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?

We don’t live in a 2D Mondrian world

Nearby pixels are close in 2D

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

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

Context for object detection

Information on how to navigate in an environment
Edge interpretation

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

Interposition / occlusion

Texture Gradient

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

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.

Shadows

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.

Linear Perspective

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

Gibson’s “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...
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.

Moon illusion

Absolute (monocular) depth cues

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

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

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?

Perspective vs. familiar size

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

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]
Scene vs. objects

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.

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.

Today’s class: reasoning about perspective cues via projective geometry
Readings

- Hartley and Zisserman textbook

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

![Image](image.png)

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

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: \(ax + by + cz = 0\)

  ![Vector](vector.png)

  - in vector notation: \(\begin{bmatrix} a \\ b \\ c \end{bmatrix}\)

- 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 \( \infty \) to every point (ray) \( p \) on the line: \( l \cdot p = 0 \)

What is the line \( l \) spanned by rays \( p_1 \) and \( p_2 \)?
- \( l \) is \( \infty \) to \( p_1 \) and \( p_2 \) \( \iff l = p_1 \times p_2 \)
- \( l \) is the plane normal

What is the intersection of two lines \( l_1 \) and \( l_2 \)?
- \( p \) is \( \infty \) to \( l_1 \) and \( l_2 \) \( \iff p = l_1 \times l_2 \)

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

**Ideal line**
- \( l \in (a, b, 0) \) – 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, collineation, planar projective map
- Modeled as a 2D warp using homogeneous coordinates

\[
\begin{bmatrix}
wx' \\
yw' \\
w' \\
\end{bmatrix} = \begin{bmatrix} x \\
y \\
1 \end{bmatrix}
\]

To apply a homography \( \mathbf{H} \)
- Compute \( p' = \mathbf{H}p \) (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: $wp' = Hp$
  - 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

A defines a least squares problem: $\text{minimize } ||Ah - 0||^2$

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

Vanishing points

- projection of a point at infinity

\[
\begin{bmatrix}
    x' \\
    y' \\
    1
\end{bmatrix} =
\begin{bmatrix}
    h_{00} & h_{01} & h_{02} & x_0 \\
    h_{10} & h_{11} & h_{12} & y_0 \\
    h_{20} & h_{21} & h_{22} & 1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    1
\end{bmatrix}
\]

\[
x' = (h_{00} + h_{03} + h_{06}) x + (h_{01} + h_{04} + h_{06}) y + (h_{02} + h_{05} + h_{06})
\]

\[
y' = (h_{10} + h_{13} + h_{16}) x + (h_{11} + h_{14} + h_{16}) y + (h_{12} + h_{15} + h_{16})
\]
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

Computing vanishing points

\[
P = P_0 + tD
\]

Properties $v = \Pi P_\infty$
- $P_\infty$ is a point at infinity, $v$ is its projection
- They depend only on line direction
- Parallel lines $P_1 + tD$, $P_2 + tD$ intersect at $P$
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?

Camera height

Measuring height without a ruler

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://www2.cs.cmu.edu/~ph/869/www/notes/vanishing.txt

The cross ratio

A Projective Invariant

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

The cross-ratio of 4 collinear points

$$\begin{bmatrix} P_1 - P_3 & P_4 - P_3 \\ P_1 - P_2 & P_4 - P_2 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Can permute the point ordering

$$\begin{bmatrix} P_1 - P_4 & P_2 - P_4 \\ P_1 - P_3 & P_2 - P_3 \end{bmatrix}$$

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

This is the fundamental invariant of projective geometry
Measuring height

\[ \frac{\|T - B\|}{\|R - B\|} = \frac{H}{R} \]

scene cross ratio

\[ \frac{\|t - b\|}{\|v_z - t\|} = \frac{H}{R} \]

image cross ratio

scene points represented as \( p = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \)

image points as \( p = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \)

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, …

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. BEHIND (behind, next to)
4. ABOVE (over, higher than, on top of)
5. BELOW (under, below, lower than)
6. BEHIND (in back of)
7. BEHIND (behind, next to)
8. NEAR (close to, next to)
9. FAR
10. TOUCHING
11. BETWEEN
12. INSIDE (interior)
13. OUTSIDE

Guzman, 1969
Ballard & Brown, 1982

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


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]

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

Key idea: analyze overlap statistics of labeled objects

- Object - parts relations
  - Car - door
  - Car - road
- Completed objects behind occlusions
  - Occlusion relations
  - Support - object relations

Object labels: tree, road, person, ...

Object labels: tree, road, person, ...
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

2. High relative overlap between part and object

area(part) / area(object)

3. Probability of coincidental overlap

area(object) / area(image)

e.g. building
Learned/inferred attachment relationships

Relationships between polygons
- Part-of
- Supported-by

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

\[ P_{support} = \frac{N_d}{N + \alpha} \]
Learned/inferred support relations

- Polygon types
  - Ground
  - Standing
  - Attached

- Edge types
  - Contact
  - Attached
  - Occluded

- Camera parameters

Recovering scene geometry
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?

Camera parameters

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

\[
\frac{t-b}{X} = \frac{v-b}{C}
\]

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

Object heights

Database image

Recovered object heights

(Average, in meters)

<table>
<thead>
<tr>
<th>Standing objects</th>
<th>Attached objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>Wheel</td>
</tr>
<tr>
<td>1.65</td>
<td>0.62</td>
</tr>
<tr>
<td>Car</td>
<td>Window</td>
</tr>
<tr>
<td>1.46</td>
<td>2.16</td>
</tr>
<tr>
<td>Bicycle</td>
<td>Arm</td>
</tr>
<tr>
<td>1.05</td>
<td>0.72</td>
</tr>
<tr>
<td>Trash</td>
<td>Windshield</td>
</tr>
<tr>
<td>1.24</td>
<td>0.47</td>
</tr>
<tr>
<td>Parking meter</td>
<td>Head</td>
</tr>
<tr>
<td>1.58</td>
<td>0.41</td>
</tr>
<tr>
<td>Fence</td>
<td>Tail light</td>
</tr>
<tr>
<td>1.89</td>
<td>0.34</td>
</tr>
<tr>
<td>Van</td>
<td>Headlight</td>
</tr>
<tr>
<td>1.89</td>
<td>0.26</td>
</tr>
<tr>
<td>Firehydrant</td>
<td>License plate</td>
</tr>
<tr>
<td>0.87</td>
<td>0.23</td>
</tr>
<tr>
<td>Cone</td>
<td>Mirror</td>
</tr>
<tr>
<td>0.74</td>
<td>0.22</td>
</tr>
</tbody>
</table>
System outputs

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)

How does labeling accuracy affect outputs?

Input image  Range scan  System output

Toy example…

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

Cut and glue!
3D measuring tool

Car is 13.68 meters away