What excites me ... and hopefully you too ...

Developing autonomous systems that are able to help us in everyday’s tasks
My view of how to get there …

Autonomous systems need to:

- Sense the environment
- Recognize the 3D world
- Interact with it

What’s important?

- Representation
- Learning
- Inference
- Data

I believe in **holistic approaches** that solve multiple tasks jointly and MRFs provide a great mathematical framework
Contents

1) Data

2) Learning and Inference

3) Sensing the environment

4) Recognition
Contents

1) Data

2) Learning and Inference

3) Sensing the environment

4) Recognition
Problems of Interest are Structured

- We are interested in predicting multiple variables which are related.
Structured Prediction

- Input: $x \in \mathcal{X}$, typically an image
- Output: structured label $y \in \mathcal{Y}$
- The score function $\theta(x, y)$ called potential or feature such that

$$\theta(x, y) = \begin{cases} 
\text{high} & \text{if } y \text{ is a good label for } x \\
\text{low} & \text{if } y \text{ is a bad label for } x 
\end{cases}$$

- We assume that the score decomposes

$$\theta(x, y) = \sum_i \theta_i(x, y_i) + \sum_\alpha \theta_\alpha(x, y_\alpha)$$

- This represents a Markov Random Field (MRF)

$$p(x, y) = \frac{1}{Z} \prod_i \psi_i(x, y_i) \prod_\alpha \psi_\alpha(x, y_\alpha)$$
Region Representation

- Factor graph representation

\[ \theta(x, y) = \sum_{i} \theta_i(x, y_i) + \sum_{\alpha} \theta_{\alpha}(x, y_{\alpha}) \]

- Region representation

\[ \theta(x, y) = \sum_{r \in \mathcal{R}} \theta_r(x, y_r) \]
We want to compute the MAP estimate

\[ y_1^*, \ldots, y_n^* = \arg \max_{y_1, \ldots, y_n} \sum_{r \in R} \theta_r(x, y_r) \]

This is NP hard in general.

Challenges for inference:

1. Come up with good approximations to solve MAP
2. Deal with large-scale problems: exploit parallel architectures in terms of memory and computation
3. Deal with continuous output variables
We want to compute the MAP estimate

\[ y_1^*, \cdots, y_n^* = \arg \max_{y_1, \cdots, y_n} \sum_{r \in R} \theta_r(x, y_r) \]

This is NP hard in general.

Challenges for inference:

1. Come up with good approximations to solve MAP
2. Deal with large-scale problems: exploit parallel architectures in terms of memory and computation
3. Deal with continuous output variables
Alternative Formulation

- Solving the problem is hard, as it involves exponential many labels
  \[
  \max_{y_1, \cdots, y_n} \sum_{r \in R} \theta_r(x, y_r)
  \]

- The ILP formulation introduces indicator variables \( b_r(y_r) \) for each region
  \[
  \max_b \sum_{r \in R, y_r} b_r(y_r) \theta_r(y_r)
  \]
  \[
  \text{s.t.} \quad \forall r, y_r \quad b_r(y_r) \in \{0, 1\}
  \]
  \[
  \forall r, y_r, p \in P(r) \quad \sum_{y_p \setminus y_r} b_p(y_p) = b_r(y_r).
  \]

- This is still NP hard in general
Solving the problem is hard, as it involves exponential many labels

$$\max_{y_1, \ldots, y_n} \sum_{r \in \mathcal{R}} \theta_r(x, y_r)$$

The ILP formulation introduces indicator variables $b_r(y_r)$ for each region

$$\max_b \sum_{r \in \mathcal{R}, y_r} b_r(y_r)\theta_r(y_r)$$

subject to

$$\forall r, y_r \quad b_r(y_r) \in \{0, 1\}$$
$$\forall r, y_r, p \in P(r) \quad \sum_{y_p \setminus y_r} b_p(y_p) = b_r(y_r).$$

This is still NP hard in general

Smooth the problem
Solving the problem is hard, as it involves exponential many labels

$$\max_{y_1, \ldots, y_n} \sum_{r \in \mathcal{R}} \theta_r(x, y_r)$$

The ILP formulation introduces indicator variables $b_r(y_r)$ for each region

$$\max_b \sum_{r \in \mathcal{R}, y_r} b_r(y_r) \theta_r(y_r)$$

s.t.

$$\forall r, y_r \quad b_r(y_r) \in \{0, 1\}$$

$$\forall r, y_r, p \in P(r) \quad \sum_{y_p \setminus y_r} b_p(y_p) = b_r(y_r).$$

This is still NP hard in general

Smooth the problem

$$\max_b \sum_{r \in \mathcal{R}, y_r} b_r(y_r) \theta_r(y_r) + \epsilon H(b)$$

s.t.

$$\forall r, y_r \quad b_r(y_r) \in \{0, 1\}$$

$$\forall r, y_r, p \in P(r) \quad \sum_{y_p \setminus y_r} b_p(y_p) = b_r(y_r).$$
Smooth LP relaxations

- Relax the integrality constraints (i.e., $b_r(y_r) \in [0, 1]$) and approximate the entropy barrier functions

\[
\begin{align*}
\max_b & \quad \sum_{r \in \mathcal{R}, y_r} b_r(y_r) \theta_r(y_r) + \sum_r \epsilon c_r H(b_r) \\
\text{s.t.} & \quad \forall r \quad b_r \in \Delta \\
& \quad \forall r, y_r, p \in P(r) \quad \sum_{y_p \not= y_r} b_p(y_p) = b_r(y_r).
\end{align*}
\]

- Dualizing this program,

\[
\sum_r \epsilon c_r \ln \sum_{y_r} \exp \frac{\theta_r(y_r) - \sum_{p \in P(r)} \lambda_{r \rightarrow p}(y_r) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c)}{\epsilon c_r}.
\]
Smooth LP relaxations

- Relax the integrality constraints (i.e., $b_r(y_r) \in [0, 1]$) and approximate the entropy barrier functions

$$
\begin{align*}
\max_b & \quad \sum_{r \in \mathcal{R}, y_r} b_r(y_r) \theta_r(y_r) + \sum_r \epsilon c_r H(b_r) \\
\text{s.t.} & \quad \forall r \quad b_r \in \Delta \\
& \quad \forall r, y_r, p \in P(r) \quad \sum_{y_p \setminus y_r} b_p(y_p) = b_r(y_r).
\end{align*}
$$

- Dualizing this program,

$$
\sum_r \epsilon c_r \ln \sum_{y_r} \exp \left( \frac{\theta_r(y_r) - \sum_{p \in P(r)} \lambda_{r \rightarrow p}(y_r) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c)}{\epsilon c_r} \right).
$$

- Solving for the Lagrange multipliers results in very effective message passing algorithms [Hazan and Sashua, 2010] [Globerson et al. 07]
Smooth LP relaxations

- Relax the integrality constraints (i.e., $b_r(y_r) \in [0, 1]$) and approximate the entropy barrier functions

$$\max_b \sum_{r \in \mathcal{R}, y_r} b_r(y_r) \theta_r(y_r) + \sum_r \epsilon c_r H(b_r)$$

s.t.

$$\forall r \quad b_r \in \Delta$$

$$\forall r, y_r, p \in P(r) \quad \sum_{y_p \setminus y_r} b_p(y_p) = b_r(y_r).$$

- Dualizing this program,

$$\sum_r \epsilon c_r \ln \sum_{y_r} \exp \frac{\theta_r(y_r) - \sum_{p \in P(r)} \lambda_{r \rightarrow p} y_r + \sum_{c \in C(r)} \lambda_{c \rightarrow r} y_c}{\epsilon c_r}.$$

- Solving for the Lagrange multipliers results in very effective message passing algorithms [Hazan and Sashua, 2010] [Globerson et al. 07]
Convex Belief Propagation

Algorithm: Message Passing Inference
Repeat until convergence
Iterate over $r$:

\[
\forall p \in P(r), y_r \quad \mu_{p \rightarrow r}(y_r) = \epsilon_c \ln \sum_{y_p \setminus y_r} \exp \left( \frac{\theta_p(y_p) - \sum_{p' \in P(p)} \lambda_{p' \rightarrow p}(y_p) + \sum_{r' \in C(p) \setminus r} \lambda_{r' \rightarrow p}(y_{r'})}{\epsilon_c} \right)
\]

\[
\forall p \in P(r), y_r \quad \lambda_{r \rightarrow p}(y_r) \propto \frac{c_p}{c_r + \sum_{p \in P(r)} c_p} \left( \theta_r(y_r) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c) + \sum_{p \in P(r)} \mu_{p \rightarrow r}(y_r) \right) - \mu_{p \rightarrow r}(y_r)
\]
**Algorithm: Message Passing Inference (Max Product)**

Repeat until convergence

Iterate over $r$:

\[
\forall p \in P(r), y_r, \quad \mu_{p \rightarrow r}(y_r) = \max_{y_p \setminus y_r} \left( \theta_p(y_p) - \sum_{p' \in P(p)} \lambda_{p \rightarrow p'}(y_p) + \sum_{r' \in C(p) \setminus r} \lambda_{r' \rightarrow p}(y_{r'}) \right)
\]

\[
\forall p \in P(r), y_r, \quad \lambda_{r \rightarrow p}(y_r) \propto \frac{1}{1 + |P(r)|} \left( \theta_r(y_r) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c) + \sum_{p \in P(r)} \mu_{p \rightarrow r}(y_r) \right) - \mu_{p \rightarrow r}(y_r)
\]

**Algorithm: Message Passing Inference (Sum Product)**

Repeat until convergence

Iterate over $r$:

\[
\forall p \in P(r), y_r, \quad \mu_{p \rightarrow r}(y_r) = \ln \sum_{y_p \setminus y_r} \exp \left( \theta_p(y_p) - \sum_{p' \in P(p)} \lambda_{p \rightarrow p'}(y_p) + \sum_{r' \in C(p) \setminus r} \lambda_{r' \rightarrow p}(y_{r'}) \right)
\]

\[
\forall p \in P(r), y_r, \quad \lambda_{r \rightarrow p}(y_r) \propto \frac{1}{1 + |P(r)|} \left( \theta_r(y_r) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c) + \sum_{p \in P(r)} \mu_{p \rightarrow r}(y_r) \right) - \mu_{p \rightarrow r}(y_r)
\]
**Sum Product and Max Product Convex BP**

**Algorithm: Message Passing Inference (Max Product)**

Repeat until convergence

Iterate over \( r \):

\[
\forall p \in P(r), y_r \quad \mu_{p \rightarrow r}(y_r) = \max_{y_p \setminus y_r} \left( \theta_p(y_p) - \sum_{p' \in P(p)} \lambda_{p \rightarrow p'}(y_p) + \sum_{r' \in C(p) \setminus r} \lambda_{r' \rightarrow p}(y_{r'}) \right)
\]

\[
\forall p \in P(r), y_r \quad \lambda_{r \rightarrow p}(y_r) \propto \frac{1}{1 + |P(r)|} \left( \theta_r(y_r) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c) + \sum_{p \in P(r)} \mu_{p \rightarrow r}(y_r) \right) - \mu_{p \rightarrow r}(y_r)
\]

**Question:** How can we make this scale to large scale problems, e.g., 10Mpixel images?
Sum Product and Max Product Convex BP

Algorithm: Message Passing Inference (Max Product)
Repeat until convergence
Iterate over $r$:

$$
\forall p \in P(r), y_r \quad \mu_{p \rightarrow r}(y_r) = \max_{y_p \setminus y_r} \left( \theta_p(y_p) - \sum_{p' \in P(p)} \lambda_{p \rightarrow p'}(y_p) + \sum_{r' \in C(p) \setminus r} \lambda_{r' \rightarrow p}(y_{r'}) \right)
$$

$$
\forall p \in P(r), y_r \quad \lambda_{r \rightarrow p}(y_r) \propto \frac{1}{1 + |P(r)|} \left( \theta_r(y_r) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c) + \sum_{p \in P(r)} \mu_{p \rightarrow r}(y_r) \right) - \mu_{p \rightarrow r}(y_r)
$$

Algorithm: Message Passing Inference (Sum Product)
Repeat until convergence
Iterate over $r$:

$$
\forall p \in P(r), y_r \quad \mu_{p \rightarrow r}(y_r) = \ln \sum_{y_p \setminus y_r} \exp \left( \theta_p(y_p) - \sum_{p' \in P(p)} \lambda_{p \rightarrow p'}(y_p) + \sum_{r' \in C(p) \setminus r} \lambda_{r' \rightarrow p}(y_{r'}) \right)
$$

$$
\forall p \in P(r), y_r \quad \lambda_{r \rightarrow p}(y_r) \propto \frac{1}{1 + |P(r)|} \left( \theta_r(y_r) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c) + \sum_{p \in P(r)} \mu_{p \rightarrow r}(y_r) \right) - \mu_{p \rightarrow r}(y_r)
$$

Question: How can we make this scale to large scale problems, e.g., 10Mpixel images?
Cloud computing: Very large problems

[A. Schwing, T. Hazan, M. Pollefeys and R. Urtasun, CVPR11]

- **Dual decomposition**: partition the problem and add consistency constraints

![Diagram of dual decomposition]

- We obtain a dual problem with one additional set of Lagrange multipliers which are messages between machines

  solve sum of LP relaxations for each subgraph

  subject to:

  marginalization constraints

  \[ \forall \kappa, r \in R(\kappa), y_r \ b_r^\kappa(y_r) = b_r(y_r). \]
Cloud computing: Very large problems

[A. Schwing, T. Hazan, M. Pollefeys and R. Urtasun, CVPR11]

- **Dual decomposition**: partition the problem and add consistency constraints.

  ![Diagram of dual decomposition](image)

  We obtain a dual problem with one additional set of Lagrange multipliers which are messages between machines.

  - Solve sum of LP relaxations for each subgraph subject to:
    - Marginalization constraints:
      \[
      \forall \kappa, r \in R(\kappa), y_r \quad b_r^\kappa(y_r) = b_r(y_r).
      \]
    - Last constraint enforces consistency between the regions assigned to multiple computers.
    - Only constraint that couples the problems between different computers.
Cloud computing: Very large problems

Dual decomposition: partition the problem and add consistency constraints

We obtain a dual problem with one additional set of Lagrange multipliers which are messages between machines

solve sum of LP relaxations for each subgraph

subject to:

marginalization constraints

\[ \forall \kappa, r \in R(\kappa), y_r \ b_r^{\kappa}(y_r) = b_r(y_r) \].

Last constraint enforces consistency between the regions assigned to multiple computers.

Only constraint that couples the problems between different computers.
Distributed Convex BP Algorithm

[A. Schwing, M. Pollefeys, T. Hazan and R. Urtasun, CVPR11]

Algorithm: Distributed Message Passing Inference
Repeat until convergence

1. For every $\kappa$ in parallel: iterate over $r$:

For $p \in P(r), y_r$  
\[ \mu_{p \rightarrow r}(y_r) = \epsilon \hat{c}_p \ln \sum_{y_p \setminus y_r} \exp \left( \frac{\hat{\theta}_p(y_p) - \sum_{p' \in P(p)} \lambda_{p' \rightarrow p}(y_p) + \sum_{r' \in C(p) \cap \kappa \setminus r} \lambda_{r' \rightarrow p}(y_{r'}) + \nu_{\kappa \rightarrow p}(y_p)}{\epsilon \hat{c}_p} \right) \]

For $p \in P(r), y_r$  
\[ \lambda_{r \rightarrow p}(y_r) \propto \frac{\hat{c}_p}{\hat{c}_r + \sum_{p \in P(r)} \hat{c}_p} \left( \frac{\hat{\theta}_r(y_r) + \sum_{c \in C(r) \cap \kappa \setminus r} \lambda_{c \rightarrow r}(y_c) + \nu_{\kappa \rightarrow r}(y_r) + \sum_{p \in P(r)} \mu_{p \rightarrow r}(y_r)}{1} \right) - \mu_{p \rightarrow r}(y_r) \]

2. Iterate over $r \in G_P$

For $\kappa \in M(r)$  
\[ \nu_{\kappa \rightarrow r}(y_r) = \frac{1}{|M(r)|} \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c) - \sum_{c \in C(r) \cap \kappa \setminus r} \lambda_{c \rightarrow r}(y_c) + \sum_{p \in P(r)} \lambda_{r \rightarrow p}(y_r) - \frac{1}{|M(r)|} \sum_{\kappa \in M(r), p \in P(r)} \lambda_{r \rightarrow p}(y_r) \]

- Guaranteed to have same solution as original problem
- SW for general graphs up and running on all platforms and EC2
Distributed Convex BP Algorithm

[Schwing, Pollefeys, Hazan and Urtasun, CVPR11]

Algorithm: Distributed Message Passing Inference

Repeat until convergence

1. For every $\kappa$ in parallel: iterate over $r$:

   $\forall p \in P(r), y_r \quad \mu_{p \rightarrow r}(y_r) = \epsilon \hat{c}_p \ln \sum_{y_p \setminus y_r} \exp \left( \frac{\hat{\theta}_p(y_p) - \sum_{p' \in P(p)} \lambda_{p' \rightarrow p}(y_p) + \sum_{r' \in C(p) \cap \kappa \setminus r} \lambda_{r' \rightarrow p}(y_{r'}) + \nu_{\kappa \rightarrow p}(y_p)}{\epsilon \hat{c}_p} \right)$

2. Iterate over $r \in G_P$

   $\forall \kappa \in M(r) \quad \nu_{\kappa \rightarrow r}(y_r) = \frac{1}{|M(r)|} \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c) - \sum_{c \in C(r) \cap \kappa} \lambda_{c \rightarrow r}(y_c) + \sum_{p \in P(r)} \lambda_{r \rightarrow p}(y_r) - \frac{1}{|M(r)|} \sum_{\kappa \in M(r), p \in P(r)} \lambda_{r \rightarrow p}(y_r)$

- Guaranteed to have same solution as original problem
- SW for general graphs up and running on all platforms and EC2
Experiments on Stereo Task

Figure: Time vs iteration: distributed version (dotted magenta)

<table>
<thead>
<tr>
<th></th>
<th>50</th>
<th>25</th>
<th>10</th>
<th>5</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16.60</td>
<td>15.74</td>
<td>14.95</td>
<td>13.51</td>
<td>10.48</td>
<td>7.67</td>
</tr>
<tr>
<td>0.01</td>
<td>15.57</td>
<td>15.27</td>
<td>14.34</td>
<td>13.19</td>
<td>10.87</td>
<td>8.32</td>
</tr>
<tr>
<td>0.10</td>
<td>15.38</td>
<td>15.10</td>
<td>14.23</td>
<td>13.24</td>
<td>10.76</td>
<td>8.24</td>
</tr>
</tbody>
</table>

Table: Speedup for different $\epsilon$ (rows) exchange rates (columns).
Inference for continuous MRFs

We developed **particle convex-BP**

- Iterate sampling to discretize MRF and solving the discrete MRF
- Use our distributed convex BP to solve the discrete MRF

**Algorithm 1** PCBP for stereo estimation and occlusion boundary reasoning

```plaintext
Set N
Initialize slanted planes \( y_i^0 = (\alpha_i^0, \beta_i^0, \gamma_i^0) \) via local fitting \( \forall i \)
Initialize \( \sigma_\alpha, \sigma_\beta \) and \( \sigma_\gamma \)
for \( t = 1 \) to \#iters do
    Sample \( N \) times \( \forall i \) from \( \alpha_i \sim \mathcal{N}(\alpha_i^{t-1}, \sigma_\alpha), \beta_i \sim \mathcal{N}(\beta_i^{t-1}, \sigma_\beta), \gamma_i \sim \mathcal{N}(\gamma_i^{t-1}, \sigma_\gamma) \)
    \((o^t, y^t) \leftarrow \text{Solve the discretized MRF using convex BP}\)
    Update \( \sigma_\alpha^c = \sigma_\beta^c = 0.5 \times \exp(-c/10) \) and \( \sigma_\gamma^c = 5.0 \times \exp(-c/10) \)
end for
Return \( o^t, y^t \)
```

- Works very well in practice
Learning in Structured Problems

- We want to learn the parameters of our MRF, assuming a log-linear model

\[ \theta(x, y) = \sum_{r \in \mathcal{R}} w_r \phi_r(y_r) \]

- Regularized loss minimization: Given input pairs \((x, y) \in S\), minimize

\[ \min_w \sum_{(x, y) \in S} \hat{\ell}(w, x, y) + \frac{C}{p} \|w\|_p^p, \]

with \(\hat{\ell}(w, x, y)\) a surrogate, typically convex loss

- This is difficult in general because it involves exponentially many labels.
Learning in Structured Problems

- We want to learn the parameters of our MRF, assuming a log-linear model

\[ \theta(x, y) = \sum_{r \in \mathcal{R}} w_r \phi_r(y_r) \]

- Regularized loss minimization: Given input pairs \((x, y) \in S\), minimize

\[ \min_w \sum_{(x, y) \in S} \hat{\ell}(w, x, y) + \frac{C}{p} \| w \|_p^p, \]

with \( \hat{\ell}(w, x, y) \) a surrogate, typically convex loss

- This is difficult in general because it involves exponentially many labels.

Challenges:

1. **Scalable algorithms** to approximately solve the problem
2. Learn the structure of the model
3. Go beyond log-linear models

I’ll only talk about the first challenge
Learning in Structured Problems

- We want to learn the parameters of our MRF, assuming a log-linear model

\[ \theta(x, y) = \sum_{r \in R} w_r \phi_r(y_r) \]

- Regularized loss minimization: Given input pairs \((x, y) \in S\), minimize

\[ \min_w \sum_{(x, y) \in S} \hat{\ell}(w, x, y) + \frac{C}{p} \|w\|_p^p, \]

with \(\hat{\ell}(w, x, y)\) a surrogate, typically convex loss

- This is difficult in general because it involves exponentially many labels.

- Challenges:
  1. **Scalable algorithms** to approximately solve the problem
  2. Learn the structure of the model
  3. Go beyond log-linear models

- I’ll only talk about the first challenge
Loss functions

- Regularized loss minimization: Given input pairs \((x, y) \in S\), minimize

\[
\sum_{(x, y) \in S} \hat{\ell}(w, x, y) + \frac{C}{p} \|w\|_p^p,
\]

- In structured SVMs

\[
\bar{\ell}_{hinge}(w, x, y) = \max_{\hat{y} \in \mathcal{Y}} \{ \ell(y, \hat{y}) + w^\top \Phi(x, \hat{y}) - w^\top \Phi(x, y) \}
\]
Loss functions

- Regularized loss minimization: Given input pairs \((x, y) \in S\), minimize
  \[
  \sum_{(x, y) \in S} \hat{\ell}(w, x, y) + \frac{C}{p} \|w\|_p^p,
  \]

- In structured SVMs
  \[
  \bar{\ell}_{hinge}(w, x, y) = \max_{\hat{y} \in \mathcal{Y}} \{\ell(y, \hat{y}) + w^\top \Phi(x, \hat{y}) - w^\top \Phi(x, y)\}
  \]

- CRF loss: The conditional distribution is
  \[
  p_{x,y}(\hat{y}; w) = \frac{1}{Z(x, y)} \exp (\ell(y, \hat{y}) + w^\top \Phi(x, \hat{y}))
  \]

  \[
  Z(x, y) = \sum_{\hat{y} \in \mathcal{Y}} \exp (\ell(y, \hat{y}) + w^\top \Phi(x, \hat{y}))
  \]

  where \(\ell(y, \hat{y})\) is a prior distribution and \(Z(x, y)\) the partition function, and

  \[
  \bar{\ell}_{log}(w, x, y) = \ln \frac{1}{p_{x,y}(y; w)}.
  \]
Loss functions

- Regularized loss minimization: Given input pairs \((x, y) \in S\), minimize
\[
\sum_{(x, y) \in S} \hat{\ell}(w, x, y) + \frac{C}{p} \|w\|_p^p,
\]

- In structured SVMs
\[
\bar{\ell}_{\text{hinge}}(w, x, y) = \max_{\hat{y} \in \mathcal{Y}} \{ \ell(y, \hat{y}) + w^T \Phi(x, \hat{y}) - w^T \Phi(x, y) \}
\]

- CRF loss: The conditional distribution is
\[
pxy(\hat{y}; w) = \frac{1}{Z(x, y)} \exp \left( \ell(y, \hat{y}) + w^T \Phi(x, \hat{y}) \right)
\]
\[
Z(x, y) = \sum_{\hat{y} \in \mathcal{Y}} \exp \left( \ell(y, \hat{y}) + w^T \Phi(x, \hat{y}) \right)
\]
where \(\ell(y, \hat{y})\) is a prior distribution and \(Z(x, y)\) the partition function, and
\[
\bar{\ell}_{\text{log}}(w, x, y) = \ln \frac{1}{pxy(y; w)}.
\]
Relation between loss functions

The CRF program is

\[
\text{(CRF)} \quad \min_w \left\{ \sum_{(x,y) \in S} \ln Z(x, y) - d^T w + \frac{C}{p} \|w\|_p^p \right\},
\]

where \((x, y) \in S\) ranges over training pairs and \(d = \sum_{(x,y) \in S} \Phi(x, y)\) is the vector of "empirical means", and

\[
Z(x, y) = \sum_{\hat{y} \in Y} \exp \left( \ell(y, \hat{y}) + w^T \Phi(x, \hat{y}) \right)
\]

In structured SVMs

\[
\text{(structured SVM)} \quad \min_w \left\{ \sum_{(x,y) \in S} \max_{\hat{y} \in Y} \left\{ \ell(y, \hat{y}) + w^T \Phi(x, \hat{y}) \right\} - d^T w + \frac{C}{p} \|w\|_p^p \right\},
\]
Relation between loss functions

- The CRF program is

\[
\text{\text{CRF}} \quad \min_w \left\{ \sum_{(x,y) \in S} \ln Z(x, y) - d^\top w + \frac{C}{p} \|w\|_p^p \right\},
\]

where \((x, y) \in S\) ranges over training pairs and \(d = \sum_{(x,y) \in S} \Phi(x, y)\) is the vector of "empirical means", and

\[
Z(x, y) = \sum_{\hat{y} \in \mathcal{Y}} \exp \left( \ell(y, \hat{y}) + w^\top \Phi(x, \hat{y}) \right)
\]

- In structured SVMs

\[
\text{\text{structured SVM}} \quad \min_w \left\{ \sum_{(x,y) \in S} \max_{\hat{y} \in \mathcal{Y}} \left\{ \ell(y, \hat{y}) + w^\top \Phi(x, \hat{y}) \right\} - d^\top w + \frac{C}{p} \|w\|_p^p \right\},
\]
One parameter extension of CRFs and structured SVMs

\[
\min_w \left\{ \sum_{(x,y) \in S} \ln Z_\epsilon(x,y) - d^T w + \frac{C}{p} \| w \|^p_p \right\},
\]

\(d\) is the empirical means, and

\[
\ln Z_\epsilon(x,y) = \epsilon \ln \sum_{\hat{y} \in \mathcal{Y}} \exp \left( \frac{\ell(y, \hat{y}) + w^T \Phi(x, \hat{y})}{\epsilon} \right)
\]

CRF if \(\epsilon = 1\), Structured SVM if \(\epsilon = 0\) respectively.
A family of structured prediction problems

One parameter extension of CRFs and structured SVMs

\[
\min_w \left\{ \sum_{(x,y) \in S} \ln Z_\epsilon(x, y) - d^T w + \frac{C}{p} \|w\|^p_p \right\},
\]

\(d\) is the empirical means, and

\[
\ln Z_\epsilon(x, y) = \epsilon \ln \sum_{\hat{y} \in \mathcal{Y}} \exp \left( \frac{\ell(y, \hat{y}) + w^T \Phi(x, \hat{y})}{\epsilon} \right)
\]

- CRF if \(\epsilon = 1\), Structured SVM if \(\epsilon = 0\) respectively.

- Introduces the notion of loss in CRFs.
A family of structured prediction problems

[T. Hazan and R. Urtasun, NIPS 2010]

- One parameter extension of CRFs and structured SVMs

\[
\min_w \left\{ \sum_{(x,y) \in S} \ln Z_\epsilon(x,y) - d^T w + \frac{C}{p} \|w\|_p^p \right\},
\]

- \(d\) is the empirical means, and

\[
\ln Z_\epsilon(x,y) = \epsilon \ln \sum_{\hat{y} \in \mathcal{Y}} \exp \left( \frac{\ell(y, \hat{y}) + w^T \Phi(x, \hat{y})}{\epsilon} \right)
\]

- CRF if \(\epsilon = 1\), Structured SVM if \(\epsilon = 0\) respectively.

- Introduces the notion of loss in CRFs.

- One can compute the dual, do approximations and compute the dual again to get an approximate primal.

- Then we can derive a block coordinate descent algorithm
A family of structured prediction problems

[T. Hazan and R. Urtasun, NIPS 2010]

- One parameter extension of CRFs and structured SVMs

\[
\min_w \left\{ \sum_{(x,y) \in S} \ln Z_\epsilon(x, y) - d^T w + \frac{C}{p} \|w\|^p \right\},
\]

\( d \) is the empirical means, and

\[
\ln Z_\epsilon(x, y) = \epsilon \ln \sum_{\hat{y} \in Y} \exp \left( \frac{\ell(y, \hat{y}) + w^T \Phi(x, \hat{y})}{\epsilon} \right)
\]

- CRF if \( \epsilon = 1 \), Structured SVM if \( \epsilon = 0 \) respectively.
- Introduces the notion of loss in CRFs.
- One can compute the dual, do approximations and compute the dual again to get an approximate primal.
- Then we can derive a block coordinate descent algorithm.
Convex Structured Prediction

Algorithm: Convex Structured Prediction
Repeat until convergence

1. Iterate over all samples \((x, y)\) and regions \(r\):
   \[
   \begin{align*}
   \forall p \in P(r), y_r & \quad \mu_{(x,y),p \rightarrow r(y_r)} = \epsilon_c p \ln \sum_{y_p \setminus y_r} \exp \left( \phi_{(x,y),p(y_p)} - \sum_{p' \in P(p)} \lambda_{(x,y),p \rightarrow p'(y_p)} + \frac{\sum_{r' \in C(p) \setminus r} \lambda_{(x,y),r' \rightarrow p(y_{r'})}}{c_p} \right) \\
   \forall p \in P(r), y_r & \quad \lambda_{(x,y),r \rightarrow p(y_r)} \propto \frac{c_p}{c_r + \sum_{p \in P(r)} c_p} \left( \phi_{(x,y),r(y_r)} + \sum_{c \in C(r)} \lambda_{(x,y),c \rightarrow r(y_c)} + \sum_{p \in P(r)} \mu_{(x,y),p \rightarrow r(y_r)} \right) - \mu_{(x,y),p \rightarrow r(y_r)}
   \end{align*}
   \]

2. Weight vector update with stepsize \(\alpha\) determined via Armijo iterations for \(p = 2\):
   \[
   w \leftarrow w - \alpha \left( \sum_{(x,s),r} \left( \sum_{\hat{s}_r} b_{(x,s),r(\hat{s}_r)} \phi_{(x,s),r(\hat{s}_r)} - \phi_{(x,s),r} \right) + Cw \right)
   \]

- This is a message passing algorithm that blends learning and inference
- Distributed version which distributes memory and computation
- SW for general graphs running on all platforms and EC2 very soon
Algorithm: Convex Structured Prediction

Repeat until convergence

1. Iterate over all samples \((x, y)\) and regions \(r\):

\[
\forall p \in P(r), y_r \quad \mu(x, y), p \rightarrow r(y_r) = \epsilon c_p \ln \sum_{y_p \setminus y_r} \exp \frac{\tilde{\phi}(x, y), p(y_p) - \sum_{p' \in P(p)} \lambda(x, y), p \rightarrow p'(y_p) + \sum_{r' \in C(p) \setminus r} \lambda(x, y), r' \rightarrow p(y_{r'})}{\epsilon c_p}
\]

\[
\forall p \in P(r), y_r \quad \lambda(x, y), r \rightarrow p(y_r) \propto \frac{c_p}{c_r + \sum_{p \in P(r)} c_p} \left( \tilde{\phi}(x, y), r(y_r) + \sum_{c \in C(r)} \lambda(x, y), c \rightarrow r(y_c) + \sum_{p \in P(r)} \mu(x, y), p \rightarrow r(y_r) \right) - \mu(x, y), p \rightarrow r(y_r)
\]

2. Weight vector update with stepsize \(\alpha\) determined via Armijo iterations for \(p = 2\)

\[
w \leftarrow w - \alpha \left( \sum_{(x, s), r} \left( \sum_{\hat{s}_r} b(x, s), r, \hat{s}_r \tilde{\phi}(x, s), r, \hat{s}_r - \phi(x, s), r \right) + Cw \right)
\]

- This is a message passing algorithm that blends learning and inference
- Distributed version which distributes memory and computation
- SW for general graphs running on all platforms and EC2 very soon
Experiments on Denoising Task

[Schwing, T. Hazan, M. Pollefeys and R. Urtasun, To appear hopefully soon]

\( \epsilon = 0 \)

\( \epsilon = 0.1 \)

\( \epsilon = 1.0 \)

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figures.png}
\caption{Time vs Iteration: non-distributed version (dotted magenta)}
\end{figure}

\begin{table}
\begin{tabular}{|c|cccccc|}
\hline
 & 25 & 10 & 5 & 2 & 1 \\
\hline
0 & 15.74 & 14.84 & 13.41 & 10.62 & 7.92 \\
0.01 & 12.81 & 14.61 & 13.44 & 10.72 & 8.13 \\
0.10 & 15.17 & 14.38 & 13.14 & 10.59 & 8.02 \\
1.00 & 15.09 & 14.02 & 12.69 & 9.74 & 7.14 \\
\hline
\end{tabular}
\caption{Speedup for different \( \epsilon \) (rows) and information exchange rates (columns).}
\end{table}
Contents

1) Data

Stereo Camera Rig

Monochrome, Color

2) Learning and Inference

3) Sensing the environment

4) Recognition

R. Urtasun (TTIC)
Sense the Environment

- Goal: given 2 cameras mounted on top of the car, reconstruct the environment in 3D.
Slanted-Plane MRFs

- Slanted-plane MRF with explicit occlusion handling
- Start with an over-segmentation of the image into superpixels
- MRF on continuous variables (slanted planes) and discrete var. (boundary)

Takes as input disparities computed by any local algorithm
Energy of PCB-P-Stereo

- \( y \) the set of slanted 3D planes, \( o \) the set of discrete boundary variables

\[
E(y, o) = E_{\text{color}}(o) + E_{\text{match}}(y, o) + E_{\text{compatibility}}(y, o) + E_{\text{junction}}(o)
\]
Energy of PCBP-Stereo

- $y$ the set of slanted 3D planes, $o$ the set of discrete boundary variables

$$E(y, o) = E_{\text{color}}(o) + E_{\text{match}}(y, o) + E_{\text{compatibility}}(y, o) + E_{\text{junction}}(o)$$

Agreement with result of input disparity map

Computed by any matching method (Modified semi-global matching)

Truncated quadratic function

$$\phi_i^{TP}(p, y, K) = \min \left( \left| D(p) - \hat{d}_i(p, y_i) \right|, K \right)^2$$

Disparity map  Slanted plane

On boundary

“Oclusion” – Foreground segment owns boundary
Energy of PCBP-Stereo

- $y$ the set of slanted 3D planes, $o$ the set of discrete boundary variables

$$E(y, o) = E_{\text{color}}(o) + E_{\text{match}}(y, o) + E_{\text{compatibility}}(y, o) + E_{\text{junction}}(o)$$

1. Preference of boundary label (Coplanar > Hinge > Occlusion)
   - Impose penalty $\lambda_{\text{occ}} > \lambda_{\text{hinge}} > 0$

2. Boundary labels match Slanted planes
   - "Occlusion" $\hat{d}_{\text{front}}(p) > \hat{d}_{\text{back}}(p)$
   - "Hinge" $\hat{d}_i(p) = \hat{d}_j(p)$ on boundary
   - "Coplanar" $\hat{d}_i(p) = \hat{d}_j(p)$ in both segments
Energy of PCBP-Stereo

- $y$ the set of slanted 3D planes, $o$ the set of discrete boundary variables

$$E(y, o) = E_{\text{color}}(o) + E_{\text{match}}(y, o) + E_{\text{compatibility}}(y, o) + E_{\text{junction}}(o)$$

Occlusion boundary reasoning \cite{Malik 1987}
Penalize impossible junctions

Impossible cases

Front
Back
Occlusion
Hinge
Coplanar
Stereo Evaluation

Comparison on test set of KITTI dataset

- GC+occ [Kolmogorov, et al. 2001] 33.50%
- OCV-BM [Bradski 2000] 25.39%
- CostFilter [Rhemann, et al. 2007] 19.96%
- GCS [Cech, et al. 2007] 13.37%
- GCSF [Cech, et al. 2011] 12.06%
- SDM [Kostkova 2003] 10.98%
- OCV-SGBM [Hirschmueller 2008] 7.64%
- Ours 4.13%

Error > 3 pixels (Non-occluded)
Stereo Evaluation

[K. Yamaguchi, T. Hazan, D. McAllester and R. Urtasun, ECCV12]

Easy Scenarios:

- Natural scenes, lots of texture, no objects
- A couple of errors at thin structures (poles)

Errors: < 0.5%

Errors: < 0.5%
Easy Scenarios:

- Shadows help the disambiguation process
- Errors at thin structures and far away textureless regions

Errors: < 0.5%
Stereo Evaluation

[K. Yamaguchi, T. Hazan, D. McAllester and R. Urtasun, ECCV12]

**Hard Scenarios:**

- Textureless or saturated areas
- Ambiguous reflections

Errors: 22.1%

Errors: 17.4%
Sense the Environment

- Depth is not all!
- Recover the motion of the scene given a single camera

[K. Yamaguchi, D. McAllester and R. Urtasun, CVPR 2013]
Comparison on the test set of KITTI
(At the time of submission)

<table>
<thead>
<tr>
<th>Method</th>
<th>General Flow Error</th>
<th>Epipolar Flow Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>C+NL [Sun, et al. 2010]</td>
<td>24.64%</td>
<td></td>
</tr>
<tr>
<td>HS [Horn, et al. 1993]</td>
<td>19.92%</td>
<td></td>
</tr>
<tr>
<td>TGV2CENSUS [Werberger 2012]</td>
<td>11.14%</td>
<td></td>
</tr>
<tr>
<td>fSGM [Hermann, et al. 2012]</td>
<td>11.03%</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>4.08%</td>
<td></td>
</tr>
</tbody>
</table>

Error > 3 pixels (Non-Occluded)
Optical Flow Evaluation

Comparison on the test set of KITTI (Current)

- fSGM [Hermann, et al. 2012]: 11.03%
- C+NL [Sun, et al. 2013]: 10.60%
- C++ [Sun, et al. 2013]: 10.16%
- CRT-Flow [Anonymous]: 9.71%
- MLDP-OF [Anonymous]: 8.91%
- TVL1-HOG [Anonymous]: 8.31%
- Data-Flow [Anonymous]: 8.22%
- TGV2ADCSIFT [Anonymous]: 6.55%
- Ours: 4.08%

Error > 3 pixels (Non-Occluded)
What about the indoor scenario?
Task: Given an image, predict the 3D parametric cuboid that best describes the layout.

\[ \hat{y} = \arg \max_y \mathbf{w}^T \phi(x, y) \]

with \( \phi(x, y) \) potentials based on image features.

- First exact solution to this problem: great performance and much faster than the state of the art.
- We employ branch and bound towards this goal.
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**
We parameterize a layout with 4 variables \( y_i \in \mathcal{Y}, \ i \in \{1, \ldots, 4\} \) [Lee et al. 09]

Layout fully specified if the VPs are known and the world is Manhattan.

We parameterize the sets by intervals of minimum and maximum angles

\[
\{[y_1^{\text{min}}, y_1^{\text{max}}], \ldots, [y_4^{\text{min}}, y_4^{\text{max}}]\}
\]
Scoring function

- Features count frequencies of image cues in each face.

\[ \mathbf{w}^T \cdot \phi(x, y) = \sum_{\alpha \in \mathcal{F}} \mathbf{w}_{o,\alpha}^T \phi_{o,\alpha}(x, y_\alpha) + \sum_{\alpha \in \mathcal{F}} \mathbf{w}_{g,\alpha}^T \phi_{g,\alpha}(x, y_\alpha) \]

with \( \mathcal{F} = \{ \text{left-wall, right-wall, ceiling, floor, front-wall} \} \)

(Orientation map [Lee et al. 09]) (Geometric Context [Hoiem et al. 05])

- Faces are defined by four (front-wall) or three angles (otherwise), so we have to deal with high order potentials, e.g., \( 50^4 \) for fourth-order.

- I’ll show you in a few slides that it’s not the case!
Deriving bounds

- Decompose potentials into those that have positive and negative weights
  \[ \hat{y} = \arg\max_{y \in Y} f^+(x, y) + f^-(x, y), \]

- \( f^+ \) and \( f^- \) decompose into weighted sums over the different faces
  \[ \overline{f}(\hat{Y}) = \sum_{\alpha \in F} \left( \overline{f}^+_{\alpha}(\hat{Y}) + \overline{f}^-_{\alpha}(\hat{Y}) \right) \]

- Bounds are max positive and min negative contributions for each face
  \[ \overline{f}_{\text{front-wall}}(\hat{Y}) = f^+_{\text{front-wall}}(x, y_{up}) + f^-_{\text{front-wall}}(x, y_{low}), \]

(Front Wall) (Minimal left wall) (Maximal left wall)
Trick: Integral Geometry

- To be practical, we need to be able to compute the bounds very efficiently.
- Same concept as integral images, but in accordance with the VP.
- Now the potentials can be computed in constant time by accessing some accumulators.
Trick: Integral Geometry

- To be practical, we need to be able to compute the bounds very efficiently.
- Same concept as integral images, but in accordance with the VP.
- Now the potentials can be computed in constant time by accessing some accumulators.

\[ \phi_{i,j,k}(x, y_i, y_j, y_k) = H_{i,j}(x, y_i, y_j) - H_{j,k}(x, y_j, y_k) \]
Table: Pixel classification error in the layout dataset of [Hedau et al. 09].

<table>
<thead>
<tr>
<th>Method</th>
<th>OM</th>
<th>GC</th>
<th>OM + GC</th>
<th>Other</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Hoiem07]</td>
<td></td>
<td>28.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[Hedau09] (a)</td>
<td></td>
<td>26.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[Hedau09] (b)</td>
<td></td>
<td>21.2</td>
<td></td>
<td></td>
<td>10-30 min</td>
</tr>
<tr>
<td>[Wang10]</td>
<td>22.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[Lee10]</td>
<td>24.7</td>
<td>22.7</td>
<td>18.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[delPero11]</td>
<td></td>
<td></td>
<td></td>
<td>16.3</td>
<td>12 min</td>
</tr>
<tr>
<td>Ours</td>
<td>18.6</td>
<td>15.4</td>
<td>13.6</td>
<td></td>
<td>0.007s</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>[delPero11]</th>
<th>[Hoiem07]</th>
<th><a href="a">Hedau09</a></th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o box</td>
<td>29.59</td>
<td>23.04</td>
<td>22.94</td>
<td>16.46</td>
</tr>
</tbody>
</table>

- Takes on average **0.007s** for exact solution over $50^4$ possibilities!
- It’s 6 orders of magnitude faster!
Qualitative Results

[Images of qualitative results showing different scenes with color-coded segmentation, likely indicating different objects or categories within the scenes.]

R. Urtasun (TTIC)  Holistic Models  June 24, 2013  45 / 72
Contents

1) Data

2) Learning and Inference

3) Sensing the environment

4) Recognition
For an image we would like to reason about:

- **Objects**: which class, where, how many?
- **Segmentation**: which semantic label does each pixel take?
- **Scene classification**: which scene am I looking at?
- **High-order semantics**: e.g., traffic patterns at intersections
Can we leverage segmentation to improve detection?
DPM is arguably state-of-the-art in object detection [Felzenszwalb et al. 10]

Let $p_0$ be the root, and $\{p_i\}_{i=1}^P$ the parts, the score of a detection

$$E(p) = \sum_{i=0}^P w_i^T \cdot \phi(x, p_i) + \sum_{i=1}^P w_{i,\text{def}}^T \cdot \phi(x, p_0, p_i)$$

with $x$ an image.
New deformable part-based model which exploits **bottom-up grouping**.

Let $h = \{0, \cdots, H(x)\}$ be the index over the set of candidate segments, with $h = 0$ implies that no segment is selected.

Allows every detection hypothesis to **select a segment** (including void), and scores each box in the image using both HOG filters as well as novel segmentation features.

$$E(p, h) = \sum_{i=0}^{P} w_i^T \cdot \phi(x, p_i) + \sum_{i=1}^{P} w_{i, \text{def}}^T \cdot \phi(x, p_0, p_i) + w_{\text{seg}}^T \phi(x, h, p_0)$$

with $x$ an image
New deformable part-based model which exploits **bottom-up grouping**.

Let \( h = \{0, \cdots, H(x)\} \) be the index over the set of candidate segments, with \( h = 0 \) implies that no segment is selected.

Allows every detection hypothesis to **select a segment** (including void), and scores each box in the image using both HOG filters as well as novel segmentation features.

\[
E(p, h) = \sum_{i=0}^{P} w_i^T \cdot \phi(x, p_i) + \sum_{i=1}^{P} w_{i, \text{def}}^T \cdot \phi(x, p_0, p_i) + w_{\text{seg}}^T \phi(x, h, p_0)
\]

with \( x \) an image

- Exact inference is possible as it forms a tree
- Our features can be computed **very efficiently** given the segments, and we thus maintain the same complexity as the original DPM.
New deformable part-based model which exploits **bottom-up grouping**.

Let \( h = \{0, \cdots, H(x)\} \) be the index over the set of candidate segments, with \( h = 0 \) implies that no segment is selected.

Allows every detection hypothesis to **select a segment** (including void), and scores each box in the image using both HOG filters as well as novel segmentation features.

\[
E(p, h) = \sum_{i=0}^{P} w_i^T \cdot \phi(x, p_i) + \sum_{i=1}^{P} w_{i,\text{def}}^T \cdot \phi(x, p_0, p_i) + w_{\text{seg}}^T \phi(x, h, p_0)
\]

with \( x \) an image

Exact inference is possible as it forms a tree

Our features can be computed **very efficiently** given the segments, and we thus maintain the same complexity as the original DPM.
### Segmentation Features

- $\phi_{\text{seg-in}}(x, h, p_0)$ and $\phi_{\text{seg-out}}(x, h, p_0)$ encourage the box to contain as many segment pixels as possible.

- $\phi_{\text{back-in}}(x, h, p_0)$ and $\phi_{\text{back-out}}(x, h, p_0)$ try to minimize the number of background pixels inside the box and maximize its number outside.

- Together they try to fit a tight bounding box around the object.

- Efficient computation of the features can be done via a **single integral image** per segment.
Results on validation set of PASCAL VOC 2010

[S. Fidler, R. Mottaghi, A. Yuille and R. Urtasun, CVPR13]

- **No parts**: segDPM outperforms Dalal & Triggs detector on all classes, and achieves 14% higher average AP

<table>
<thead>
<tr>
<th>wo parts</th>
<th>plane</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>motor</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dalal</td>
<td>29.1</td>
<td>36.9</td>
<td>2.9</td>
<td>3.4</td>
<td>15.6</td>
<td>47.1</td>
<td>27.1</td>
<td>11.4</td>
<td>9.8</td>
<td>5.8</td>
<td>6.0</td>
<td>5.0</td>
<td>24.8</td>
<td>28.4</td>
<td>27.5</td>
<td>2.2</td>
<td>18.4</td>
<td>9.2</td>
<td>27.4</td>
<td>23.2</td>
<td>18.1</td>
</tr>
<tr>
<td>CPMC</td>
<td>49.9</td>
<td>15.5</td>
<td>18.5</td>
<td>14.7</td>
<td>7.4</td>
<td>35.0</td>
<td>19.9</td>
<td>41.4</td>
<td>3.9</td>
<td>16.2</td>
<td>8.5</td>
<td>24.4</td>
<td>26.0</td>
<td>32.1</td>
<td>18.9</td>
<td>5.7</td>
<td>15.3</td>
<td>14.1</td>
<td>29.8</td>
<td>18.7</td>
<td>20.8</td>
</tr>
<tr>
<td>segDPM</td>
<td>51.9</td>
<td>43.3</td>
<td>23.5</td>
<td>18.8</td>
<td>21.6</td>
<td>55.3</td>
<td>44.9</td>
<td>14.7</td>
<td>24.2</td>
<td>17.7</td>
<td>33.6</td>
<td>41.6</td>
<td>44.7</td>
<td>35.2</td>
<td>14.9</td>
<td>28.8</td>
<td>24.0</td>
<td>43.6</td>
<td>33.6</td>
<td>32.5</td>
<td></td>
</tr>
</tbody>
</table>

- **With parts**: segDPM outperforms the original DPM in 19 out of 20 classes and achieves an improvement of 9% AP
Results on validation set of PASCAL VOC 2010

[S. Fidler, R. Mottaghi, A. Yuille and R. Urtasun, CVPR13]

- **No parts**: segDPM outperforms Dalal & Triggs detector on all classes, and achieves 14% higher average AP

<table>
<thead>
<tr>
<th></th>
<th>plane</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>motor</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dalal</td>
<td>29.1</td>
<td>36.9</td>
<td>2.9</td>
<td>3.4</td>
<td>15.6</td>
<td>47.1</td>
<td>27.1</td>
<td>11.4</td>
<td>9.8</td>
<td>5.8</td>
<td>6.0</td>
<td>5.0</td>
<td>24.8</td>
<td>28.4</td>
<td>27.5</td>
<td>2.2</td>
<td>18.4</td>
<td>9.2</td>
<td>27.4</td>
<td>23.2</td>
<td>18.1</td>
</tr>
<tr>
<td>CPMC</td>
<td>49.9</td>
<td>15.5</td>
<td>18.5</td>
<td>14.7</td>
<td>7.4</td>
<td>35.0</td>
<td>19.9</td>
<td>41.4</td>
<td>3.9</td>
<td>16.2</td>
<td>8.5</td>
<td>24.4</td>
<td>26.0</td>
<td>32.1</td>
<td>18.9</td>
<td>5.7</td>
<td>15.3</td>
<td>14.1</td>
<td>29.8</td>
<td>18.7</td>
<td>20.8</td>
</tr>
<tr>
<td>segDPM</td>
<td>51.9</td>
<td>43.3</td>
<td>23.5</td>
<td>18.8</td>
<td>21.6</td>
<td>55.3</td>
<td>34.0</td>
<td>44.9</td>
<td>14.7</td>
<td>24.2</td>
<td>17.7</td>
<td>33.6</td>
<td>41.6</td>
<td>44.7</td>
<td>35.2</td>
<td>14.9</td>
<td>28.8</td>
<td>24.0</td>
<td>43.6</td>
<td>33.6</td>
<td>32.5</td>
</tr>
</tbody>
</table>

- **With parts**: segDPM outperforms the original DPM in 19 out of 20 classes and achieves an improvement of 9% AP.

<table>
<thead>
<tr>
<th></th>
<th>plane</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>motor</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM</td>
<td>46.3</td>
<td>49.5</td>
<td>4.8</td>
<td>6.4</td>
<td>22.6</td>
<td>53.5</td>
<td>38.7</td>
<td>24.8</td>
<td>14.2</td>
<td>10.5</td>
<td>10.9</td>
<td>12.9</td>
<td>36.4</td>
<td>38.7</td>
<td>42.6</td>
<td>3.6</td>
<td>26.9</td>
<td>22.7</td>
<td>34.2</td>
<td>31.2</td>
<td>26.6</td>
</tr>
<tr>
<td>segDPM</td>
<td>55.8</td>
<td>49.7</td>
<td>25</td>
<td>18.8</td>
<td>29.4</td>
<td>57.1</td>
<td>42.7</td>
<td>49.6</td>
<td>17.9</td>
<td>25.8</td>
<td>19.6</td>
<td>34.4</td>
<td>43.4</td>
<td>47.8</td>
<td>42.5</td>
<td>15.7</td>
<td>33.1</td>
<td>27.3</td>
<td>44.2</td>
<td>40.9</td>
<td>36.0</td>
</tr>
</tbody>
</table>
Results on Test PASCAL VOC 2010

[S. Fidler, R. Mottaghi, A. Yuille and R. Urtasun, CVPR13]
Can detection help segmentation?
Let’s use a classifier for each task independently. What’s in the patch?

- detector: *bird*
- seg classif.: *water*
- scene: *boat*
Why Holistic?

Let’s use a classifier for each task independently. What’s in the patch?

- detector: *bird*
- seg classif.: *water*
- scene: *boat*
Why Holistic?

Let’s use a classifier for each task independently. What’s in the patch?

- detector: *bird*
- seg classif.: *water*
- scene: *boat*

![Image of a boat with annotations showing ground truth, segmentation, detection, and scene types.](image-url)
Why Holistic?

Let’s use a classifier for each task independently. What’s in the patch?

- detector: *bird*
- seg classif.: *water*
- scene: *boat*
Holistic Scene Understanding

We want to reason about the scene as a **whole**.

- Joint inference of scene type, class presence, objects and segmentation
- Efficient learning and inference with home-made structured prediction
Compact Holistic Model

- \( s \in \{1, \ldots, S\} \): scene type label
- \( z_k \in \{0, 1\} \): presence of class \( k \)
- \( b_l \in \{0, 1\} \): object det. \( l \) is on/off
- \( y_j \in \{1, \ldots, C\} \): class \( j \)-th super-seg
- \( x_i \in \{1, \ldots, C\} \): class \( i \)-th super-pixel

**Unary potentials for each task**: scene, segmentation, detection and presence/absence of a class
Compact Holistic Model

- $s \in \{1, \ldots, S\}$: scene type label
- $z_k \in \{0, 1\}$: presence of class $k$
- $b_l \in \{0, 1\}$: object det. $l$ is on/off
- $y_j \in \{1, \ldots, C\}$: class $j$-th super-seg
- $x_i \in \{1, \ldots, C\}$: class $i$-th super-pixel

**Long range dependencies** via Pn Potentials [Kohli et al 08]
Compact Holistic Model

- $s \in \{1, \ldots, S\}$: scene type label

- $z_k \in \{0, 1\}$: presence of class $k$

- $b_l \in \{0, 1\}$: object det. $l$ is on/off

- $y_j \in \{1, \ldots, C\}$: class $j$-th super-seg

- $x_i \in \{1, \ldots, C\}$: class $i$-th super-pixel

Compatibility between class presence/absence and segmentation via Pn Potentials [Kohli et al 08]
Compact Holistic Model

- $s \in \{1, \ldots, S\}$: scene type label
- $z_k \in \{0, 1\}$: presence of class $k$
- $b_l \in \{0, 1\}$: object det. $l$ is on/off
- $y_j \in \{1, \ldots, C\}$: class j-th super-seg
- $x_i \in \{1, \ldots, C\}$: class i-th super-pixel

**Co-occurance** potentials
Compact Holistic Model

- $s \in \{1, \ldots, S\}$: scene type label
- $z_k \in \{0, 1\}$: presence of class $k$
- $b_l \in \{0, 1\}$: object det. $l$ is on/off
- $y_j \in \{1, \ldots, C\}$: class $j$-th super-seg
- $x_i \in \{1, \ldots, C\}$: class $i$-th super-pixel

Compatibility between detection and class absence/presence via Pn Potentials [Kohli et al 08]
Compact Holistic Model

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

$ z_k \in \{0, 1\} $: presence of class $ k $

$ b_l \in \{0, 1\} $: object det. $ l $ is on/off

$ y_j \in \{1, \ldots, C\} $: class $ j $-th super-seg

$ x_i \in \{1, \ldots, C\} $: class $ i $-th super-pixel

**Shape prior** inside the bounding box linking detection and segmentation

aeroplane  |  chair  |  car, component 1  |  car, cmp. 3  |  bicycle  |  bird  |  cow  |  flower
Compact Holistic Model

- $s \in \{1, \ldots, S\}$: scene type label
- $z_k \in \{0, 1\}$: presence of class $k$
- $b_l \in \{0, 1\}$: object det. $l$ is on/off
- $y_j \in \{1, \ldots, C\}$: class $j$-th super-seg
- $x_i \in \{1, \ldots, C\}$: class $i$-th super-pixel

**Scene context** potentials encoding statistics of each type of classes present for each type of scene
Really fast and accurate!

Figure: Segmentation accuracy on MSRC as a function of (left) training time, (right) inference time.

- State of the art achieve with < 1 min training and 0.02 s for inference!
- Holistic helps segmentation, scene classification and detection
Figure: Segmentation examples: (image, groundtruth, our holistic scene model)

Figure: Examples of failure modes.
Images do not appear in isolation
Images and text appear everywhere, e.g., web, human-robot interactions

History

The founding of a colonial college had long been the desire of John Graves Simcoe, the first Lieutenant-Governor of Upper Canada. As an Oxford-educated military commander who had fought in the American Revolutionary War, Simcoe believed a college was needed to counter the spread of republicanism from the United States. The Upper Canada Executive Committee recommended in 1798 that a college be established in York, the colonial capital.

On March 15, 1827, a royal charter was formally issued by King George IV, proclaiming "from this time one College, with the style and privileges of an University ... for the education of youth in the principles of the Christian Religion, and for their instruction in the various branches of Science and Literature ... to continue for ever, to be called King's College." The granting of the charter was largely the result of intense lobbying by John Strachan, the influential Anglican Bishop of Toronto who took office as the first president of the college. The original three-storey Greek Revival school building was constructed on the present site of Queen's Park.

Under Strachan's guidance, King's College was a religious institution that closely aligned with the Church of England and the British colonial elite, known as the Family Compact. Reformist politicians opposed the clergy's control over colonial institutions and fought to have the college secularized. In 1842, after a lengthy and heated debate, the newly-elected responsible government of Upper Canada voted to rename King's College as the University of Toronto and severed the school's ties with the church. Having anticipated this decision, the enraged Strachan had resigned a year earlier to open Trinity College as a private Anglican seminary. University College was created as the non-denominational teaching branch of the University of Toronto. During the American Civil War, the threat of Union blockade on British North America prompted the creation of the University Rifle Corps, which saw battle in resisting the Fenian raids on the Niagara border in 1866.
Images and text appear everywhere, e.g., web, human-robot interactions

That’s not my mug. My mug is in the kitchen next to the pile of dishes you should clean.
What to do with Images and Text?

Current approaches

- Generate sentences given an image
- Learn a common latent space between images and tags
- Sense disambiguation, e.g., mouse

Use the sentential descriptions to better do visual holistic scene understanding

- Parse syntactically and semantically the sentences
- Use it in our holistic model to better parse the visual scene
What to do with Images and Text?

Current approaches

- Generate sentences given an image
- Learn a common latent space between images and tags
- Sense disambiguation, e.g., mouse

Use the sentential descriptions to better do **visual holistic scene understanding**

- Parse syntactically and semantically the sentences
- Use it in our holistic model to better parse the visual scene

Challenges

- People don’t talk about everything in the image
- People describe things not visually relevant
- NLP is not solved
What to do with Images and Text?

Current approaches

- Generate sentences given an image
- Learn a common latent space between images and taggs
- Sense disambiguation, e.g., mouse

Use the sentential descriptions to better do **visual holistic scene understanding**

- Parse syntactically and semantically the sentences
- Use it in our holistic model to better parse the visual scene

Challenges

- People don’t talk about everything in the image
- People describe things not visually relevant
- NLP is not solved
What do we need to do?

Original image
Ground truth
Holistic visual
What do we need to do?

sentence:  A black dog walking in a field next to two sheep
What do we need to do?

Original image | Ground truth | Holistic visual

**sentence:** A black dog walking in a field next to two sheep
What do we need to do?

sentence: A black dog walking in a field next to two sheep
What do we need to do?

Original image  | Ground truth  | Holistic visual

**sentence:**  A black dog walking in a field next to two sheep
What do we need to do?

sentence: A black dog walking in a field next to two sheep
What do we need to do?

sentence: A black dog walking in a field next to two sheep

Extract Attributes

Original image: A black dog walking in a field next to two sheep

Ground truth: A black dog walking in a field next to two sheep

Holistic visual: A black dog walking in a field next to two sheep
What do we need to do?

Original image

Ground truth

Holistic visual

sentence: A black dog walking in a field next to two sheep
How do we do it?

Original image  Ground truth  Holistic visual

Class presence  Segm level 2  Segm level 1  scene  detection
How do we do it?

<table>
<thead>
<tr>
<th>Original image</th>
<th>Ground truth</th>
<th>Holistic visual</th>
<th>visual + text</th>
</tr>
</thead>
</table>

Scene from text: BOW, nouns and semantic class

Class presence from mentioned/non-mentioned

cardinality

detection

Detection from nouns and prepositions
Quantitative Results

<table>
<thead>
<tr>
<th>Textonboost</th>
<th>back</th>
<th>aerop.</th>
<th>bicycle</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>dtable</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>person</th>
<th>pplant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>monitor</th>
<th>averg.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>77.8</td>
<td>14.1</td>
<td>3.4</td>
<td>0.7</td>
<td>11.3</td>
<td>3.3</td>
<td>25.5</td>
<td>30.9</td>
<td>10.3</td>
<td>0.7</td>
<td>13.2</td>
<td>10.8</td>
<td>5.2</td>
<td>15.1</td>
<td>31.8</td>
<td>41.0</td>
<td>0.0</td>
<td>3.7</td>
<td>2.4</td>
<td>17.1</td>
<td>33.7</td>
<td>16.8</td>
</tr>
<tr>
<td>[Yao12]</td>
<td>77.3</td>
<td>25.6</td>
<td>12.9</td>
<td>14.2</td>
<td>19.2</td>
<td>31.0</td>
<td>34.6</td>
<td>38.6</td>
<td>16.1</td>
<td>7.4</td>
<td>11.9</td>
<td>9.0</td>
<td>13.9</td>
<td>25.4</td>
<td>31.7</td>
<td>38.1</td>
<td>11.2</td>
<td>18.8</td>
<td>6.2</td>
<td>23.6</td>
<td>34.4</td>
<td>23.9</td>
</tr>
<tr>
<td>ours</td>
<td>76.9</td>
<td>31.3</td>
<td>29.7</td>
<td>37.3</td>
<td>27.7</td>
<td>29.5</td>
<td>52.1</td>
<td>40.0</td>
<td>38.0</td>
<td>6.6</td>
<td>55.9</td>
<td>25.2</td>
<td>33.2</td>
<td>38.2</td>
<td>44.3</td>
<td>42.5</td>
<td>15.2</td>
<td>32.0</td>
<td>20.2</td>
<td>40.7</td>
<td>48.4</td>
<td>36.4</td>
</tr>
</tbody>
</table>

**Table:** Comparison to baselines in the UIUC sentence dataset (subset of PASCAL VOC 2008). By leveraging text information our approach improves 13% AP.

<table>
<thead>
<tr>
<th># train</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>all</th>
<th>all</th>
<th>all</th>
<th>all</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td># test</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>all</td>
</tr>
<tr>
<td>T-DPM</td>
<td>35.6</td>
<td>36.1</td>
<td>36.3</td>
<td>36.7</td>
<td>35.6</td>
<td>36.3</td>
<td>36.4</td>
<td>36.6</td>
<td>36.6</td>
</tr>
<tr>
<td>DPM</td>
<td></td>
<td>31.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Detection AP (%) improves 5% over DPM when employing text
Acknowledgements

- Alex Schwing
- Tamir Hazan
- Sanja Fidler
- David McAllester
- Roozbeh Mottaghi
- Jian Peng
- Abhishek Sharma
- Koichiro Yamaguchi
- Jian Yao
- Alan Yuille
- Marc Pollefeys
Conclusions

I've shown you examples of how to

- Sense the environment: stereo, flow, layout estimation
- Recognize the 3D world: detection, segmentation
- Interact with it

I advocate for **MRF holistic models** which require

- Representation: decomposition and exact inference
- Learning
- Inference
- Data

Important to bridge the gap between communities: ML, Optimization, Vision, NLP, Robotics