]	88.4	85.0	62.1	82.7	40.0	71.7	58.5	55.0	67.9
ALM Root	**92.5**	89.2	83.4	**86.0**	58.7	82.7	69.2	59.6	77.7
ALM Full	91.4	**93.4**	**85.0**	84.6	**66.5**	**87.0**	**72.8**	**65.2**	**80.7**



Predicting the Full Extent of Objects

- Get a detailed description of objects, going beyond what's visible
- Predict accurate viewpoint, style, full extent of objects



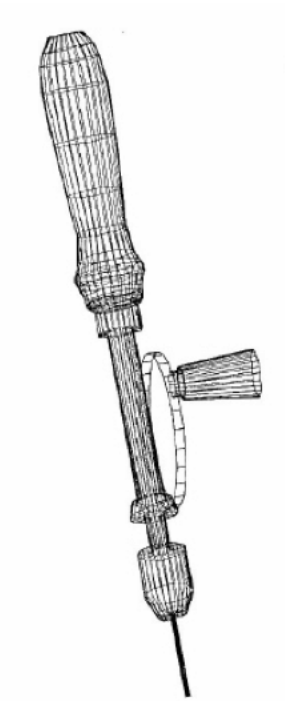
[Guo, Hoiem, Support surface prediction in indoor scenes, *ICCV* 2013]

motivation video by Efros et al.

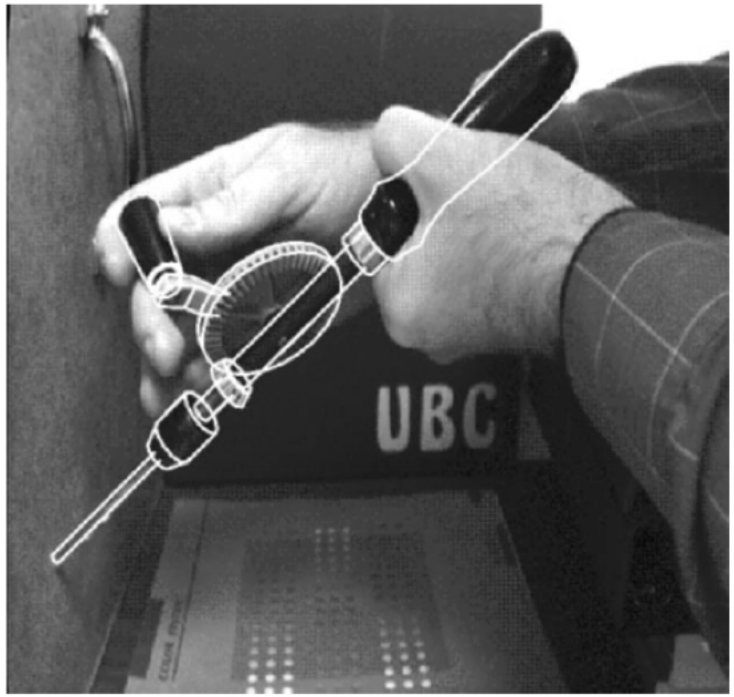
Fitting CAD Models

Goal: Match known detailed 3D CAD model to image:

- Before: Do some grouping on the image side to get corners, lines, etc
- Before: match **one** known 3D model to the image evidence



3D Model



Alignment

Refs: Dickinson, Lowe, Huttenlocher, etc

Fitting CAD Models

- Now: 3D Warehouse (https://3dwarehouse.sketchup.com/) has millions of accurate CAD models of objects. 8,375 search results for query "IKEA".

Figure: http://ikea.csail.mit.edu/

Fitting CAD Models

- 127,915 CAD models for 662 object categories in **modelnet**

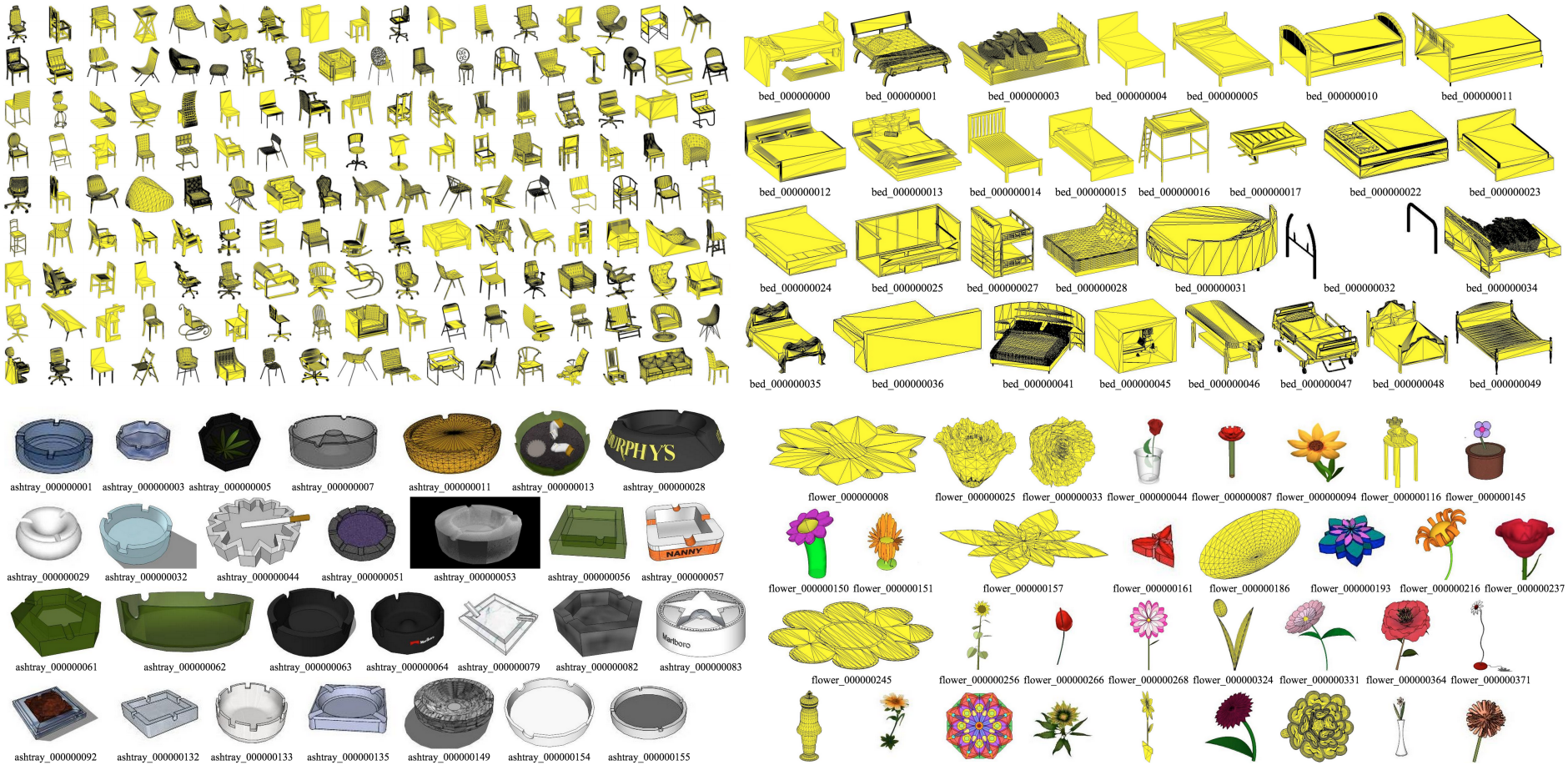


Figure: http://modelnet.cs.princeton.edu/

Fitting CAD Models

- **Idea:** Train classifiers and learn which local patches can be reliably detected for each 3D model.
- Refs: [Lim et al., ICCV 2013], [Aubry et al., CVPR 2014]

Parsing IKEA Objects [Lim et al., 2013]

J. J. Lim, H. Pirsiavash, Antonio Torralba. Parsing IKEA Objects: Fine Pose Estimation. ICCV'13]

Data: http://ikea.csail.mit.edu/

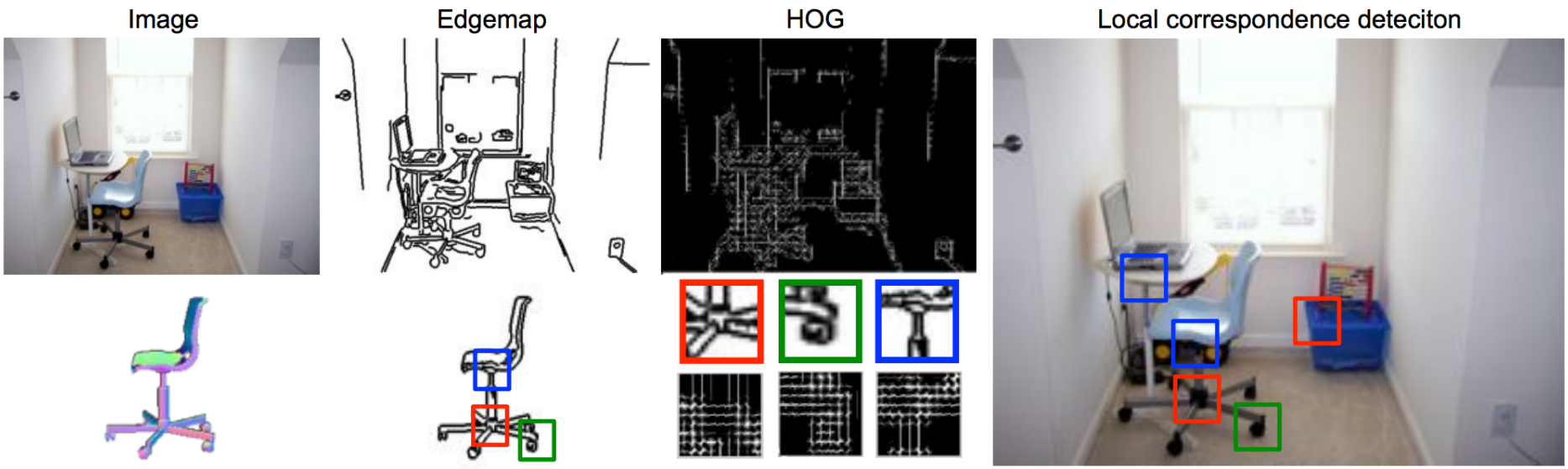


Figure 2. **Local correspondence:** for each 3D interest point $\mathbf{X}_i$ (red, green, and blue), we train an LDA patch detector on an edgemap

- Train an LDA classifier for each local patch, find discriminative patches
- Feature space: HOG on edge-map
- Global alignment via global features (agreement on edges, superpixels, texture) and RANSAC-style optimization

Parsing IKEA Objects

[Lim et al., 2013]

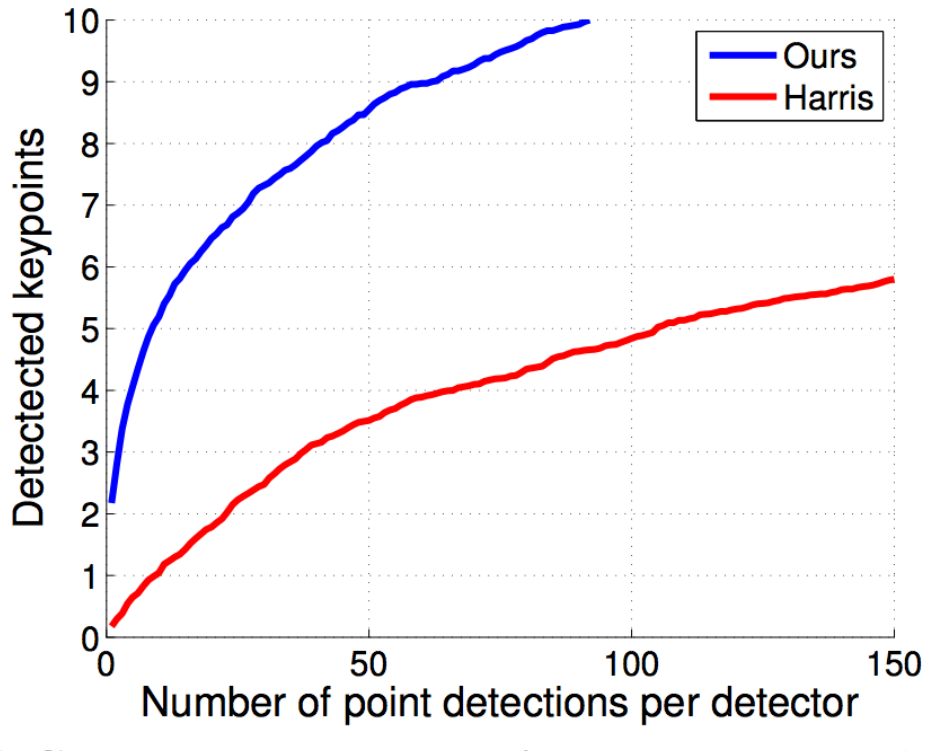


Figure: Learned discriminative patches vs Harris corners

Parsing IKEA Objects

[Lim et al., 2013]

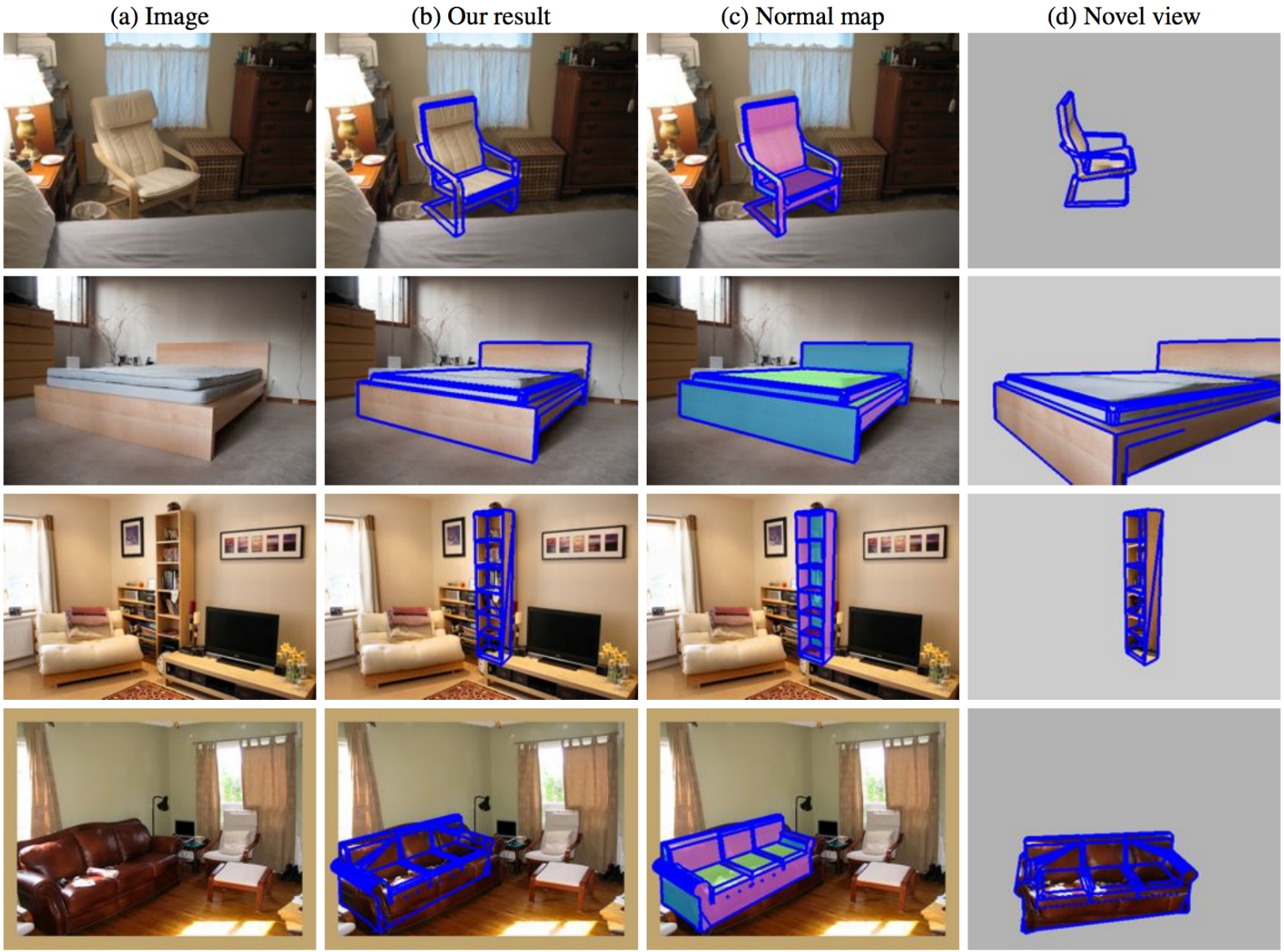


Figure: Results

Parsing IKEA Objects

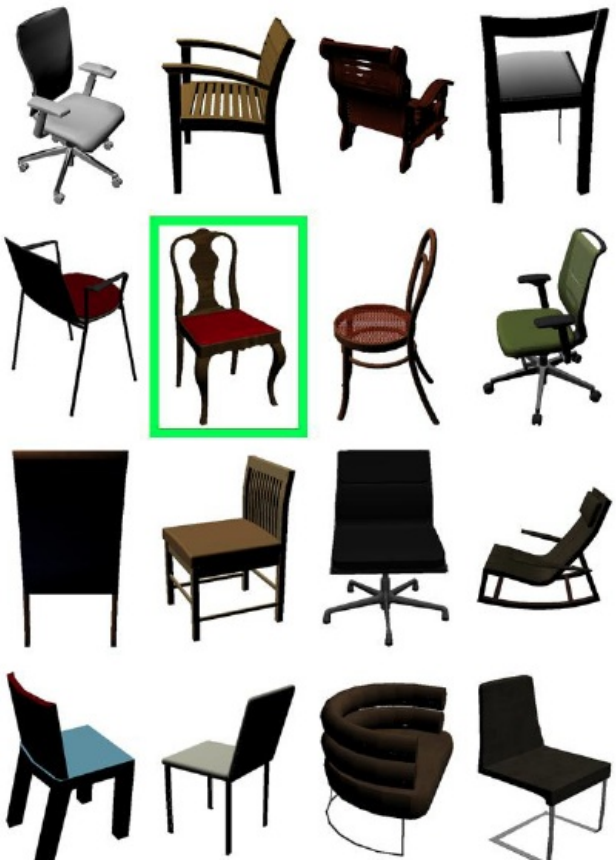
[Lim et al., 2013]

Figure: Some failure modes

Seeing 3D Chairs [Aubry et al., 2010]

M. Aubry, D. Maturana, A. A. Efros, B. Russell, J. Sivic, Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models, *CVPR* 2014

Code, data: http://www.di.ens.fr/willow/research/seeing3Dchairs/



Seeing 3D Chairs

[Aubry et al., 2010]

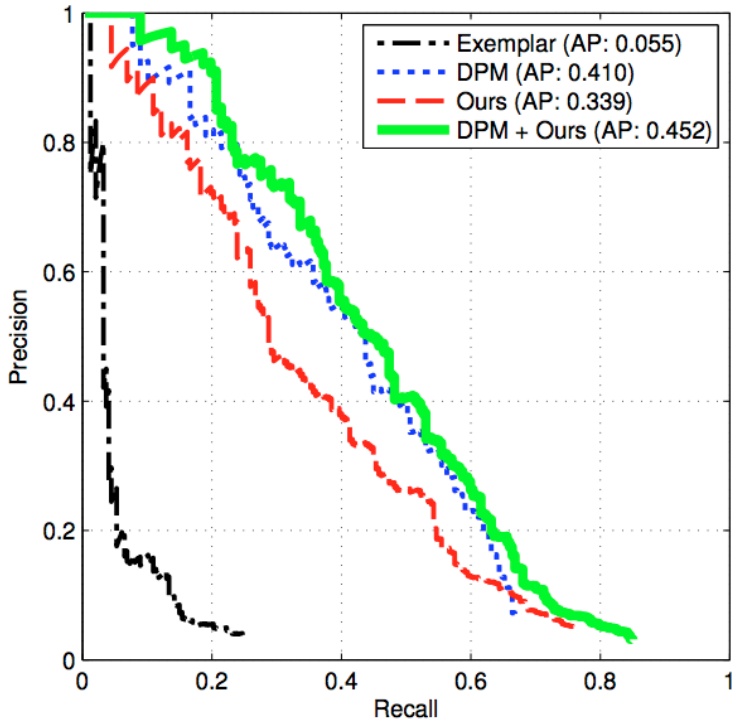
Detection:

(a) Input images (b) DPM [14] output (c) Our aligned outputs (d) Retrieved 3D chairs

Retrieval:

Seeing 3D Chairs [Aubry et al., 2010]

Detection results:



User study:

	Alignment		Style		
	Good	Bad	Good	Ok	Bad
Exemplar-LDA	52%	48%	3%	31%	66%
Ours	90%	10%	21%	64%	15%

CAD Model Datasets

- 219 models of IKEA furniture from 3D Warehouse:
  http://ikea.csail.mit.edu/
- 1,393 *chairs*:
  http://www.di.ens.fr/willow/research/seeing3Dchairs/
- 200 *cars*, 200 *beds*, 296 *sofas*, 90 *tables*, where all models are annotated with viewpoint and aligned:
  http://www.cs.toronto.edu/~fidler/projects/CAD.html
- 128,000 models for 662 categories, where 10 classes (*bathtub*, *bed*, *chair*, *desk*, *dresser*, *monitor*, *night-stand*, *sofa*, *table*, *toilet*) are annotated with viewpoint (aligned up to scale):
  http://modelnet.cs.princeton.edu/

Indoor Object Detection Datasets

- Indoor dataset by Hedau et al., CVPR 2013:

  http://vision.cs.uiuc.edu/~vhedau2/Research/data/indoordataset.zip

- Indoor-Scene-Objects dataset:

  http://wwweb.eecs.umich.edu/vision/3DGP/

- Parsing IKEA dataset (has CAD models aligned with images):

  http://wwweb.eecs.umich.edu/vision/3DGP/

- NYUv2 dataset:

  http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

  Additional annotations:

  http://aqua.cs.uiuc.edu/site/projects/scenemodel.html

- RMRC challenge:

  http://cs.nyu.edu/~silberman/rmrc2014/indoor.php