Geometric camera models and calibration
Course announcements

• Homework 3 is out.
  - Due October 12th **tonight**.
  - Any questions?

• Homework 4 will be posted tonight and will be on lightfields.

• Due October 21st: Project ideas posted on Piazza.

• Next week I’ll schedule extra office hours for those of you who cannot make it to the make-up lecture, and also so that you can discuss with me about the final project.

• Additional guest lecture next Monday: Anat Levin, “Coded photography.”
Overview of today’s lecture

- Leftover from lecture 12
- Reminder about pinhole and lens cameras
- Camera matrix.
- Perspective distortion.
- Other camera models.
- Geometric camera calibration.
Many of these slides were adapted from:

- Kris Kitani (15-463, Fall 2016).
- Srinivasa Narasimhan (16-720, Fall 2017).
- Noah Snavely (Cornell).

Some slides inspired from:

- Fredo Durand (MIT).
Pinhole and lens cameras
The lens camera
The pinhole camera
The pinhole 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.
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)
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?
The camera as a coordinate transformation

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

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
= 
\begin{bmatrix}
p_{1} & p_{2} & p_{3} & p_{4} \\
p_{5} & p_{6} & p_{7} & p_{8} \\
p_{9} & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\]

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

image plane

focal length $f$
The (rearranged) pinhole camera

Image plane

Real-world object

Focal length f

Camera center
The (rearranged) pinhole camera

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

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

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

$$\mathbf{x} = \mathbf{PX}$$
The pinhole camera matrix

Relationship from similar triangles:

\[
\begin{bmatrix} X & Y & Z \end{bmatrix}^T \leftrightarrow \begin{bmatrix} fX/Z & fY/Z \end{bmatrix}^T
\]

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:

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

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:

How does the camera matrix change?

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

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

(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: \[ P = K[I|0] \] where \[ K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \]
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

tilde means heterogeneous coordinates
World-to-camera coordinate system transformation

Coordinate of the camera center in the world coordinate frame
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:

\[
\begin{bmatrix}
X_c \\
Y_c \\
Z_c \\
1
\end{bmatrix} = \begin{bmatrix}
R & -RC \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
X_w \\
Y_w \\
Z_w \\
1
\end{bmatrix}
\]

or

\[ X_c = \begin{bmatrix} R & -R\tilde{C} \end{bmatrix} X_w \]
Incorporating the transform in the camera matrix

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

\[
x = PX_c = K[I|0]X_c
\]

We also just derived:

\[
X_c = \begin{bmatrix} R & -R\hat{C} \\ 0 & 1 \end{bmatrix} X_w
\]
Putting it all together

We can write everything into a single projection:

\[ \mathbf{x} = \mathbf{P} \mathbf{X}_w \]

The camera matrix now looks like:

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

*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-camera 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**

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

\[
t = \begin{bmatrix}
t_1 \\
t_2 \\
t_3
\end{bmatrix}
\]

**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
The camera matrix relates what two quantities?

\[ x = Px \]

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.

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

CCD camera: pixels may not be square.

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

How many degrees of freedom?
More general camera matrices

CCD camera: pixels may not be square.

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

How many degrees of freedom?

10 DOF
More general camera matrices

Finite projective camera: sensor be skewed.

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

How many degrees of freedom?
More general camera matrices

Finite projective camera: sensor be skewed.

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

How many degrees of freedom?

11 DOF
Perspective distortion
Finite projective camera

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

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

\[ x = PX \]
The 2D view of the (rearranged) pinhole camera

Perspective projection in 2D

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

Perspective distortion: magnification changes with depth
Forced perspective
The Ames room illusion
The Ames room illusion
Other camera models
What if… we continue increasing $Z$ and $f$ while maintaining the same magnification?

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

Camera is far from object and has large focal length.

Increasing focal length.

Increasing distance from camera.
Different cameras

perspective camera

weak perspective camera
Weak perspective vs perspective camera

\[ [X \ Y \ Z]^\top \rightarrow [fX/Z_0 \ fY/Z_0]^\top \]

- magnification does not change with depth
- constant magnification depending on \( f \) and \( Z_0 \)

\( Z_0 \)

\( f \)

\( Z \)

image plane

magnification changes with depth
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:

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

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

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.

What is the magnification factor in this case?
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.

\[ m = \frac{D' - f}{f} \]

- magnification is constant with depth
- remember that focal length \( f \) refers to different things in pinhole and lens cameras
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.

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

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

object distance $D$

focal length $f$

sensor distance $D'$
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
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

\[ P \]

We’ll use a perspective camera model for pose estimation
Same setup as homography estimation
(slightly different derivation here)

*Where did we see homography estimation in this class?*
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

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

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

Heterogeneous coordinates

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

(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
\[ p_2^T X - p_3^T X y' = 0 \]
\[ p_1^T X - p_3^T X x' = 0 \]

*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 ... \[
\begin{bmatrix}
X_1^\top & 0 & -x'X_1^\top \\
0 & X_1^\top & -y'X_1^\top \\
\vdots & \vdots & \vdots \\
X_N^\top & 0 & -x'X_N^\top \\
0 & X_N^\top & -y'X_N^\top \\
\end{bmatrix}
\begin{bmatrix}
p_1 \\
p_2 \\
p_3 \\
\end{bmatrix} = 0
\]

How do we solve this system?
Solve for camera matrix by

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

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

\[
x = \begin{bmatrix}
p_1 \\
p_2 \\
p_3
\end{bmatrix}
\]
Solve for camera matrix by

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

\[
A = \begin{bmatrix}
    X_1^T & 0 & -x' X_1^T \\
    0 & X_1^T & -y' X_1^T \\
    \vdots & \vdots & \vdots \\
    X_N^T & 0 & -x' X_N^T \\
    0 & X_N^T & -y' X_N^T \\
\end{bmatrix}
\]

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

Solution \( x \) is the column of \( V \) corresponding to smallest singular value of

\[ A = U \Sigma V^T \]
Solve for camera matrix by

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

\[
A = \begin{bmatrix}
X_1^T & 0 & -x'X_1^T \\
0 & X_1^T & -y'X_1^T \\
\vdots & \vdots & \vdots \\
X_N^T & 0 & -x'X_N^T \\
0 & X_N^T & -y'X_N^T
\end{bmatrix}
\]

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

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

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

Are we done?
Almost there …  

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

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 \( C \)

What is the projection of the camera center?

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

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

\[ \mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \]
\[ = \mathbf{K}[\mathbf{R}] - \mathbf{K} \mathbf{R}_c \]
\[ = [\mathbf{M}] - \mathbf{M}_c \]

Find the camera center \( \mathbf{C} \)

\[ \mathbf{P}_c = 0 \]

How do we compute the camera center from this?

Find intrinsic \( \mathbf{K} \) and rotation \( \mathbf{R} \)
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 \( C \)
\[ P_c = 0 \]

**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 \( C \)

\[ Pc = 0 \]

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 \[ P \]

Where do we get these matched points from?

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
Geometric camera calibration

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.

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
Minimizing reprojection error

Is this equivalent to what we were doing previously?

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

What causes this distortion?

no distortion  barrel distortion  pincushion distortion
Radial distortion model

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

Distorted: \[
x'' = \frac{1}{\lambda} x' \quad y'' = \frac{1}{\lambda} y' \quad \lambda = 1 + k_1 r^2 + k_2 r^4 + \cdots
\]
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!
  - Matlab version: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
  - Also available on OpenCV.

Disadvantage: Need to solve non-linear optimization problem.
Step-by-step demonstration
Step-by-step demonstration
Step-by-step demonstration
Step-by-step demonstration
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.  
  → lecture 5

• Color calibration.  
  → lecture 6

• Geometric calibration.  
  → lecture 13

• Noise calibration.  
  → later lecture

• Lens (or aberration) calibration.  
  → lecture 12, (maybe) later lecture
References

Basic reading:
• Szeliski textbook, Section 2.1.5, 6.2.
  The main resource for camera calibration in Matlab, where the screenshots in this lecture were taken from. It also has a detailed of the camera calibration algorithm and an extensive reference section.

Additional reading: