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

CVPR’15 Tutorial

Sanja Fidler and Raquel Urtasun

June 7, 2015
Reconstruction / Localization
Scene Geometry via Humans

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

Figure: In which room are these people?
Exploit human actions and location in time-lapse videos (or single image) to infer functional room geometry (walkable, seatable and reachable surfaces)

**Figure**: In which room are these people?

**Answer**: Room A
Scene Geometry via Humans

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

(a) Action and Pose Detections

Sitting

Walkable

Sittable surfaces
Scene Geometry via Humans

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

**Figure:** Poses indicate contact points with the interacting surface.
Scene Geometry via Humans

(Fouhey et al., 2012)

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

(b) Appearances + People (Our approach).

<table>
<thead>
<tr>
<th>Location</th>
<th>Appearance Only</th>
<th>People Only</th>
<th>Appearance + People</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lee et al.</td>
<td>64.1%</td>
<td>70.4%</td>
<td>74.9%</td>
</tr>
<tr>
<td>Hedau et al.</td>
<td></td>
<td></td>
<td>70.8%</td>
</tr>
<tr>
<td>Overall</td>
<td>82.5%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure:** Time-lapse videos

<table>
<thead>
<tr>
<th>Location</th>
<th>Appearance Only</th>
<th>Appearance + People with Ground Truth Poses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lee et al.</td>
<td>66.4%</td>
<td>71.3%</td>
</tr>
<tr>
<td>Hedau et al.</td>
<td></td>
<td>77.0%</td>
</tr>
<tr>
<td>Ours</td>
<td>79.6%</td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure:** Single image prediction
Scene Geometry via Humans

[Fouhey et al., 2012]
Normals from Single Image

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

Xiaolong Wang, David F. Fouhey, Abhinav Gupta, Designing Deep Networks or Surface Normal Estimation, Arxiv, Nov 2014
Train three networks:

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

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

[Wang et al., 2014]

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

[Wang et al., 2014]
Table 1: Results on NYU v2 for per-pixel surface normal estimation, evaluated over valid pixels.

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary Stats. (°)</th>
<th>% Good Pixels (Higher Better)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Median</td>
</tr>
<tr>
<td>Our Network</td>
<td>25.0</td>
<td>13.8</td>
</tr>
<tr>
<td>UNFOLD [7]</td>
<td>35.1</td>
<td>19.2</td>
</tr>
<tr>
<td>Discr. [20]</td>
<td>32.5</td>
<td>22.4</td>
</tr>
<tr>
<td>3DP (MW) [6]</td>
<td>36.0</td>
<td>20.5</td>
</tr>
<tr>
<td>3DP [6]</td>
<td>34.2</td>
<td>30.0</td>
</tr>
</tbody>
</table>
Inserting Objects

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

link to video
How Many Times Have You Looked for Apartments?

United States:
- 11.7% per year

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

[From Rent3D slides]
Example Rental Data

- Plus some meta information e.g. wall height

[From Rent3D slides]
Rent3D: View Rental Ads in 3D

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

Data: http://www.cs.utoronto.ca/~fidler/projects/rent3D.html
Camera localization within apartment
Overview

[S. Fidler, R. Urtasun]

[Liu et al., 2015]
Accurate **camera localization:**

- **Scene cues**
Accurate camera localization:

- Scene cues
- Semantic cues
Accurate **camera localization**:

- **Scene cues**
- **Semantic cues**
- **Geometric cues** by exploiting the dimension information
\( r \in \{1, \ldots, R\} \) ... discrete random variable representing the room
Formulation

- \( r \in \{1, \ldots, R\} \) ... discrete random variable representing the room

Front wall is the plane defined by \( v_{P0} \) and \( v_{P1} \)
Formulation

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

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

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

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

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

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

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

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

The problem formulated as inference in a Conditional Random Field with the following energy:

\[
E(r, c_r, y) = E_{\text{scene-type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y)
\]
Energy Terms: Scene Type

\[ E(r, c_r, y) = E_{\text{scene\_type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y) \]

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

\[ E(r, c_r, y) = E_{\text{scene\_type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y) \]

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

\[ E(r, c_r, y) = E_{\text{scene\_type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y) \]

Orientation Map (Lee et al.)

Geometric Context (Hedau et al.)
Energy Terms: Layout

\[ E(r, c_r, y) = E_{\text{scene-type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y) \]

- **Potentials**: Counts of blue, red, etc, pixels inside and outside of each wall
- Fast computation using *integral geometry* [Schwing et al., 2012]
\[ E(r, c_r, y) = E_{\text{scene-type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y) \]
Energy Terms: Layout

\[ E(r, c_r, y) = E_{\text{scene\_type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y) \]

\[ y = (y_1, y_2, y_3, y_4), \quad y_4 = f(r, c_r, y_1, y_2, y_3) \]
Energy Terms: Layout

\[ E(r, c_r, y) = E_{\text{scene-type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y) \]

- \( y = (y_1, y_2, y_3, y_4) \), \( y_4 = f(r, c_r, y_1, y_2, y_3) \)
- Additional constraint on \( y \): Camera is \textbf{inside} the room
Energy Terms: Windows

\[ E(r, c_r, y) = E_{\text{scene\_type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y) \]

- Window-background segmentation
Energy Terms: Windows

\[ E(r, c_r, y) = E_{\text{scene\_type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y) \]

- Window-background segmentation
- **Potential**: count window pixels inside and outside the window area
We are minimizing the energy:

\[
(r^*, c_r^*, y^*) = \arg\min_{r, c_r, y} \left( E_{\text{scene type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y) \right)
\]
We are minimizing the energy:

$$(r^*, c^*_r, y^*) = \arg\min_{r, c_r, y} \left( E_{\text{scene-type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y) \right)$$

Inference:

- Exhaustive enumeration of $r$ and $c_r$
- Exact branch and bound inference for $y$ \cite{Schwing2012}
We are minimizing the energy:

\[(r^*, c_r^*, y^*) = \arg\min_{r, c_r, y} \left( E_{\text{scene\_type}}(r) + E_{\text{layout}}(r, c_r, y) + E_{\text{win}}(r, c_r, y) \right)\]

Inference:

- Exhaustive enumeration of \( r \) and \( c_r \)
- Exact branch and bound inference for \( y \) \[\text{[Schwing & Urtasun, 2012]}\]
- S-SVM for training
Crawled a London apartment rental site

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td># apartments</td>
<td>215</td>
</tr>
<tr>
<td># of images</td>
<td>1570</td>
</tr>
<tr>
<td># of indoor images</td>
<td>1259</td>
</tr>
<tr>
<td># images without GT alignment</td>
<td>82</td>
</tr>
<tr>
<td>avg. # rooms per apt</td>
<td>6</td>
</tr>
<tr>
<td>avg. # walls per apt</td>
<td>31</td>
</tr>
<tr>
<td>avg. # windows per apt</td>
<td>6</td>
</tr>
<tr>
<td>avg. # doors per apt</td>
<td>9</td>
</tr>
</tbody>
</table>
We assume we know which wall the camera is facing

**Metrics:** Pixel accuracy for predicting 5 walls

<table>
<thead>
<tr>
<th></th>
<th>Layout error</th>
<th>Evaluations</th>
<th>Test time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schwing'12</td>
<td>13.88</td>
<td>16012.4</td>
<td>0.0208</td>
</tr>
<tr>
<td>Rent3D</td>
<td>11.69</td>
<td>1271.5</td>
<td>0.0037</td>
</tr>
</tbody>
</table>
Results: Layout Estimation

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

<table>
<thead>
<tr>
<th></th>
<th>Layout error</th>
<th>Evaluations</th>
<th>Test time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schwing’12</td>
<td>13.88</td>
<td>16012.4</td>
<td>0.0208</td>
</tr>
<tr>
<td>Rent3D</td>
<td><strong>11.69</strong></td>
<td><strong>1271.5</strong></td>
<td><strong>0.0037</strong></td>
</tr>
</tbody>
</table>

- 2% reduction in layout error
We assume we know which wall the camera is facing.

**Metrics:** Pixel accuracy for predicting 5 walls

<table>
<thead>
<tr>
<th></th>
<th>Layout error</th>
<th>Evaluations</th>
<th>Test time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schwing'12</td>
<td>13.88</td>
<td>16012.4</td>
<td>0.0208</td>
</tr>
<tr>
<td>Rent3D</td>
<td>11.69</td>
<td>1271.5</td>
<td>0.0037</td>
</tr>
</tbody>
</table>

2% reduction in layout error

10 times less branching operations
We assume we know which wall the camera is facing

**Metrics**: Pixel accuracy for predicting 5 walls

<table>
<thead>
<tr>
<th></th>
<th>Layout error</th>
<th>Evaluations</th>
<th>Test time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schwing’12</td>
<td>13.88</td>
<td>16012.4</td>
<td>0.0208</td>
</tr>
<tr>
<td>Rent3D</td>
<td><strong>11.69</strong></td>
<td><strong>1271.5</strong></td>
<td><strong>0.0037</strong></td>
</tr>
</tbody>
</table>

- 2% reduction in layout error
- 10 times less branching operations
- 10x speedup
Results: Camera Localization

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

<table>
<thead>
<tr>
<th></th>
<th>Aspect</th>
<th>+Scene</th>
<th>+Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.0328</td>
<td>0.1138</td>
<td>0.1954</td>
</tr>
<tr>
<td>Rent3D (no windows)</td>
<td>0.0686</td>
<td>0.1945</td>
<td>0.2654</td>
</tr>
<tr>
<td>Rent3D (windowGT)</td>
<td>0.2128</td>
<td>0.4737</td>
<td>0.5995</td>
</tr>
<tr>
<td>Rent3D (window)</td>
<td>0.1670</td>
<td>0.3982</td>
<td>0.5080</td>
</tr>
</tbody>
</table>
**Metrics:** % of correct assignments of front wall to the apartment wall

<table>
<thead>
<tr>
<th></th>
<th>Aspect</th>
<th>+Scene</th>
<th>+Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.0328</td>
<td>0.1138</td>
<td>0.1954</td>
</tr>
<tr>
<td>Rent3D (no windows)</td>
<td>0.0686</td>
<td>0.1945</td>
<td>0.2654</td>
</tr>
<tr>
<td>Rent3D (windowGT)</td>
<td>0.2128</td>
<td>0.4737</td>
<td>0.5995</td>
</tr>
<tr>
<td>Rent3D (window)</td>
<td>0.1670</td>
<td>0.3982</td>
<td>0.5080</td>
</tr>
</tbody>
</table>

**Aspect:** Only aspect ratio information (and not scene) used
**Metrics:** % of correct assignments of front wall to the apartment wall

<table>
<thead>
<tr>
<th></th>
<th>Aspect</th>
<th>+Scene</th>
<th>+Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.0328</td>
<td>0.1138</td>
<td>0.1954</td>
</tr>
<tr>
<td>Rent3D (no windows)</td>
<td>0.0686</td>
<td>0.1945</td>
<td>0.2654</td>
</tr>
<tr>
<td>Rent3D (windowGT)</td>
<td>0.2128</td>
<td>0.4737</td>
<td>0.5995</td>
</tr>
<tr>
<td>Rent3D (window)</td>
<td>0.1670</td>
<td>0.3982</td>
<td>0.5080</td>
</tr>
</tbody>
</table>

**Scene:** Aspect information and scene classifier are used
*Metrics:* % of correct assignments of front wall to the apartment wall

<table>
<thead>
<tr>
<th></th>
<th>Aspect</th>
<th>+Scene</th>
<th>+Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.0328</td>
<td>0.1138</td>
<td>0.1954</td>
</tr>
<tr>
<td>Rent3D (no windows)</td>
<td>0.0686</td>
<td>0.1945</td>
<td>0.2654</td>
</tr>
<tr>
<td>Rent3D (windowGT)</td>
<td>0.2128</td>
<td>0.4737</td>
<td>0.5995</td>
</tr>
<tr>
<td>Rent3D (window)</td>
<td>0.1670</td>
<td>0.3982</td>
<td>0.5080</td>
</tr>
</tbody>
</table>

*Room:* We know which room the picture was taken in
**Metrics:** % of correct assignments of front wall to the apartment wall

<table>
<thead>
<tr>
<th></th>
<th>Aspect</th>
<th>+Scene</th>
<th>+Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.0328</td>
<td>0.1138</td>
<td>0.1954</td>
</tr>
<tr>
<td>Rent3D (no windows)</td>
<td>0.0686</td>
<td>0.1945</td>
<td>0.2654</td>
</tr>
<tr>
<td>Rent3D (windowGT)</td>
<td>0.2128</td>
<td>0.4737</td>
<td>0.5995</td>
</tr>
<tr>
<td>Rent3D (window)</td>
<td>0.1670</td>
<td>0.3982</td>
<td>0.5080</td>
</tr>
</tbody>
</table>
**Metrics**: % of correct assignments of front wall to the apartment wall

<table>
<thead>
<tr>
<th></th>
<th>Aspect</th>
<th>+Scene</th>
<th>+Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.0328</td>
<td>0.1138</td>
<td>0.1954</td>
</tr>
<tr>
<td>Rent3D (no windows)</td>
<td>0.0686</td>
<td>0.1945</td>
<td>0.2654</td>
</tr>
<tr>
<td>Rent3D (windowGT)</td>
<td>0.2128</td>
<td>0.4737</td>
<td>0.5995</td>
</tr>
<tr>
<td>Rent3D (window)</td>
<td>0.1670</td>
<td>0.3982</td>
<td>0.5080</td>
</tr>
</tbody>
</table>
Results: Joint Layout and Localization

Red arrow: Groundtruth camera  Green arrow: Predicted camera

Liu et al., 2015
Results

[Liu et al., 2015]

Window + Aspect

+ Scene

+ Room

Ground-truth

1 images out of 4
2 walls out of 8

4 images out of 4
8 walls out of 8

4 images out of 4
8 walls out of 8

-
Reconstructing Museums

- Virtual tour of large indoor spaces (e.g., museums)
- Uses a rig of cameras and three linear laser range sensors
Reconstructing Museums [Xiao and Furukawa, 2014]

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

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

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

3D point cloud → Segmented 2D slices → Stacked 2D CSG models → 3D CSG model → Wall model → Final textured model

for each slice

Free space constraint → Line extraction → Iterative 2D CSG model reconstruction → 2D CSG model
Reconstructing Museums

[Reconstruction of museums: The Frick Collection, National Gallery, The State Tretyakov Gallery]

[Link to video]
Reconstructing Indoor Tourist Sites

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

Reconstructing Indoor Tourist Sites

Figure: Localization results

Figure: Interactive visualization (link to video)
Indoor Scene Understanding
with RGB-D Data
Difficult problem?

- Noisy depth
- Missing depth
- Occlusion
- Viewpoint, aspect-ratio variation
R-CNN with Depth

Observation: The standard R-CNN pipeline doesn’t work well for detection on NYU-v2

Can we train a better network that includes depth?

<table>
<thead>
<tr>
<th></th>
<th>bath tub</th>
<th>bed</th>
<th>book shelf</th>
<th>box</th>
<th>chair</th>
<th>countertop</th>
<th>desk</th>
<th>door</th>
<th>dress er</th>
<th>garbage bin</th>
<th>lamp</th>
<th>monitor</th>
<th>nightstand</th>
<th>pillow</th>
<th>sink</th>
<th>sofa</th>
<th>table</th>
<th>television</th>
<th>toilet</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB DPM</td>
<td>9.0</td>
<td>0.9</td>
<td>27.6</td>
<td>9.0</td>
<td>0.1</td>
<td>7.8</td>
<td>7.3</td>
<td>0.7</td>
<td>2.5</td>
<td>1.4</td>
<td>6.6</td>
<td>22.2</td>
<td>10.0</td>
<td>9.2</td>
<td>4.3</td>
<td>5.9</td>
<td>5.5</td>
<td>5.8</td>
<td>34.4</td>
</tr>
<tr>
<td>RGBD-DPM</td>
<td>23.9</td>
<td>19.3</td>
<td>56.0</td>
<td>17.5</td>
<td>0.6</td>
<td>23.5</td>
<td>24.0</td>
<td>6.2</td>
<td>9.5</td>
<td>16.4</td>
<td>26.7</td>
<td>34.9</td>
<td>32.6</td>
<td>20.7</td>
<td>22.8</td>
<td>34.2</td>
<td>17.2</td>
<td>19.5</td>
<td>45.1</td>
</tr>
<tr>
<td>RGB R-CNN</td>
<td>22.5</td>
<td>16.9</td>
<td>45.3</td>
<td>28.5</td>
<td>0.7</td>
<td>25.9</td>
<td>30.4</td>
<td>9.7</td>
<td>16.3</td>
<td>18.9</td>
<td>27.9</td>
<td>32.5</td>
<td>17.0</td>
<td>11.1</td>
<td>16.6</td>
<td>29.4</td>
<td>12.7</td>
<td>27.4</td>
<td>44.1</td>
</tr>
</tbody>
</table>
R-CNN with Depth

- Trick: Use network pre-trained on e.g. ImageNet and fine-tune it on a 3D depth encoding “HHA”

- HHA: horizontal disparity, height above ground, and the angle between pixel’s normal and the inferred gravity direction
R-CNN with Depth

- Fine-tune network on synthetic views generated with Guo & Hoiem’s models
### R-CNN with Depth

[GuPTa et al., 2014]

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

[Gupta et al., 2014]
Goal: Align CAD models in RGB-D scenes
Aligning CAD Models in RGB-D [Gupta et al., 2015]

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

3 layer CNN on normal images trained on synthetic data

Search over scale, rotation, translation and CAD model to minimize re-projection error
### Table: 3D Detection Performance

<table>
<thead>
<tr>
<th>Task</th>
<th>AP&lt;sub&gt;b&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;r&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>[13]</td>
<td>train</td>
<td>35.9</td>
</tr>
<tr>
<td>[13] + Region Features</td>
<td>train</td>
<td>39.3</td>
</tr>
<tr>
<td>[13]</td>
<td>trainval</td>
<td>38.8</td>
</tr>
<tr>
<td>[13] + Region Features</td>
<td>trainval</td>
<td>41.2</td>
</tr>
<tr>
<td>[13] (Random Forests)</td>
<td>train</td>
<td>32.1</td>
</tr>
<tr>
<td>[13] + Region Features</td>
<td>train</td>
<td>34.0</td>
</tr>
<tr>
<td>[13] + Region Features</td>
<td>trainval</td>
<td>37.5</td>
</tr>
</tbody>
</table>

#### Figure: Detection and instance segmentation

<table>
<thead>
<tr>
<th>Method</th>
<th>3D all</th>
<th>3D clean</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our (3D Box on instance segm. from [13])</strong></td>
<td>48.4</td>
<td>66.1</td>
</tr>
<tr>
<td><strong>Our (3D Box around estimated model)</strong></td>
<td>58.5</td>
<td>71.1</td>
</tr>
<tr>
<td><strong>Song and Xiao [34]</strong></td>
<td>39.6</td>
<td>64.6</td>
</tr>
<tr>
<td><strong>Our [no RGB&lt;sup&gt;1&lt;/sup&gt;] (3D Box on instance segm. from [13])</strong></td>
<td>46.5</td>
<td>62.3</td>
</tr>
<tr>
<td><strong>Our [no RGB&lt;sup&gt;1&lt;/sup&gt;] (3D Box around estimated model)</strong></td>
<td>57.6</td>
<td>70.7</td>
</tr>
</tbody>
</table>

#### Figure: 3D detection
Holistic Scene Understanding

- Reasoning jointly about multiple related tasks may help

**bathroom**
Holistic Scene Understanding

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

**image**

**depth**

point cloud with **cuboids around objects**
Cuboid Candidates

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

example regions  
regions in 3D  
fit cuboids
Holistic 3D Scene Model

\[ p(y, s) \propto \exp \left( w_s^T \phi_s(s) + w_y^T \sum_{i=1}^{K} \phi_y(y_i) + w_{yy}^T \sum_{(i,j)} \phi_{yy}(y_i, y_j) + w_{sy}^T \sum_{i=1}^{K} \phi_{sy}(s, y_i) \right) \]

- **Unary:**
  - appearance
  - geometry

- **Pairwise:**
  - spatial relations
  - semantic relations
Unary Potentials

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

**Geometry features:**

**RGB-D features:**

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

Other features:
- horiz. aspect = long width / short width
- vert. aspect = height / long width
- area = long width * short width
- volume = area * height
- close to wall = exp(dist to wall / 0.1)
- parallel to wall = exp(radian / 0.1)
- close to ground = exp(dist to floor / 0.1)
Pairwise Potentials

Semantic context:

- **scene-object potential**:

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

- **object-object potential**

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

Geometric relations:

- **close-to**: Two objects are *close to* each other if their distance is less than 0.5 meters.

- **on-top-of**: Object A is *on top of* B if A is higher than B and (at least) 80% of A’s bottom face is contained within the top face of B.
Learning and Inference

- **Loss**: how far from GT is each hypothesis
  - Object: 0/1 loss based on IOU with GT
  - Scene: 0/1 loss

- **Learning**: Primal dual method blending learning and inference [Hazan and Urtasun, NIPS 2010]

- **Inference**: Distributed message passing [Schwing et al., CVPR 2011]

- **Timings**:
  - **learning** takes 2 minutes (∼800 images)
  - **inference** takes 15 ms per image (15 cuboids per image)

On Intel i7 quad-core CPU (4 threads)
- NYUv2 [Silberman et al, 2012]: 1449 scenes, 6680 objects, 21 object classes + background
- Ground truth: Fit 3D cuboids around GT regions and correct bad fits
- Standard split: 60% of images used for training and 40% for test
Results on GT Cuboids

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

<table>
<thead>
<tr>
<th>configuration</th>
<th>object</th>
<th>scene</th>
</tr>
</thead>
<tbody>
<tr>
<td>scene appearance only</td>
<td>-</td>
<td>55.20</td>
</tr>
<tr>
<td>segmentation only</td>
<td>54.46</td>
<td>-</td>
</tr>
<tr>
<td>geometry only</td>
<td>42.85</td>
<td>-</td>
</tr>
<tr>
<td>all unaries</td>
<td>59.02</td>
<td>55.20</td>
</tr>
<tr>
<td>unaries + scene-obj</td>
<td>60.00</td>
<td>57.65</td>
</tr>
<tr>
<td>full model</td>
<td>60.49</td>
<td>58.72</td>
</tr>
</tbody>
</table>

![Bar chart showing performance of different configurations]
Full Detection Pipeline

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

<table>
<thead>
<tr>
<th></th>
<th>DPM</th>
<th>seg.</th>
<th>seg.+geo.</th>
<th>all unaries</th>
<th>+scene-object</th>
<th>full model</th>
</tr>
</thead>
<tbody>
<tr>
<td>K = 8</td>
<td>8.01</td>
<td>28.98</td>
<td>30.22</td>
<td>35.17</td>
<td>35.18</td>
<td>35.23</td>
</tr>
<tr>
<td>K = 15</td>
<td>6.54</td>
<td>28.33</td>
<td>29.44</td>
<td>34.92</td>
<td>34.95</td>
<td>35.56</td>
</tr>
<tr>
<td>K = 30</td>
<td>4.96</td>
<td>24.81</td>
<td>25.58</td>
<td>32.54</td>
<td>32.57</td>
<td>33.10</td>
</tr>
</tbody>
</table>

S. Fidler, R. Urtasun
3D Indoor Scene Understanding
Example Detections

[Lin et al., 2013]
Generates layout and object candidates, and re-reasons about the best configuration in a holistic way.
Predicting Complete 3D Models

[Guo et al., 2015]

Input Image

Automatic 3D Model (two views)

link to video
Indoor RGB-D Datasets

- **NYUv2 dataset:**
  
  http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

- **RMRC challenge:**
  
  http://cs.nyu.edu/~silberman/rmrc2014/indoor.php

- **B3DO: Berkeley 3-D Object Dataset:**
  
  http://kinectdata.com/

- **SUN RGB-D:**
  
  http://rgbd.cs.princeton.edu/
Discussion

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