Geometric and Semantic 3D Reconstruction
Part 1: Single View

Srikumar Ramalingam
School of Computing
University of Utah
Coarse Single View Modeling

Manhattan Scene Modeling
[Delage’2005]

Outdoor layout estimation
[Hoiem’2007]

Pixel-wise depth estimation
[Saxena’2007]

Indoor layout estimation
[Hedau’2009]
Semantic Boundary Detection
Applications (depth and semantics from single view)

Car and pedestrian detection [Hoiem’2006]

Rendering synthetic objects [Karsh’ 2011]

SKYLINE2GPS [Ramalingam’09, Baatz’2002, Ho’2014]

Change detection [Taneja’2013]
Outline

• **Interpretation of Line Drawings**
• **Interactive 3D Modeling algorithms**
• **Prior-based 3D Modeling**
  – Semantic segmentation based approaches
  – Shape priors, Manhattan priors, dense modeling
  – Manhattan priors, sparse modeling
• **Deep learning methods**
  – Indoor Layout Estimation
  – Dense depth Estimation
  – Semantic Boundary detection
Single View 3D Reconstruction is Hard

- One unknown parameter for every pixel
- Need priors for depth estimation

[Sinha and Adelson 1993]

Infinite number of possible solutions!
Early Work in Computer Vision

• 1966: Marvin Minsky gave a summer project to a 1st year student to extract semantics from simple polyhedral objects.

Junction detection

L L W W L T Y W W L L W T Y Y W
Line Labeling

- Convex: $>180^\circ$
- Concave: $<180^\circ$
- Occluding

Diagram showing the labeling of convex and concave sides on a 3D object.
Line Labeling

Trihedral drawings - not more than 3 planes meet at a point.

Trihedral Junction Catalogue

<table>
<thead>
<tr>
<th>Junction Types</th>
<th>Sample Labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>(-&gt;,&lt;-),(&lt;-,-&gt;)</td>
</tr>
<tr>
<td>T</td>
<td>(&lt;-&gt;,&lt;-),(&lt;-,-&gt;,-)</td>
</tr>
<tr>
<td>W</td>
<td>(-&gt;,+,-),(+,+,+)</td>
</tr>
<tr>
<td>Y</td>
<td>(+,+,+),(-,-,-)</td>
</tr>
</tbody>
</table>

![Diagram of Junction Types]

![Diagram of Junction Types]
Given $n$ line segments, we have $n^4$ possibilities.
The junction catalogue significantly reduces the possible solutions.
Solved as constraint satisfaction problems with heuristics.
Line Labeling Results
Line Labeling Results
Line Labeling Results
Sugihara’s Lifting Procedure

- The 3D points are lifted along the projection rays.
- Sugihara in 1982 proposed a linear programming solution for lifting the line drawings to 3D space, given the line labeling and projection matrix.
- This approach can also verify whether a drawing is physically realizable or not for trihedral drawings.
Tetrahedral and General Line Drawings

• Sugihara’s formulation was applicable to trihedral drawings.
• Whiteley in 1987 extended it to general case using matroid theory.
• There are still some pathological cases and unsolved line drawings. This is still and active problem in constraint satisfaction community.
Pathological cases
Limitations of Classical Approaches

- Several assumptions – synthetic, polyhedral, graph connectivity
- Real scenes are not polyhedral
- No connectivity information
- Missing and spurious lines
Line Labeling in Real Images

- Recovering occlusion boundaries can provide a good sense of relative depths of different objects.

[Hoiem’2007]
Unfolding the Origami World

Input RGB image

Contour Labeling

[Fouhey’2014]
Semantic Classification of Boundary pixels from RGBD data

red (occluding), green (planar), blue (convex), yellow (concave)

[Sonì’2015]
Outline

• Interpretation of Line Drawings
• Interactive 3D Modeling algorithms
• Prior-based 3D Modeling
  – Semantic segmentation based approaches
  – Shape priors, Manhattan priors, dense modeling
  – Manhattan priors, sparse modeling
• Deep learning methods
  – Indoor Layout Estimation
  – Dense depth Estimation
  – Semantic Boundary detection
Interactive 3D Modeling Methods

- Interactive piecewise-planar reconstruction [Sturm’99]
- Interactive single view metrology [Criminisi’2002]

Google SketchUp
Interactive 3D Modeling

[Sturm’99]
Interactive 3D Modeling

[Sturm’99]
Interactive 3D Modeling

- Extract vanishing points and lines.
- Calibrate the camera.
- Back-project the points up to a scale.
- From vanishing points, compute plane normals.
- Partition the planes with known normals in sets of planes connected by at least one point.
- Compute the point and plane coordinates by satisfying all the user constraints.

[Sturm’99]
Interactive 3D Modeling

[Sturm’99]
Interactive Single View

[Criminisi’2002]
Single View Methods

• Interpretation of Line Drawings
• Interactive 3D Modeling algorithms
• Prior-based 3D Modeling
  – Semantic segmentation based approaches
  – Shape priors, Manhattan priors, dense modeling
  – Manhattan priors, sparse modeling
• Deep learning methods
  – Indoor Layout Estimation
  – Dense depth Estimation
  – Semantic Boundary detection
Overview

Input

Geometric Labels

Cut’n’Fold

3D Model

Image

Ground

Vertical

Sky

Learned Models

[Hoiem’2007]
Geometric Cues

<table>
<thead>
<tr>
<th>Feature Descriptions</th>
<th>Num</th>
<th>Used</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Color</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1. RGB values: mean</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>C2. HSV values: conversion from mean RGB values</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>C3. Hue: histogram (5 bins) and entropy</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>C4. Saturation: histogram (3 bins) and entropy</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td><strong>Texture</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1. DOOG Filters: mean abs response</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>T2. DOOG Filters: mean of variables in T1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>T3. DOOG Filters: id of max of variables in T1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>T4. DOOG Filters: (max - median) of variables in T1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>T5. Textons: mean abs response</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>T6. Textons: max of variables in T5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>T7. Textons: (max - median) of variables in T5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Location and Shape</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1. Location: normalized x and y, mean</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>L2. Location: norm. x and y, 10th and 90th percentile</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>L3. Location: norm. y wrt horizon, 10th and 90th pctl</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>L4. Shape: number of superpixels in constellation</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>L5. Shape: number of sides of convex hull</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>L6. Shape: num pixels/area(convex hull)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>L7. Shape: whether the constellation region is contiguous</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>3D Geometry</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G1. Long Lines: total number in constellation region</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>G2. Long Lines: % of nearly parallel pairs of lines</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>G3. Line Intersection: hist. over 12 orientations, entropy</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>G4. Line Intersection: % right of center</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>G5. Line Intersection: % above center</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>G6. Line Intersection: % far from center at 8 orientations</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>G7. Line Intersection: % very far from center at 8 orientations</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>G8. Texture gradient: x and y “edginess” (T2) center</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Geometric Layouts (image, model)
Geometric Layout Failures

Labeling Errors

[Hoiem’2007]
Ordering Prior in Layout Estimation using a Conditional Random Field
Most vision problems involving discrete variables can be solved using graph cuts, BP, or DNNs.

Typically formulated as minimization of an energy function involving “data” and “smoothness” terms.

Goal: find most probable interpretation of scene
Pseudo Boolean Functions (PBF)

• variables: $x_1, x_2, ..., x_n \in \{0,1\}$

• pseudo-Boolean functions (PBF):
  » maps a Boolean vector to a real number.
  \[ f : \{0,1\}^n \rightarrow \mathbb{R} \]

• has unique multi-linear representation:
  » For example:
  \[ f(x_1, x_2, x_3, x_4) = 2 - 3x_2x_4 + 5x_1x_2x_3 \]

[Boros&Hammer’2002]
Submodular Quadratic Pseudo-Boolean Functions

- A QPBF is submodular if and only if all quadratic coefficients are non-positive.

\[
f_3(x_1, x_2, x_3) = 15 + x_1 - 3x_2 - x_1x_2 - 5x_2x_3
\]

- Minimizing the function \( f \) is equivalent to solving maxflow/mincut algorithm. [Hammer 1965]

[Kohli&Torr’2005] [Boykov and Jolly’2001, Rother et al. 2004]
Multi-label problems

Segmentation (sky, building, and ground)

- Choose the disparities from the discrete set: \((1,2,\ldots,L)\)
Encoding multi-label variables using Boolean ones

1. Choose the encoding.

<table>
<thead>
<tr>
<th>y</th>
<th>x₁</th>
<th>x₂</th>
<th>x₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Using 3 Boolean variables to denote a 4-label variable

2. Generate encoding functions using Boolean variables.

\[ \delta^1_y = x_1, \]
\[ \delta^2_y = x_2 - x_1, \]
\[ \delta^3_y = x_3 - x_2, \]
\[ \delta^4_y = 1 - x_3 \]

3. Penalty terms to prohibit certain Boolean configurations.

4. There is a one-one correspondence at their respective minima:

\[ \arg\min_{y_i, i=\{1, \ldots, m\}} E^y(y_{1}, \ldots, y_{m}) = \arg\min_{x_i, i=\{1, \ldots, n\}} E^x(x_{1}, \ldots, x_{n}) \]

[Ishikawa et al. 2003, Ramalingam et al. 2008]
Triple clique terms

• Consider the following function:

\[ E(x_1, x_2, x_3) = \alpha_{ijk} x_1 x_2 x_3 + L_2 \]

• If \( a_{ijk} \leq 0 \)

\[ a_{ijk} x_1 x_2 x_3 = \min_{z \in \{0,1\}} a_{ijk} (x_i + x_j + x_k - 2)z \]

• Here \( z \) is auxiliary variable.

\[
\begin{align*}
\text{Multi-label} & \quad \text{Boolean} \\
\text{higher-order} & \quad \text{pairwise} \\
\arg\min_{y_i, i=\{1,..,m\}} E^y(y_1,..,y_m) & = \arg\min_{x_i, i=\{1,..,n\}} E^x(x_1,..,x_n)
\end{align*}
\]
Single View 3D Reconstruction

Original

Ground

Vertical

Sky
Single View Methods

Original       Superpixel       Ground truth       Hoiem’07       Ramalingam’08
Multi-scale MRFs for depth estimation

[Saxena et al. 2007]
Multi-scale MRFs for depth estimation

The convolutional filters used for texture energies and gradients. The relative depth features for each patch use histograms of the filter outputs.

[Saxena et al. 2007]
Multi-scale MRFs for depth estimation

[Saxena et al. 2007]
Multi-scale MRFs for depth estimation

[Saxena et al. 2007]
Outline

• Interpretation of Line Drawings
• Interactive 3D Modeling algorithms
• Prior-based 3D Modeling
  – Semantic segmentation based approaches
  – Shape priors, Manhattan priors, dense modeling
  – Manhattan priors, sparse modeling
• Deep learning methods
  – Indoor Layout Estimation
  – Dense depth Estimation
  – Semantic Boundary detection
Indoor layout Estimation

• Introduced by Hedau et al. 2009.
• Geometric context (Hoiem’2007) for pixel probabilities.
• Structured SVM for constrained room layout estimation.

Vanishing Point Detection and Line Clustering

- 3 vanishing points are detected
- lines are clustered

[Hedau’2009]
Four random variables \( \{y_1, y_2, y_3, y_4\} \) are used to denote the box. When the variables take a value outside the image, then the box corresponds to the one where not all four corners are not visible in the image.

[Ha09]
Layout Estimation

[Hedau’2009]
Geometric Context Features

[Hoiem’2007]
Orientation Maps

[Lee’2010]
Structured SVM

Let the image be given by $x_i$, the box layouts by $y_i$, and their corresponding features (orientation maps, geometric context) by $\psi(x_i, y_i)$.

Given the true layout and possible hypothesis, learn a linear classifier maximizing the score $w^T \psi(x_i, y_i)$ of the true layout:

$$\min_{w, \epsilon} \frac{1}{2} |w|^2 + C \sum_i \epsilon_i$$

S.T $\epsilon_i \geq 0, \forall i$

$$w^T \psi(x_i, y_i) - w^T \psi(x_i, y) \geq \Delta(y_i, y) - \epsilon_i$$

$\forall_i, \forall y \in Y / y_i$
Junction-Based Features

- Corners of a room correspond to specific types of Y-junctions.
- X-junctions mostly occur at windows and not at the boundaries of a layout.
- The type of L-junction give useful cues about the orientation of the face.
Different types of junctions

- Let $S = \{ \vec{x}, \vec{x}, \vec{y}, \vec{y}, \vec{z}, \vec{z} \}$
- Every subset $A \subseteq S$ defines a junction type
- Compute junction type for every pixel using voting
Junction Detection
CRF Modeling

- Sampling rays are chosen based on Y-junctions.
- Our goal is to find the labels for each variable \( x_i = \{ \text{Left wall, middle wall, Right wall, Ceiling, and Floor} \} \).
- We would like to define an objective function that gives high scores for correct layouts and low scores for incorrect ones.

[Ramalingam’2013]
Unary Term

- Unary term captures the orientation properties of every polygon.
- We use L, X and T junction scores inside every polygon to denote the unary potentials. The junctions are chosen according to the orientations.
- We also used two features based on orientation maps and geometric context.
• A line separating two polygons act as pairwise potentials. Two nodes can more easily take different labels if there is a line between them.
• We also use the junction scores of Y and W junctions on the line segment to construct the pairwise potentials.

[Ramalingam’2013]
We use Y junctions to construct triple clique potentials.

[Ramalingam’2013]
Objective function

\[ E(x, \omega) = \sum_{i=1}^{n} \omega_i \Psi(i, a) \delta_{ia} + \sum_{\{i,j\} \in E} \omega_{i,j} \Psi(i, j, a, b) \delta_{ia} \delta_{jb} + \sum_{\{i,j,k\} \in \mathcal{T}} \omega_{i,j,k} \Psi(i, j, k, a, b, c) \delta_{ia} \delta_{jb} \delta_{kc} \]

\[ i, j, k = \{1, 2, \ldots, n\} \quad a, b, c = \{L, M, R, F, C\} \]

- Unary potential
- Pairwise potential
- Triple clique potential

\[ \Psi(i, a) \]
\[ \Psi(i, j, a, b) \]
\[ \Psi(i, j, k, a, b, c) \]

\[ \omega = \{\omega_i, \omega_{ij}, \omega_{ijk}\} \]

- Parameters

\[ \delta_{ia} = 1 \quad \text{when} \quad x_i = a \]
\[ \delta_{ia} = 0 \quad \text{otherwise} \]
Parameter Learning using Structured Prediction

- Our objective function can be written as

\[ E(x, \omega) = \omega^T \Phi(\Psi, x) \]

- The parameters of our energy function is computed using structured SVM.
- Features (Junction features, normal maps and geometric context)
- During testing, choose the hypothesis with the highest score.
Top 10 results

Worst 5 results

[Ramalingam’2013]
Single View Methods

• Interpretation of Line Drawings
• Interactive 3D Modeling algorithms
• Prior-based 3D Modeling
  – Semantic segmentation based approaches
  – Shape priors, Manhattan priors, dense modeling
  – Manhattan priors, sparse modeling
• Deep learning methods
Single View Line-Based 3D Modeling

Given a single image:

3D model

Applications: cadastral modeling, localization, self-calibration, collision avoidance

[Ramalingam’2013]
Basic Idea: Straws and Connector City
Algorithm

1. lines, VP, graph, and junctions
2. calibration and rotation
3. LP and MST

aligns 3D Manhattan lines to X, Y, and Z directions
Our Main Idea

Assumption: largest subset that satisfies all the constraints will consist of correct intersections.

The intersections don’t simultaneously satisfy all constraints: projection, orthogonality and parallelism.

8 possible intersections, H is a wrong intersection

7 intersections when we remove H

Only 6 intersections when we include H

Assumption: largest subset that satisfies all the constraints will consist of correct intersections.
Linear Program

- L0 minimization is NP-hard
- Relax to L1 norm minimization problem
- Get penalty terms $w_{ij}$ from junctions

\[
\begin{align*}
\min_{\lambda_i} & \quad \sum_{(i,j) \in \mathcal{E}} (\|w_{ij}s_{ij}\|_1) \\
\text{s.t.} & \quad |\lambda_id_{ia} - \lambda_jd_{ja}| \leq s_{ij}, a \in \{x, y, z\} \setminus \{D_i, D_j\} \\
& \quad \lambda_i \geq 1, \quad i \in \mathcal{V}. \\
& \quad D_i \in \{x, y, z\}
\end{align*}
\]
Junction-Breaking Costs

- L and X junctions occur on planar surfaces
- T junctions occur on both planar surfaces and on occluding edges
- Y and W junctions are common on concave and convex edges
- We use different parameters for $w_{ij}$ depending on junctions
Qualitative Evaluation

image

detected lines

two perspective views of the line reconstruction
Challenging Cases

transparent objects and too much clutter lead to the collapse of two planes into one.

spurious lines on the shiny floor lead to incorrect reconstruction

detected lines

two perspective views of the line reconstruction
Quantitative Evaluation
(York urban database – 102 images)

<table>
<thead>
<tr>
<th>Mean</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detected Lines</td>
<td>235</td>
</tr>
<tr>
<td>Lines in MST</td>
<td>152</td>
</tr>
<tr>
<td>Orthogonality</td>
<td>659</td>
</tr>
<tr>
<td>Collinearity</td>
<td>27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Incorrect constraints (%)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifting Algorithm</td>
<td>4.91</td>
<td>&lt;1 s</td>
</tr>
</tbody>
</table>
Pencil Sketches and Drawings
Ames Room

Top view of the room

viewpoint
Impossible Drawings
Non-Manhattan Scenes

Non-Manhattan lines are mostly ignored or approximated.
Reconstruction of Partially Occluded Regions

- Second glass door behind the right wall
- Identifies partially occluded structures
Outline

• Interpretation of Line Drawings
• Interactive 3D Modeling algorithms
• Prior-based 3D Modeling
  – Semantic segmentation based approaches
  – Shape priors, Manhattan priors, dense modeling
  – Manhattan priors, sparse modeling
• Deep learning methods
  – Indoor Layout Estimation
  – Dense depth Estimation
  – Semantic Boundary detection
Indoor Layout Estimation

Input image -> Deep FCNN -> Heat maps for different classes

replaced by

Geometric context
Orientation maps
Manhattan Junctions

Indoor scene layout

[Dasgupta’2016]
<table>
<thead>
<tr>
<th>Image</th>
<th>DL</th>
<th>Result</th>
<th>GT</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.jpg" alt="Image" /></td>
<td><img src="result1.jpg" alt="Image" /></td>
<td><img src="gt1.jpg" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td><img src="image2.jpg" alt="Image" /></td>
<td><img src="result2.jpg" alt="Image" /></td>
<td><img src="gt2.jpg" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td><img src="image3.jpg" alt="Image" /></td>
<td><img src="result3.jpg" alt="Image" /></td>
<td><img src="gt3.jpg" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td><img src="image4.jpg" alt="Image" /></td>
<td><img src="result4.jpg" alt="Image" /></td>
<td><img src="gt4.jpg" alt="Image" /></td>
<td></td>
</tr>
</tbody>
</table>

[DL - Dasgupta’2016]
# Quantitative Evaluation

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Pixel Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hedau et al. (2009)</td>
<td>21.20</td>
</tr>
<tr>
<td>Del Pero et al. (2012)</td>
<td>16.30</td>
</tr>
<tr>
<td>Gupta et al. (2010)</td>
<td>16.20</td>
</tr>
<tr>
<td>Zhao et al. (2013)</td>
<td>14.50</td>
</tr>
<tr>
<td>Ramalingam et al. (2013)</td>
<td>13.34</td>
</tr>
<tr>
<td>Mallya et al. (2015)</td>
<td>12.83</td>
</tr>
<tr>
<td>Schwing et al. (2012)</td>
<td>12.8</td>
</tr>
<tr>
<td>Del Pero et al. (2013)</td>
<td>12.7</td>
</tr>
<tr>
<td>Dasgupta et al. (2016)</td>
<td>9.73</td>
</tr>
</tbody>
</table>
Outline

• Interpretation of Line Drawings
• Interactive 3D Modeling algorithms
• Prior-based 3D Modeling
  – Semantic segmentation based approaches
  – Shape priors, Manhattan priors, dense modeling
  – Manhattan priors, sparse modeling
• Deep learning methods
  – Indoor Layout Estimation
  – Dense depth Estimation
  – Semantic Boundary detection
Normals, Depth, and Semantic Labels

• Same architecture for depth, normals, and labels
• Fast (30 Hertz)

[Eigen and Fergus’ 2015]
Depth Estimation

[Image: A set of images showing depth estimation results from Eigen and Fergus' 2015 work.]
Deeper depth prediction

CNN Architecture

Upsampling

(s) nxn convolution (stride s)

ReLU

3x3 max-pooling stride 2

Batch normalization

Residual blocks

[Laina’ 2016]
Loss function

Typical loss for regression:

L2 loss: \[ \frac{1}{n} \sum_i (d_i - d_i^*)^2. \]

L1 loss: \[ \frac{1}{n} \sum_i |d_i - d_i^*|. \]

Berhu loss (reverse Huber loss): \[ \frac{1}{n} \sum_i B(d_i - d_i^*), \]
where \( B(x) = \begin{cases} |x|, & |x| \leq c, \\ \frac{1}{2c} (x^2 + c^2), & |x| > c. \end{cases} \)

\( n \): number of pixels
\( d \): predicted depth map
\( d^* \): ground truth depth map

[Laina’ 2016]
## Evaluation

<table>
<thead>
<tr>
<th>NYU Depth v2</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liu’2015</td>
<td>0.821</td>
</tr>
<tr>
<td>Wang’2015</td>
<td>0.745</td>
</tr>
<tr>
<td>Roy’2016</td>
<td>0.744</td>
</tr>
<tr>
<td>Eigen’2015</td>
<td>0.641</td>
</tr>
<tr>
<td>Laina’2016 (uses ResNet)</td>
<td>0.573</td>
</tr>
</tbody>
</table>
Normal Estimation

Input  3DP  Ladicky’14  Wang’15  Eigen’15  GT

[ Eigen and Fergus’ 2015 ]
Semantic Labels

RGB Input  4-class pred.  13-class pred.  13-class GT

[Eigen and Fergus’ 2015]
Outline

• Interpretation of Line Drawings
• Interactive 3D Modeling algorithms
• Prior-based 3D Modeling
  – Semantic segmentation based approaches
  – Shape priors, Manhattan priors, dense modeling
  – Manhattan priors, sparse modeling
• Deep learning methods
  – Indoor Layout Estimation
  – Dense depth Estimation
  – Semantic Boundary detection
CASENet: Category-Aware Semantic Edge Detection

Zhiding Yu, Chen Feng, Ming-Yu Liu, Srikumar Ramalingam, Teng-Yok Lee
Conventional Edge
Conventional Edge
Semantic Edge?
Semantic Edge?
Category-Aware Semantic Edge?
Category-Aware Semantic Edge?
Why Do We Care?

• Boundary cues are highly beneficial:
  – Semantic segmentation [Chen’16]
  – Object detection/recognition [Karsch’13]
  – 3D reconstruction [Shan’14]
  – Depth from single image [Hoiem’07]

• Semantic boundary helps towards a holistic interpretation of natural images
Previous Methods: SBD

• Semantic Contours from Inverse Detectors
  [Hariharan’11]
Previous Methods: HED

- Holistically-Nested Edge Detection [Xie and Tu, ICCV’15]
Problem Formulation

• Input: RGB image $I$

• Output: $K$ boundary maps $\{B_1, \cdots, B_K\}$
  
  $K$: # object categories
  
  $B_k(p) \in [0,1]$: the confidence of pixel $p$ on the boundary of the $k$-th object category
  
  Non-mutual-exclusive: $\sum_k B_k(p)$ need not be 1
CASENet

(e) Side Feature

(h) Shared Concatenation

res5
res4
res3
res2
res1
data

1024
2048
256
64
3

fused classification
side 5 classification
side 3 feature extraction
side 2 feature extraction
side 1 feature extraction

shared concatenation

shared concatenation

4K
Loss Function

- Self-balanced cross-entropy loss
  - severe skewness of pos/neg training examples
  - Sum of each category’s loss
  - $\beta$: percentage of non-boundary pixels in the image

$$l(W) = \sum_k l_k(W)$$

$$= \sum_k \sum_p \{-\beta \log Pr(B_k(p) = 1|I; W)$$

$$- (1 - \beta) \log Pr(B_k(p) = 0|I; W)\},$$
Experiments: SBD

<table>
<thead>
<tr>
<th>Metric</th>
<th>Method</th>
<th>aero</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>mbike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF</td>
<td>InvDet</td>
<td>42.6</td>
<td>49.5</td>
<td>15.7</td>
<td>16.8</td>
<td>36.7</td>
<td>43.0</td>
<td>40.8</td>
<td>22.6</td>
<td>18.1</td>
<td>26.6</td>
<td>10.2</td>
<td>18.0</td>
<td>35.2</td>
<td>29.4</td>
<td>48.2</td>
<td>14.3</td>
<td>26.8</td>
<td>11.2</td>
<td>22.2</td>
<td>32.0</td>
<td>28.0</td>
</tr>
<tr>
<td></td>
<td>HFL-FC8</td>
<td>71.6</td>
<td>59.6</td>
<td>68.0</td>
<td>54.1</td>
<td>57.2</td>
<td>68.0</td>
<td>58.8</td>
<td>69.3</td>
<td>43.3</td>
<td>65.8</td>
<td>33.3</td>
<td>67.9</td>
<td>67.5</td>
<td>62.2</td>
<td>69.0</td>
<td>43.8</td>
<td>68.5</td>
<td>33.9</td>
<td>57.7</td>
<td>54.8</td>
<td>58.7</td>
</tr>
<tr>
<td></td>
<td>HFL-CRF</td>
<td>73.9</td>
<td>61.4</td>
<td>74.6</td>
<td>57.2</td>
<td>58.8</td>
<td>70.4</td>
<td>61.6</td>
<td>71.9</td>
<td>46.5</td>
<td>72.3</td>
<td>36.2</td>
<td>71.1</td>
<td>73.0</td>
<td>68.1</td>
<td>70.3</td>
<td>44.4</td>
<td>73.2</td>
<td>42.6</td>
<td>62.4</td>
<td>60.1</td>
<td>62.5</td>
</tr>
<tr>
<td>MF (Ours)</td>
<td>CASENet-VGG</td>
<td>85.1</td>
<td>79.3</td>
<td>79.3</td>
<td>70.4</td>
<td>65.7</td>
<td>85.1</td>
<td>78.4</td>
<td>86.3</td>
<td>63.4</td>
<td>86.5</td>
<td>57.9</td>
<td>86.2</td>
<td>86.1</td>
<td>81.4</td>
<td>82.6</td>
<td>69.5</td>
<td>86.2</td>
<td>72.4</td>
<td>78.1</td>
<td>65.5</td>
<td>77.3</td>
</tr>
<tr>
<td></td>
<td>Basic</td>
<td>88.1</td>
<td>84.6</td>
<td>88.7</td>
<td>76.0</td>
<td>81.2</td>
<td>90.8</td>
<td>84.5</td>
<td>91.9</td>
<td>72.3</td>
<td>88.4</td>
<td>70.7</td>
<td>90.5</td>
<td>89.2</td>
<td>85.3</td>
<td>86.7</td>
<td>73.9</td>
<td>87.8</td>
<td>79.9</td>
<td>86.9</td>
<td>76.8</td>
<td>83.7</td>
</tr>
<tr>
<td></td>
<td>DSN</td>
<td>88.3</td>
<td><strong>85.4</strong></td>
<td>87.9</td>
<td>75.4</td>
<td>79.1</td>
<td><strong>91.0</strong></td>
<td>84.4</td>
<td><strong>92.1</strong></td>
<td>72.2</td>
<td>87.9</td>
<td>70.2</td>
<td>90.8</td>
<td>89.7</td>
<td>85.4</td>
<td>86.4</td>
<td>74.5</td>
<td>90.7</td>
<td>78.3</td>
<td><strong>87.5</strong></td>
<td><strong>79.7</strong></td>
<td>83.8</td>
</tr>
<tr>
<td></td>
<td>CASENet^-</td>
<td>88.5</td>
<td>85.1</td>
<td>88.5</td>
<td>73.0</td>
<td>78.8</td>
<td>89.9</td>
<td>84.2</td>
<td>91.0</td>
<td>71.8</td>
<td>88.5</td>
<td>67.4</td>
<td>90.1</td>
<td>89.8</td>
<td>85.6</td>
<td>85.9</td>
<td>74.2</td>
<td>88.0</td>
<td>80.0</td>
<td>86.0</td>
<td>75.4</td>
<td>83.0</td>
</tr>
<tr>
<td></td>
<td>CASENet</td>
<td><strong>88.7</strong></td>
<td>85.4</td>
<td><strong>89.0</strong></td>
<td>76.7</td>
<td>80.5</td>
<td>90.8</td>
<td>84.4</td>
<td>92.0</td>
<td><strong>72.8</strong></td>
<td><strong>89.4</strong></td>
<td>69.7</td>
<td><strong>90.8</strong></td>
<td><strong>90.5</strong></td>
<td><strong>85.8</strong></td>
<td><strong>86.9</strong></td>
<td><strong>75.1</strong></td>
<td>88.9</td>
<td>79.0</td>
<td>87.0</td>
<td>76.9</td>
<td><strong>84.0</strong></td>
</tr>
<tr>
<td>AP</td>
<td>InvDet</td>
<td>38.4</td>
<td>29.6</td>
<td>9.6</td>
<td>9.9</td>
<td>24.2</td>
<td>33.6</td>
<td>31.3</td>
<td>17.3</td>
<td>10.7</td>
<td>16.4</td>
<td>3.7</td>
<td>12.1</td>
<td>28.5</td>
<td>20.4</td>
<td>45.7</td>
<td>7.6</td>
<td>16.1</td>
<td>5.7</td>
<td>14.6</td>
<td>22.7</td>
<td>19.9</td>
</tr>
<tr>
<td></td>
<td>HFL-FC8</td>
<td>66.0</td>
<td>50.7</td>
<td>58.9</td>
<td>40.6</td>
<td>47.1</td>
<td>62.9</td>
<td>51.0</td>
<td>59.0</td>
<td>25.6</td>
<td>54.6</td>
<td>15.3</td>
<td>57.8</td>
<td>57.3</td>
<td>55.9</td>
<td>62.2</td>
<td>27.5</td>
<td>55.6</td>
<td>18.0</td>
<td>50.1</td>
<td>40.6</td>
<td>47.8</td>
</tr>
<tr>
<td></td>
<td>HFL-CRF</td>
<td>71.2</td>
<td>55.2</td>
<td>69.3</td>
<td>45.7</td>
<td>48.9</td>
<td>71.1</td>
<td>56.8</td>
<td>65.7</td>
<td>29.1</td>
<td>65.9</td>
<td>17.7</td>
<td>64.5</td>
<td>68.3</td>
<td>64.7</td>
<td>65.9</td>
<td>29.1</td>
<td>66.5</td>
<td>25.7</td>
<td>60.0</td>
<td>49.8</td>
<td>54.6</td>
</tr>
<tr>
<td>AP (Ours)</td>
<td>CASENet-VGG</td>
<td>70.9</td>
<td>58.4</td>
<td>62.4</td>
<td>47.2</td>
<td>48.7</td>
<td>66.4</td>
<td>62.3</td>
<td>67.9</td>
<td>36.0</td>
<td>47.5</td>
<td>22.4</td>
<td>62.8</td>
<td>56.9</td>
<td>61.5</td>
<td>77.5</td>
<td>30.9</td>
<td>57.7</td>
<td>28.7</td>
<td>58.8</td>
<td>44.5</td>
<td>53.5</td>
</tr>
<tr>
<td></td>
<td>Basic</td>
<td>80.9</td>
<td>70.8</td>
<td>79.2</td>
<td>57.8</td>
<td>67.3</td>
<td>84.3</td>
<td>74.1</td>
<td>86.1</td>
<td>47.4</td>
<td>71.6</td>
<td>37.1</td>
<td>81.9</td>
<td>75.8</td>
<td>74.3</td>
<td>82.1</td>
<td>43.7</td>
<td>72.5</td>
<td>45.0</td>
<td>77.0</td>
<td><strong>62.9</strong></td>
<td>68.6</td>
</tr>
<tr>
<td></td>
<td>DSN</td>
<td>80.0</td>
<td>73.2</td>
<td>77.3</td>
<td>56.8</td>
<td>65.8</td>
<td>84.8</td>
<td>74.7</td>
<td>85.2</td>
<td>45.9</td>
<td>69.5</td>
<td>37.3</td>
<td>81.4</td>
<td>75.2</td>
<td>74.4</td>
<td>81.9</td>
<td>44.6</td>
<td>71.8</td>
<td>45.4</td>
<td>77.3</td>
<td>62.5</td>
<td>68.2</td>
</tr>
<tr>
<td></td>
<td>CASENet^-</td>
<td>81.3</td>
<td>70.8</td>
<td>79.6</td>
<td>56.5</td>
<td>68.6</td>
<td>84.4</td>
<td>74.5</td>
<td>86.1</td>
<td>47.0</td>
<td><strong>73.6</strong></td>
<td>36.8</td>
<td>81.6</td>
<td>78.6</td>
<td>73.6</td>
<td>82.9</td>
<td>44.0</td>
<td>73.1</td>
<td>43.4</td>
<td>76.1</td>
<td>61.3</td>
<td>68.7</td>
</tr>
<tr>
<td></td>
<td>CASENet</td>
<td><strong>82.0</strong></td>
<td>73.6</td>
<td><strong>80.6</strong></td>
<td>59.0</td>
<td><strong>70.0</strong></td>
<td><strong>85.3</strong></td>
<td><strong>75.9</strong></td>
<td><strong>86.9</strong></td>
<td><strong>49.4</strong></td>
<td><strong>73.6</strong></td>
<td><strong>38.7</strong></td>
<td><strong>82.7</strong></td>
<td><strong>79.4</strong></td>
<td><strong>75.6</strong></td>
<td><strong>83.4</strong></td>
<td><strong>45.9</strong></td>
<td><strong>74.5</strong></td>
<td><strong>46.7</strong></td>
<td><strong>77.9</strong></td>
<td><strong>62.4</strong></td>
<td><strong>70.2</strong></td>
</tr>
</tbody>
</table>

- **CASENet^-<CASENet**
  - side supervision at high level helps
- **HFL-FC8<<CASENet-VGG**
  - CASENet on VGGNet still works better
Videos

https://www.youtube.com/watch?v=BNE1hAP6Qho

107
Roadmap

- Single view 3D modeling deals with the complete description of a scene using semantics, depth, contours, objects, and junctions.
- We need inference algorithms that can handle complex interdependencies and long-range interactions.
- Deep learning methods provide good unary potentials. Can we have an end-to-end method satisfying common scene constraints?
Thank You!
Experiments: SBD

<table>
<thead>
<tr>
<th>DSN</th>
<th>CASENet~</th>
<th>CASENet</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="DSN-Boat" /></td>
<td><img src="image2" alt="DSN-Person" /></td>
<td><img src="image3" alt="CASENet~" /></td>
</tr>
<tr>
<td><img src="image5" alt="Side1" /></td>
<td><img src="image6" alt="Side2" /></td>
<td><img src="image7" alt="Side3" /></td>
</tr>
<tr>
<td><img src="image9" alt="Side5" /></td>
<td><img src="image10" alt="Side5-Person" /></td>
<td><img src="image11" alt="Side5-Boat" /></td>
</tr>
</tbody>
</table>
Experiments: SBD

<table>
<thead>
<tr>
<th>Input</th>
<th>Ground Truth</th>
<th>Basic</th>
<th>DSN</th>
<th>CASENet</th>
<th>CASENet-VGG</th>
</tr>
</thead>
</table>

111
Experiments: Cityscapes
CASENet

- **Lower level feature**
  - RF: 7, 21, 41
  - Not directly useful for multi-label classification
  - Useful for suppressing non-boundary pixel activations
  - Side supervision: No
    - Overly finetune earlier layers
    - Disturbing context extraction

- **High level feature**
  - RF: 137, 161
  - Side supervision: Yes
    - Necessary