# Single view 3D Scene Layout

3D Vision

University of Illinois

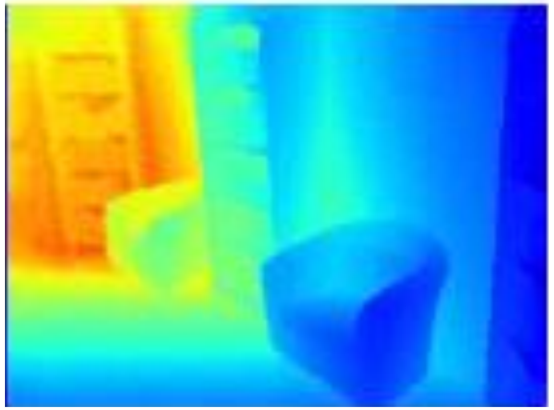
Derek Hoiem

# Agenda

- Scene representations
- Outdoor scene layout
  - Photo popup, WorldSheet
- Indoor scene layout
  - Room as box
  - 360-based layout
  - Complete scene parsing

**Geometric Pixel Labeling**

- Structured by **pixels**
- **View-dependent**: depth, normals, boundaries are relative to current view
- **Translation** between two measurable things (e.g. intensity to depth)



**Scene Layout**

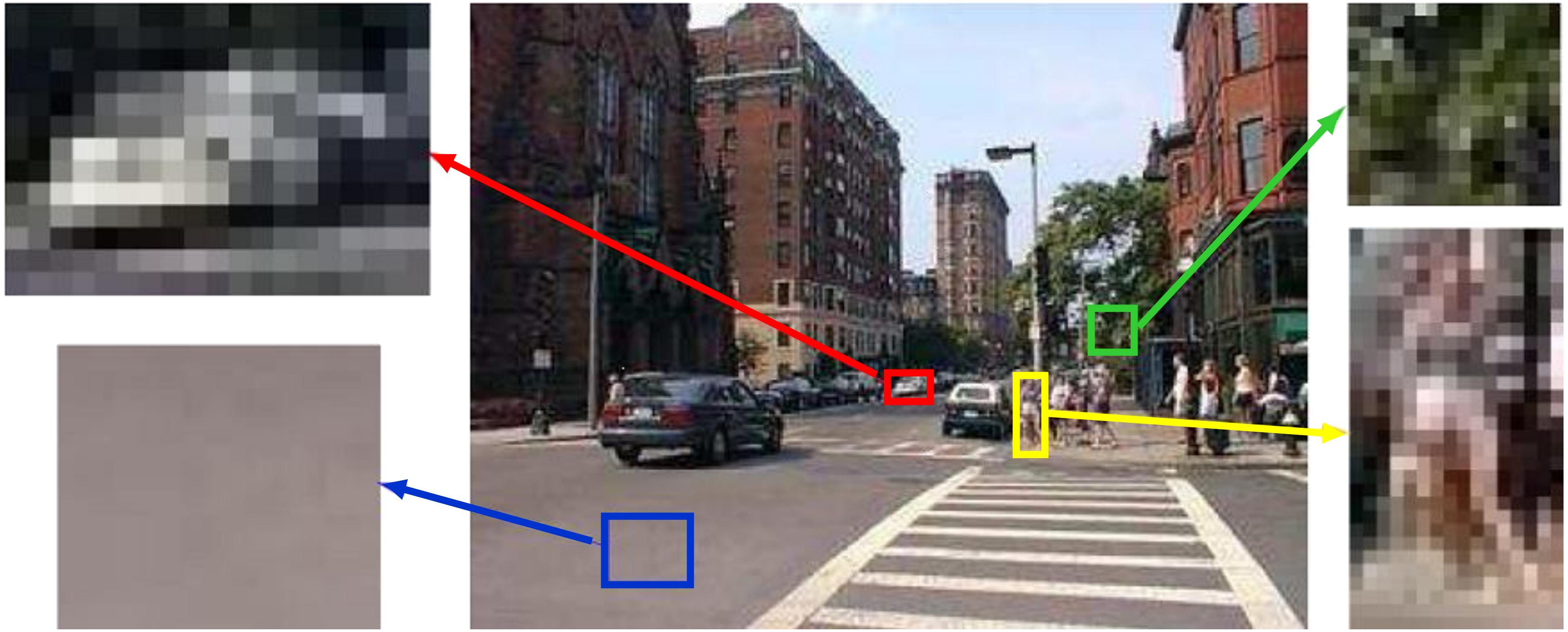
- Structured with **objects and surfaces**
- **View-independent**: same representation applies to many perspectives
- **Interpretation** of models from measurement

# Uses of scene layout

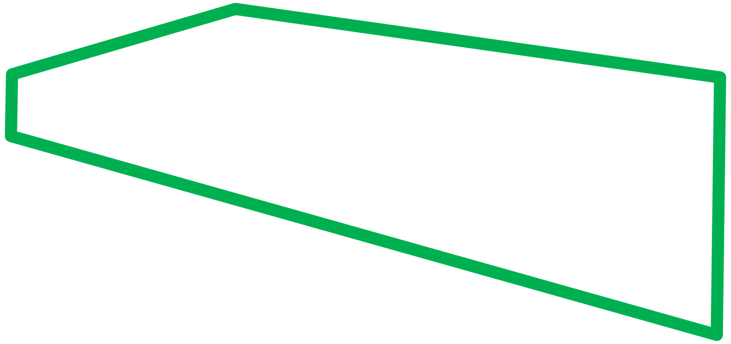
Context for recognition

# Uses of scene layout

Context for recognition



# Physical space helpful for recognition



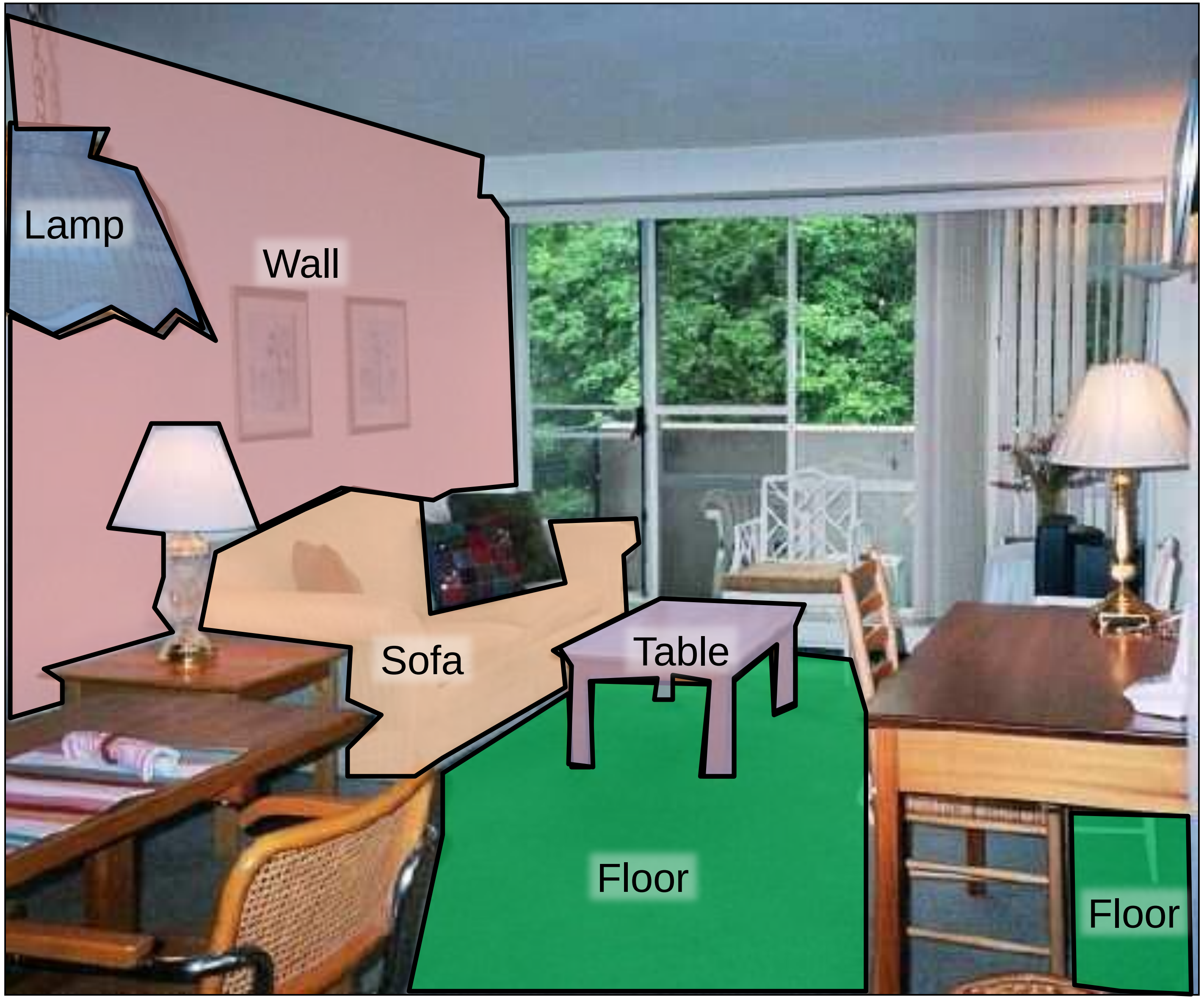
Apparent shape depends strongly on viewpoint

# Physical space needed to predict appearance

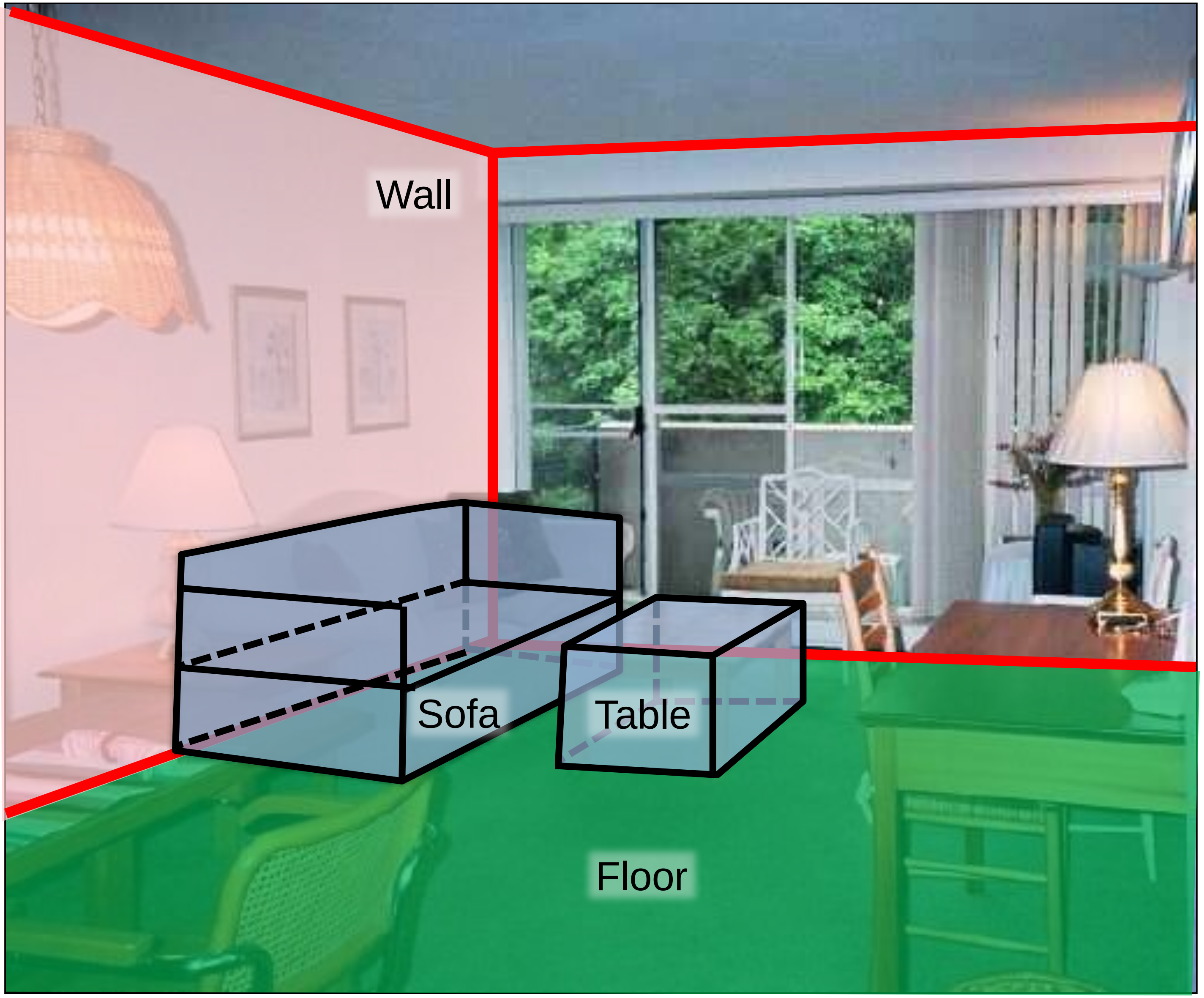
# Physical space needed to predict appearance

# Scene understanding

# Do pixel labels provide scene understanding?



# Interpreting scene layout in a physical space



# Physical space needed for affordance

Is this a good place to sit?

Could I stand over here?

Can I put my cup here?

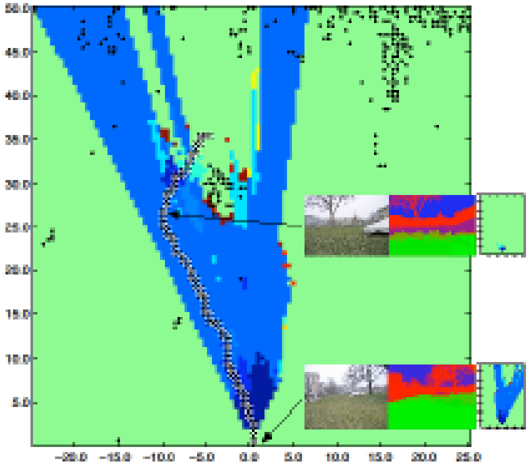
Walkable path

# Uses of scene layout

Other direct applications

a) Assisted driving
b) Robot navigation/interaction
c) Object insertion

3D Reconstruction: Input, Mesh, Novel View

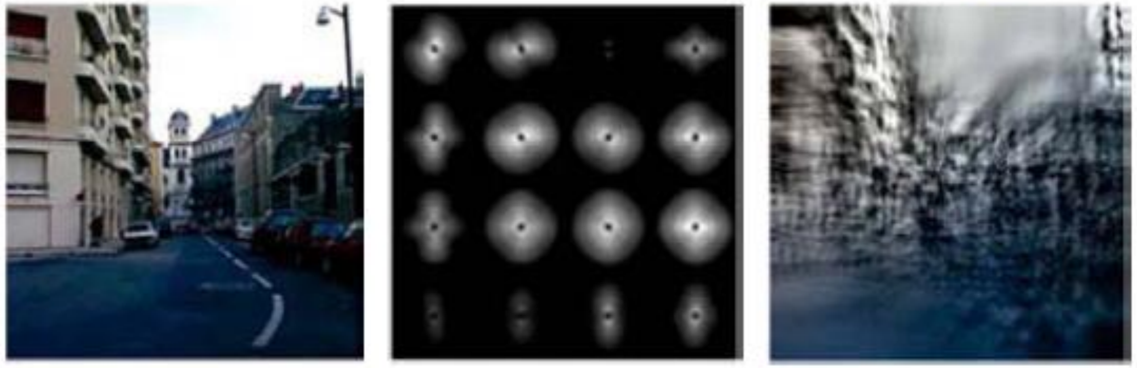


Robot Navigation: Path Planning

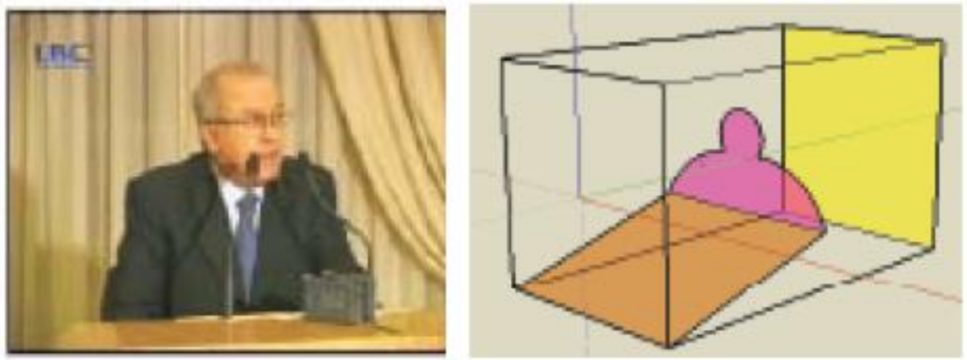
# How to represent scene space?

Wide variety of possible representations

## Scene-Level Geometric Description



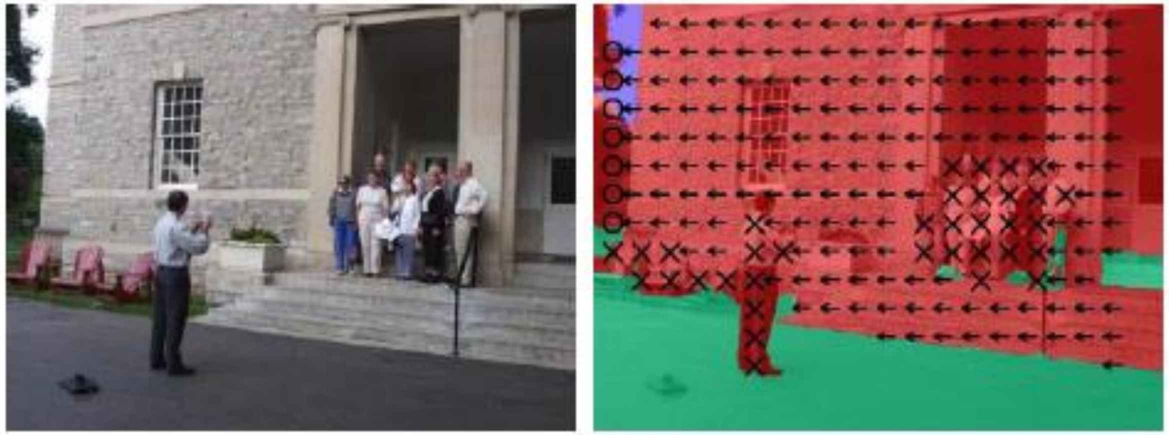
a) Gist, Spatial Envelope



b) Stages

Figs from Hoiem - Savarese 2011 book

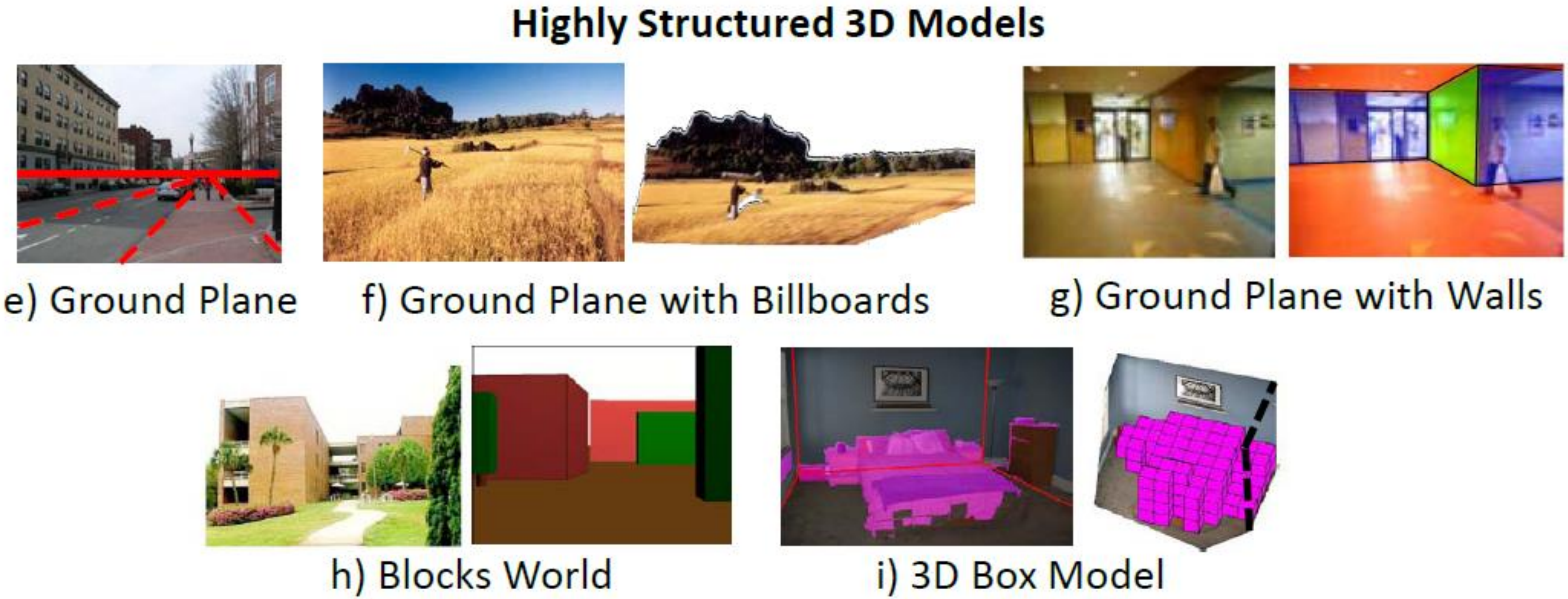
# Retinotopic Maps



c) Geometric Context

d) Depth Maps

Figs from Hoiem - Savarese 2011 book

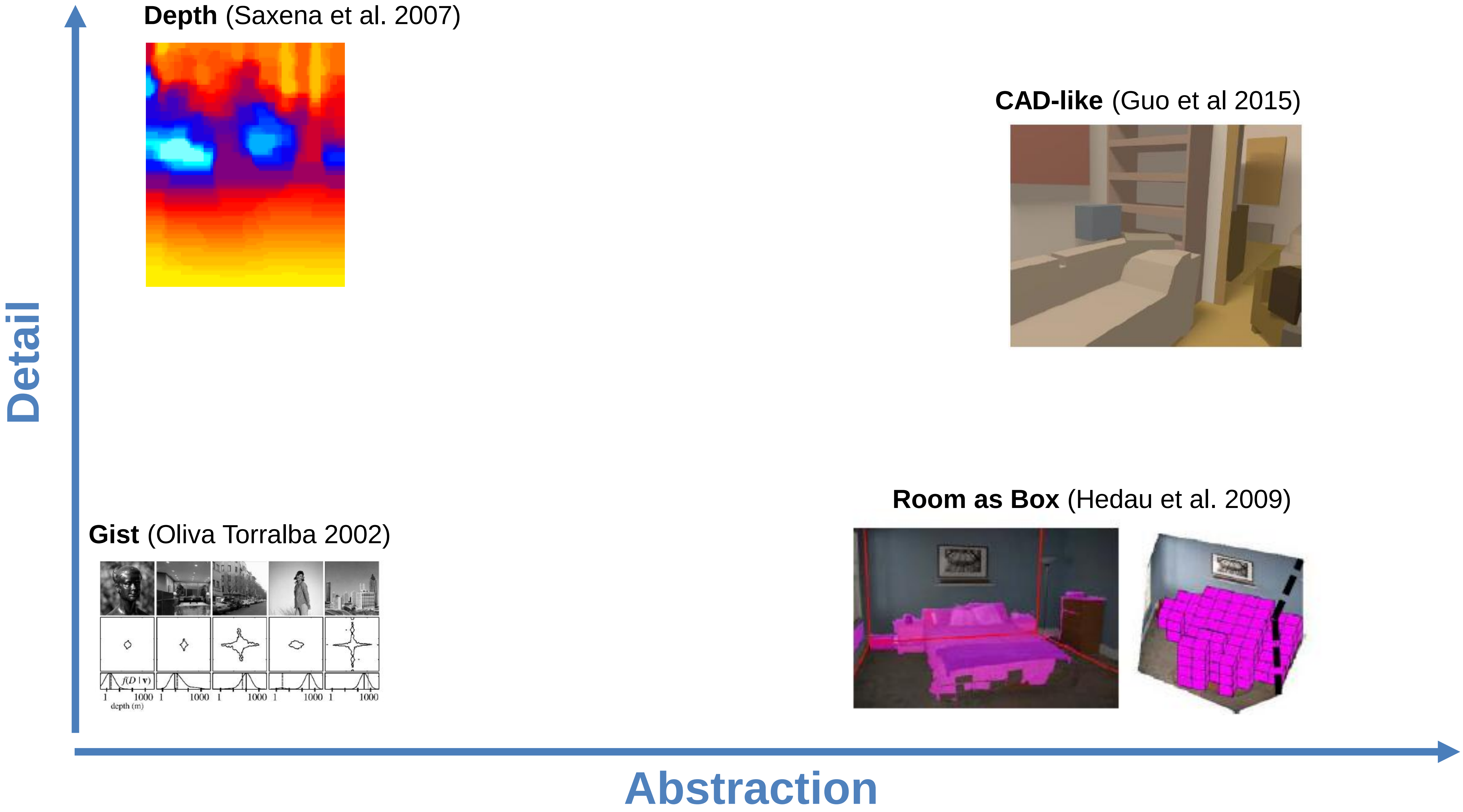



CAD-like: layout + objects

Figs from Hoiem - Savarese 2011 book

# Key Trade-offs

- Level of detail: rough “gist”, or detailed point cloud?
  - Precision vs. accuracy
  - Difficulty of inference
- Abstraction: depth at each pixel, or ground planes and walls?
  - What is it for: e.g., metric reconstruction vs. interaction


Depth (Saxena et al. 2007)
CAD-like (Guo et al 2015)
Room as Box (Hedau et al. 2009)
Gist (Oliva Torralba 2002)
f(D | v)
1 1000 1 1000 1 1000 1 1000 1 1000
depth (m)
Detail
Abstraction

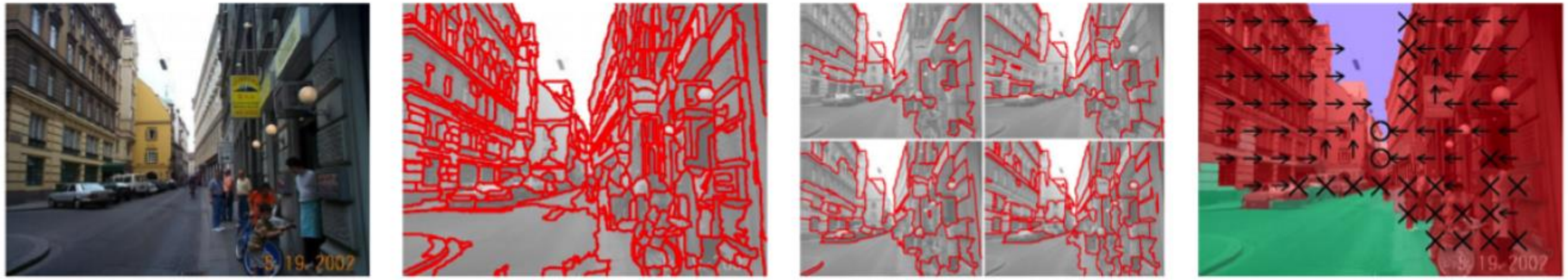
# Outdoor Scenes

- Highly irregular
- ~ Things sitting on the ground
- Ground-object boundary informs distance

# Surface Layout (“Geometric Context”)

| SURFACE CUES |
| --- |
| **Location and Shape** |
| L1. Location: normalized x and y, mean |
| L2. Location: normalized x and y, $10^{th}$ and $90^{th}$ pctl |
| L3. Location: normalized y wrt estimated horizon, $10^{th}$, $90^{th}$ pctl |
| L4. Location: whether segment is above, below, or straddles estimated horizon |
| L5. Shape: number of superpixels in segment |
| L6. Shape: normalized area in image |
| **Color** |
| C1. RGB values: mean |
| C2. HSV values: C1 in HSV space |
| C3. Hue: histogram (5 bins) |
| C4. Saturation: histogram (3 bins) |
| **Texture** |
| T1. LM filters: mean absolute response (15 filters) |
| T2. LM filters: histogram of maximum responses (15 bins) |
| **Perspective** |
| P1. Long Lines: (number of line pixels)/sqrt(area) |
| P2. Long Lines: percent of nearly parallel pairs of lines |
| P3. Line Intersections: histogram over 8 orientations, entropy |
| P4. Line Intersections: percent right of image center |
| P5. Line Intersections: percent above image center |
| P6. Line Intersections: percent far from image center at 8 orientations |
| P7. Line Intersections: percent very far from image center at 8 orientations |
| P8. Vanishing Points: (num line pixels with vertical VP membership)/sqrt(area) |
| P9. Vanishing Points: (num line pixels with horizontal VP membership)/sqrt(area) |
| P10. Vanishing Points: percent of total line pixels with vertical VP membership |
| P11. Vanishing Points: x-pos of horizontal VP - segment center (0 if none) |
| P12. Vanishing Points: y-pos of highest/lowest vertical VP wrt segment center |
| P13. Vanishing Points: segment bounds wrt horizontal VP |
| P14. Gradient: x, y center of mass of gradient magnitude wrt segment center |

- Compute superpixels
- For each superpixel compute several interesting features that make use of vanishing points, color, texture, lines…
- Train classifiers to predict several geometric classes: support, vertical sky



Input Superpixels Multiple Segmentations Surface Layout

Hoiem Efros Hebert 2005,2007

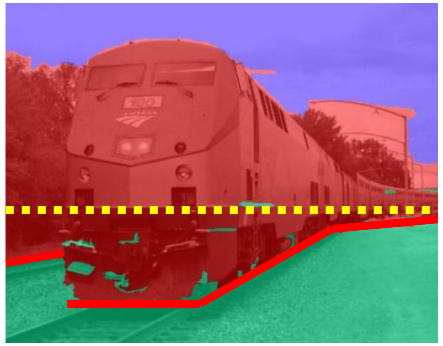
Slide from Sanja Fidler

# Automatic Photo Popup

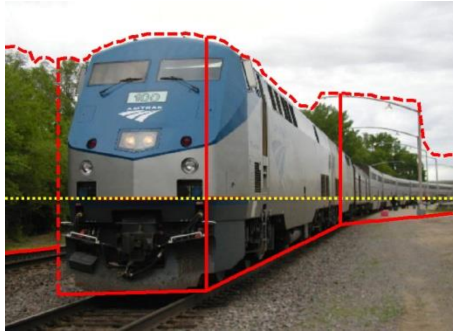
Labeled Image

Fit Ground-Vertical Boundary with Line Segments

Form Segments into Polylines



Cut and Fold

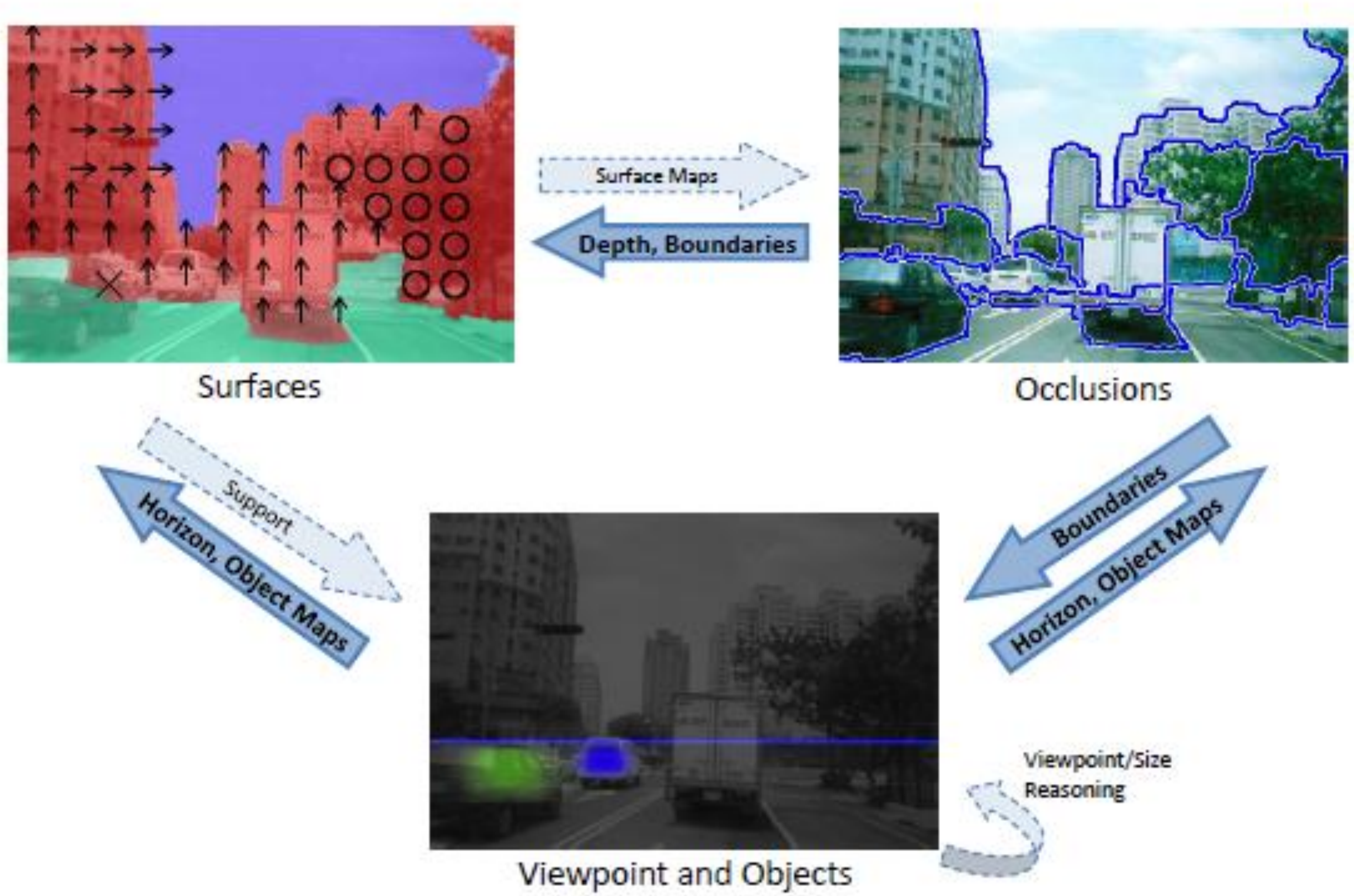


Final Pop-up Model

[Hoiem Efros Hebert 2005]

# Automatic Photo Popup

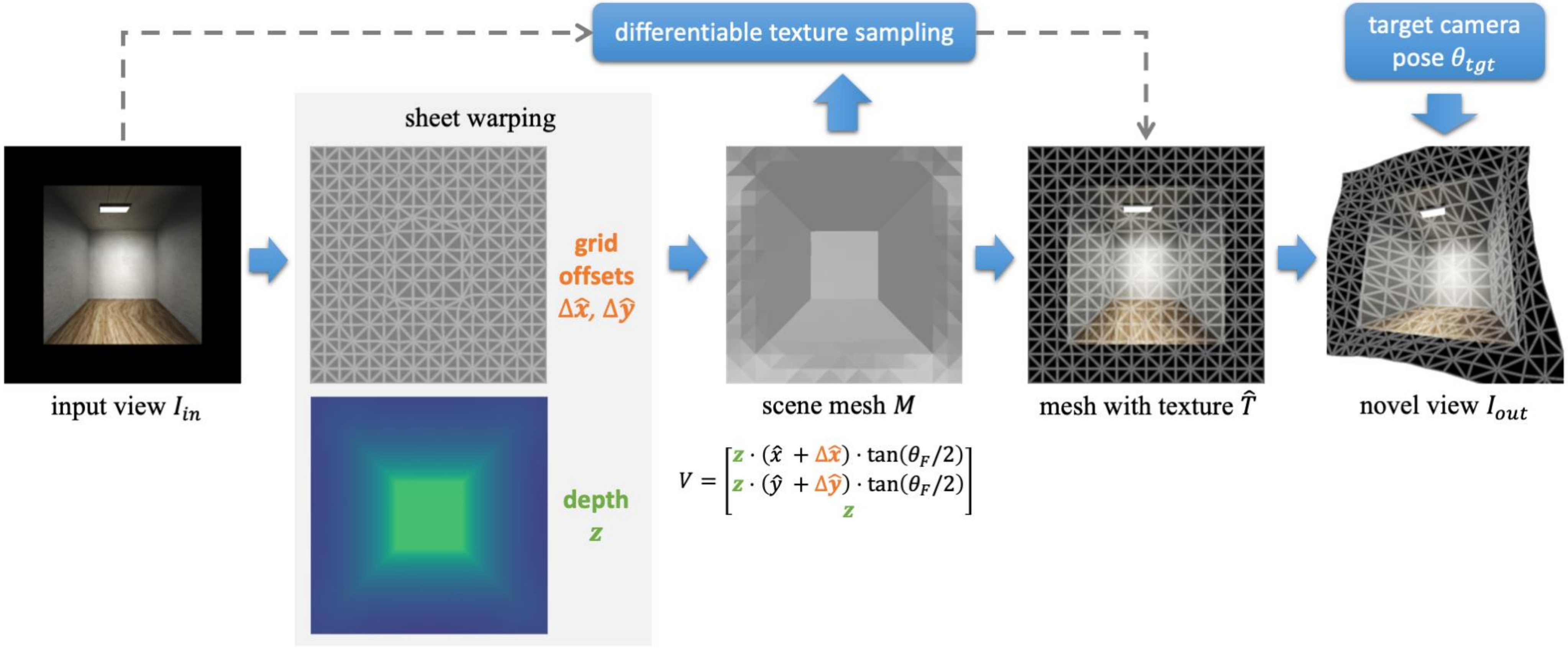
# Surface Layout + Boundaries + Viewpoint




[Hoiem et al. 2008]

# Worldsheet (Hu et al. ICCV 2021)

- https://worldsheet.github.io/
- https://www.youtube.com/watch?v=j5aT3zRxFlk



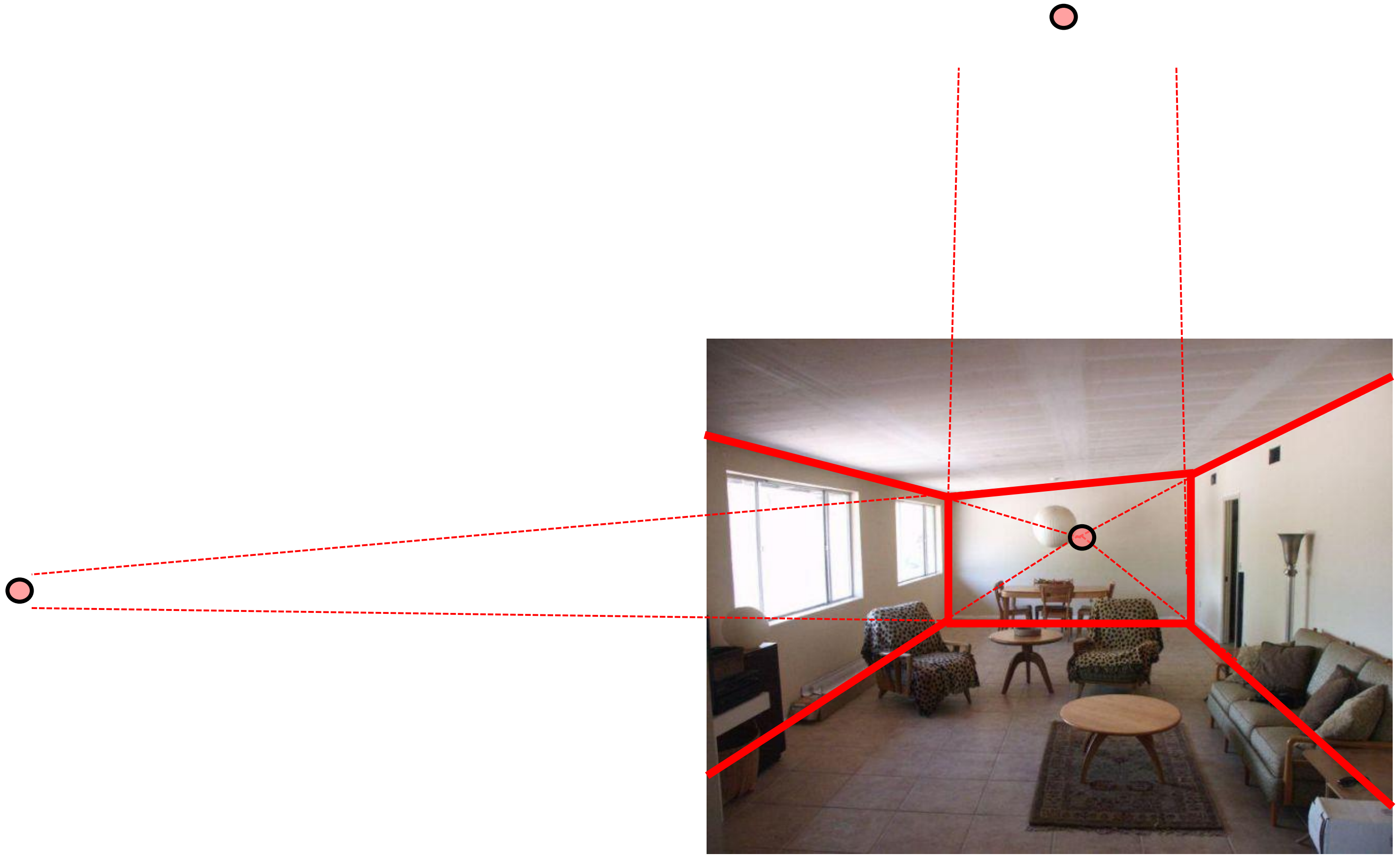
# Indoor scenes

- Highly regular
- Lots of things close to each other
- Things on other things
- Ground contact often not visible

# Simplest Model: Box Layout

Hedau Hoiem Forsyth, ICCV 2009

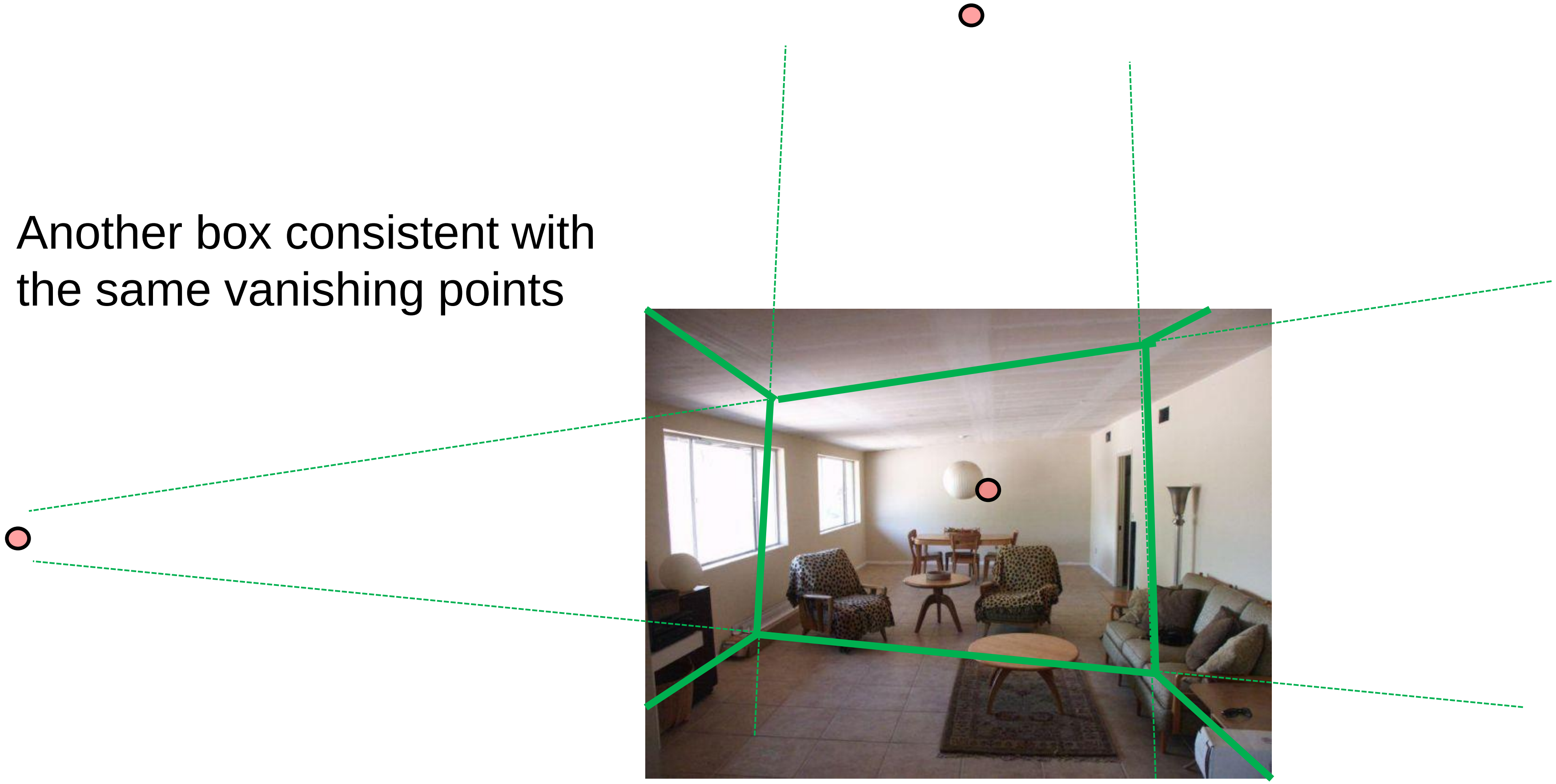
- Room is an oriented 3D box
  - Three vanishing points specify orientation
  - Two pairs of sampled rays specify position/size



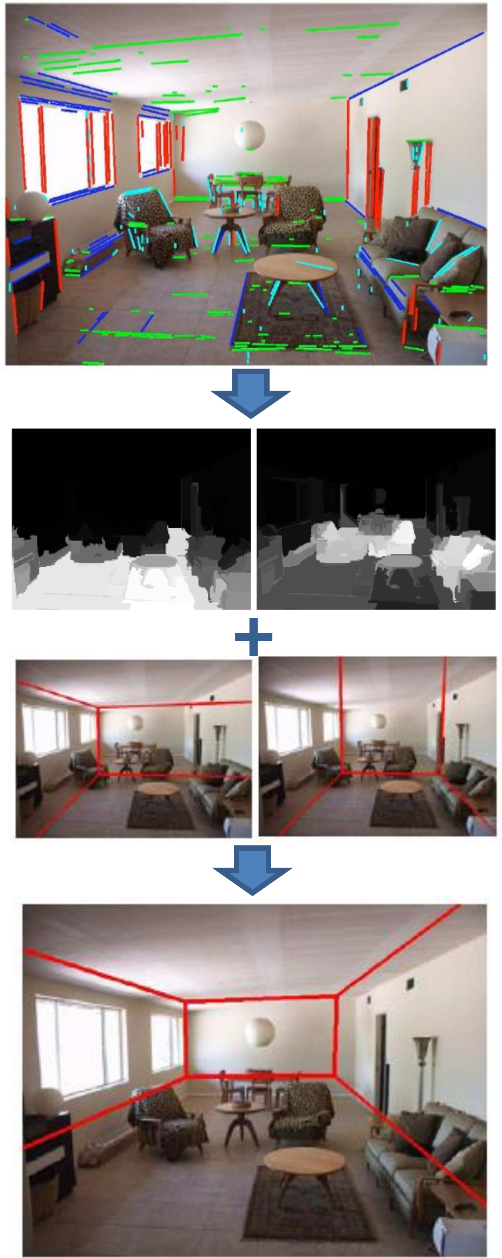
# Simplest Model: Box Layout

- Room is an oriented 3D box
  - Three vanishing points (VPs) specify orientation
  - Two pairs of sampled rays specify position/size

Another box consistent with the same vanishing points



# Box Layout Algorithm



1. Detect edges
2. Estimate 3 orthogonal vanishing points
3. Apply region classifier to label pixels with visible surfaces
   – Boosted decision trees on region based on color, texture, edges, position
4. Generate box candidates by sampling pairs of rays from VPs
5. Score each box based on edges and pixel labels
   – Learn score via structured learning
6. Jointly refine box layout and pixel labels to get final estimate

# Evaluation

- Dataset: 308 indoor images
  - Train with 204 images, test with 104 images






# Experimental results

Detected Edges

Surface Labels

Box Layout

Detected Edges

Surface Labels

Box Layout

# Experimental results

Detected Edges

Surface Labels

Box Layout

Detected Edges

Surface Labels

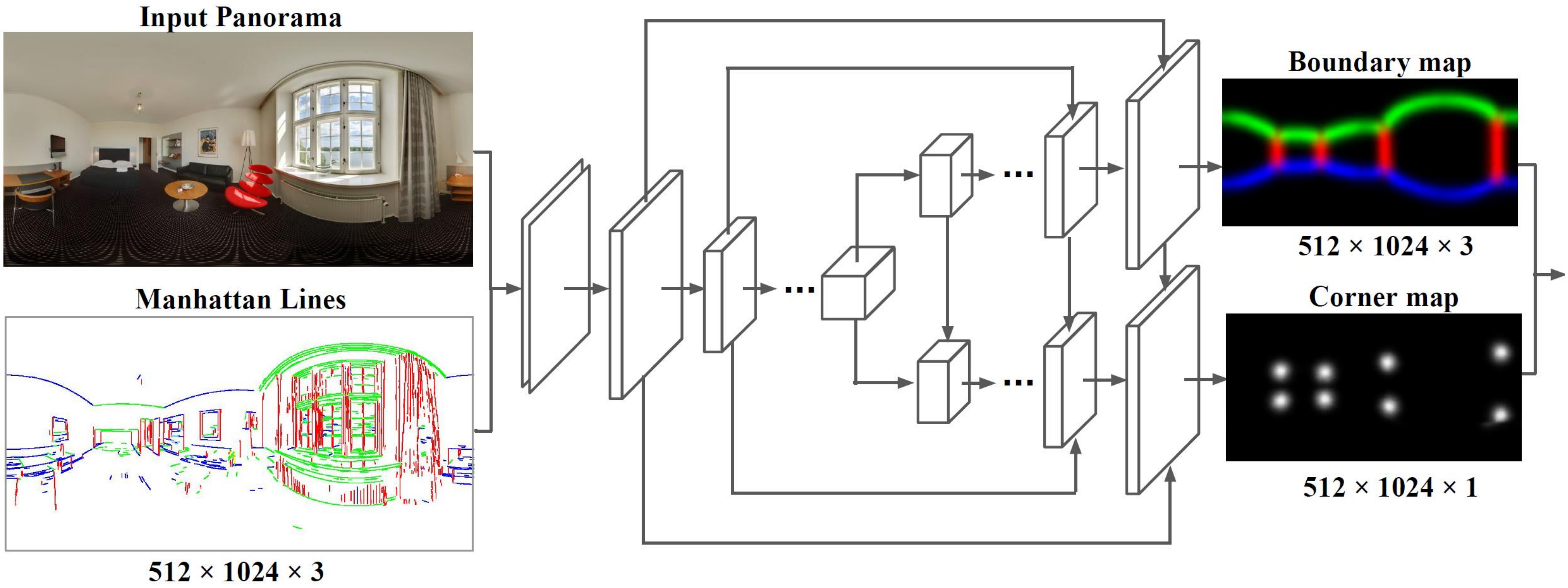
Box Layout

# Experimental results

- Joint reasoning of surface label / box layout helps
  - Pixel error: 26.5% → 21.2%
  - Corner error: 7.4% → 6.3%

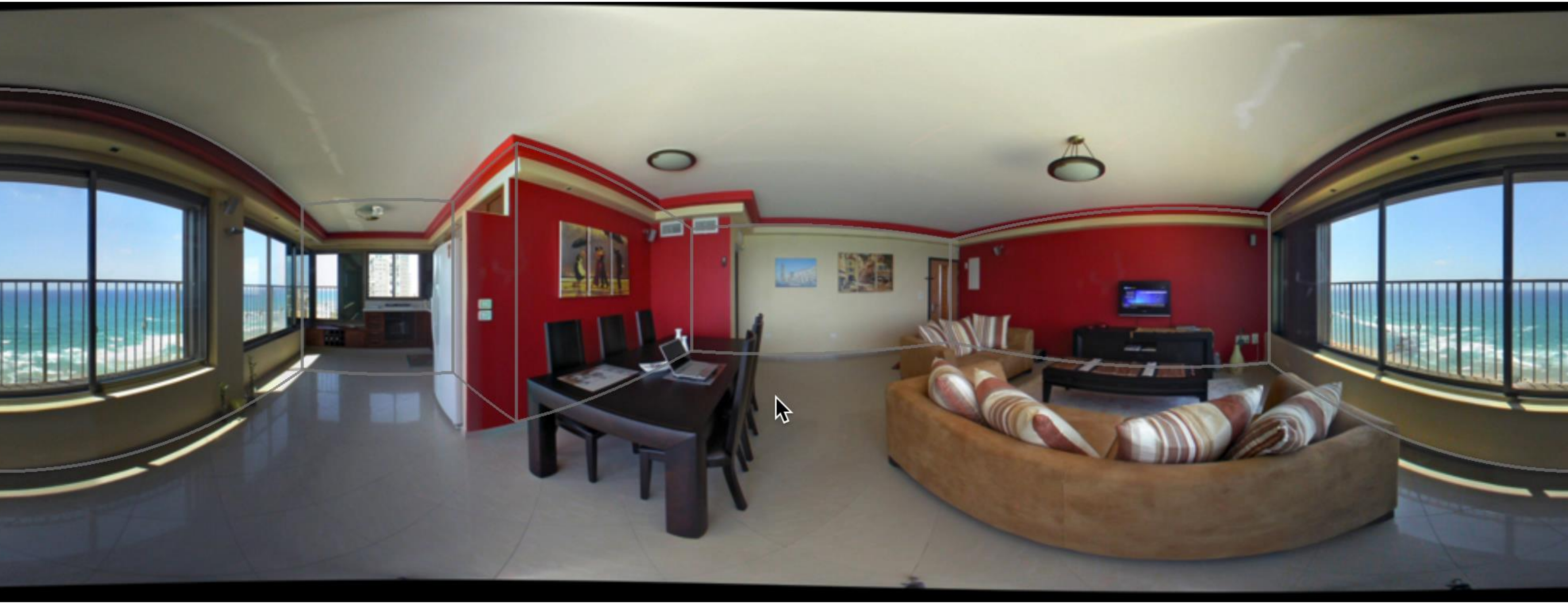
- Similar performance for cluttered and uncluttered rooms

# Similar idea for 360 images: “recognize” features of geometry, and fit simple model



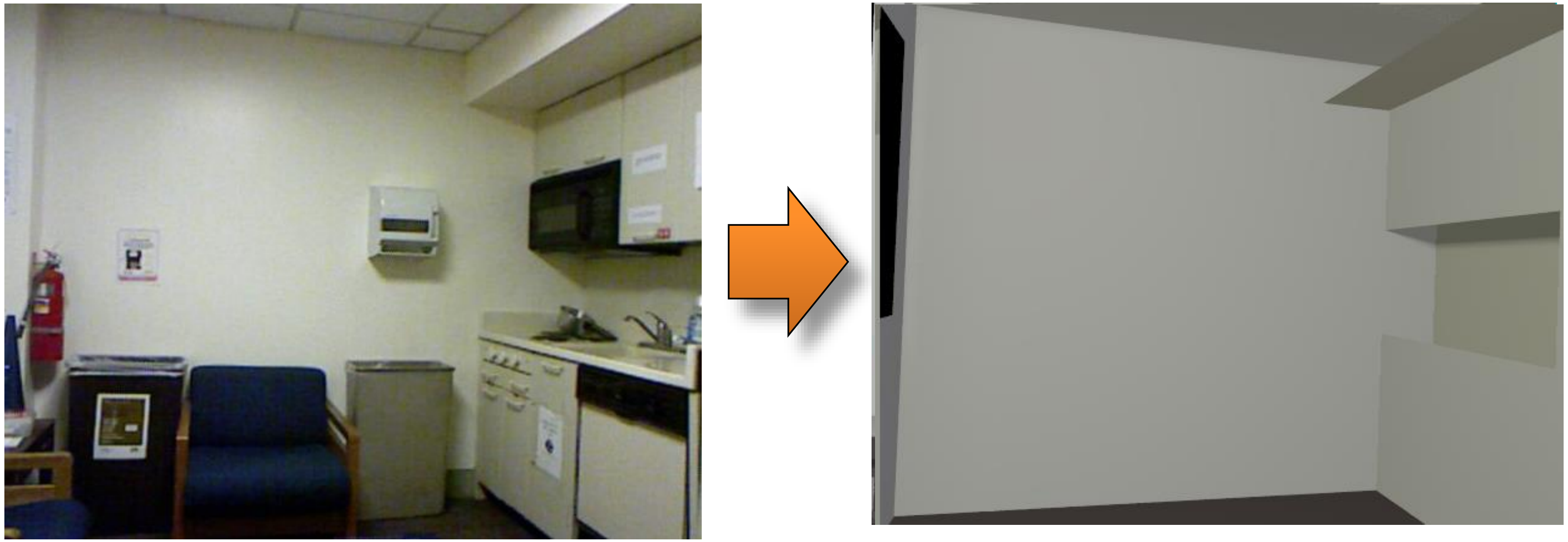

“LayoutNet”: Zou Colburn Shan Hoiem 2018  (collaboration with Zillow)

# LayoutNet example



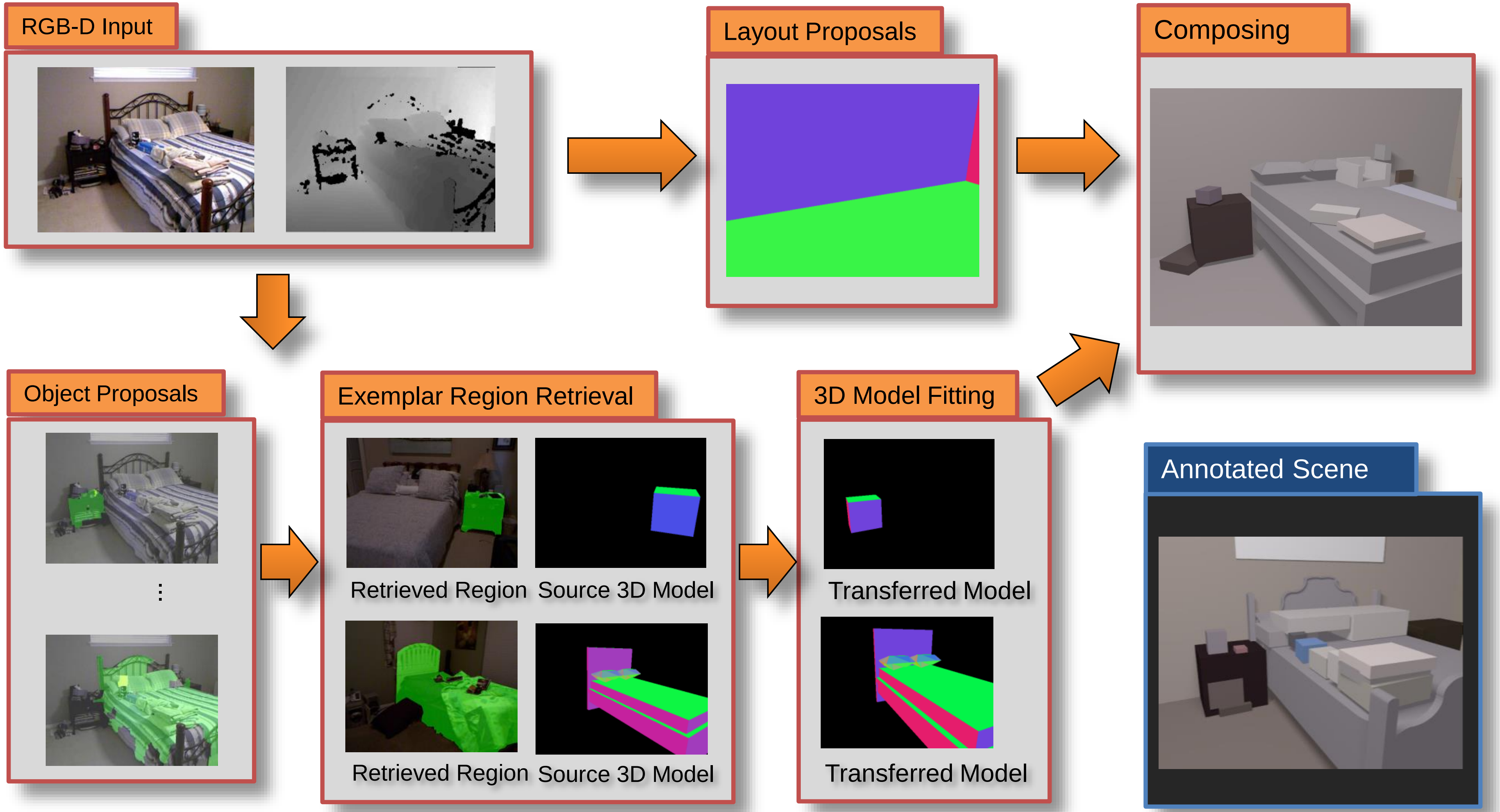
# Predicting complete models from RGBD

Key idea: create **complete** 3D scene hypothesis that is **consistent** with observed depth and appearance

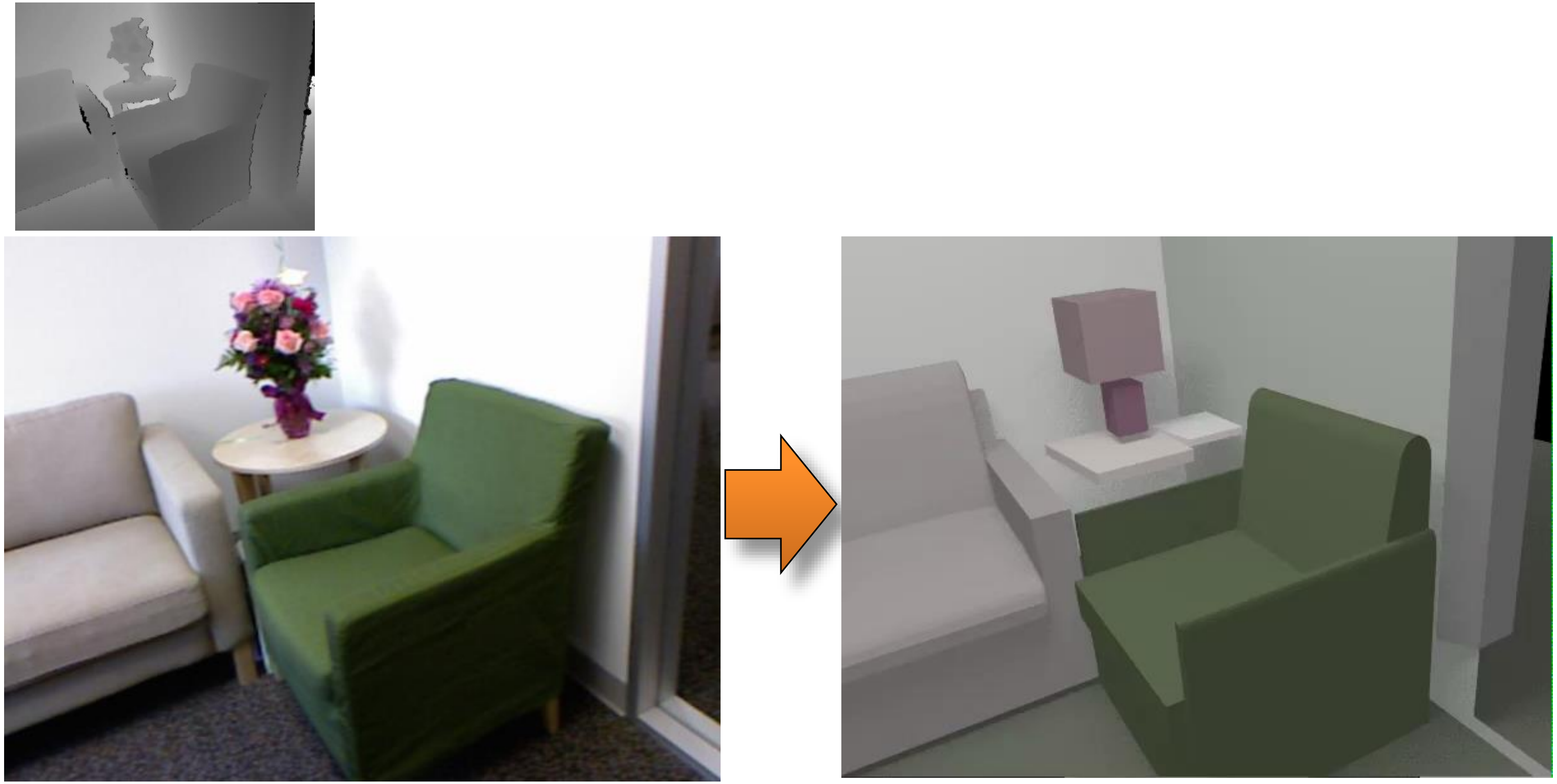


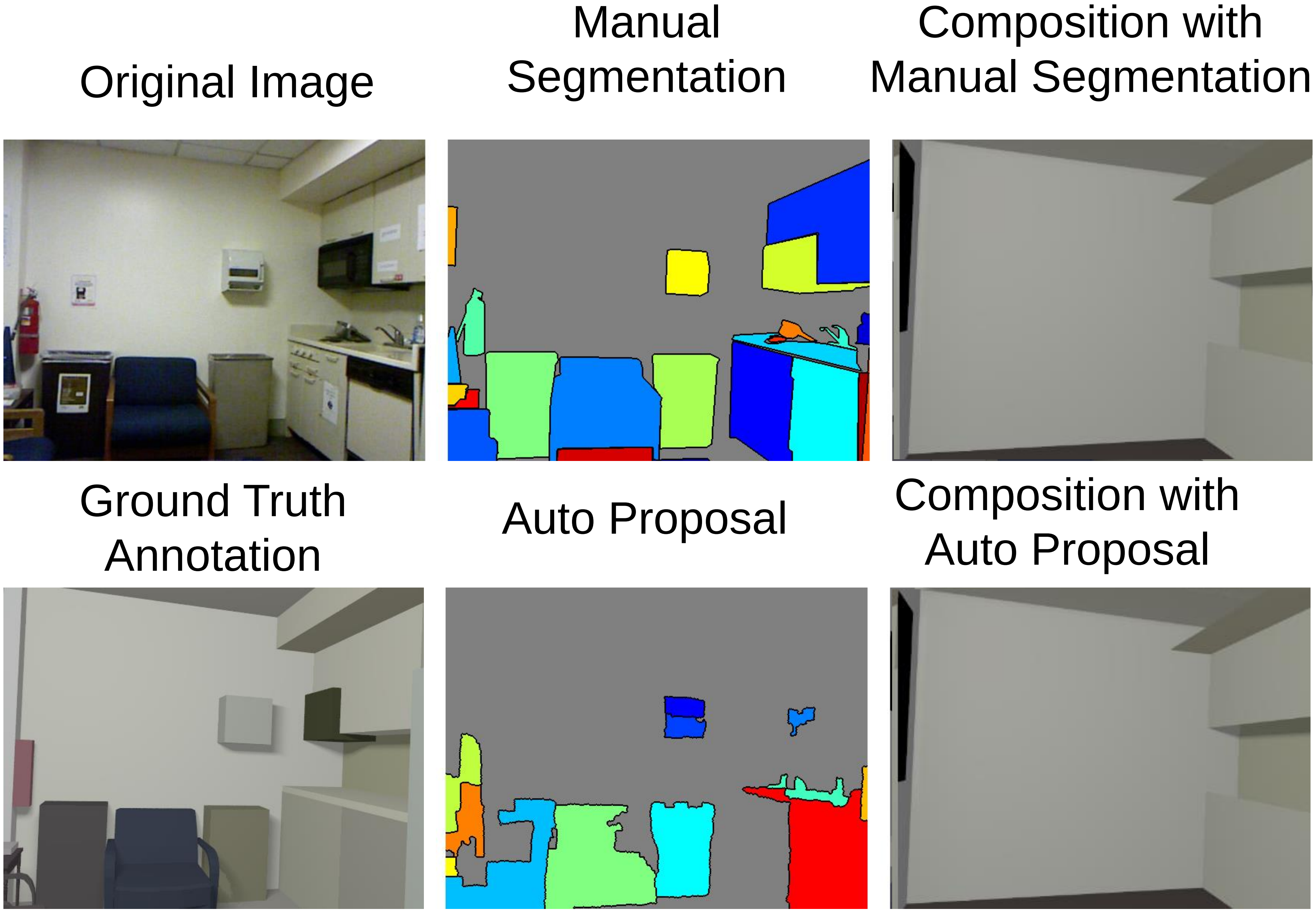
Guo Zou Hoiem 2015, 2016

# Overview of approach



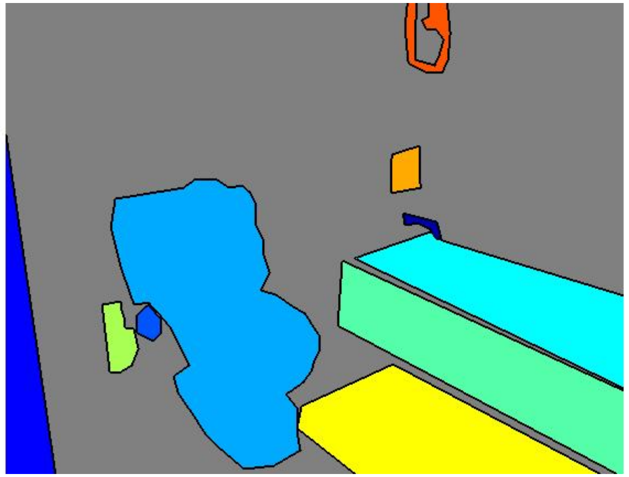
# Example result



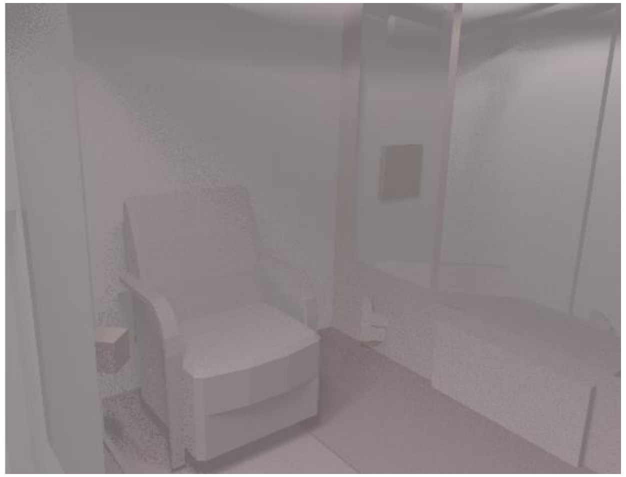

Original Image
Manual Segmentation
Composition with Manual Segmentation
Ground Truth Annotation
Auto Proposal
Composition with Auto Proposal

Original Image

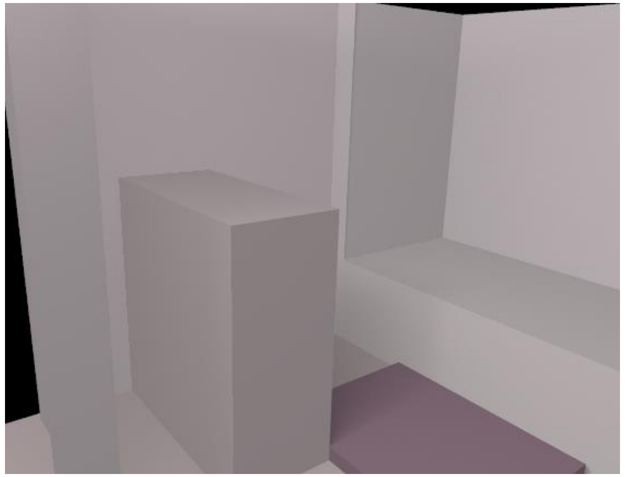
Manual Segmentation



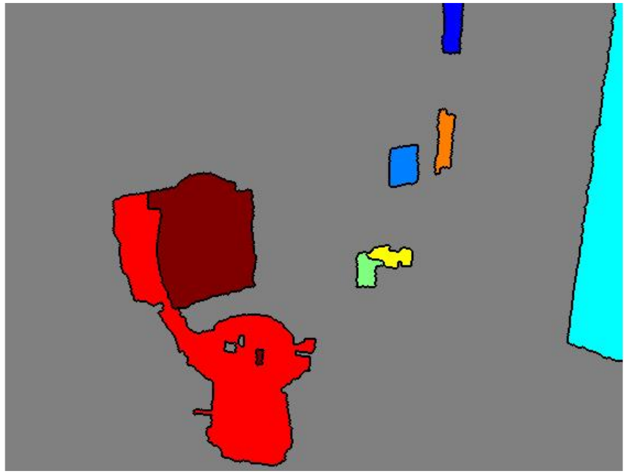
Composition w. Manual Segmentation



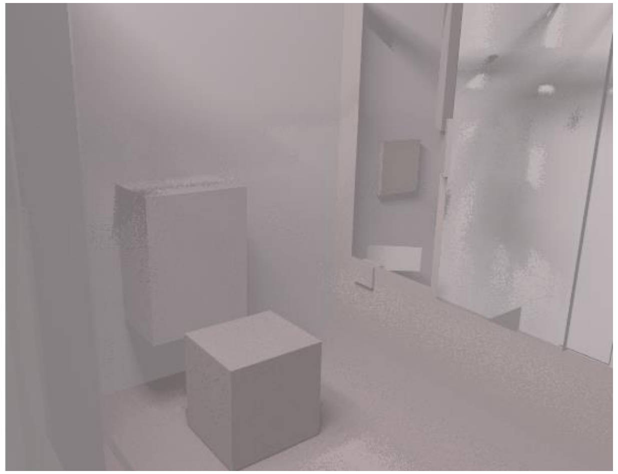
Ground Truth Annotation



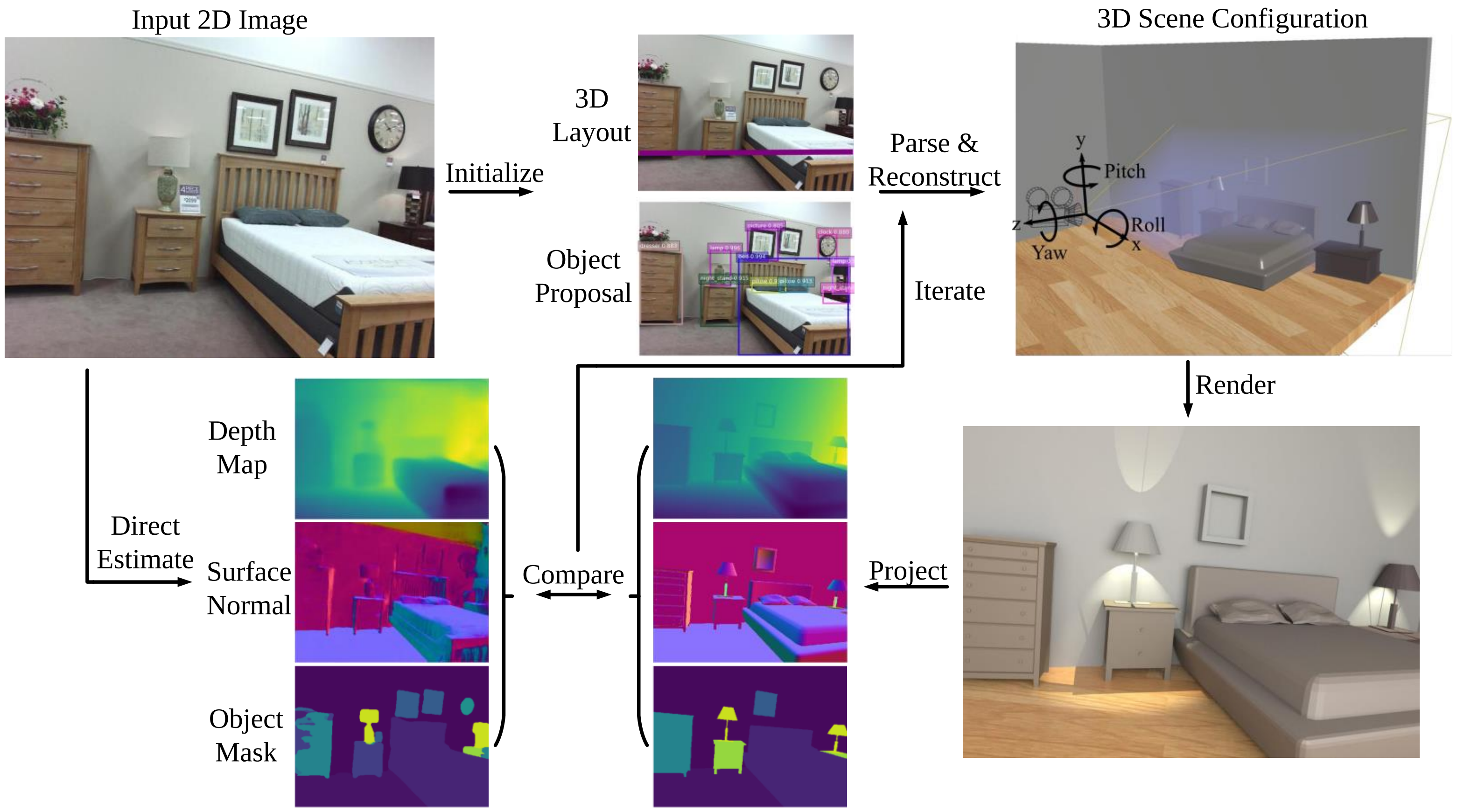
Auto Proposal



Composition w. Auto Proposal



# Scene parsing via rendering consistency

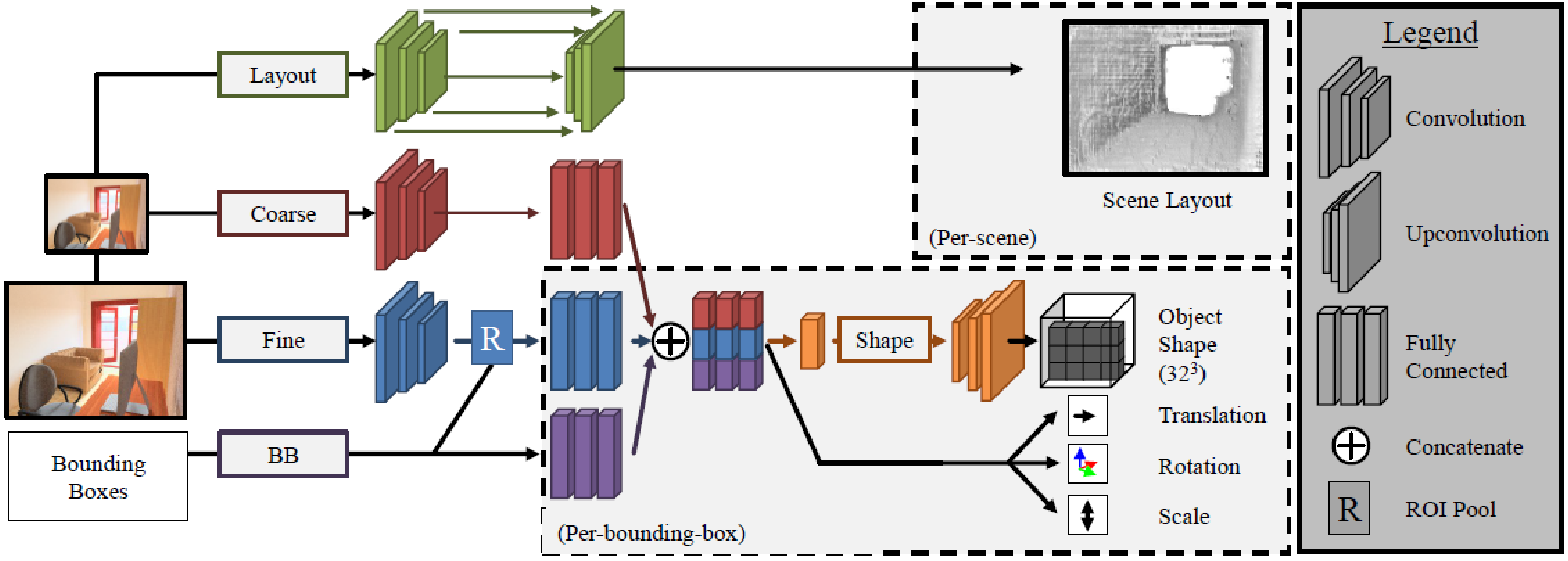


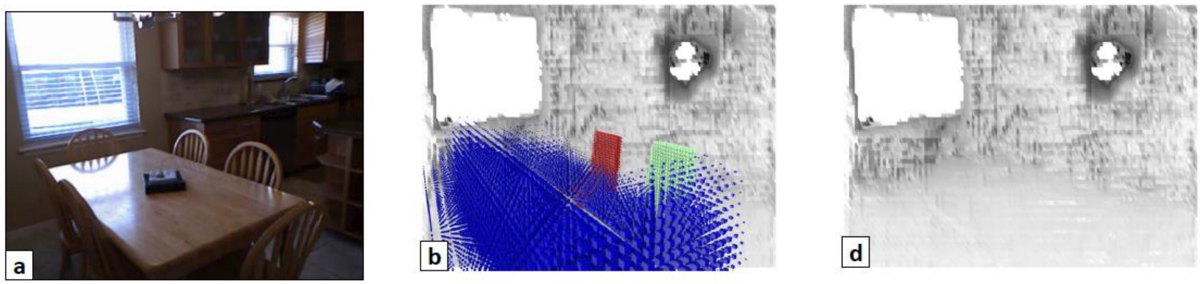

“Holistic 3D Scene Parsing”: Huang et al. 2018

# Factoring Shape, Pose, and Layout from the 2D Image of a 3D Scene

CVPR 2018

Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A. Efros, Jitendra Malik
University of California, Berkeley




Input Image    Layout + Object Shape/Pose    Layout Only

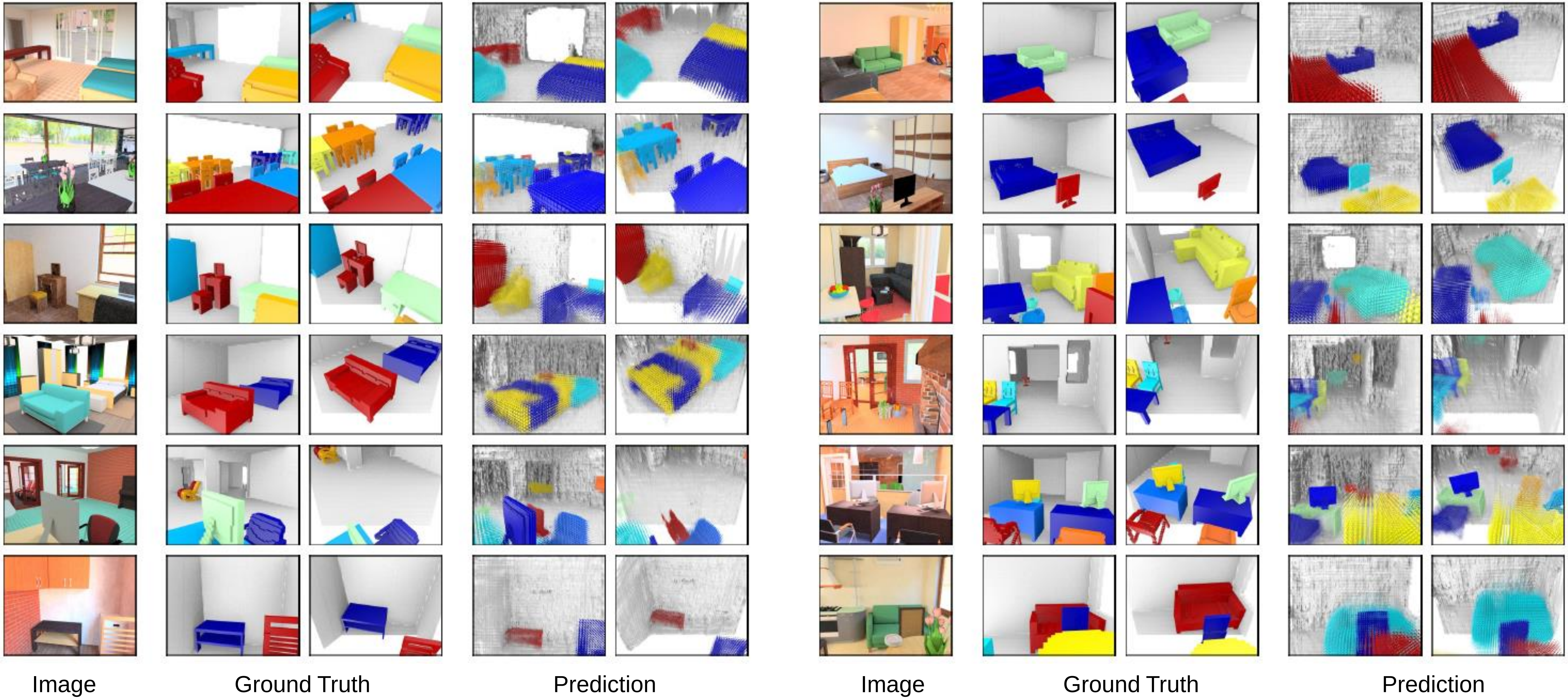


Image
Ground Truth
Prediction
Image
Ground Truth
Prediction

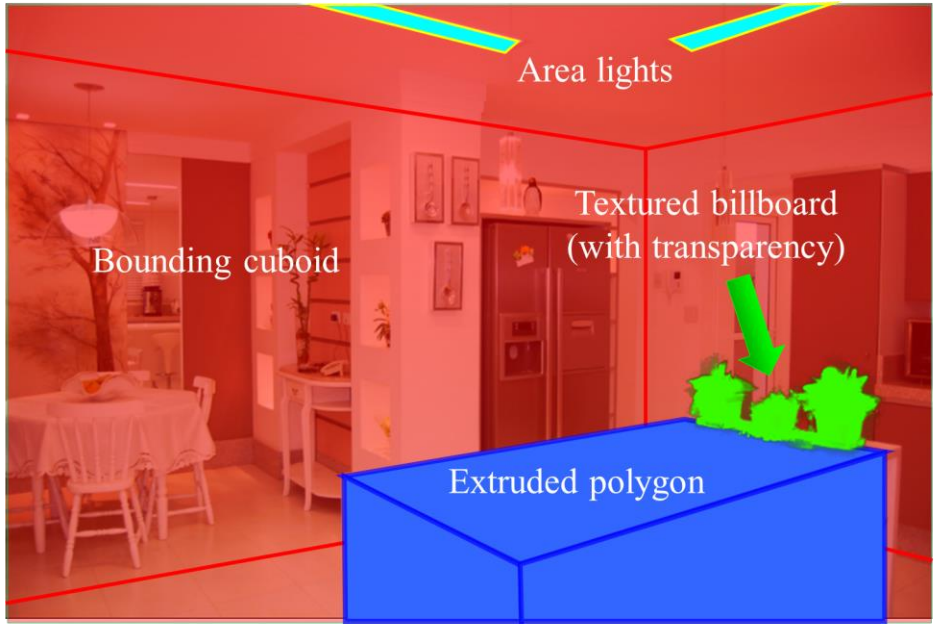
# Ultimate goal of 3D scene layout

- Recover layout surfaces (walls, floor, counters, etc.)
- Recognize objects where possible
- Estimate pose and shape of object(s) of interest
- Estimate space occupancy of all other objects (for movement)

# Can we combine representations of detail and structure?

Detailed Geometry
from Multiview

Structure and Semantics
from Single View



# Things to remember

- Most vision tasks are about *representing the image*, but 3D scene layout is about *representing the world*

- Difficult to maintain both precision and abstraction in a single representation – maybe best to maintain separate representations
  - Viewer-centric depth, normals, boundaries
  - Viewer-independent 3D layout of surfaces and shapes/positions of objects

- Biggest barrier to progress is complexity and challenge of evaluating, given that a central aim is to produce useful representations for unspecified downstream tasks