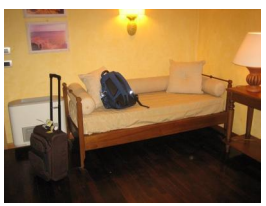
Understanding indoor scenes from images

Luca Del Pero

Main contributors: Joseph Schlecht, Emily Hartley
Joshua Bowdish, Bonnie Kermgard, Kobus Barnard
Other contributors: Ernesto Brau, Andrew Emmot,
Daniel Fried, Jinyan Guan

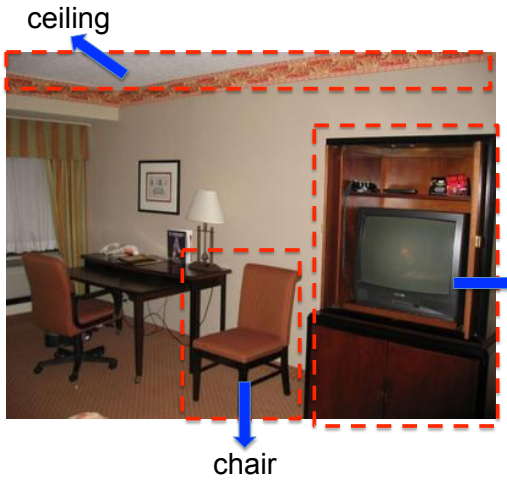
ISTA 352: Images: Past, Present, and Future
09/26/2012, Tucson, AZ

Goal: understand images of indoor environments





What does understanding an image mean?

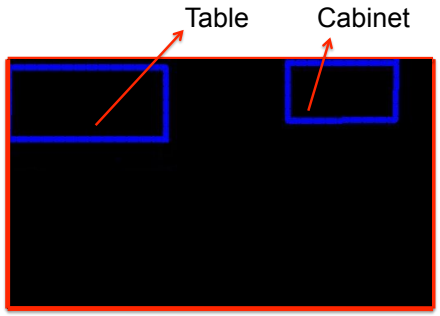


Understand content in 2D (indexing, image search)

cabinet

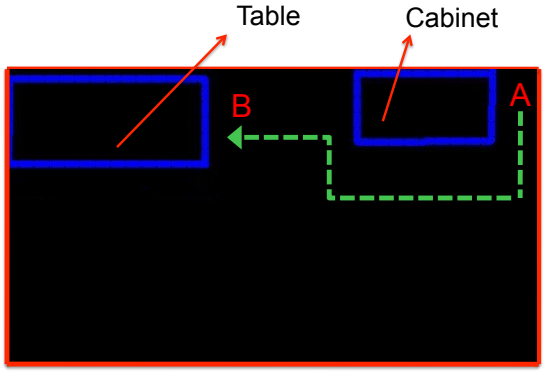
What does understanding an image mean?

Reconstructing the 3D world that generated an image:
What is where in 3D

3D information is useful (1)

Application: Unmanned navigation



Based on the information extracted from the image, plan a path from A to B

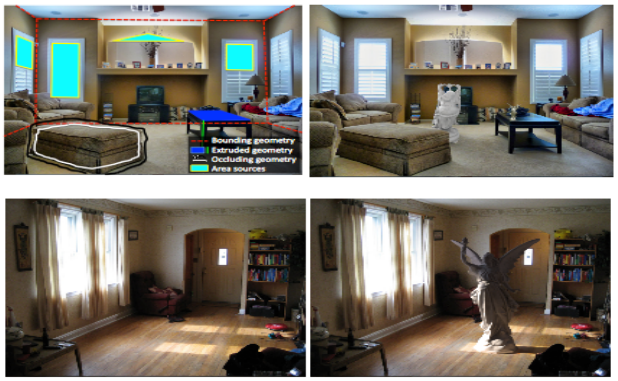
3D information is useful (2)

Application: Predicting human actions

Based on the geometry extracted from the image, find where humans could sit, lie, stand, etc. (Gupta et al., CVPR 2011)

3D information is useful (3)

Application: Computer graphics



Inserting synthetic objects in images, deal with occlusions (top) and shadows (bottom), Karsch et al, SIGGRAPH Asia 2011

3D information is useful (4)

Occlusions and context

- Thinking in 3D explains why the second chair is barely visible
- Chairs are positioned symmetrically around the table (context)

3D information is useful (5)

Prior information on the 3D world

- A 3D model can encode that tables have right angles
- 2D projections of right angles are not right!

Overview of our approach

- Recover 3D geometry of indoors scenes from indoor images
  - Recover room layout (walls, ceiling, floor)
  - Identify objects in it (beds, couches, doors, windows, etc.)

Flat image

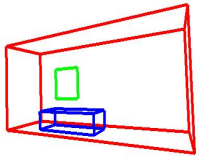
Overview of our approach

- Recover 3D geometry of indoors scenes from single images
  - Recover room layout (walls, ceiling, floor)
  - Identify objects in it (beds, couches, doors, windows, etc.)

Flat image

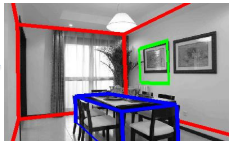
3D layout

A generative model for rooms



3D Layout

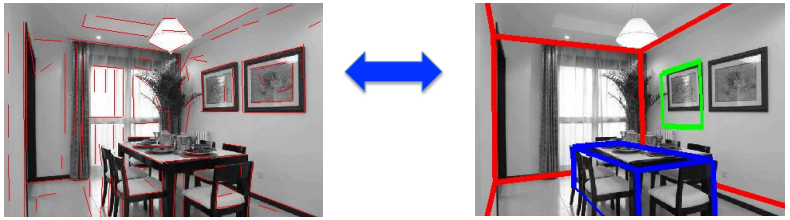
Camera



- Indoor images are generated by projecting the 3D scene
- Jointly estimate:
  - Camera parameters (unknown)
  - Room layout (size, position)
  - Number of objects (pieces of furniture, doors, windows,…)
  - Position, size and identity of each object

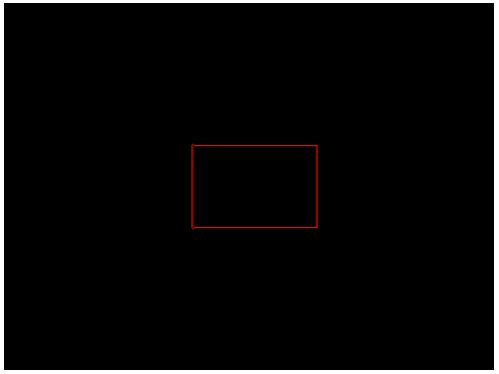
Finding the optimal parameters (inference)

- General idea: measure how well the model fits the data
- Example: image edges
- Edge likelihood compares:
  1) Edge points detected on the image plane
  2) Edge points generated by projecting the 3D layout under the camera



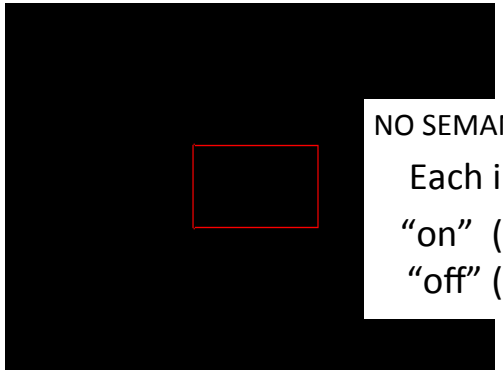
Toy example: find the rectangle!

- Edge points are detected when there is a change in color
- Given the detected edge points, where is the rectangle?



Toy example: find the rectangle!

- Edge points are detected when there is a change in color
- Given the detected edge points, where is the rectangle?

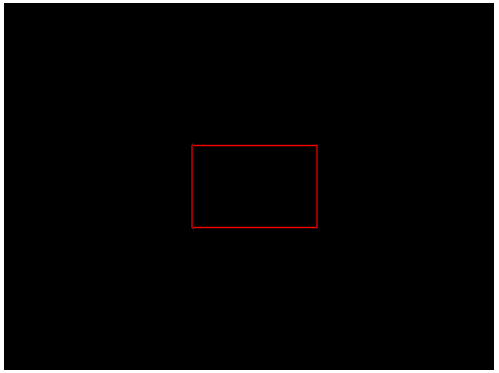


NO SEMANTICS!!!!

Each image point is "on" (edge) or "off" (no edge)

Toy example: find the rectangle!

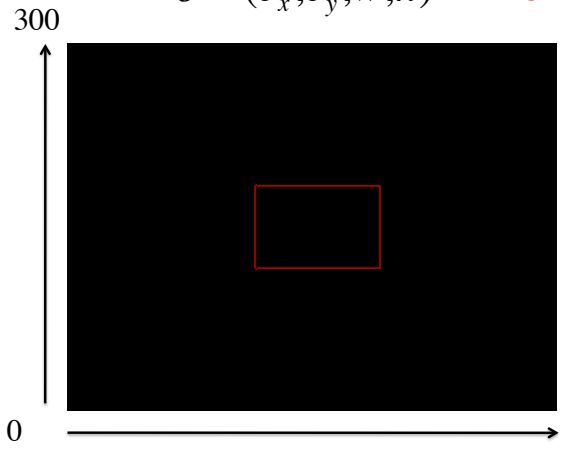
- What is a good model for the rectangle ?



Toy example: find the rectangle!

- What is a good model for the rectangle ?

$$\vartheta = (c_x, c_y, w, h)$$ MODELING

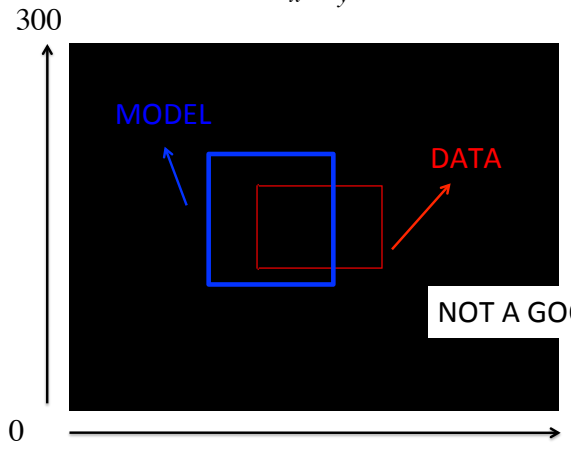


Toy example: find the rectangle!

- Find the parameters that best fit the image

Example $\vartheta = (c_x, c_y, w, h) = (120, 90, 80, 90)$

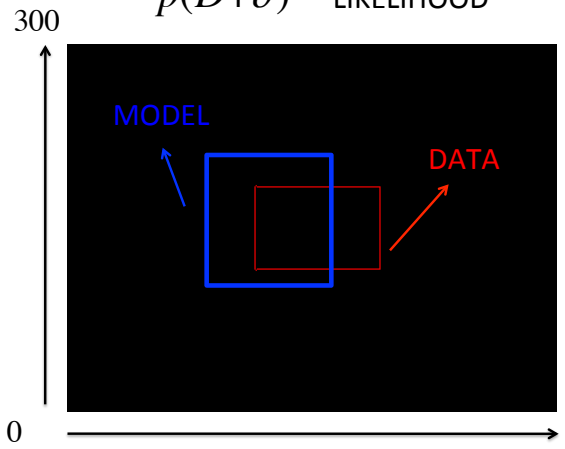
INFERENCE



Toy example: find the rectangle!

- Measure how well the hypothesis matches the data, e.g.

$$p(D \mid \vartheta)$$ LIKELIHOOD



Toy example: find the rectangle!

- Measure how well the hypothesis matches the data, e.g.

$$p(D \mid \vartheta)$$ LIKELIHOOD

$$p(D \mid \vartheta_1) \quad < \quad p(D \mid \vartheta_2)$$

Toy example: find the rectangle!

- Measure how well the hypothesis matches the data
  - Model edges are close to data edges $p(D \mid \vartheta) \uparrow$
  - More data edges than model edges $p(D \mid \vartheta) \downarrow$
  - More model edges than data edges $p(D \mid \vartheta) \downarrow$

Toy example: find the rectangle!

- INFERENCE: find $\vartheta = (c_x, c_y, w, h)$ maximizing $p(D \mid \vartheta)$
- Several techniques: gradient descent, sampling, etc.

  DEMO 1 → find c_y

  DEMO 2 → find all parameters
- Same ideas can be applied in 3D -> model edges come from 3D model hypothesis projected using camera hypothesis

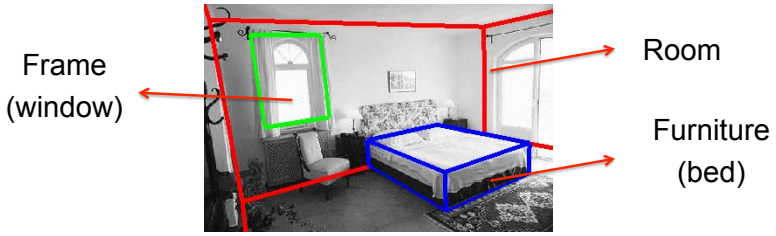
  DEMO 3 -> find 3D position of the cuboid

…and back to the original problem

- Indoor images are generated by projecting the 3D scene
- Jointly estimate:
  - Camera parameters (unknown)
  - Room layout (size, position)
  - Number of objects (pieces of furniture, doors, windows,…)
  - Position, size and identity of each objects

Room geometry

- Model room and objects as cuboids (blocks)



- Approximate furniture (couches, beds) with bounding boxes, frames (windows, doors) with thin blocks
- Furniture (blue) is on the floor, frames (green) are attached to a wall, objects can not overlap

Camera parameters

- Focal length and extrinsic parameters
- The camera determines the perspective distortion

Wrong camera

Correct camera

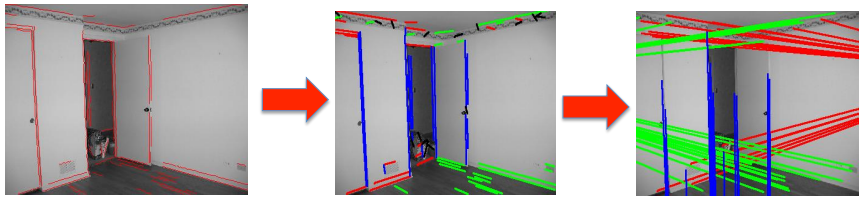
Inference

- Find model parameters that maximize the edge likelihood
- Challenges:
  - vast parameter space
  - very structured output (e.g. objects have to touch the floor, they cannot overlap, etc.)
- In general, recovering 3D from 2D is under constrained

Manhattan world assumption

- Most surfaces are aligned to three orthogonal directions
- This assumption enables reconstruction from a single image (2D to 3D)

Manhattan world assumption



Detect edges

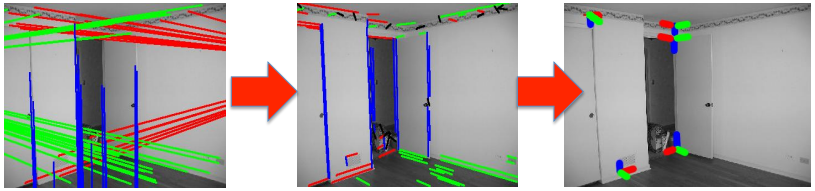
Group them based on convergence

- Compute camera parameters (e.g., focal length) from three vanishing points (analytically)
- Good initialization in most cases -> simpler inference

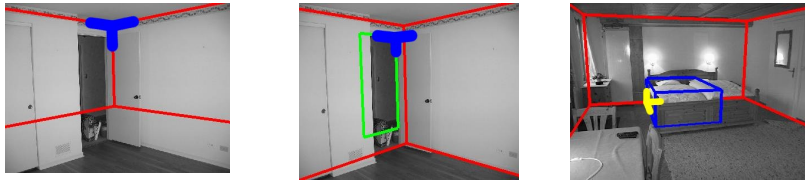
Data-driven inference

(let the data speak)

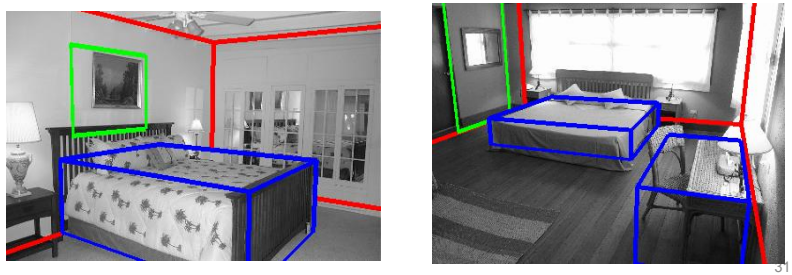
Detect corners



Propose 3D structure from corners

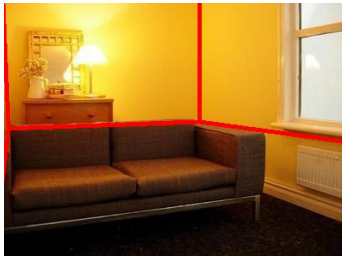


Results

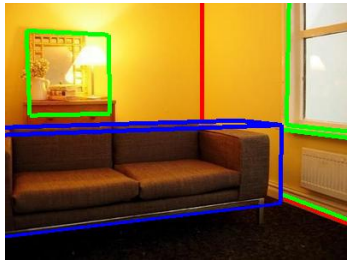


3D reasoning at work

Explaining occlusions

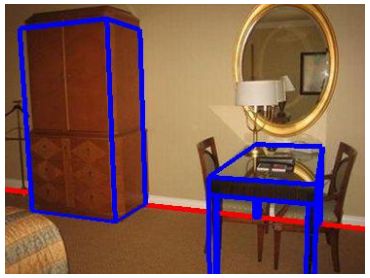


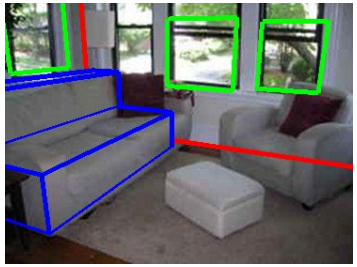
Fitting the room only



Fitting the room and the objects

Ongoing work





Use detailed models for furniture,
instead of simple blocks