Part II: Monocular Room Layout Estimation

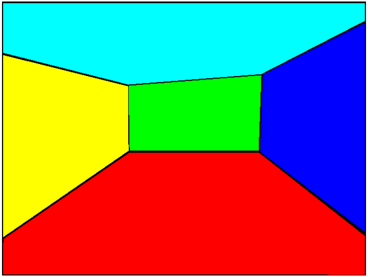
Sanja Fidler and Raquel Urtasun

University of Toronto

June 7, 2015

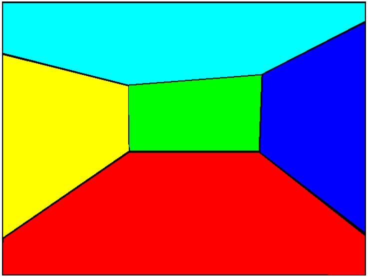
Room Layout Estimation

Task: Estimate the 3D layout from a single image



Room Layout Estimation

Task: Estimate the 3D layout from a single image



QUESTION: How would you do this?

Contents

- Definition of the Problem

Contents

- Definition of the Problem
- Parameterization (in order of structure)
  - Pixel labeling
  - 3D cuboid
  - Rays originating from Vanishing points

Contents

- Definition of the Problem
- Parameterization (in order of structure)
    - Pixel labeling
    - 3D cuboid
    - Rays originating from Vanishing points
- Inference (depends on the parameterization and energy)
    - Greedy
    - Sampling
    - Move making algorithms
    - Dynamic Programming
    - Message Passing
    - Exact inference: branch and bound

Contents

- Definition of the Problem
- Parameterization (in order of structure)
  - Pixel labeling
  - 3D cuboid
  - Rays originating from Vanishing points
- Inference (depends on the parameterization and energy)
  - Greedy
  - Sampling
  - Move making algorithms
  - Dynamic Programming
  - Message Passing
  - Exact inference: branch and bound
- Learning:
  - Ad hoc
  - Structure prediction: ranking, structure SVMs, CRFs (log loss)

No structure: Pixel Labeling

Underlying Assumption: Manhattan World

- Layout and the Objects are oriented with 3 dominant orientations which are orthogonal [Lee et al. 09]

Geometric Context

[Hoiem et al., 2007]

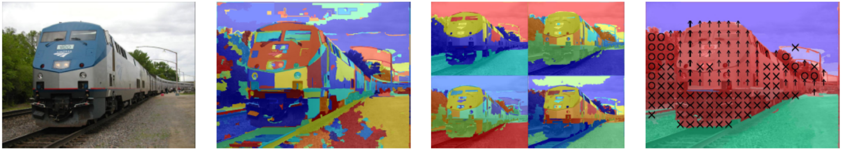
D. Hoiem, A. A. Efros, M. Hebert, Recovering Surface Layout from an Image, *IJCV*, Vol. 75, No. 1, 2007

Code and data: http://web.engr.illinois.edu/~dhoiem/projects/context/

- A rough sense of the scene geometry can be obtained from a single image by learning appearance-based models of surfaces at various orientations
- Originally developed for outdoor scenes: Ground, Sky, Vertical (left, center, right, porous, solid)

Figure : (Hoiem et al. 07)

- Built sequentially: from pixel to super pixels to regions



- **Generating segmentations**: Use agglomerative clustering with learned affinities to merge regions. Different segmentations use different feature combinations.
- **Generate Labelings**: build classifiers and average the likelihood of the classifiers on the different segmentations. They used Adaboost with decision trees.
- **Inference**: Greedy (independent for each pixel)

Feature Descriptions	Num
Color	**16**
C1. RGB values: mean	3
C2. HSV values: C1 in HSV space	3
C3. Hue: histogram (5 bins) and entropy	6
C4. Saturation: histogram (3 bins) and entropy	4
Texture	**15**
T1. DOOG filters: mean abs response of 12 filters	12
T2. DOOG stats: mean of variables in T1	1
T3. DOOG stats: argmax of variables in T1	1
T4. DOOG stats: (max - median) of variables in T1	1
Location and Shape	**12**
L1. Location: normalized x and y, mean	2
L2. Location: norm. x and y, 10^{th} and 90^{th} pctl	4
L3. Location: norm. y wrt horizon, 10^{th}, 90^{th} pctl	2
L4. Shape: number of superpixels in region	1
L5. Shape: number of sides of convex hull	1
L6. Shape: $num\ pixels/area(convex\ hull)$	1
L7. Shape: whether the region is contiguous $\in \{0, 1\}$	1
3D Geometry	**35**
G1. Long Lines: total number in region	1
G2. Long Lines: % of nearly parallel pairs of lines	1
G3. Line Intsctn: hist. over 12 orientations, entropy	13
G4. Line Intsctn: % right of center	1
G5. Line Intsctn: % above center	1
G6. Line Intsctn: % far from center at 8 orientations	8
G7. Line Intsctn: % very far from center at 8 orient.	8
G8. Texture gradient: x and y “edginess” (T2) center	2

V. Hedau, D. Hoiem, D. Forsyth, Recovering the Spatial Layout of Cluttered Rooms, *ICCV*, 2009
Code and data: http://vision.cs.uiuc.edu/~vhedau2/Research/research_spatialLayout.html

- GC modified by (Hedau et al. 09) to handle indoor scenes
- 6 Classes: Left-wall, right-wall, front-wall, ceiling, floor and object
- **Features**: color, texture, edge, and vanishing point cues computed over each segment
- A boosted decision tree classifier estimates the likelihood that a segment is valid (contains only one type of label) and likelihood of each possible label
- These likelihoods are then integrated pixel-wise over the segmentations to provide label confidences for each superpixel

- Was created by (Hedau et al. 09)
- Contains 204 training and 104 test images collected from the web
- GT surface labeling: floor, left-wall, right-wall, ceiling, object

Figure : Projection of the 3D box into the image

Metrics

- Despite the fact that we are after 3D, the metric used is the % of pixels that have the correct wall labeled

Metrics

- Despite the fact that we are after 3D, the metric used is the % of pixels that have the correct wall labeled
- Corner Error: RMSE of the image diagonal.

Metrics

- Despite the fact that we are after 3D, the metric used is the % of pixels that have the correct wall labeled
- Corner Error: RMSE of the image diagonal.
- What happens when the front wall is not present?

Metrics

- Despite the fact that we are after 3D, the metric used is the % of pixels that have the correct wall labeled
- Corner Error: RMSE of the image diagonal.
- What happens when the front wall is not present?
- Alternatively we could compute IOU of free-space

Metrics

- Despite the fact that we are after 3D, the metric used is the % of pixels that have the correct wall labeled
- Corner Error: RMSE of the image diagonal.
- What happens when the front wall is not present?
- Alternatively we could compute IOU of free-space
- 3D metrics are tricky as a small change in 2D can be a large change in 3D

Metrics

- Despite the fact that we are after 3D, the metric used is the % of pixels that have the correct wall labeled
- Corner Error: RMSE of the image diagonal.
- What happens when the front wall is not present?
- Alternatively we could compute IOU of free-space
- 3D metrics are tricky as a small change in 2D can be a large change in 3D
- But, that's the reason why is difficult in the first place!

Geometric Context: Results

	OM	GC	OM/GC	Other	GC/Oth	OM/Oth	Time
[Hoiem07]	-	28.9	-	-	-	-	-
[Hedau09] (a)	-	26.5	-	-	-	-	-

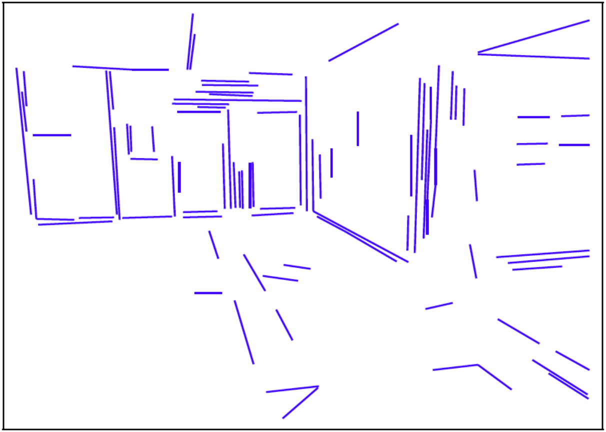
Table : Pixel classification error in the layout dataset of (Hedau et al. 09).

Orientation maps

[Lee et al., 2009]

D. C. Lee, M. Hebert, T. Kanade, Geometric Reasoning for Single Image Structure Recovery. *CVPR*, 2009
Code: https://www.cs.cmu.edu/~dclee/code/index.html

- Can you recognize the structure given only lines?



Orientation maps

[Lee et al., 2009]

- Given a line segment with end points p_1 and p_2, create the convex hull by sweeping the line α in the direction of the VP
- Do the sweep until the region contains a line that "blocks" the sweep
- A pixel is believed to have orientation z when two lines of different orientation x and y support the pixel, and only when it is exclusively supported to be z

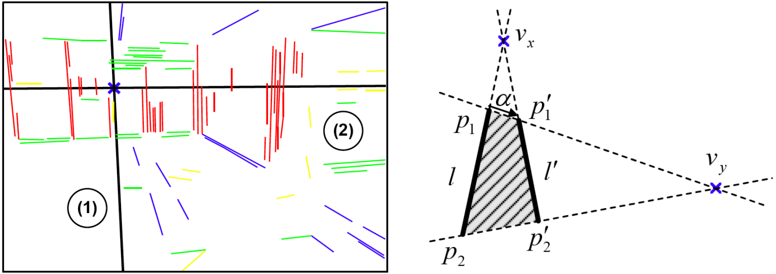


Figure : (Lee et al. 09)

Orientation maps

- Given a line segment with end points p_1 and p_2, create the convex hull by sweeping the line α in the direction of the VP
- Do the sweep until the region contains a line that "blocks" the sweep
- A pixel is believed to have orientation z when two lines of different orientation x and y support the pixel, and only when it is exclusively supported to be z

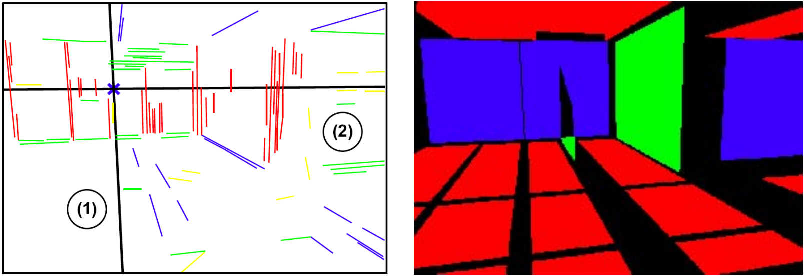


Figure : (Lee et al. 09)

OM Results

	OM	GC	OM/GC	Other	GC/Oth	OM/Oth	Time
[Hoiem07]	-	28.9	-	-	-	-	-
[Hedau09] (a)	-	26.5	-	-	-	-	-
[Lee10] w/o	24.7	22.7	18.6	-	-	-	-

Table : Pixel classification error in the layout dataset of (Hedau et al. 09).

A bit more Structure: Pixel Labeling

Semantic Segmentation

- Hey, I know about semantic segmentation!!!

Semantic Segmentation

- Hey, I know about semantic segmentation!!!
- Formulate the problem with one label per face

Semantic Segmentation

- Hey, I know about semantic segmentation!!!
- Formulate the problem with one label per face
- Write down a Markov Random Field (MRF) that uses the independent predictors (that we just reviewed) ...

$$p(\mathbf{f}|\mathbf{x}) = \frac{1}{Z}\exp\left(-E(\mathbf{f})\right)$$

Semantic Segmentation

- Hey, I know about semantic segmentation!!!
- Formulate the problem with one label per face
- Write down a Markov Random Field (MRF) that uses the independent predictors (that we just reviewed) ...

$$p(\mathbf{f}|\mathbf{x}) = \frac{1}{Z} \exp\left(-E(\mathbf{f})\right)$$

- ... but also relationships between neighboring pixels

$$E(\mathbf{f}) = \lambda \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, f_q)$$

Semantic Segmentation

- Hey, I know about semantic segmentation!!!
- Formulate the problem with one label per face
- Write down a Markov Random Field (MRF) that uses the independent predictors (that we just reviewed) ...

$$p(\mathbf{f}|\mathbf{x}) = \frac{1}{Z}\exp\left(-E(\mathbf{f})\right)$$

- ... but also relationships between neighboring pixels

$$E(\mathbf{f}) = \lambda \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q)\in\mathcal{N}} V_{pq}(f_p, f_q)$$

- How would you define $V_{pq}(f_p, f_q)$?

Semantic Segmentation

- Hey, I know about semantic segmentation!!!
- Formulate the problem with one label per face
- Write down a Markov Random Field (MRF) that uses the independent predictors (that we just reviewed) ...

$$p(\mathbf{f}|\mathbf{x}) = \frac{1}{Z} \exp\left(-E(\mathbf{f})\right)$$

- ... but also relationships between neighboring pixels

$$E(\mathbf{f}) = \lambda \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, f_q)$$

- How would you define $V_{pq}(f_p, f_q)$?
- How would you do inference?

Semantic Segmentation

- Hey, I know about semantic segmentation!!!
- Formulate the problem with one label per face
- Write down a Markov Random Field (MRF) that uses the independent predictors (that we just reviewed) ...

$$p(\mathbf{f}|\mathbf{x}) = \frac{1}{Z} \exp\left(-E(\mathbf{f})\right)$$

- ... but also relationships between neighboring pixels

$$E(\mathbf{f}) = \lambda \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, f_q)$$

- How would you define $V_{pq}(f_p, f_q)$?
- How would you do inference?
- The answer depends on your choose of $V_{pq}(f_p, f_q)$

Semantic Segmentation

- Hey, I know about semantic segmentation!!!
- Formulate the problem with one label per face
- Write down a Markov Random Field (MRF) that uses the independent predictors (that we just reviewed) ...

$$p(\mathbf{f}|\mathbf{x}) = \frac{1}{Z} \exp\left(-E(\mathbf{f})\right)$$

- ... but also relationships between neighboring pixels

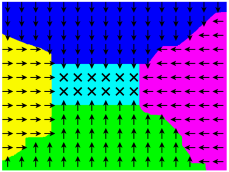
$$E(\mathbf{f}) = \lambda \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, f_q)$$

- How would you define $V_{pq}(f_p, f_q)$?
- How would you do inference?
- The answer depends on your choose of $V_{pq}(f_p, f_q)$
- Let's think of less general potentials, but more specific for the problem

Ordering Constraints

[Liu et al., 2008]

X. Liu, O. Veksler, J. Samarabandu, Graph Cut with Ordering Constraints on Labels and its Applications, *CVPR*, 2009

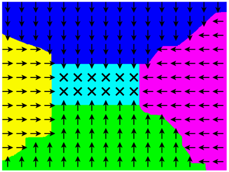


- Five Labeling problem: "center", "left", "right", "top", and "bottom"
- The front wall is a rectangle!

Ordering Constraints

[Liu et al., 2008]

X. Liu, O. Veksler, J. Samarabandu, Graph Cut with Ordering Constraints on Labels and its Applications, *CVPR*, 2009



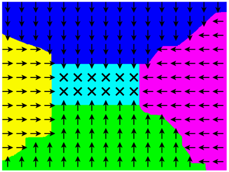
- Five Labeling problem: "center", "left", "right", "top", and "bottom"
- The front wall is a rectangle!

1. a "left" pixel cannot be to the right of a pixel with any other label;

Ordering Constraints

[Liu et al., 2008]

X. Liu, O. Veksler, J. Samarabandu, Graph Cut with Ordering Constraints on Labels and its Applications, *CVPR*, 2009



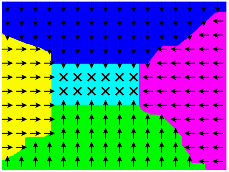
- Five Labeling problem: "center", "left", "right", "top", and "bottom"
- The front wall is a rectangle!

1. a "left" pixel cannot be to the right of a pixel with any other label;
2. a "right" pixel cannot be the left of a pixel with any other label;

Ordering Constraints

[Liu et al., 2008]

X. Liu, O. Veksler, J. Samarabandu, Graph Cut with Ordering Constraints on Labels and its Applications, *CVPR*, 2009



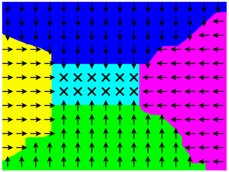
- Five Labeling problem: "center", "left", "right", "top", and "bottom"
- The front wall is a rectangle!

1. a "left" pixel cannot be to the right of a pixel with any other label;
2. a "right" pixel cannot be the left of a pixel with any other label;
3. a "top"pixel cannot be below a pixel with any other label;

Ordering Constraints

[Liu et al., 2008]

X. Liu, O. Veksler, J. Samarabandu, Graph Cut with Ordering Constraints on Labels and its Applications, *CVPR*, 2009

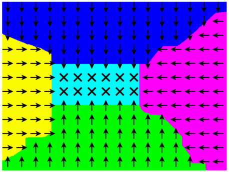


- Five Labeling problem: "center", "left", "right", "top", and "bottom"
- The front wall is a rectangle!

1. a "left" pixel cannot be to the right of a pixel with any other label;
2. a "right" pixel cannot be the left of a pixel with any other label;
3. a "top"pixel cannot be below a pixel with any other label;
4. a "bottom" pixel cannot be above a pixel with any other label;

Ordering Constraints [Liu et al., 2008]

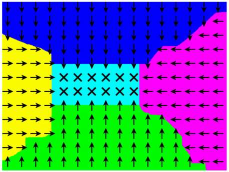
X. Liu, O. Veksler, J. Samarabandu, Graph Cut with Ordering Constraints on Labels and its Applications, *CVPR*, 2009



- Five Labeling problem: "center", "left", "right", "top", and "bottom"
- The front wall is a rectangle!

1. a "left" pixel cannot be to the right of a pixel with any other label;
2. a "right" pixel cannot be the left of a pixel with any other label;
3. a "top" pixel cannot be below a pixel with any other label;
4. a "bottom" pixel cannot be above a pixel with any other label;
5. if the neighbor of a "center" pixel has other label, then the neighbor has to be labeled as "left", "right", "top", or "bottom" if it is to the L,R,A,B respectively.

X. Liu, O. Veksler, J. Samarabandu, Graph Cut with Ordering Constraints on Labels and its Applications, *CVPR*, 2009



- Five Labeling problem: "center", "left", "right", "top", and "bottom"
- The front wall is a rectangle!

1. a "left" pixel cannot be to the right of a pixel with any other label;
2. a "right" pixel cannot be the left of a pixel with any other label;
3. a "top" pixel cannot be below a pixel with any other label;
4. a "bottom" pixel cannot be above a pixel with any other label;
5. if the neighbor of a "center" pixel has other label, then the neighbor has to be labeled as "left", "right", "top", or "bottom" if it is to the L,R,A,B respectively.
6. The "center" region is a rectangle.

- Let f_p be the label for each pixel
- Formulate the problem as Energy Minimization

$$E(\mathbf{f}) = \lambda \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, f_q)$$

- The pairwise potential defines ordering constraints

Vertical Neighbors $p=(x,y), q=(x,y+1)$					
$f_p \backslash f_q$	L	R	C	T	B
L	0	∞	∞	∞	w_{pq}
R	∞	0	∞	∞	w_{pq}
C	∞	∞	0	∞	w_{pq}
T	w_{pq}	w_{pq}	w_{pq}	0	∞
B	∞	∞	∞	∞	0

Horizontal Neighbors $p=(x,y), q=(x+1,y)$					
$f_p \backslash f_q$	L	R	C	T	B
L	0	∞	w_{pq}	w_{pq}	w_{pq}
R	∞	0	∞	∞	∞
C	∞	w_{pq}	0	∞	∞
T	∞	w_{pq}	∞	0	∞
B	∞	w_{pq}	∞	∞	0

MRF formulation

[Liu et al., 2008]

- Let f_p be the label for each pixel
- Formulate the problem as Energy Minimization

$$E(\mathbf{f}) = \lambda \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, f_q)$$

- The pairwise potential defines ordering constraints

Vertical Neighbors $p = (x, y), q = (x, y+1)$					
$f_p \backslash f_q$	L	R	C	T	B
L	0	∞	∞	∞	w_{pq}
R	∞	0	∞	∞	w_{pq}
C	∞	∞	0	∞	w_{pq}
T	w_{pq}	w_{pq}	w_{pq}	0	∞
B	∞	∞	∞	∞	0

Horizontal Neighbors $p = (x, y), q = (x+1, y)$					
$f_p \backslash f_q$	L	R	C	T	B
L	0	∞	w_{pq}	w_{pq}	w_{pq}
R	∞	0	∞	∞	∞
C	∞	w_{pq}	0	∞	∞
T	∞	w_{pq}	∞	0	∞
B	∞	w_{pq}	∞	∞	0

- Question: How can we do inference?

Move Making Algorithms

- Unlike regular binary energies, optimal solution is not possible in multi-label problems
- Proceed by solving to optimality subproblems that include current iterate
- This guarantees decrease in the objective

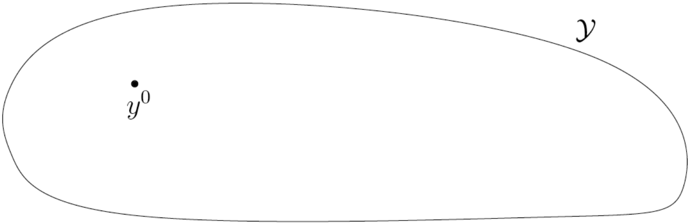


Figure : from (Nowozin et al)

Move Making Algorithms

- Unlike regular binary energies, optimal solution is not possible in multi-label problems
- Proceed by solving to optimality subproblems that include current iterate
- This guarantees decrease in the objective

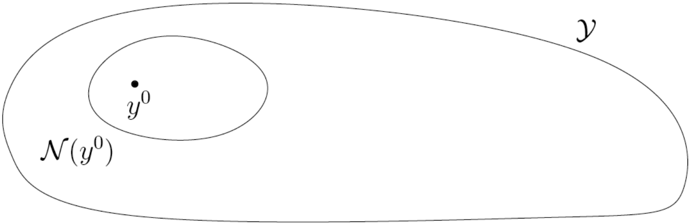


Figure : from (Nowozin et al)

Move Making Algorithms

- Unlike regular binary energies, optimal solution is not possible in multi-label problems
- Proceed by solving to optimality subproblems that include current iterate
- This guarantees decrease in the objective

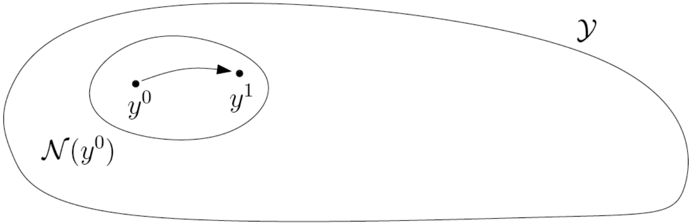


Figure : from (Nowozin et al)

Move Making Algorithms

- Unlike regular binary energies, optimal solution is not possible in multi-label problems
- Proceed by solving to optimality subproblems that include current iterate
- This guarantees decrease in the objective

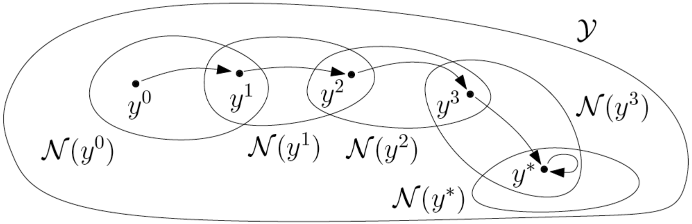


Figure : from (Nowozin et al)

Move Making Algorithms

- **Alpha Expansion**: Checks if current nodes want to switch to label α
- **Alpha - Beta Swaps**: Checks if a node with class α wants to switch to β.
- Binary problems that can be solve exactly for certain type of potentials

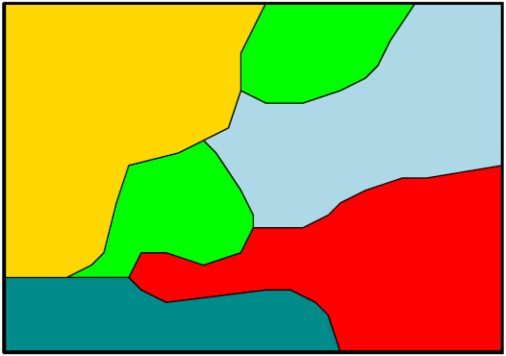


Figure : Alpha-beta Swaps. Figure from (Nowozin et al)

Move Making Algorithms

- **Alpha Expansion**: Checks if current nodes want to switch to label α
- **Alpha - Beta Swaps**: Checks if a node with class α wants to switch to β.
- Binary problems that can be solve exactly for certain type of potentials

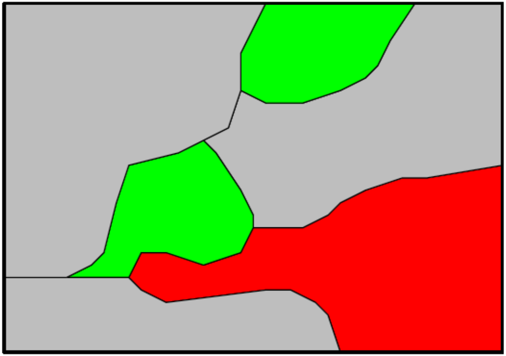


Figure : Alpha-beta Swaps. Figure from (Nowozin et al)

Move Making Algorithms

- **Alpha Expansion**: Checks if current nodes want to switch to label α
- **Alpha - Beta Swaps**: Checks if a node with class α wants to switch to β.
- Binary problems that can be solve exactly for certain type of potentials

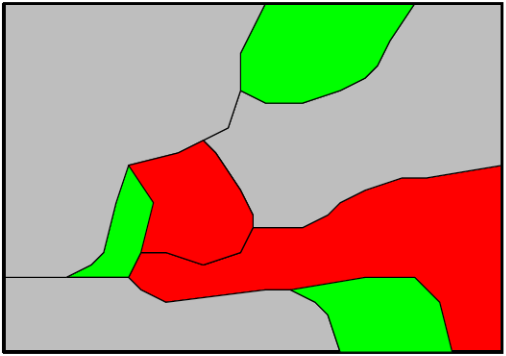


Figure : Alpha-beta Swaps. Figure from (Nowozin et al)

Move Making Algorithms

- **Alpha Expansion**: Checks if current nodes want to switch to label α
- **Alpha - Beta Swaps**: Checks if a node with class α wants to switch to β.
- Binary problems that can be solve exactly for certain type of potentials

Figure : Alpha-beta Swaps. Figure from (Nowozin et al)

Binary Moves

- $\alpha - \beta$ moves works for semi-metrics
- α expansion works for V being a metric

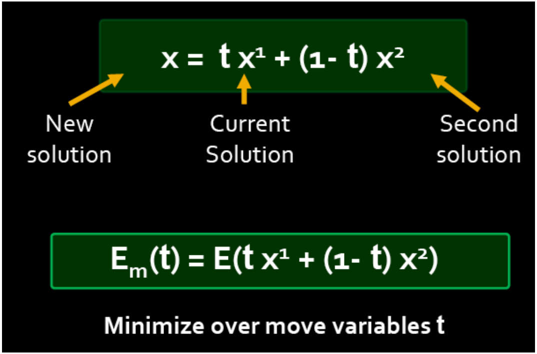


Figure : from P. Kohli tutorial on graph-cuts

- For certain x^1 and x^2, the move energy is sub-modular and can be solved via graph-cuts

α-Expansion on Our problem

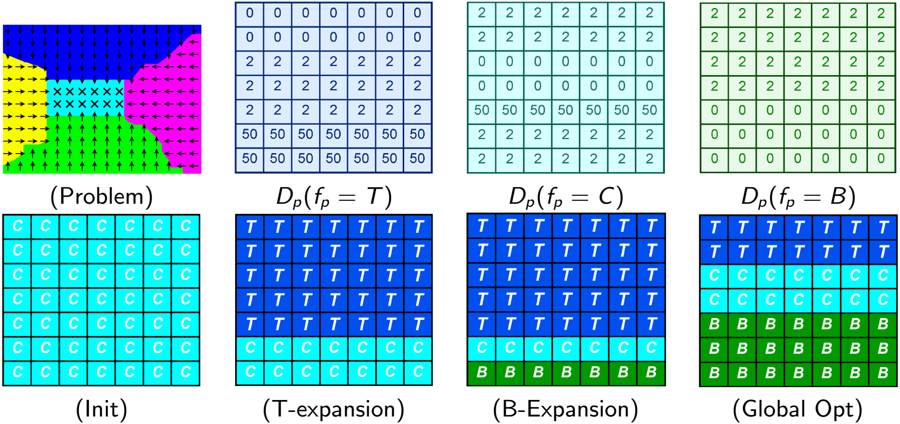
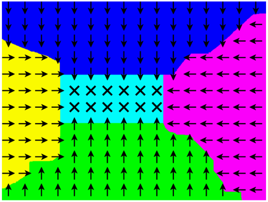


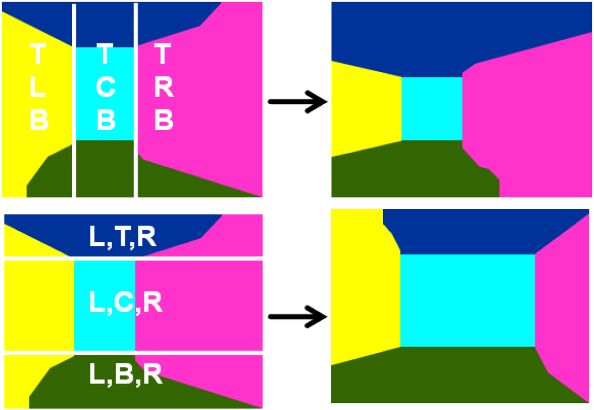
Figure : Illustration of Local Minima Problem (Liu et al. 08)

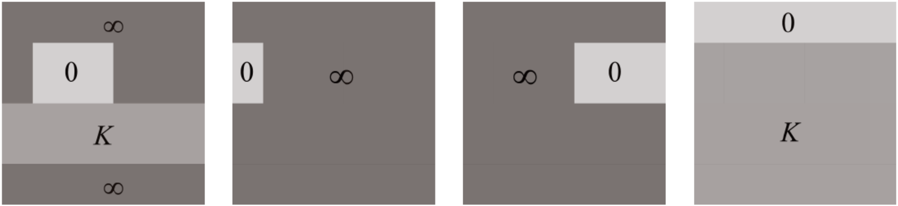
Can we derive an inference algorithm that uses the structure of the problem?

Problem-Specific Moves

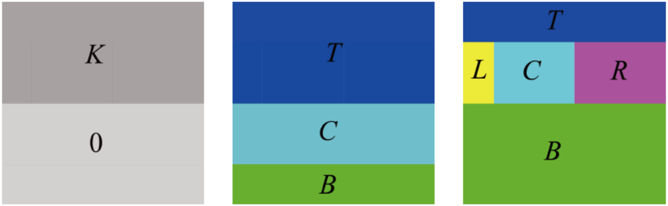
- Use the structure to derive specific moves: vertical and horizontal
- Although its a 3-label problem, it can be optimally solved via graph-cuts (see Liu et al. 08 for graph construction)
- Why 3 labels?







(a) Data terms C (b) Data terms L (c) Data terms R (d) Data terms T



(e) Data terms B (f) Local minima (g) Optimum

Figure : Illustration of the local minima problem (Bai et al. 12)

Can we do even better and get the global optima?

Yes we can!

J. Bai, Q. Song, O. Veksler, X. Wu, Fast Dynamic Programming for Labeling Problems with Ordering Constraints, *CVPR*, 2012

- It turns out that this problem is NOT NP-hard
- Caution: This assumes that the front wall is a rectangle, and the curves are monotonic!
- Trick: Go over all possible rectangles, and for each the computation is much simpler

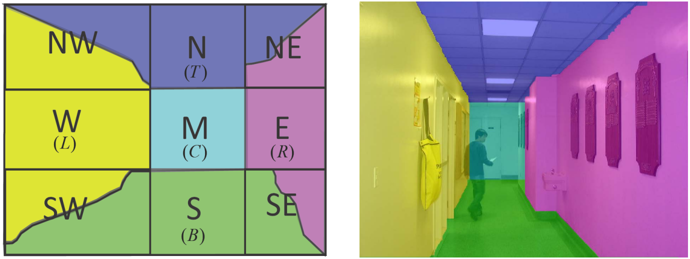
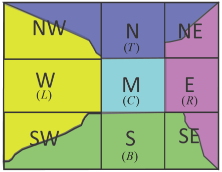
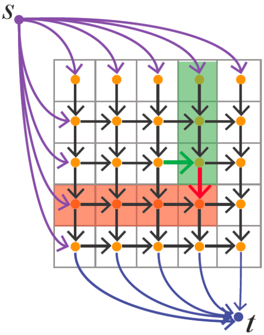


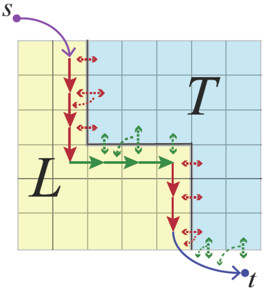
Figure : (Bai et al. 12)

Efficient Dynamic Programming [Bai et al., 2012]

- The quadrants N, W, M, E and S are fixed given the front wall.
- NW, SW, NE and SE, we want to estimate a monotonic curve
- Dynamic programing algorithm that does shortest path
- Use of integral images to accelerate computation
- $\mathcal{O}(N^{1.5})$ computation: and $\mathcal{O}(N)$ memory, with $N = w \times h$







- Use 300 images of (Liu et al 08)
- Same results as (Liu et al 08), but half the time (≈20s/image)

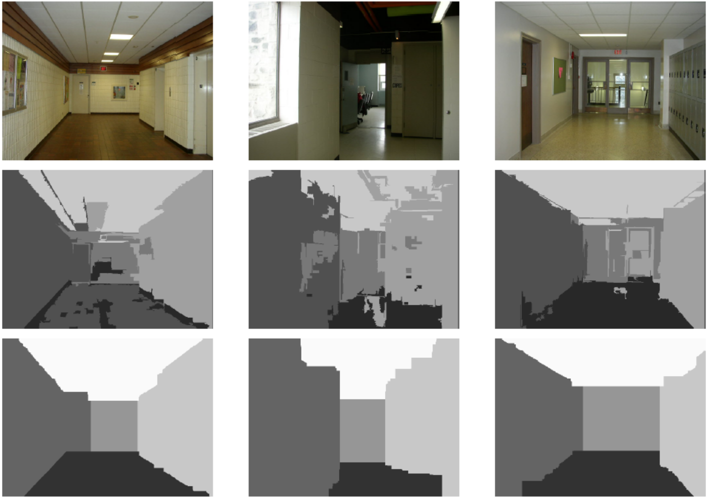


Figure : (Bai et al. 12)

Qualitative Results

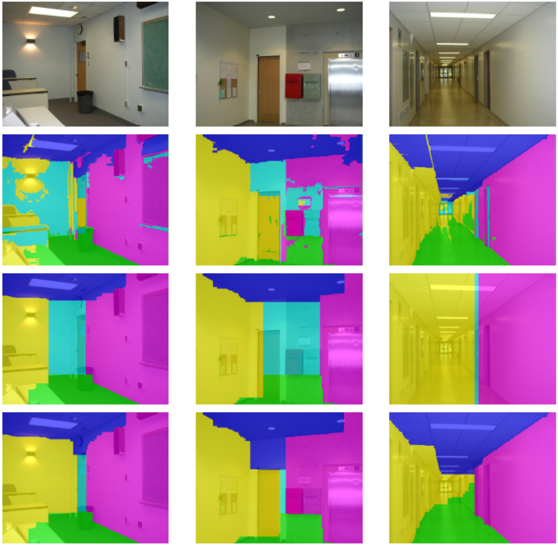


Figure : (Bai et al. 12)

Beyond Pixels: Use the <u>Structure</u> of the Problem

Room layout as a 3D Bounding Box

- Predict the 3D parametric cuboid that best describes the layout.

Room layout as a 3D Bounding Box

- Predict the 3D parametric cuboid that best describes the layout.

Room layout as a 3D Bounding Box

- Predict the 3D parametric cuboid that best describes the layout.

- How many degrees of freedom do we need?

Room as a cuboid

L. Del Pero, J. G. E. Brau, J. Schlecht, K. Barnard, Sampling Bedrooms, *CVPR*, 2011

- The floor is constrained to be parallel to the $x - z$ plane, and the room box can only rotate around the vertical axis

L. Del Pero, J. G. E. Brau, J. Schlecht, K. Barnard, Sampling Bedrooms, *CVPR*, 2011

- The floor is constrained to be parallel to the $x - z$ plane, and the room box can only rotate around the vertical axis
- The room is represented

$$r_b = (x_r, y_b, z_b, w_b, h_b, l_b, \gamma)$$

with (x_r, y_b, z_b) the coordinates of the room centre in 3D, (w_b, h_b, l_b) are the with, height and length and γ is the angle of rotation

Room as a cuboid [del Pero et al., 2011]

L. Del Pero, J. G. E. Brau, J. Schlecht, K. Barnard, Sampling Bedrooms, *CVPR*, 2011

- The floor is constrained to be parallel to the $x - z$ plane, and the room box can only rotate around the vertical axis
- The room is represented

$$r_b = (x_r, y_b, z_b, w_b, h_b, l_b, \gamma)$$

  with (x_r, y_b, z_b) the coordinates of the room centre in 3D, (w_b, h_b, l_b) are the with, height and length and γ is the angle of rotation
- Intrinsics: Assume no skew and unity aspect ratio, and principal point in the center.

Room as a cuboid [del Pero et al., 2011]

L. Del Pero, J. G. E. Brau, J. Schlecht, K. Barnard, Sampling Bedrooms, *CVPR*, 2011

- The floor is constrained to be parallel to the $x - z$ plane, and the room box can only rotate around the vertical axis
- The room is represented

$$r_b = (x_r, y_b, z_b, w_b, h_b, l_b, \gamma)$$

  with (x_r, y_b, z_b) the coordinates of the room centre in 3D, (w_b, h_b, l_b) are the with, height and length and γ is the angle of rotation
- Intrinsics: Assume no skew and unity aspect ratio, and principal point in the center.
- Camera model is fully specify with

$$c = (\psi, \phi, f)$$

  with ψ, ϕ the pitch an roll angles and f the focal length

Generative Model of Rooms [del Pero et al., 2011]

- Generative model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

- The **likelihood** $p(E|\theta)$ is the prob. of matching edges (after projecting the cuboid into the image)
- The **prior** $p(\theta)$ are box constraints

- Generative model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

- The **likelihood** $p(E|\theta)$ is the prob. of matching edges (after projecting the cuboid into the image)
- The **prior** $p(\theta)$ are box constraints
- Learning: Parameters set by hand

- Generative model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

- The **likelihood** $p(E|\theta)$ is the prob. of matching edges (after projecting the cuboid into the image)
- The **prior** $p(\theta)$ are box constraints
- Learning: Parameters set by hand
- Inference: Sampling with different proposal distributions

- Generative model

$$\underbrace{p(\theta|E)}_{\textit{posterior}} \propto \underbrace{p(E|\theta)}_{\textit{likelihood}} \underbrace{p(\theta)}_{\textit{prior}}$$

- The **likelihood** $p(E|\theta)$ is the prob. of matching edges (after projecting the cuboid into the image)
- The **prior** $p(\theta)$ are box constraints
- Learning: Parameters set by hand
- Inference: Sampling with different proposal distributions
  - Use moves that change the random variable values

- Generative model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

- The **likelihood** $p(E|\theta)$ is the prob. of matching edges (after projecting the cuboid into the image)
- The **prior** $p(\theta)$ are box constraints
- Learning: Parameters set by hand
- Inference: Sampling with different proposal distributions
  - Use moves that change the random variable values
  - Use proposal distribution that proposes the camera parameters and 3D orthogonal corner given 2D corner and f (Shi et al. 04).

- Generative model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

- The **likelihood** $p(E|\theta)$ is the prob. of matching edges (after projecting the cuboid into the image)
- The **prior** $p(\theta)$ are box constraints
- Learning: Parameters set by hand
- Inference: Sampling with different proposal distributions
  - Use moves that change the random variable values
  - Use proposal distribution that proposes the camera parameters and 3D orthogonal corner given 2D corner and f (Shi et al. 04).
  - Trick: Use a lot of samples!

Generative Model of Rooms [del Pero et al., 2011]

- Generative model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

- The **likelihood** $p(E|\theta)$ is the prob. of matching edges (after projecting the cuboid into the image)
- The **prior** $p(\theta)$ are box constraints
- Learning: Parameters set by hand
- Inference: Sampling with different proposal distributions
  - Use moves that change the random variable values
  - Use proposal distribution that proposes the camera parameters and 3D orthogonal corner given 2D corner and f (Shi et al. 04).
  - Trick: Use a lot of samples!
  - Thus you need a fairly efficient likelihood computation, as the prior is usually easy

Results

	OM	GC	OM/GC	Other	GC/Oth	OM/Oth	Time
[Hoiem07]	-	28.9	-	-	-	-	-
[Hedau09] (a)	-	26.5	-	-	-	-	-
[Lee10] w/o	24.7	22.7	18.6	-	-	-	-
[delPero11]	-	-	-	26.8	-	-	10s?

Table : Pixel classification error in the layout dataset of (Hedau et al. 09).

What's next?

- Generative Model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

- How can we improve results?

What's next?

- Generative Model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

- How can we improve results?
  - Better Priors

What's next?

- Generative Model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

- How can we improve results?
  - Better Priors
  - Better Likelihood: more features

What's next?

- Generative Model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood}\,\underbrace{p(\theta)}_{prior}$$

- How can we improve results?
  - Better Priors
  - Better Likelihood: more features
  - Better Inference

What's next?

- Generative Model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

- How can we improve results?
  - Better Priors
  - Better Likelihood: more features
  - Better Inference
  - Use of other information, e.g. VPs

L. Del Pero J. Bowdish, D. Fried, B. Kermgard, E. Hartley, K. Barnard, Bayesian geometric modeling of indoor scenes, *CVPR*, 2012

- Generative Model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

L. Del Pero J. Bowdish, D. Fried, B. Kermgard, E. Hartley, K. Barnard, Bayesian geometric modeling of indoor scenes, *CVPR*, 2012

- Generative Model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood}\,\underbrace{p(\theta)}_{prior}$$

- Better Priors: Gaussian priors over ratio of width and length and over ratio of width and height

More Powerful Generative Models

[del Pero et al., 2012]

L. Del Pero J. Bowdish, D. Fried, B. Kermgard, E. Hartley, K. Barnard, Bayesian geometric modeling of indoor scenes, *CVPR*, 2012

- Generative Model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

- Better Priors: Gaussian priors over ratio of width and length and over ratio of width and height
- Better Likelihood: count "right" OM features on the faces of the room

More Powerful Generative Models [del Pero et al., 2012]

L. Del Pero J. Bowdish, D. Fried, B. Kermgard, E. Hartley, K. Barnard, Bayesian geometric modeling of indoor scenes, *CVPR*, 2012

- Generative Model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood}\, \underbrace{p(\theta)}_{prior}$$

- Better Priors: Gaussian priors over ratio of width and length and over ratio of width and height
- Better Likelihood: count "right" OM features on the faces of the room
- Better Inference:
  - Init camera parameters from the VPs
  - Init proposals from corners detected in the image
  - Keep best 20 and multithread sampling strategy

Results on Layout Dataset

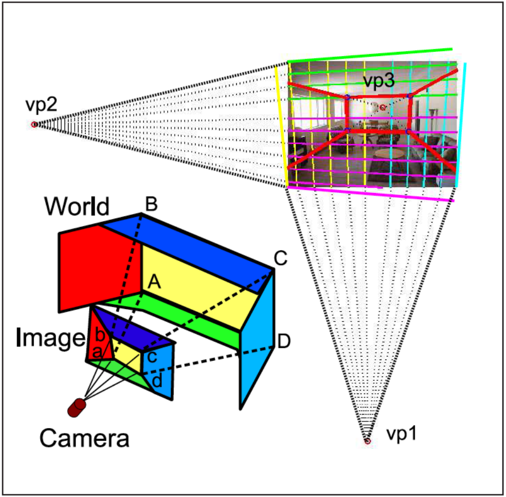
	OM	GC	OM/GC	Other	GC/Oth	OM/Oth	Time
[Hoiem07]	-	28.9	-	-	-	-	-
[Hedau09] (a)	-	26.5	-	-	-	-	-
[Lee10] w/o	24.7	22.7	18.6	-	-	-	-
[delPero11]	-	-	-	26.8	-	-	X min
[delPero12]	-	-	-	24.7	-	21.3	10s?

Table : Pixel classification error in the layout dataset of (Hedau et al. 09).

Even more structure

Utilizing Vanishing Points

V. Hedau, D. Hoiem, D. Forsyth, Recovering the Spatial Layout of Cluttered Rooms, *ICCV*, 2009



- If you know VPs, there are only 4 dof left, and e.g., 50^4 boxes!

Formal Parameterization

[Hedau et al., 2009]

- $\mathbf{x}$ is an image, and $\mathbf{y}$ is a layout
- Energy minimization task (max score/probability):

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$$

with $\phi(\mathbf{x}, \mathbf{y})$ potentials based on image features

Formal Parameterization

[Hedau et al., 2009]

- $\mathbf{x}$ is an image, and $\mathbf{y}$ is a layout
- Energy minimization task (max score/probability):

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$$

  with $\phi(\mathbf{x}, \mathbf{y})$ potentials based on image features

- How do we incorporate our prior knowledge?

Formal Parameterization

[Hedau et al., 2009]

- $\mathbf{x}$ is an image, and $\mathbf{y}$ is a layout
- Energy minimization task (max score/probability):

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$$

  with $\phi(\mathbf{x}, \mathbf{y})$ potentials based on image features
- How do we incorporate our prior knowledge?
- How do we construct $\phi(\mathbf{x}, \mathbf{y})$?

Formal Parameterization

[Hedau et al., 2009]

- **x** is an image, and **y** is a layout
- Energy minimization task (max score/probability):

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$$

  with $\phi(\mathbf{x}, \mathbf{y})$ potentials based on image features
- How do we incorporate our prior knowledge?
- How do we construct $\phi(\mathbf{x}, \mathbf{y})$?
- **Learning**: How do we score a 3D box?

Formal Parameterization

[Hedau et al., 2009]

- $\mathbf{x}$ is an image, and $\mathbf{y}$ is a layout
- Energy minimization task (max score/probability):

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$$

  with $\phi(\mathbf{x}, \mathbf{y})$ potentials based on image features
- How do we incorporate our prior knowledge?
- How do we construct $\phi(\mathbf{x}, \mathbf{y})$?
- **Learning**: How do we score a 3D box?
- **Inference**: How do we reason about all possible 3D boxes?

We need to compute $\phi(\mathbf{x}, \mathbf{y})$

1. **Weighted line membership**: Sum the lines of a particular VP vs all other lines in the face

Figure : (Hedau et al. 09)

We need to compute $\phi(\mathbf{x}, \mathbf{y})$

1. **Weighted line membership**: Sum the lines of a particular VP vs all other lines in the face

Figure : (Hedau et al. 09)

- For a wall, lines appear mainly on two orientations.

We need to compute $\phi(\mathbf{x}, \mathbf{y})$

1. **Weighted line membership**: Sum the lines of a particular VP vs all other lines in the face

Figure : (Hedau et al. 09)

- For a wall, lines appear mainly on two orientations.
- Objects violate this: weight the lines by conf. of been inside an object region

Let's look at Hedau et al. 09

2. For each face, compute the **normalized sum of the geometric context features**

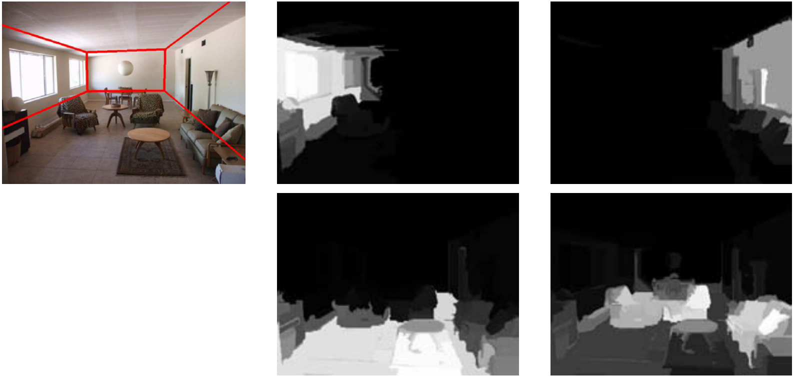


Figure : (Hedau et al. 09)

- "Sample" a set of 3D box candidates, e.g., 200

Figure : (Hedau et al. 09)

Learning a Scoring Function

[Hedau et al., 2009]

- Use Structure Prediction to learn the scoring function

- Use Structure Prediction to learn the scoring function
- Formulate the problem as **structured ranking**, which involves minimizing the following QP:

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_i \xi_i \\ \text{s.t.} \quad & \xi_i \geq 0 \qquad \forall i \\ & \mathbf{w}^T\phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \mathbf{w}^T\phi(\mathbf{x}^{(i)}, \mathbf{y}) \geq \Delta(\mathbf{y}^{(i)}, \mathbf{y}) - \xi_i \qquad \forall i, \forall \mathbf{y} \in \mathcal{Y} \end{aligned}$$

with ξ_i the slack variables and $\Delta(\mathbf{y}^{(i)}, \mathbf{y})$ the loss function

- Use Structure Prediction to learn the scoring function
- Formulate the problem as **structured ranking**, which involves minimizing the following QP:

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_i \xi_i \\ \text{s.t.} \quad & \xi_i \geq 0 \qquad \forall i \\ & \mathbf{w}^T\phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \mathbf{w}^T\phi(\mathbf{x}^{(i)}, \mathbf{y}) \geq \Delta(\mathbf{y}^{(i)}, \mathbf{y}) - \xi_i \qquad \forall i, \forall \mathbf{y} \in \mathcal{Y} \end{aligned}$$

  with ξ_i the slack variables and $\Delta(\mathbf{y}^{(i)}, \mathbf{y})$ the loss function
- The loss function $\Delta(\mathbf{y}^{(i)}, \mathbf{y})$ penalizes deviation from the GT

Learning a Scoring Function

[Hedau et al., 2009]

- Use Structure Prediction to learn the scoring function
- Formulate the problem as **structured ranking**, which involves minimizing the following QP:

$$
\begin{aligned}
\min_{w,\xi} \quad & \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_i \xi_i \\
\text{s.t.} \quad & \xi_i \geq 0 \qquad \forall i \\
& \mathbf{w}^T\phi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) - \mathbf{w}^T\phi(\mathbf{x}^{(i)}, \mathbf{y}) \geq \Delta(\mathbf{y}^{(i)}, \mathbf{y}) - \xi_i \qquad \forall i, \forall \mathbf{y} \in \mathcal{Y}
\end{aligned}
$$

  with ξ_i the slack variables and $\Delta(\mathbf{y}^{(i)}, \mathbf{y})$ the loss function
- The loss function $\Delta(\mathbf{y}^{(i)}, \mathbf{y})$ penalizes deviation from the GT
- Their loss function penalizes
  - the absence of a face,
  - the shift of the centroid of the faces
  - the sum of pixel errors for all faces.

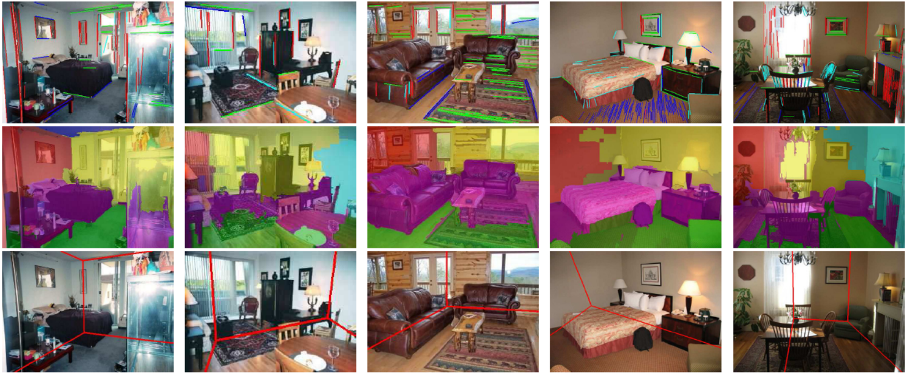
Results on Layout Dataset

	OM	GC	OM/GC	Other	GC/Oth	OM/Oth	Time
[Hoiem07]	-	28.9	-	-	-	-	-
[Hedau09] (a)	-	26.5	-	-	-	-	-
[Hedau09] (b)	-	-	-	-	21.2	-	10-30 min
[Lee10] w/o	24.7	22.7	18.6	-	-	-	-
[delPero11]	-	-	-	26.8	-	-	X min
[delPero12]	-	-	-	24.7	-	21.3	10s?

Table : Pixel classification error in the layout dataset of (Hedau et al. 09).

Qualitative Results

[Hedau et al., 2009]



Can we solve this problem more efficiently?

Efficient 3D Room Layout Estimation

- **Task**: Given an image, predict the 3D parametric cuboid that best describes the layout

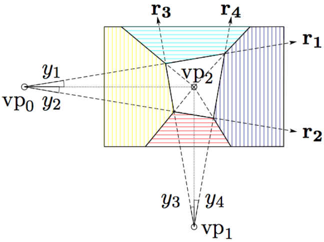
- $\mathbf{x}$ is an image, and $\mathbf{y}$ is a layout , solve via structure prediction

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$$

with $\phi(\mathbf{x}, \mathbf{y})$ potentials based on image features

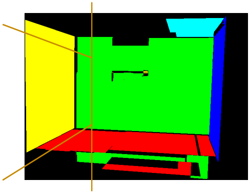
Parameterizing The Layout

- We parameterize a layout with 4 variables $y_i \in \mathcal{Y}$, $i \in \{1, ..., 4\}$ (Hedau et al. 09)

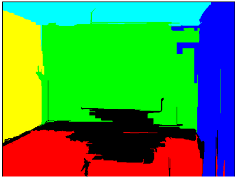


Layout Energy or Scoring Function

- Image feaures



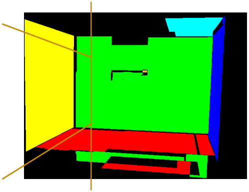
OM (Lee et al. 09)



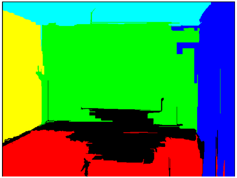
GC (Hoiem et al. 05)

Layout Energy or Scoring Function

- Image feaures



OM (Lee et al. 09)



GC (Hoiem et al. 05)

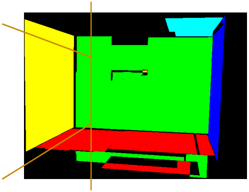
- The potentials count for each layout face the occurrence of each feature type

$$E_{full-layout}(x, \mathbf{y}) = \mathbf{w}^T \phi_{layout}(x, \mathbf{y}) = \sum_{\alpha \in \mathcal{F}} \mathbf{w}_\alpha^T \phi_\alpha(\mathbf{x}, \mathbf{y}_\alpha)$$

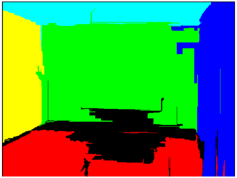
with $\mathcal{F} = \{$*left-wall*, *right-wall*, *ceiling*, *floor*, *front-wall*$\}$

Layout Energy or Scoring Function

- Image feaures



OM (Lee et al. 09)



GC (Hoiem et al. 05)

- The potentials count for each layout face the occurrence of each feature type

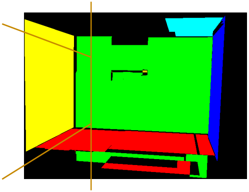
$$E_{full-layout}(x, \mathbf{y}) = \mathbf{w}^T \phi_{layout}(x, \mathbf{y}) = \sum_{\alpha \in \mathcal{F}} \mathbf{w}_\alpha^T \phi_\alpha(\mathbf{x}, \mathbf{y}_\alpha)$$

with $\mathcal{F} = \{$*left-wall*, *right-wall*, *ceiling*, *floor*, *front-wall*$\}$

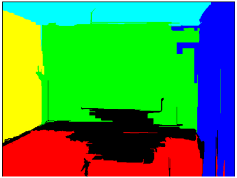
- High-order potentials a priori. Why?

Layout Energy or Scoring Function

- Image feaures



OM (Lee et al. 09)



GC (Hoiem et al. 05)

- The potentials count for each layout face the occurrence of each feature type

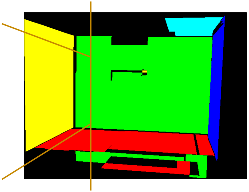
$$E_{full-layout}(x, \mathbf{y}) = \mathbf{w}^T \phi_{layout}(x, \mathbf{y}) = \sum_{\alpha \in \mathcal{F}} \mathbf{w}_\alpha^T \phi_\alpha(\mathbf{x}, \mathbf{y}_\alpha)$$

  with $\mathcal{F} = \{$*left-wall*, *right-wall*, *ceiling*, *floor*, *front-wall*$\}$

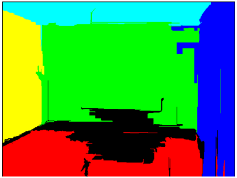
- High-order potentials a priori. Why?
- Faces are defined by four (*front-wall*) or three angles (otherwise)

Layout Energy or Scoring Function

- Image feaures



OM (Lee et al. 09)



GC (Hoiem et al. 05)

- The potentials count for each layout face the occurrence of each feature type

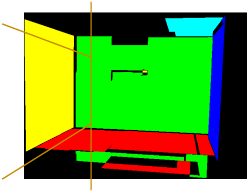
$$E_{full-layout}(x, \mathbf{y}) = \mathbf{w}^T \phi_{layout}(x, \mathbf{y}) = \sum_{\alpha \in \mathcal{F}} \mathbf{w}_\alpha^T \phi_\alpha(\mathbf{x}, \mathbf{y}_\alpha)$$

with $\mathcal{F} = \{$*left-wall*, *right-wall*, *ceiling*, *floor*, *front-wall*$\}$

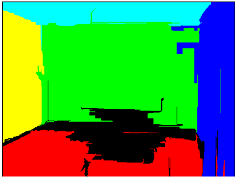
- High-order potentials a priori. Why?
- Faces are defined by four (*front-wall*) or three angles (otherwise)
- Learning done via structured prediction

Layout Energy or Scoring Function

- Image feaures



OM (Lee et al. 09)



GC (Hoiem et al. 05)

- The potentials count for each layout face the occurrence of each feature type

$$E_{full-layout}(x, \mathbf{y}) = \mathbf{w}^T \phi_{layout}(x, \mathbf{y}) = \sum_{\alpha \in \mathcal{F}} \mathbf{w}_\alpha^T \phi_\alpha(\mathbf{x}, \mathbf{y}_\alpha)$$

with $\mathcal{F} = \{$*left-wall*, *right-wall*, *ceiling*, *floor*, *front-wall*$\}$

- High-order potentials a priori. Why?
- Faces are defined by four (*front-wall*) or three angles (otherwise)
- Learning done via structured prediction
- What do you expect learning to "learn"

Inference

- Is inference easy in this model? Why?

Inference

- Is inference easy in this model? Why?
- What can we do?

Inference

- Is inference easy in this model? Why?
- What can we do?
- Multi-label problem, message passing seems the best option

Inference

- Is inference easy in this model? Why?
- What can we do?
- Multi-label problem, message passing seems the best option
- Problem: High order potentials $\rightarrow$ very very slow !

Inference

- Is inference easy in this model? Why?
- What can we do?
- Multi-label problem, message passing seems the best option
- Problem: High order potentials $\rightarrow$ very very slow !
- Let's think about it for a second, maybe we can do something

Inference

- Is inference easy in this model? Why?
- What can we do?
- Multi-label problem, message passing seems the best option
- Problem: High order potentials $\rightarrow$ very very slow !
- Let's think about it for a second, maybe we can do something
- Remember we want to compute sum of features in faces, and search over all possible faces

Inference

- Is inference easy in this model? Why?
- What can we do?
- Multi-label problem, message passing seems the best option
- Problem: High order potentials $\rightarrow$ very very slow !
- Let's think about it for a second, maybe we can do something
- Remember we want to compute sum of features in faces, and search over all possible faces
- Let's first take a detour

Integral Images

- We are interested in computing the sum of some features inside a rectangle, and we want to vary the rectangle
- How can we do this efficiently?
- Compute the **sum area table**, also called **integral image**

3	2	7	2	3
1	5	1	3	4
5	1	**3**	5	1
4	3	2	1	6
2	4	1	4	8

$$s(i,j) = \sum_{k=0}^{i} \sum_{l=0}^{j} f(k,l)$$

Integral Images

- We are interested in computing the sum of some features inside a rectangle, and we want to vary the rectangle
- How can we do this efficiently?
- Compute the **sum area table**, also called **integral image**

3	2	7	2	3
1	5	1	3	4
5	1	**3**	5	1
4	3	2	1	6
2	4	1	4	8

3	5	12	14	17
4	*11*	**19**	24	31
9	**17**	28	38	46
13	24	37	48	62
15	30	44	59	81

$$s(i,j) = \sum_{k=0}^{i} \sum_{l=0}^{j} f(k,l)$$

- This can be efficiently computed using a recursive (raster-scan) algorithm

$$s(i,j) = s(i-1,j) + s(i,j-1) - s(i-1,j-1) + f(i,j)$$

Integral Images

- We are interested in computing the sum of some features inside a rectangle, and we want to vary the rectangle
- How can we do this efficiently?
- Compute the **sum area table**, also called **integral image**

3	2	7	2	3
1	5	1	3	4
5	1	**3**	5	1
4	3	2	1	6
2	4	1	4	8

3	5	12	*14*	17
4	11	19	24	31
9	17	28	38	46
13	24	37	**48**	62
15	30	44	59	81

$$s(i,j) = \sum_{k=0}^{i} \sum_{l=0}^{j} f(k,l)$$

- This can be efficiently computed using a recursive (raster-scan) algorithm

$$s(i,j) = s(i-1,j) + s(i,j-1) - s(i-1,j-1) + f(i,j)$$

- Then compute the sum on the rectangle by accessing 4 numbers

$$S([i_0,i_1] \times [j_0,j_1]) = s(i_1,j_1) - s(i_1,j_0-1) - s(i_0-1,j_1) + s(i_0-1,j_0-1)$$

Integral Images

- We are interested in computing the sum of some features inside a rectangle, and we want to vary the rectangle
- How can we do this efficiently?
- Compute the **sum area table**, also called **integral image**

3	2	7	2	3
1	5	1	3	4
5	1	**3**	5	1
4	3	2	1	6
2	4	1	4	8

3	5	12	*14*	17
4	11	19	24	31
9	17	28	38	46
13	24	37	**48**	62
15	30	44	59	81

$$s(i,j) = \sum_{k=0}^{i} \sum_{l=0}^{j} f(k,l)$$

- This can be efficiently computed using a recursive (raster-scan) algorithm

$$s(i,j) = s(i-1,j) + s(i,j-1) - s(i-1,j-1) + f(i,j)$$

- Then compute the sum on the rectangle by accessing 4 numbers

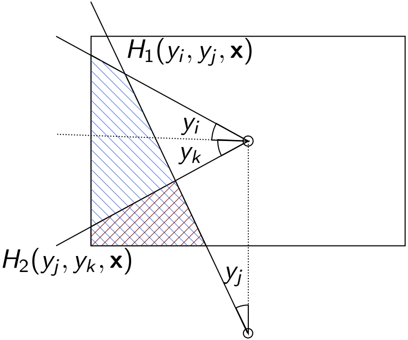
$$S([i_0,i_1] \times [j_0,j_1]) = s(i_1,j_1) - s(i_1,j_0-1) - s(i_0-1,j_1) + s(i_0-1,j_0-1)$$

- Can we do something similar in our case?

A. Schwing, T. Hazan, M. Pollefeys and R. Urtasun, Efficient Structured Prediction for 3D Indoor Scene Understanding, *CVPR*, 2012

- Faces are generalizations of rectangles
- We need to extend the concept of integral images to 3D
- This is called **integral geometry** (Schwing et al. 12a)
- How does this work?

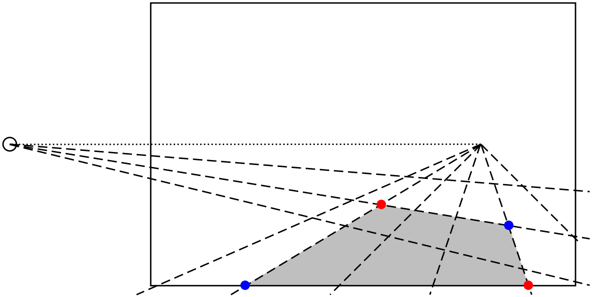
$$\phi_{\{left\_w\}}(y_i, y_j, y_k, \mathbf{x}) \quad = \quad H_1(y_i, y_j, \mathbf{x}) - H_2(y_j, y_k, \mathbf{x})$$



A. Schwing, T. Hazan, M. Pollefeys and R. Urtasun, Efficient Structured Prediction for 3D Indoor Scene Understanding, *CVPR*, 2012

- Faces are generalizations of rectangles
- We need to extend the concept of integral images to 3D
- This is called **integral geometry** (Schwing et al. 12a)
- How does this work?

$$\phi_{\{floor\}}(y_i, y_j, y_k, \mathbf{x}) \quad = \quad H_1(y_i, y_j, \mathbf{x}) - H_2(y_j, y_k, \mathbf{x})$$



What are the implications?

[Schwing et al., 2012]

- We can now write the problem in terms of potentials of order at most 2

$$E(y_1, \cdots, y_4) = \sum_r \mathbf{w}_r^T(\mathbf{y}_r, \mathbf{x})$$

and r only contains sets of 2 random variables

What are the implications?

- We can now write the problem in terms of potentials of order at most 2

$$E(y_1, \cdots, y_4) = \sum_r \mathbf{w}_r^T(\mathbf{y}_r, \mathbf{x})$$

  and r only contains sets of 2 random variables

- Life is a bit more complicated than what I showed you as I was varying the parameterization to make you understand easily

What are the implications?

[Schwing et al., 2012]

- We can now write the problem in terms of potentials of order at most 2

$$E(y_1, \cdots, y_4) = \sum_r \mathbf{w}_r^T(\mathbf{y}_r, \mathbf{x})$$

  and r only contains sets of 2 random variables

- Life is a bit more complicated than what I showed you as I was varying the parameterization to make you understand easily
- Good news is that it still depends on pairwise potentials (which are accumulators) but there is quite a few more

What are the implications? [Schwing et al., 2012]

- We can now write the problem in terms of potentials of order at most 2

$$E(y_1, \cdots, y_4) = \sum_r \mathbf{w}_r^T(\mathbf{y}_r, \mathbf{x})$$

  and r only contains sets of 2 random variables

- Life is a bit more complicated than what I showed you as I was varying the parameterization to make you understand easily
- Good news is that it still depends on pairwise potentials (which are accumulators) but there is quite a few more
- Some of these r share the same weights, as they come from the integral geometry.

- We can now write the problem in terms of potentials of order at most 2

$$E(y_1, \cdots, y_4) = \sum_r \mathbf{w}_r^T(\mathbf{y}_r, \mathbf{x})$$

  and r only contains sets of 2 random variables
- Life is a bit more complicated than what I showed you as I was varying the parameterization to make you understand easily
- Good news is that it still depends on pairwise potentials (which are accumulators) but there is quite a few more
- Some of these r share the same weights, as they come from the integral geometry.
- If they are not shared then they do not represent the same problem

What are the implications? [Schwing et al., 2012]

- We can now write the problem in terms of potentials of order at most 2

$$E(y_1, \cdots, y_4) = \sum_r \mathbf{w}_r^T (\mathbf{y}_r, \mathbf{x})$$

  and r only contains sets of 2 random variables

- Life is a bit more complicated than what I showed you as I was varying the parameterization to make you understand easily
- Good news is that it still depends on pairwise potentials (which are accumulators) but there is quite a few more
- Some of these r share the same weights, as they come from the integral geometry.
- If they are not shared then they do not represent the same problem
- This speeds up the message passing inference by a few orders of magnitude

Results on Layout Dataset

	OM	GC	OM/GC	Other	GC/Oth	OM/Oth	Time
[Hoiem07]	-	28.9	-	-	-	-	-
[Hedau09] (a)	-	26.5	-	-	-	-	-
[Hedau09] (b)	-	-	-	-	21.2	-	10-30 min
[Lee10] w/o	24.7	22.7	18.6	-	-	-	-
[delPero11]	-	-	-	26.8	-	-	10s?
[delPero12]	-	-	-	24.7	-	21.3	X min
Schwing12a	**18.6**	**15.4**	**13.6**	-	-	-	**0.15s**

Table : Pixel classification error in the layout dataset of (Hedau et al. 09).

Can we get the global optima?

Branch and Bound

Algorithm 1 branch and bound (BB) inference

put pair $(\bar{f}(\mathcal{Y}), \mathcal{Y})$ into queue and set $\hat{\mathcal{Y}} = \mathcal{Y}$
repeat
  split $\hat{\mathcal{Y}} = \hat{\mathcal{Y}}_1 \times \hat{\mathcal{Y}}_2$ with $\hat{\mathcal{Y}}_1 \cap \hat{\mathcal{Y}}_2 = \emptyset$
  put pair $(\bar{f}(\hat{\mathcal{Y}}_1), \hat{\mathcal{Y}}_1)$ into queue
  put pair $(\bar{f}(\hat{\mathcal{Y}}_2), \hat{\mathcal{Y}}_2)$ into queue
  retrieve $\hat{\mathcal{Y}}$ having highest score
until $|\hat{\mathcal{Y}}| = 1$

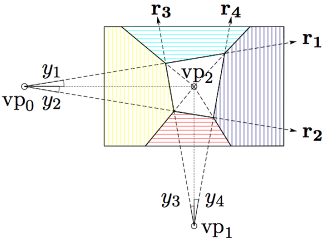
We have to define:

1. A parameterization that defines **sets of hypothesis**.
2. A **scoring function** f
3. **Tight bounds** on the scoring function that can be computed very **efficiently**

Parameterization of the Problem [Schwing & Urtasun, 2012]

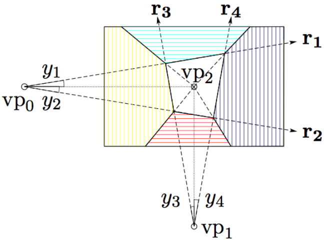
A. Schwing and R. Urtasun, Efficient Exact Inference for 3D Indoor Scene Understanding, *ECCV*, 2012

- Layout with 4 variables $y_i \in \mathcal{Y}$, $i \in \{1, ..., 4\}$
- How do we define $\mathcal{Y}$?
- Is this problem continuous or discrete?



A. Schwing and R. Urtasun, Efficient Exact Inference for 3D Indoor Scene Understanding, *ECCV*, 2012

- Layout with 4 variables $y_i \in \mathcal{Y}$, $i \in \{1, ..., 4\}$
- How do we define $\mathcal{Y}$?
- Is this problem continuous or discrete?

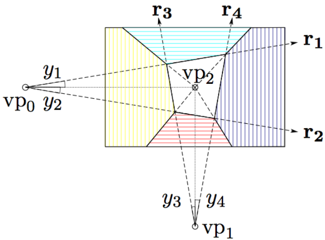


- We parameterize the sets by **intervals** of minimum and maximum angles

$$\{[y_1^{min}, y_1^{max}], \cdots, [y_4^{min}, y_4^{max}]\}$$

A. Schwing and R. Urtasun, Efficient Exact Inference for 3D Indoor Scene Understanding, *ECCV*, 2012

- Layout with 4 variables $y_i \in \mathcal{Y}$, $i \in \{1, ..., 4\}$
- How do we define $\mathcal{Y}$?
- Is this problem continuous or discrete?



- We parameterize the sets by **intervals** of minimum and maximum angles

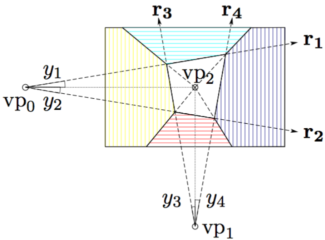
$$\{[y_1^{min}, y_1^{max}], \cdots, [y_4^{min}, y_4^{max}]\}$$

- Why intervals?

Parameterization of the Problem [Schwing & Urtasun, 2012]

A. Schwing and R. Urtasun, Efficient Exact Inference for 3D Indoor Scene Understanding, *ECCV*, 2012

- Layout with 4 variables $y_i \in \mathcal{Y}$, $i \in \{1, ..., 4\}$
- How do we define $\mathcal{Y}$?
- Is this problem continuous or discrete?



- We parameterize the sets by **intervals** of minimum and maximum angles

$$\{[y_1^{min}, y_1^{max}], \cdots, [y_4^{min}, y_4^{max}]\}$$

- Why intervals?
- We have defined already the scoring function. What about the bounds?

Properties of the Bounds

Derive bounds $\bar{f}$ for the original scoring function $\mathbf{w}^T \phi(\mathbf{y}, \mathbf{x})$ that satisfy:

1. The bound of the interval $\hat{\mathcal{Y}}$ has to upper-bound the true cost of each hypothesis $y \in \hat{\mathcal{Y}}$,
$$\forall y \in \hat{\mathcal{Y}}, \quad \bar{f}(\hat{\mathcal{Y}}) \geq \mathbf{w}^T \phi(\mathbf{y}, \mathbf{x}).$$
2. The bound has to be exact for every single hypothesis,
$$\forall y \in \mathcal{Y}, \quad \bar{f}(y) = \mathbf{w}^T \phi(\mathbf{y}, \mathbf{x}).$$

Properties of the Bounds

Derive bounds $\bar{f}$ for the original scoring function $\mathbf{w}^T \phi(\mathbf{y}, \mathbf{x})$ that satisfy:

1. The bound of the interval $\hat{\mathcal{Y}}$ has to upper-bound the true cost of each hypothesis $y \in \hat{\mathcal{Y}}$,
$$\forall y \in \hat{\mathcal{Y}}, \quad \bar{f}(\hat{\mathcal{Y}}) \geq \mathbf{w}^T \phi(\mathbf{y}, \mathbf{x}).$$
2. The bound has to be exact for every single hypothesis,
$$\forall y \in \mathcal{Y}, \quad \bar{f}(y) = \mathbf{w}^T \phi(\mathbf{y}, \mathbf{x}).$$

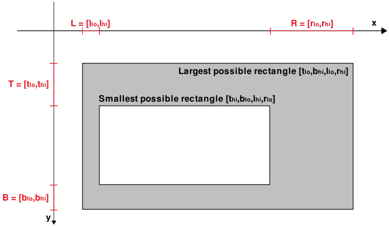
Can we define this for our problem?

Intuitions from 2D

C. H. Lampert, M. B. Blaschko, T. Hofmann: Efficient Subwindow Search: A Branch and Bound Framework for Object Localization. *IEEE T-PAMI*, 31(12):2129-2142, 2009
Code: http://www.robots.ox.ac.uk/~blaschko/software/ESS-1_2.zip

Let's look at the 2D case again

- We want to compute the bounding box that maximizes a scoring function
- Let's try to do this with branch and bound
- We define an interval as the max and min of the x and y axis of the rectangle

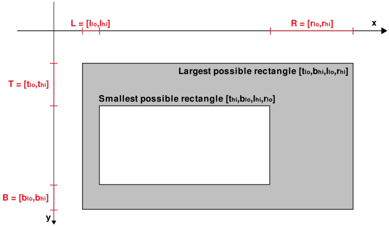


Intuitions from 2D

C. H. Lampert, M. B. Blaschko, T. Hofmann: Efficient Subwindow Search: A Branch and Bound Framework for Object Localization. *IEEE T-PAMI*, 31(12):2129-2142, 2009
Code: http://www.robots.ox.ac.uk/~blaschko/software/ESS-1_2.zip

Let's look at the 2D case again

- We want to compute the bounding box that maximizes a scoring function
- Let's try to do this with branch and bound
- We define an interval as the max and min of the x and y axis of the rectangle



- The scoring function sums features in the rectangle defined by the BBox

$$E(y_1, \cdots, y_4) = \sum_{i \in BBox(\mathbf{y})} f_i(\mathbf{x})$$

Branch and Bound for BBox prediction [Lampert & Blaschko, 2009]

- The scoring function sums features in the rectangle defined by the BBox

$$E(y_1, \cdots, y_4) = \sum_{i \in BBox(\mathbf{y})} f_i(\mathbf{x})$$

Branch and Bound for BBox prediction [Lampert & Blaschko, 2009]

- The scoring function sums features in the rectangle defined by the BBox

$$E(y_1, \cdots, y_4) = \sum_{i \in BBox(\mathbf{y})} f_i(\mathbf{x})$$

- Some features are positive and some are negative

Branch and Bound for BBox prediction [Lampert & Blaschko, 2009]

- The scoring function sums features in the rectangle defined by the BBox

$$E(y_1, \cdots, y_4) = \sum_{i \in BBox(\mathbf{y})} f_i(\mathbf{x})$$

- Some features are positive and some are negative
- **Trick**: Divide the space into negative and positive features

$$E(y_1, \cdots, y_4) = \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^+(\mathbf{x})}_{f^+(\mathbf{y},\mathbf{x})} + \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^-(\mathbf{x})}_{f^-(\mathbf{y},\mathbf{x})}$$

Branch and Bound for BBox prediction [Lampert & Blaschko, 2009]

- The scoring function sums features in the rectangle defined by the BBox

$$E(y_1, \cdots, y_4) = \sum_{i \in BBox(\mathbf{y})} f_i(\mathbf{x})$$

- Some features are positive and some are negative
- **Trick**: Divide the space into negative and positive features

$$E(y_1, \cdots, y_4) = \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^+(\mathbf{x})}_{f^+(\mathbf{y},\mathbf{x})} + \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^-(\mathbf{x})}_{f^-(\mathbf{y},\mathbf{x})}$$

- Bound the positive and negative independently

$$bound(E(\bar{\mathcal{Y}})) = \bar{f}^+(\bar{\mathcal{Y}}, \mathbf{x}) + \bar{f}^-(\bar{\mathcal{Y}}, \mathbf{x})$$

- Energy was defined as

$$E(y_1, \cdots, y_4) = \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^+(\mathbf{x})}_{f^+(\mathbf{y},\mathbf{x})} + \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^-(\mathbf{x})}_{f^-(\mathbf{y},\mathbf{x})}$$

- Bound the positive and negative independently

$$bound(E(\bar{\mathcal{Y}})) = \bar{f}^+(\bar{\mathcal{Y}}, \mathbf{x}) + \bar{f}^-(\bar{\mathcal{Y}}, \mathbf{x})$$

Bounding the functions [Lampert & Blaschko, 2009]

- Energy was defined as

$$E(y_1, \cdots, y_4) = \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^+(\mathbf{x})}_{f^+(\mathbf{y},\mathbf{x})} + \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^-(\mathbf{x})}_{f^-(\mathbf{y},\mathbf{x})}$$

- Bound the positive and negative independently

$$bound(E(\bar{\mathcal{Y}})) = \bar{f}^+(\bar{\mathcal{Y}}, \mathbf{x}) + \bar{f}^-(\bar{\mathcal{Y}}, \mathbf{x})$$

- These bounds are very simple? What are they?

Bounding the functions [Lampert & Blaschko, 2009]

- Energy was defined as

$$E(y_1, \cdots, y_4) = \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^+(\mathbf{x})}_{f^+(\mathbf{y},\mathbf{x})} + \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^-(\mathbf{x})}_{f^-(\mathbf{y},\mathbf{x})}$$

- Bound the positive and negative independently

$$bound(E(\bar{\mathcal{Y}})) = \bar{f}^+(\bar{\mathcal{Y}}, \mathbf{x}) + \bar{f}^-(\bar{\mathcal{Y}}, \mathbf{x})$$

- These bounds are very simple? What are they?
- How can we compute them very fast?

Bounding the functions

[Lampert & Blaschko, 2009]

- Energy was defined as

$$E(y_1, \cdots, y_4) = \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^+(\mathbf{x})}_{f^+(\mathbf{y},\mathbf{x})} + \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^-(\mathbf{x})}_{f^-(\mathbf{y},\mathbf{x})}$$

- Bound the positive and negative independently

$$bound(E(\bar{\mathcal{Y}})) = \bar{f}^+(\bar{\mathcal{Y}}, \mathbf{x}) + \bar{f}^-(\bar{\mathcal{Y}}, \mathbf{x})$$

- These bounds are very simple? What are they?
- How can we compute them very fast?
- What's the complexity of computing them?

Bounding the functions

[Lampert & Blaschko, 2009]

- Energy was defined as

$$E(y_1, \cdots, y_4) = \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^+(\mathbf{x})}_{f^+(\mathbf{y},\mathbf{x})} + \underbrace{\sum_{i \in BBox(\mathbf{y})} f_i^-(\mathbf{x})}_{f^-(\mathbf{y},\mathbf{x})}$$

- Bound the positive and negative independently

$$bound(E(\bar{\mathcal{Y}})) = \bar{f}^+(\bar{\mathcal{Y}}, \mathbf{x}) + \bar{f}^-(\bar{\mathcal{Y}}, \mathbf{x})$$

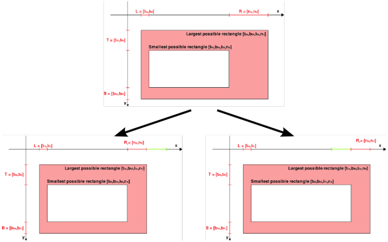
- These bounds are very simple? What are they?
- How can we compute them very fast?
- What's the complexity of computing them?
- How many integral images do we need?

Algorithm 1 Efficient Subwindow Search

Require: image x
Require: quality bounding function $\hat{f}$ (see Sect.III)
Ensure: $(t_{\text{opt}}, b_{\text{opt}}, l_{\text{opt}}, r_{\text{opt}}) = \operatorname{argmax}_{y \in \mathcal{Y}} f(y)$

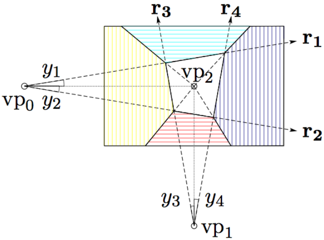
  initialize P as empty priority queue
  set $[T, B, L, R] = [1, n] \times [1, n] \times [1, m] \times [1, m]$
  repeat
    split $[T, B, L, R] \rightarrow [T_1, B_1, L_1, R_1] \,\dot{\cup}\, [T_2, B_2, L_2, R_2]$
    push $\big([T_1, B_1, L_1, R_1];\ \hat{f}([T_1, B_1, L_1, R_1]) \big)$ onto P
    push $\big([T_2, B_2, L_2, R_2];\ \hat{f}([T_2, B_2, L_2, R_2]) \big)$ onto P
    retrieve top state $[T, B, L, R]$ from P
  until $[T, B, L, R]$ consists of only one rectangle
  set $(t_{\text{opt}}, b_{\text{opt}}, l_{\text{opt}}, r_{\text{opt}}) = [T, B, L, R]$

- How do we split?



- When do we terminate?

- Let's go back to our problem



- We parameterize the sets by **intervals** of minimum and maximum angles

$$\{[y_1^{min}, y_1^{max}], \cdots, [y_4^{min}, y_4^{max}]\}$$

- The scoring function sums features over the faces

$$E(y_1, \cdots, y_4) = \sum_r \mathbf{w}_r^T \phi(\mathbf{y}_r, \mathbf{x}) = \sum_\alpha f_\alpha(\mathbf{y}, \mathbf{x})$$

with $\alpha = \{floor, left\_w, right\_w, ceiling, front\_w\}$

- What about the bounds?

Bounds for 3D layout

[Schwing & Urtasun, 2012]

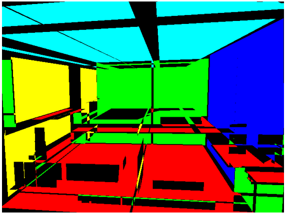
- The scoring function sums features over the faces

$$E(y_1, \cdots, y_4) = \sum_r \mathbf{w}_r^T \phi(\mathbf{y}_r, \mathbf{x}) = \sum_\alpha f_\alpha(\mathbf{y}, \mathbf{x})$$

with $\alpha = \{floor, left\_w, right\_w, ceiling, front\_w\}$

- Let's bound each "face" α separately
- Recall where the features come from

original image



orientation map

geometric context

- Some features are positive, some are negative. Why? How do I know which ones are positive/negative?

- Inference can be then done by

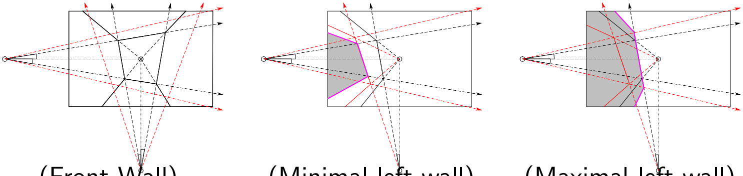
$$E(y_1, \cdots, y_4) = \sum_{\alpha} f_{\alpha}^{+}(x, y) + f_{\alpha}^{-}(x, y),$$

- We can bound each of this terms separately

$$bound(E(\hat{\mathcal{Y}}, \mathbf{x})) = \sum_{\alpha \in \mathcal{F}} \bar{f}_{\alpha}^{+}(\hat{\mathcal{Y}}, \mathbf{x}) + \bar{f}_{\alpha}^{-}(\hat{\mathcal{Y}}, \mathbf{x})$$

- We construct bounds by computing the max positive and min negative contribution of the score within the set $\hat{\mathcal{Y}}$ for each face $\alpha \in \mathcal{F}$.

$$\bar{f}_{\text{front-wall}}(\hat{\mathcal{Y}}) = f_{\text{front-wall}}^{+}(x, y_{up}) + f_{\text{front-wall}}^{-}(x, y_{low}),$$



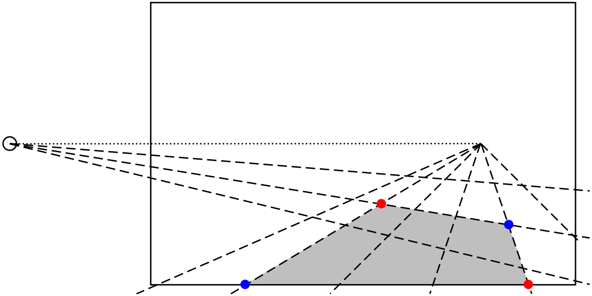
(Front Wall) (Minimal left wall) (Maximal left wall)

Efficient bounds [Schwing & Urtasun, 2012]

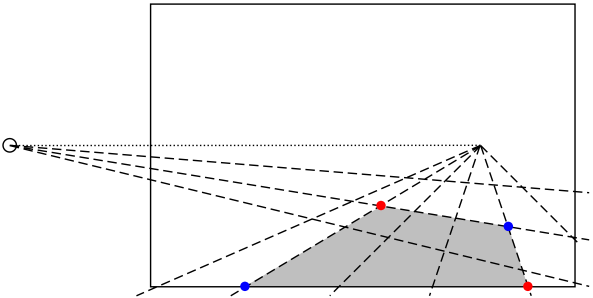
- How can we compute the bounds efficiently?

Efficient bounds

- How can we compute the bounds efficiently?



- How can we compute the bounds efficiently?

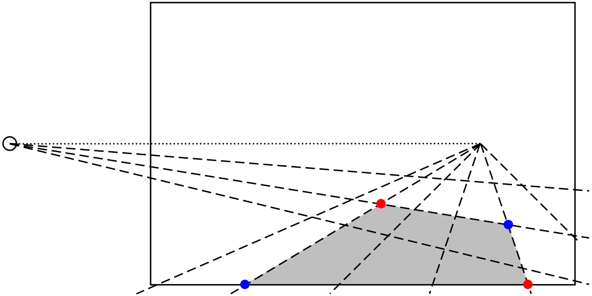


- What's the complexity?

Efficient bounds

[Schwing & Urtasun, 2012]

- How can we compute the bounds efficiently?

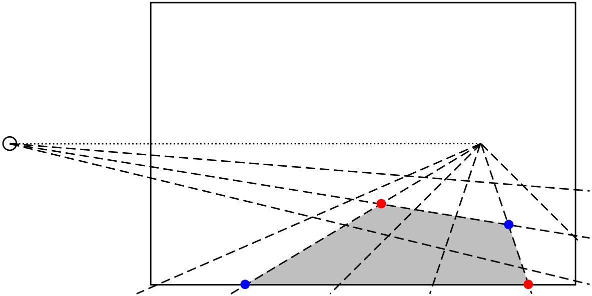


- What's the complexity?
- How many evaluations?

Efficient bounds

[Schwing & Urtasun, 2012]

- How can we compute the bounds efficiently?



- What's the complexity?
- How many evaluations?
- Learning uses Structured SVMs, trains in 1min!

Results on Layout Dataset

	OM	GC	OM/GC	Other	GC/Oth	OM/Oth	Time
[Hoiem07]	-	28.9	-	-	-	-	-
[Hedau09] (a)	-	26.5	-	-	-	-	-
[Hedau09] (b)	-	-	-	-	21.2	-	10-30 min
[Lee10] w/o	24.7	22.7	18.6	-	-	-	-
[delPero11]	-	-	-	26.8	-	-	10s?
[delPero12]	-	-	-	24.7	-	21.3	X min
Schwing12a	**18.6**	**15.4**	**13.6**	-	-	-	0.15s
Schwing12b	**18.6**	**15.4**	**13.6**	-	-	-	**0.007s**

Table : Pixel classification error in the layout dataset of (Hedau et al. 09).

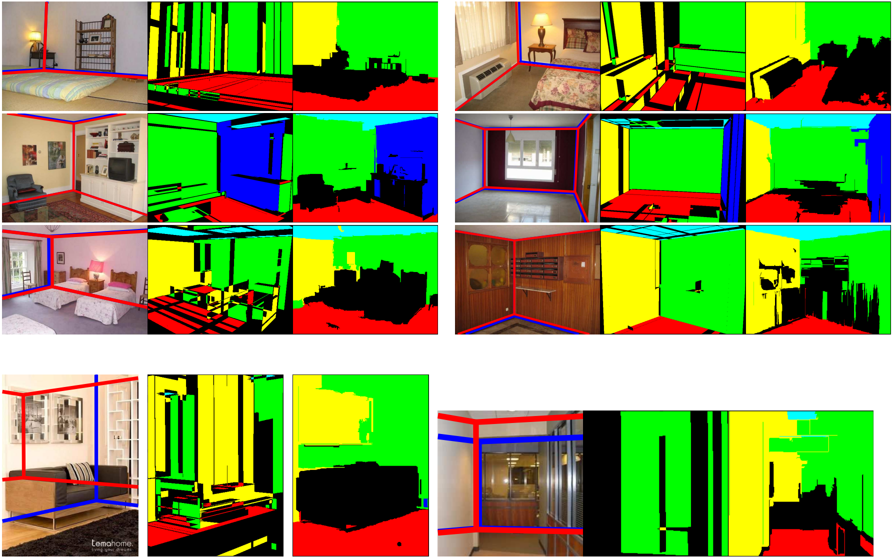
	[delPero11]	[Hoiem07]	[Hedau09](a)	Schwing12b
w/o box	29.59	23.04	22.94	**16.46**

Table : Pixel classification error in the bedroom data set [Hedau et al. 10].

- Takes on average 0.007s for exact solution over 50^4 possibilities !
- It's 6 orders of magnitude faster!

Qualitative Results

[Schwing & Urtasun, 2012]



But rooms are not empty, what about the objects?

Joint inference over layout and 3D objects

Objects as Clutter

[Wang et al., 2010]

H. Wang, S. Gould, D. Koller (2010), Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding, *ECCV*, 2010

- (Wang et al. 10) formulate the problem as inference of the room (4 rays) and clutter
- Clutter as a latent variable $\rightarrow$ no need for annotations of clutter

H. Wang, S. Gould, D. Koller (2010), Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding, *ECCV*, 2010

- (Wang et al. 10) formulate the problem as inference of the room (4 rays) and clutter
- Clutter as a latent variable $\rightarrow$ no need for annotations of clutter
- Let $\mathbf{x}$ image, $\mathbf{y}$ the layout and h the clutter, the enegy

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}, \mathbf{h}) - (\alpha E^a(\mathbf{x}, \mathbf{y}, \mathbf{h}) + \beta E^c(\mathbf{y}, \mathbf{h}))$$

Objects as Clutter

H. Wang, S. Gould, D. Koller (2010), Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding, *ECCV*, 2010

- (Wang et al. 10) formulate the problem as inference of the room (4 rays) and clutter
- Clutter as a latent variable $\rightarrow$ no need for annotations of clutter
- Let **x** image, **y** the layout and h the clutter, the enegy

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}, \mathbf{h}) - (\alpha E^a(\mathbf{x}, \mathbf{y}, \mathbf{h}) + \beta E^c(\mathbf{y}, \mathbf{h}))$$

- Ψ contains a rich set of features: color, texture, perspective consistency, and overall layout

Objects as Clutter

[Wang et al., 2010]

H. Wang, S. Gould, D. Koller (2010), Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding, *ECCV*, 2010

- (Wang et al. 10) formulate the problem as inference of the room (4 rays) and clutter
- Clutter as a latent variable $\rightarrow$ no need for annotations of clutter
- Let $\mathbf{x}$ image, $\mathbf{y}$ the layout and h the clutter, the enegy

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}, \mathbf{h}) - (\alpha E^a(\mathbf{x}, \mathbf{y}, \mathbf{h}) + \beta E^c(\mathbf{y}, \mathbf{h}))$$

- Ψ contains a rich set of features: color, texture, perspective consistency, and overall layout
- E^a is the variance of the appearance value within a layout face excluding clutter

H. Wang, S. Gould, D. Koller (2010), Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding, *ECCV*, 2010

- (Wang et al. 10) formulate the problem as inference of the room (4 rays) and clutter
- Clutter as a latent variable $\rightarrow$ no need for annotations of clutter
- Let **x** image, **y** the layout and h the clutter, the enegy

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}, \mathbf{h}) - (\alpha E^a(\mathbf{x}, \mathbf{y}, \mathbf{h}) + \beta E^c(\mathbf{y}, \mathbf{h}))$$

- Ψ contains a rich set of features: color, texture, perspective consistency, and overall layout
- E^a is the variance of the appearance value within a layout face excluding clutter
- E^c penalizes clutterness of each face

H. Wang, S. Gould, D. Koller (2010), Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding, *ECCV*, 2010

- (Wang et al. 10) formulate the problem as inference of the room (4 rays) and clutter
- Clutter as a latent variable $\rightarrow$ no need for annotations of clutter
- Let $\mathbf{x}$ image, $\mathbf{y}$ the layout and h the clutter, the enegy

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}, \mathbf{h}) - (\alpha E^a(\mathbf{x}, \mathbf{y}, \mathbf{h}) + \beta E^c(\mathbf{y}, \mathbf{h}))$$

- Ψ contains a rich set of features: color, texture, perspective consistency, and overall layout
- E^a is the variance of the appearance value within a layout face excluding clutter
- E^c penalizes clutterness of each face
- **Learning:** latent structured SVM

Objects as Clutter

[Wang et al., 2010]

H. Wang, S. Gould, D. Koller (2010), Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding, *ECCV*, 2010

- (Wang et al. 10) formulate the problem as inference of the room (4 rays) and clutter
- Clutter as a latent variable $\rightarrow$ no need for annotations of clutter
- Let $\mathbf{x}$ image, $\mathbf{y}$ the layout and h the clutter, the enegy

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}, \mathbf{h}) - (\alpha E^a(\mathbf{x}, \mathbf{y}, \mathbf{h}) + \beta E^c(\mathbf{y}, \mathbf{h}))$$

- Ψ contains a rich set of features: color, texture, perspective consistency, and overall layout
- E^a is the variance of the appearance value within a layout face excluding clutter
- E^c penalizes clutterness of each face
- **Learning:** latent structured SVM
- **Inference**: Alternate optimization scheme with local search

Results

[Wang et al., 2010]

Results on Layout Dataset

	OM	GC	OM/GC	Other	GC/Oth	OM/Oth	Time
[Hoiem07]	-	28.9	-	-	-	-	-
[Hedau09](a)	-	26.5	-	-	-	-	-
[Hedau09](b)	-	-	-	-	21.2	-	10-30 min
[Wang10]	-	-	-	22.2	-	-	-
[Lee10] w/o	24.7	22.7	18.6	-	-	-	-
[delPero11]	-	-	-	26.8	-	-	10s?
[delPero12]	-	-	-	24.7	-	21.3	X min
Schwing12a	18.6	15.4	**13.6**	-	-	-	0.15s
Schwing12b	18.6	15.4	**13.6**	-	-	-	**0.007s**

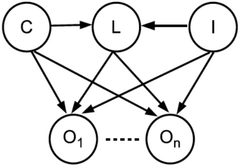
Table : Pixel classification error in the layout dataset of (Hedau et al. 09).

Rescoring Candidates

[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

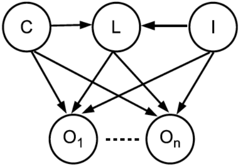
- Model Interactions between a small set of layout hypothesis (i.e., 100), camera and objects



$$p(o_1, \cdots, o_N, L, C) = p(C)p(L|C)\prod_i p(o_i|L, C)$$

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

- Model Interactions between a small set of layout hypothesis (i.e., 100), camera and objects

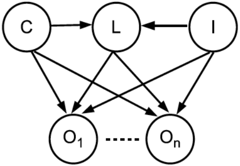


$$p(o_1, \cdots, o_N, L, C) = p(C)p(L|C)\prod_i p(o_i|L, C)$$

- What's the non-reasonable assumption?

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

- Model Interactions between a small set of layout hypothesis (i.e., 100), camera and objects



$$p(o_1, \cdots, o_N, L, C) = p(C)p(L|C)\prod_i p(o_i|L, C)$$

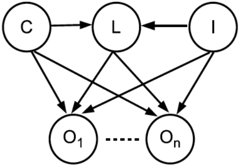
- What's the non-reasonable assumption?
- Potentials: overlap between object's footprint and the floor, distance between object and the walls, scores from our object detector, inferred object height

Rescoring Candidates

[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

- Model Interactions between a small set of layout hypothesis (i.e., 100), camera and objects



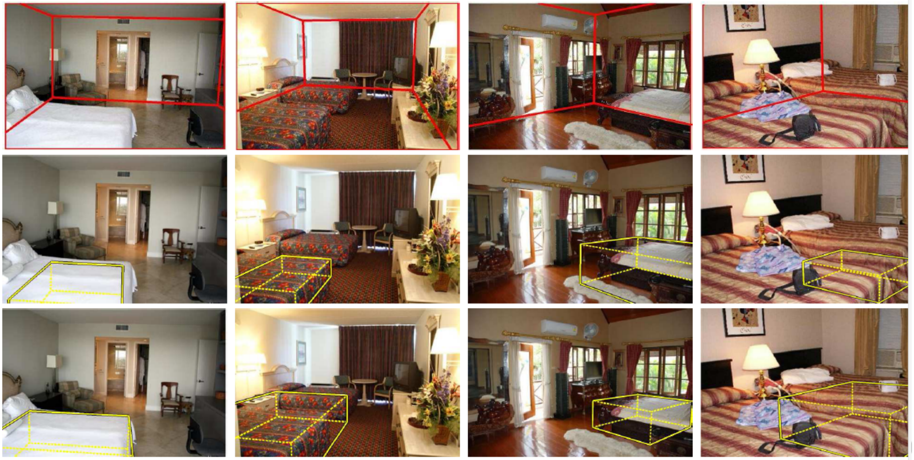
$$p(o_1, \cdots, o_N, L, C) = p(C)p(L|C)\prod_i p(o_i|L, C)$$

- What's the non-reasonable assumption?
- Potentials: overlap between object's footprint and the floor, distance between object and the walls, scores from our object detector, inferred object height
- Due to the assumptions of the approach, inference is very easy

Results

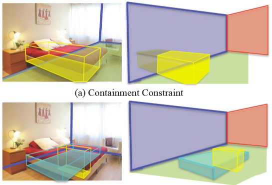
[Hedau et al., 2010]

	3D Cuboid	DPM	Both	Both + layout
AP	0.513	0.542	0.596	0.628



D. C. Lee, A. Gupta, M. Hebert, T. Kanade, Estimating Spatial Layout of Rooms using Volumetric Reasoning about Objects and Surfaces, *NIPS* 2010
Code: https://www.cs.cmu.edu/~dclee/code/index.html

- Jointly extract the spatial layout of the room and the configuration of objects in the scene.
- Objects parameterized as 3D cuboids which occupy 3D volumes in the free space defined by the room walls
- Select configuration that best matches local surface geometry estimated via image cues and satisfies the volumetric constraints of the physical world
  - Each object has non-zero finite volume
  - The objects cannot intersect
  - The objects are inside the room

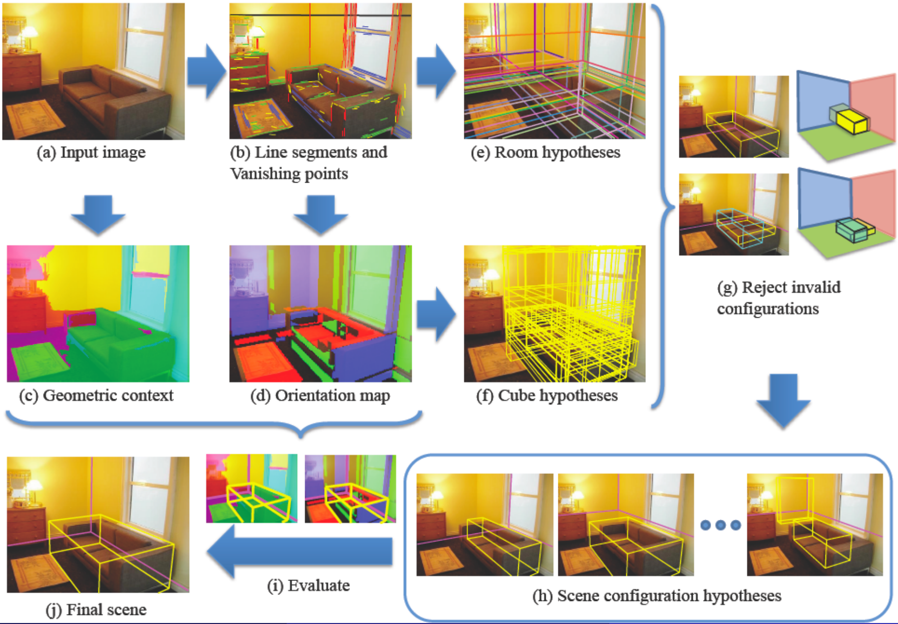


(a) Containment Constraint

(b) Spatial Exclusion Constraint

Model Overview

[Lee et al., 2010]



Details and Results

[Lee et al., 2010]

- Learning via Structured SVMs
- Loss function: percentage of pixels in the entire image having
- Inference via Beam Search incorrect label

Input image Orientation map Geometric context Room only Room and objects

Results on Layout Dataset

	OM	GC	OM/GC	Other	GC/Oth	OM/Oth	Time
[Hoiem07]	-	28.9	-	-	-	-	-
[Hedau09](a)	-	26.5	-	-	-	-	-
[Hedau09](b)	-	-	-	-	21.2	-	10-30 min
[Wang10]	-	-	-	22.2	-	-	-
[Lee10] w/o	24.7	22.7	18.6	-	-	-	-
[Lee10] + o	19.5	20.2	16.2	-	-	-	-
[delPero11]	-	-	-	26.8	-	-	10s?
[delPero12]	-	-	-	24.7	-	21.3	X min
Schwing12a	18.6	15.4	**13.6**	-	-	-	0.15s
Schwing12b	18.6	15.4	**13.6**	-	-	-	**0.007s**

Table : Pixel classification error in the layout dataset of (Hedau et al. 09).

L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, K. Barnard, Bayesian geometric modeling of indoor scenes, *CVPR* 2012

- Generative Model
$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$
- Room is represented
$$r_b = (x_r, y_b, z_b, w_b, h_b, l_b, \gamma)$$
with (x_r, y_b, z_b) the coordinates of the room centre in 3D, (w_b, h_b, l_b) are the with, height and length and γ is the angle of rotation
- Intrinsics: no skew and unity aspect ratio, and principal point in the center.
- Camera model is fully specify with
$$c = (\psi, \phi, f)$$
with ψ, ϕ the pitch an roll angles and f the focal length

More Powerful Generative Models [del Pero et al., 2012]

L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, K. Barnard, Bayesian geometric modeling of indoor scenes, *CVPR* 2012

- Generative Model

$$\underbrace{p(\theta|E)}_{posterior} \propto \underbrace{p(E|\theta)}_{likelihood} \underbrace{p(\theta)}_{prior}$$

- Room is represented

$$r_b = (x_r, y_b, z_b, w_b, h_b, l_b, \gamma)$$

  with (x_r, y_b, z_b) the coordinates of the room centre in 3D, (w_b, h_b, l_b) are the with, height and length and γ is the angle of rotation
- Intrinsics: no skew and unity aspect ratio, and principal point in the center.
- Camera model is fully specify with

$$c = (\psi, \phi, f)$$

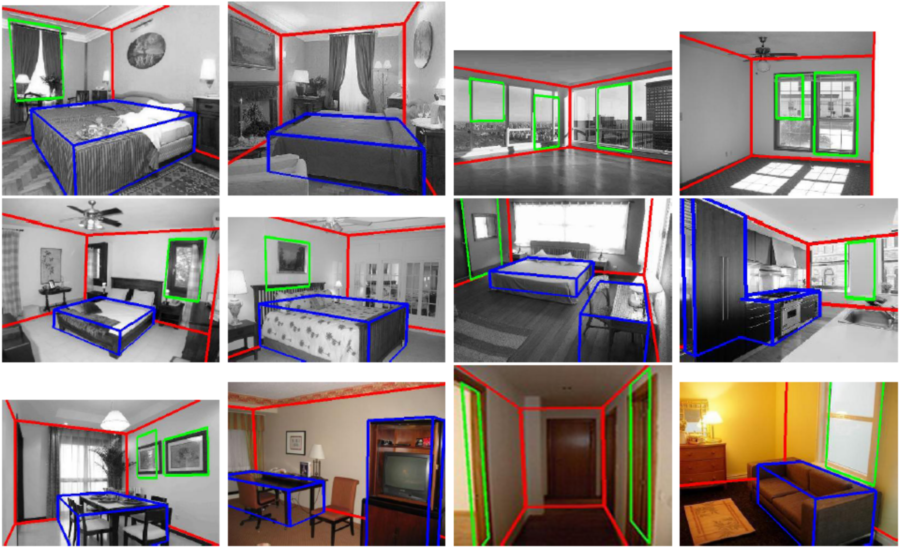
  with ψ, ϕ the pitch an roll angles and f the focal length
- Add objects $(o_1, o_2, \cdots)$, where the object

$$o_i = (b_i, t_i)$$

  with b_i the bounding box and t_i the type of object

Complex and Slow Inference [del Pero et al., 2012]

- Likelihood uses lines and GCs
- Inference via Sampling
- **Diffusion moves**: sample parameters
- **Jump Moves**: change the structure of the model by adding and removing objects.
- Need to use Reversible Jumps $\rightarrow$ complicated!



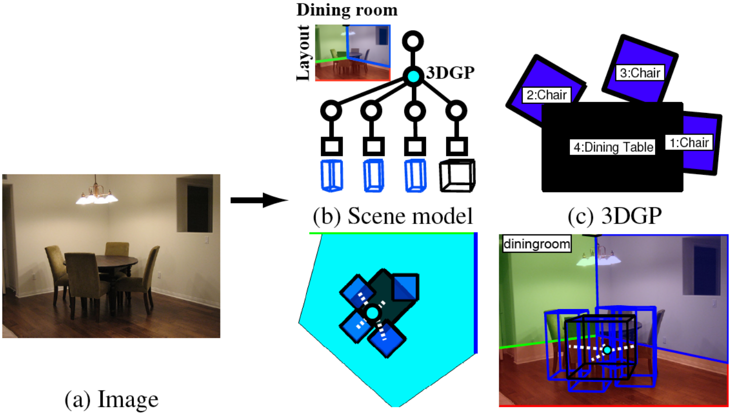
Results on Layout Dataset

	OM	GC	OM/GC	Other	GC/Oth	OM/Oth	Time
[Hoiem07]	-	28.9	-	-	-	-	-
[Hedau09](a)	-	26.5	-	-	-	-	-
[Hedau09](b)	-	-	-	-	21.2	-	10-30 min
[Wang10]	-	-	-	22.2	-	-	-
[Lee10] w/o	24.7	22.7	18.6	-	-	-	-
[Lee10] + o	19.5	20.2	16.2	-	-	-	-
[delPero11]	-	-	-	26.8	-	-	10s?
[delPero12]	-	-	-	24.7	-	21.3	X min
[delPero12]+o	-	-	-	-	-	16.3	12 min
Schwing12a	18.6	15.4	**13.6**	-	-	-	0.15s
Schwing12b	18.6	15.4	**13.6**	-	-	-	**0.007s**

Table : Pixel classification error in the layout dataset of (Hedau et al. 09).

W. Choi, Y. -W. Chao, C. Pantofaru, S. Savarese. Understanding Indoor Scenes Using 3D Geometric Phrases, *CVPR*, 2013
Code and data: http://wwweb.eecs.umich.edu/vision/3DGP/

- Learn the typical configuration of objects in 3D
- Solve jointly for scene type, layout and objects



(a) Image (b) Scene model (c) 3DGP (d) 3D model (e) Final labeling

Energy Formulation

[Choi et al., 2013]

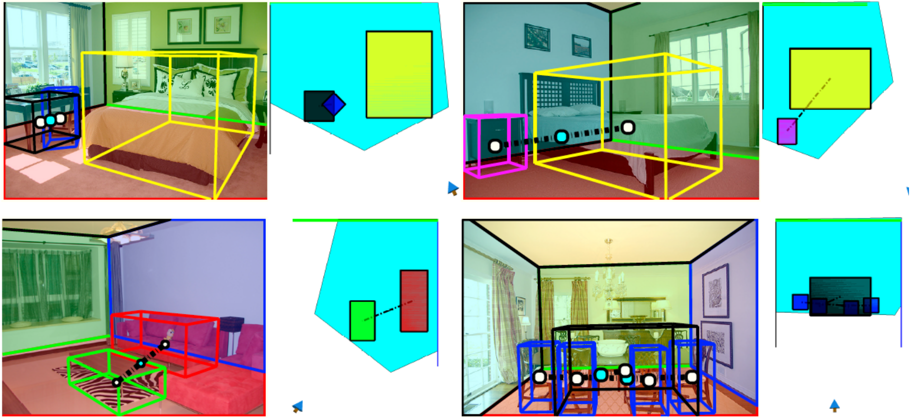
- The energy is defined

$$
\begin{aligned}
E_{\Pi,\theta}(G,I) = {} & \underbrace{\alpha^\top \phi(C,O_s)}_{\text{scene observation}} + \underbrace{\beta^\top \phi(H,O_l)}_{\text{layout observation}} + \underbrace{\sum_{V\in\mathbb{V}_T} \gamma^\top \phi(V,O_o)}_{\text{object observation}} \\
& + \underbrace{\sum_{V\in\mathbb{V}_T} \eta^\top \psi(V,C)}_{\text{object-scene}} + \underbrace{\sum_{V\in\mathbb{V}_T} \nu^\top \psi(V,H)}_{\text{object-layout}} \\
& + \underbrace{\sum_{V,W\in\mathbb{V}_T} \mu^\top \varphi(V,W)}_{\text{object overlap}} + \underbrace{\sum_{V\in\mathbb{V}_I} \lambda^\top \varphi(V,Ch(V))}_{\text{3DGP}} \qquad (1)
\end{aligned}
$$

- Learning by "clustering" and fitting parameters with max-margin
- Inference via Reversible Jump MCMC

Positive Results

[Choi et al., 2013]



Results on Layout Dataset

	OM	GC	OM/GC	Other	GC/Oth	OM/Oth	Time
[Hoiem07]	-	28.9	-	-	-	-	-
[Hedau09](a)	-	26.5	-	-	-	-	-
[Hedau09](b)	-	-	-	-	21.2	-	10-30 min
[Wang10]	-	-	-	22.2	-	-	-
[Lee10] w/o	24.7	22.7	18.6	-	-	-	-
[Lee10] + o	19.5	20.2	16.2	-	-	-	-
[delPero11]	-	-	-	26.8	-	-	10s?
[delPero12]	-	-	-	24.7	-	21.3	X min
[delPero12]+o	-	-	-	-	-	16.3	12 min
Schwing12a	18.6	15.4	**13.6**	-	-	-	0.15s
Schwing12b	18.6	15.4	**13.6**	-	-	-	**0.007s**

Table : Pixel classification error in the layout dataset of (Hedau et al. 09).

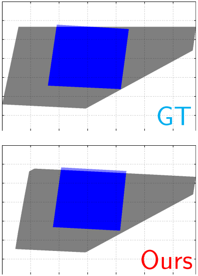
- They didn't evaluate on this dataset but in their own data, performance for layout is 1% better than Hedau09

Optimal solution to the joint layout and object problem?

3D Scene Understanding from Single Image

A. Schwing, S. Fidler, M. Pollefeys, R. Urtasun, Box In the Box: Joint 3D Layout and Object Reasoning from Single Images, *ICCV*, 2013

- **Task:** Given a single image, obtain the layout as well as the 3D objects present in the scene

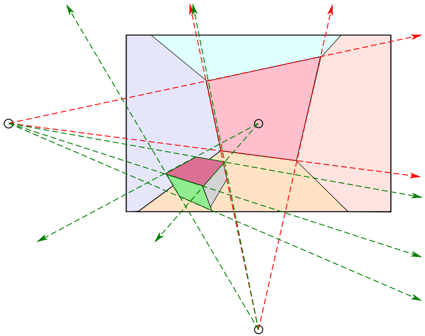


- **Assumption:** The world is Manhattan, objects and room are 3D cuboids oriented in accordance with the vanishing points (VPs)
- **Conjecture:** A holistic approach that does joint inference over layout and objects should be better than serial reasoning

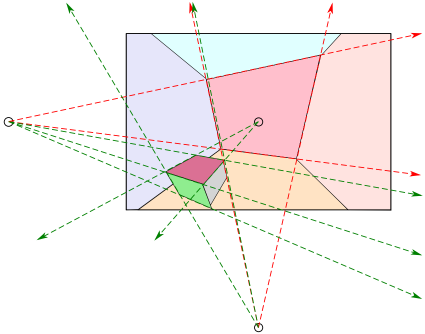
Parameterization

[Schwing et al., 2013]

- Given the VPs, we need 4 angles to describe the room layout and 5 angles to describe each object
- For simplicity let's consider a single object

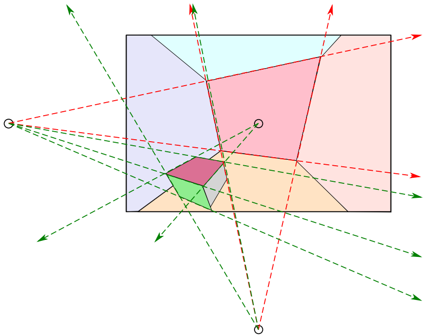


- Given the VPs, we need 4 angles to describe the room layout and 5 angles to describe each object
- For simplicity let's consider a single object



- Let $\mathbf{y}$ be the layout and $\mathbf{z}$ the object

- Given the VPs, we need 4 angles to describe the room layout and 5 angles to describe each object
- For simplicity let's consider a single object

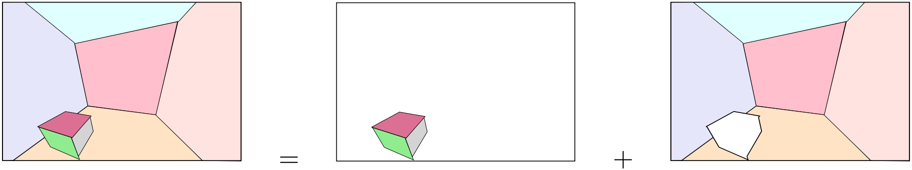


- Let $\mathbf{y}$ be the layout and $\mathbf{z}$ the object
- Branch and bound for exact inference

Scoring Function Over Joint Problem [Schwing et al., 2013]

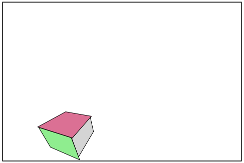
- Combined energy is

$$E_{total}(x, \mathbf{y}, \mathbf{z}) = E_{object}(x, \mathbf{z}) + E_{layout}(x, \mathbf{y}, \mathbf{z})$$

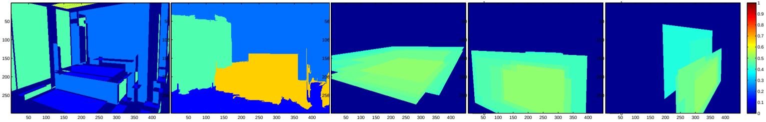


$E_{total}(x, \mathbf{y}, \mathbf{z})$ $=$ $E_{object}(x, \mathbf{z})$ $+$ $E_{layout}(x, \mathbf{y}, \mathbf{z})$

- Log linear model $E_{object}(x, \mathbf{z}) = \mathbf{w}^T \phi_{object}(x, \mathbf{z})$,



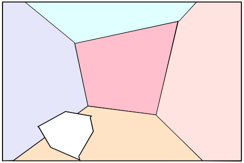
- Count for each face of the object **geometric features** (i.e., normal direction), as well as probability map generated by a 3D **detector**



(OM) (GC) (3D detection)

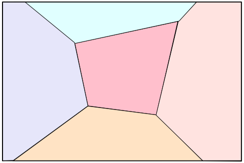
- Take into account occlusion to not over-count evidence

$$E_{layout}(x, \mathbf{y}, \mathbf{z}) = E_{full-layout}(x, \mathbf{y}) - E_{occ}(x, \mathbf{y}, \mathbf{z}) + E_{pen}(x, \mathbf{y}, \mathbf{z})$$



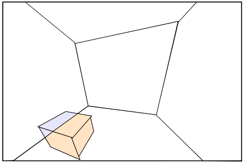
$E_{layout}(x, \mathbf{y}, \mathbf{z})$

=



$E_{full\_layout}(x, \mathbf{y})$

-

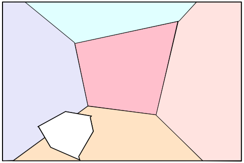


$E_{occ}(x, \mathbf{y}, \mathbf{z})$

Layout Scoring Function with Occlusion

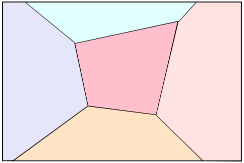
- Take into account occlusion to not over-count evidence

$$E_{layout}(x, \mathbf{y}, \mathbf{z}) = E_{full-layout}(x, \mathbf{y}) - E_{occ}(x, \mathbf{y}, \mathbf{z}) + E_{pen}(x, \mathbf{y}, \mathbf{z})$$



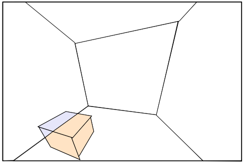
$E_{layout}(x, \mathbf{y}, \mathbf{z})$

=



$E_{full\_layout}(x, \mathbf{y})$

-



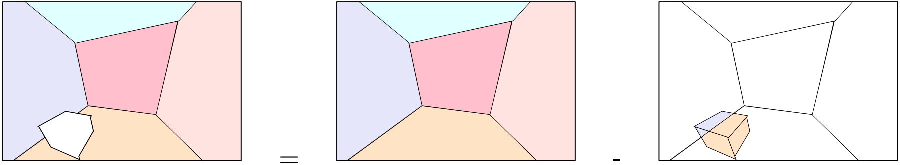
$E_{occ}(x, \mathbf{y}, \mathbf{z})$

- We have seen how to compute $E_{full-layout}(x, \mathbf{y})$ before

Layout Scoring Function with Occlusion [Schwing et al., 2013]

- Take into account occlusion to not over-count evidence

$$E_{layout}(x, \mathbf{y}, \mathbf{z}) = E_{full-layout}(x, \mathbf{y}) - E_{occ}(x, \mathbf{y}, \mathbf{z}) + E_{pen}(x, \mathbf{y}, \mathbf{z})$$



$E_{layout}(x, \mathbf{y}, \mathbf{z})$ = $E_{full\_layout}(x, \mathbf{y})$ - $E_{occ}(x, \mathbf{y}, \mathbf{z})$

- We have seen how to compute $E_{full-layout}(x, \mathbf{y})$ before
- $E_{pen}(x, \mathbf{y}, \mathbf{z})$ ensures that the object does not penetrate the walls

$$E = E_{object}(x, \mathbf{z}) + \underbrace{E_{full-layout}(x, \mathbf{y}) - E_{occ}(x, \mathbf{y}, \mathbf{z}) + E_{pen}(x, \mathbf{y}, \mathbf{z})}_{E_{layout}(x,\mathbf{y},\mathbf{z})}$$

- E_{occ} subtracts the object from the layout for the OM and GC features

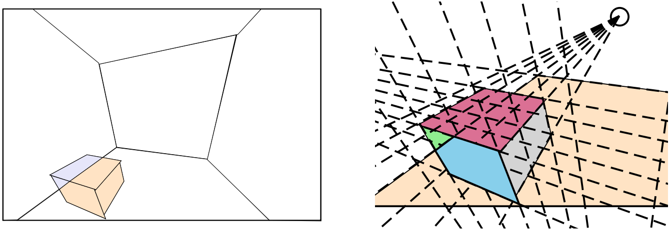


Figure : Example of how the front face of the object affects the floor estimation of the layout

$$E = E_{object}(x, \mathbf{z}) + \underbrace{E_{full-layout}(x, \mathbf{y}) - E_{occ}(x, \mathbf{y}, \mathbf{z}) + E_{pen}(x, \mathbf{y}, \mathbf{z})}_{E_{layout}(x, \mathbf{y}, \mathbf{z})}$$

- E_{occ} subtracts the object from the layout for the OM and GC features

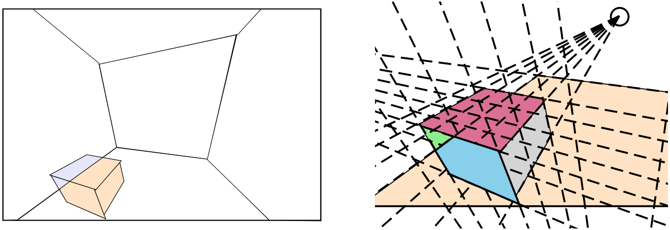


Figure : Example of how the front face of the object affects the floor estimation of the layout

- **Difficulty**: The shape varies depending on where the object is relative to the layout

Algorithm 1 branch and bound (BB) inference

put pair $(\bar{f}(\mathcal{Y}), \mathcal{Y})$ into queue and set $\hat{\mathcal{Y}} = \mathcal{Y}$
repeat
  split $\hat{\mathcal{Y}} = \hat{\mathcal{Y}}_1 \times \hat{\mathcal{Y}}_2$ with $\hat{\mathcal{Y}}_1 \cap \hat{\mathcal{Y}}_2 = \emptyset$
  put pair $(\bar{f}(\hat{\mathcal{Y}}_1), \hat{\mathcal{Y}}_1)$ into queue
  put pair $(\bar{f}(\hat{\mathcal{Y}}_2), \hat{\mathcal{Y}}_2)$ into queue
  retrieve $\hat{\mathcal{Y}}$ having highest score
until $|\hat{\mathcal{Y}}| = 1$

We have to define:

1. A parameterization that defines **sets of hypothesis**.
2. A **scoring function**
3. **Tight bounds** on the scoring function that can be computed very **efficiently**

Branch & Bound for Exact Inference [Schwing et al., 2013]

Algorithm 1 branch and bound (BB) inference

put pair $(\bar{f}(\mathcal{Y}), \mathcal{Y})$ into queue and set $\hat{\mathcal{Y}} = \mathcal{Y}$
repeat
  split $\hat{\mathcal{Y}} = \hat{\mathcal{Y}}_1 \times \hat{\mathcal{Y}}_2$ with $\hat{\mathcal{Y}}_1 \cap \hat{\mathcal{Y}}_2 = \emptyset$
  put pair $(\bar{f}(\hat{\mathcal{Y}}_1), \hat{\mathcal{Y}}_1)$ into queue
  put pair $(\bar{f}(\hat{\mathcal{Y}}_2), \hat{\mathcal{Y}}_2)$ into queue
  retrieve $\hat{\mathcal{Y}}$ having highest score
until $|\hat{\mathcal{Y}}| = 1$

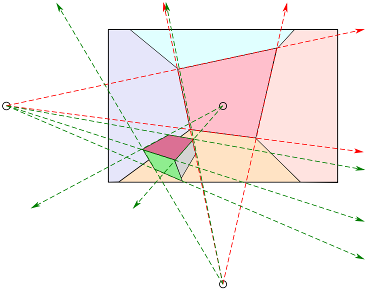
We have to define:

1. A parameterization that defines **sets of hypothesis**.
2. A **scoring function**
3. **Tight bounds** on the scoring function that can be computed very **efficiently**

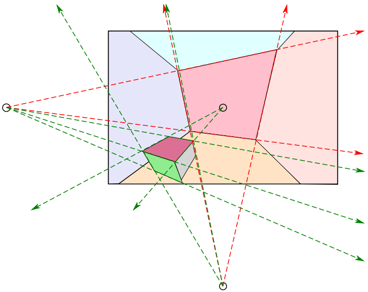
Energy is a sum of terms, we bound them individually

Parameterization of the Problem

- Param. layout with 4 variables $y_i \in \mathcal{Y}$, $i \in \{1, ..., 4\}$
- We parameterize an object with 5 variables $z_i \in \mathcal{Z}$, $i \in \{1, ..., 5\}$



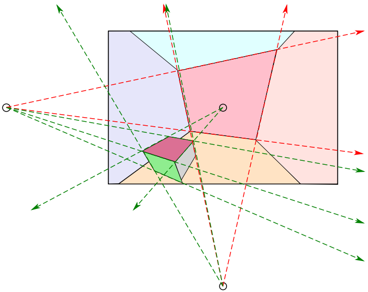
- Param. layout with 4 variables $y_i \in \mathcal{Y}$, $i \in \{1, ..., 4\}$
- We parameterize an object with 5 variables $z_i \in \mathcal{Z}$, $i \in \{1, ..., 5\}$



- We parameterize the sets by intervals of minimum and maximum angles

$$\{[y_1^{min}, y_1^{max}], \cdots, [y_4^{min}, y_4^{max}]\}$$

- Param. layout with 4 variables $y_i \in \mathcal{Y}$, $i \in \{1, ..., 4\}$
- We parameterize an object with 5 variables $z_i \in \mathcal{Z}$, $i \in \{1, ..., 5\}$



- We parameterize the sets by intervals of minimum and maximum angles

$$\{[y_1^{min}, y_1^{max}], \cdots, [y_4^{min}, y_4^{max}]\}$$

- The same thing for the object, use intervals for the angles

$$\{[z_1^{min}, z_1^{max}], \cdots, [z_5^{min}, z_5^{max}]$$

Deriving bounds for the Layout

[Schwing et al., 2013]

- Decompose potential into positive and negative contributions

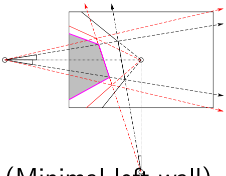
$$E_{full-layout}(x, \mathbf{y}) = {w_{fl}^{+}}^{\top} \phi_{fl}^{+}(x, \mathbf{y}) + {w_{fl}^{-}}^{\top} \phi_{fl}^{-}(x, \mathbf{y})$$

- Bound each face individually

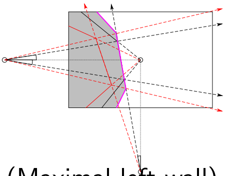
$$\bar{f}(\hat{\mathcal{Y}}) = \sum_{\alpha \in \mathcal{F}} \left(\bar{f}_{\alpha}^{+}(\hat{\mathcal{Y}}) + \bar{f}_{\alpha}^{-}(\hat{\mathcal{Y}}) \right)$$

- Bounds are max positive and min negative contributions for each face

$$\bar{f}_{\textit{left-wall}}(\hat{\mathcal{Y}}) = f_{\textit{left-wall}}^{+}(x, y_{up}) + f_{\textit{left-wall}}^{-}(x, y_{low}),$$



(Minimal left wall)



(Maximal left wall)

- Decompose potential into positive and negative contributions

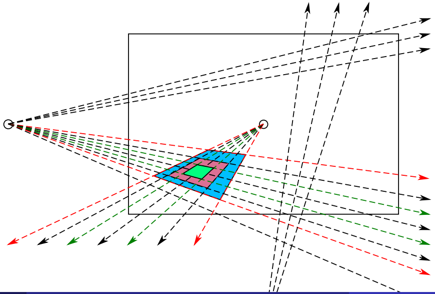
$$E_{obj}(x, \mathbf{z}) = w_{obj}^{+\top} \phi_{obj}^{+}(x, \mathbf{z}) + w_{obj}^{-\top} \phi_{obj}^{-}(x, \mathbf{z})$$

- Bound each face individually, using integral geometry

$$\bar{g}(\hat{\mathcal{Z}}) = \sum_{\alpha \in \mathcal{F}} \left(\bar{g}_{\alpha}^{+}(\hat{\mathcal{Z}}) + \bar{g}_{\alpha}^{-}(\hat{\mathcal{Z}}) \right)$$

- Bounds are max positive and min negative contributions for each face

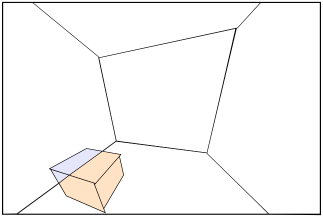
$$\bar{g}_{top\text{-}obj}(\hat{\mathcal{Z}}) = g_{top\text{-}obj}^{+}(x, z_{up}) + g_{top\text{-}obj}^{-}(x, z_{low}),$$



Bounds for Penetration and Occlusion [Schwing et al., 2013]

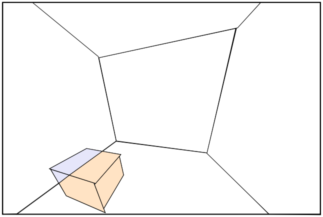
- Penetration is implicitly bounded by **carving** out the space, i.e., removing hypothesis that do not satisfy the penetration constraint

- Penetration is implicitly bounded by **carving** out the space, i.e., removing hypothesis that do not satisfy the penetration constraint
- Life gets harder with the occlusion constraint: integral geometry does not work anymore!!!



Bounds for Penetration and Occlusion [Schwing et al., 2013]

- Penetration is implicitly bounded by **carving** out the space, i.e., removing hypothesis that do not satisfy the penetration constraint
- Life gets harder with the occlusion constraint: integral geometry does not work anymore!!!

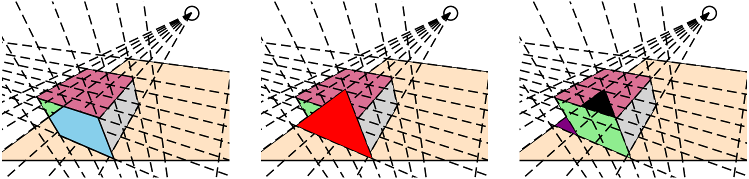


- Decompose intersections into **triangles** and compute more accumulators so that you can get constant time access

Bounds for Penetration and Occlusion [Schwing et al., 2013]

- It looks complicated and high order!

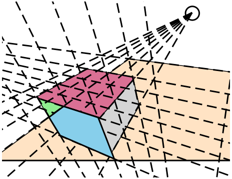
- It looks complicated and high order!
- But look at pairs of faces and decompose intersections into triangles



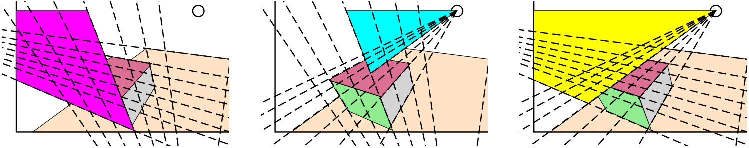
Bounds for Penetration and Occlusion

[Schwing et al., 2013]

- It looks complicated and high order!
- But look at pairs of faces and decompose intersections into triangles

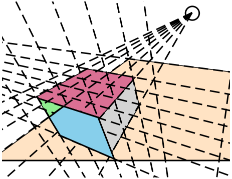


- Compute more accumulators so that you can get constant time access

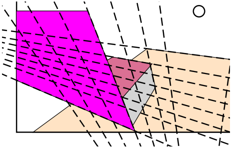


Bounds for Penetration and Occlusion [Schwing et al., 2013]

- It looks complicated and high order!
- But look at pairs of faces and decompose intersections into triangles

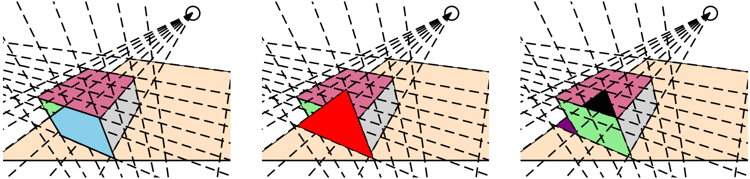


- Compute more accumulators so that you can get constant time access

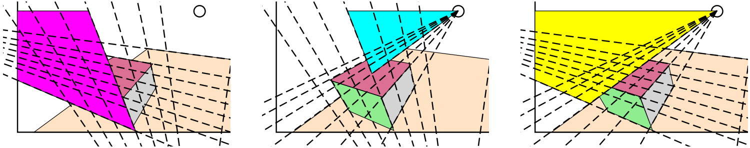


- This accumulators are also pairwise potentials!

- It looks complicated and high order!
- But look at pairs of faces and decompose intersections into triangles



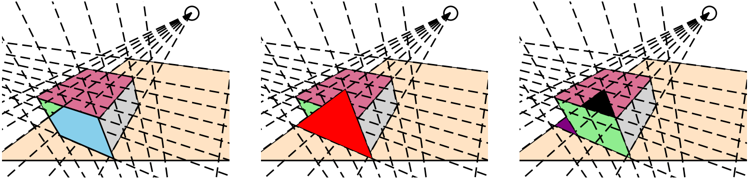
- Compute more accumulators so that you can get constant time access



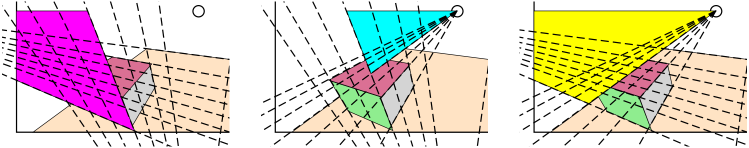
- This accumulators are also pairwise potentials!
- Bounds computed also by looking at min and max areas of each accumulator

Bounds for Penetration and Occlusion [Schwing et al., 2013]

- It looks complicated and high order!
- But look at pairs of faces and decompose intersections into triangles



- Compute more accumulators so that you can get constant time access



- This accumulators are also pairwise potentials!
- Bounds computed also by looking at min and max areas of each accumulator
- Sounds easy... but it's a nightmare ;)

- Experiments on the bedroom dataset (Hedau et al. 10)
- The layout is improved by 1.5%

		Top	Side	Hull	BB
loc	DPM (Felzenszwalb et al. 10)	-	-	56.12	57.14
	3D-DPM (Fidler et al. 12)	30.61	35.71	53.06	66.33
	Sup. DPM	-	-	61.22	63.27
	Ours	**35.05**	**39.18**	**68.04**	**74.23**

Table : Comparison to state-of-the-art in 3D detection.

Results: Importance of the features [Schwing et al., 2013]

		Intersection over union								Labeling measures			
		joint				greedy				joint		greedy	
		Top	Side	Hull	BB	Top	Side	Hull	BB	9L	5L	9L	5L
loc	Geo	25.51	19.39	48.98	64.29	26.53	24.49	50.00	63.27	26.16	22.00	26.62	22.70
	Geo+2D	33.67	27.55	60.20	65.31	33.67	27.55	60.20	65.31	24.34	21.44	24.46	21.45
	Geo+3D	**37.76**	38.78	60.20	71.43	**35.71**	37.76	60.20	69.39	23.20	20.43	23.95	**21.03**
	Geo+2D+3D	35.05	**39.18**	**68.04**	**74.23**	34.69	**38.78**	**65.31**	**74.49**	**22.65**	**20.30**	**23.81**	21.22
det	Geo	36.30	32.59	51.11	54.07	36.30	34.07	49.63	51.11	27.84	23.81	26.95	23.05
	Geo+2D	42.22	38.52	62.22	66.67	43.70	40.74	62.96	65.93	25.77	22.94	24.50	21.64
	Geo+3D	**44.44**	43.70	58.52	60.74	42.96	43.70	57.78	60.00	**24.45**	**21.64**	**24.28**	**21.37**
	Geo+2D+3D	42.96	**47.41**	**66.67**	**69.63**	**45.19**	**48.89**	**65.93**	**70.37**	24.66	21.67	24.57	21.73

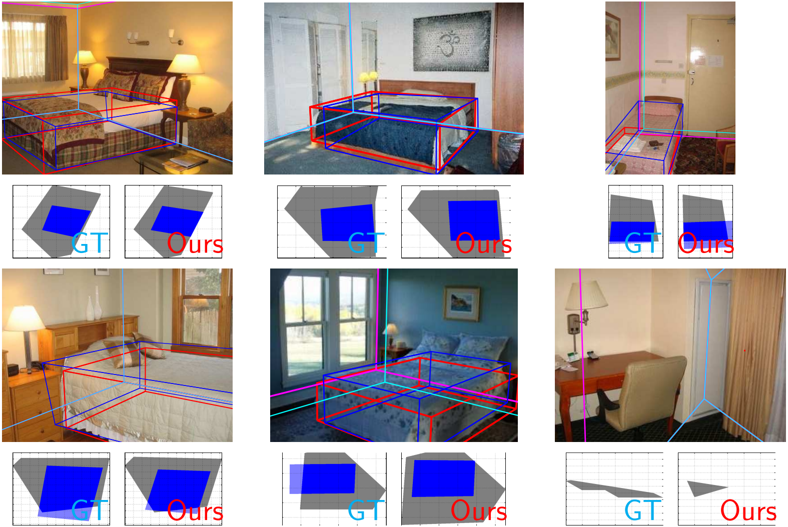
Table : Importance of the features: note that every feature we add generally improves detection. We refer to OM+GC features via *Geo*, the 2D detector via *2D*, and the 3D detector via *3D*.

	joint	greedy
Oracle 9L	12.88s	0.07s
Oracle 5L	6.95s	0.07s
Geo	331.43s	0.37s
Geo+2D	230.68s	0.30s
Geo+3D	583.18s	0.43s
Geo+2D+3D	3333.09s	1.58s

Table : Average inference time in seconds for the joint and greedy approach with different features provided

	Pascal			Average		
	Floor	Object	Free	Floor	Object	Free
Oracle 9L	89.76	62.22	77.95	77.22	62.83	64.64
Oracle 5L	90.55	60.00	77.95	78.37	60.81	64.88
Geo	63.78	29.63	35.43	57.21	35.07	40.47
Geo+2D	**71.65**	29.63	39.37	**59.24**	37.76	42.40
Geo+3D	68.50	**37.78**	**40.94**	58.36	40.95	**43.33**
Geo+2D+3D	70.63	37.04	38.89	58.64	**41.92**	42.05

Table : Computation of average F1 score for intersection over union of floor, object footprint and free-space for joint inference with indicated features. While the Pascal approach counts scores larger than 0.5 as correct detections, we also provide the mean.


GT
Ours
GT
Ours
GT
Ours
GT
Ours
GT
Ours
GT
Ours