Single-view 3D reasoning

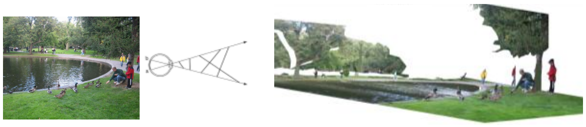
Lecturer: Bryan Russell

UW CSE 576, May 2013

Slides borrowed from Antonio Torralba, Alyosha Efros, Antonio Criminisi, Derek Hoiem, Steve Seitz, Stephen Palmer, Abhinav Gupta, James Coughlan, Aude Oliva, and others

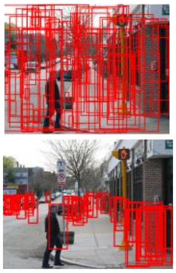
Depth Perception

The inverse problem

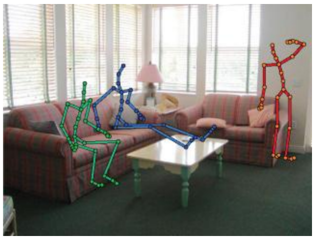


Slide by A. Torralba

Why is depth perception important?



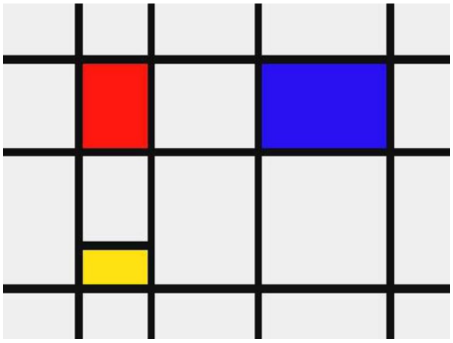
Context for object detection



Information on how to navigate in an environment

We don’t live in a 2D Mondrian world

Nearby pixels are close in 2D



Nearby pixels in 2D can be far away in 3D

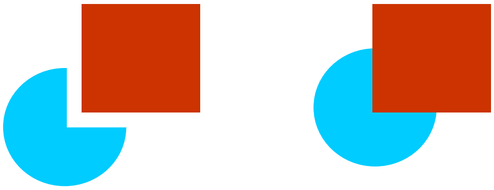
What are clues for recovering depth information from a single image?

Edge interpretation

Simple and powerful cue, but hard to make it work in practice…

Slide by A. Torralba

Interposition / occlusion



Slide by A. Torralba

Texture Gradient

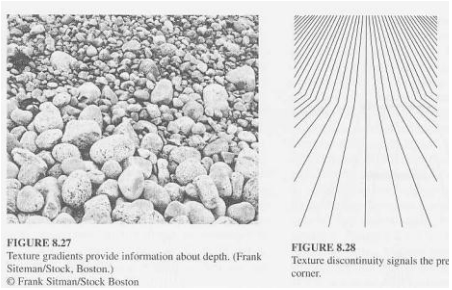


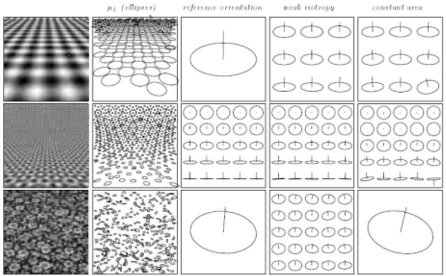
FIGURE 8.27
Texture gradients provide information about depth. (Frank Siteman/Stock, Boston.)
© Frank Sitman/Stock Boston

FIGURE 8.28
Texture discontinuity signals the pre
corner.

A Witkin. Recovering Surface Shape and Orientation from Texture (1981)

Slide by A. Torralba

Texture Gradient

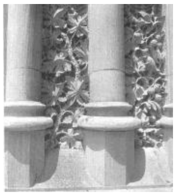


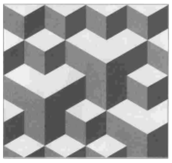
Shape from Texture from a Multi-Scale Perspective. Tony Lindeberg and Jonas Garding. ICCV 93

Slide by A. Torralba

Shading

- Based on 3 dimensional modeling of objects in light, shade and shadows.

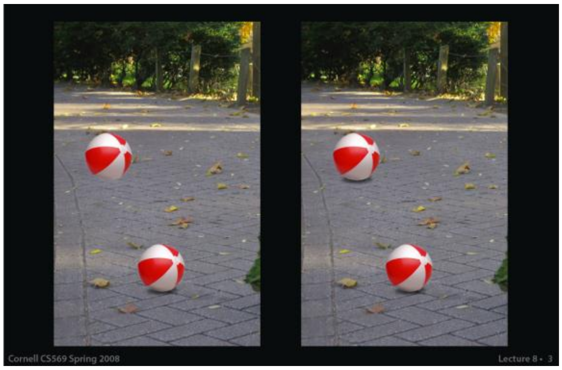




- Perception of depth through shading alone is always subject to the concave/convex inversion. The pattern shown can be perceived as stairsteps receding towards the top and lighted from above, or as an overhanging structure lighted from below.

Slide by A. Torralba

Shadows



Slide by Steve Marschner

http://www.cs.cornell.edu/courses/cs569/2008sp/schedule.stm

Atmospheric perspective

- Based on the effect of air on the color and visual acuity of objects at various distances from the observer.
- Consequences:
  - Distant objects appear bluer
  - Distant objects have lower contrast.

Slide by A. Torralb

Atmospheric perspective

http://encarta.msn.com/medias_761571997/Perception_(psychology).html

Slide by A. Torralb

Claude Lorrain (artist)
French, 1600 - 1682
***Landscape with Ruins, Pastoral Figures, and Trees*, 1643/1655**

Slide by A. Torralb

Linear Perspective

Based on the apparent convergence of parallel lines to common vanishing points with increasing distance from the observer.

(Gibson : “perspective order”)

In Gibson’s term, perspective is a characteristic of the visual field rather than the visual world. It approximates how we see (the retinal image) rather than what we see, the objects in the world.

Perspective : a representation that is specific to one individual, in one position in space and one moment in time (a powerful immediacy).

Is perspective a universal fact of the visual retinal image ? Or is perspective something that is learned ?

Simple and powerful cue, and easy to make it work in practice…

Slide by A. Torralb

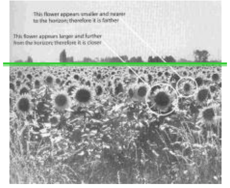
Linear Perspective

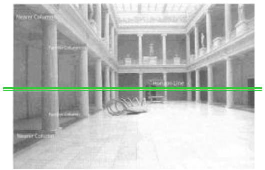
(c) 2006 Walt Anthony

Slide by A. Torralb

Distance from the horizon line

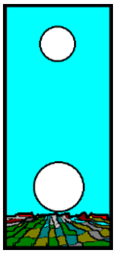
- Based on the tendency of objects to appear nearer the horizon line with greater distance to the horizon.
- Objects approach the horizon line with greater distance from the viewer. The base of a nearer column will appear lower against its background floor and further from the horizon line. Conversely, the base of a more distant column will appear higher against the same floor, and thus nearer to the horizon line.





Slide by A. Torralb

Moon illusion



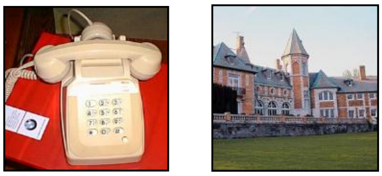
Slide by A. Torralb

Absolute (monocular) depth cues

Are there any monocular cues that can give us absolute depth from a single image?

Slide by A. Torralb

Familiar size



Which “object” is closer to the camera?
How close?

Slide by A. Torralb

Familiar size

Apparent reduction in size of objects at a greater distance from the observer

Size perspective is thought to be conditional, requiring knowledge of the objects.

But, material textures also get smaller with distance, so possibly, no need of perceptual learning ?

Slide by A. Torralb

Perspective vs. familiar size

3D percept is driven by the scene, which imposes its ruling to the objects

Slide by A. Torralb

Scene vs. objects

What do you see? A big apple or a small room?

I see a big apple and a normal room
The scene seems to win again?

[*The Listening Room* Rene Magritte]

Slide by A. Torralba

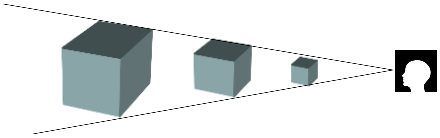
Scene vs. objects

Slide by A. Torralba

[*Personal Values* Rene Magritte]

Depth Perception from Image Structure

Mean depth refers to a global measurement of the mean distance between the observer and the main objects and structures that compose the scene.

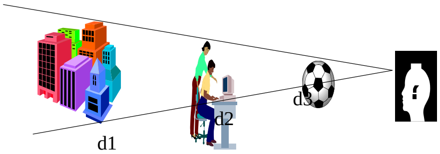


Stimulus ambiguity: the three cubes produce the same retinal image. Monocular information cannot give absolute depth measurements. Only relative depth information such as shape from shading and junctions (occlusions) can be obtained.

Slide by A. Torralba

Depth Perception from Image Structure

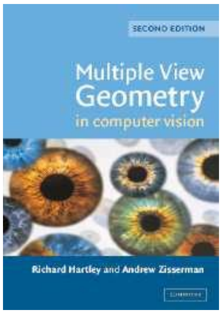
However, nature (and man) do not build in the same way at different scales.



If d1>>d2>>d3 the structures of each view strongly differ. **Structure** provides monocular information about the scale (mean depth) of the space in front of the observer.

Slide by A. Torralba

Today's class: reasoning about perspective cues via projective geometry



Readings

- Hartley and Zisserman textbook
- Mundy, J.L. and Zisserman, A., Geometric Invariance in Computer Vision, Appendix: Projective Geometry for Machine Vision, MIT Press, Cambridge, MA, 1992, **(read 23.1 - 23.5, 23.10)**
  - available online: http://www.cs.cmu.edu/~ph/869/papers/zisser-mundy.pdf

Projective geometry—what's it good for?

Uses of projective geometry

- Drawing
- Measurements
- Mathematics for projection
- Undistorting images
- Focus of expansion
- Camera pose estimation, match move
- Object recognition

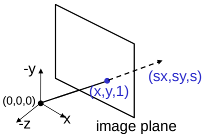
The projective plane

Why do we need homogeneous coordinates?

- represent points at infinity, homographies, perspective projection, multi-view relationships

What is the geometric intuition?

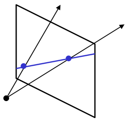
- a point in the image is a *ray* in ***projective space***



- Each *point* (x,y) on the plane is represented by a *ray* (sx,sy,s)
  - all points on the ray are equivalent: (x, y, 1) E (sx, sy, s)

Projective lines

What does a line in the image correspond to in projective space?



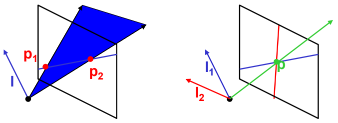
- A line is a *plane* of rays through origin
  - all rays (x,y,z) satisfying: ax + by + cz = 0

$$\text{in vector notation:}\quad 0 = \underset{\mathbf{l}}{\begin{bmatrix} a & b & c \end{bmatrix}} \underset{\mathbf{p}}{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}$$

- A line is also represented as a homogeneous 3-vector **l**

Point and line duality

- A line **l** is a homogeneous 3-vector = [a b c]
- It is ∞ to every point (ray) **p** on the line: **l p**=0



What is the line **l** spanned by rays $\mathbf{p_1}$ and $\mathbf{p_2}$?

- **l** is ∞ to $\mathbf{p_1}$ and $\mathbf{p_2}$ ® $\mathbf{l = p_1 \times p_2}$
- **l** is the plane normal

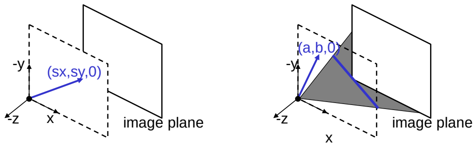
What is the intersection of two lines $\mathbf{l_1}$ and $\mathbf{l_2}$?

- **p** is ∞ to $\mathbf{l_1}$ and $\mathbf{l_2}$ ® $\mathbf{p = l_1 \times l_2}$

Points and lines are *dual* in projective space

- given any formula, can switch the meanings of points and lines to get another formula

Ideal points and lines



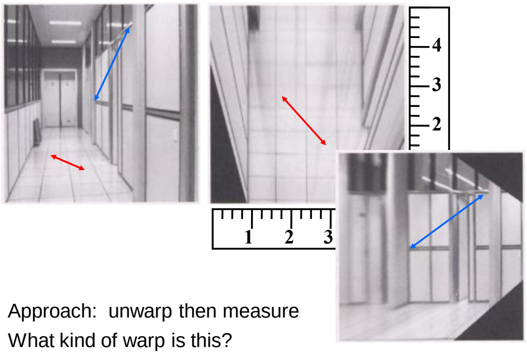
Ideal point ("point at infinity")

- p E (x, y, 0) – parallel to image plane
- It has infinite image coordinates

Ideal line

- l E (a, b, 0) – parallel to image plane
  - Corresponds to a line in the image (finite coordinates)
    - goes through image origin (*principal point*)

Measurements on planes



Approach: unwarp then measure

What kind of warp is this?

Homographies

Perspective projection of a plane

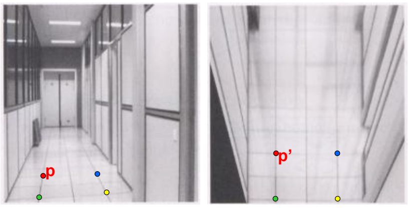
- Lots of names for this:
  - **homography**, texture-map, colineation, planar projective map
- Modeled as a 2D warp using homogeneous coordinates

$$\underset{\mathbf{p'}}{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}} = \underset{\mathbf{H}}{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}} \underset{\mathbf{p}}{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}$$

To apply a homography **H**

- Compute **p'** = **Hp** (regular matrix multiply)
- Convert **p'** from homogeneous to image coordinates
  - divide by w (third) coordinate

Image rectification



To unwarp (rectify) an image

- solve for homography **H** given **p** and **p'**
- solve equations of the form: w**p'** = **Hp**
  - linear in unknowns: w and coefficients of **H**
  - **H** is defined up to an arbitrary scale factor
  - how many points are necessary to solve for **H**?

Solving for homographies

$$\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$\begin{aligned} x_i' &= \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}} \\ y_i' &= \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}} \end{aligned}$$

$$\begin{aligned} x_i'(h_{20}x_i + h_{21}y_i + h_{22}) &= h_{00}x_i + h_{01}y_i + h_{02} \\ y_i'(h_{20}x_i + h_{21}y_i + h_{22}) &= h_{10}x_i + h_{11}y_i + h_{12} \end{aligned}$$

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -y_i'x_i & -y_i'y_i & -y_i' \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

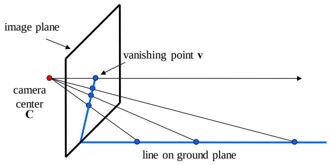
Solving for homographies

$$\underset{2n \times 9}{\mathbf{A}}\;\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1'x_1 & -x_1'y_1 & -x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y_1'x_1 & -y_1'y_1 & -y_1' \\ & & & & \vdots & & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_n'x_n & -x_n'y_n & -x_n' \\ 0 & 0 & 0 & x_n & y_n & 1 & -y_n'x_n & -y_n'y_n & -y_n' \end{bmatrix} \underset{9}{\mathbf{h}}\;\begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \underset{2n}{\mathbf{0}}\;\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Defines a least squares problem: minimize $\|\mathbf{A}\mathbf{h} - \mathbf{0}\|^2$

- Since **h** is only defined up to scale, solve for unit vector $\hat{\mathbf{h}}$
- Solution: $\hat{\mathbf{h}}$ = eigenvector of $\mathbf{A}^T\mathbf{A}$ with smallest eigenvalue
- Works with 4 or more points

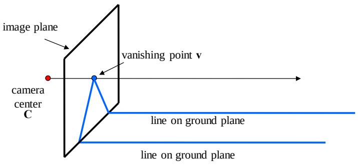
Vanishing points



Vanishing point

- projection of a point at infinity

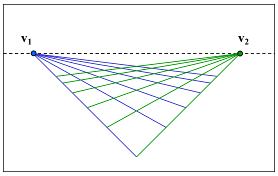
Vanishing points



Properties

- Any two parallel lines have the same vanishing point **v**
- The ray from **C** through **v** is parallel to the lines
- An image may have more than one vanishing point
  - in fact every pixel is a potential vanishing point

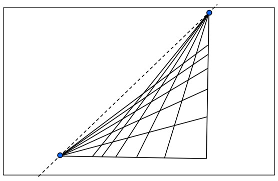
Vanishing lines



Multiple Vanishing Points

- Any set of parallel lines on the plane define a vanishing point
- The union of all of vanishing points from lines on the same plane is the *vanishing line*
  - For the ground plane, this is called the *horizon*

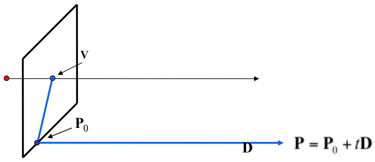
Vanishing lines



Multiple Vanishing Points

- Different planes define different vanishing lines

Computing vanishing points

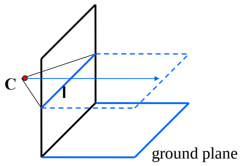


$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} P_X/t + D_X \\ P_Y/t + D_Y \\ P_Z/t + D_Z \\ 1/t \end{bmatrix} \qquad t \rightarrow \infty \qquad \mathbf{P}_\infty \cong \begin{bmatrix} D_X \\ D_Y \\ D_Z \\ 0 \end{bmatrix}$$

Properties $\mathbf{v} = \Pi \mathbf{P}_\infty$ ($\prod$ is camera projection matrix)

- $\mathbf{P}_\infty$ is a point at *infinity*, **v** is its projection
- They depend only on line *direction*
- Parallel lines $\mathbf{P}_0$ + t**D**, $\mathbf{P}_1$ + t**D** intersect at $\mathbf{P}_\infty$

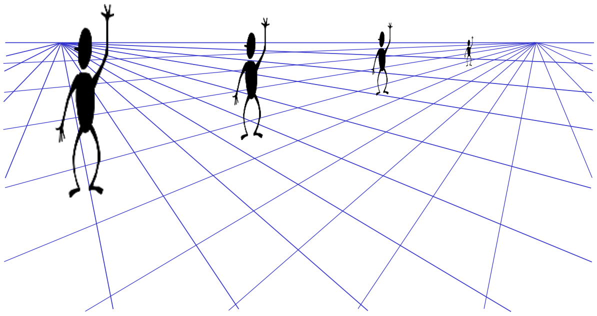
Computing the horizon



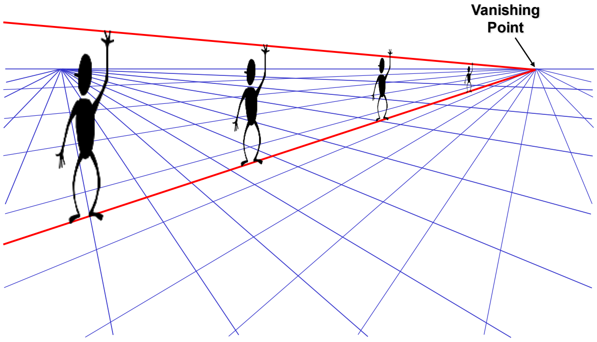
Properties

- **l** is intersection of horizontal plane through **C** with image plane
- Compute **l** from two sets of parallel lines on ground plane
- All points at same height as **C** project to **l**
    - points higher than **C** project above **l**
- Provides way of comparing height of objects in the scene

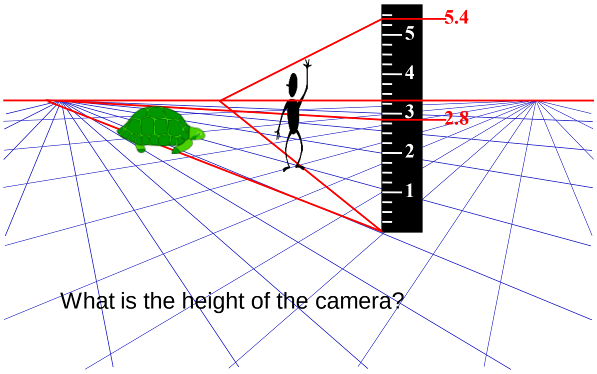
Are these guys the same height?



Comparing heights

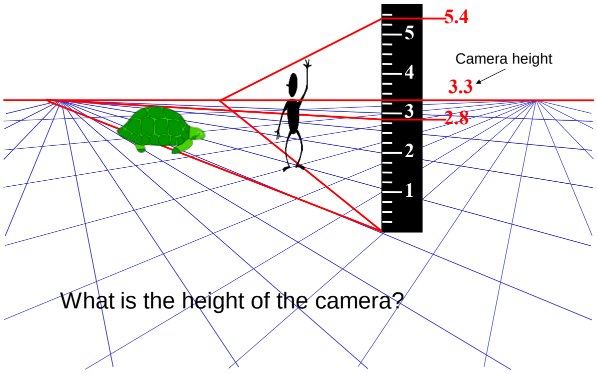


Measuring height



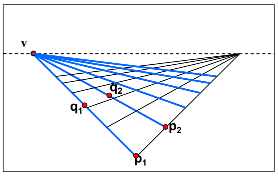
What is the height of the camera?

Measuring height



What is the height of the camera?

Computing vanishing points (from lines)



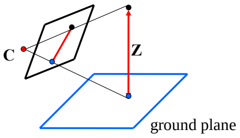
Intersect p_1q_1 with p_2q_2

$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

Least squares version

- Better to use more than two lines and compute the "closest" point of intersection
- See notes by Bob Collins for one good way of doing this:
  - http://www-2.cs.cmu.edu/~ph/869/www/notes/vanishing.txt

Measuring height without a ruler



Compute Z from image measurements

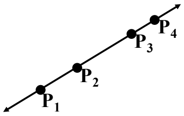
- Need more than vanishing points to do this

The cross ratio

A Projective Invariant

- Something that does not change under projective transformations (including perspective projection)

The cross-ratio of 4 collinear points



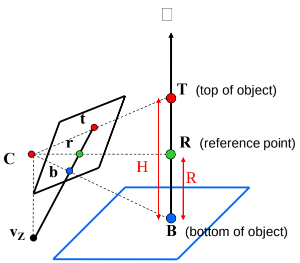
$$\frac{\|\mathbf{P}_3 - \mathbf{P}_1\| \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_3 - \mathbf{P}_2\| \|\mathbf{P}_4 - \mathbf{P}_1\|} \qquad \mathbf{P}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Can permute the point ordering $\frac{\|\mathbf{P}_1 - \mathbf{P}_3\| \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_1 - \mathbf{P}_2\| \|\mathbf{P}_4 - \mathbf{P}_3\|}$

- 4! = 24 different orders (but only 6 distinct values)

This is the fundamental invariant of projective geometry

Measuring height



$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|}=\frac{H}{R}$$

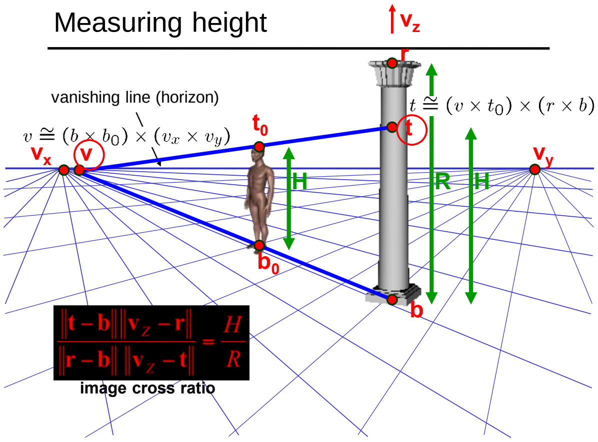
scene cross ratio

$$\frac{\|\mathbf{t}-\mathbf{b}\|\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}=\frac{H}{R}$$

image cross ratio

scene points represented as $\mathbf{P}=\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$ image points as $\mathbf{p}=\begin{bmatrix}x\\y\\1\end{bmatrix}$

Measuring height



$v \cong (b \times b_0) \times (v_x \times v_y)$

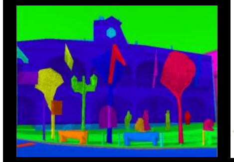
$t \cong (v \times t_0) \times (r \times b)$

$$\frac{\|\mathbf{t}-\mathbf{b}\|\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}=\frac{H}{R}$$

image cross ratio

LabelMe3D: Building a database of 3D scenes from user annotations

Goal: Collect a large labeled 3D dataset in absolute coordinates over many different scene types and object categories



Object labels: tree, road, person, …

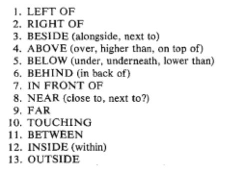
tree (4.2 meters tall) person (1.7 meters tall)

[B.C. Russell and A. Torralba, CVPR 2009]

Benefits of a 3D database

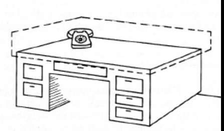
- Can be used as a validation dataset
- Techniques used to generate database can be incorporated into scene understanding system
- Useful as a prior for 3D tasks (e.g. recognition, image/video pop-up)
- Other creative applications (object attribute queries, studying 3D relationships between objects)

Reasoning about spatial relationships between objects

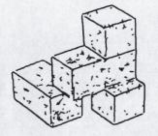


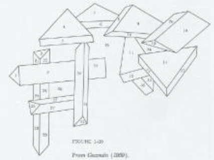
1. LEFT OF
2. RIGHT OF
3. BESIDE (alongside, next to)
4. ABOVE (over, higher than, on top of)
5. BELOW (under, underneath, lower than)
6. BEHIND (in back of)
7. IN FRONT OF
8. NEAR (close to, next to?)
9. FAR
10. TOUCHING
11. BETWEEN
12. INSIDE (within)
13. OUTSIDE

Freeman, 1974



Ballard & Brown, 1982





Guzman, 1969

Our approach

Use object labels provided by humans to discover relationships between objects and recover 3D scene structure

Similar to the line analysis work of the 70s, but with more data

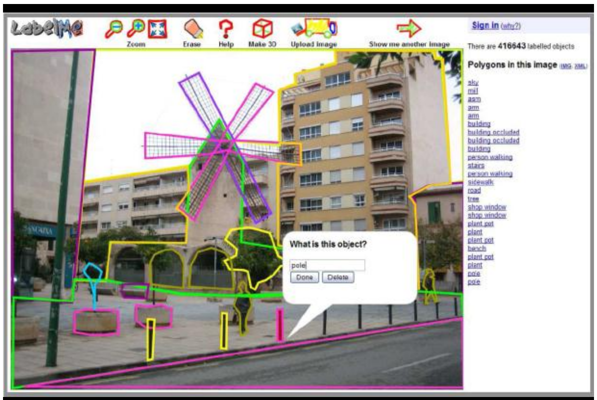
Clowes, 1971 Barrow & Tenenbaum, 1978 Huffman, 1977 Sugihara, 1984

Goals of LabelMe

- Build large collection of images depicting scenes and objects in their natural context

- Collect detailed annotations of many objects in the scene

[B.C. Russell, A. Torralba, K.P. Murphy, W.T. Freeman, IJCV 2008]



There are 416643 labelled objects

Polygons in this image (IMG, XML)

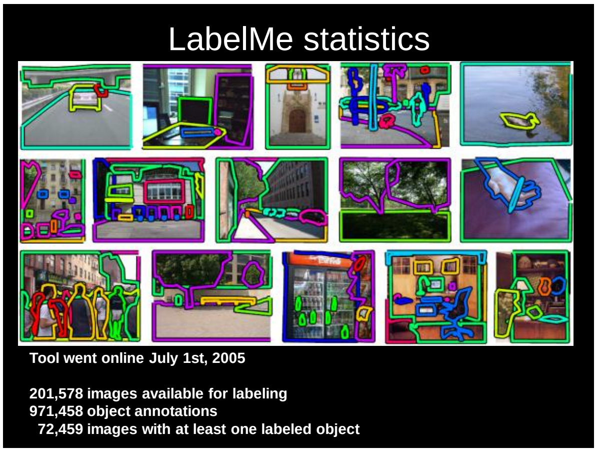
sky
mill
arm
arm
arm
building
building occluded
building occluded
building
person walking
stairs
person walking
sidewalk
road
tree
shop window
shop window
plant pot
plant
plant pot
bench
plant pot
plant
pole
pole

What is this object?

pole

http://labelme.csail.mit.edu

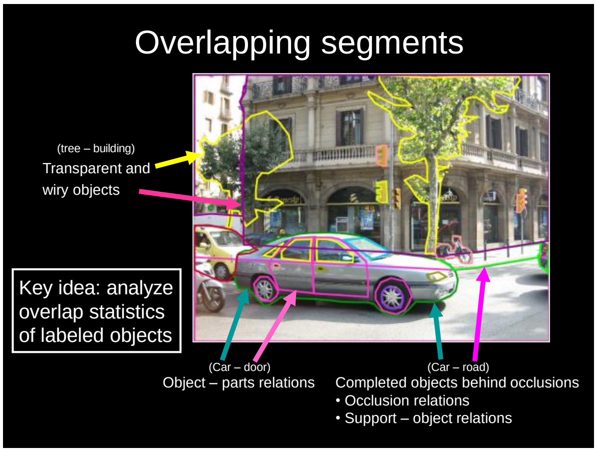
Matlab toolbox and annotation tool source code available



LabelMe statistics

Tool went online July 1st, 2005

201,578 images available for labeling
971,458 object annotations
72,459 images with at least one labeled object



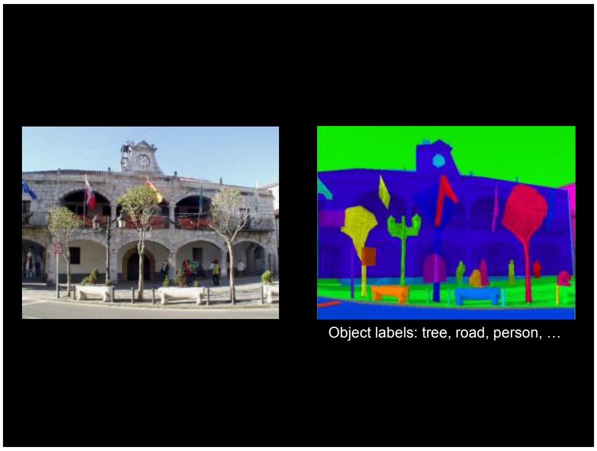
Overlapping segments

(tree – building)
Transparent and
wiry objects

Key idea: analyze overlap statistics of labeled objects

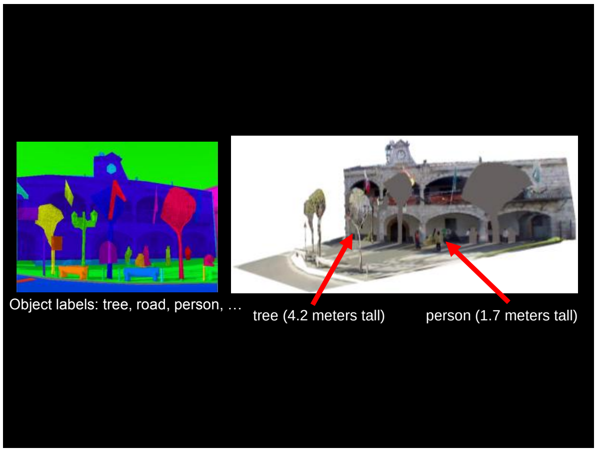
(Car – door)
Object – parts relations

(Car – road)
Completed objects behind occlusions
- Occlusion relations
- Support – object relations



Object labels: tree, road, person, …

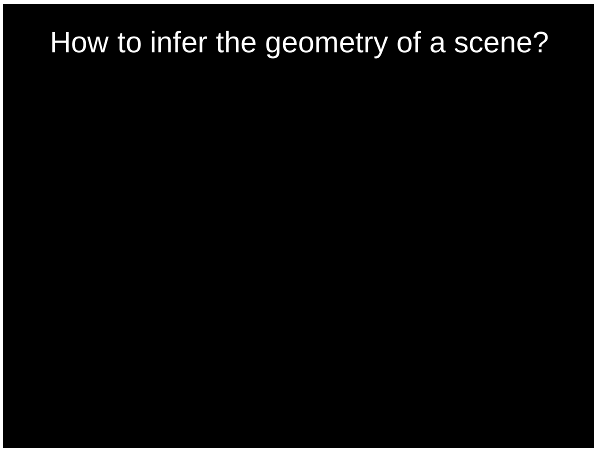
Object labels: tree, road, person, …



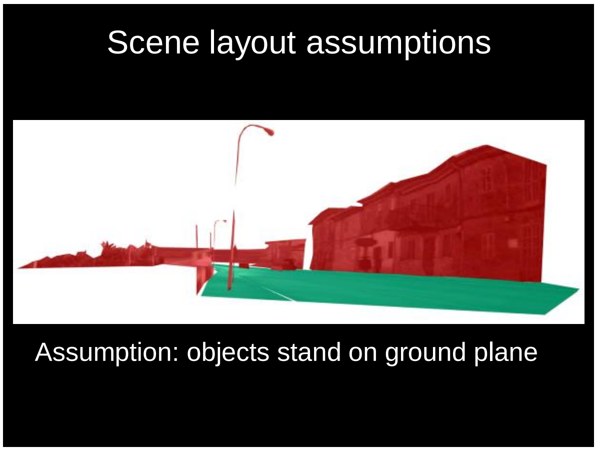
Object labels: tree, road, person, …

tree (4.2 meters tall)

person (1.7 meters tall)

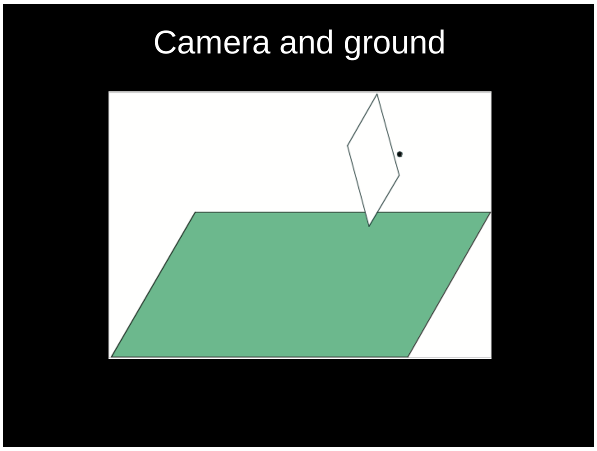


How to infer the geometry of a scene?

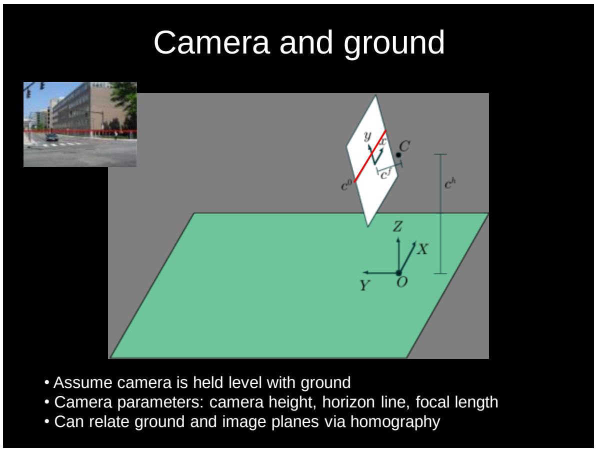


Scene layout assumptions

Assumption: objects stand on ground plane

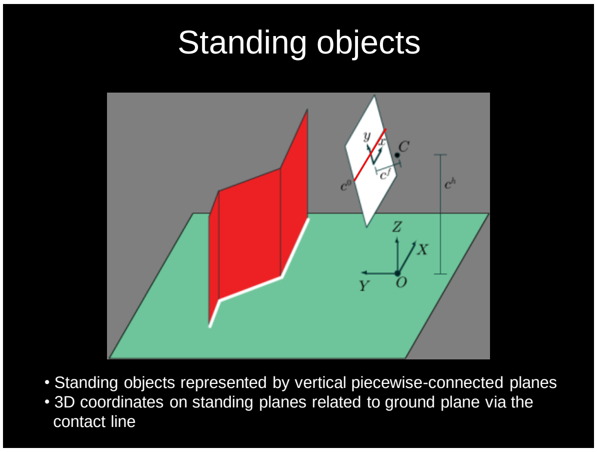


Camera and ground



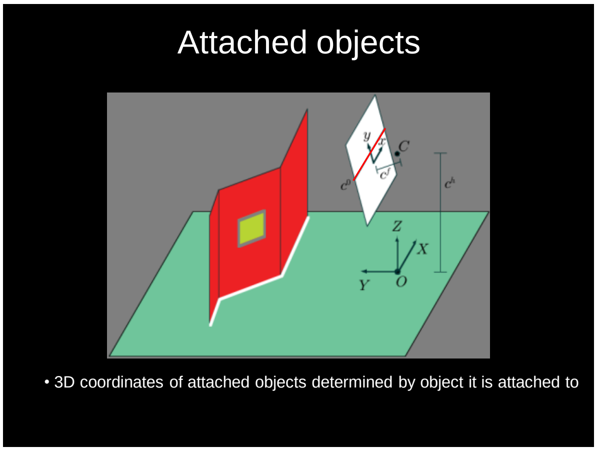
Camera and ground

- Assume camera is held level with ground
- Camera parameters: camera height, horizon line, focal length
- Can relate ground and image planes via homography



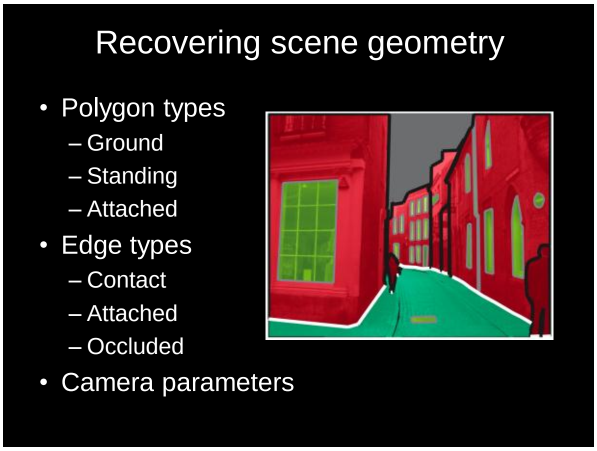
Standing objects

- Standing objects represented by vertical piecewise-connected planes
- 3D coordinates on standing planes related to ground plane via the contact line



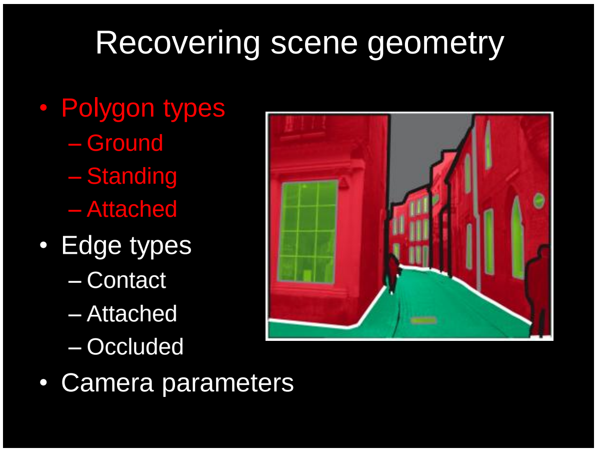
Attached objects

- 3D coordinates of attached objects determined by object it is attached to



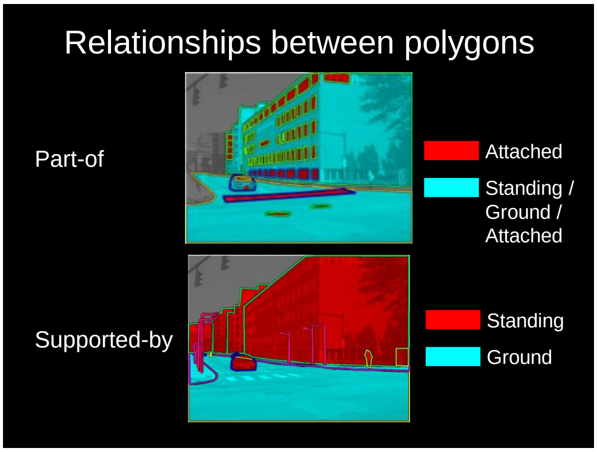
Recovering scene geometry

- Polygon types
  - Ground
  - Standing
  - Attached
- Edge types
  - Contact
  - Attached
  - Occluded
- Camera parameters



Recovering scene geometry

- Polygon types
  - Ground
  - Standing
  - Attached
- Edge types
  - Contact
  - Attached
  - Occluded
- Camera parameters



Relationships between polygons

Part-of

🟥 Attached

🟦 Standing / Ground / Attached

Supported-by

🟥 Standing

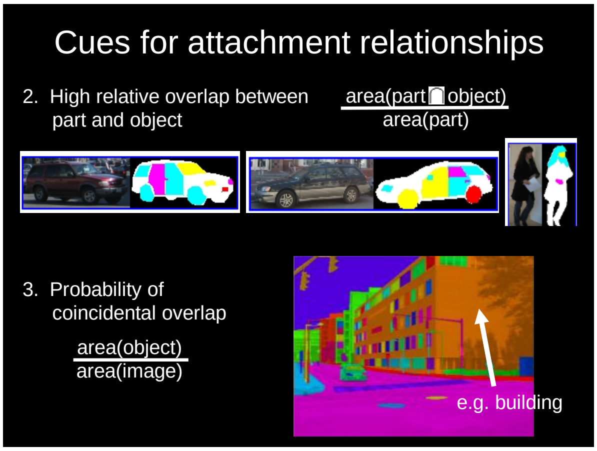
🟦 Ground

Cues for attachment relationships

1. Consistency of relationship across database

building, windows

building, person



Cues for attachment relationships

2. High relative overlap between part and object

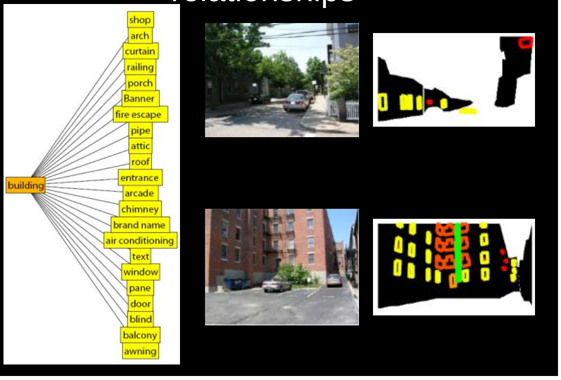
$$\frac{\text{area}(\text{part} \cap \text{object})}{\text{area}(\text{part})}$$

3. Probability of coincidental overlap

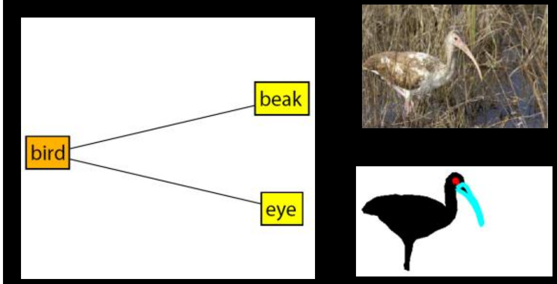
$$\frac{\text{area}(\text{object})}{\text{area}(\text{image})}$$

e.g. building

Learned/inferred attachment relationships

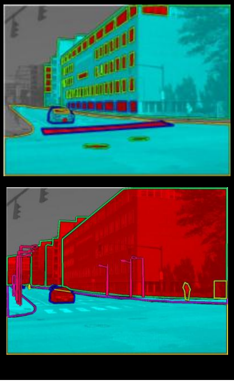


Learned/inferred attachment relationships



Relationships between polygons

Part-of



🟥 Attached

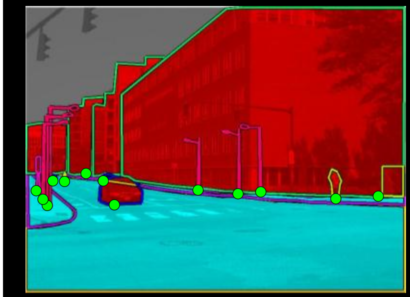
🟦 Standing / Ground / Attached

Supported-by

🟥 Standing

🟦 Ground

Recover support relations



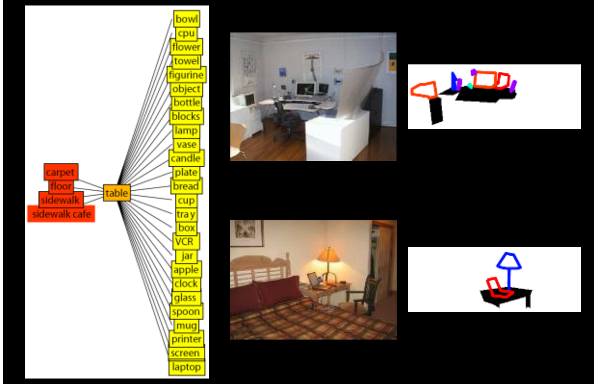
🟥 Object

🟦 Support

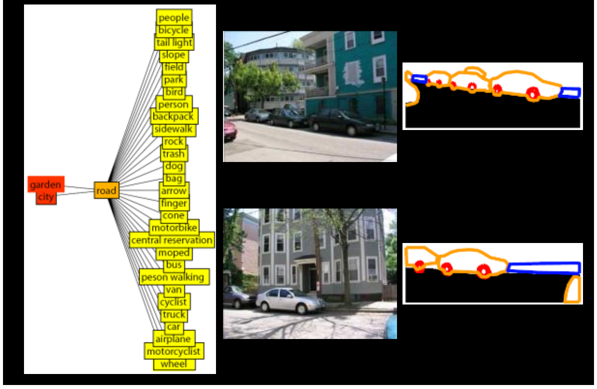
$$P_{support} = \frac{N_s}{N + \alpha}$$

Over entire dataset, count number of images where bottom of object is inside support object

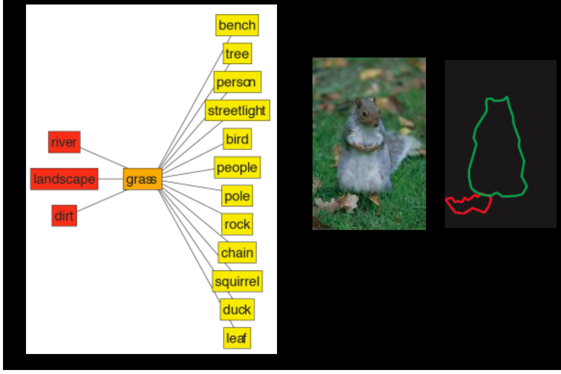
Learned/inferred support relations



Learned/inferred support relations

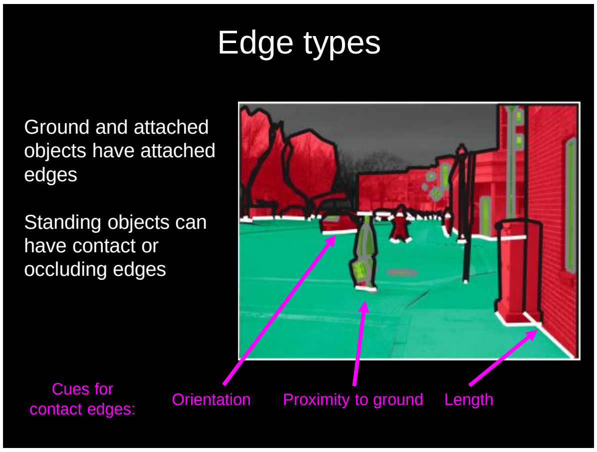


Learned/inferred support relations



Recovering scene geometry

- Polygon types
  - Ground
  - Standing
  - Attached
- Edge types
  - Contact
  - Attached
  - Occluded
- Camera parameters

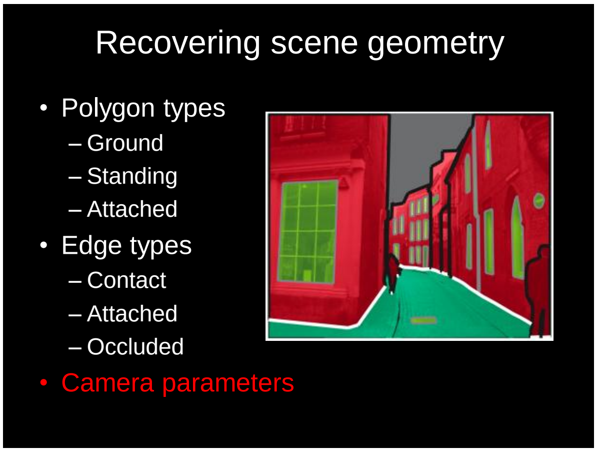


Edge types

Ground and attached objects have attached edges

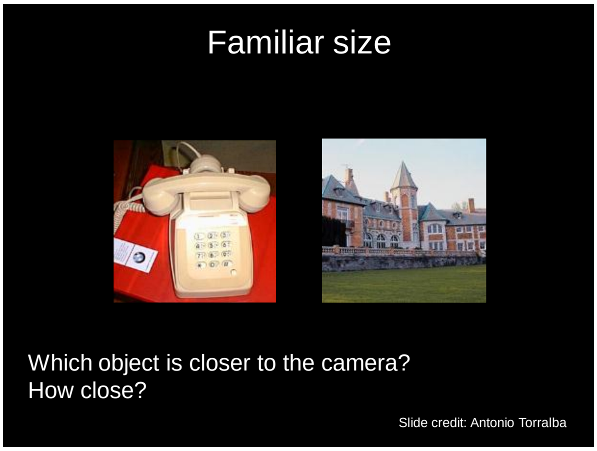
Standing objects can have contact or occluding edges

Cues for contact edges: Orientation Proximity to ground Length



Recovering scene geometry

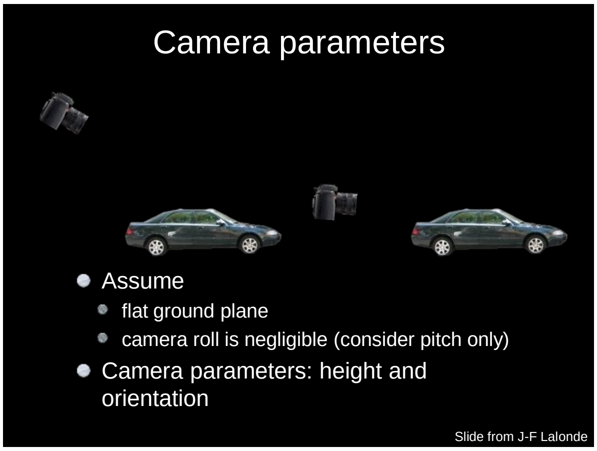
- Polygon types
  - Ground
  - Standing
  - Attached
- Edge types
  - Contact
  - Attached
  - Occluded
- Camera parameters



Familiar size

Which object is closer to the camera?
How close?

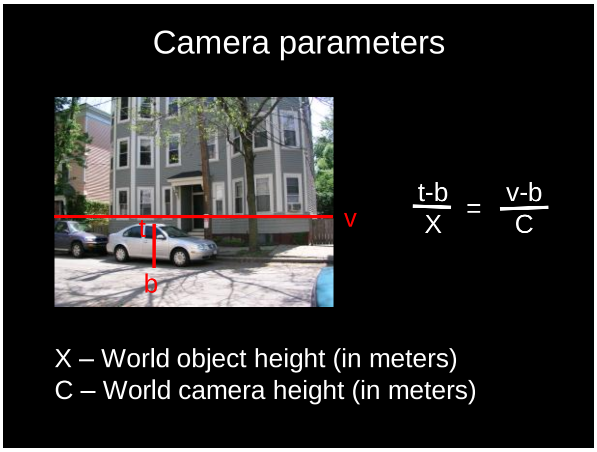
Slide credit: Antonio Torralba



Camera parameters

- Assume
  - flat ground plane
  - camera roll is negligible (consider pitch only)
- Camera parameters: height and orientation

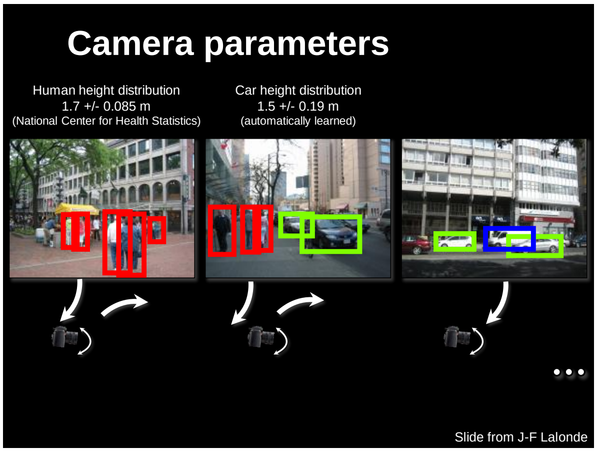
Slide from J-F Lalonde



Camera parameters

$$\frac{t-b}{X} = \frac{v-b}{C}$$

X – World object height (in meters)
C – World camera height (in meters)

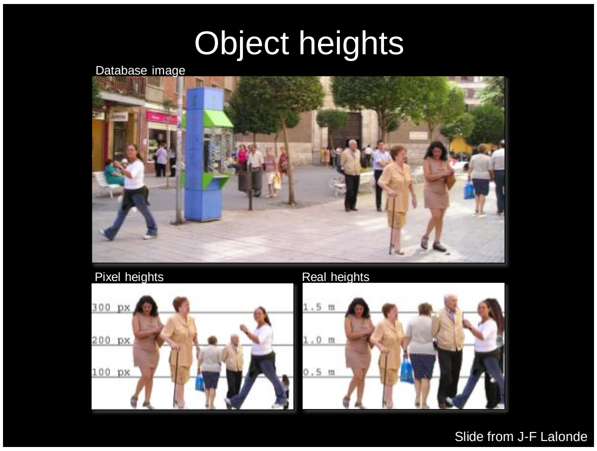


Camera parameters

Human height distribution
1.7 +/- 0.085 m
(National Center for Health Statistics)

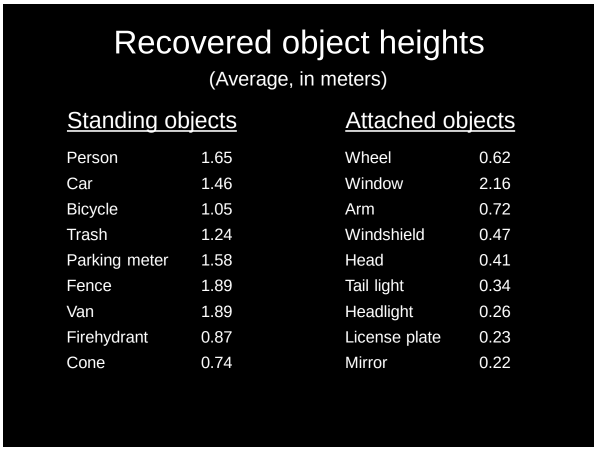
Car height distribution
1.5 +/- 0.19 m
(automatically learned)

Slide from J-F Lalonde



Object heights

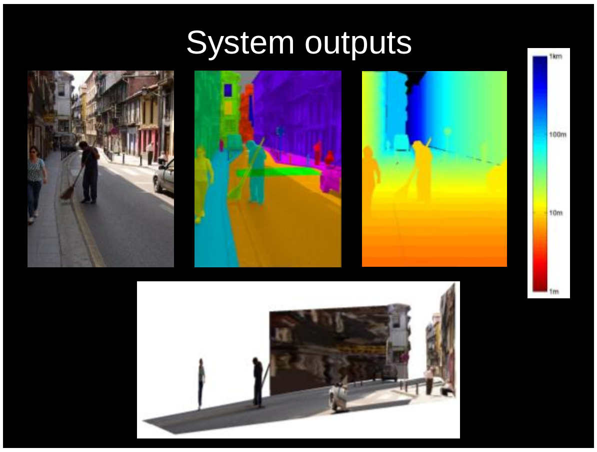
Slide from J-F Lalonde



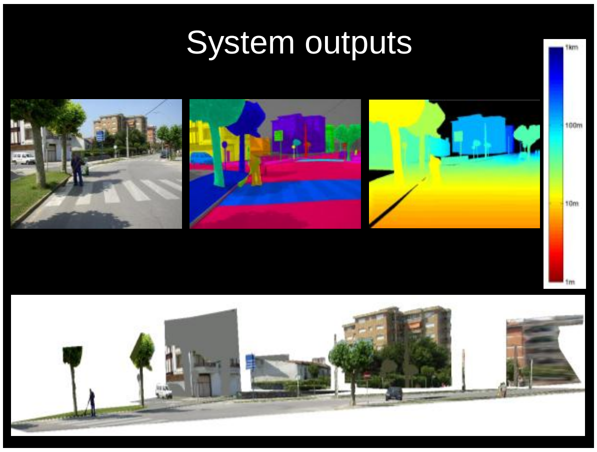
Recovered object heights

(Average, in meters)

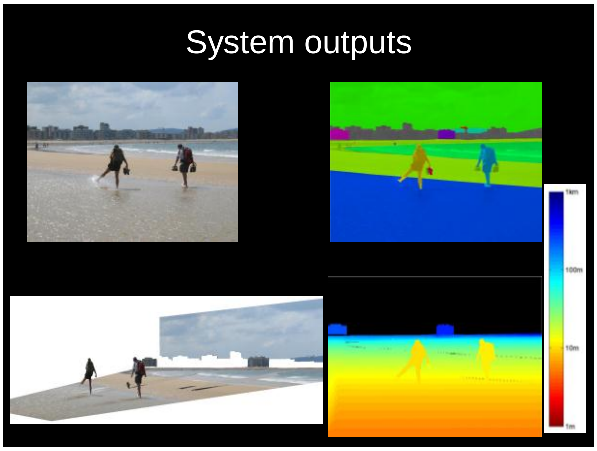
Standing objects		Attached objects	
Person	1.65	Wheel	0.62
Car	1.46	Window	2.16
Bicycle	1.05	Arm	0.72
Trash	1.24	Windshield	0.47
Parking meter	1.58	Head	0.41
Fence	1.89	Tail light	0.34
Van	1.89	Headlight	0.26
Firehydrant	0.87	License plate	0.23
Cone	0.74	Mirror	0.22



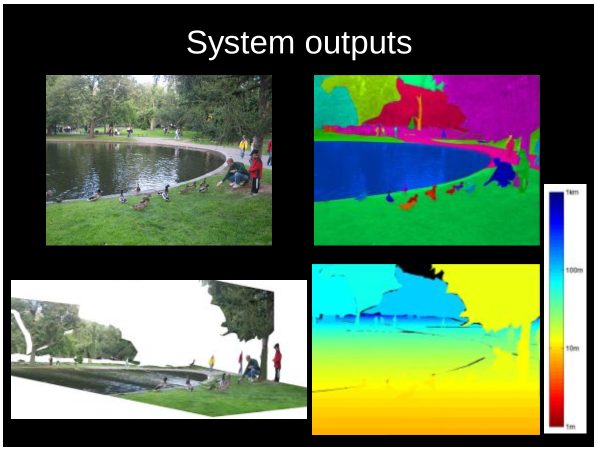
System outputs



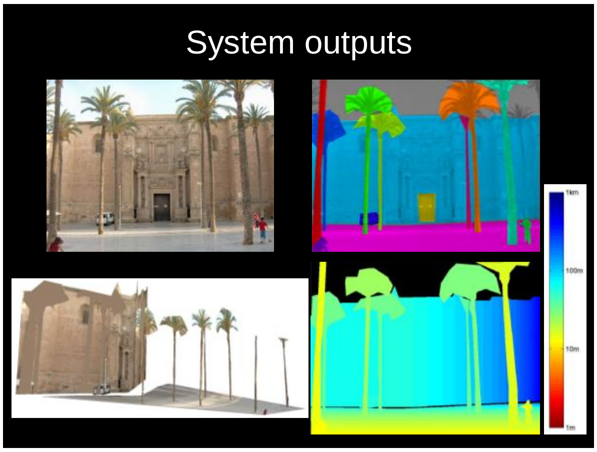
System outputs



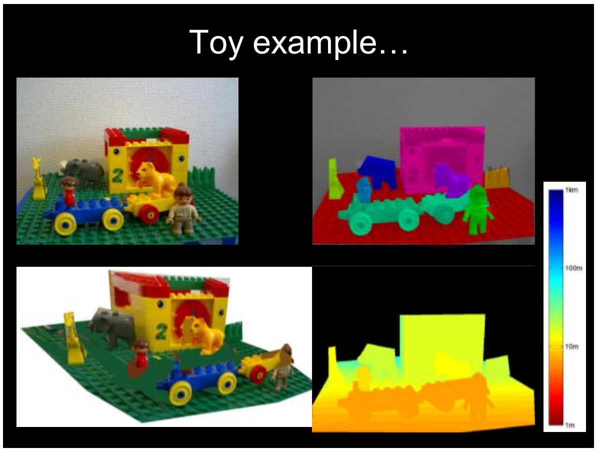
System outputs



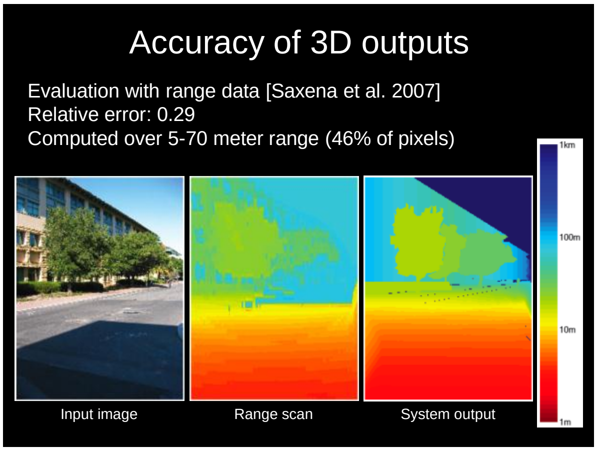
System outputs



System outputs



Toy example…



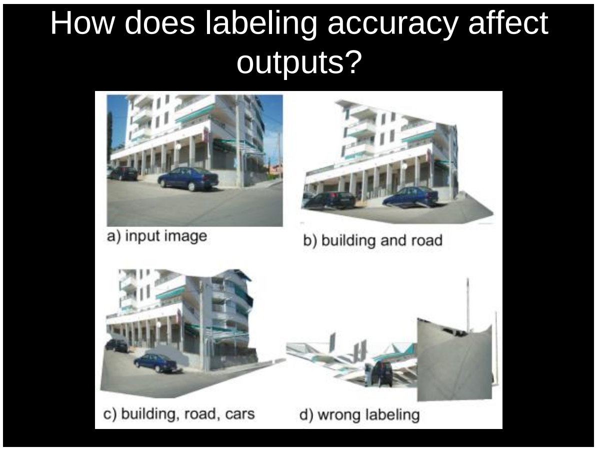
Accuracy of 3D outputs

Evaluation with range data [Saxena et al. 2007]
Relative error: 0.29
Computed over 5-70 meter range (46% of pixels)

Input image

Range scan

System output



How does labeling accuracy affect outputs?

a) input image

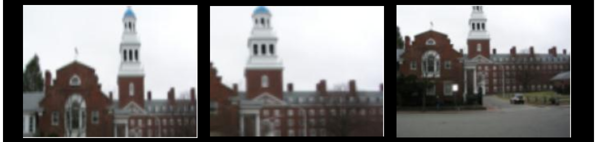
b) building and road

c) building, road, cars

d) wrong labeling

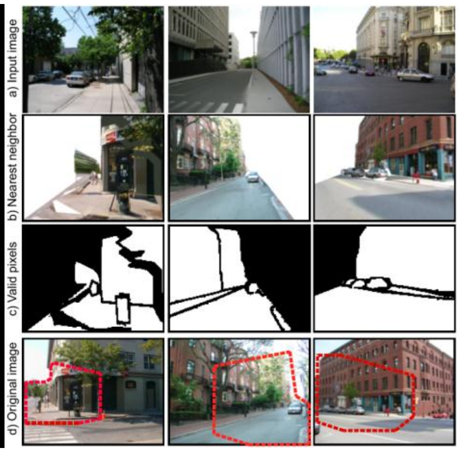
Application: Extending database with virtual views

Randomly cropping

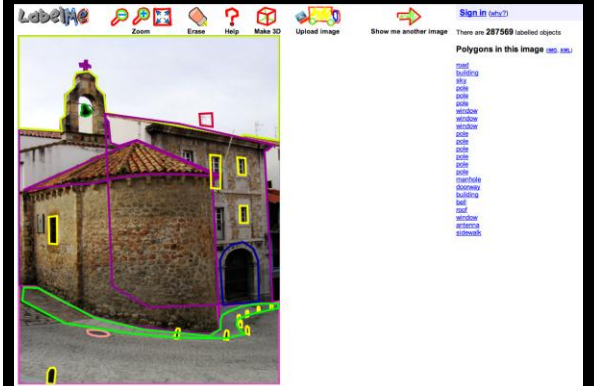


Virtual viewpoints

20 new views generated per image (6000 images)



Labeling 3D

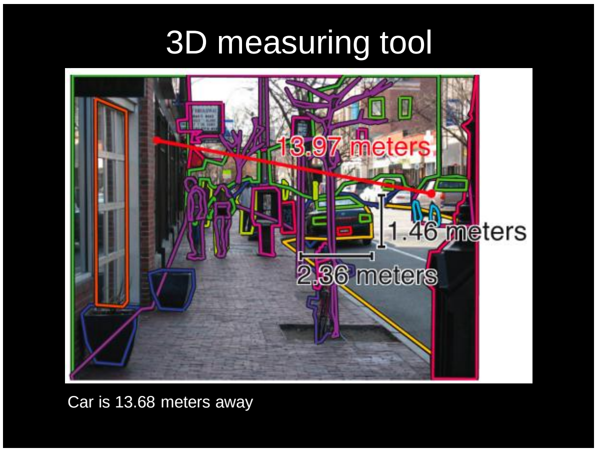


There are 287569 labelled objects

Polygons in this image (IMG, XML)

road
building
sky
pole
pole
pole
window
window
window
pole
pole
pole
pole
pole
pole
manhole
doorway
building
bell
roof
window
antenna
sidewalk

Cut and glue!



3D measuring tool

Car is 13.68 meters away