Part III: Reconstruction, Localization, Semantics in RGB-D

CVPR'15 Tutorial

Sanja Fidler and Raquel Urtasun

June 7, 2015

UNIVERSITY OF TORONTO

Reconstruction / Localization

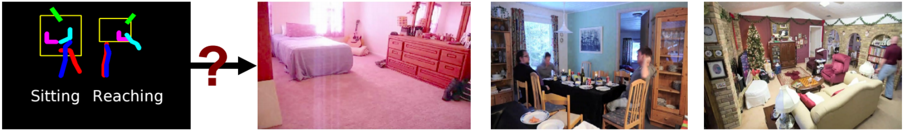
D. F. Fouhey, V. Delaitre, A. Gupta A Efros, I. Laptev, J. Sivic, People Watching: Human Actions as a Cue for Single View Geometry, *ECCV*, 2012

- Exploit human actions and location in time-lapse videos (or single image) to infer functional room geometry (walkable, seatable and reachable surfaces)

Figure: In which room are these people?

D. F. Fouhey, V. Delaitre, A. Gupta A Efros, I. Laptev, J. Sivic, People Watching: Human Actions as a Cue for Single View Geometry, *ECCV*, 2012

- Exploit human actions and location in time-lapse videos (or single image) to infer functional room geometry (walkable, seatable and reachable surfaces)

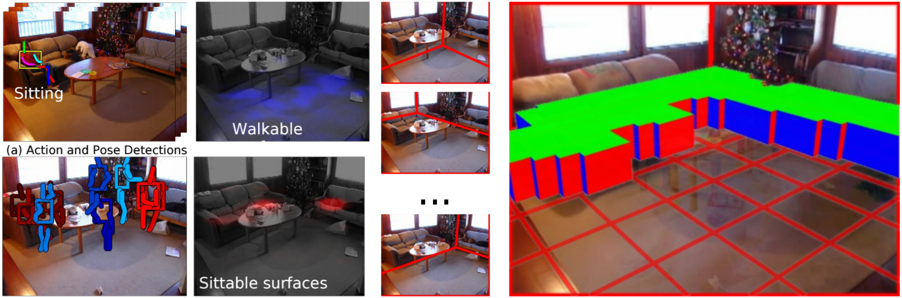


Room A Room B Room C

Figure: In which room are these people?

Answer: Room A

- Detect people and parse their pose
- Infer room layout by imposing that humans are inside the room
- Use layout and human pose to predict the interacting surfaces
- Human pose used to predict *contact* points with the surfaces



- Detect people and parse their pose
- Infer room layout by imposing that humans are inside the room
- Use layout and human pose to predict the interacting surfaces
- Human pose used to predict *contact* points with the surfaces

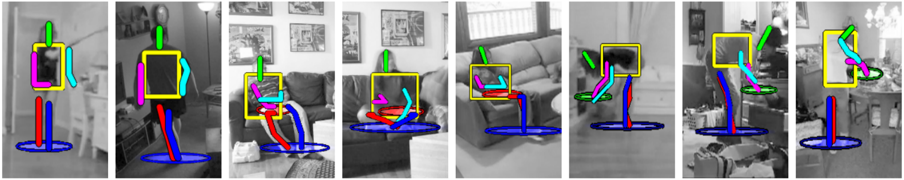


Figure: Poses indicate contact points with the interacting surface.

(a) Appearances Only (Hedau *et al*).

(b) Appearances + People (Our approach).

	Location	Appearance Only		People Only	Appearance + People
		Lee *et al.*	Hedau *et al.*		
Overall	64.1%	70.4 %	74.9%	70.8%	**82.5%**

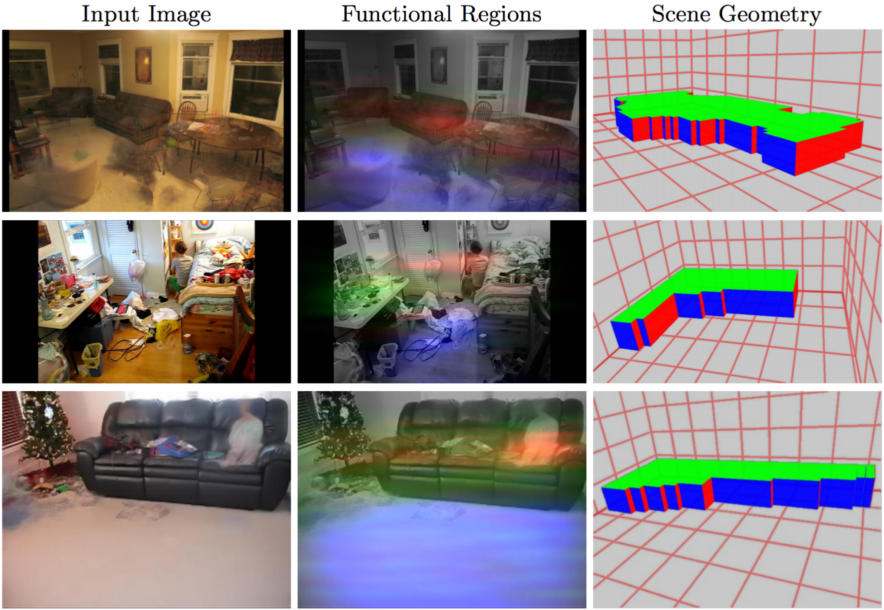
Figure: Time-lapse videos

	Location	Appearance Only		Appearance + People	
		Lee *et al.*	Hedau *et al.*	Ours	with Ground Truth Poses
Overall	66.4%	71.3%	77.0%	79.6%	**80.8%**

Figure: Single image prediction

Scene Geometry via Humans

[Fouhey et al., 2012]

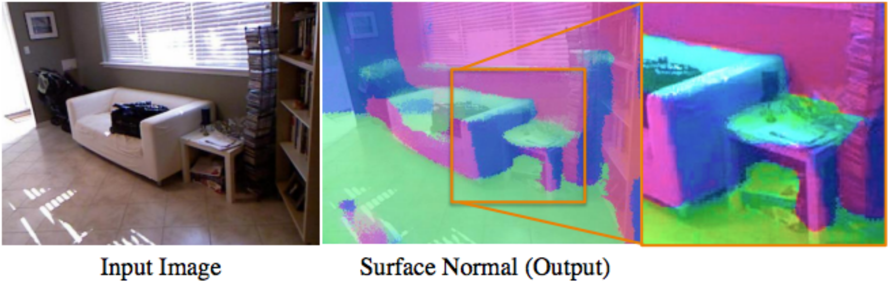


Normals from Single Image

[Wang et al., 2014]

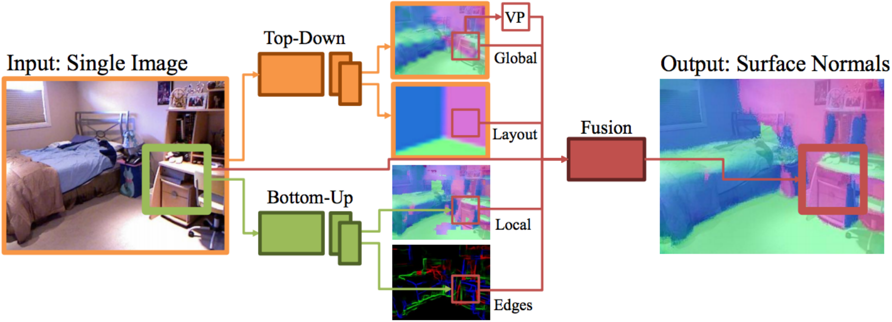
Xiaolong Wang, David F. Fouhey, Abhinav Gupta, Designing Deep Networks or Surface Normal Estimation, *Arxiv*, Nov 2014

- Goal is to predict surface normals from a single image
- For amazing performance use deep learning



Normals from Single Image

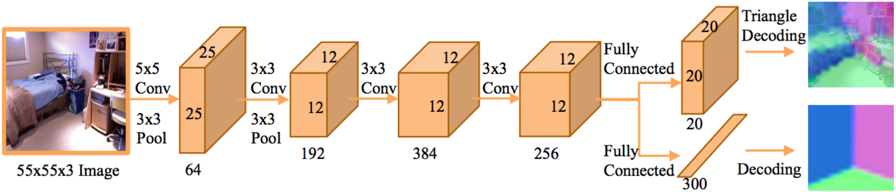
- Train three networks:
  - *Global*: input full image, output coarse normals and layout
  - *Local*: local image patches, output finer normals and edge classification (concave, convex, occlusion)
  - *Fusion*: take a result form both networks and feed it to another network



Normals from Single Image

[Wang et al., 2014]

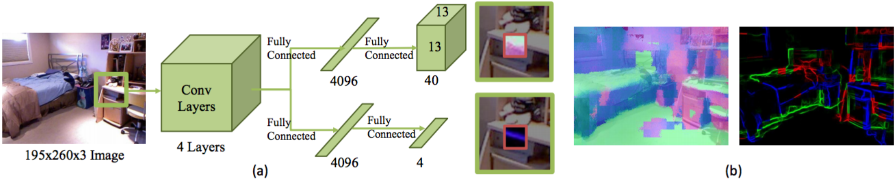
- Train three networks:
  - **Global: input full image, output coarse normals and layout**
  - *Local*: local image patches, output finer normals and edge classification (concave, convex, occlusion)
  - *Fusion*: take a result form both networks and feed it to another network



Normals from Single Image

[Wang et al., 2014]

- Train three networks:
  - *Global*: input full image, output coarse normals and layout
  - **Local: local image patches, output finer normals and edge classification (concave, convex, occlusion)**
  - *Fusion*: take a result form both networks and feed it to another network



Normals from Single Image

[Wang et al., 2014]

Input | Ground Truth | Output | Input | Ground Truth | Output

Input Image | Output | Input Image | Output | Input Image | Output

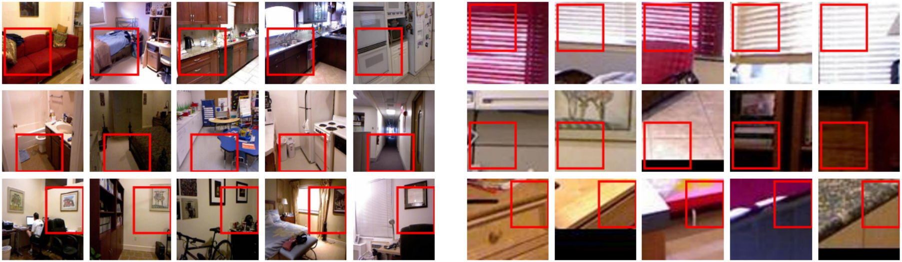


Table 1: Results on NYU v2 for per-pixel surface normal estimation, evaluated over valid pixels.

	Summary Stats. (°) (Lower Better)			% Good Pixels (Higher Better)		
	Mean	Median	RMSE	11.25°	22.5°	30°
Our Network	**25.0**	**13.8**	**35.9**	**44.2**	**63.2**	**70.3**
UNFOLD [7]	35.1	19.2	48.7	37.6	53.3	58.9
Discr. [20]	32.5	22.4	43.3	27.4	50.2	60.2
3DP (MW) [6]	36.0	20.5	49.4	35.9	52.0	57.8
3DP [6]	34.2	30.0	41.4	18.6	38.6	49.9

K. Karsch, V. Hedau, D. Forsyth, D. Hoiem, Rendering synthetic objects into legacy photographs, SIGGRAPH'11

link to video

How Many Times Have You Looked for Apartments?

United States:

- 11.7% per year

Craigslist:

- 90,000 rental ads per day only in New York
- 10 million people visit the website per day

[From Rent3D slides]

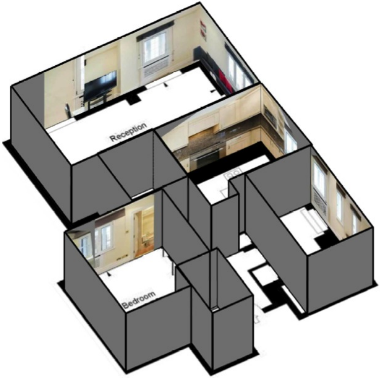
Example Rental Data

- Plus some meta information e.g. wall height

[From Rent3D slides]

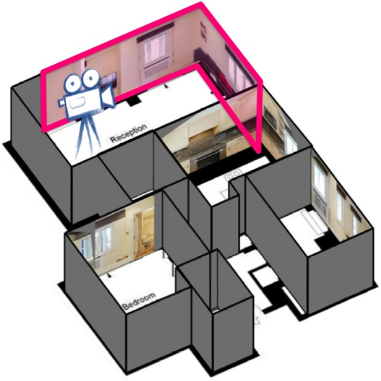
Rent3D: View Rental Ads in 3D

[Liu et al., 2015]

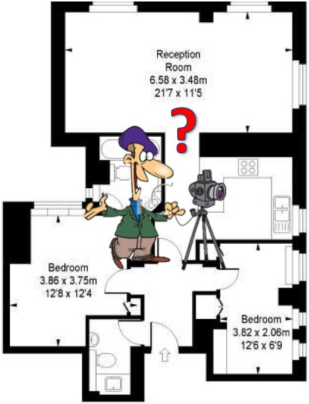


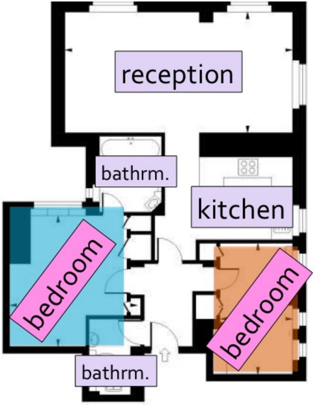
C. Liu, A. Schwing, K. Kundu, R. Urtasun, S. Fidler, Rent3D: Floor-Plan Priors for Monocular Layout Estimation, *CVPR'15* 2015

Data: http://www.cs.utoronto.ca/~fidler/projects/rent3D.html



- Camera localization within apartment

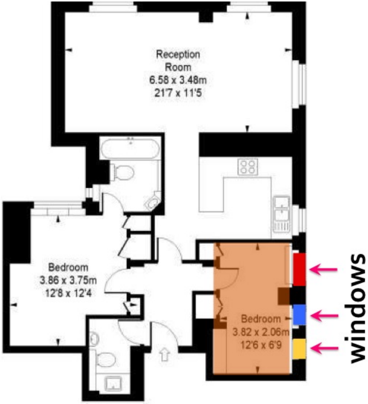

Reception
Room
6.58 x 3.48m
21'7 x 11'5
Bedroom
3.86 x 3.75m
12'8 x 12'4
Bedroom
3.82 x 2.06m
12'6 x 6'9



bedroom

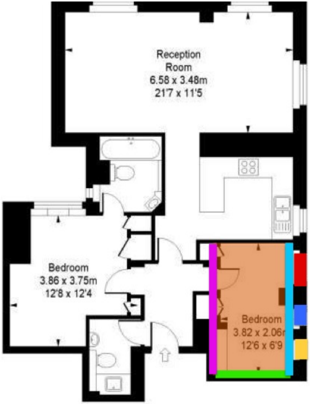
Accurate **camera localization**:

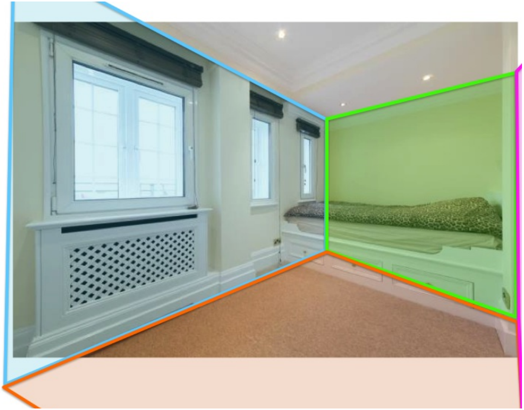
- **Scene cues**



Accurate **camera localization**:

- **Scene cues**
- **Semantic cues**

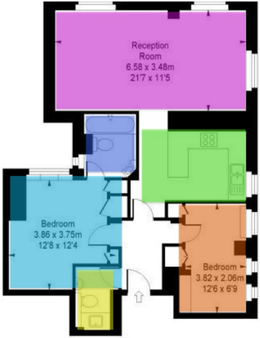




Accurate **camera localization**:

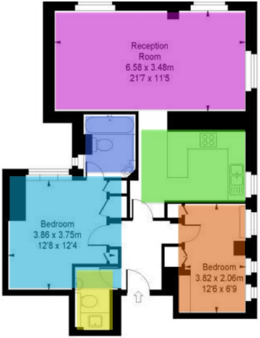
- **Scene cues**
- **Semantic cues**
- **Geometric cues** by exploiting the dimension information

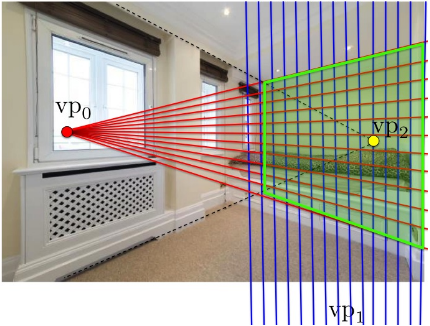
- $r \in \{1, \ldots, R\}$ … discrete random variable representing the room



- $r \in \{1, \dots, R\}$ ... discrete random variable representing the room

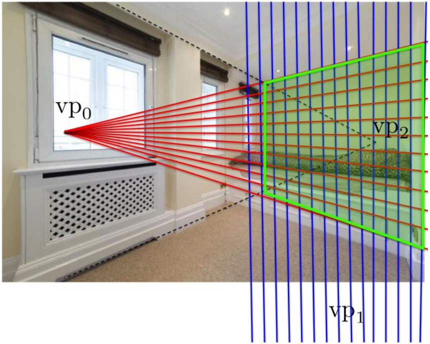
Front wall is the plane defined by vp_0 and vp_1



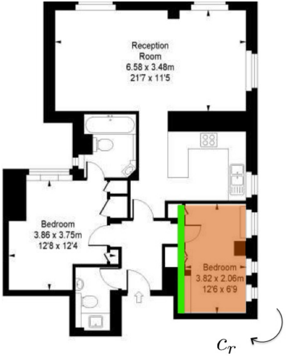


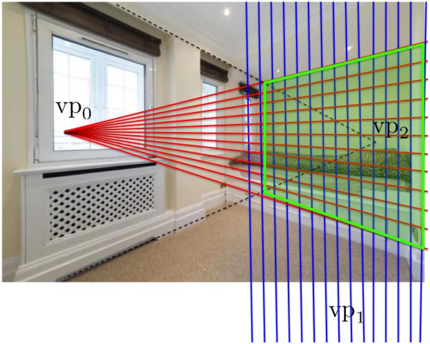
Formulation

- $r \in \{1, \ldots, R\}$ ... discrete random variable representing the room
- $c_r \in \{1, \ldots, |C_r|\}$ ... a discrete variable representing within room r which wall the picture is facing ($|C_r|$ the number of walls in a room)

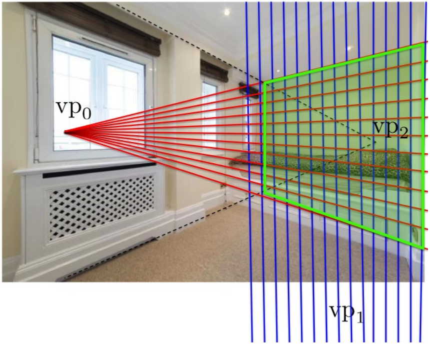


- $r \in \{1, \ldots, R\}$ ... discrete random variable representing the room
- $c_r \in \{1, \ldots, |C_r|\}$ ... a discrete variable representing within room r which wall the picture is facing ($|C_r|$ the number of walls in a room)

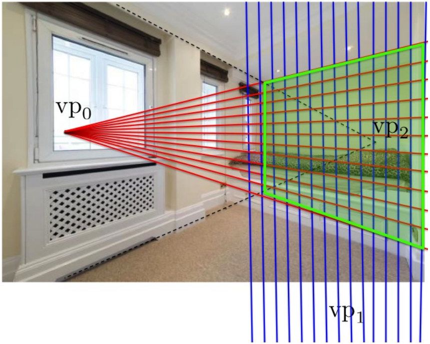




- $r \in \{1, \ldots, R\}$ ... discrete random variable representing the room
- $c_r \in \{1, \ldots, |C_r|\}$ ... a discrete variable representing within room r which wall the picture is facing ($|C_r|$ the number of walls in a room)

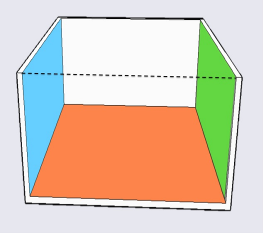


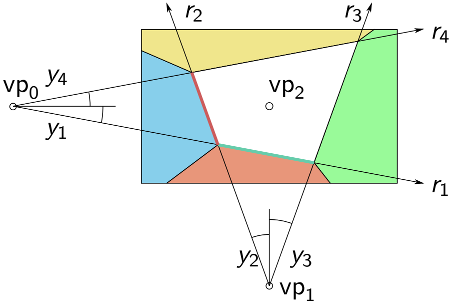
- $r \in \{1, \ldots, R\}$ ... discrete random variable representing the room
- $c_r \in \{1, \ldots, |C_r|\}$ ... a discrete variable representing within room r which wall the picture is facing ($|C_r|$ the number of walls in a room)



- $r \in \{1, \ldots, R\}$ ... discrete random variable representing the room
- $c_r \in \{1, \ldots, |C_r|\}$ ... a discrete variable representing within room r which wall the picture is facing ($|C_r|$ the number of walls in a room)
- $\mathbf{y}$ ... rays representing a room layout

Typical parametrization for room layout (Hedau et al.):





- Room is a 3D cuboid
- $\mathbf{y} = (y_1, y_2, y_3, y_4)$
- 4 rays needed to define it

Formulation

[Liu et al., 2015]

- $r \in \{1, \ldots, R\}$ ... discrete random variable representing the room
- $c_r \in \{1, \ldots, |C_r|\}$ ... a discrete variable representing within room r which wall the picture is facing ($|C_r|$ the number of walls in a room)
- $\mathbf{y}$ ... rays representing a room layout
- The problem formulated as inference in a Conditional Random Field with the following energy:

$$E(r, c_r, \mathbf{y}) = E_{scene\_type}(r) + E_{layout}(r, c_r, \mathbf{y}) + E_{win}(r, c_r, \mathbf{y})$$

Energy Terms: Scene Type [Liu et al., 2015]

$$E(r, c_r, \mathbf{y}) = E_{scene\_type}(r) + E_{layout}(r, c_r, \mathbf{y}) + E_{win}(r, c_r, \mathbf{y})$$

- **Potential:** Score of a scene classifier predicting scene type (e.g., bedroom, kitchen, reception)

Energy Terms: Scene Type [Liu et al., 2015]

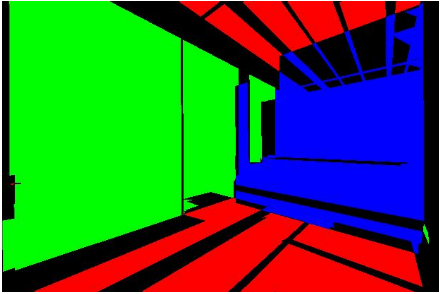
$$E(r, c_r, \mathbf{y}) = E_{scene\_type}(r) + E_{layout}(r, c_r, \mathbf{y}) + E_{win}(r, c_r, \mathbf{y})$$

- **Potential:** Score of a scene classifier predicting scene type (e.g., bedroom, kitchen, reception)

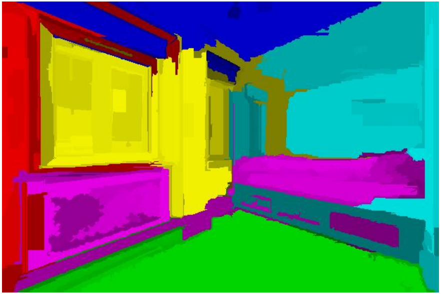
Energy Terms: Layout

[Liu et al., 2015]

$$E(r, c_r, \mathbf{y}) = E_{scene\_type}(r) + E_{layout}(r, c_r, \mathbf{y}) + E_{win}(r, c_r, \mathbf{y})$$

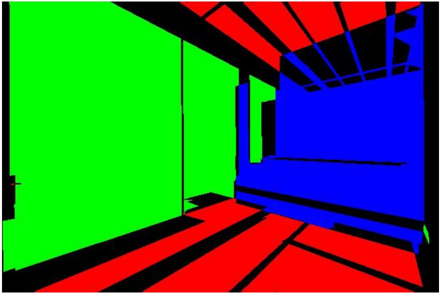


Orientation Map (Lee et al.)

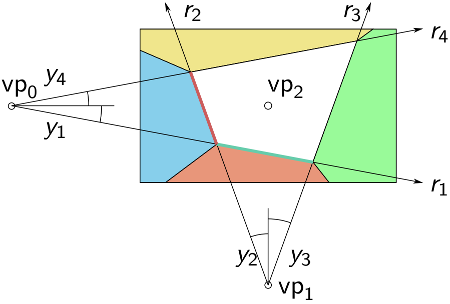


Geometric Context (Hedau et al.)

$$E(r, c_r, \mathbf{y}) = E_{scene\_type}(r) + E_{layout}(r, c_r, \mathbf{y}) + E_{win}(r, c_r, \mathbf{y})$$



Orientation Map (Lee et al.)

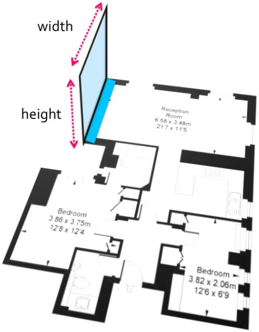


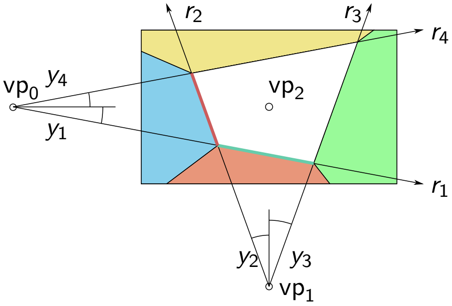
- **Potentials**: Counts of blue, red, etc, pixels inside and outside of each wall
- Fast computation using *integral geometry* [Schwing et al., 2012]

Energy Terms: Layout

[Liu et al., 2015]

$$E(r, c_r, \mathbf{y}) = E_{scene\_type}(r) + E_{layout}(\boxed{r, c_r}, \mathbf{y}) + E_{win}(r, c_r, \mathbf{y})$$

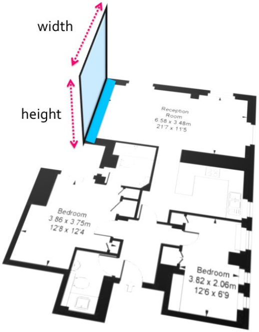


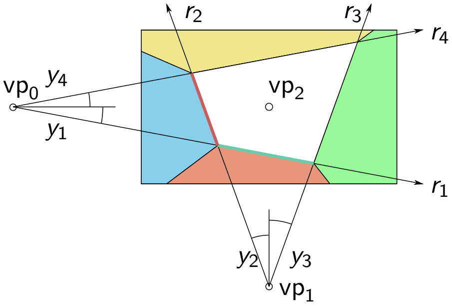


Energy Terms: Layout

[Liu et al., 2015]

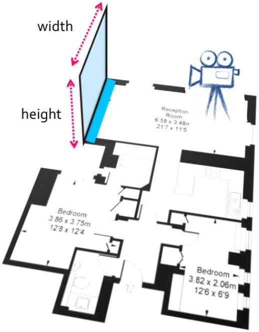
$$E(r, c_r, \mathbf{y}) = E_{scene\_type}(r) + E_{layout}(\boxed{r, c_r}, \mathbf{y}) + E_{win}(r, c_r, \mathbf{y})$$

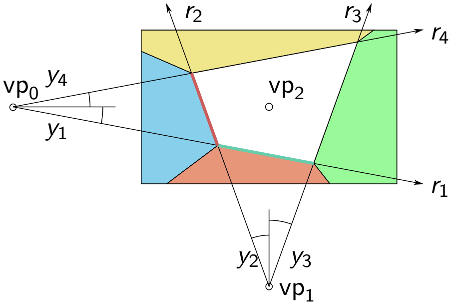




- $\mathbf{y} = (y_1, y_2, y_3, \cancel{y_4})$, $\quad y_4 = f(r, c_r, y_1, y_2, y_3)$

$$E(r, c_r, \mathbf{y}) = E_{scene\_type}(r) + E_{layout}(\boxed{r, c_r}, \mathbf{y}) + E_{win}(r, c_r, \mathbf{y})$$





- $\mathbf{y} = (y_1, y_2, y_3, \cancel{y_4})$, $\quad y_4 = f(r, c_r, y_1, y_2, y_3)$
- Additional constraint on $\mathbf{y}$: Camera is **inside** the room

$$E(r, c_r, \mathbf{y}) = E_{scene\_type}(r) + E_{layout}(r, c_r, \mathbf{y}) + E_{win}(r, c_r, \mathbf{y})$$

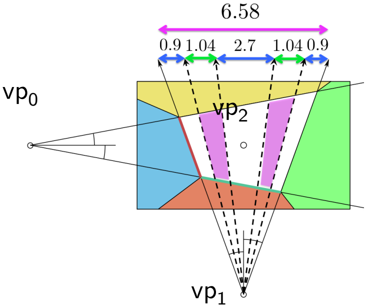
- Window-background segmentation

Energy Terms: Windows

[Liu et al., 2015]

$$E(r, c_r, \mathbf{y}) = E_{scene\_type}(r) + E_{layout}(r, c_r, \mathbf{y}) + E_{win}(\boxed{r, c_r}, \mathbf{y})$$

- Window-background segmentation
- **Potential**: count window pixels inside and outside the window area



- We are minimizing the energy:

$$(r^*, c_r^*, \mathbf{y}^*) = \underset{r, c_r, \mathbf{y}}{\operatorname{argmin}} \ \left(E_{scene\_type}(r) + E_{layout}(r, c_r, \mathbf{y}) + E_{win}(r, c_r, \mathbf{y})\right)$$

- We are minimizing the energy:

$$(r^*, c_r^*, \mathbf{y}^*) = \operatorname*{argmin}_{r, c_r, \mathbf{y}} \left(E_{scene\_type}(r) + E_{layout}(r, c_r, \mathbf{y}) + E_{win}(r, c_r, \mathbf{y}) \right)$$

- Inference:
  - Exhaustive enumeration of r and c_r
  - Exact branch and bound inference for $\mathbf{y}$ [Schwing & Urtasun, 2012]

- We are minimizing the energy:

$$(r^*, c_r^*, \mathbf{y}^*) = \underset{r, c_r, \mathbf{y}}{\operatorname{argmin}} \left(E_{scene\_type}(r) + E_{layout}(r, c_r, \mathbf{y}) + E_{win}(r, c_r, \mathbf{y}) \right)$$

- Inference:
  - Exhaustive enumeration of r and c_r
  - Exact branch and bound inference for $\mathbf{y}$ [Schwing & Urtasun, 2012]
- S-SVM for training

Dataset

[Liu et al., 2015]

- Crawled a London apartment rental site

# apartments	215
# of images	1570
# of indoor images	1259
# images without GT alignment	82
avg. # rooms per apt	6
avg. # walls per apt	31
avg. # windows per apt	6
avg. # doors per apt	9

Results: Layout Estimation

[Liu et al., 2015]

- We assume we know which wall the camera is facing
- **Metrics**: Pixel accuracy for predicting 5 walls

	Layout error	Evaluations	Test time [s]
Schwing'12	13.88	16012.4	0.0208
Rent3D	**11.69**	**1271.5**	**0.0037**

Results: Layout Estimation

[Liu et al., 2015]

- We assume we know which wall the camera is facing
- **Metrics**: Pixel accuracy for predicting 5 walls

	Layout error	Evaluations	Test time [s]
Schwing'12	13.88	16012.4	0.0208
Rent3D	**11.69**	**1271.5**	**0.0037**

- 2% reduction in layout error

Results: Layout Estimation

[Liu et al., 2015]

- We assume we know which wall the camera is facing
- **Metrics**: Pixel accuracy for predicting 5 walls

	Layout error	Evaluations	Test time [s]
Schwing'12	13.88	16012.4	0.0208
Rent3D	**11.69**	**1271.5**	**0.0037**

- 2% reduction in layout error
- 10 times less branching operations

Results: Layout Estimation

- We assume we know which wall the camera is facing
- **Metrics**: Pixel accuracy for predicting 5 walls

	Layout error	Evaluations	Test time [s]
Schwing'12	13.88	16012.4	0.0208
Rent3D	**11.69**	**1271.5**	**0.0037**

- 2% reduction in layout error
- 10 times less branching operations
- 10x speedup

- **Metrics**: % of correct assignments of front wall to the apartment wall

	Aspect	+Scene	+Room
Random	0.0328	0.1138	0.1954
Rent3D (no windows)	0.0686	0.1945	0.2654
Rent3D (windowGT)	0.2128	0.4737	0.5995
Rent3D (window)	0.1670	0.3982	0.5080

Results: Camera Localization

- **Metrics**: % of correct assignments of front wall to the apartment wall

	Aspect	+Scene	+Room
Random	0.0328	0.1138	0.1954
Rent3D (no windows)	0.0686	0.1945	0.2654
Rent3D (windowGT)	0.2128	0.4737	0.5995
Rent3D (window)	0.1670	0.3982	0.5080

Aspect: Only aspect ratio information (and not scene) used

Results: Camera Localization

- **Metrics**: % of correct assignments of front wall to the apartment wall

	Aspect	+Scene	+Room
Random	0.0328	0.1138	0.1954
Rent3D (no windows)	0.0686	0.1945	0.2654
Rent3D (windowGT)	0.2128	0.4737	0.5995
Rent3D (window)	0.1670	0.3982	0.5080

+*Scene*: Aspect information and scene classifier are used

Results: Camera Localization [Liu et al., 2015]

- **Metrics**: % of correct assignments of front wall to the apartment wall

	Aspect	+Scene	+Room
Random	0.0328	0.1138	0.1954
Rent3D (no windows)	0.0686	0.1945	0.2654
Rent3D (windowGT)	0.2128	0.4737	0.5995
Rent3D (window)	0.1670	0.3982	0.5080

+*Room*: We know which room the picture was taken in

Results: Camera Localization

- **Metrics**: % of correct assignments of front wall to the apartment wall

	Aspect	+Scene	+Room
Random	0.0328	0.1138	0.1954
Rent3D (no windows)	0.0686	0.1945	0.2654
Rent3D (windowGT)	0.2128	0.4737	0.5995
Rent3D (window)	0.1670	0.3982	0.5080

Results: Camera Localization

- **Metrics**: % of correct assignments of front wall to the apartment wall

	Aspect	+Scene	+Room
Random	0.0328	0.1138	0.1954
Rent3D (no windows)	0.0686	0.1945	0.2654
Rent3D (windowGT)	0.2128	0.4737	0.5995
Rent3D (window)	0.1670	0.3982	0.5080

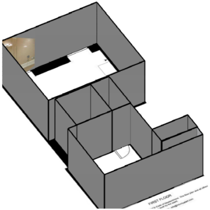
Results: Joint Layout and Localization [Liu et al., 2015]

Red arrow: Groundtruth camera Green arrow: Predicted camera

Results

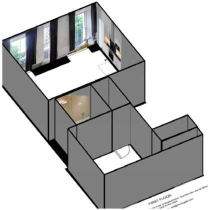
[Liu et al., 2015]

Window+Aspect	+Scene	+Room	Ground-truth

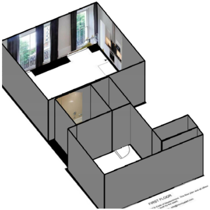


1 images out of 4
2 walls out of 8

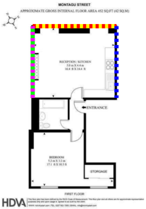
4 images out of 4
8 walls out of 8

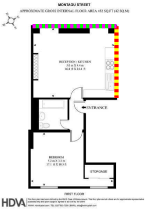


4 images out of 4
8 walls out of 8



-
-





J. Xiao and Y. Furukawa, Reconstructing the Worlds Museums, *IJCV*, 2014

- Virtual tour of large indoor spaces (e.g., museums)
- Uses a rig of cameras and three linear laser range sensors

Reconstructing Museums

[Xiao and Furukawa, 2014]

- Virtual tour of large indoor spaces (e.g., museums)
- Uses a rig of cameras and three linear laser range sensors

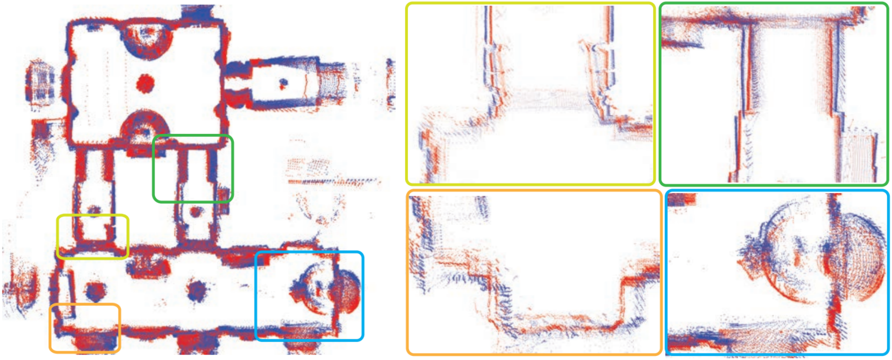
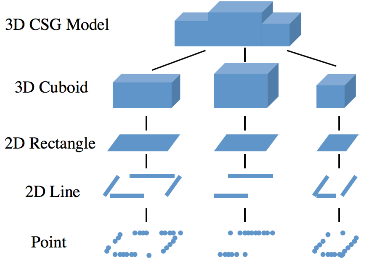


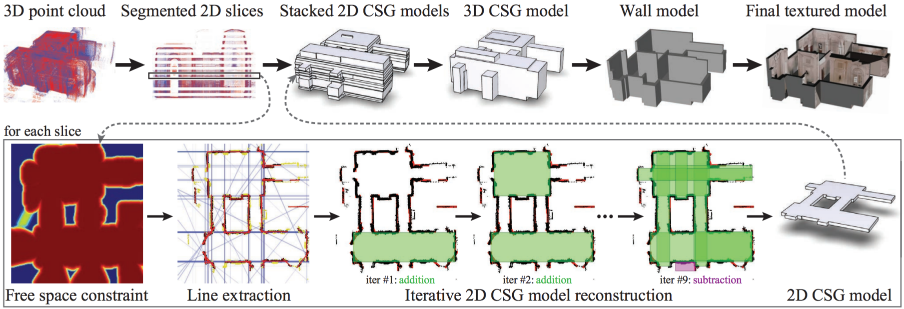
Figure: Red and blue points obtained with two different laser scanners

Reconstructing Museums

[Xiao and Furukawa, 2014]

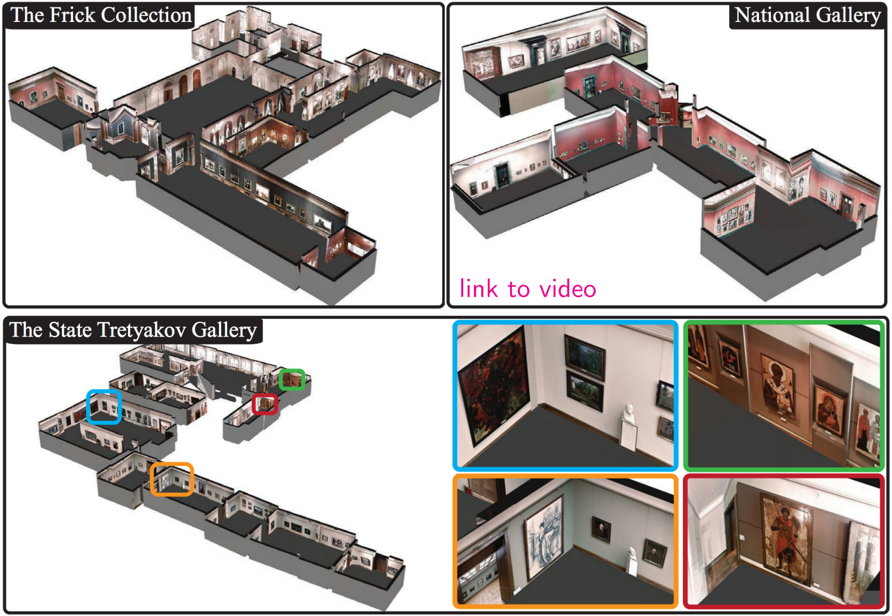
Construction of a Constructive Solid Geometry (CSG) model consisting of volumetric primitives





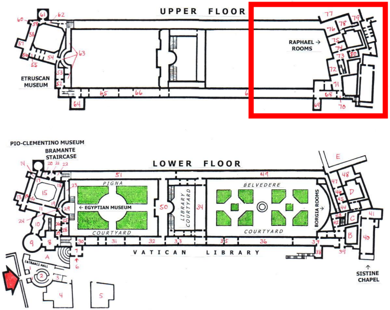
Reconstructing Museums

[Xiao and Furukawa, 2014]



R. Martin-Brualla, Y. He, B. C. Russell, S. M. Seitz, The 3D jigsaw puzzle: mapping large indoor spaces, *ECCV*, 2014
Project page: http://grail.cs.washington.edu/projects/jigsaw3d/

- SfM using Internet photos of popular tourist sites
- Place 3D models in a global reference frame (a floormap)



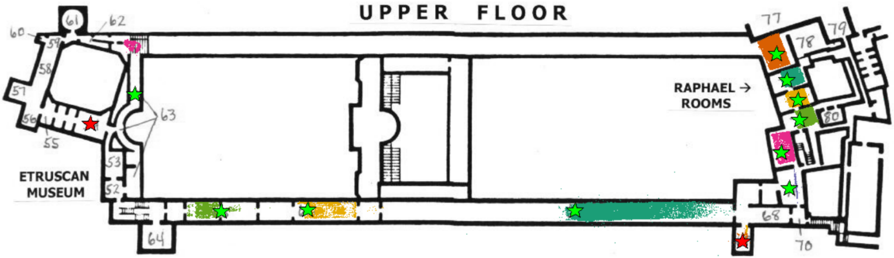


Figure: Localization results

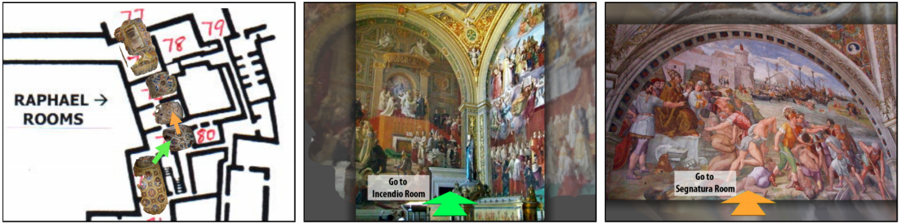
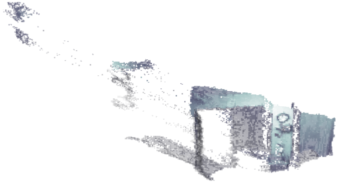


Figure: Interactive visualization (link to video)

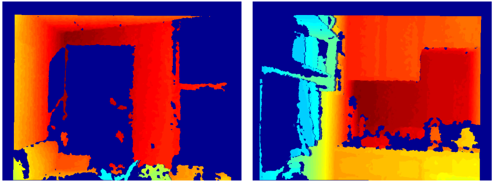
Indoor Scene Understanding with RGB-D Data

Difficult problem?

Noisy depth

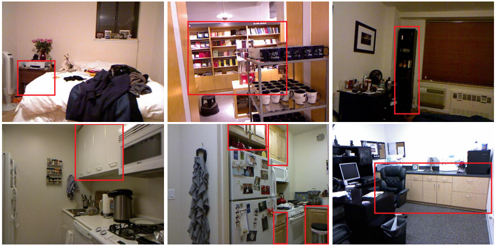


Missing depth



Occlusion

Viewpoint, aspect-ratio variation



R-CNN with Depth

[Gupta et al., 2014]

S. Gupta, R. Girshick, P. Arbelaez, J. Malik, Learning Rich Features from RGB-D Images for Object Detection and Segmentation, *ECCV'14*
Code, data: https://github.com/s-gupta/rcnn-depth

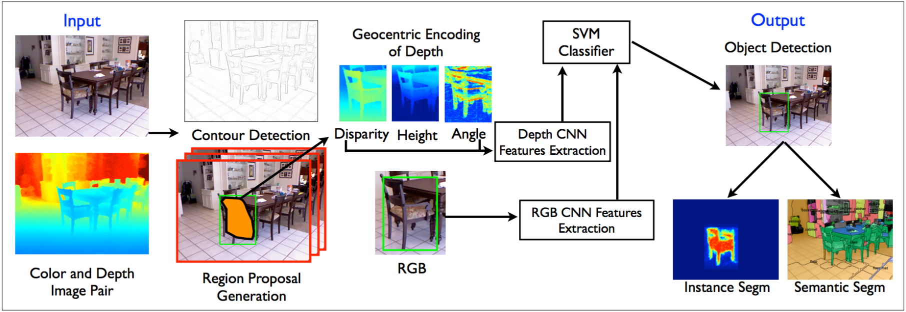
- Observation: The standard R-CNN pipeline doesn't work well for detection on NYU-v2
- Can we train a better network that includes depth?

	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

R-CNN with Depth

[Gupta et al., 2014]

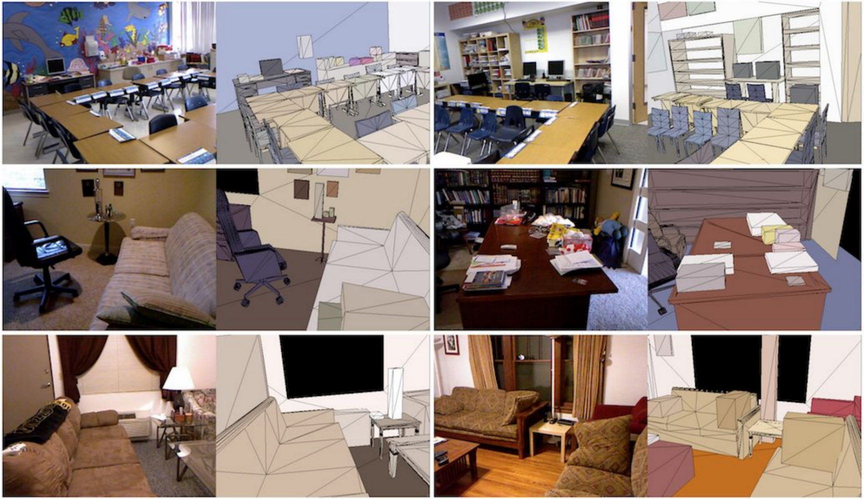
- Trick: Use network pre-trained on e.g. ImageNet and fine-tune it on a 3D depth encoding "HHA"
- HHA: horizontal disparity, height above ground, and the angle between pixel's normal and the inferred gravity direction



R-CNN with Depth

[Gupta et al., 2014]

- Fine-tune network on synthetic views generated with Guo & Hoiem's models



R-CNN with Depth

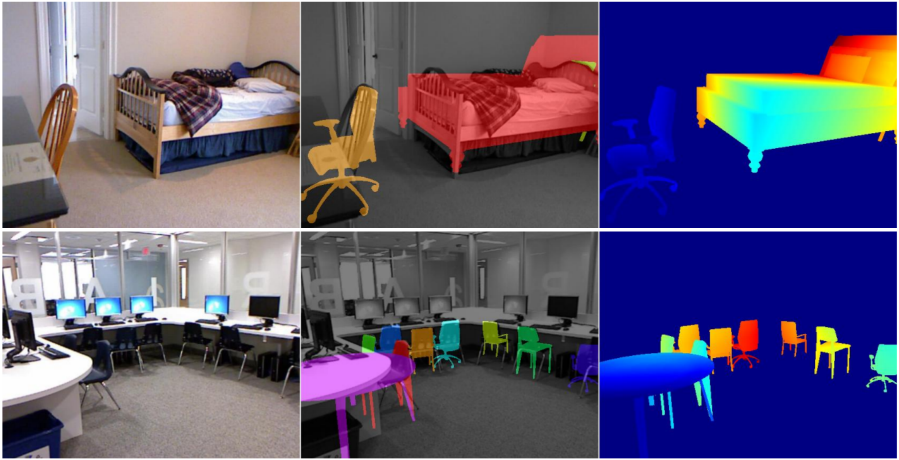
[Gupta et al., 2014]

	A	B	C	D	E	F	G	H	I	J	K	L
	DPM	DPM	CNN	CNN	CNN	CNN	CNN	CNN	CNN	CNN	CNN	CNN
finetuned?			no	yes	no	yes	yes	yes	yes	yes	yes	yes
input channels	RGB	RGBD	RGB	RGB	disparity	disparity	HHA	HHA	HHA	HHA	HHA	RGB+HHA
synthetic data?								2x	15x	2x	2x	2x
CNN layer			fc6	fc6	fc6	fc6	fc6	fc6	fc6	pool5	fc7	fc6
bathtub	0.1	12.2	4.9	5.5	3.5	6.1	20.4	20.7	20.7	11.1	19.9	**22.9**
bed	21.2	56.6	44.4	52.6	46.5	63.2	60.6	67.2	**67.8**	61.0	62.2	66.5
bookshelf	3.4	6.3	13.8	19.5	14.2	16.3	20.7	18.6	16.5	20.6	18.1	**21.8**
box	0.1	0.5	1.3	1.0	0.4	0.4	0.9	1.4	1.0	1.0	1.1	**3.0**
chair	6.6	22.5	21.4	24.6	23.8	36.1	38.7	38.2	35.2	32.6	37.4	**40.8**
counter	2.7	14.9	20.7	20.3	18.5	32.8	32.4	33.6	36.3	24.1	35.0	**37.6**
desk	0.7	2.3	2.8	6.7	1.8	3.1	5.0	5.1	7.8	4.2	5.4	**10.2**
door	1.0	4.7	10.6	14.1	0.9	2.3	3.8	3.7	3.4	2.8	3.3	**20.5**
dresser	1.9	23.2	11.2	16.2	3.7	5.7	18.4	18.9	**26.3**	13.1	24.7	26.2
garbage-bin	8.0	26.6	17.4	17.8	2.4	12.7	26.9	29.1	16.4	21.4	25.3	**37.6**
lamp	16.7	25.9	13.1	12.0	10.5	21.3	24.5	26.5	23.6	22.3	23.2	**29.3**
monitor	27.4	27.6	24.8	32.6	0.4	5.0	11.5	14.0	12.3	17.7	13.5	**43.4**
night-stand	7.9	16.5	9.0	18.1	3.9	19.1	25.2	27.3	22.1	25.9	27.8	**39.5**
pillow	2.6	21.1	6.6	10.7	3.8	23.4	35.0	32.2	30.7	31.1	31.2	**37.4**
sink	7.9	**36.1**	19.1	6.8	20.0	28.5	30.2	22.7	24.9	18.9	23.0	24.2
sofa	4.3	28.4	15.5	21.6	7.6	17.3	36.3	37.5	39.0	30.2	34.3	**42.8**
table	5.3	14.2	6.9	10.0	12.0	18.0	18.8	22.0	22.6	21.0	22.8	**24.3**
television	16.2	23.5	29.1	31.6	9.7	14.7	18.4	23.4	26.3	18.9	22.9	**37.2**
toilet	25.1	48.3	39.6	52.0	31.2	**55.7**	51.4	54.2	52.6	38.4	48.8	53.0
mean	8.4	21.7	16.4	19.7	11.3	20.1	25.2	26.1	25.6	21.9	25.3	**32.5**

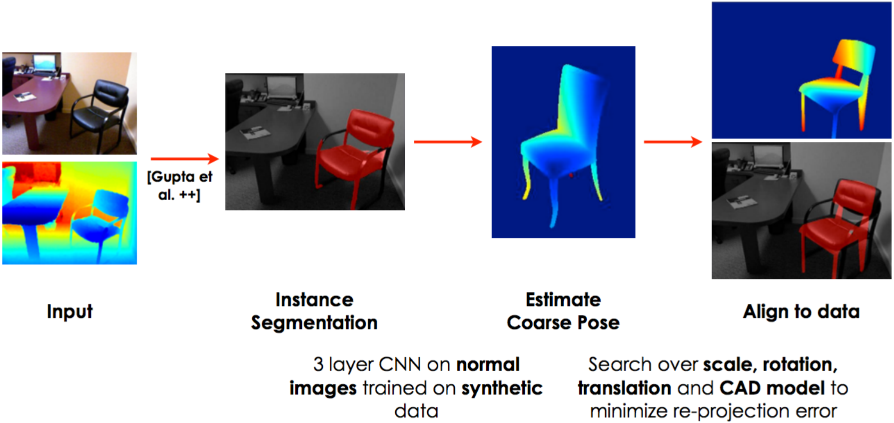
Aligning CAD Models in RGB-D [Gupta et al., 2015]

S. Gupta, P. Arbelaez, R. Girshick, J. Malik , Aligning 3D Models to RGB-D Images of Cluttered Scenes, *CVPR'15*

- Goal: Align CAD models in RGB-D scenes



- Generate object candidates using previous approach
- A deep net that predicts coarse pose (trained with *model net*)
- A modified ICP to match a small number of category CAD models



Aligning CAD Models in RGB-D

[Gupta et al., 2015]

task		fine tuning set	mean	bath tub	bed	book shelf	box	chair	counter	desk	door	dresser	garbage bin	lamp	monitor	night stand	pillow	sink	sofa	table	tele vision	toilet
AP^b	[13]	train	35.9	39.5	69.4	32.8	1.3	41.9	44.3	13.3	21.2	31.4	35.8	35.8	50.1	31.4	39.0	42.4	50.1	23.5	33.3	46.4
	[13] + Region Features	train	39.3	**50.0**	70.6	34.9	3.0	45.2	**48.7**	**15.2**	23.5	32.6	48.3	34.9	50.2	32.2	**44.2**	43.1	**54.9**	23.4	41.5	49.9
	[13]	trainval	38.8	36.4	70.8	35.1	3.6	47.3	46.8	14.9	23.3	38.6	43.9	**37.6**	**52.7**	40.7	42.4	**43.5**	51.6	22.0	38.0	47.7
	[13] + Region Features	trainval	**41.2**	39.4	**73.6**	**38.4**	**5.9**	**50.1**	47.3	14.6	**24.4**	**42.9**	**51.5**	36.2	52.1	**41.5**	42.9	42.6	54.6	**25.4**	**48.6**	**50.2**
AP^r	[13] (Random Forests)	train	32.1	18.9	**66.1**	10.2	1.5	35.5	32.8	**10.2**	**22.8**	33.7	38.3	**35.5**	53.3	**42.7**	31.5	34.4	40.7	14.3	37.4	50.3
	[13] + Region Features	train	34.0	33.8	64.4	9.8	2.3	36.6	41.3	9.7	20.4	30.9	47.4	26.6	51.6	27.5	**42.1**	37.1	44.8	14.7	42.7	62.6
	[13] + Region Features	trainval	**37.5**	**42.0**	65.1	**12.7**	**5.1**	**42.0**	**42.1**	9.5	20.5	**38.0**	**50.3**	32.8	**54.5**	38.2	42.0	**39.4**	**46.6**	**14.8**	**48.0**	**68.4**

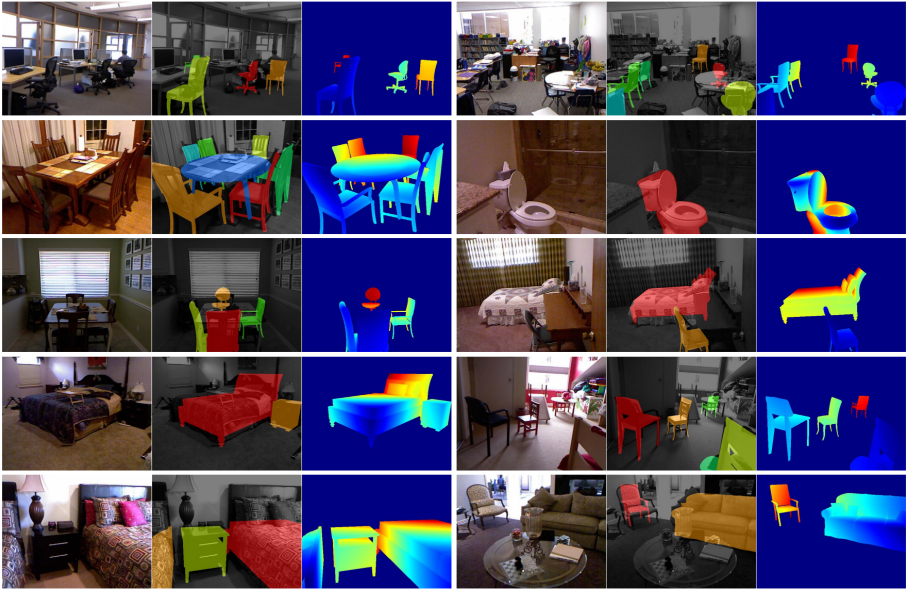
Figure: Detection and instance segmentation

	3D all						3D clean					
	mean	bed	chair	sofa	table	toilet	**mean**	bed	chair	sofa	table	toilet
Our (3D Box on instance segm. from [13])	48.4	74.7	18.6	50.3	28.6	69.7	66.1	90.9	45.9	68.2	25.5	100.0
Our (3D Box around estimated model)	58.5	73.4	44.2	57.2	33.4	84.5	71.1	82.9	72.5	75.3	24.6	100.0
Song and Xiao [34]	39.6	33.5	29.0	34.5	33.8	67.3	64.6	71.2	**78.7**	41.0	**42.8**	89.1
Our [no RGB[1]] (3D Box on instance segm. from [13])	46.5	71.0	18.2	49.6	30.4	63.4	62.3	**86.9**	43.6	57.4	26.6	**96.7**
Our [no RGB[1]] (3D Box around estimated model)	**57.6**	**72.7**	**47.5**	**54.6**	**40.6**	**72.7**	**70.7**	84.9	75.7	**62.8**	33.7	**96.7**

Figure: 3D detection

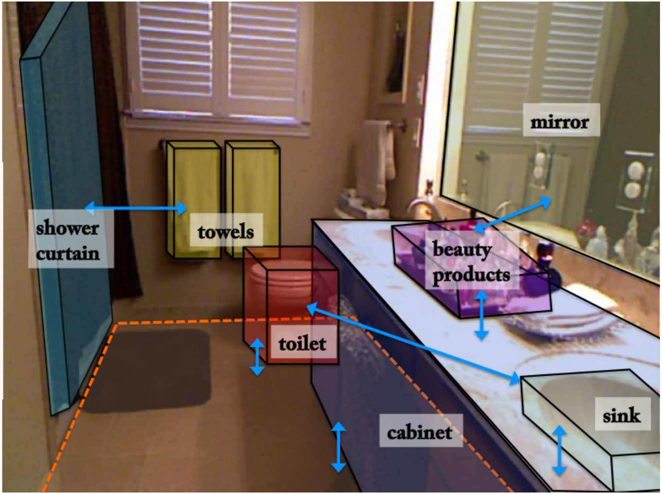
Aligning CAD Models in RGB-D

[Gupta et al., 2015]



- Reasoning jointly about multiple related tasks may help

bathroom

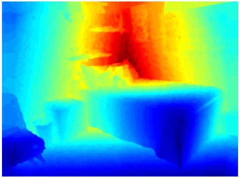


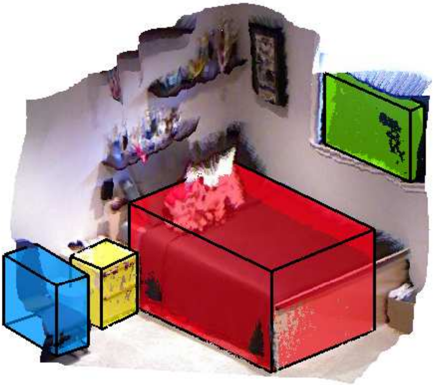
D. Lin, S. Fidler, R. Urtasun, Holistic Scene Understanding for 3D Object Detection with RGBD cameras, *ICCV'13*
Code, data: http://www.cs.utoronto.ca/~fidler/projects/scenes3D.html

- Exploit **RGBD imagery** for **category-level 3D object detection**
- **Holistic approach**: jointly reason about **scene**, **objects**, and **context**

image

depth





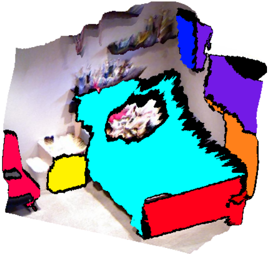
point cloud with **cuboids around objects**

Cuboid Candidates

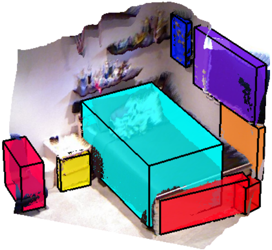
[Lin et al., 2013]

- Get candidate "objectness" regions with CPMC [Carreira et al., PAMI 2012] extended to 3D
- Take top K candidates ranked by objectness score
- Project each region to 3D
- Fit a minimal cube that contains 95% of the 3D points
- Enforce the gravity vector of each cube to be orthogonal to the floor

example regions



regions in 3D



fit cuboids

$$p(\mathbf{y}, s) \propto \exp\left(\mathbf{w}_\mathbf{s}^\mathbf{T}\phi_s(s) + \mathbf{w}_\mathbf{y}^\mathbf{T}\sum_{i=1}^{K}\phi_y(y_i) + \mathbf{w}_\mathbf{yy}^\mathbf{T}\sum_{(i,j)}\phi_{yy}(y_i, y_j) + \mathbf{w}_\mathbf{sy}^\mathbf{T}\sum_{i=1}^{K}\phi_{sy}(s, y_i)\right)$$

cuboid class:
$y_i \in \{0, \ldots, C\}$

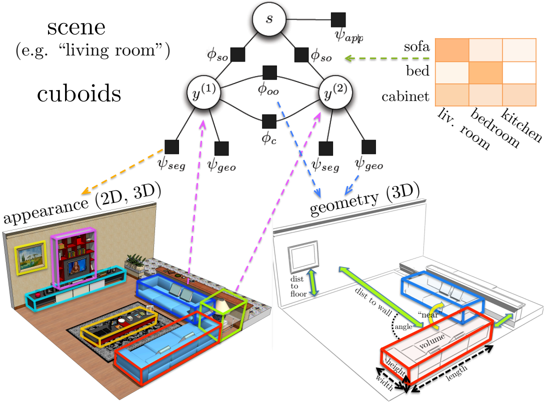
scene class:
$s \in \{1, \ldots, S\}$

Unary:

- appearance
- geometry

Pairwise:

- spatial relations
- semantic relations

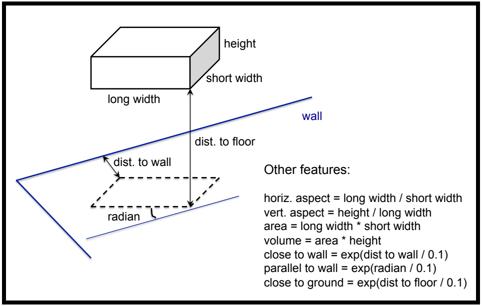


- **Scene appearance:** Classifier on RGB-D features
- **Ranking potential:** Predicts amount of overlap of object candidate with ground-truth [CPMC-o2p, Carreira et al., 2012]
- **Segmentation potential:** Classifier on superpixels using RGB-D kernel descriptors
- **Object geometry:** Classifier on geometric features

RGB-D features:

- RGB: gradient, color, LBP, self-similarity, SIFT
- Depth: depth gradient, spin/surface normal

Geometry features:



Semantic context:

- **scene-object potential**:

$$\phi_{sy}(s = k, y = l) = \text{scene-object co-occurrence stats}$$

- **object-object potential**

$$\phi_{yy}(y = l, y' = l') = \text{object-object co-occurrence stats}$$

Geometric relations:

- **close-to**: Two objects are *close to* each other if their distance is less than 0.5 meters.
- **on-top-of**: Object A is *on top of* B if A is higher than B and (at least) 80% of A's bottom face is contained within the top face of B.

Learning and Inference

[Lin et al., 2013]

- **Loss:** how far from GT is each hypothesis
  - Object: 0/1 loss based on IOU with GT
  - Scene: 0/1 loss
- **Learning:** Primal dual method blending learning and inference [Hazan and Urtasun, NIPS 2010]
- **Inference:** Distributed message passing [Schwing et al., CVPR 2011]
- **Timings**:
  - **learning** takes **2 minutes** ($\sim$ 800 images)
  - **inference** takes **15 ms per image** (15 cuboids per image)

On Intel i7 quad-core CPU (4 threads)

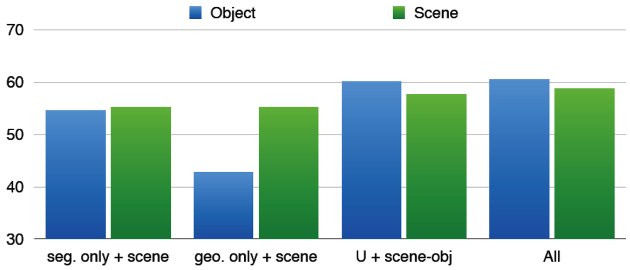
Results

- NYUv2 [Silberman et al, 2012]: 1449 scenes, 6680 objects, 21 object classes + background
- Ground truth: Fit 3D cuboids around GT regions and correct bad fits
- Standard split: 60% of images used for training and 40% for test

Results on GT Cuboids [Lin et al., 2013]

- Performance of scene measured in classification accuracy
- Performance evaluated on GT cuboids, measured as classification accuracy

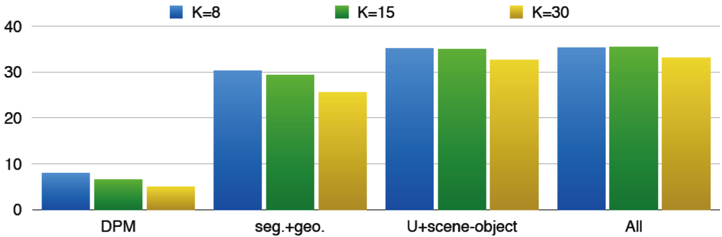
configuration	object	scene
scene appearance only	-	55.20
segmentation only	54.46	-
geometry only	42.85	-
all unaries	59.02	55.20
unaries + scene-obj	60.00	57.65
full model	**60.49**	**58.72**



Full Detection Pipeline

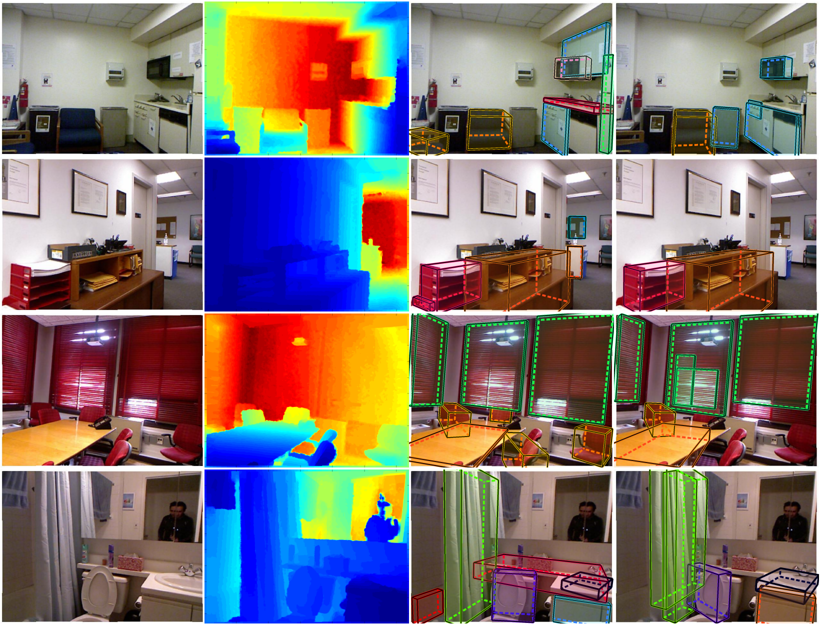
- Performance measured as average of per-class F-measures
- DPM: [Felzenswalb et al., TPAMI, 2010]
- Jiang'13: Cuboids from [H. Jiang and J. Xiao, CVPR, 2013]

	DPM	seg.	seg.+geo.	all unaries	+scene-object	full model
[Jiang'13]	-	11.11	21.13	21.90	22.19	**22.3**
K = 8	8.01	28.98	30.22	35.17	35.18	**35.23**
K = 15	6.54	28.33	29.44	34.92	34.95	**35.56**
K = 30	4.96	24.81	25.58	32.54	32.57	**33.10**



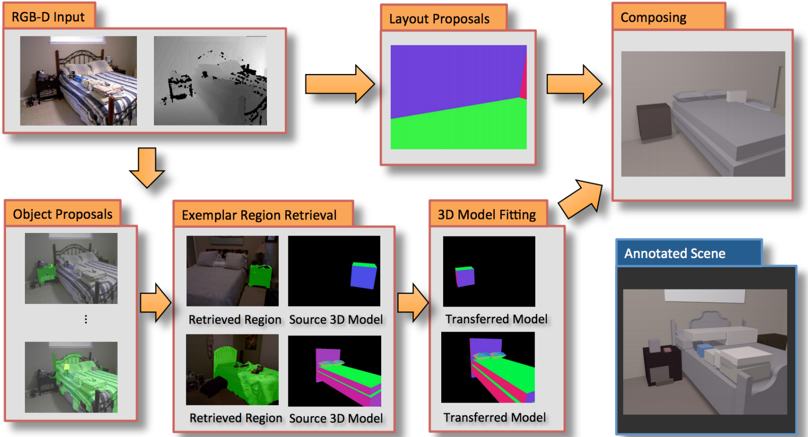
Example Detections

[Lin et al., 2013]



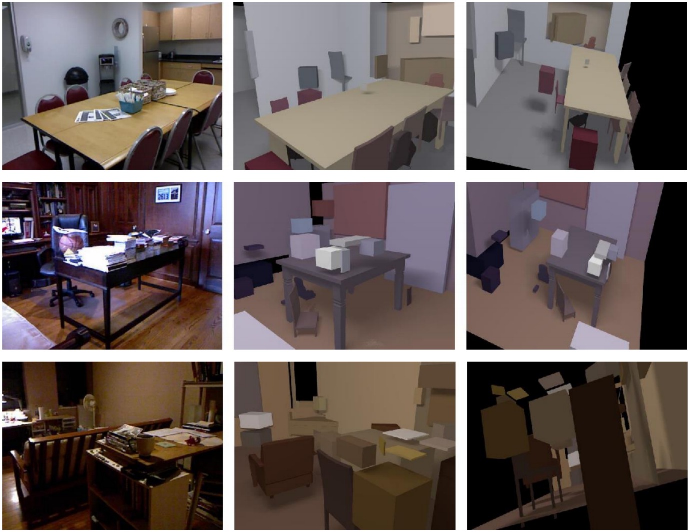
R. Guo, C. Zou, D. Hoiem, Predicting Complete 3D Models of Indoor Scenes , *arXiv:1504.02437*, 2015

- Generates layout and object candidates, and re-reasons about the best configuration in a holistic way



Predicting Complete 3D Models

[Guo et al., 2015]



Input Image

Automatic 3D Model (two views)

link to video

Indoor RGB-D Datasets

- NYUv2 dataset:

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

- RMRC challenge:

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

- B3DO: Berkeley 3-D Object Dataset:

  http://kinectdata.com/

- SUN RGB-D:

  http://rgbd.cs.princeton.edu/

Discussion

- What is missing?
- What are the next steps?