Structure Prediction for 3D Scene Understanding

Raquel Urtasun

TTI Chicago

August 2, 2013

Which problems we will be looking at?

- 3D Layout estimation
- 3D object detection

A little bit about structure prediction

Recipe for Success using Structure Prediction

- What are my random variables?
- How are they related? i.e., graph

Recipe for Success using Structure Prediction

- What are my random variables?
- How are they related? i.e., graph

Recipe for Success using Structure Prediction

- What are my random variables?
- How are they related? i.e., graph
- How do I encode my prior knowledge about the problem?

$$E(y_1, \cdots, y_n, \mathbf{x}) = \sum_{r \in \mathcal{R}} \mathbf{w}_r^T \phi_r(\mathbf{y}_r, \mathbf{x})$$

Recipe for Success using Structure Prediction

- What are my random variables?
- How are they related? i.e., graph
- How do I encode my prior knowledge about the problem?

$$E(y_1, \cdots, y_n, \mathbf{x}) = \sum_{r \in \mathcal{R}} \mathbf{w}_r^T \phi_r(\mathbf{y}_r, \mathbf{x})$$

- **Advise**: Forget about probabilities in your potentials, the partition function will take care of that!

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp(-E(\mathbf{y}, \mathbf{x}))$$

Recipe for Success using Structure Prediction

- What are my random variables?
- How are they related? i.e., graph
- How do I encode my prior knowledge about the problem?

$$E(y_1, \cdots, y_n, \mathbf{x}) = \sum_{r \in \mathcal{R}} \mathbf{w}_r^T \phi_r(\mathbf{y}_r, \mathbf{x})$$

- **Advise**: Forget about probabilities in your potentials, the partition function will take care of that!

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp(-E(\mathbf{y}, \mathbf{x}))$$

- How can I do inference? Why is this complicated?

$$\min_{y_1, \cdots, y_n} E(y_1, \cdots, y_n)$$

Recipe for Success using Structure Prediction

- What are my random variables?
- How are they related? i.e., graph
- How do I encode my prior knowledge about the problem?

$$E(y_1, \cdots, y_n, \mathbf{x}) = \sum_{r \in \mathcal{R}} \mathbf{w}_r^T \phi_r(\mathbf{y}_r, \mathbf{x})$$

- **Advise**: Forget about probabilities in your potentials, the partition function will take care of that!

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp(-E(\mathbf{y}, \mathbf{x}))$$

- How can I do inference? Why is this complicated?

$$\min_{y_1, \cdots, y_n} E(y_1, \cdots, y_n)$$

- If you know how to do inference you will know how to do learning! Where does the complication come from?

Why Would I Use Structure Prediction?

- Why to worry about math if I can hack up something quickly? $\rightarrow$ there is still room for hackers!
- It allows you to abstract and encode models to solve your problems

Why Would I Use Structure Prediction?

- Why to worry about math if I can hack up something quickly? $\rightarrow$ there is still room for hackers!
- It allows you to abstract and encode models to solve your problems
- Captures well the combinatorial structure of some problems

Why Would I Use Structure Prediction?

- Why to worry about math if I can hack up something quickly? $\rightarrow$ there is still room for hackers!
- It allows you to abstract and encode models to solve your problems
- Captures well the combinatorial structure of some problems
- Easy to reason jointly about multiple problems

Why Would I Use Structure Prediction?

- Why to worry about math if I can hack up something quickly? $\rightarrow$ there is still room for hackers!
- It allows you to abstract and encode models to solve your problems
- Captures well the combinatorial structure of some problems
- Easy to reason jointly about multiple problems
- Why do I care about holistic (i.e., joint) models?

Why Would I Use Structure Prediction?

- Why to worry about math if I can hack up something quickly? $\rightarrow$ there is still room for hackers!
- It allows you to abstract and encode models to solve your problems
- Captures well the combinatorial structure of some problems
- Easy to reason jointly about multiple problems
- Why do I care about holistic (i.e., joint) models?
- Well understood inference algorithms, some of them exact!

Why Would I Use Structure Prediction?

- Why to worry about math if I can hack up something quickly? $\rightarrow$ there is still room for hackers!
- It allows you to abstract and encode models to solve your problems
- Captures well the combinatorial structure of some problems
- Easy to reason jointly about multiple problems
- Why do I care about holistic (i.e., joint) models?
- Well understood inference algorithms, some of them exact!
- Good learning algorithms exist as well

Why Would I Use Structure Prediction?

- Why to worry about math if I can hack up something quickly? $\rightarrow$ there is still room for hackers!
- It allows you to abstract and encode models to solve your problems
- Captures well the combinatorial structure of some problems
- Easy to reason jointly about multiple problems
- Why do I care about holistic (i.e., joint) models?
- Well understood inference algorithms, some of them exact!
- Good learning algorithms exist as well

What's not so good?

- Use as a keyword, approaches that don't think about how the problem is represented, how the energy looks like, etc.
- Particularly overloaded terms, e.g., high-order potentials

What's not so good?

- Use as a keyword, approaches that don't think about how the problem is represented, how the energy looks like, etc.
- Particularly overloaded terms, e.g., high-order potentials
- Problems with continuous variables: we need better algorithms!

What's not so good?

- Use as a keyword, approaches that don't think about how the problem is represented, how the energy looks like, etc.
- Particularly overloaded terms, e.g., high-order potentials
- Problems with continuous variables: we need better algorithms!
- Do I need to understand inference? Yes, yes and yes! I don't think this is a negative point though

What's not so good?

- Use as a keyword, approaches that don't think about how the problem is represented, how the energy looks like, etc.
- Particularly overloaded terms, e.g., high-order potentials
- Problems with continuous variables: we need better algorithms!
- Do I need to understand inference? Yes, yes and yes! I don't think this is a negative point though
- Is a log-linear model expressive enough?

What's not so good?

- Use as a keyword, approaches that don't think about how the problem is represented, how the energy looks like, etc.
- Particularly overloaded terms, e.g., high-order potentials
- Problems with continuous variables: we need better algorithms!
- Do I need to understand inference? Yes, yes and yes! I don't think this is a negative point though
- Is a log-linear model expressive enough?
- Where does the structure come from?

What's not so good?

- Use as a keyword, approaches that don't think about how the problem is represented, how the energy looks like, etc.
- Particularly overloaded terms, e.g., high-order potentials
- Problems with continuous variables: we need better algorithms!
- Do I need to understand inference? Yes, yes and yes! I don't think this is a negative point though
- Is a log-linear model expressive enough?
- Where does the structure come from?
- Can I learn everything from unlabeled data? How deep are you?

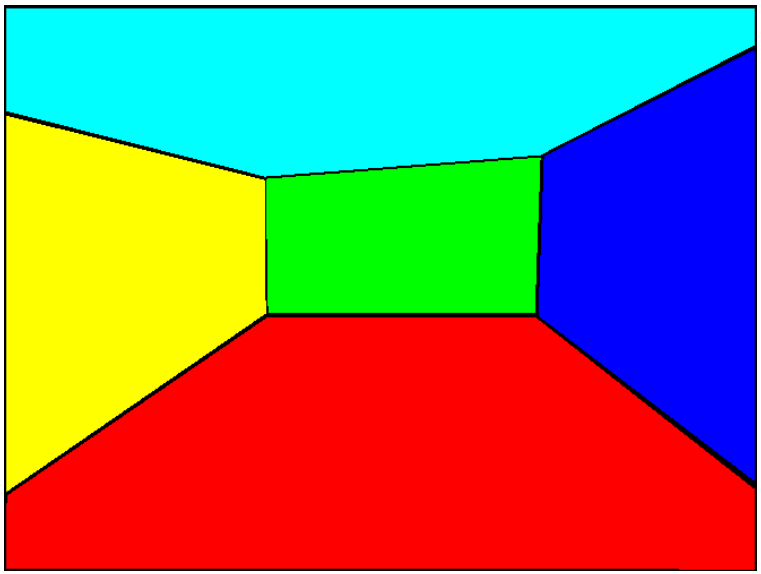
What's not so good?

- Use as a keyword, approaches that don't think about how the problem is represented, how the energy looks like, etc.
- Particularly overloaded terms, e.g., high-order potentials
- Problems with continuous variables: we need better algorithms!
- Do I need to understand inference? Yes, yes and yes! I don't think this is a negative point though
- Is a log-linear model expressive enough?
- Where does the structure come from?
- Can I learn everything from unlabeled data? How deep are you?

First task: 3D indoor scene understanding

3D layout for Indoors

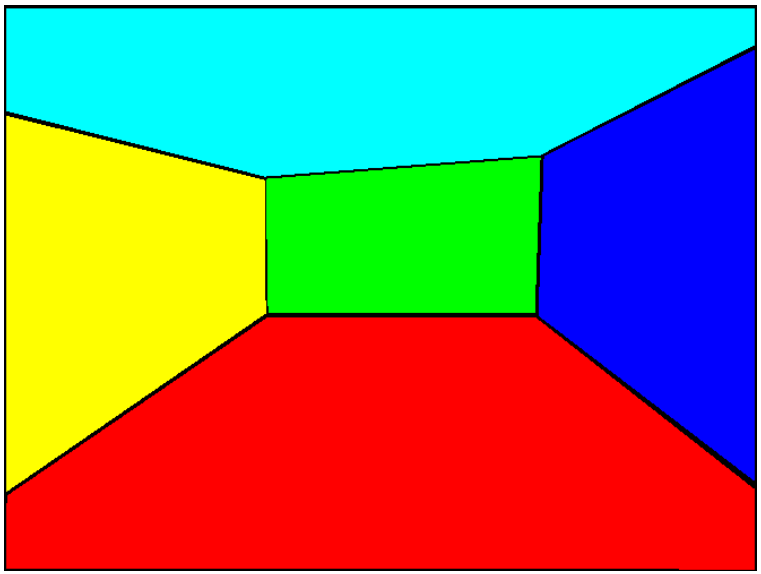
Task: Estimate the 3D layout from a single image



- What's the metric? how do I know if I did well?
- How would you parameterize this problem? (i.e., what are your random variables?)

3D layout for Indoors

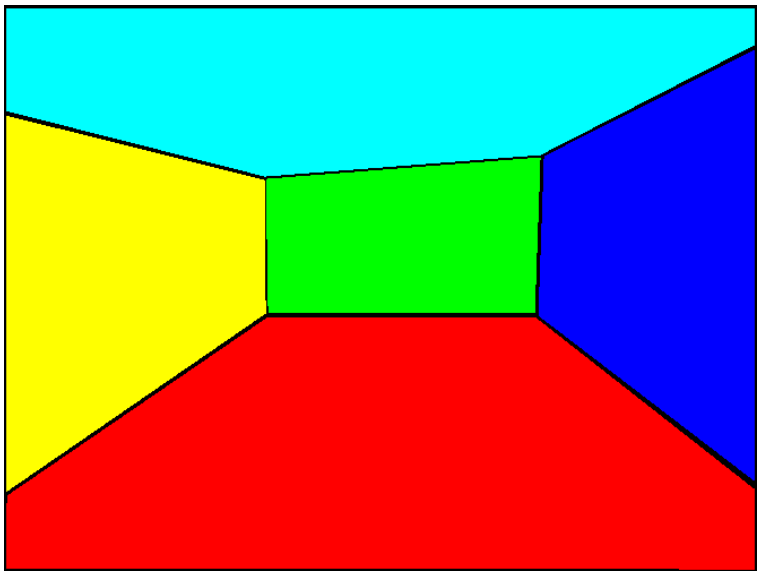
Task: Estimate the 3D layout from a single image



- What's the metric? how do I know if I did well?
- How would you parameterize this problem? (i.e., what are your random variables?)
- What prior knowledge would you like to encode?

3D layout for Indoors

Task: Estimate the 3D layout from a single image



- What's the metric? how do I know if I did well?
- How would you parameterize this problem? (i.e., what are your random variables?)
- What prior knowledge would you like to encode?

3D layout for Indoors

- Isn't this a segmentation task where each pixel can be labeled as a wall?

3D layout for Indoors

- Isn't this a segmentation task where each pixel can be labeled as a wall?
- Let's start with the most simple parameterization: split the image into super pixels, and for each define

$$y_i \in \{1, \cdots, 5\}$$

the label the super pixel is associated with

3D layout for Indoors

- Isn't this a segmentation task where each pixel can be labeled as a wall?
- Let's start with the most simple parameterization: split the image into super pixels, and for each define

$$y_i \in \{1, \cdots, 5\}$$

  the label the super pixel is associated with
- Define the energy as

$$E(y_1, \cdots, y_n, \mathbf{x}) = \sum_{r \in \mathcal{R}} \mathbf{w}_r^T \phi_r(\mathbf{y}_r, \mathbf{x})$$

- What are the $\phi_r(\mathbf{y}_r, \mathbf{x})$?

3D layout for Indoors

- Isn't this a segmentation task where each pixel can be labeled as a wall?
- Let's start with the most simple parameterization: split the image into super pixels, and for each define

$$y_i \in \{1, \cdots, 5\}$$

  the label the super pixel is associated with
- Define the energy as

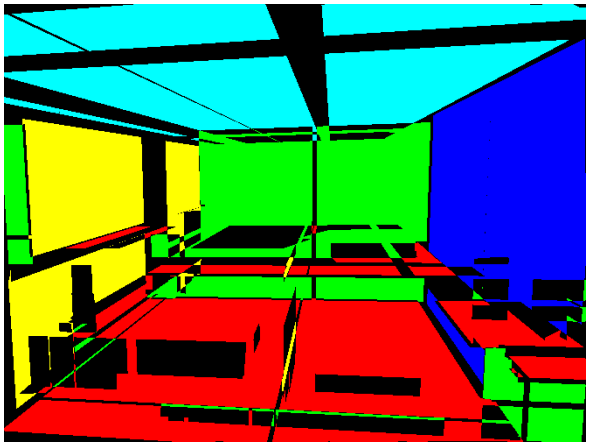
$$E(y_1, \cdots, y_n, \mathbf{x}) = \sum_{r \in \mathcal{R}} \mathbf{w}_r^T \phi_r(\mathbf{y}_r, \mathbf{x})$$

- What are the $\phi_r(\mathbf{y}_r, \mathbf{x})$?

Geometric Features as Unaries

- Orientation maps [Leet el al 09], geometric context [Hoiem et al. 05]

original image



orientation map

geometric context

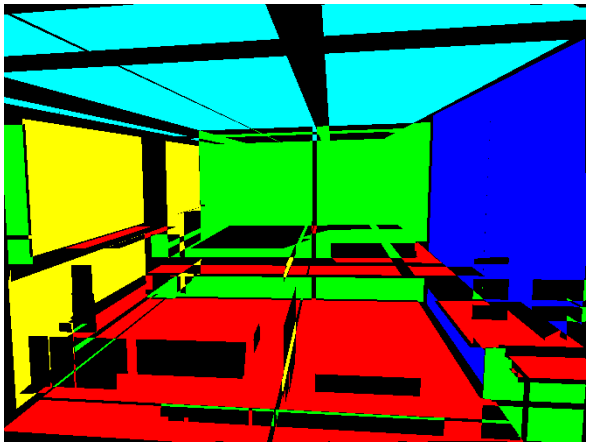
How do I construct my unaries $\phi_i(\mathbf{x}, y_i)$?

- What are my pairwise potentials $\phi_{ij}(\mathbf{x}, y_i, y_j)$?

Geometric Features as Unaries

- Orientation maps [Leet el al 09], geometric context [Hoiem et al. 05]

original image



orientation map

geometric context

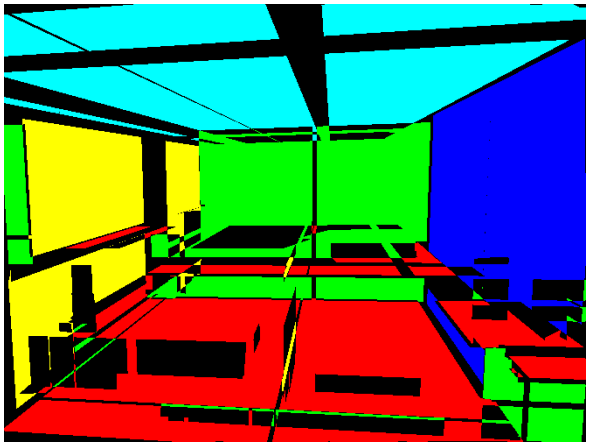
How do I construct my unaries $\phi_i(\mathbf{x}, y_i)$?

- What are my pairwise potentials $\phi_{ij}(\mathbf{x}, y_i, y_j)$?
- What's the problem with smoothness potentials?

Geometric Features as Unaries

- Orientation maps [Leet el al 09], geometric context [Hoiem et al. 05]

original image



orientation map

geometric context

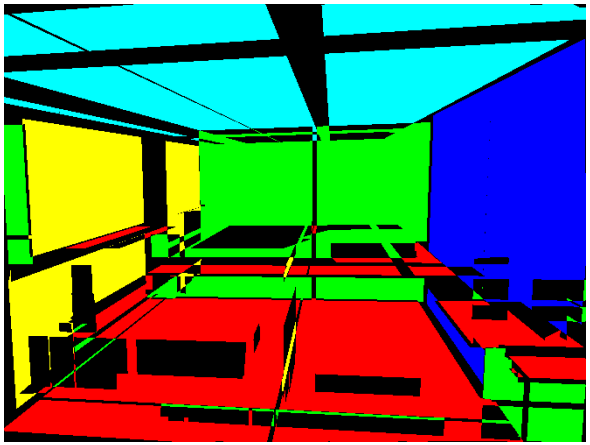
How do I construct my unaries $\phi_i(\mathbf{x}, y_i)$?

- What are my pairwise potentials $\phi_{ij}(\mathbf{x}, y_i, y_j)$?
- What's the problem with smoothness potentials?
- Are we missing something? What extra knowledge do we have?

Geometric Features as Unaries

- Orientation maps [Leet el al 09], geometric context [Hoiem et al. 05]

original image



orientation map

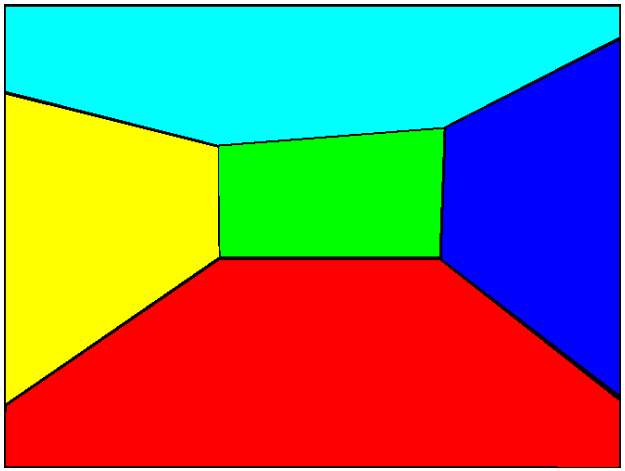
geometric context

How do I construct my unaries $\phi_i(\mathbf{x}, y_i)$?

- What are my pairwise potentials $\phi_{ij}(\mathbf{x}, y_i, y_j)$?
- What's the problem with smoothness potentials?
- Are we missing something? What extra knowledge do we have?

Manhattan World for Segmentation

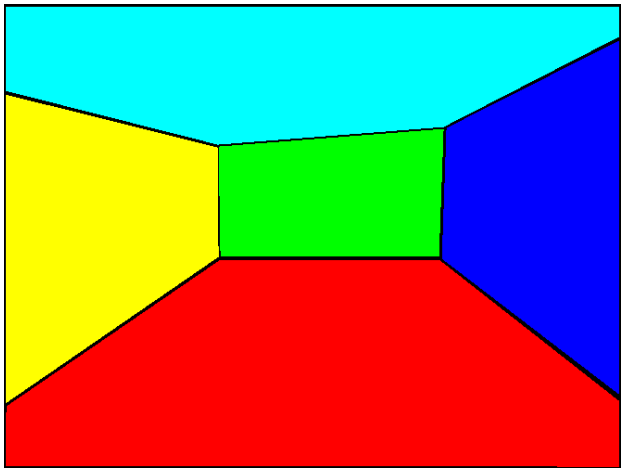
- Labels are not appearing at random in the image



- We can encode that the world is Manhattan by expressing **ordering** constraints

Manhattan World for Segmentation

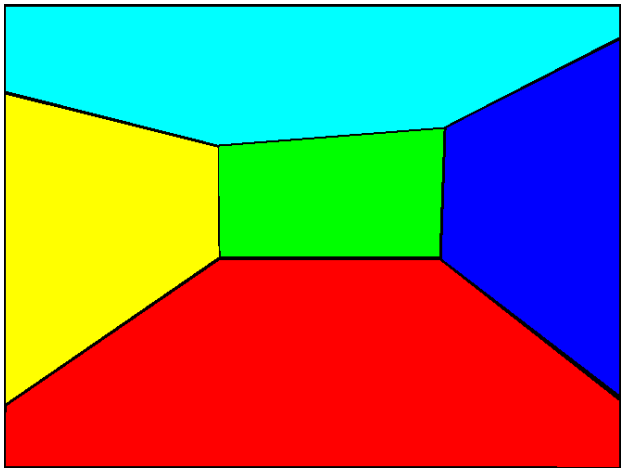
- Labels are not appearing at random in the image



- We can encode that the world is Manhattan by expressing **ordering** constraints
- What would that be?

Manhattan World for Segmentation

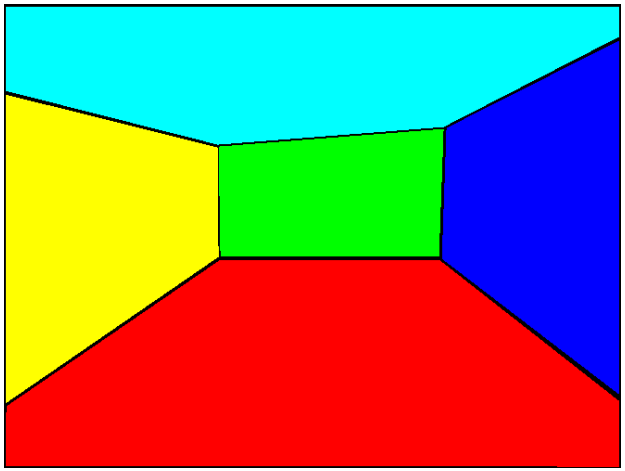
- Labels are not appearing at random in the image



- We can encode that the world is Manhattan by expressing **ordering** constraints
- What would that be?
- What's the order of the potentials?

Manhattan World for Segmentation

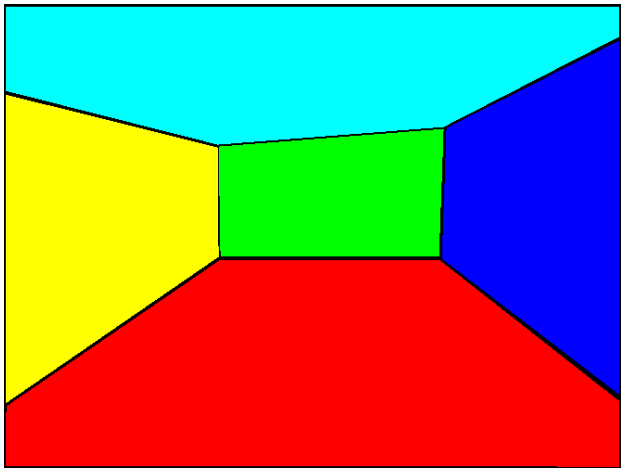
- Labels are not appearing at random in the image



- We can encode that the world is Manhattan by expressing **ordering** constraints
- What would that be?
- What's the order of the potentials?
- Can we do inference easily?

Manhattan World for Segmentation

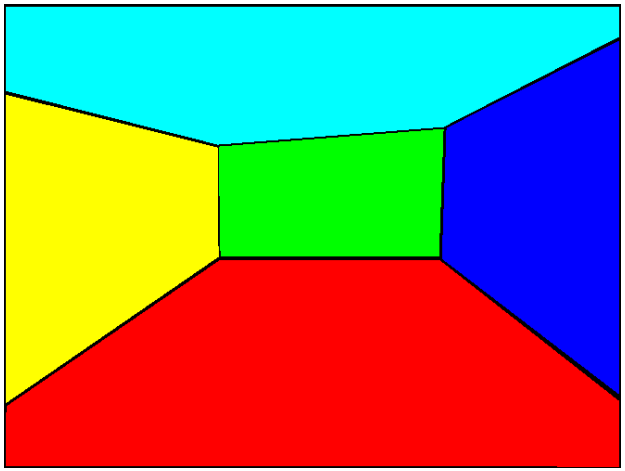
- Labels are not appearing at random in the image



- We can encode that the world is Manhattan by expressing **ordering** constraints
- What would that be?
- What's the order of the potentials?
- Can we do inference easily?
- Which algorithm will you use? would it take a long time? would it be optimal?

Manhattan World for Segmentation

- Labels are not appearing at random in the image



- We can encode that the world is Manhattan by expressing **ordering** constraints
- What would that be?
- What's the order of the potentials?
- Can we do inference easily?
- Which algorithm will you use? would it take a long time? would it be optimal?

Encoding Manhattan World Structure

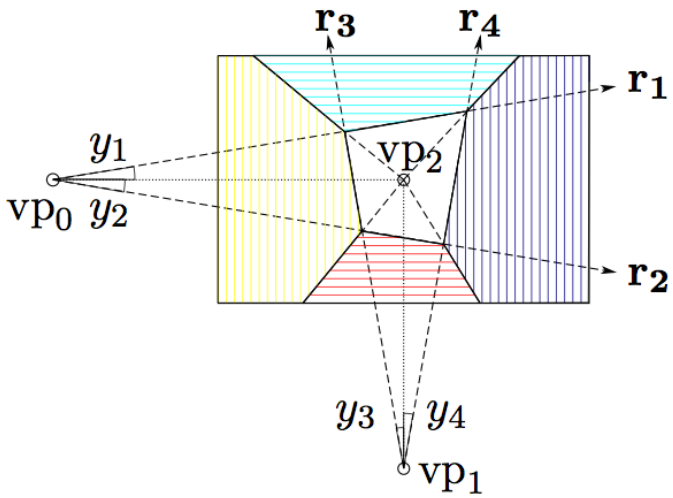
- Let's assume that I can compute vanishing points
- How should I express the problem? how many degrees of freedom do I have?

Encoding Manhattan World Structure

- Let's assume that I can compute vanishing points
- How should I express the problem? how many degrees of freedom do I have?

Encoding Manhattan World Structure

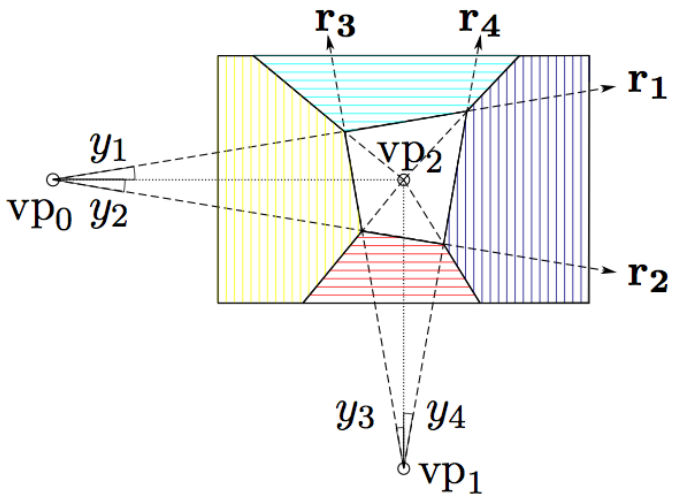
- Let's assume that I can compute vanishing points
- How should I express the problem? how many degrees of freedom do I have?
- We parameterize a layout with 4 variables $y_i \in \mathcal{Y}$, $i \in \{1, ..., 4\}$ [Lee et al. 09]



- What have I lost with respect to before?

Encoding Manhattan World Structure

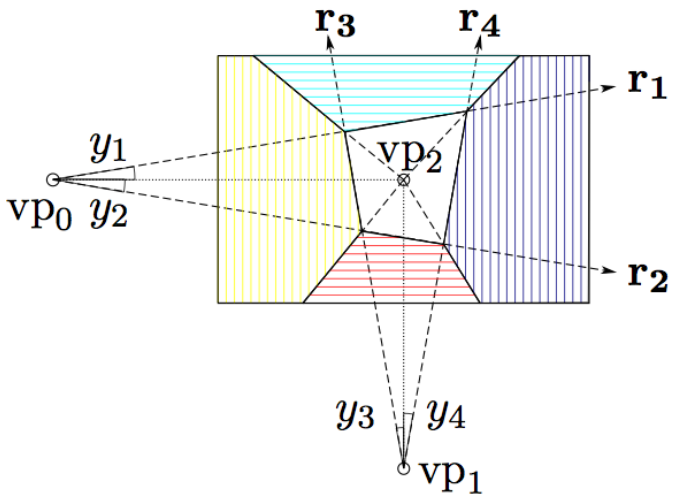
- Let's assume that I can compute vanishing points
- How should I express the problem? how many degrees of freedom do I have?
- We parameterize a layout with 4 variables $y_i \in \mathcal{Y}$, $i \in \{1, ..., 4\}$ [Lee et al. 09]



- What have I lost with respect to before?
- What have I won?

Encoding Manhattan World Structure

- Let's assume that I can compute vanishing points
- How should I express the problem? how many degrees of freedom do I have?
- We parameterize a layout with 4 variables $y_i \in \mathcal{Y}$, $i \in \{1, ..., 4\}$ [Lee et al. 09]



- What have I lost with respect to before?
- What have I won?

Energy of the problem

- Let's define the energy. Which potentials will you use?

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

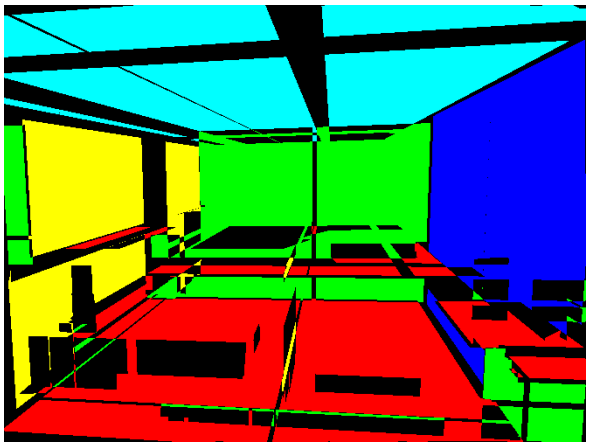
Energy of the problem

- Let's define the energy. Which potentials will you use?

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

- Let's start with the geometric features

original image



orientation map

geometric context

- We will like to maximize the yellow pixels in the left wall, green in the frontal wall, etc

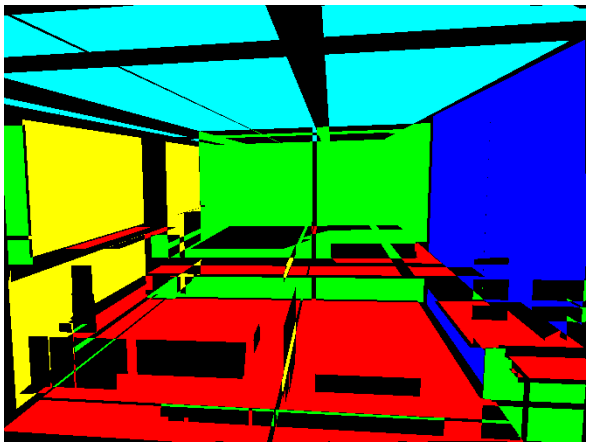
Energy of the problem

- Let's define the energy. Which potentials will you use?

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

- Let's start with the geometric features

original image



orientation map

geometric context

- We will like to maximize the yellow pixels in the left wall, green in the frontal wall, etc
- We will also like to minimize the other colors in those walls, e.g., all but yellow in left wall

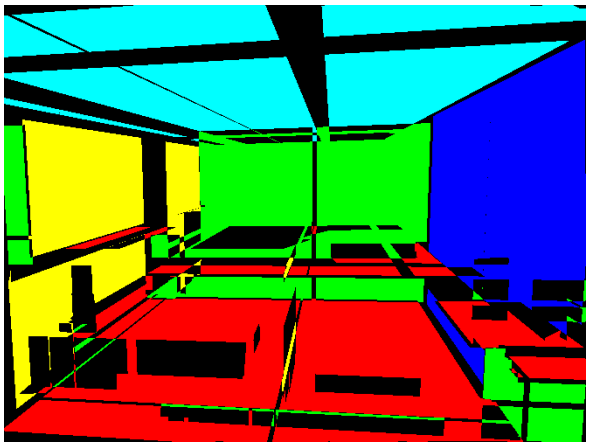
Energy of the problem

- Let's define the energy. Which potentials will you use?

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

- Let's start with the geometric features

original image



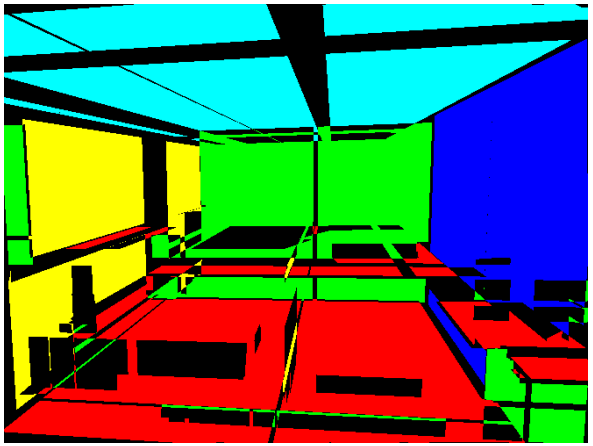
orientation map

geometric context

- We will like to maximize the yellow pixels in the left wall, green in the frontal wall, etc
- We will also like to minimize the other colors in those walls, e.g., all but yellow in left wall

More on energy

original image



orientation map

geometric context

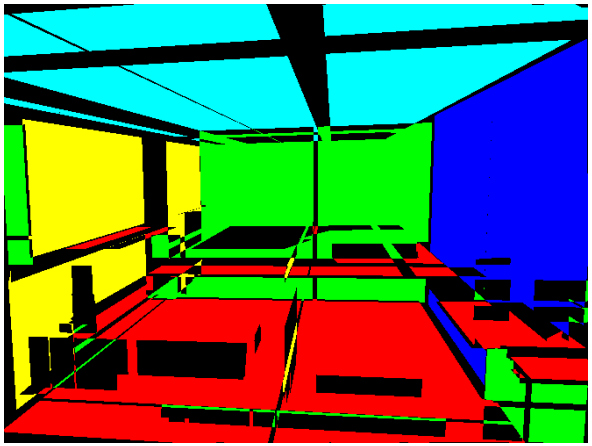
- How do I express this in my potentials?

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

- How many y_i's do I need to define them?

More on energy

original image



orientation map

geometric context

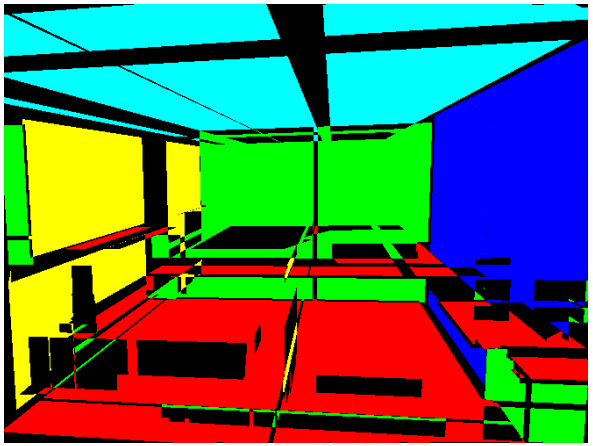
- How do I express this in my potentials?

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

- How many y_i's do I need to define them?
- Do I need other potentials?

More on energy

original image



orientation map

geometric context

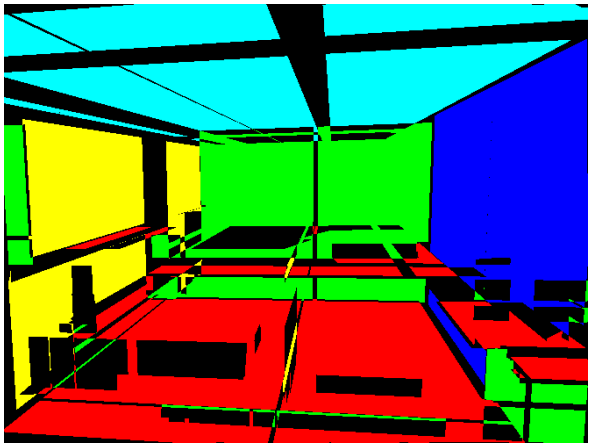
- How do I express this in my potentials?

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

- How many y_i's do I need to define them?
- Do I need other potentials?
- Why did I need more potentials than just geometric features before?

More on energy

original image



orientation map

geometric context

- How do I express this in my potentials?

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

- How many y_i's do I need to define them?
- Do I need other potentials?
- Why did I need more potentials than just geometric features before?

Inference

- It's inference easy in this model? Why?
- What can we do?

Inference

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

Inference

- It's 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

- It's 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

- It's 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

- It's 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

Inference

- It's 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

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

- 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?

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

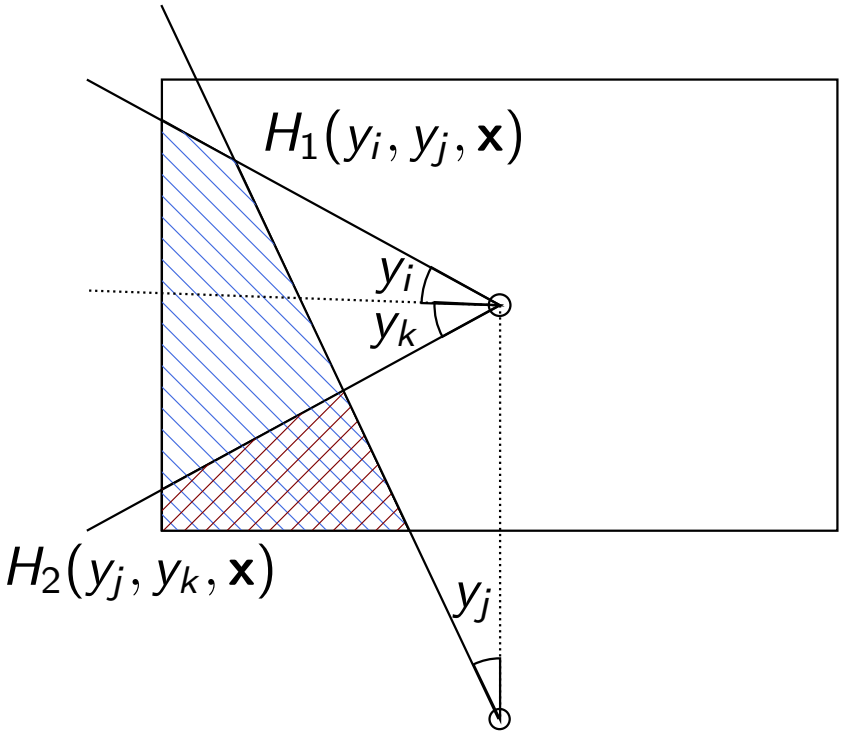
$$S([i_0,i_1] \times [j_0,j_1]) = s(i_1,j_1) - s(i_1,j_0-1) - s(i_0-1,j_1) + s(i_0-1,j_0-1)$$

- Can we do something similar in our case?

Generalization to 3D

- 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?

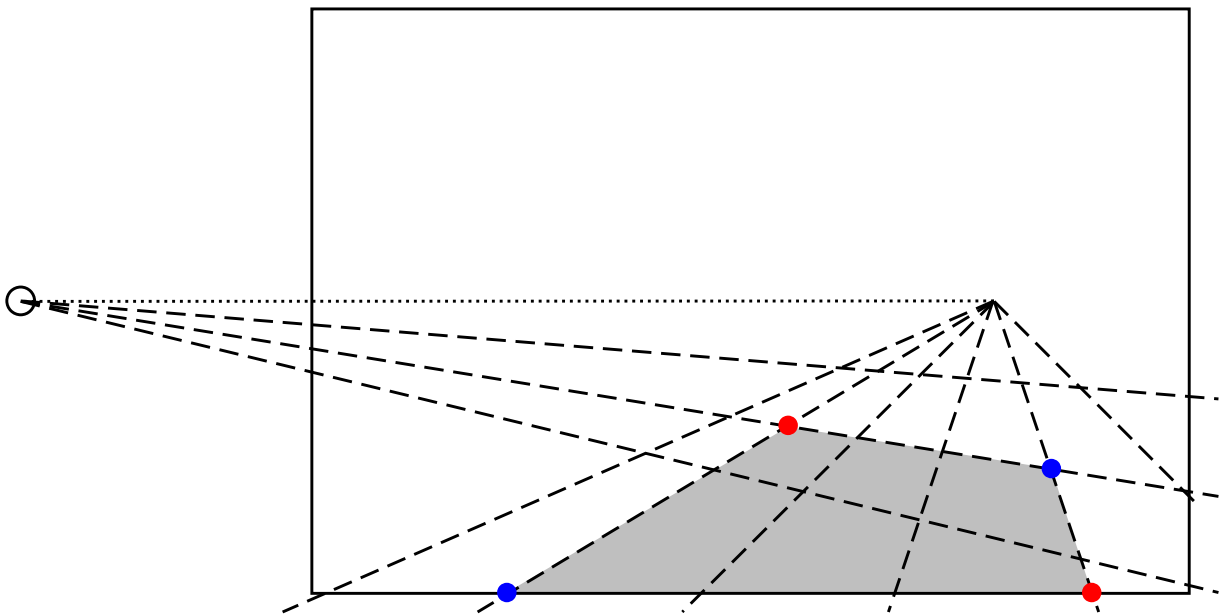
$$\phi_{\{left\_w\}}(y_i, y_j, y_k, \mathbf{x}) \quad = \quad H_1(y_i, y_j, \mathbf{x}) - H_2(y_j, y_k, \mathbf{x})$$



Generalization to 3D

- 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?

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



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?

- 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?

- 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 this r share the same weights, as they come from the integral geometry.

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
- Good news is that it still depends on pairwise potentials (which are accumulators) but there is quite a few more
- Some of this 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?

- 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 this 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 speed ups the message passing inference by a few orders of magnitude

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
- Good news is that it still depends on pairwise potentials (which are accumulators) but there is quite a few more
- Some of this 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 speed ups the message passing inference by a few orders of magnitude

Exact Inference?

- Can we compute the optimal solution?
- The graph of the previous problem has a single loop

Exact Inference?

- Can we compute the optimal solution?
- The graph of the previous problem has a single loop
- Message passing will not give the optimal

Exact Inference?

- Can we compute the optimal solution?
- The graph of the previous problem has a single loop
- Message passing will not give the optimal
- What other algorithms do you know that give the optimal solution?

Exact Inference?

- Can we compute the optimal solution?
- The graph of the previous problem has a single loop
- Message passing will not give the optimal
- What other algorithms do you know that give the optimal solution?
- Let's look at branch and bound

Exact Inference?

- Can we compute the optimal solution?
- The graph of the previous problem has a single loop
- Message passing will not give the optimal
- What other algorithms do you know that give the optimal solution?
- Let's look at branch and bound

Branch and Bound

Algorithm 1 branch and bound (BB) inference

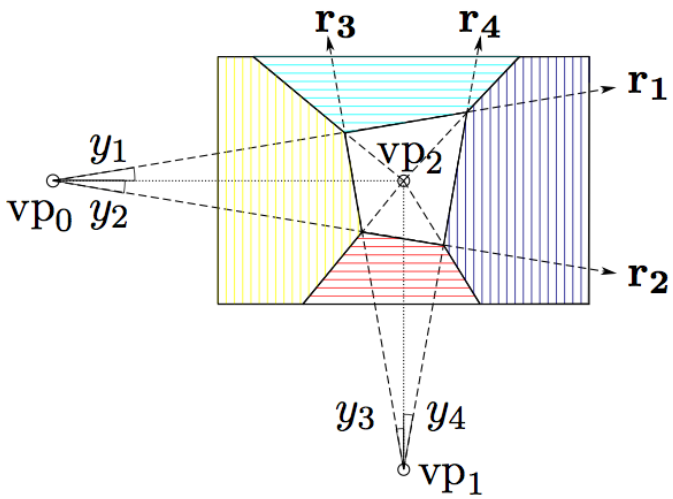
put pair $(\bar{f}(\mathcal{Y}), \mathcal{Y})$ into queue and set $\hat{\mathcal{Y}} = \mathcal{Y}$
repeat
  split $\hat{\mathcal{Y}} = \hat{\mathcal{Y}}_1 \times \hat{\mathcal{Y}}_2$ with $\hat{\mathcal{Y}}_1 \cap \hat{\mathcal{Y}}_2 = \emptyset$
  put pair $(\bar{f}(\hat{\mathcal{Y}}_1), \hat{\mathcal{Y}}_1)$ 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

- Layout with 4 variables $y_i \in \mathcal{Y}$, $i \in \{1, ..., 4\}$ [Lee et al. 09]
- 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:

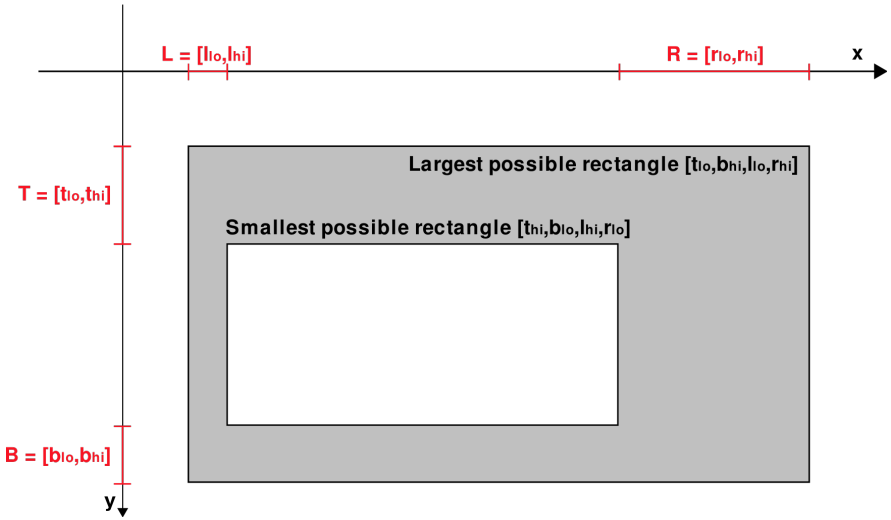
1. The bound of the interval $\hat{\mathcal{Y}}$ has to upper-bound the true cost of each hypothesis $y \in \hat{\mathcal{Y}}$,
$$\forall y \in \hat{\mathcal{Y}}, \quad \bar{f}(\hat{\mathcal{Y}}) \geq \mathbf{w}^T \phi(\mathbf{y}, \mathbf{x}).$$
2. The bound has to be exact for every single hypothesis,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

Can we define this for our problem?

Intuitions from 2D

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



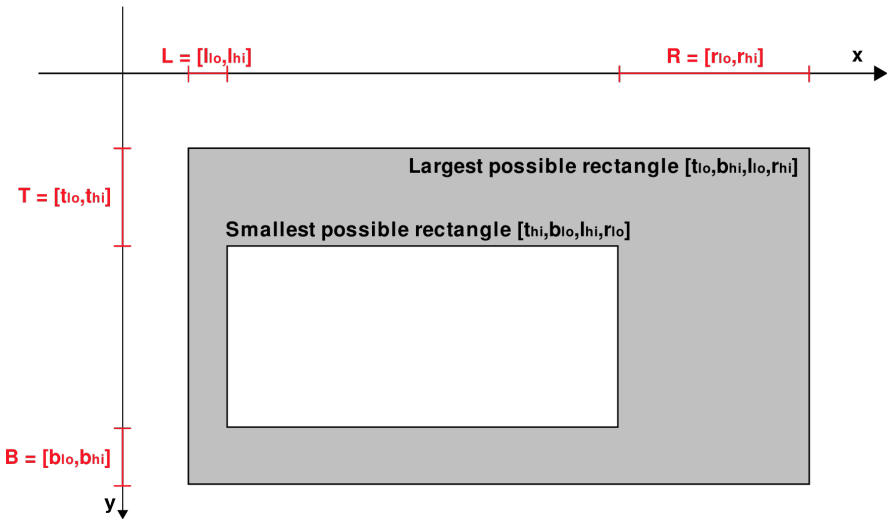
- The scoring function sums features in the rectangle defined by the BBox

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

Intuitions from 2D

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

- 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

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

  $\Rightarrow$ show an illustration

- 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

- 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

- 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

- 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

- 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?

Bounding the functions

- 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

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

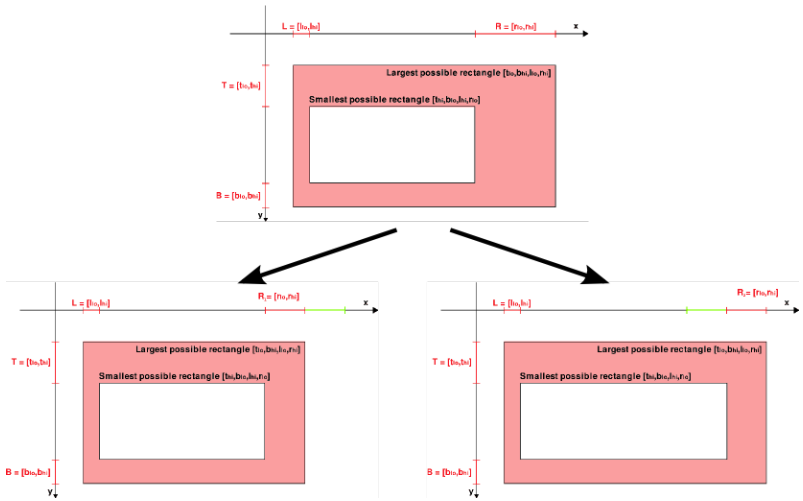
- These bounds are very simple? What are they?
- How can we compute them very fast?
- What's the complexity of computing them?
- How many integral images do we need?

Algorithm for 2D BBox [Lampert et al. 06]

Algorithm 1 Efficient Subwindow Search

Require: image x
Require: quality bounding function $\hat{f}$ (see Sect.III)
Ensure: $(t_{\text{opt}}, b_{\text{opt}}, l_{\text{opt}}, r_{\text{opt}}) = \text{argmax}_{y \in \mathcal{Y}} f(y)$
  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 $([T_1, B_1, L_1, R_1]; \hat{f}([T_1, B_1, L_1, R_1]))$ onto P
    push $([T_2, B_2, L_2, R_2]; \hat{f}([T_2, B_2, L_2, R_2]))$ onto P
    retrieve top state $[T, B, L, R]$ from P
  until $[T, B, L, R]$ consists of only one rectangle
  set $(t_{\text{opt}}, b_{\text{opt}}, l_{\text{opt}}, r_{\text{opt}}) = [T, B, L, R]$

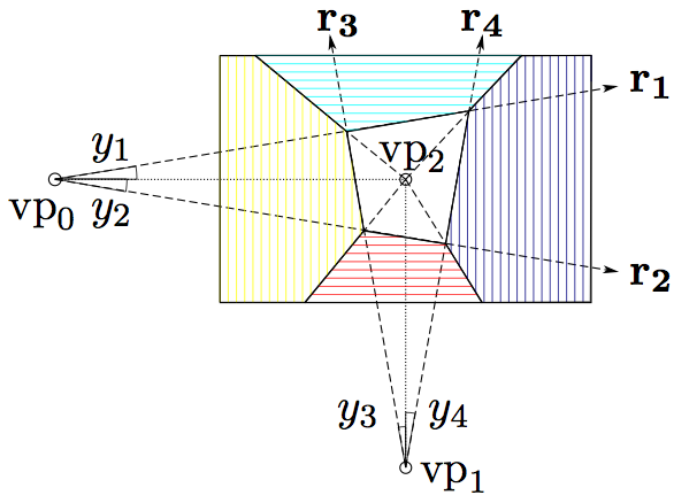
- How do we split?



- When do we terminate?

3D layout estimation

- 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

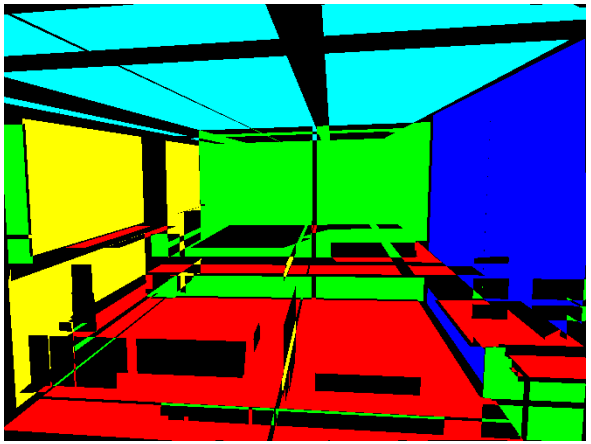
- The scoring function sums features over the faces

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

  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?

Deriving bounds

- Inference can be then done by

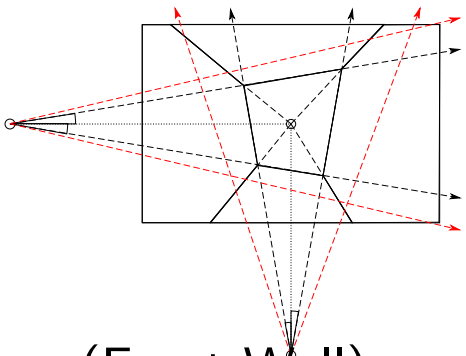
$$E(y_1, \cdots, y_4) = \sum_{\alpha} f_{\alpha}^{+}(x, y) + f_{\alpha}^{-}(x, y),$$

- We can bound each of this terms separately

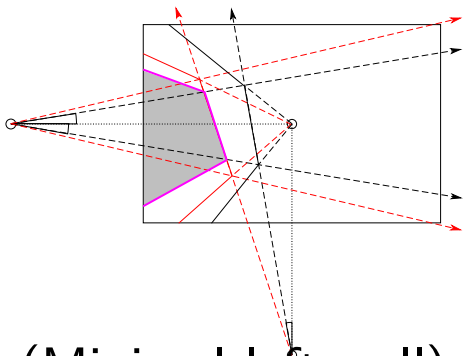
$$bound(E(\hat{\mathcal{Y}}, \mathbf{x})) = \sum_{\alpha \in \mathcal{F}} \bar{f}_{\alpha}^{+}(\hat{\mathcal{Y}}, \mathbf{x}) + \bar{f}_{\alpha}^{-}(\hat{\mathcal{Y}}, \mathbf{x})$$

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

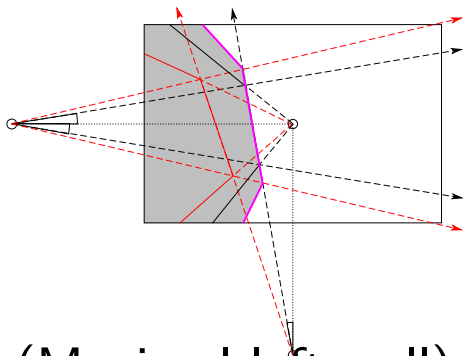
$$\bar{f}_{\textit{front-wall}}(\hat{\mathcal{Y}}) = f^{+}_{\textit{front-wall}}(x, y_{up}) + f^{-}_{\textit{front-wall}}(x, y_{low}),$$



(Front Wall)



(Minimal left wall)



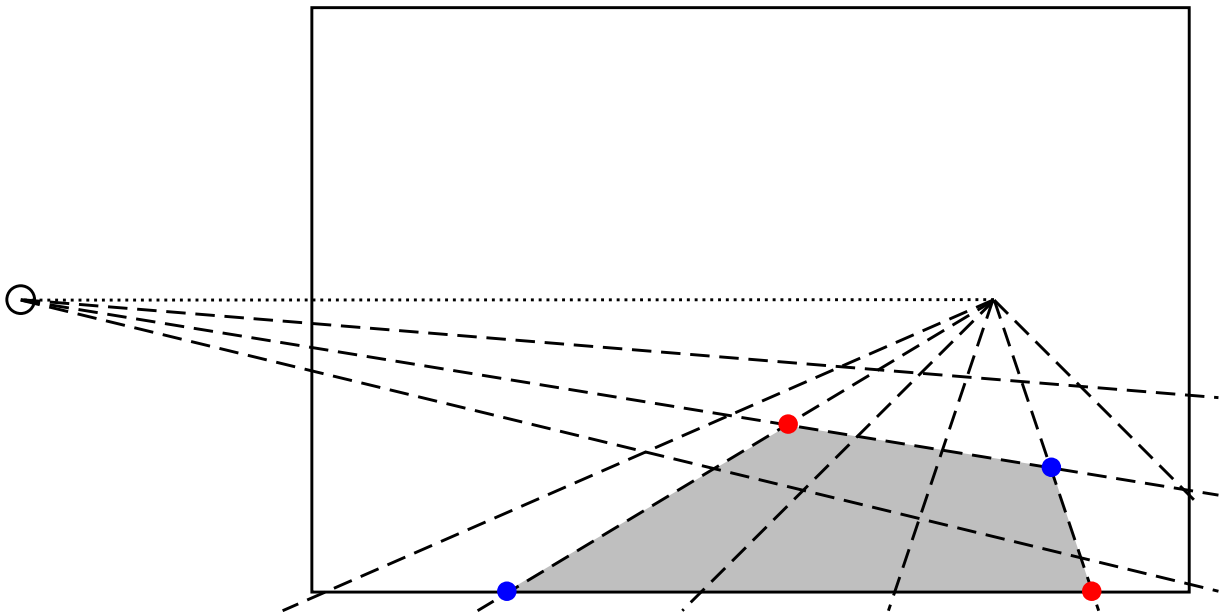
(Maximal left wall)

Efficient bounds

- How can we compute the bounds efficiently?

Efficient bounds

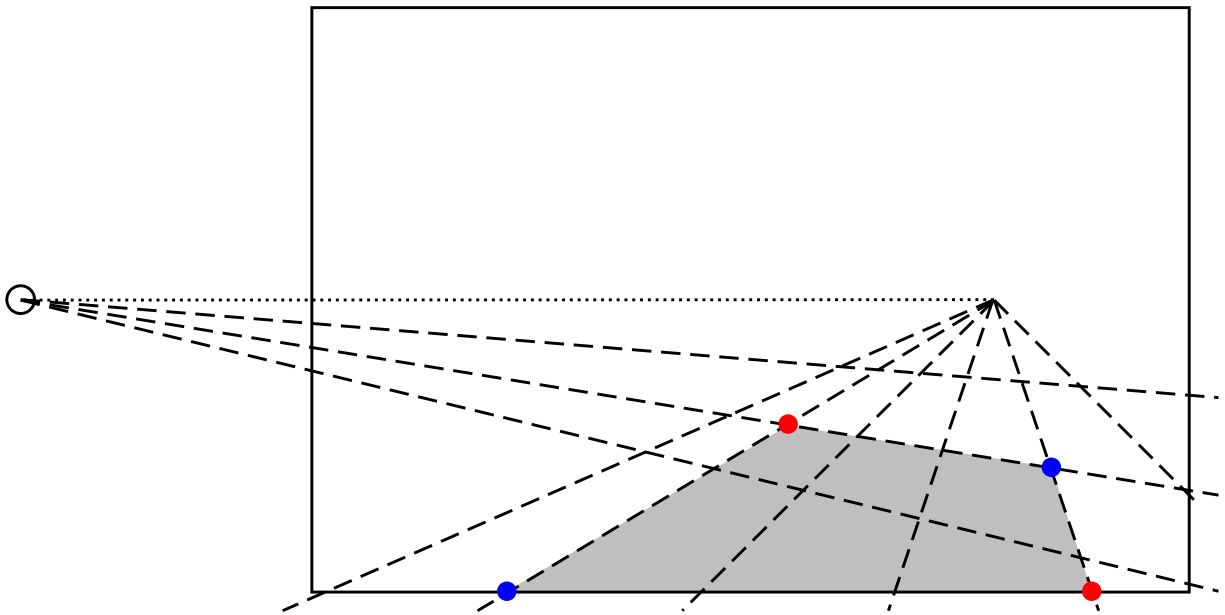
- How can we compute the bounds efficiently?



- What's the complexity?

Efficient bounds

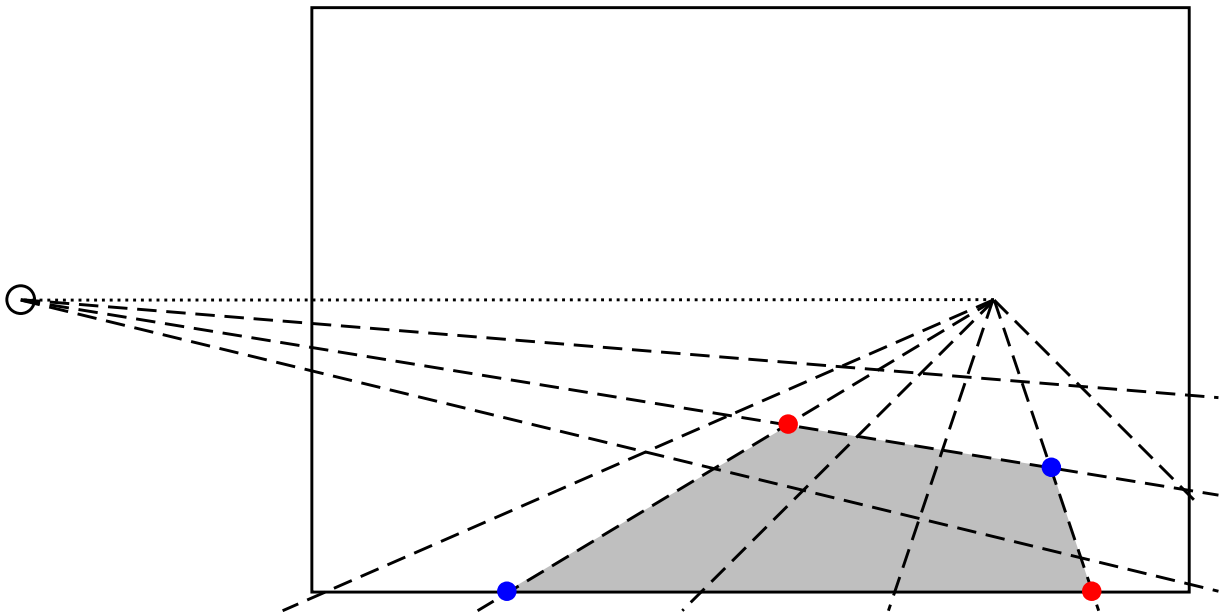
- How can we compute the bounds efficiently?



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

Efficient bounds

- How can we compute the bounds efficiently?



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

Results

[A. Schwing and R. Urtasun, ECCV12]

Table: Pixel classification error in the layout dataset of [Hedau et al. 09].

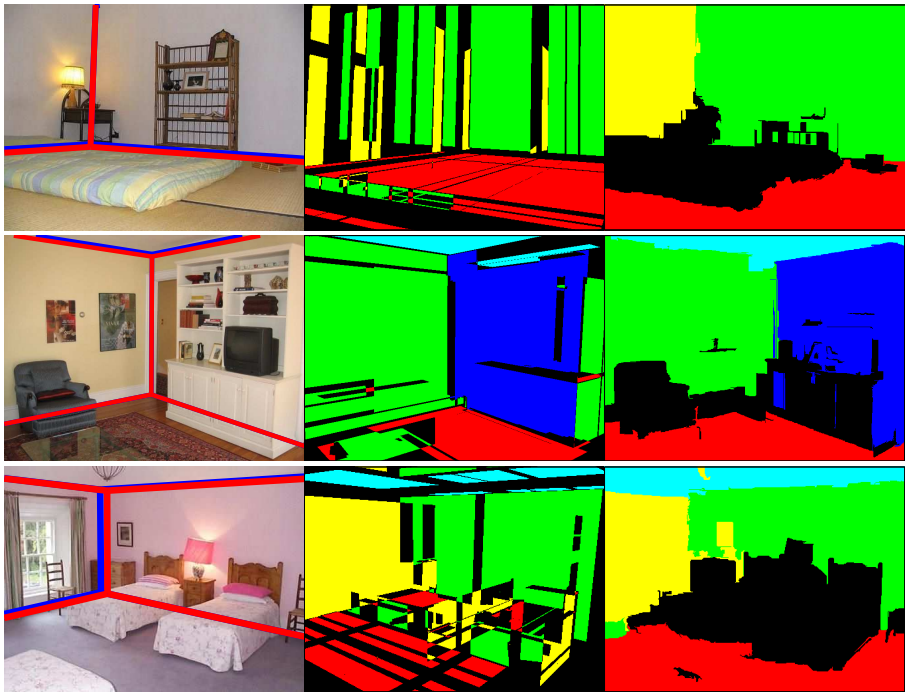
	OM	GC	OM + GC	Other	Time
[Hoiem07]	-	28.9	-	-	-
[Hedau09] (a)	-	26.5	-	-	-
[Hedau09] (b)	-	21.2	-	-	10-30 min
[Wang10]	22.2	-	-	-	
[Lee10]	24.7	22.7	18.6	-	-
[delPero11]	-	-	-	16.3	12 min
Ours	**18.6**	**15.4**	**13.6**	-	0.007s

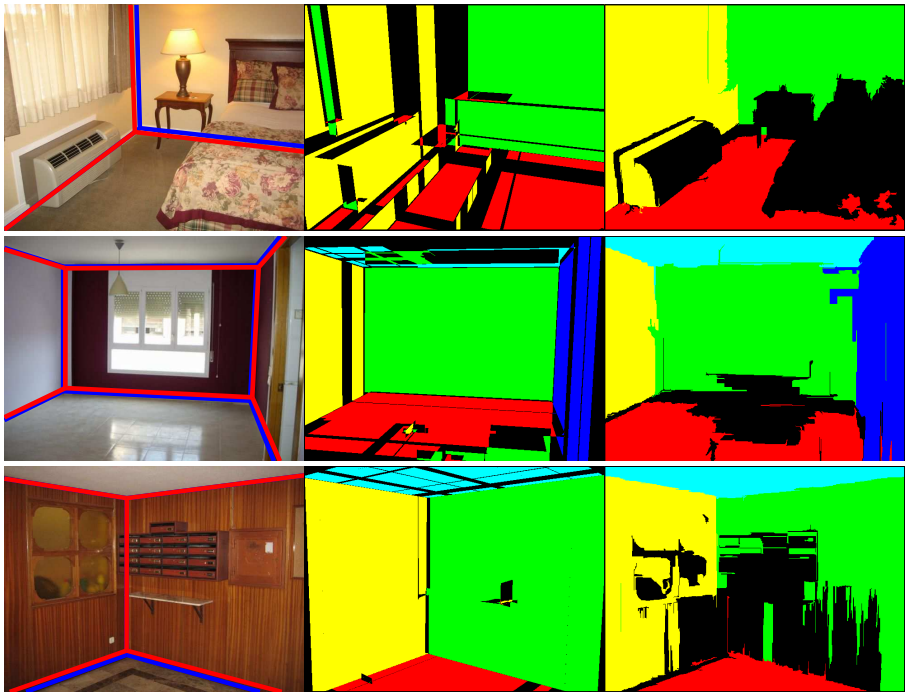
Table: Pixel classification error in the bedroom data set [Hedau et al. 10].

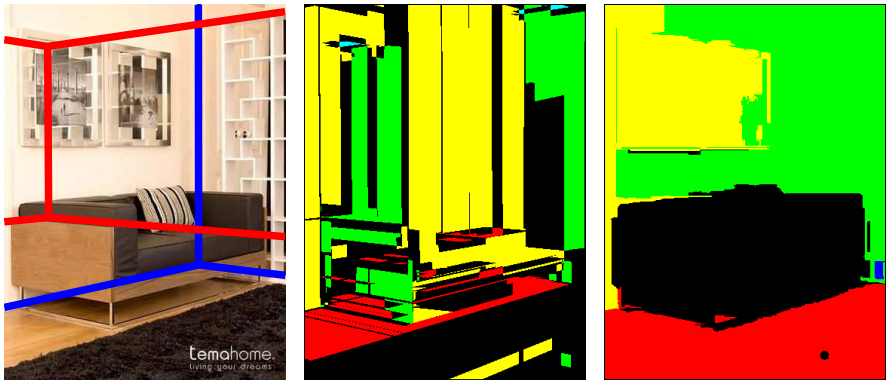
	[delPero11]	[Hoiem07]	[Hedau09](a)	Ours
w/o box	29.59	23.04	22.94	**16.46**

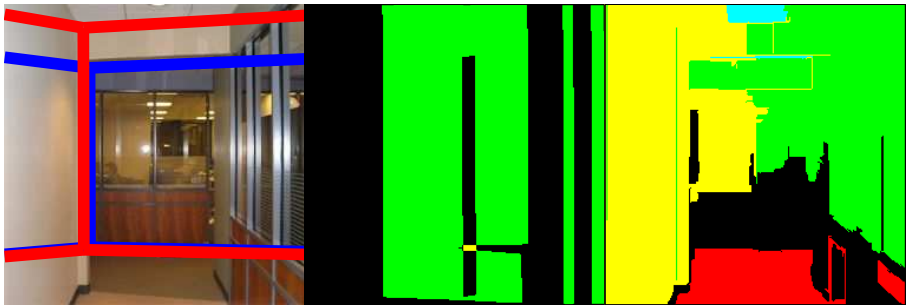
- Takes on average 0.007s for exact solution over 50^4 possibilities !
- It's 6 orders of magnitude faster than the state-of-the-art!

Qualitative Results









Let's try to detect objects in 3D

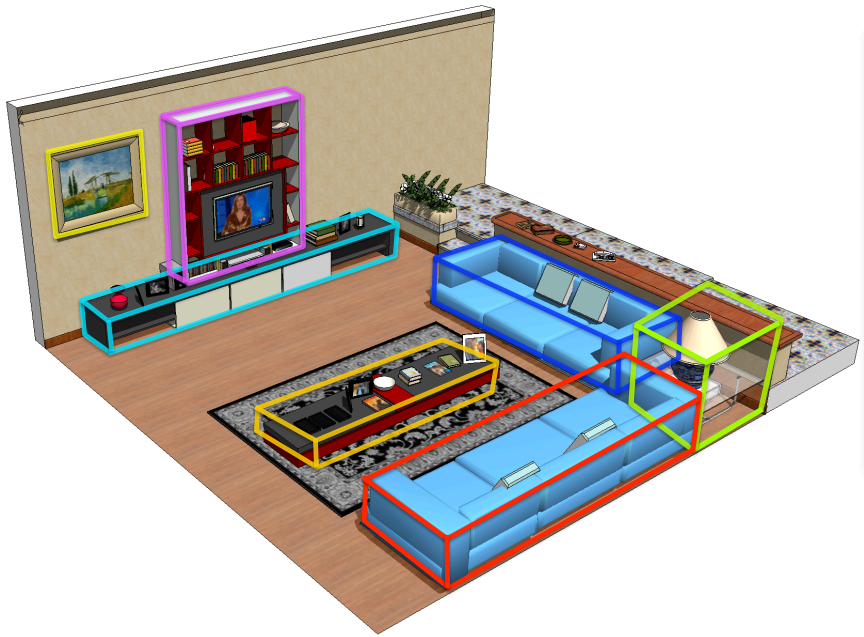
3D Object Detection

- **Task:** Given an image (e.g., rgb, rgbd, video), detect the 3D objects present in the scene

Figure: Image from [Jia et al. 13]

Contextual Models for 3D Object Detection

- Simple approach: Imagine you were able "somehow" to get candidate 3D bounding boxes
- **Task**: identify the object labels labels (e.g., bed, table) as well as which ones are outliers
- Objects are not independent!
- This is however the assumption of most object detectors (both 2D and 3D)
- Can we create a model which reasons about multiple objects?



Contextual Models for 3D Object Detection

- What would be the random variables?
- For each bounding box, $y_i \in \{0, 1\}$ saying whether it is correct or not, or $y_i \in \{0, 1, \cdots, C\}$
- When to use which parameterization?
- We can then write the energy

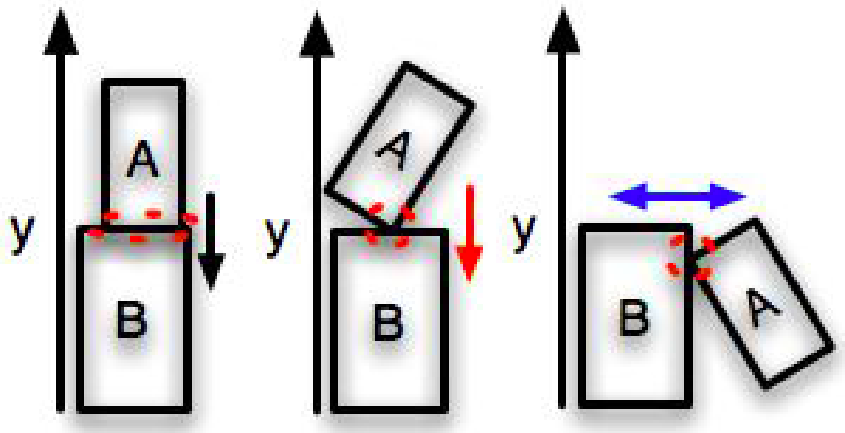
$$E(y_1, \cdots, y_n) = \sum_r \mathbf{w}^T \phi_r(\mathbf{y}, \mathbf{x})$$

- What would you encode in the potentials?
- What's the underlying graph?

Our prior knowledge about the problem

If we have 3D blocks, then physics can be used to constrained object location, orientation, size, etc

- **Stability:** blocks are put such that they don't fell [Gupta et al. 10, Jia et al. 13]

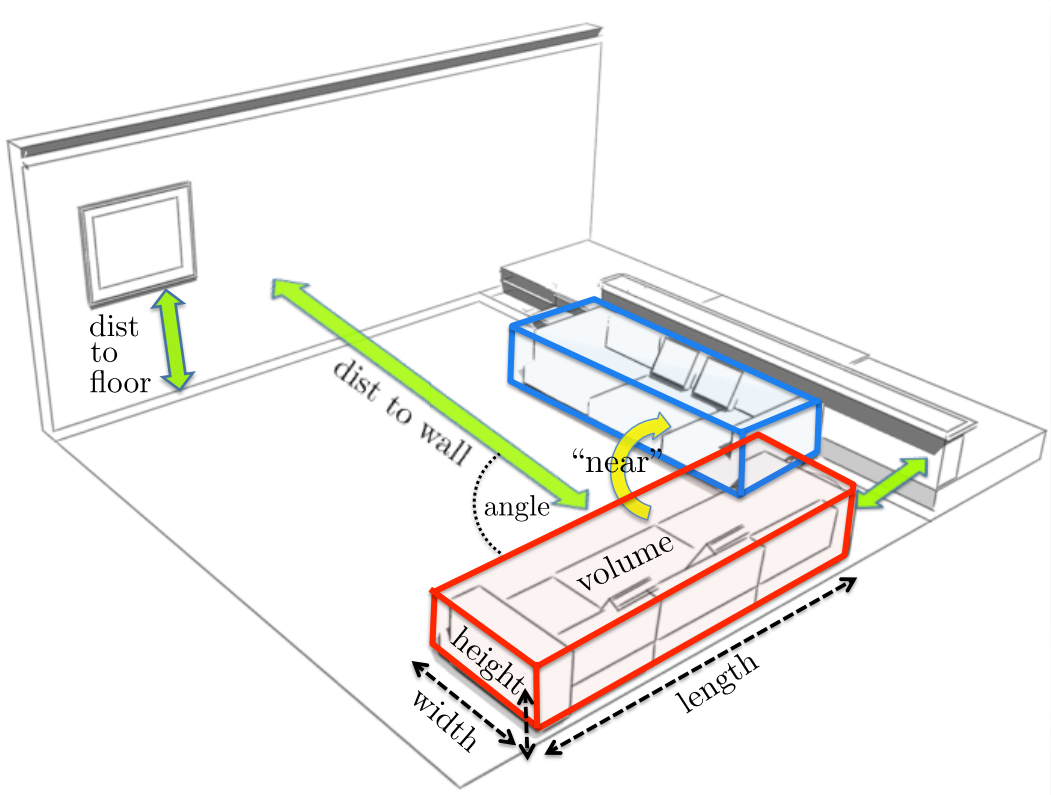


- **Support:** a cup is on the table, but a table is not on a cup [Silberman et al. 12, Jia et al. 13]
- **Semantic coherence:** co-occurance of objects [Jia et al. 13, Lin et al. 13]

Our prior knowledge about the problem

If we have 3D blocks, then physics can be used to constrained object location, orientation, size, etc

- **Location:** where objects appear with respect to the room [Hedau?] and each other [Lin et al. 13]



Our prior knowledge about the problem

If we have 3D blocks, then physics can be used to constrained object location, orientation, size, etc

- **Size coherence:** Relative scale of objects [Lin et al. 13]
- **High level context:** type of scene, e.g., a cow can't be in a living room [Lin et al. 13]
- **Layout:** objects do not penetrate the room layout [Lee10, Hedau 10, Schwing12a, delPero12]
- more?

Inference

- All these things mentioned are pairwise potentials (i.e., relations between two objects)
- If those are sub modular, use graph cuts!
- If not message passing
- In any case, you can do inference in ms [Lin et al. 13]
- Use standard methods for learning, e.g., CRF log loss or structured SVMs

Results