Geometric camera models
Course announcements

• Homework 2 is available online.
  - Due on **February 27**\textsuperscript{th} at 23:59.
  - How many of you have read/started/finished HW2?

• There was some confusion about spring break.
  - Course website has been adjusted.
  - There is *no* homework due on spring break.

• Yannis has extra office hours this week:
  - Wednesday 3-4 pm (right after class).
  - Thursday 3-4 pm.
  - Friday 2-3 pm (in addition to the usual 3-5 pm).
Overview of today’s lecture

• Leftover from lecture 8: RANSAC.

• Some motivational imaging experiments.

• Pinhole camera.

• Accidental pinholes.

• Camera matrix.

• Perspective.

• Other camera models.

• Pose estimation.
Most of these slides were adapted from:

• Kris Kitani (15-463, Fall 2016).

Some slides inspired from:

• Fredo Durand (MIT).
Some motivational imaging experiments
Let’s say we have a sensor...

digital sensor
(CCD or CMOS)
... and an object we like to photograph

What would an image taken like this look like?
Bare-sensor imaging

real-world object

digital sensor (CCD or CMOS)
Bare-sensor imaging

real-world object

digital sensor
(CCD or CMOS)
Bare-sensor imaging

real-world object

digital sensor (CCD or CMOS)
Bare-sensor imaging

real-world object

digital sensor (CCD or CMOS)

All scene points contribute to all sensor pixels

What does the image on the sensor look like?
Bare-sensor imaging

All scene points contribute to all sensor pixels
Let’s add something to this scene

What would an image taken like this look like?
Pinhole imaging

real-world object

most rays are blocked

one makes it through

digital sensor (CCD or CMOS)
Pinhole imaging

most rays are blocked

one makes it through

digital sensor (CCD or CMOS)

real-world object
Pinhole imaging

Each scene point contributes to only one sensor pixel

What does the image on the sensor look like?
Pinhole imaging

real-world object

copy of real-world object (inverted and scaled)
Pinhole camera
Pinhole camera a.k.a. camera obscura
Pinhole camera a.k.a. camera obscura

First mention ...

Chinese philosopher Mozi (470 to 390 BC)

First camera ...

Greek philosopher Aristotle (384 to 322 BC)
Pinhole camera terms

- Real-world object
- Barrier (diaphragm)
- Pinhole (aperture)
- Digital sensor (CCD or CMOS)
Pinhole camera terms

real-world object

barrier (diaphragm)

pinhole (aperture)

camera center (center of projection)

digital sensor (CCD or CMOS)

image plane
Focal length

real-world object

focal length f
What happens as we change the focal length?
What happens as we change the focal length?
Focal length

What happens as we change the focal length?

object projection is half the size

real-world object

focal length 0.5 f
Ideal pinhole has infinitesimally small size
• In practice that is impossible.
Pinhole size

What happens as we change the pinhole diameter?
What happens as we change the pinhole diameter?
What happens as we change the pinhole diameter?
What happens as we change the pinhole diameter?

Object projection becomes blurrier

Real-world object
What about light efficiency?

- What is the effect of doubling the pinhole diameter?
- What is the effect of doubling the focal length?
What about light efficiency?

- 2x pinhole diameter $\rightarrow$ 4x light
- 2x focal length $\rightarrow$ $\frac{1}{4}$x light
The lens camera

How does this mapping work exactly?

Lenses map “bundles” of rays from points on the scene to the sensor.
The pinhole camera
The lens camera
Central rays propagate in the same way for both models!
Describing both lens and pinhole cameras

We can derive properties and descriptions that hold for both camera models if:

• We use only central rays.
• We assume the lens camera is in focus.
Important difference: focal length

In a pinhole camera, focal length is distance between aperture and sensor
Important difference: focal length

In a lens camera, focal length is distance where parallel rays intersect.

object distance $D$

focal length $f$

focus distance $D'$
Describing both lens and pinhole cameras

We can derive properties and descriptions that hold for both camera models if:

• We use only central rays.
• We assume the lens camera is in focus.
• We assume that the focus distance of the lens camera is equal to the focal length of the pinhole camera.

Remember: *focal length* $f$ refers to different things for lens and pinhole cameras.
• In this lecture, we use it to refer to the aperture-sensor distance, as in the pinhole camera case.
Accidental pinholes
What does this image say about the world outside?
Accidental pinhole camera

Antonio Torralba, William T. Freeman
Computer Science and Artificial Intelligence Laboratory (CSAIL)
MIT
torralba@mit.edu, billf@mit.edu
Accidental pinhole camera

window is an aperture

projected pattern on the wall

upside down

window with smaller gap

view outside window
Pinhole cameras

What are we imaging here?
Camera matrix
The camera as a coordinate transformation

A camera is a mapping from:

- the 3D world

to:

- a 2D image

3D object

3D to 2D transform (camera)

2D image

2D to 2D transform (image warping)

2D image
The camera as a coordinate transformation

A camera is a mapping from:
- the 3D world

to:
- a 2D image

What are the dimensions of each variable?

\[ x = PX \]

- \( x \): 2D image point
- \( P \): camera matrix
- \( X \): 3D world point

homogeneous coordinates
The camera as a coordinate transformation

\[ x = PX \]

\[
\begin{bmatrix}
X \\
Y \\
Y \\
Z
\end{bmatrix}
= 
\begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Y \\
1
\end{bmatrix}
\]

homogeneous image coordinates
3 x 1
camera matrix
3 x 4
homogeneous world coordinates
4 x 1
The pinhole camera

real-world object

camera center

focal length f

image plane
The (rearranged) pinhole camera

- real-world object
- image plane
- focal length $f$
- camera center
What is the equation for image coordinate $x$ in terms of $X$?
The 2D view of the (rearranged) pinhole camera

What is the equation for image coordinate $x$ in terms of $X$?
The 2D view of the (rearranged) pinhole camera

\[ [X \ Y \ Z]^\top \mapsto [fX/Z \ fY/Z]^\top \]
The (rearranged) pinhole camera

What is the camera matrix $P$ for a pinhole camera?

$x = PX$
The pinhole camera matrix

Relationship from similar triangles:

$$[X \ Y \ Z]^\top \rightarrow [fX/Z \ fY/Z]^\top$$

General camera model:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

What does the pinhole camera projection look like?

The pinhole camera matrix

Relationship from similar triangles:

\[
[X \ Y \ Z]^\top \mapsto [fX/Z \ fY/Z]^\top
\]

General camera model:

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} =
\begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\]

What does the pinhole camera projection look like?

\[
P =
\begin{bmatrix}
f & 0 & 0 & 0 \\
0 & f & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]
Generalizing the camera matrix

In general, the camera and image have *different* coordinate systems.
Generalizing the camera matrix

In particular, the camera origin and image origin may be different:

How does the camera matrix change?

\[
P = \begin{bmatrix}
    f & 0 & 0 & 0 \\
    0 & f & 0 & 0 \\
    0 & 0 & 1 & 0 \\
\end{bmatrix}
\]
Generalizing the camera matrix

In particular, the camera origin and image origin may be different:

\[ \mathbf{P} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \]
Camera matrix decomposition

We can decompose the camera matrix like this:

$$ P = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} $$

What does each part of the matrix represent?
Camera matrix decomposition

We can decompose the camera matrix like this:

\[
\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}
\]

(homogeneous) transformation from 2D to 2D, accounting for not unit focal length and origin shift

(homogeneous) projection from 3D to 2D, assuming image plane at \( z = 1 \) and shared camera/image origin

Also written as: \( \mathbf{P} = \mathbf{K}[\mathbf{I}|\mathbf{0}] \) where

\[
\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}
\]
Generalizing the camera matrix

In general, there are *three*, generally different, coordinate systems.

We need to know the transformations between them.
World-to-camera coordinate system transformation

$tilded$ means $heterogeneous$ coordinates
World-to-camera coordinate system transformation

World coordinate system

Camera coordinate system

Coordinate of the camera center in the world coordinate frame

\( \tilde{X}_w \)

\( X_w \)

\( Y_w \)

\( Z_w \)

\( x_c \)

\( y_c \)

\( z_c \)
World-to-camera coordinate system transformation

Why aren’t the points aligned?

\((\tilde{X}_w - \tilde{C})\)

translate
World-to-camera coordinate system transformation

\[ R \cdot (\tilde{X}_w - \tilde{C}) \]

rotate translate

points now coincide
Modeling the coordinate system transformation

In heterogeneous coordinates, we have:

\[
\tilde{X}_c = R \cdot (\tilde{X}_w - \tilde{C})
\]

How do we write this transformation in homogeneous coordinates?
Modeling the coordinate system transformation

In heterogeneous coordinates, we have:

\[ \tilde{X}_c = R \cdot (\tilde{X}_w - \tilde{C}) \]

In homogeneous coordinates, we have:

\[
\begin{bmatrix}
X_c \\
Y_c \\
Z_c \\
1
\end{bmatrix} = \begin{bmatrix}
R & -RC \\
0 & 1
\end{bmatrix} \begin{bmatrix}
X_w \\
Y_w \\
Z_w \\
1
\end{bmatrix} \quad \text{or} \quad X_c = \begin{bmatrix}
R & -R\tilde{C} \\
0 & 1
\end{bmatrix} X_w
\]
Incorporating the transform in the camera matrix

The previous camera matrix is for homogeneous 3D coordinates in camera coordinate system:

\[ \mathbf{x} = \mathbf{P} \mathbf{X}_c = \mathbf{K} [\mathbf{I} | \mathbf{0}] \mathbf{X}_c \]

We also just derived:

\[ \mathbf{X}_c = \begin{bmatrix} \mathbf{R} & -\mathbf{R} \mathbf{C} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{x}_w \]
Putting it all together

We can write everything into a single projection:

\[ \mathbf{x} = \mathbf{PX}_w \]

The camera matrix now looks like:

\[
\mathbf{P} = \begin{bmatrix}
    f & 0 & p_x \\
    0 & f & p_y \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    \mathbf{R} \\
    -\mathbf{RC}
\end{bmatrix}
\]

*intrinsic parameters* (3 x 3): correspond to camera internals (sensor not at f = 1 and origin shift)

*extrinsic parameters* (3 x 4): correspond to camera externals (world-to-image transformation)
General pinhole camera matrix

We can decompose the camera matrix like this:

\[
P = KR[I| - C]
\]

(translate first then rotate)

Another way to write the mapping:

\[
P = K[R|t]
\]

where \( t = -RC \)

(rotate first then translate)
General pinhole camera matrix

\[ P = K[R|t] \]

\[ P = \begin{bmatrix}
  f & 0 & p_x \\
  0 & f & p_y \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  r_1 & r_2 & r_3 & t_1 \\
  r_4 & r_5 & r_6 & t_2 \\
  r_7 & r_8 & r_9 & t_3
\end{bmatrix} \]

- **intrinsic parameters**
- **extrinsic parameters**

\[ R = \begin{bmatrix}
  r_1 & r_2 & r_3 \\
  r_4 & r_5 & r_6 \\
  r_7 & r_8 & r_9
\end{bmatrix} \quad t = \begin{bmatrix}
  t_1 \\
  t_2 \\
  t_3
\end{bmatrix} \]

- **3D rotation**
- **3D translation**
Recap

What is the size and meaning of each term in the camera matrix?

\[ P = KR[I| - C] \]

Recap

What is the size and meaning of each term in the camera matrix?

\[ P = KR[I| - C] \]

- 3x3 intrinsics
- ?
- ?
- ?
Recap

What is the size and meaning of each term in the camera matrix?

\[ P = KR[I] - C \]

- 3x3 intrinsics
- 3x3 3D rotation
- ?
- ?
Recap

What is the size and meaning of each term in the camera matrix?

\[ P = KR[I] - C \]

- 3x3 intrinsics
- 3x3 3D rotation
- 3x3 identity
- ?
Recap

What is the size and meaning of each term in the camera matrix?

\[ P = KR[I| - C] \]

- 3x3 intrinsics
- 3x3 3D rotation
- 3x3 identity
- 3x1 3D translation
Quiz

The camera matrix relates what two quantities?
Quiz

The camera matrix relates what two quantities?

\[ x = PX \]

homogeneous 3D points to 2D image points
Quiz

The camera matrix relates what two quantities?

\[ \mathbf{x} = \mathbf{PX} \]

homogeneous 3D points to 2D image points

The camera matrix can be decomposed into?
Quiz

The camera matrix relates what two quantities?

\[ x = PX \]

homogeneous 3D points to 2D image points

The camera matrix can be decomposed into?

\[ P = K[R|t] \]

intrinsic and extrinsic parameters
More general camera matrices

The following is the standard camera matrix we saw.

\[
P = \begin{bmatrix}
f & 0 & p_x \\
0 & f & p_y \\
0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
R & -RC \\
\end{bmatrix}
\]
More general camera matrices

CCD camera: pixels may not be square.

\[ P = \begin{bmatrix} \alpha_x & 0 & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & -RC \end{bmatrix} \]

How many degrees of freedom?
More general camera matrices

CCD camera: pixels may not be square.

\[
P = \begin{bmatrix}
\alpha_x & 0 & p_x \\
0 & \alpha_y & p_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
R & -RC
\end{bmatrix}
\]

How many degrees of freedom?

10 DOF
More general camera matrices

Finite projective camera: sensor be skewed.

\[
P = \begin{bmatrix}
\alpha_x & s & p_x \\
0 & \alpha_y & p_y \\
0 & 0 & 1
\end{bmatrix}
[\begin{bmatrix}
R & -RC
\end{bmatrix}

How many degrees of freedom?
More general camera matrices

Finite projective camera: sensor be skewed.

\[
P = \begin{bmatrix}
\alpha_x & s & p_x \\
0 & \alpha_y & p_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
R & -RC
\end{bmatrix}
\]

How many degrees of freedom?

11 DOF
Perspective distortion
Finite projective camera

$$P = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & -RC \end{bmatrix}$$

What does this matrix look like if the camera and world have the same coordinate system?
Finite projective camera

The pinhole camera and all of the more general cameras we have seen so far have “perspective distortion”.

\[
P = \begin{bmatrix}
    \alpha_x & s & p_x \\
    0 & \alpha_y & p_y \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 \\
    0 & 0 & 1 & 0
\end{bmatrix}
\]

*Perspective projection from (homogeneous) 3D to 2D coordinates*
The (rearranged) pinhole camera

Perspective projection in 3D

\[ \mathbf{x} = \mathbf{P} \mathbf{X} \]
The 2D view of the (rearranged) pinhole camera

Perspective projection in 2D

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}^\top \mapsto \begin{bmatrix}
fX/Z \\
fY/Z
\end{bmatrix}^\top
\]
Forced perspective
The Ames room illusion
The Ames room illusion

- Actual position of Person A
- Apparent position of person A
- Apparent shape of room
- Viewing peephole
- Actual and apparent position of person B

The Ames room illusion is a classic optical illusion where two individuals of different sizes appear to be the same size when viewed through a specific angle. This is due to the distortion created by the viewing peephole's shape and the room's perspective.
The arrow illusion
Magnification depends on depth

What happens as we change the focal length?

real-world object

depth Z

depth 2 Z
Magnification depends on focal length

real-world object

focal length $f$
focal length $2f$
What if…

1. Set focal length to half

real-world object

focal length \( f \)

depth 2 \( Z \)

focal length 2 \( f \)
What if…

1. Set focal length to half
2. Set depth to half

Is this the same image as the one I had at focal length $2f$ and distance $2Z$?
Perspective distortion

long focal length  
mid focal length  
short focal length
Perspective distortion
Vertigo effect

Named after Alfred Hitchcock’s movie
• also known as “dolly zoom”
Vertigo effect

How would you create this effect?
Other camera models
What if... real-world object

... we continue increasing $Z$ and $f$ while maintaining same magnification?

$f \to \infty$ and $\frac{f}{Z} = \text{constant}$
camera is close to object and has small focal length

camera is far from object and has large focal length
Different cameras

perspective camera

weak perspective camera
Weak perspective vs perspective camera

\[
\begin{bmatrix} X & Y & Z \end{bmatrix}^\top \rightarrow \begin{bmatrix} fX/Z_0 & fY/Z_0 \end{bmatrix}^\top
\]

- magnification does not change with depth
- \textit{constant} magnification depending on \( f \) and \( Z_0 \)
Comparing camera matrices

Let’s assume that the world and camera coordinate systems are the same.

- The *perspective* camera matrix can be written as:

\[
P = \begin{bmatrix}
f & 0 & p_x \\
0 & f & p_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]

- What would the matrix of the weak perspective camera look like?
Comparing camera matrices

Let’s assume that the world and camera coordinate systems are the same.

- The *perspective* camera matrix can be written as:

\[
P = \begin{bmatrix}
f & 0 & p_x \\
0 & f & p_y \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]

- The *weak perspective* camera matrix can be written as:

\[
P = \begin{bmatrix}
f & 0 & p_x \\
0 & f & p_y \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & Z_0
\end{bmatrix}
\]
Comparing camera matrices

Let’s assume that the world and camera coordinate systems are the same.

• The *finite projective* camera matrix can be written as:

\[
P = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}
\]

where we now have the more general intrinsic matrix

\[
K = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix}
\]

• The *affine* camera matrix can be written as:

\[
P = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & Z_0 \end{bmatrix}
\]

In both cameras, we can incorporate extrinsic parameters same as we did before.
When can we assume a weak perspective camera?
When can we assume a weak perspective camera?

1. When the scene (or parts of it) is very far away.

Weak perspective projection applies to the mountains.
When can we assume a weak perspective camera?

2. When we use a telecentric lens.

Place a pinhole at focal length, so that only rays parallel to primary ray pass through.
Orthographic camera

Special case of weak perspective camera where:

• constant magnification is equal to 1.
• there is no shift between camera and image origins.
• the world and camera coordinate systems are the same.

What is the camera matrix in this case?
Orthographic camera

Special case of weak perspective camera where:
• constant magnification is equal to 1.
• there is no shift between camera and image origins.
• the world and camera coordinate systems are the same.

\[ P = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix} \]
Orthographic projection using a telecentric lens

How do we make the telecentric lens act as an orthographic camera?

We set the sensor distance as:

\[ D' = 2f \]

in order to achieve unit magnification.
Many other types of cameras
Geometric camera calibration
<table>
<thead>
<tr>
<th></th>
<th>Structure (scene geometry)</th>
<th>Motion (camera geometry)</th>
<th>Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera Calibration (a.k.a. Pose Estimation)</td>
<td>known</td>
<td>estimate</td>
<td>3D to 2D correspondences</td>
</tr>
<tr>
<td>Triangulation</td>
<td>estimate</td>
<td>known</td>
<td>2D to 2D correspondences</td>
</tr>
<tr>
<td>Reconstruction</td>
<td>estimate</td>
<td>estimate</td>
<td>2D to 2D correspondences</td>
</tr>
</tbody>
</table>
Given a single image, estimate the exact position of the photographer
Geometric camera calibration

Given a set of matched points

\[ \{X_i, x_i\} \]

point in 3D space \hspace{1cm} point in the image

and camera model

\[ x = f(X; p) = PX \]

projection model \hspace{1cm} parameters \hspace{1cm} Camera matrix

Find the (pose) estimate of \( P \)

We’ll use a **perspective** camera model for pose estimation
Same setup as homography estimation
(slightly different derivation here)
Mapping between 3D point and image points

\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} = \begin{bmatrix}
  p_1 & p_2 & p_3 & p_4 \\
  p_5 & p_6 & p_7 & p_8 \\
  p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix} \begin{bmatrix}
  X \\
  Y \\
  Z \\
  1
\end{bmatrix}
\]

What are the unknowns?
Mapping between 3D point and image points

\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} =
\begin{bmatrix}
  p_1 & p_2 & p_3 & p_4 \\
  p_5 & p_6 & p_7 & p_8 \\
  p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\begin{bmatrix}
  X \\
  Y \\
  Z \\
  1
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} =
\begin{bmatrix}
  \text{---} & \text{---} & p_1^\top & \text{---} \\
  \text{---} & \text{---} & p_2^\top & \text{---} \\
  \text{---} & \text{---} & p_3^\top & \text{---}
\end{bmatrix}
\begin{bmatrix}
  X
\end{bmatrix}
\]

Heterogeneous coordinates

\[
x' = \frac{p_1^\top X}{p_3^\top X} \quad y' = \frac{p_2^\top X}{p_3^\top X}
\]

(non-linear relation between coordinates)

*How can we make these relations linear?*
How can we make these relations linear?

\[ x' = \frac{p_1^T X}{p_3^T X}, \quad y' = \frac{p_2^T X}{p_3^T X} \]

Make them linear with algebraic manipulation…

\[ p_2^T X - p_3^T X y' = 0 \]

\[ p_1^T X - p_3^T X x' = 0 \]

Now we can setup a system of linear equations with multiple point correspondences.
\[ p_2^\top X - p_3^\top X y' = 0 \]

\[ p_1^\top X - p_3^\top X x' = 0 \]

*How do we proceed?*
\[ p_2^\top X - p_3^\top X y' = 0 \]
\[ p_1^\top X - p_3^\top X x' = 0 \]

In matrix form …

\[
\begin{bmatrix}
  X^\top & 0 & -x' X^\top \\
  0 & X^\top & -y' X^\top \\
\end{bmatrix}
\begin{bmatrix}
  p_1 \\
  p_2 \\
  p_3 \\
\end{bmatrix} = 0
\]

*How do we proceed?*
In matrix form ...

\[
\begin{bmatrix}
X^\top & 0 & -x'X^\top \\
0 & X^\top & -y'X^\top
\end{bmatrix}
\begin{bmatrix}
p_1 \\
p_2 \\
p_3
\end{bmatrix} = 0
\]

For N points ...

\[
\begin{bmatrix}
X_1^\top & 0 & -x'X_1^\top \\
0 & X_1^\top & -y'X_1^\top \\
\vdots & \vdots & \vdots \\
X_N^\top & 0 & -x'X_N^\top \\
0 & X_N^\top & -y'X_N^\top
\end{bmatrix}
\begin{bmatrix}
p_1 \\
p_2 \\
p_3
\end{bmatrix} = 0
\]
Solve for camera matrix by

\[ \hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \| \mathbf{A} \mathbf{x} \|^2 \text{ subject to } \| \mathbf{x} \|^2 = 1 \]

\[ \mathbf{A} = \begin{bmatrix}
\mathbf{X}_1^\top & 0 & -x' \mathbf{X}_1^\top \\
0 & \mathbf{X}_1^\top & -y' \mathbf{X}_1^\top \\
\vdots & \vdots & \vdots \\
\mathbf{X}_N^\top & 0 & -x' \mathbf{X}_N^\top \\
0 & \mathbf{X}_N^\top & -y' \mathbf{X}_N^\top
\end{bmatrix} \]

\[ \mathbf{x} = \begin{bmatrix}
p_1 \\
p_2 \\
p_3
\end{bmatrix} \]

**SVD!**
Solve for camera matrix by

\[ \hat{x} = \arg \min_{x} \|Ax\|^2 \text{ subject to } \|x\|^2 = 1 \]

\[
A = \begin{bmatrix}
X_1^\top & 0 & -x'X_1^\top \\
0 & X_1^\top & -y'X_1^\top \\
\vdots & \vdots & \vdots \\
X_N^\top & 0 & -x'X_N^\top \\
0 & X_N^\top & -y'X_N^\top
\end{bmatrix}
\]

\[
x = \begin{bmatrix}
p_1 \\
p_2 \\
p_3
\end{bmatrix}
\]

Solution \( x \) is the column of \( V \) corresponding to smallest singular value of \( A = U\Sigma V^\top \)
Solve for camera matrix by

\[ \hat{x} = \arg \min_x \| A x \|^2 \text{ subject to } \| x \|^2 = 1 \]

\[
A = \begin{bmatrix}
X_1^\top & 0 & -x' X_1^\top \\
0 & X_1^\top & -y' X_1^\top \\
\vdots & \vdots & \vdots \\
X_N^\top & 0 & -x' X_N^\top \\
0 & X_N^\top & -y' X_N^\top 
\end{bmatrix}
\]

\[ x = \begin{bmatrix}
p_1 \\
p_2 \\
p_3 
\end{bmatrix} \]

Equivalently, solution \( x \) is the Eigenvector corresponding to smallest Eigenvalue of

\[ A^\top A \]
Now we have: $\mathbf{P} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix}$

Are we done?
Almost there … 

\[
P = \begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\]

How do you get the intrinsic and extrinsic parameters from the projection matrix?
Decomposition of the Camera Matrix

\[ P = \begin{bmatrix}
  p_1 & p_2 & p_3 & p_4 \\
  p_5 & p_6 & p_7 & p_8 \\
  p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix} \]
Decomposition of the Camera Matrix

\[
P = \begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\]

\[
P = K[R | t]
\]
Decomposition of the Camera Matrix

\[ P = \begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix} \]

\[ P = K[R|t] \]
\[ = K[R| - Rc] \]
\[ = [M| - Mc] \]
Decomposition of the Camera Matrix

\[
P = \begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\]

\[
P = K[R|t]
\]

\[
= K[R] - Rc
\]

\[
= [M] - Mc
\]

Find the camera center \( C \)

Find intrinsic \( K \) and rotation \( R \)

*What is the projection of the camera center?*
Decomposition of the Camera Matrix

\[
P = \begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\]

\[
P = K[R|t]
\]

\[
= K[R| - Rc]
\]

\[
= [M| - Mc]
\]

Find the camera center \( C \)

\[
P_c = 0
\]

*How do we compute the camera center from this?*

Find intrinsic \( K \) and rotation \( R \)
Decomposition of the Camera Matrix

\[
P = \begin{bmatrix}
  p_1 & p_2 & p_3 & p_4 \\
  p_5 & p_6 & p_7 & p_8 \\
  p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\]

\[
P = K[R|t]
\]

\[
= K[R| - Rc]
\]

\[
= [M| - Mc]
\]

Find the camera center \( C \)

\[
Pc = 0
\]

SVD of \( P! \)

\( c \) is the Eigenvector corresponding to smallest Eigenvalue

Find intrinsic \( K \) and rotation \( R \)
Decomposition of the Camera Matrix

\[
P = \begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\]

\[
P = K[R|t]
\]
\[
= K[R] - Rc
\]
\[
= [M] - Mc
\]

Find the camera center \( C \)

\[
Pc = 0
\]

\[
\text{SVD of } P!
\]

\( c \) is the Eigenvector corresponding to smallest Eigenvalue

Find intrinsic \( K \) and rotation \( R \)

\[
M = KR
\]

Any useful properties of \( K \) and \( R \) we can use?
Decomposition of the Camera Matrix

\[ P = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \]

\[ P = K[R | t] \]
\[ = K[R | - Rc] \]
\[ = [M | - Mc] \]

Find the camera center \( c \)

\[ Pc = 0 \]
SVD of \( P \)

\( c \) is the Eigenvector corresponding to smallest Eigenvalue

Find intrinsic \( K \) and rotation \( R \)

\[ M = KR \]
right upper triangle
orthogonal

How do we find \( K \) and \( R \)?
Decomposition of the Camera Matrix

\[ P = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \]

\[ P = K[R|t] \]
\[ = K[R] - Rc \]
\[ = [M] - Mc \]

Find the camera center \( C \)
\[ Pc = 0 \]
SVD of \( P \!
\]
\( c \) is the Eigenvector corresponding to smallest Eigenvalue

Find intrinsic \( K \) and rotation \( R \)
\[ M = KR \]
QR decomposition
Geometric camera calibration

Given a set of matched points
\[ \{X_i, x_i\} \]

point in 3D space
point in the image

and camera model
\[ x = f(X; p) = PX \]

projection model
parameters
Camera matrix

Find the (pose) estimate of

We’ll use a **perspective** camera model for pose estimation
Calibration using a reference object

Place a known object in the scene:

- identify correspondences between image and scene
- compute mapping from scene to image

Issues:

- must know geometry very accurately
- must know 3D->2D correspondence
For these reasons, **nonlinear methods** are preferred

- Define error function $E$ between projected 3D points and image positions
  - $E$ is nonlinear function of intrinsics, extrinsics, radial distortion
- Minimize $E$ using nonlinear optimization techniques

**Advantages:**
- Very simple to formulate.
- Analytical solution.

**Disadvantages:**
- Doesn’t model radial distortion.
- Hard to impose constraints (e.g., known $f$).
- Doesn’t minimize the correct error function.

Geometric camera calibration
Minimizing reprojection error

\[
\left( u_i - \frac{m_1 \cdot P_i}{m_3 \cdot P_i} \right)^2 + \left( v_i - \frac{m_2 \cdot P_i}{m_3 \cdot P_i} \right)^2
\]

Is this equivalent to what we were doing previously?
Radial distortion

What causes this distortion?

no distortion  barrel distortion  pincushion distortion
Radial distortion model

Ideal:

\[ x' = f \frac{x}{z} \]
\[ y' = f \frac{y}{z} \]

Distorted:

\[ x'' = \frac{1}{\lambda} x' \]
\[ y'' = \frac{1}{\lambda} y' \]

\[ \lambda = 1 + k_1 r^2 + k_2 r^4 + \cdots \]
Minimizing reprojection error with radial distortion

Add distortions to reprojection error:

$$\left( u_i - \frac{1}{\lambda m_3 \cdot P_i} \right)^2 + \left( v_i - \frac{1}{\lambda m_3 \cdot P_i} \right)^2$$
Correcting radial distortion

before

after
Advantages:

- Only requires a plane
- Don’t have to know positions/orientations
- Great code available online!
  - Also available on OpenCV.

Disadvantage: Need to solve non-linear optimization problem.
Step-by-step demonstration
Step-by-step demonstration

Click on the four extreme corners of the rectangular pattern...

Click #1 (origin)

Click #2

Click #3

Click #4
Step-by-step demonstration
Step-by-step demonstration

Image points (+) and reprojected grid points (o)
Step-by-step demonstration
What does it mean to “calibrate a camera”?
What does it mean to “calibrate a camera”?

Many different ways to calibrate a camera:

• Radiometric calibration.

• Color calibration.

• Geometric calibration.

• Noise calibration.

• Lens (or aberration) calibration.

We’ll briefly discuss radiometric and color calibration in later lectures. For the rest, see 15-463/663/862.
Simple AR program

1. Compute point correspondences (2D and AR tag)
2. Estimate the pose of the camera $P$
3. Project 3D content to image plane using $P$

3D locations of planar marker features are known in advance

3D content prepared in advance
References

Basic reading:
• Szeliski textbook, Section 2.1.5, 6.2.

Additional reading:
• Torralba and Freeman, “Accidental Pinhole and Pinspeck Cameras,” CVPR 2012. the eponymous paper discussed in the slides.