EE 417 Introduction to Computer Vision / EE 569 3D vision

Image Formation and Camera Calibration

Instructor: Associate Prof. Mehmet Keskinöz

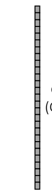
Communication Theory &Technologies (CTT) Group,
Electronics Engineering Program,
Computer Science and Engineering,
Cyber-Security Program
Faculty of Engineering and Natural Sciences

Sabancı Üniversitesi

Email: keskinoz@sabanciuniv.edu

1

Let's say we have a sensor...



2

... and an object we like to photograph

What would an image taken like this look like?

3

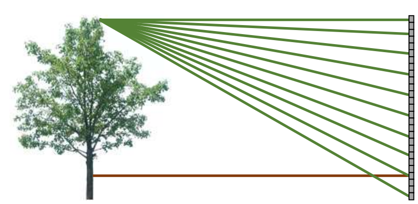
Bare-sensor imaging

4

Bare-sensor imaging

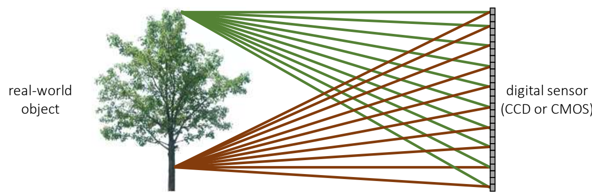
5

Bare-sensor imaging



6

Bare-sensor imaging



All scene points contribute to all sensor pixels

What does the image on the sensor look like?

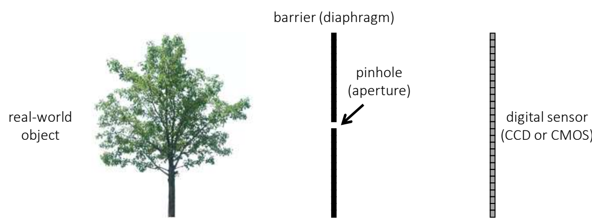
7

Bare-sensor imaging

All scene points contribute to all sensor pixels

8

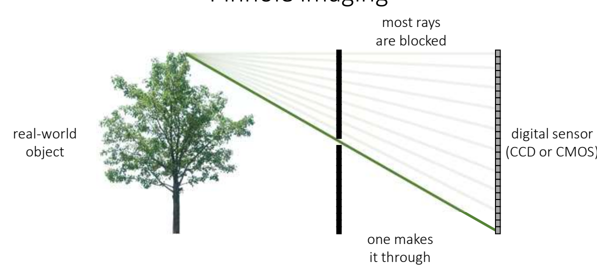
Let's add something to this scene



What would an image taken like this look like?

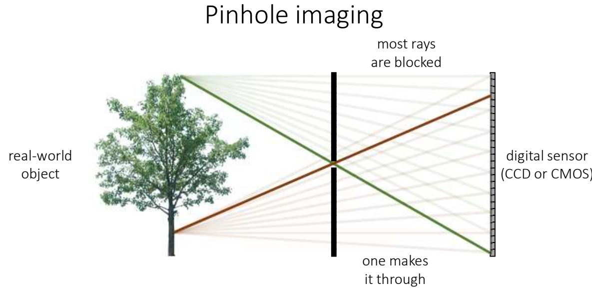
9

Pinhole imaging



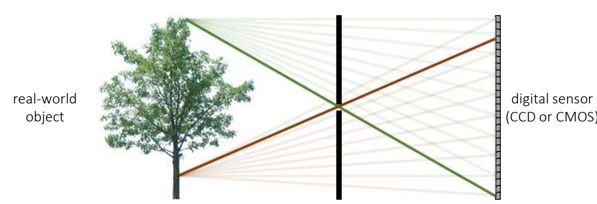
10

Pinhole imaging



11

Pinhole imaging

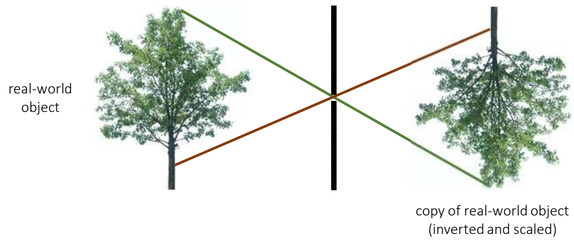


Each scene point contributes to only one sensor pixel

What does the image on the sensor look like?

12

Pinhole imaging

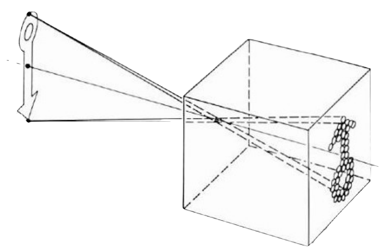


13

Pinhole camera

14

Pinhole camera a.k.a. camera obscura



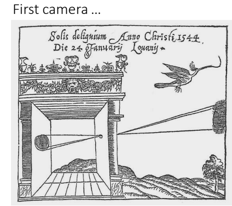
15

Pinhole camera a.k.a. camera obscura

First mention ...

Chinese philosopher Mozi (470 to 390 BC)

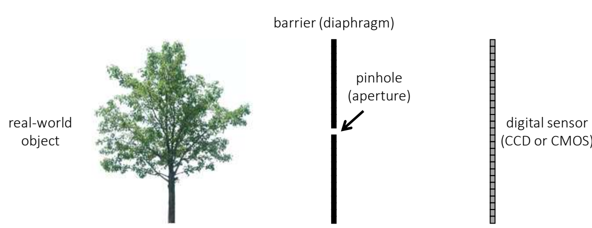
First camera ...



Greek philosopher Aristotle (384 to 322 BC)

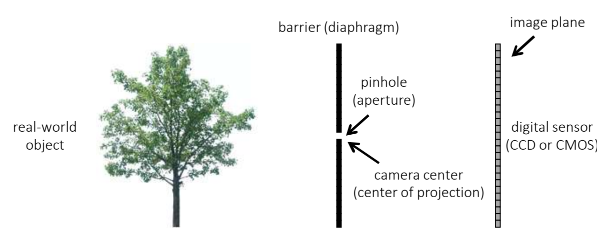
16

Pinhole camera terms



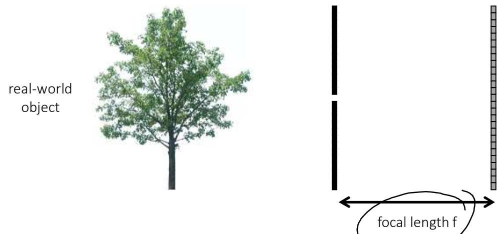
17

Pinhole camera terms



18

Focal length

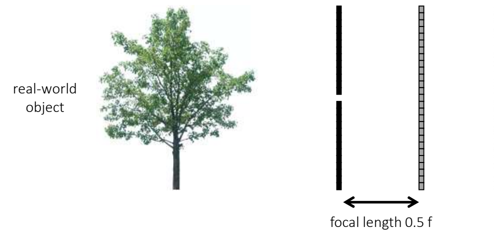


distance from the aperture to the sensor

19

Focal length

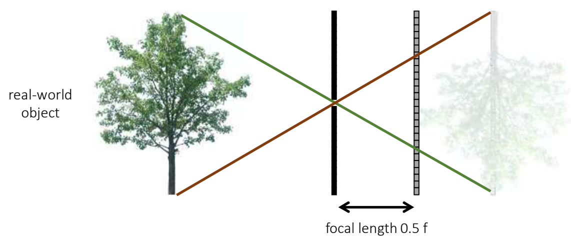
What happens as we change the focal length?



20

Focal length

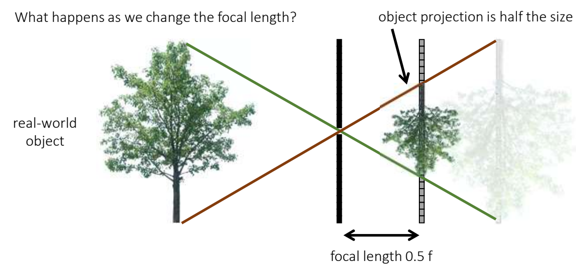
What happens as we change the focal length?



21

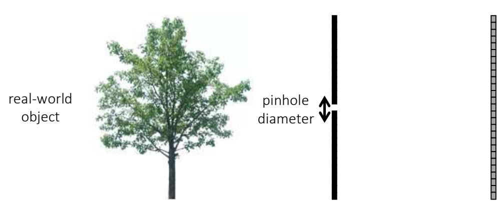
Focal length

What happens as we change the focal length?



22

Pinhole size

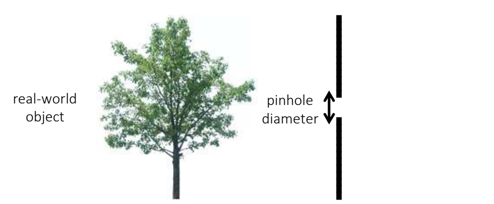


Ideal pinhole has infinitesimally small size
- In practice that is impossible.

23

Pinhole size

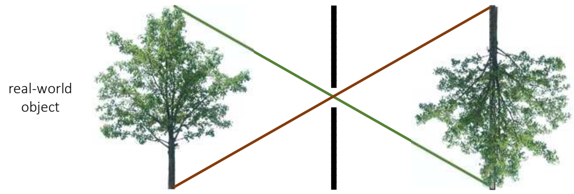
What happens as we change the pinhole diameter?



24

Pinhole size

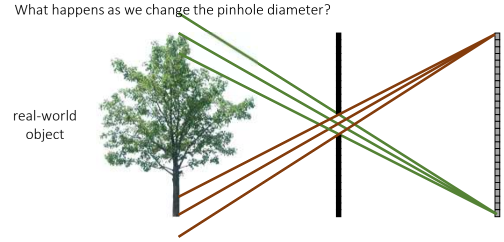
What happens as we change the pinhole diameter?



25

Pinhole size

What happens as we change the pinhole diameter?

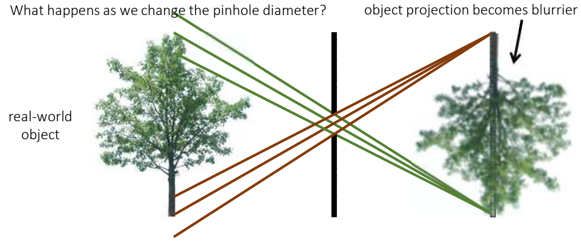


26

Pinhole size

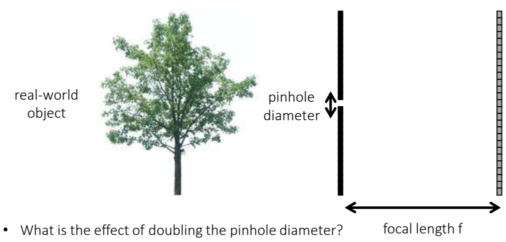
What happens as we change the pinhole diameter?

object projection becomes blurrier



27

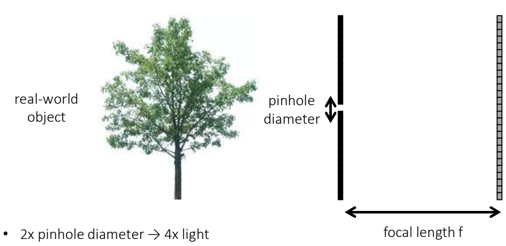
What about light efficiency?



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

28

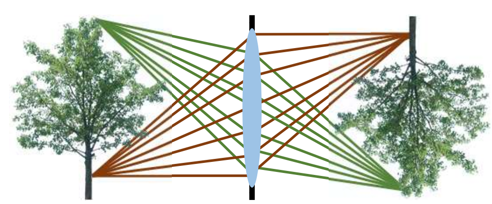
What about light efficiency?



- 2x pinhole diameter → 4x light
- 2x focal length → ¼x light

29

The lens camera

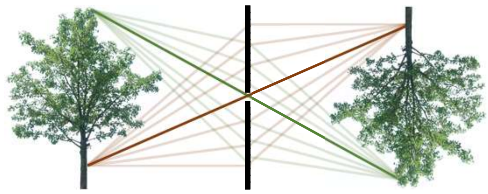


Lenses map “bundles” of rays from points on the scene to the sensor.

How does this mapping work exactly?

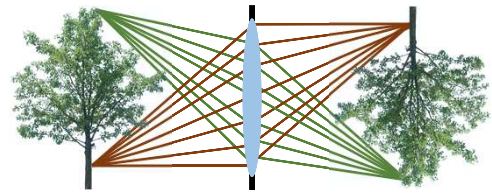
30

The pinhole camera



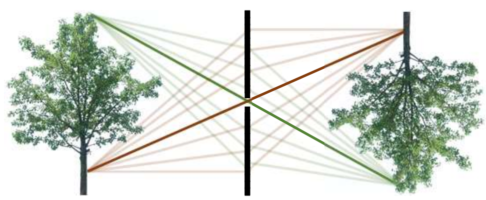
31

The lens camera



32

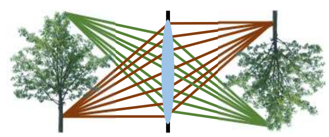
The pinhole camera

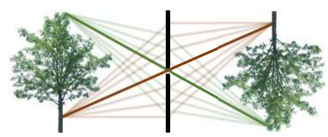


Central rays propagate in the same way for both models!

33

Describing both lens and pinhole cameras





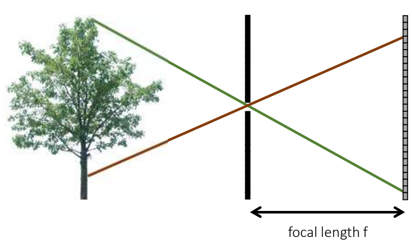
We can derive properties and descriptions that hold for both camera models if:

- We use only central rays.
- We assume the lens camera is in focus.

34

Important difference: focal length

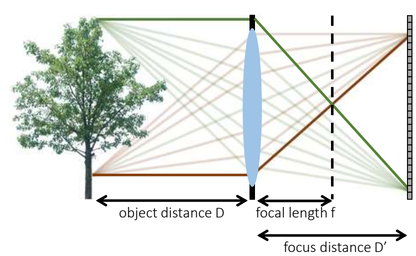
In a pinhole camera, focal length is distance between aperture and sensor



35

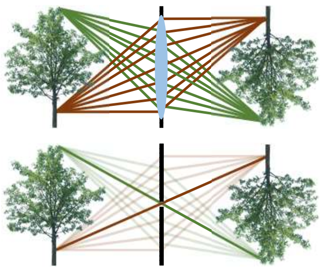
Important difference: focal length

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



36

Describing both lens and pinhole cameras

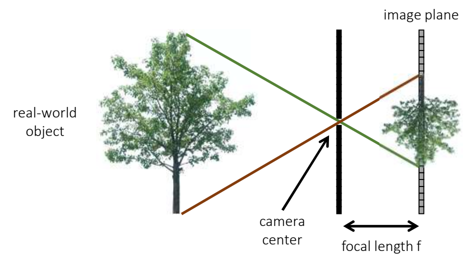


We can derive properties and descriptions that hold for both camera models if:
- We use only central rays.
- We assume the lens camera is in focus.
- We assume that the focus distance of the lens camera is equal to the focal length of the pinhole camera.

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.

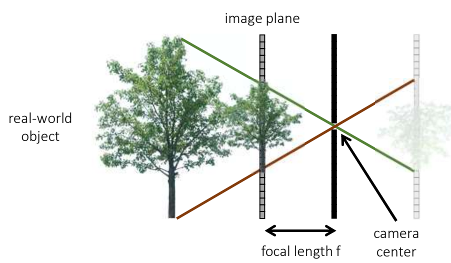
37

The pinhole camera



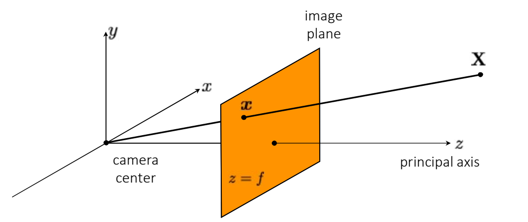
38

The (rearranged) pinhole camera



39

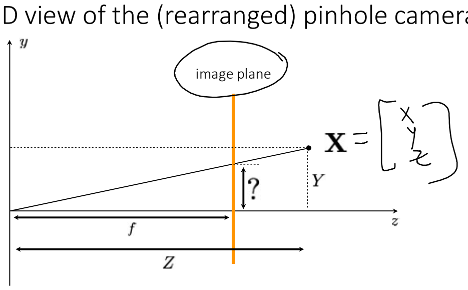
The (rearranged) pinhole camera



What is the equation for image coordinate **x** in terms of **X**?

40

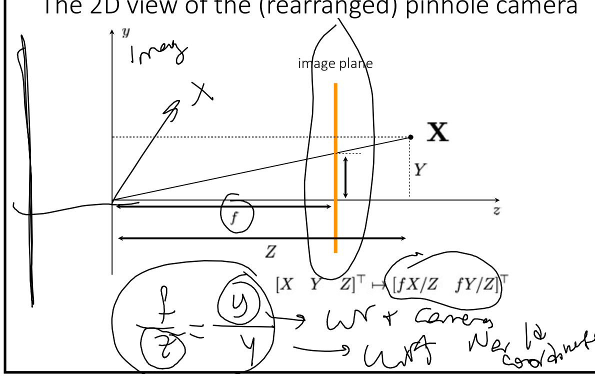
The 2D view of the (rearranged) pinhole camera



What is the equation for image coordinate **x** in terms of **X**?

41

The 2D view of the (rearranged) pinhole camera



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

$\frac{f}{Z} = \frac{y}{Y}$ → wrt camera → wrt World coordinate

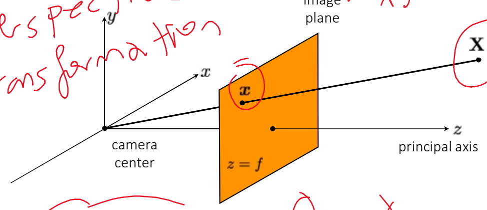
42

Orthographic projection depth Z of the object

The (rearranged) pinhole camera

$x = X'$, $y = Y'$

Perspective Transformation



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

division by Z is non-linear

43

Trick: add one more coordinate:

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous image (2D) coordinates — homogeneous scene (3D) coordinates

Converting *from* homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w) \qquad \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Homogenous coordinates invariant under scale

Inhomogeneous = heterogeneous = world coordinate

44

Homogeneous coordinates

heterogeneous coordinates — homogeneous coordinates

$$\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

add a 1 here

- Represent 2D point with a 3D vector

45

Homogeneous coordinates

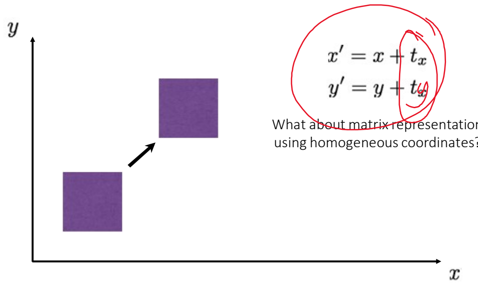
heterogeneous coordinates — homogeneous coordinates

$$\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} ax \\ ay \\ a \end{bmatrix}$$

- Represent 2D point with a 3D vector
- 3D vectors are only defined up to scale

46

2D translation



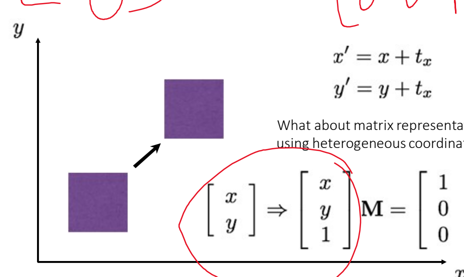
$$x' = x + t_x$$
$$y' = y + t_y$$

What about matrix representation using homogeneous coordinates?

47

$$\begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2D translation



$$x' = x + t_x$$
$$y' = y + t_x$$

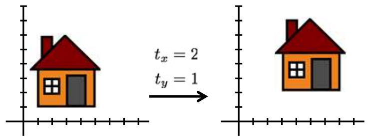
What about matrix representation using heterogeneous coordinates?

$$\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \mathbf{M} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

48

2D translation using homogeneous coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$



49

Homogeneous coordinates

Conversion:

•heterogeneous → homogeneous

$$\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

•homogeneous → heterogeneous

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow \begin{bmatrix} x/w \\ y/w \end{bmatrix}$$

•scale invariance

$$\begin{bmatrix} x & y & w \end{bmatrix}^\top = \lambda \begin{bmatrix} x & y & w \end{bmatrix}^\top$$

Special points:

- point at infinity

$$\begin{bmatrix} x & y & 0 \end{bmatrix}$$

- undefined

$$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

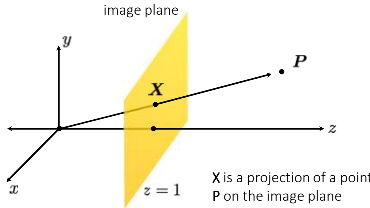
50

Projective geometry

image point in pixel coordinates $\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

⇕

image point in homogeneous coordinates $\boldsymbol{X} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$



X is a projection of a point **P** on the image plane

What does scaling **X** correspond to?

51

2D transformations in heterogeneous coordinates

Re-write these transformations as 3x3 matrices:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} & & \\ & ? & \\ & & \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

scaling

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} & & \\ & ? & \\ & & \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

rotation

52

2D transformations in heterogeneous coordinates

Re-write these transformations as 3x3 matrices:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

scaling

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} & & \\ & ? & \\ & & \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

rotation

53

2D transformations in homogenous coordinates

Re-write these transformations as 3x3 matrices:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

scaling

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} & & \\ & ? & \\ & & \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

rotation

54

2D transformations in Homogenous coordinates

Re-write these transformations as 3x3 matrices:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

scaling

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

rotation

55

Matrix composition

Transformations can be combined by matrix multiplication:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

p' = ? ? ? **p**

56

Matrix composition

Transformations can be combined by matrix multiplication:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

p' = translation(t_x,t_y) rotation(θ) scale(s,s) **p**

Does the multiplication order matter?

57

Classification of 2D transformations

Translation: $\begin{bmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & 1 \end{bmatrix}$

Dof is 2

How many degrees of freedom?

(or free) // # of independent variables

58

Classification of 2D transformations

Euclidean (rigid):
rotation + translation
$\begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ 0 & 0 & 1 \end{bmatrix}$

59

Classification of 2D transformations

Euclidean (rigid):
rotation + translation
$\begin{bmatrix} \cos\theta & -\sin\theta & r_3 \\ \sin\theta & \cos\theta & r_6 \\ 0 & 0 & 1 \end{bmatrix}$

tx ty

How many degrees of freedom?

Dof = 3

60

Classification of 2D transformations

which other matrix values will change if this increases?

Euclidean (rigid): rotation + translation

$$\begin{bmatrix} \cos\theta & -\sin\theta & r_3 \\ \sin\theta & \cos\theta & r_6 \\ 0 & 0 & 1 \end{bmatrix}$$

61

Classification of 2D transformations

what will happen to the image if this increases?

Euclidean (rigid): rotation + translation

$$\begin{bmatrix} \cos\theta & -\sin\theta & r_3 \\ \sin\theta & \cos\theta & r_6 \\ 0 & 0 & 1 \end{bmatrix}$$

62

Classification of 2D transformations

what will happen to the image if this increases?

Euclidean (rigid): rotation + translation

$$\begin{bmatrix} \cos\theta & -\sin\theta & r_3 \\ \sin\theta & \cos\theta & r_6 \\ 0 & 0 & 1 \end{bmatrix}$$

63

Classification of 2D transformations

Similarity: uniform scaling + rotation + translation

$$\begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ 0 & 0 & 1 \end{bmatrix}$$

Are there any values that are related?

64

Classification of 2D transformations

multiply these four by scale **s**

Similarity: uniform scaling + rotation + translation

$$\begin{bmatrix} \boxed{\begin{matrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{matrix}} & \begin{matrix} r_3 \\ r_6 \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & 1 \end{bmatrix}$$

How many degrees of freedom?

65

Classification of 2D transformations

what will happen to the image if this increases?

Similarity: uniform scaling + rotation + translation

$$\begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ 0 & 0 & 1 \end{bmatrix}$$

66

What is camera calibration?

- The process of estimating camera parameters
- The 3D world coordinates are projected on the 2D image plane (film)
- The relationship between the world coordinates and image coordinates is defined by the camera parameters

67

Why do we need to calibrate cameras?

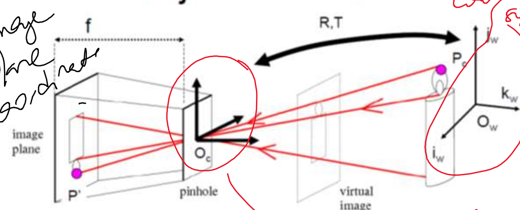
- To estimate the 3D geometry of the world → geometric calibration
- To correct imaging artefacts caused by imperfect lenses
- Examples include
  - Stereo reconstruction
  - Multiview reconstruction
  - Single view measurements such as the height of a person
  - Lens distortion correction

68

Projective camera

image plane coordinate

world coordinate system



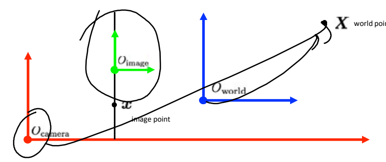
Camera coordinate system

- For convenience, the image plane is usually represented in front so that the image preserves the same orientation (i.e. not flipped)

69

Generalizing the camera matrix

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



We need to know the transformations between them.

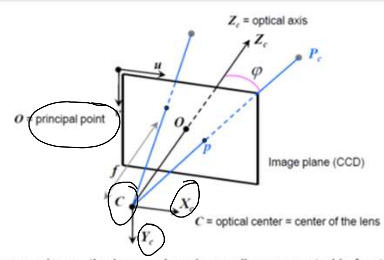
70

Geometric Camera calibration

- Made up of 2 transformations:
  - From some (arbitrary) world coordinate system to the camera's 3D coordinate system. ***Extrinisic parameters*** *(camera pose)*
  - From the 3D coordinates in the camera frame to the 2D image plane via projection. ***Intrinisic paramters***

71

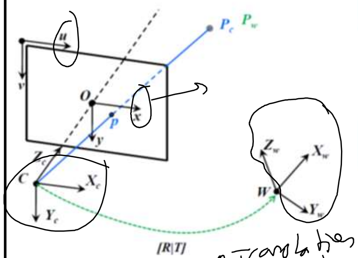
Perspective Camera



- For convenience, the image plane is usually represented in front so that the image preserves the same orientation (i.e. not flipped)
- Note: **a camera does not measure distances but angles!**
  ⇨ a camera is a "bearing sensor"

72

Perspective Projection: from world to pixel coords



Find pixel coordinates ***(u,v)*** of point P_w in the world frame:

0. Convert world point P_w to camera point P_c

Find pixel coordinates ***(u,v)*** of point P_c in the camera frame:

1. Convert P_c to image-plane coordinates ***(x,y)***

2. Convert P_c to (discretised) pixel coordinates ***(u,v)***

↑ Translation of the world wrt camera

R T

↳ wrt ~~Rco~~ rotation of the world coordinate wrt camera

73

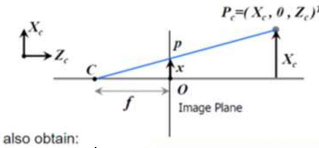
Perspective Projection (1)

From the Camera frame to the image plane

pixel size Δx Δy

- The 3D camera point $P_c = (X_c, 0, Z_c)^T$ projects to point $p = (x, y)$ onto the image plane
- Analysing similar triangles:

$$\frac{x}{f} = \frac{X_c}{Z_c} \Rightarrow x = \frac{fX_c}{Z_c}$$



- So, for the 3D case, we can also obtain:

$$\frac{y}{f} = \frac{Y_c}{Z_c} \Rightarrow y = \frac{fY_c}{Z_c}$$

$u = \frac{x}{\Delta x}$

$v = \frac{y}{\Delta y}$

x y

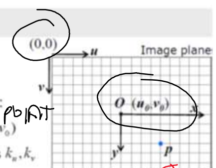
1. Convert P_c to image-plane coordinates ***(x,y)***

2. Convert P_c to (discretised) pixel coordinates ***(u,v)***

74

Perspective Projection (2)

From the Camera frame to pixel coordinates



- Convert ***p***, from the local image plane coords ***(x,y)*** to the pixel coords ***(u,v)*** we have to take account for:
  - Pixel coords of the camera ~~optical center~~ Principal point $O = (u_0, v_0)$
  - Scale factors for the pixel-size in both dimensions k_u, k_v

So:

$$u = u_0 + k_u x \Rightarrow u = u_0 + \frac{k_u f X_c}{Z_c}$$

$$v = v_0 + k_v y \Rightarrow v = v_0 + \frac{k_v f Y_c}{Z_c}$$

$u - u_0 = \frac{x}{\Delta x} = \frac{f}{\Delta x}\frac{X_c}{Z_c}$

$k_u = \frac{1}{\Delta x}$ $k_v = \frac{1}{\Delta y}$

- Use **Homogeneous Coordinates** for linear mapping from 3D to 2D, by introducing an extra element (scale):

$$p = \begin{pmatrix} u \\ v \end{pmatrix} \Rightarrow \tilde{p} = \begin{pmatrix} \lambda u \\ \lambda v \\ \lambda \end{pmatrix}$$ and similarly for the world coordinates. Note, usually $\lambda = 1$

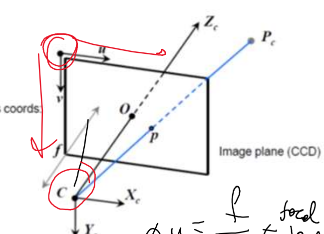
75

Perspective Projection (3)

- So:

$$u = u_0 + \frac{k_u f X_c}{Z_c}$$

$$v = v_0 + \frac{k_v f Y_c}{Z_c}$$



Expressed in matrix form & Homogenerous coords:

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Or alternatively

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Focal length in ***u***-direction

Focal length in ***v***-direction

"Calibration matrix"
or
"Matrix of Intrinsic Parameters"
internal

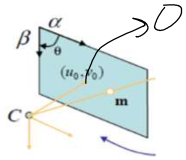
$\alpha u = \frac{f}{\Delta x}$ ← focal length in terms of pixel

DoF of $\underline{K} = 4$

76

$$K = \begin{bmatrix} \alpha & s & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$ → skew

K has 5 degrees of freedom!



→ O

$s = \alpha \cot \theta$ Axis skew causes shear distortion

77

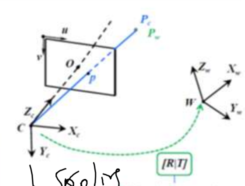
Perspective Projection (4)

From the Camera frame to the World frame

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

in homogeneous coords

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} R & T \\ \hline 0\ 0\ 0 & 1 \end{array}\right] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$



$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

↓ Scaling factor

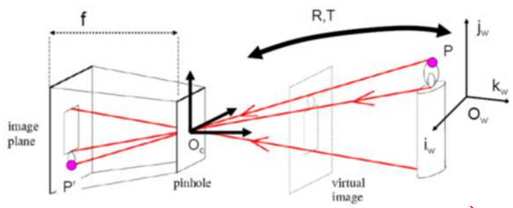
$[R\ T]$ external parameters

Projection Matrix

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boxed{K[R|T]} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$\underbrace{K[R|T]}_{M}$

78



$$P' = MP_w$$

Projection matrix

$$= \boxed{K}\boxed{\begin{bmatrix} R & T \end{bmatrix}}P_w$$

Internal parameters

External parameters

Dof of $\underline{M}$

$= 11$

79

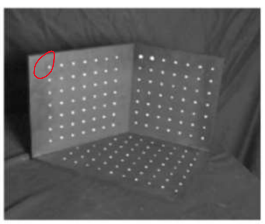
Calibration

- How to determine **M**

80

Calibration using a reference object

- Place a known object in the scene
  - identify correspondence between image and scene
  - compute mapping from scene to image



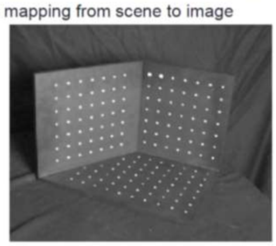
Issues

- must know geometry very accurately
- must know 3D->2D correspondence

81

Estimating the projection matrix

- Place a known object in the scene
  - identify correspondence between image and scene
  - compute mapping from scene to image



$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

82

- Projective Camera Matrix:

$$p = K[R \quad t]P = MP$$

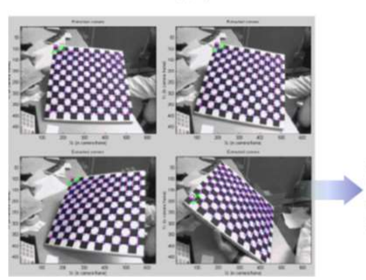
$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- Only up to a scale, so 11 DOFs.

83

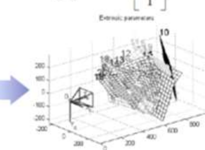
Camera Calibration

- Use our camera model to interpret the projection from world to image plane
- Using known correspondences of $p \Leftrightarrow P$, we can compute the unknown parameters K, R, T by applying the perspective projection equation
- … so associate known, physical distances in the world to pixel-distances in image



Projection Matrix

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boxed{K[R|T]} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$



84

Direct linear calibration - homogeneous

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$$\left.\begin{aligned} u_i &= \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}} \\ v_i &= \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}} \end{aligned}\right\}$$

$$u_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}$$
$$v_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}$$

P_i^T $\quad -u_i \{ P_i^T$

One pair of equations for each point

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i & -v_i \end{bmatrix} \begin{bmatrix} m_{00} \\ m_{01} \\ m_{02} \\ m_{03} \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{13} \\ m_{20} \\ m_{21} \\ m_{22} \\ m_{23} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

85

Direct linear calibration - homogeneous

$$\underbrace{\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix}}_{\mathbf{A}\ 2n \times 12} \underbrace{\begin{bmatrix} m_{00} \\ m_{10} \\ m_{02} \\ m_{03} \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{13} \\ m_{20} \\ m_{21} \\ m_{22} \\ m_{23} \end{bmatrix}}_{\mathbf{m}\ 12} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{0}\ 2n}$$

This is a homogenous set of equations.

$\underline{A}_{2n\times 12} \cdot \vec{m}_{12\times 1} = \vec{0}_{12\times 1}$ $\quad$ n # of corresp

86

Homogeneous M x N Linear Systems

M=number of equations
N=number of unknown

$$A \; x = 0$$

Rectangular system (M>N)

- 0 is always a solution
- To find non-zero solution

  Minimize $|Ax|^2$

  under the constraint $|x|^2 = 1$

87

Direct linear calibration - homogeneous

$$\underbrace{\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix}}_{\mathbf{A}\ 2n \times 12} \underbrace{\begin{bmatrix} m_{00} \\ m_{10} \\ m_{02} \\ m_{03} \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{13} \\ m_{20} \\ m_{21} \\ m_{22} \\ m_{23} \end{bmatrix}}_{\mathbf{m}\ 12} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{0}\ 2n\times 1}$$

rank(A^TA) = rank(A^T) = rank(A)

11

This is a homogenous set of equations.

When over constrained, defines a least squares problem

- minimize $\|\mathbf{Am}\|$
  - Since **m** is only defined up to scale, solve for unit vector $\mathbf{m}^*$
  - Solution: $\mathbf{m}^*$ = eigenvector of $\mathbf{A}^T\mathbf{A}$ with *smallest* eigenvalue
  - Works with 6 or more points

88

Camera Calibration

- We know that : $\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R|T] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$
- So there are 11 values to estimate: (the overall scale doesn't matter, so e.g. m_{34} could be set to 1) $\quad \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$
- Each observed point gives us a pair of equations:

$$u_i = \frac{\lambda u_i}{\lambda} = \frac{m_{11}X_i + m_{12}Y_i + m_{13}Z_i + m_{14}}{m_{31} + m_{32} + m_{33} + m_{34}}$$
$$v_i = \frac{\lambda v_i}{\lambda} = \frac{m_{21}X_i + m_{22}Y_i + m_{23}Z_i + m_{24}}{m_{31} + m_{32} + m_{33} + m_{34}}$$

- To estimate 11 unknowns, we need **at least 6** points to calibrate the camera ⇨ solved using linear least squares

89

Direct linear calibration - inhomogeneous

- Another approach: 1 in lower r.h. corner for 11 d.o.f

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- Now "regular" least squares since there is a non-variable term in the equations:

$$u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + 1}$$
$$v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + 1}$$

90

$$X_i q_{11} + Y_i q_{12} + Z_i q_{13} + q_{14} - u_i X_i q_{31} - u_i Y_i q_{32} - u_i Z_i q_{33} = u_i$$
$$X_i q_{21} + Y_i q_{22} + Z_i q_{23} + q_{24} - v_i X_i q_{31} - v_i Y_i q_{32} - v_i Z_i q_{33} = v_i$$

How many equations do we need to solve this?

How many points do we need to solve this?

$$\underbrace{\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i \end{bmatrix}}_{\text{2N x 11 Matrix}} \underbrace{\begin{bmatrix} q_{11} \\ q_{12} \\ q_{13} \\ q_{14} \\ q_{21} \\ q_{22} \\ q_{23} \\ q_{24} \\ q_{31} \\ q_{32} \\ q_{33} \end{bmatrix}}_{\text{11 vector of unknowns}} = \underbrace{\begin{bmatrix} u_i \\ v_i \end{bmatrix}}_{\text{2N vector}}$$

$\vec{m}$

$B \vec{m} = \vec{p}_i$

$\vec{m} = (B^T B)^{-1} B^T P_i$

91

Solving for q

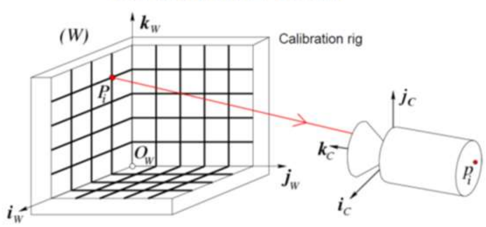
- Since every point gives two equations, we need at least 6 non-coplanar points to solve $A\mathbf{q} = \mathbf{b}$
- We can solve this using linear least squares

$$A\mathbf{q} = \mathbf{b}$$
$$\Rightarrow A^T A\mathbf{q} = A^T\mathbf{b}$$
$$\Rightarrow \mathbf{q} = (A^T A)^{-1} A^T\mathbf{b}$$

- Can be solved in one line of Matlab $\mathbf{q} = A\backslash\mathbf{b}$;
- For a unique solution, $A^T A$ must be non-singular i.e. $\mathrm{rank}(A^T A)$ or $\mathrm{rank}(A)$ must be 11. Need $2N \geq 11,\ N \geq 6$ non-coplanar points.

92

Calibration Problem

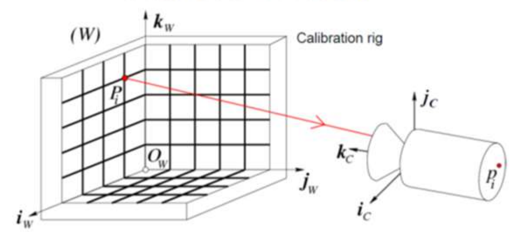


- $P_1 \ldots P_n$ with known positions in $[O_w, i_w, j_w, k_w]$
- $p_1, \ldots p_n$ known positions in the image

Goal: compute intrinsic and extrinsic parameters

93

Calibration Problem

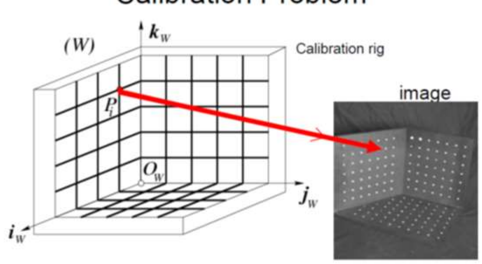


How many correspondences do we need?

- P has 11 unknown
- We need 11 equations
- 6 correspondences would do it

94

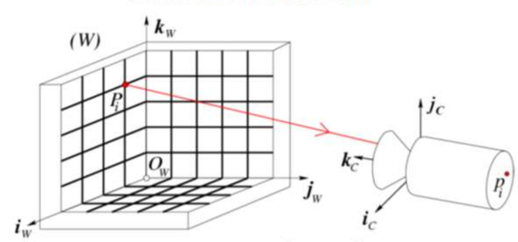
Calibration Problem



In practice: user may need to look at the image and select the n>=6 correspondences

95

Calibration Problem



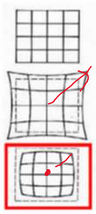
$$P_i \to M\,P_i \to p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix} \qquad M = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

in pixels

96

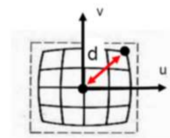
Radial Distortion

- Caused by imperfect lenses
- Deviations are most noticeable for rays that pass through the edge of the lens



97

Radial Distortion



$$\begin{bmatrix} \frac{1}{\lambda} & 0 & 0 \\ 0 & \frac{1}{\lambda} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathrm{M}\,\mathrm{P}_i \rightarrow \begin{bmatrix} u_i \\ v_i \end{bmatrix} = p_i$$

$$M = \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix}$$

$$d^2 = a\,u^2 + b\,v^2 + c\,u\,v$$

To model radial behavior

$$\lambda = 1 \pm \underbrace{\sum_{p=1}^{3} \boxed{\kappa_p} d^{2p}}_{\text{Polynomial function}}$$

Distortion coefficient

98

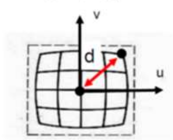
Radial Distortion

Estimating m_1 and m_2…

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix}$$

How to do that?

Hint:



$$\frac{u_i}{v_i} = \text{slope}$$

99

Radial Distortion

Estimating m_1 and m_2…

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \dfrac{\mathbf{m}_1^T P_i}{\mathbf{m}_3^T P_i} \\ \dfrac{\mathbf{m}_2^T P_i}{\mathbf{m}_3^T P_i} \end{bmatrix} \qquad \frac{u_i}{v_i} = \frac{\dfrac{(\mathbf{m}_1^T P_i)}{\cancel{(\mathbf{m}_3^T P_i)}}}{\dfrac{(\mathbf{m}_2\,P_i)}{\cancel{(\mathbf{m}_3\,P_i)}}} = \frac{\mathbf{m}_1\,P_i}{\mathbf{m}_2\,P_i}$$

$$\begin{cases} v_1(\mathbf{m}_1\,P_1) - u_1(\mathbf{m}_2\,P_1) = 0 \\ v_i(\mathbf{m}_1\,P_i) - u_i(\mathbf{m}_2\,P_i) = 0 \\ \vdots \\ v_n(\mathbf{m}_1\,P_n) - u_n(\mathbf{m}_2\,P_n) = 0 \end{cases} \qquad Q\,\mathbf{n} = 0 \qquad \mathbf{n} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}$$

Tsai technique [87]

100

Radial Distortion

Once that $\mathbf{m}_1$ and $\mathbf{m}_2$ are estimated…

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix}$$

$\mathbf{m}_3$ is non linear function of $\mathbf{m}_1$, $\mathbf{m}_2$, λ

101

General Calibration Problem

$$X = f(P)$$

f( ) is nonlinear

measurement — X; parameter — P

-Newton Method
-Levenberg-Marquardt Algorithm

- Iterative, starts from initial solution
- May be slow if initial solution far from real solution
- Estimated solution may be function of the initial solution
- Newton requires the computation of J, H
- Levenberg-Marquardt doesn't require the computation of H

102