Scene Modeling for a Single View

René MAGRITTE
Portrait d'Edward James

CS194: Intro to Comp. Vision and Comp. Photo
Alexei Efros, UC Berkeley, Fall 2020
Breaking out of 2D

...now we are ready to break out of 2D

And enter the real world!
on to 3D...

Enough of images!

We want more of the plenoptic function

We want real 3D scene walk-throughs:
  Camera rotation
  Camera translation

Can we do it from a single photograph?
Camera rotations with homographies

Original image

St. Petersburg
photo by A. Tikhonov

Virtual camera rotations
Camera translation

Does it work?

PP1

PP2

synthetic PP
Yes, with planar scene (or far away)

PP3 is a projection plane of both centers of projection, so we are OK!
So, what can we do here?

Model the scene as a set of planes!

Now, just need to find the orientations of these planes.
Some preliminaries: projective geometry

Ames Room
Silly Euclid!

Parallel lines???
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: \((x, y, 1) \cong (sx, sy, s)\)
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: \(ax + by + cz = 0\)

  \[
  \begin{bmatrix}
  x \\
  y \\
  z
  \end{bmatrix}
  \begin{bmatrix}
  a & b & c
  \end{bmatrix}
  \begin{bmatrix}
  l \\
  p
  \end{bmatrix}
  \]

- A line is also represented as a homogeneous 3-vector \(l\)
Ideal points and lines

Ideal point ("point at infinity")
- $p \approx (x, y, 0)$ – parallel to image plane
- It has infinite image coordinates

Ideal line
- $l \approx (0, 0, 1)$ – parallel to image plane
Vanishing points

Vanishing point
- projection of a point at infinity
Vanishing points (2D)
Computing vanishing points

Properties

- $P \infty$ is a point at *infinity*, $v$ is its projection
- They depend only on line *direction*
- Parallel lines $P_0 + tD, P_1 + tD$ intersect at $P \infty$

\[
P_t = \begin{bmatrix}
P_x + tD_x \\
P_y + tD_y \\
P_z + tD_z \\
1
\end{bmatrix} \approx \begin{bmatrix}
P_x / t + D_x \\
P_y / t + D_y \\
P_z / t + D_z \\
1 / t
\end{bmatrix} \quad t \to \infty \quad P_\infty \approx \begin{bmatrix}
D_x \\
D_y \\
D_z \\
0
\end{bmatrix}
\]

$v = \Pi P_\infty$
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
Vanishing lines

Multiple Vanishing Points

- Any set of parallel lines on the plane define a vanishing point
- The union of all of these vanishing points is the horizon line – also called vanishing line
- Note that different planes define different vanishing lines
Vanishing lines

Multiple Vanishing Points

- Any set of parallel lines on the plane define a vanishing point
- The union of all of these vanishing points is the horizon line – also called vanishing line
- Note that different planes define different vanishing lines
Computing vanishing lines

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
Fun with vanishing points
Create a 3D “theatre stage” of five billboards

Specify foreground objects through bounding polygons

Use camera transformations to navigate through the scene
The idea

Many scenes (especially paintings), can be represented as an axis-aligned box volume (i.e. a stage)

Key assumptions:
- All walls of volume are orthogonal
- Camera view plane is parallel to back of volume
- Camera up is normal to volume bottom

How many vanishing points does the box have?
- Three, but two at infinity
- Single-point perspective

Can use the vanishing point to fit the box to the particular Scene!
Fitting the box volume

User controls the inner box and the vanishing point placement (# of DOF???)

Q: What’s the significance of the vanishing point location?
A: It’s at eye level: ray from COP to VP is perpendicular to image plane.
Example of user input: vanishing point and back face of view volume are defined.
Example of user input: vanishing point and back face of view volume are defined
Example of user input: vanishing point and back face of view volume are defined
Example of user input: vanishing point and back face of view volume are defined.
Comparison of how image is subdivided based on two different camera positions. You should see how moving the box corresponds to moving the eyepoint in the 3D world.

High Camera

Low Camera
Another example of user input: vanishing point and back face of view volume are defined.
Another example of user input: vanishing point and back face of view volume are defined
Another example of user input: vanishing point and back face of view volume are defined
Another example of user input: vanishing point and back face of view volume are defined
Comparison of two camera placements – left and right. Corresponding subdivisions match view you would see if you looked down a hallway.
2D to 3D conversion

First, we can get ratios

![Diagram showing 2D to 3D conversion with vanishing point, left, right, top, bottom, and back plane.]
2D to 3D conversion

• Size of user-defined back plane must equal size of camera plane (orthogonal sides)
• Use top versus side ratio to determine relative height and width dimensions of box
• Left/right and top/bot ratios determine part of 3D camera placement
Depth of the box

Can compute by similar triangles (CVA vs. CV’A’)
Need to know focal length f (or FOV)

Note: can compute position on any object on the ground
• Simple unprojection
• What about things off the ground?
Now, we know the 3D geometry of the box.
We can texture-map the box walls with texture from the image.
Foreground Objects

Use separate billboard for each

For this to work, three separate images used:

- Original image.
- Mask to isolate desired foreground images.
- Background with objects removed
Foreground Objects

Add vertical rectangles for each foreground object.

Can compute 3D coordinates $P_0$, $P_1$ since they are on known plane.

$P_2$, $P_3$ can be computed as before (similar triangles).

(a) Specifying of a foreground object

(b) Estimating the vertices of the foreground object model

(c) Three foreground object models
Foreground DEMO
How can we model more complex scene?

1. Find world coordinates (X,Y,Z) for a few points
2. Connect the points with planes to model geometry
   • Texture map the planes
Finding world coordinates (X,Y,Z)

1. Define the ground plane (Z=0)
2. Compute points (X,Y,0) on that plane
3. Compute the *heights* Z of all other points
Measurements on planes

Approach: unwarp, then measure
What kind of warp is this?
Unwarp ground plane

Our old friend – the homography

Need 4 reference points with world coordinates

\[ p = (x,y) \]
\[ p' = (X,Y,0) \]
Finding world coordinates (X,Y,Z)

1. Define the ground plane (Z=0)
2. Compute points (X,Y,0) on that plane
3. Compute the heights Z of all other points
Comparing heights
Perspective cues
Perspective cues
Vanishing point

Vertical vanishing point (at infinity)

Criminisi '99
Comparing heights

Vanishing Point
Measuring height

Camera height

5.4
3.3
2.8
Measuring height

\[ v \approx (b \times b_0) \times (v_x \times v_y) \]

vanishing line (horizon)

\[ t \approx (v \times t_0) \times (r \times b) \]
What if the point on the ground plane $b_0$ is not known?

- Here the guy is standing on the box
- Use one side of the box to help find $b_0$ as shown above
The cross ratio

A Projective Invariant

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

The cross-ratio of 4 collinear points

\[
\frac{\|P_3 - P_1\|}{\|P_3 - P_2\|} \div \frac{\|P_4 - P_2\|}{\|P_4 - P_1\|}
\]

\[
P_i = \begin{bmatrix}
X_i \\
Y_i \\
Z_i \\
1
\end{bmatrix}
\]

Can permute the point ordering

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

This is the fundamental invariant of projective geometry
Measuring height

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

scene cross ratio

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

image cross ratio

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

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

vanishing line (horizon)

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

image cross ratio
Measuring heights of people

Here we go!

185.3 cm
Assessing geometric accuracy

Are the heights of the 2 groups of people consistent with each other?

*Flagellation*,
Piero della Francesca

<table>
<thead>
<tr>
<th>Estimated relative heights</th>
</tr>
</thead>
<tbody>
<tr>
<td>+3.2%</td>
</tr>
<tr>
<td>-1.8%</td>
</tr>
<tr>
<td>-0.2%</td>
</tr>
<tr>
<td>-1.2%</td>
</tr>
<tr>
<td>+1.3%</td>
</tr>
</tbody>
</table>
Assessing geometric accuracy

*The Marriage of the Virgin*, Raphael

Estimated relative heights:
- Real height: -9.0%
- Estimated height: -4.5%
- Real height: +0.3%
- Estimated height: +2.8%
Complete 3D reconstruction

- Planar measurements
- Height measurements
- Automatic vanishing point/line computation
- Interactive segmentation
- Occlusion filling
- Object placement in 3D model
A virtual museum @ Microsoft

The Virtual Museum

A. Criminisi @ Microsoft, 2002

A. Criminisi    http://research.microsoft.com/~antcrim/