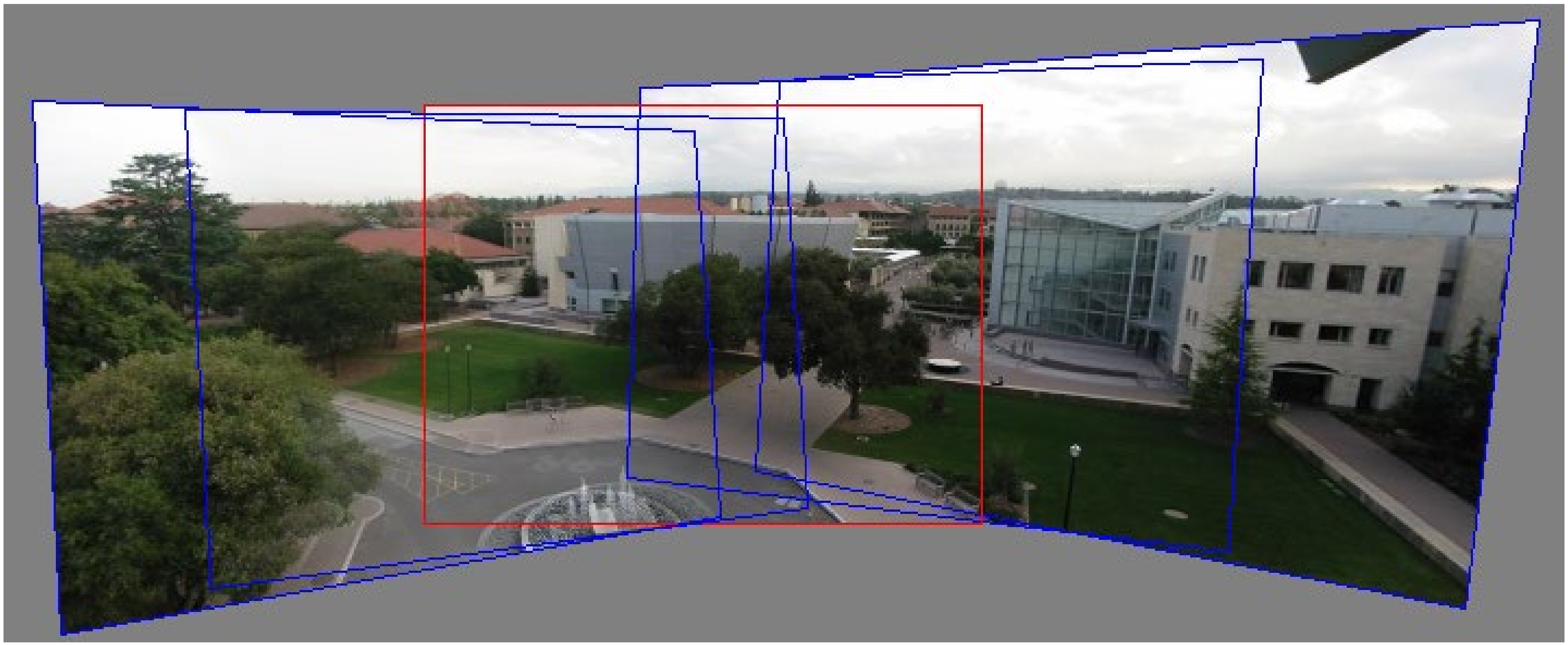
Cameras

<Vision System>

Department of Robot Engineering
Prof. Younggun Cho

Motivation: Can we use homographies to create a 360 panorama?

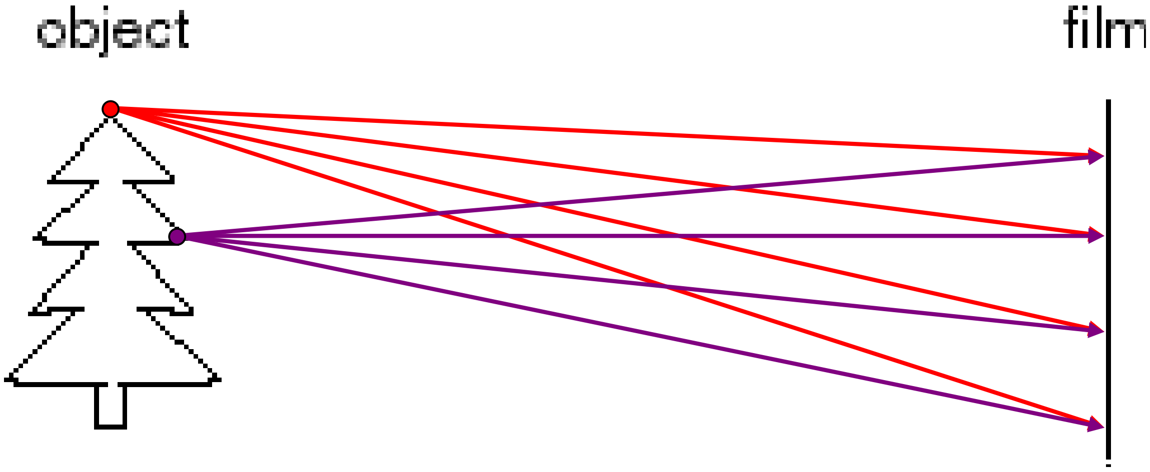


- In order to figure this out, we need to learn what a **camera** is

Overview

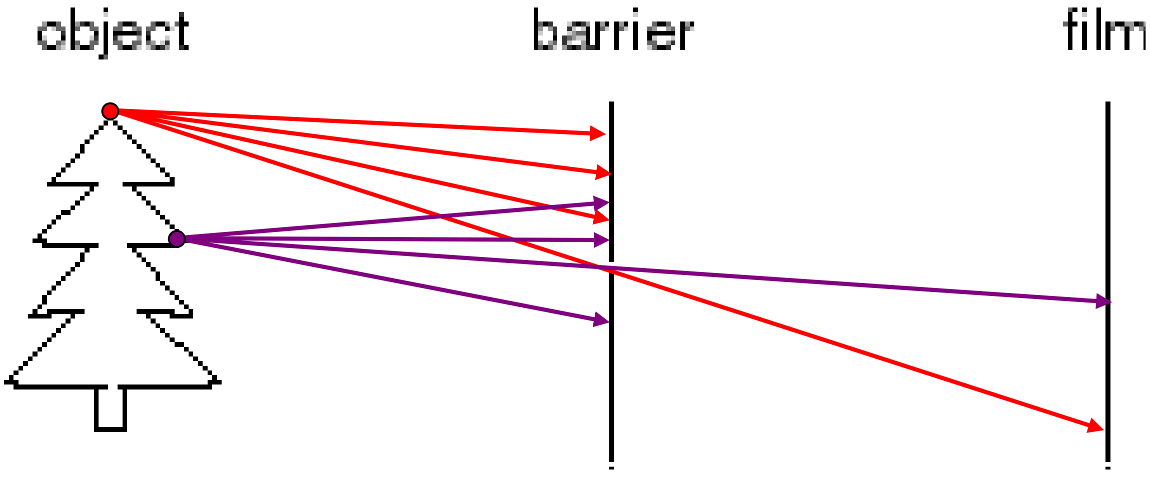
- The pinhole projection model
  - Qualitative properties
  - Perspective projection matrix
- Cameras with lenses
  - Depth of focus
  - Field of view
  - Lens aberrations
- Digital cameras
  - Types of sensors
  - Color

Image formation



- Let's design a camera
  - Idea 1: put a piece of film in front of an object
  - Do we get a reasonable image?
  - No. This is a bad camera.

Pinhole camera



- Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening known as the **aperture**
  - How does this transform the image?

Pinhole camera

Camera Obscura (어두운 방)

- Basic principle known to Mozi (470-390 BC), Aristotle (384-322 BC)
- Drawing aid for artists: described by Leonardo da Vinci (1452-1519)

Source: A. Efros

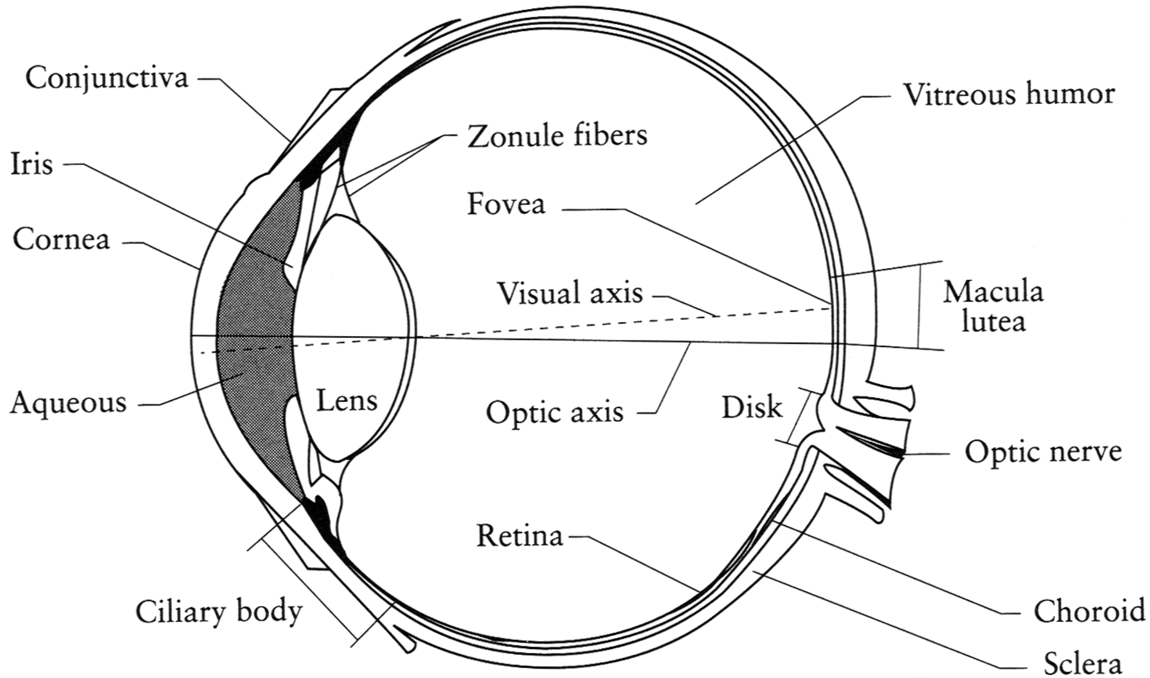
Home-made pinhole camera

Why so blurry?

http://www.debevec.org/Pinhole/

Slide by A. Efros

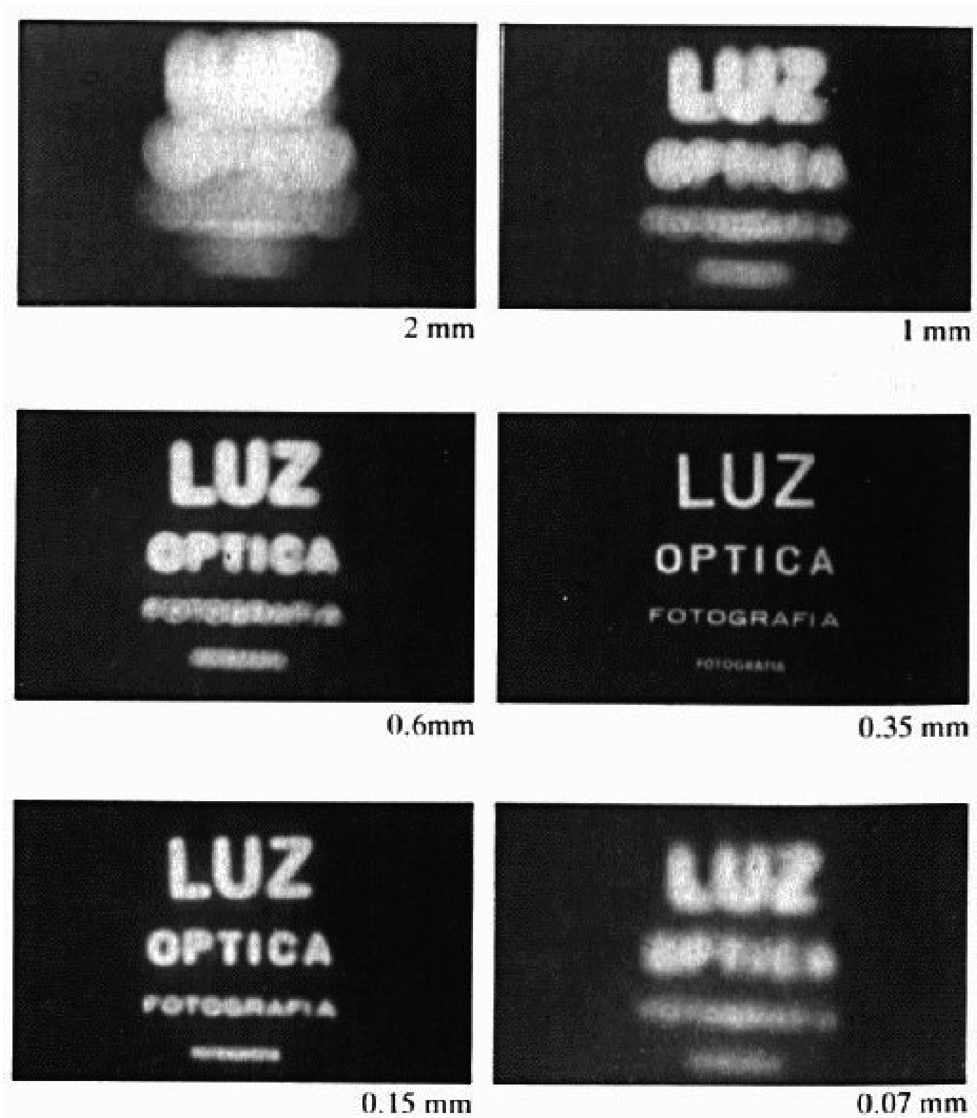
The eye



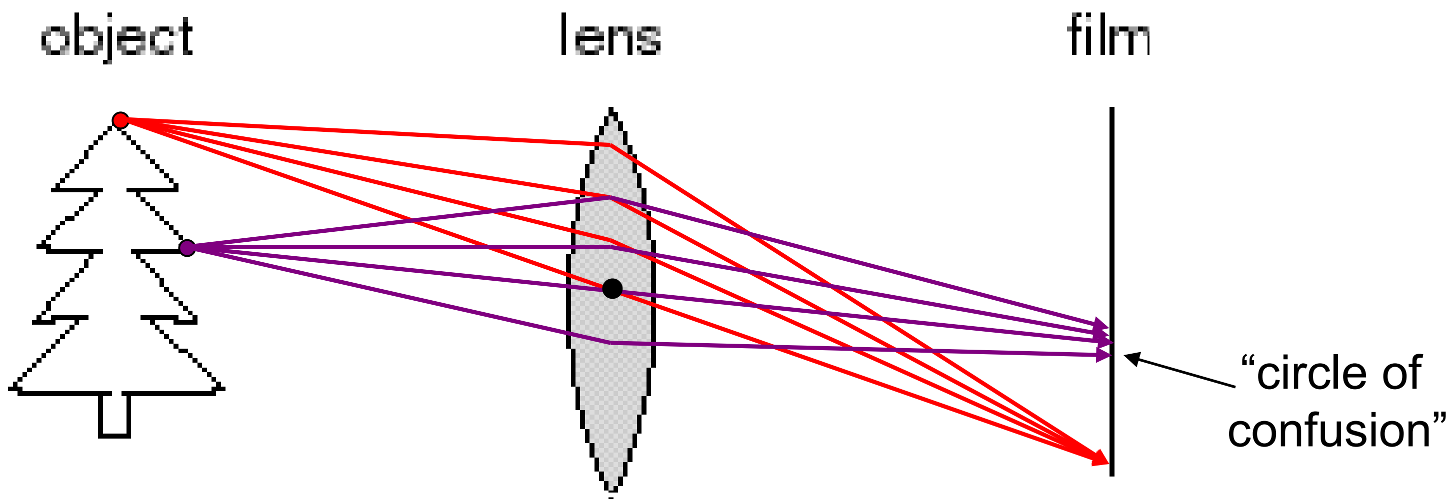
- The human eye is a camera
  - **Iris** - colored annulus with radial muscles
  - **Pupil** - the hole (aperture) whose size is controlled by the iris
  - What's the "film"?
    - photoreceptor cells (rods and cones) in the **retina**

http://www.telegraph.co.uk/news/earth/earthpicturegalleries/7598120/Animal-eyes-quiz-Can-you-work-out-which-creatures-these-are-from-their-eyes.html?image=25

Shrinking the aperture



Adding a lens



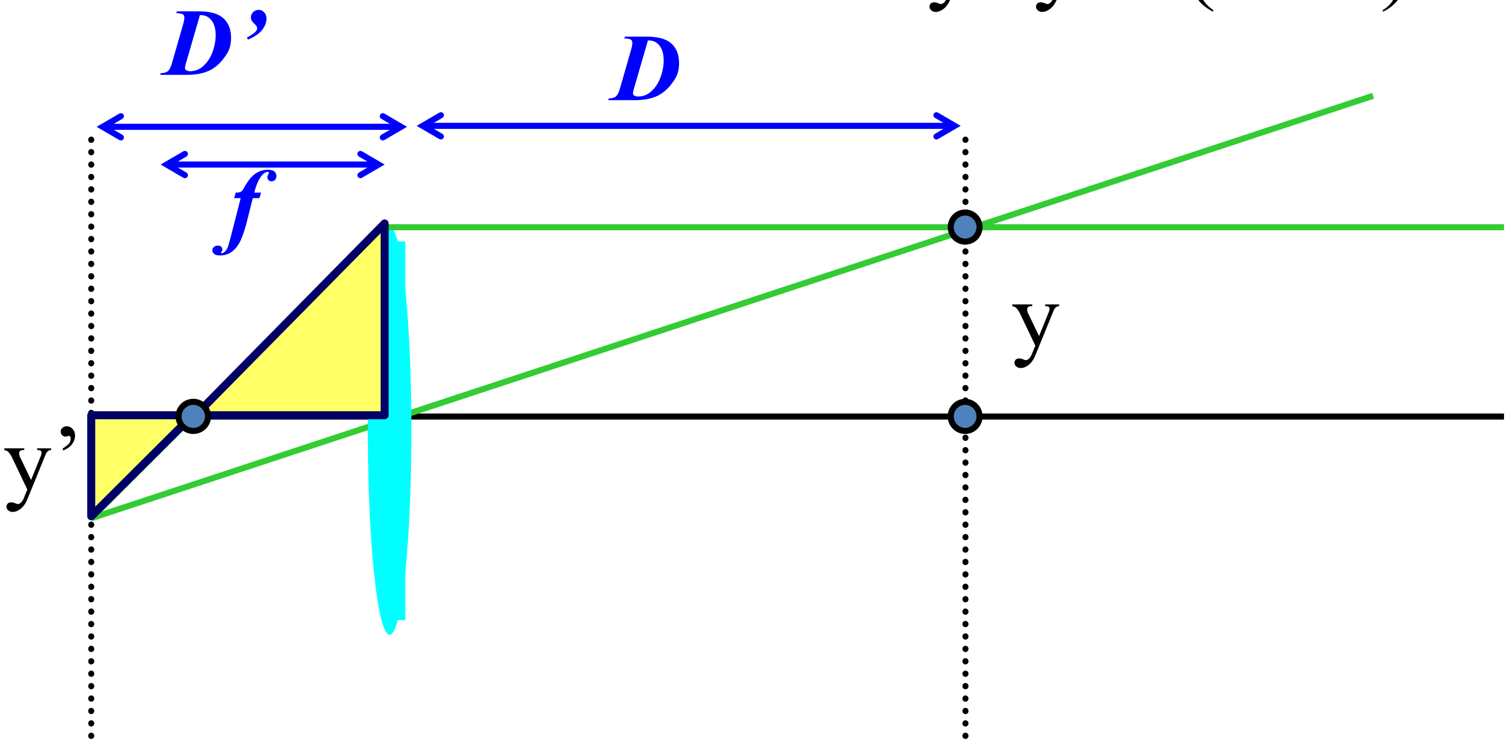
- A lens focuses light onto the film
  - There is a specific distance at which objects are “in focus”
    - other points project to a “circle of confusion” in the image
  - Changing the shape of the lens changes this distance

Thin lens formula

Similar triangles everywhere!

$$y'/y = D'/D$$

$$y'/y = (D'-f)/f$$

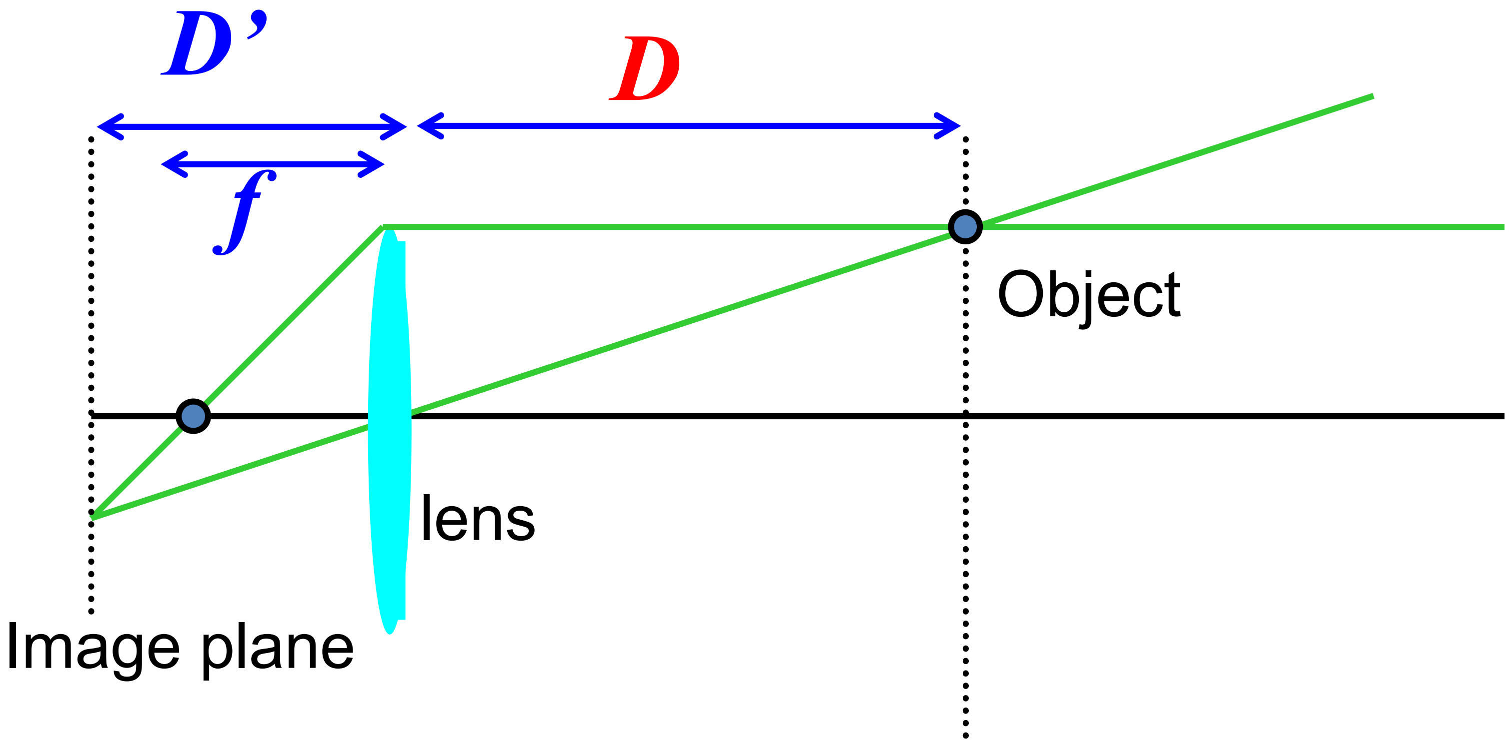


Frédo Durand's slide

Thin lens formula (포커싱 원리)

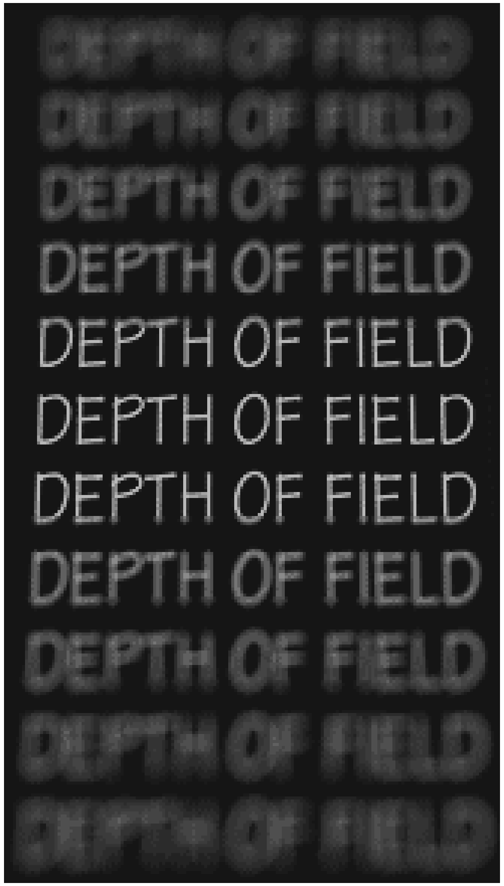
$$\frac{1}{D'} + \frac{1}{D} = \frac{1}{f}$$

Any point satisfying the thin lens equation is in focus.



Frédo Durand's slide

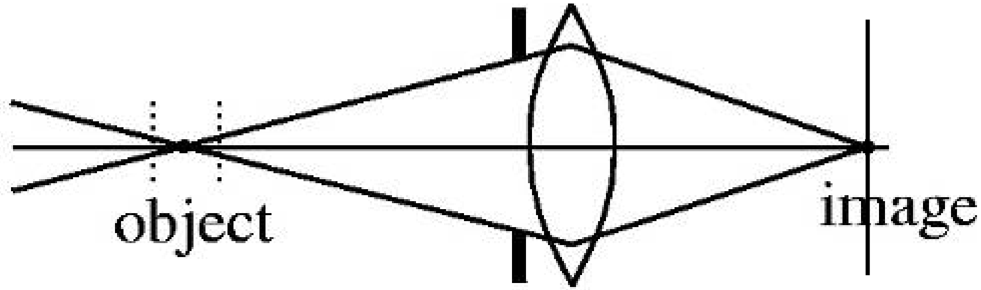
Depth of Field

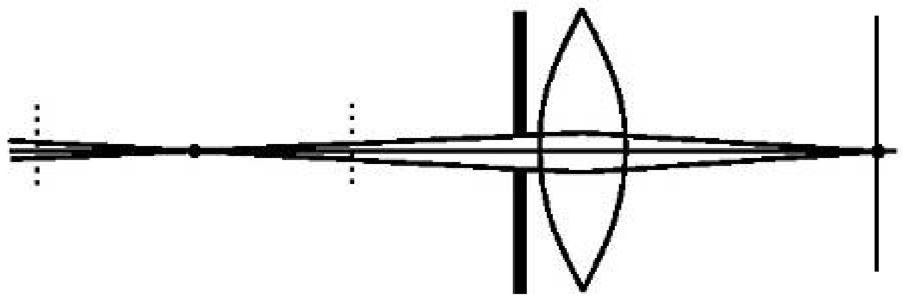


http://www.cambridgeincolour.com/tutorials/depth-of-field.htm

Slide by A. Efros

How can we control the depth of field?





중요

- Changing the aperture size affects depth of field
  - A smaller aperture increases the range in which the object is approximately in focus
  - But small aperture reduces amount of light – need to increase exposure

Slide by A. Efros

Varying the aperture

$$f/\# = N = \frac{f}{D}$$

여기서 f는 초점거리이고 D는 입사동공 지름.

Large aperture = small F#

Small aperture = large F#

Camera Model

The camera as a coordinate transformation

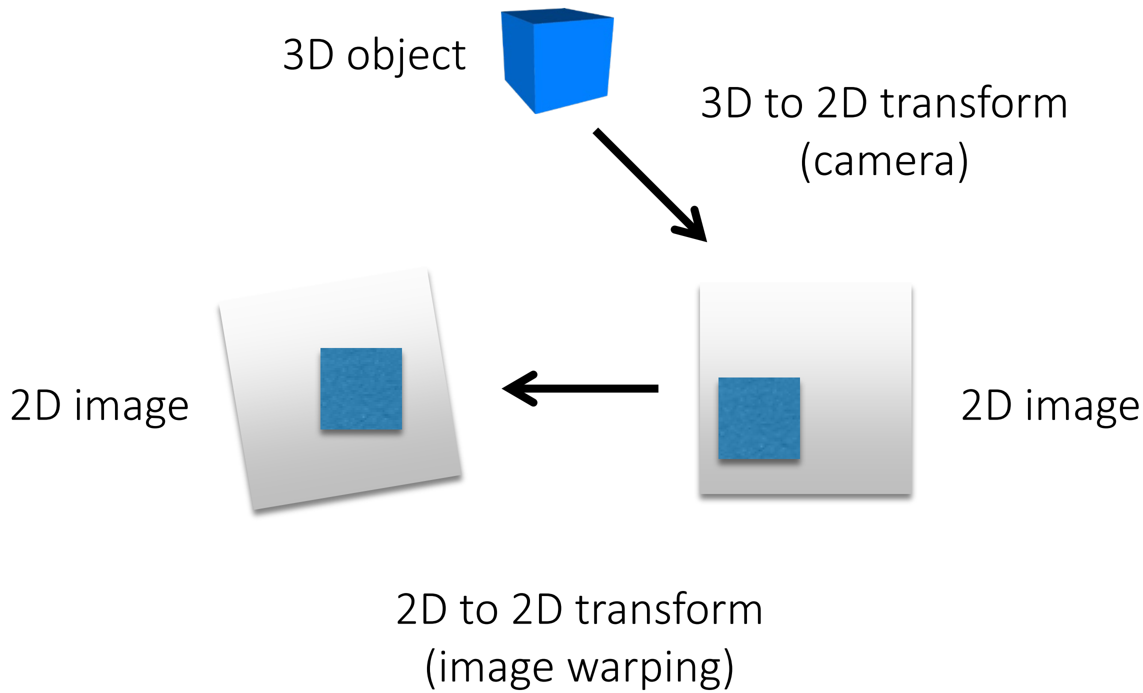
A camera is a mapping

from:

the 3D world

to:

a 2D image



The camera as a coordinate transformation

A camera is a mapping

from:

the 3D world

to:

a 2D image

homogeneous coordinates

$$\boldsymbol{x} = \mathbf{P}\mathbf{X}$$

2D image point

camera matrix

3D world point

What are the dimensions of each variable?

The camera as a coordinate transformation

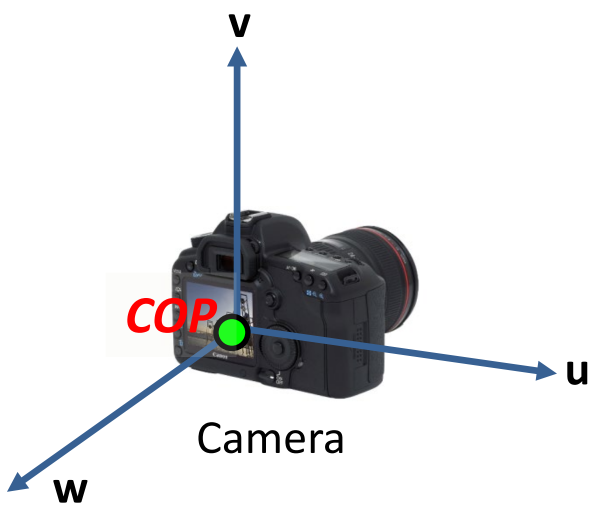
$$\boldsymbol{x} = \mathbf{PX}$$

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

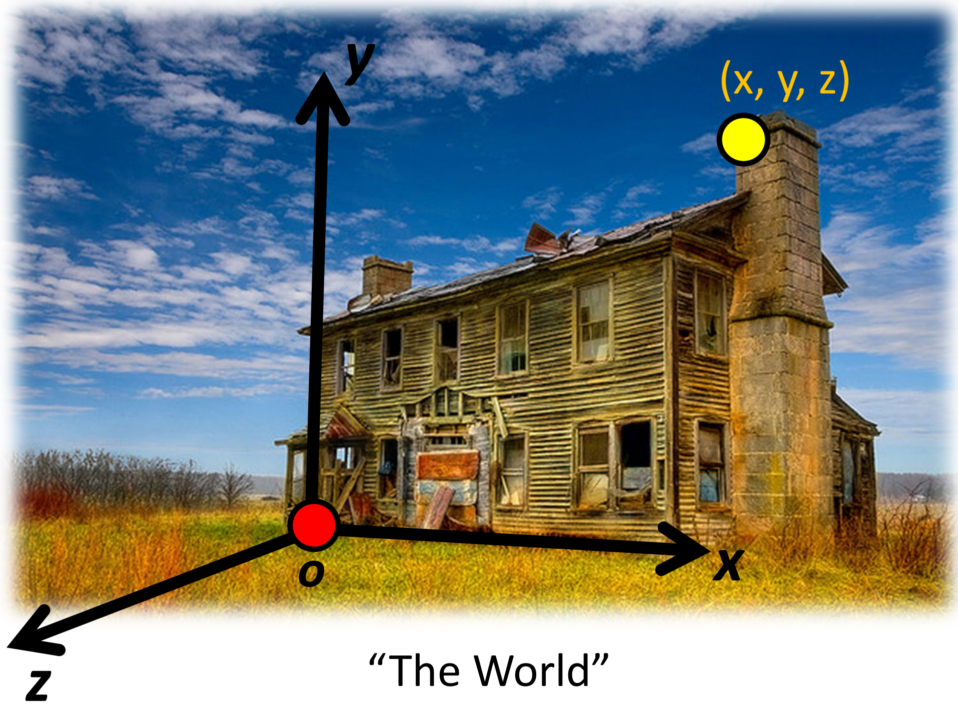
homogeneous image coordinates	camera matrix	homogeneous world coordinates
3 x 1	3 x 4	4 x 1

Camera parameters

- How can we model the geometry of a camera?



Camera



"The World"

Two important coordinate systems:

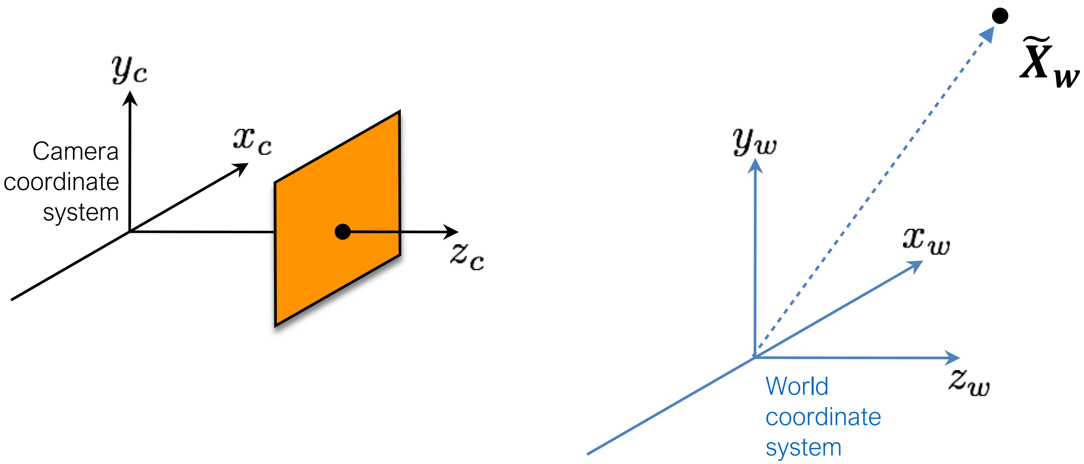
1. *World* coordinate system
2. *Camera* coordinate system

How do we project a given point (x, y, z) in world coordinates?

Camera parameters

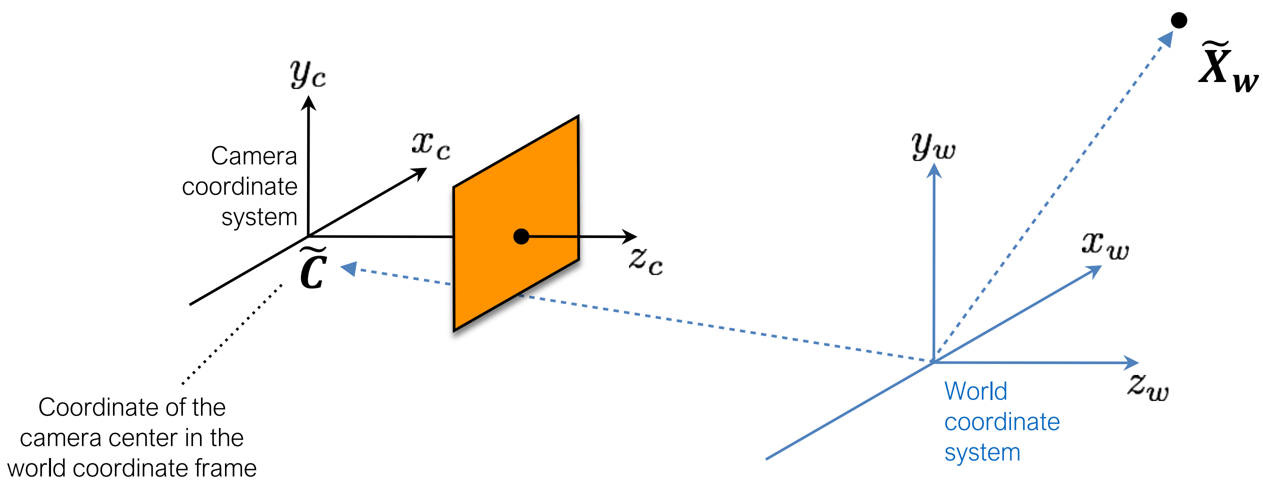
- To project a point (x,y,z) in *world* coordinates into a camera
- First transform (x,y,z) into *camera* coordinates
- Need to know
  - Camera position (in world coordinates)
  - Camera orientation (in world coordinates)
- Then project into the image plane to get a pixel coordinate
  - Need to know camera *intrinsics*

World-to-camera coordinate system transformation

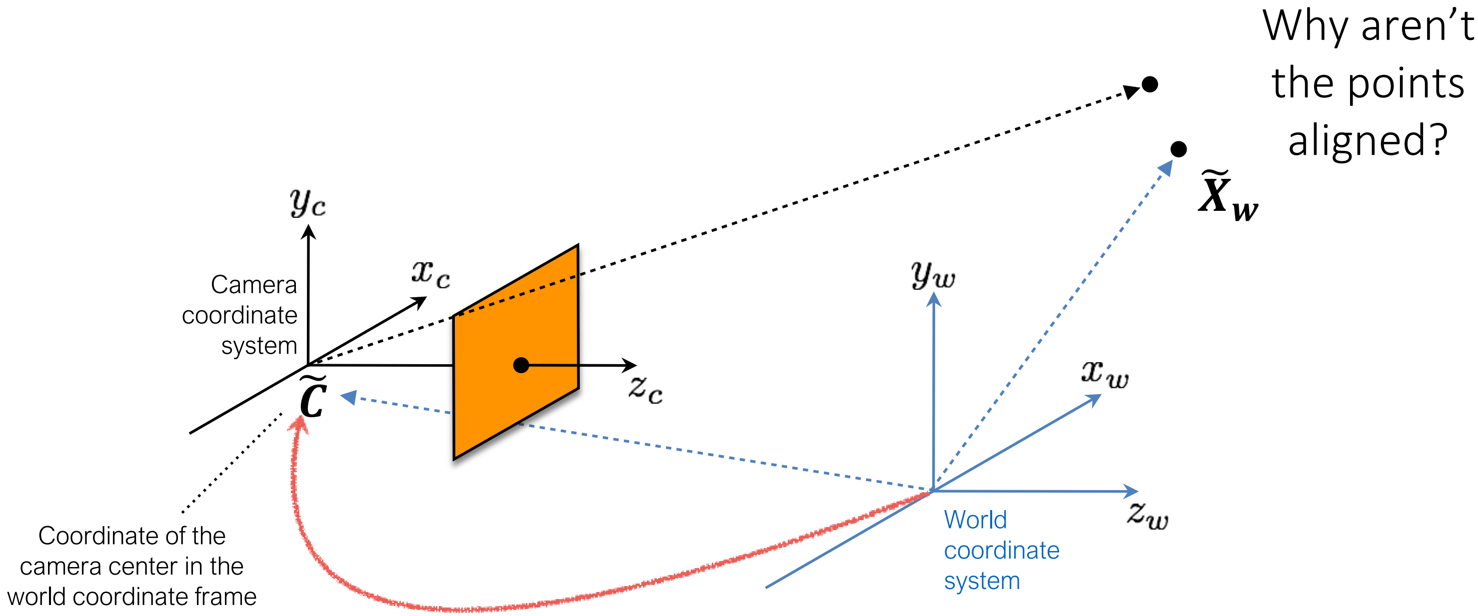


tilde means *heterogeneous* coordinates

World-to-camera coordinate system transformation



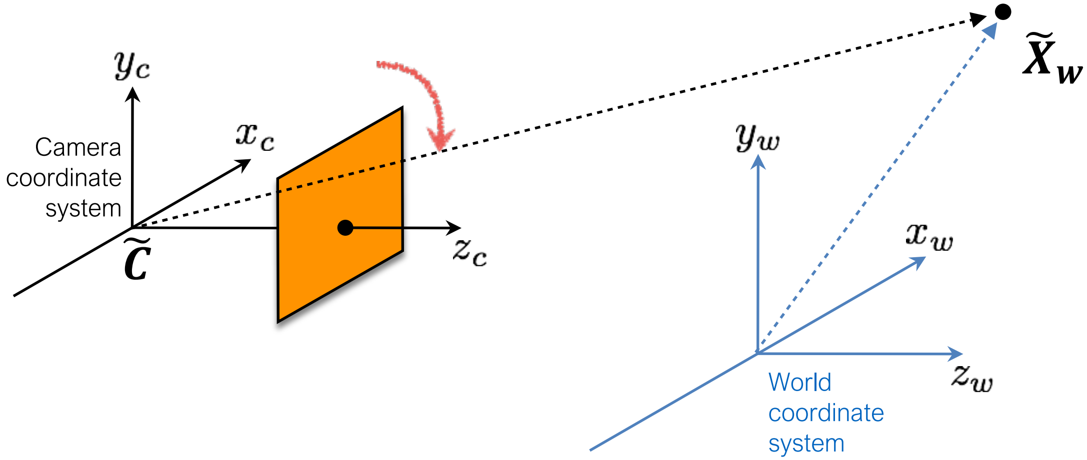
World-to-camera coordinate system transformation



$$(\widetilde{X}_w - \widetilde{C})$$

translate

World-to-camera coordinate system transformation



points now coincide

$$\boldsymbol{R} \cdot (\tilde{\boldsymbol{X}}_{\boldsymbol{w}} - \tilde{\boldsymbol{C}})$$

rotate translate

Modeling the coordinate system transformation

In heterogeneous coordinates, we have:

$$\widetilde{\mathbf{X}}_{\mathbf{c}} = \mathbf{R} \cdot \left(\widetilde{\mathbf{X}}_{\mathbf{w}} - \tilde{\mathbf{C}}\right)$$

In homogeneous coordinates, we have:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad \text{or} \quad \mathbf{X_c} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{X_w}$$

Putting it all together

We can write everything into a single projection:

$$\mathbf{x} = \mathbf{P}\mathbf{X_w}$$

The camera matrix now looks like:

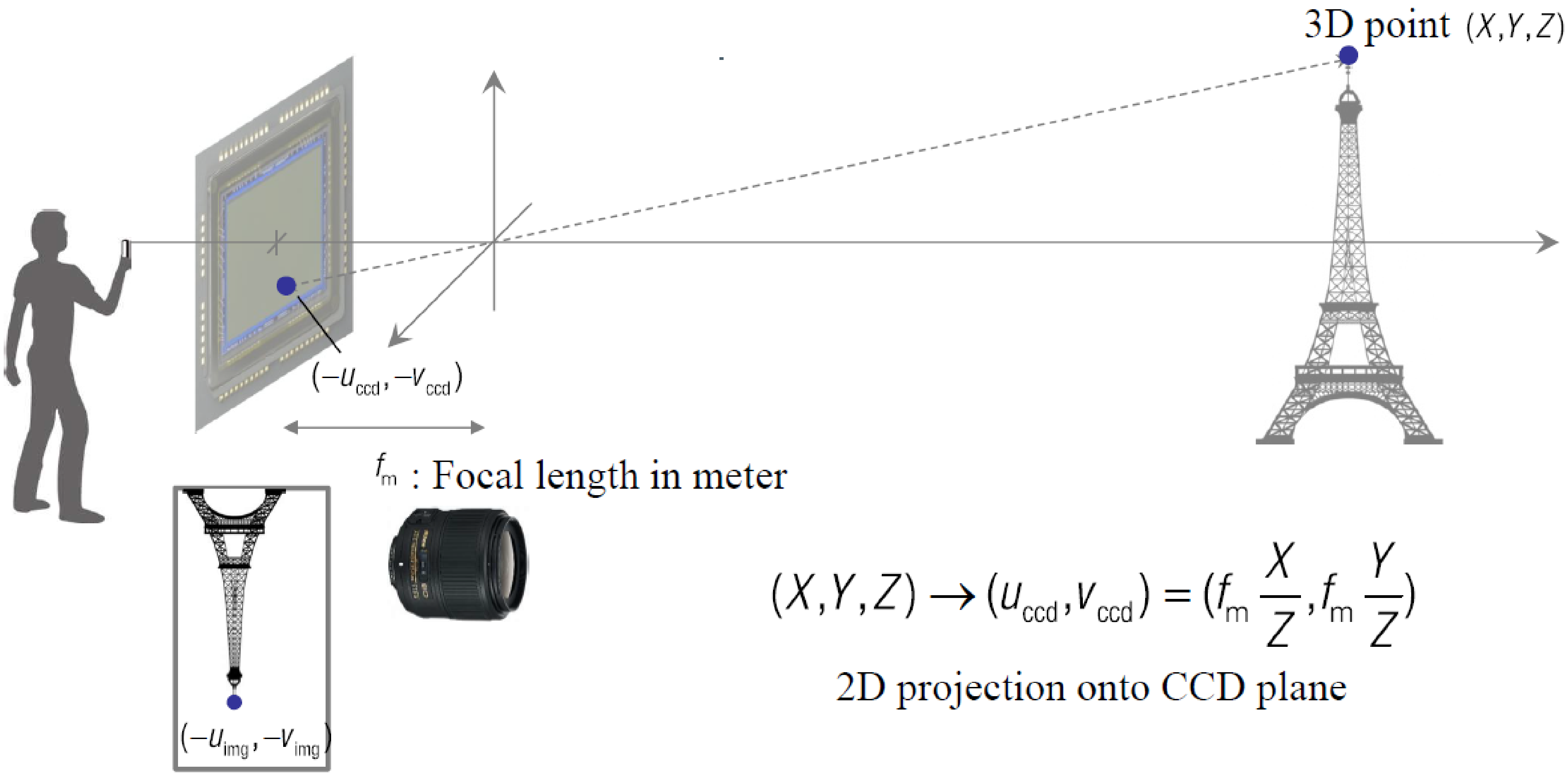
$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{I} \quad | \quad \mathbf{0}] \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix}$$

intrinsic parameters (3 x 3): correspond to camera internals (image-to-image transformation)

perspective projection (3 x 4): maps 3D to 2D points (camera-to-image transformation)

extrinsic parameters (4 x 4): correspond to camera externals (world-to-camera transformation)

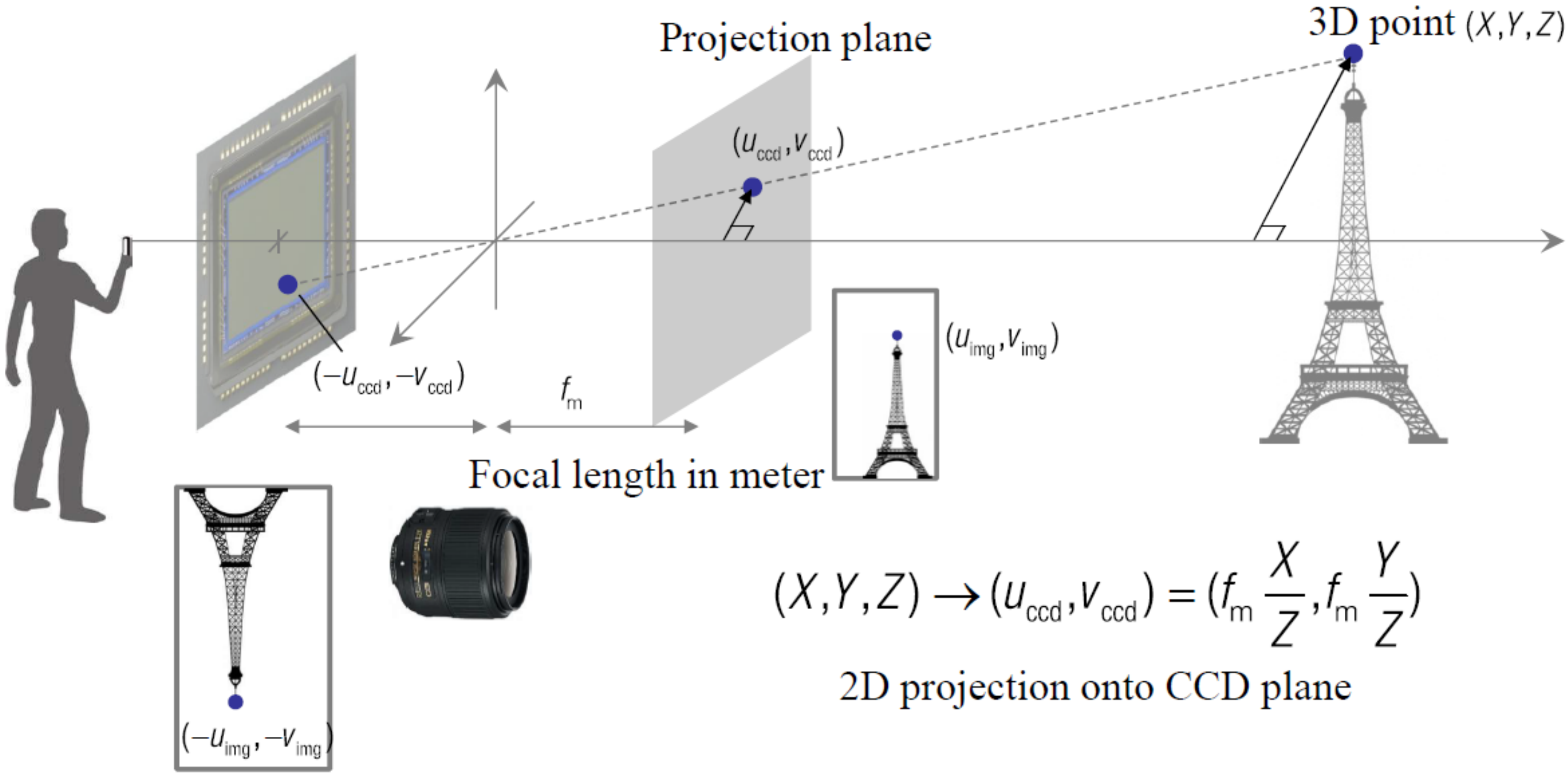
3D Point Projection (Metric Space)



$$(X,Y,Z) \rightarrow (u_{ccd}, v_{ccd}) = (f_m \frac{X}{Z}, f_m \frac{Y}{Z})$$

2D projection onto CCD plane

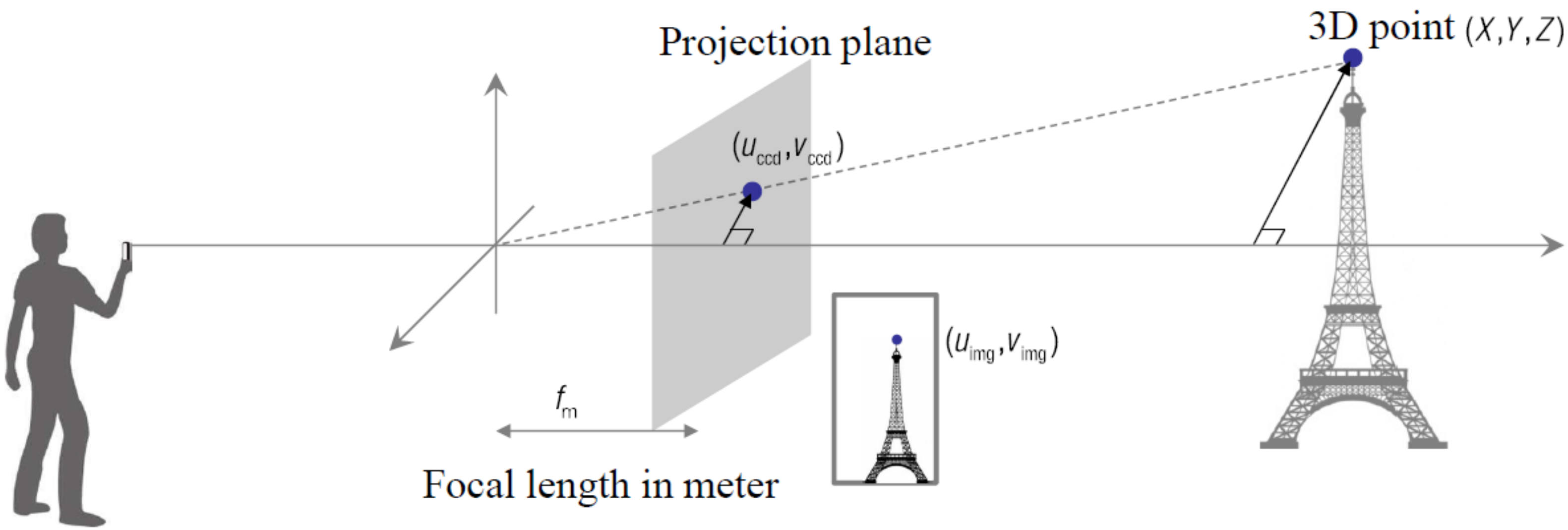
3D Point Projection (Metric Space)



$$(X,Y,Z) \rightarrow (u_{ccd}, v_{ccd}) = (f_m \frac{X}{Z}, f_m \frac{Y}{Z})$$

2D projection onto CCD plane

3D Point Projection (Metric Space)

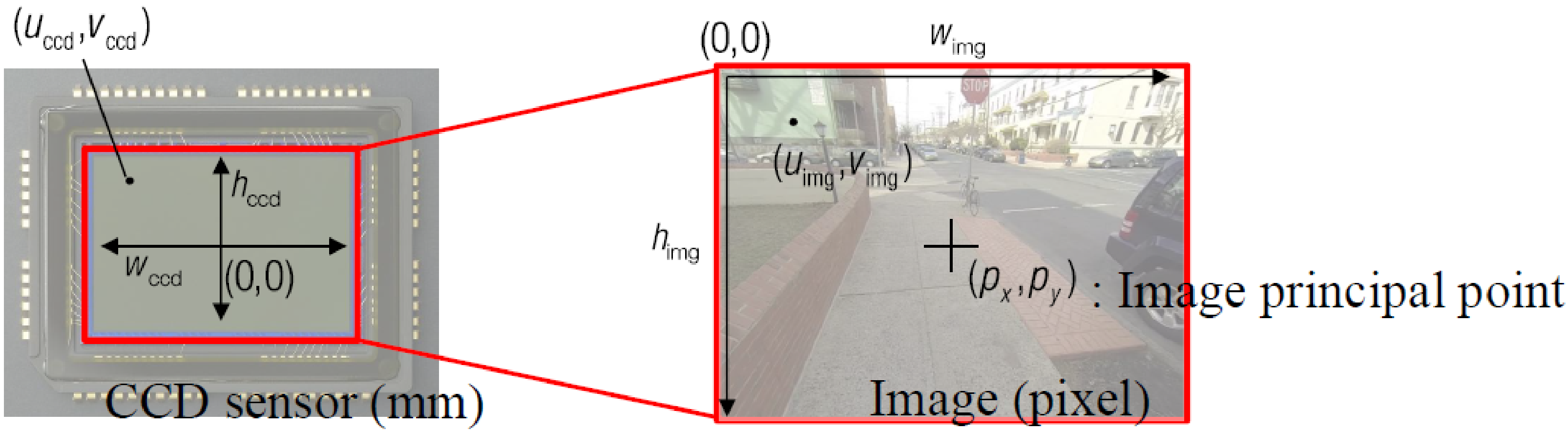


$$(X, Y, Z) \rightarrow (u_{ccd}, v_{ccd}) = (f_m \frac{X}{Z}, f_m \frac{Y}{Z})$$

2D projection onto CCD plane

Slide from Jianbo Shi

3D Point Projection (Pixel Space)

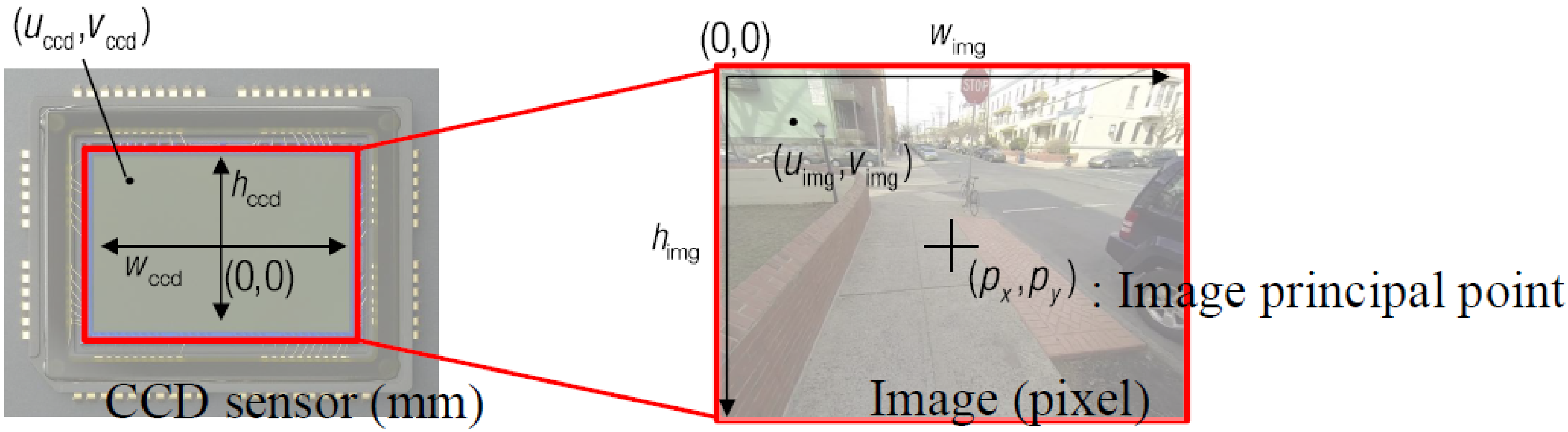


$$\frac{u_{ccd}}{w_{ccd}} = \frac{u_{img} - p_x}{w_{img}} \qquad \frac{v_{ccd}}{h_{ccd}} = \frac{v_{img} - p_y}{h_{img}}$$

$$u_{img} = u_{ccd}\frac{w_{img}}{w_{ccd}} + p_x \qquad v_{img} = v_{ccd}\frac{h_{img}}{h_{ccd}} + p_y$$

Slide from Jianbo Shi

3D Point Projection (Pixel Space)

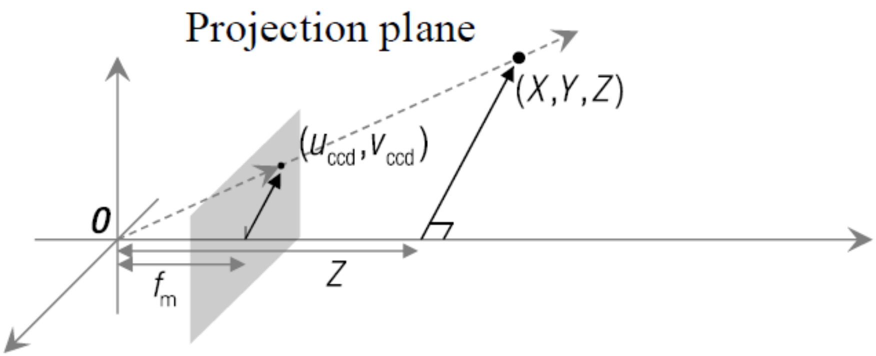


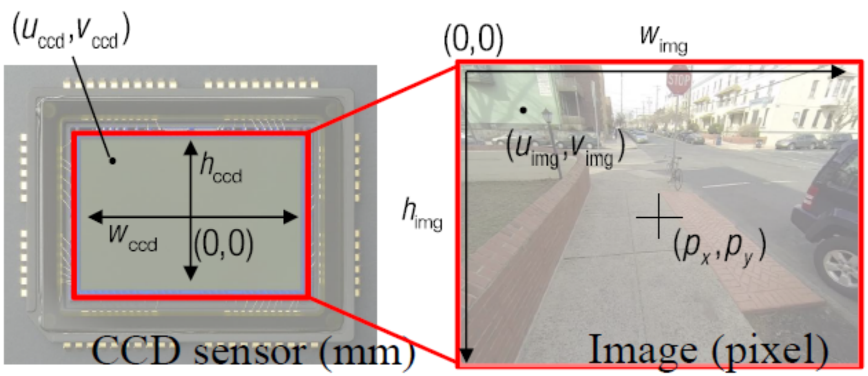
$$\frac{u_{ccd}}{w_{ccd}} = \frac{u_{img} - p_x}{w_{img}} \qquad \frac{v_{ccd}}{h_{ccd}} = \frac{v_{img} - p_y}{h_{img}}$$

$$u_{img} = u_{ccd}\frac{w_{img}}{w_{ccd}} + p_x \qquad v_{img} = v_{ccd}\frac{h_{img}}{h_{ccd}} + p_y$$

Slide from Jianbo Shi

3D Point Projection (Pixel Space)



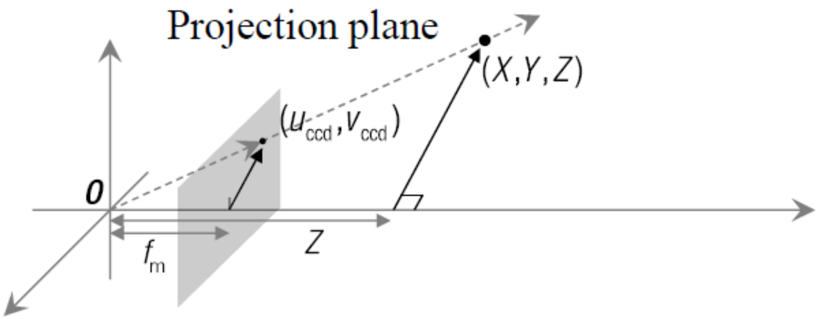


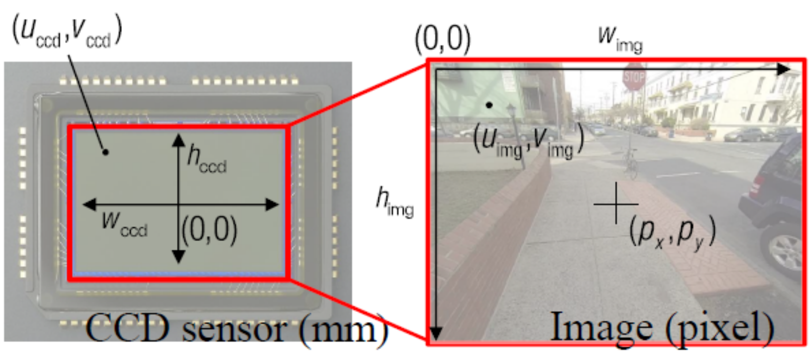
$$(X,Y,Z) \rightarrow (u_{ccd}, v_{ccd}) = (f_m \frac{X}{Z}, f_m \frac{Y}{Z})$$

$$u_{img} = u_{ccd} \frac{w_{img}}{w_{ccd}} + p_x = \underbrace{f_m \frac{w_{img}}{w_{ccd}}}_{\text{Focal length in pixel}} \frac{X}{Z} + p_x$$

$$v_{img} = v_{ccd} \frac{h_{img}}{h_{ccd}} + p_y = \underbrace{f_m \frac{h_{img}}{h_{ccd}}}_{\text{Focal length in pixel}} \frac{Y}{Z} + p_y$$

3D Point Projection (Pixel Space)





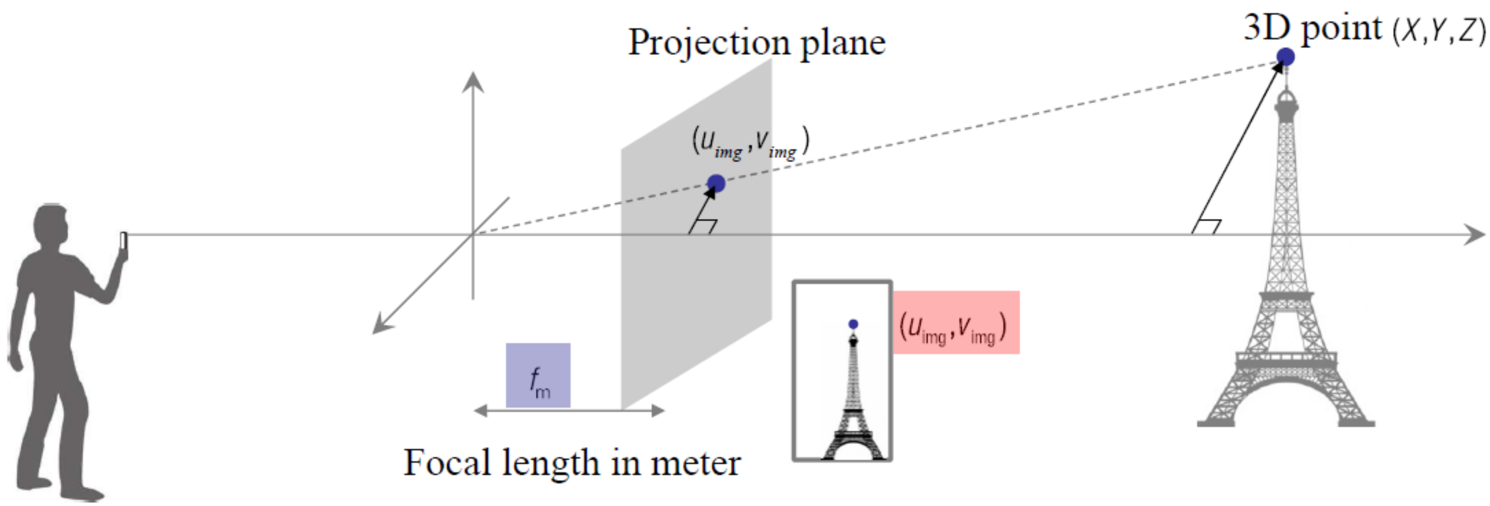
$$(X,Y,Z) \rightarrow (u_{ccd}, v_{ccd}) = (f_m \frac{X}{Z}, f_m \frac{Y}{Z})$$

$$u_{img} = u_{ccd} \frac{W_{img}}{W_{ccd}} + p_x = \underbrace{f_x}_{\text{Focal length in pixel}} \frac{X}{Z} + p_x$$

$$v_{img} = v_{ccd} \frac{h_{img}}{h_{ccd}} + p_y = \underbrace{f_y}_{\text{Focal length in pixel}} \frac{Y}{Z} + p_y$$

Slide from Jianbo Shi

3D Point Projection (Pixel Space)

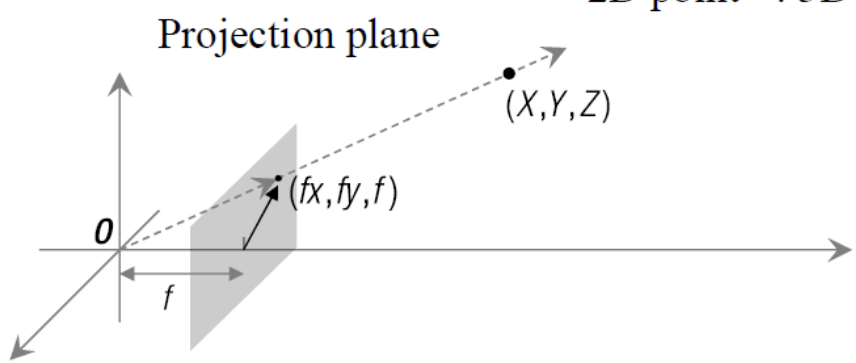


$$(X,Y,Z) \rightarrow (u_{img}, v_{img}) = (f_m \frac{w_{img}}{w_{ccd}} \frac{X}{Z}, f_m \frac{h_{img}}{h_{ccd}} \frac{Y}{Z})$$

Slide from Jianbo Shi

Homogeneous coordinates

2D point =: 3D ray $\lambda(x, y, 1)$



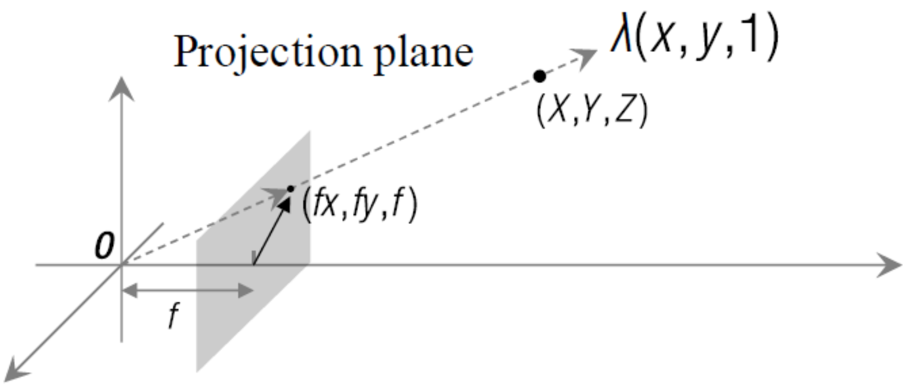
$$
\begin{aligned}
(x, y) &\rightarrow (x, y, 1) \\
&= f(x, y, 1) \\
&= \lambda(x, y, 1)
\end{aligned}
$$

: A point in Euclidean space ($\mathbb{R}^2$) can be represented by a homogeneous representation in Projective space ($\mathcal{P}^2$) (3 numbers).

Slide from Jianbo Shi

Homogeneous coordinates

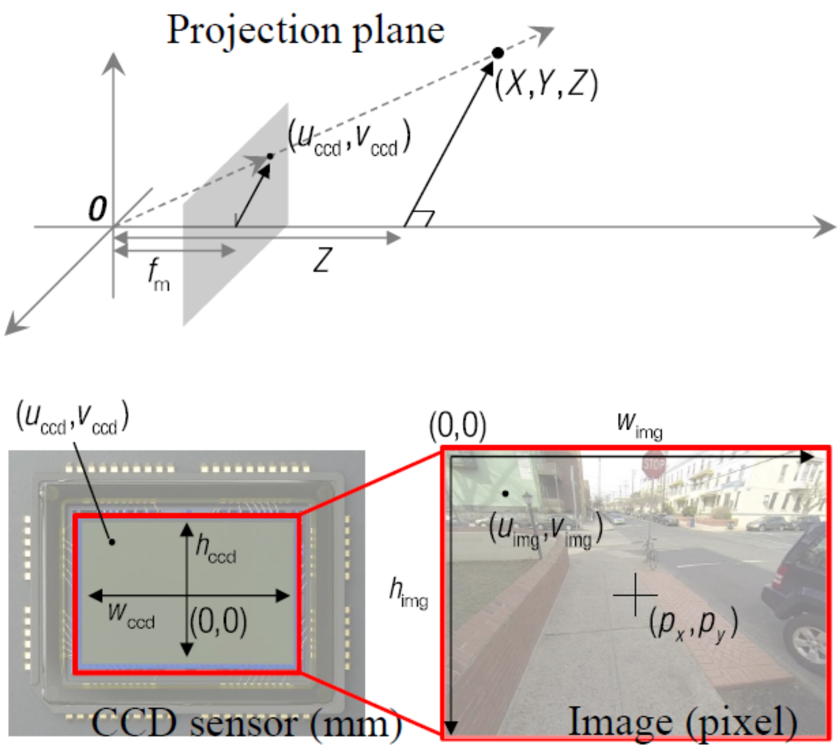
2D point =: 3D ray



$\underbrace{\lambda(x, y, 1)}_{\text{Homogeneous coordinate}} = (X, Y, Z)$: 3D point lies in the 3D ray passing 2D image point.

Slide from Jianbo Shi

3D point projection (Pixel space)



$$(X,Y,Z) \rightarrow (u_{\text{ccd}}, v_{\text{ccd}}) = (f_{\text{m}}\frac{X}{Z}, f_{\text{m}}\frac{Y}{Z})$$

$$u_{\text{img}} = f_x \frac{X}{Z} + p_x \qquad v_{\text{img}} = f_y \frac{Y}{Z} + p_y$$

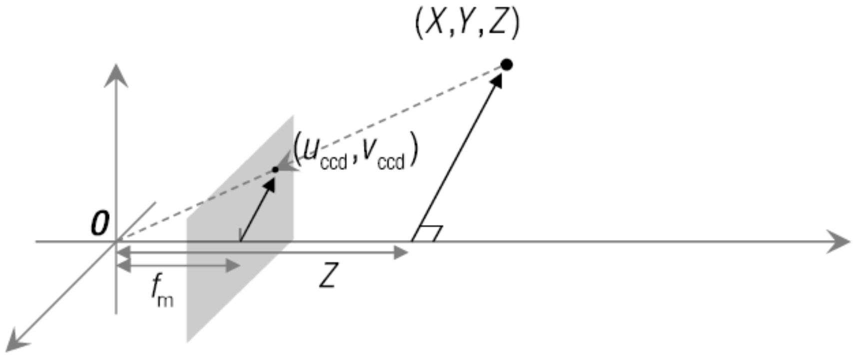
$$\lambda \begin{bmatrix} u_{\text{img}} \\ v_{\text{img}} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Homogeneous representation

Slide from Jianbo Shi

Camera Intrinsic!

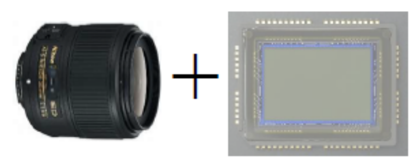
- Relationship Between 3D World to 2D Image!



Projection plane

Pixel space　　　　Metric space

$$\lambda \begin{bmatrix} u_{img} \\ v_{img} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$



Camera intrinsic parameter
: metric space to pixel space

Slide from Jianbo Shi

Real Camera

Focal length

- Can think of as "zoom"

24mm

50mm

200mm

800mm

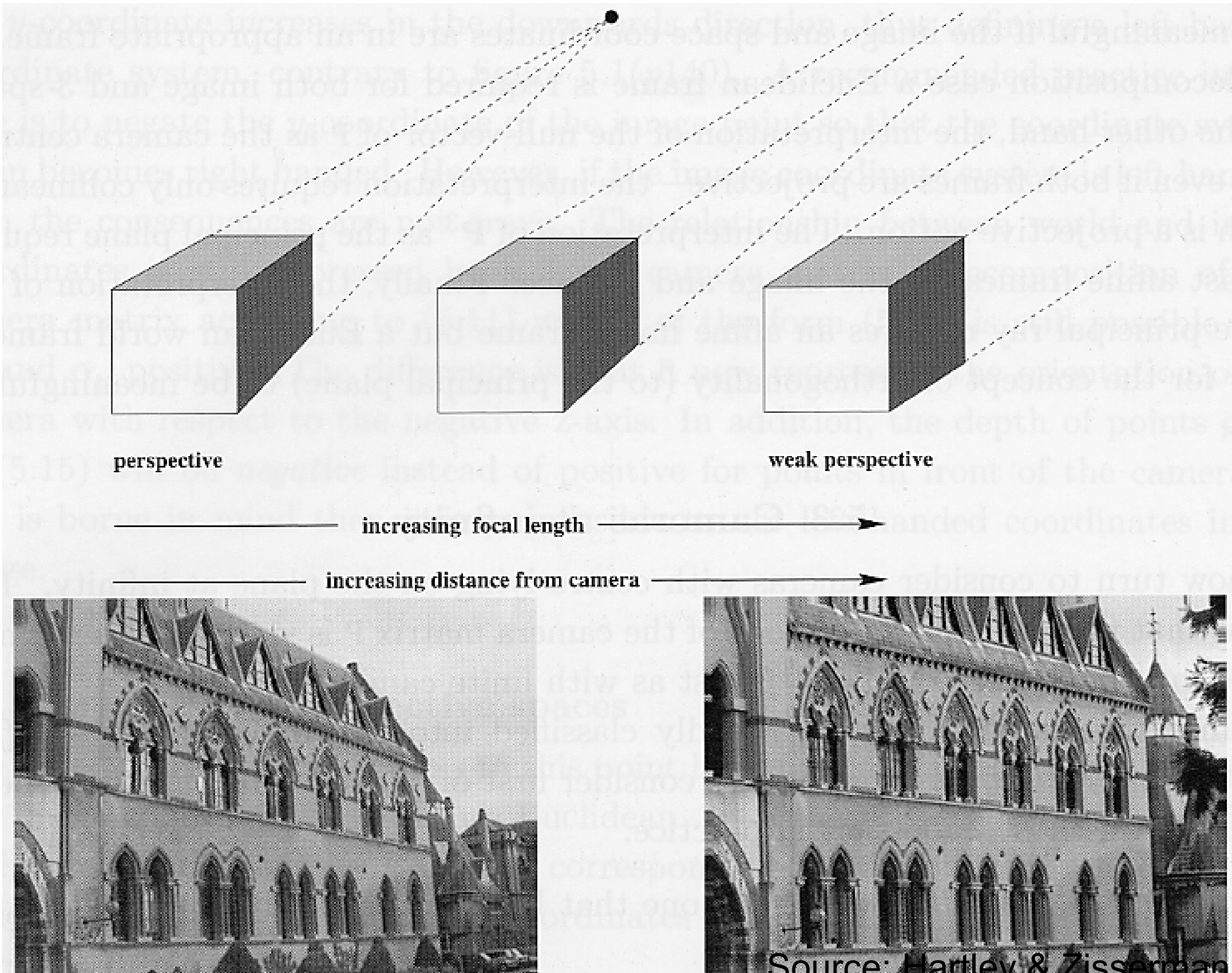
- Also related to *field of view*

Wide angle

Standard

Telephoto

Approximating an affine camera



Source: Hartley & Zisserman

Projection matrix

$$\mathbf{\Pi} = \underbrace{\mathbf{K}}_{\text{intrinsics}} \underbrace{\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} & \mathbf{R} & & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} & \mathbf{I}_{3\times 3} & & -\mathbf{c} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{translation}}}$$

The **K** matrix converts 3D rays in the camera's coordinate system to 2D image points in image (pixel) coordinates.

This part converts 3D points in world coordinates to 3D rays in the camera's coordinate system. There are 6 parameters represented (3 for position/translation, 3 for rotation).

Projection matrix

$$\Pi = \underbrace{\mathbf{K}}_{\text{intrinsics}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} & \mathbf{R} & & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} & \mathbf{I}_{3\times 3} & & -\mathbf{c} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{translation}}$$

$$\left[\; \mathbf{R} \;\middle|\; \underbrace{-\mathbf{Rc}}_{\text{(}\mathbf{t}\text{ in book's notation)}} \; \right]$$

$$\Pi = \mathbf{K} \left[\; \mathbf{R} \;\middle|\; -\mathbf{Rc} \; \right]$$

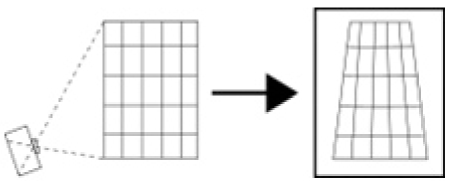
Perspective distortion

- Problem for architectural photography: converging verticals

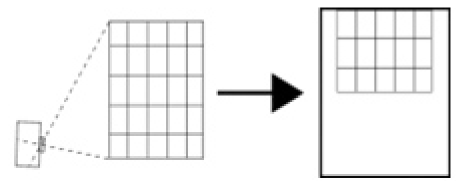
Source: F. Durand

Perspective distortion

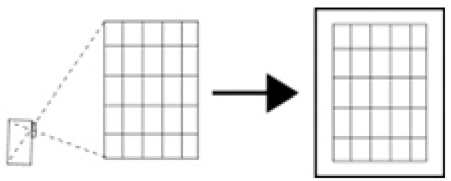
- Problem for architectural photography: converging verticals



Tilting the camera upwards results in converging verticals

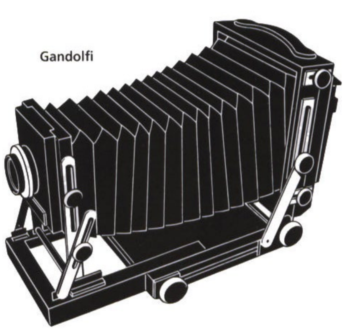


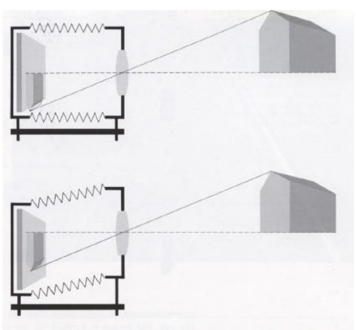
Keeping the camera level, with an ordinary lens, captures only the bottom portion of the building



Shifting the lens upwards results in a picture of the entire subject

- Solution: view camera (lens shifted w.r.t. film)





http://en.wikipedia.org/wiki/Perspective_correction_lens

Source: F. Durand

Perspective distortion

- Problem for architectural photography: converging verticals
- Result:

Source: F. Durand

Perspective distortion

- What does a sphere project to?

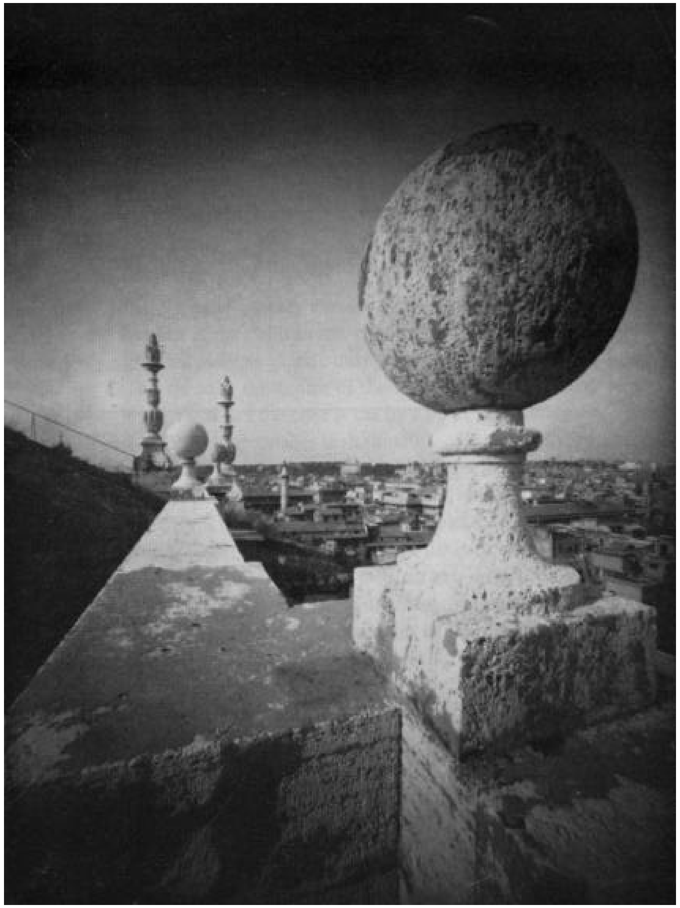
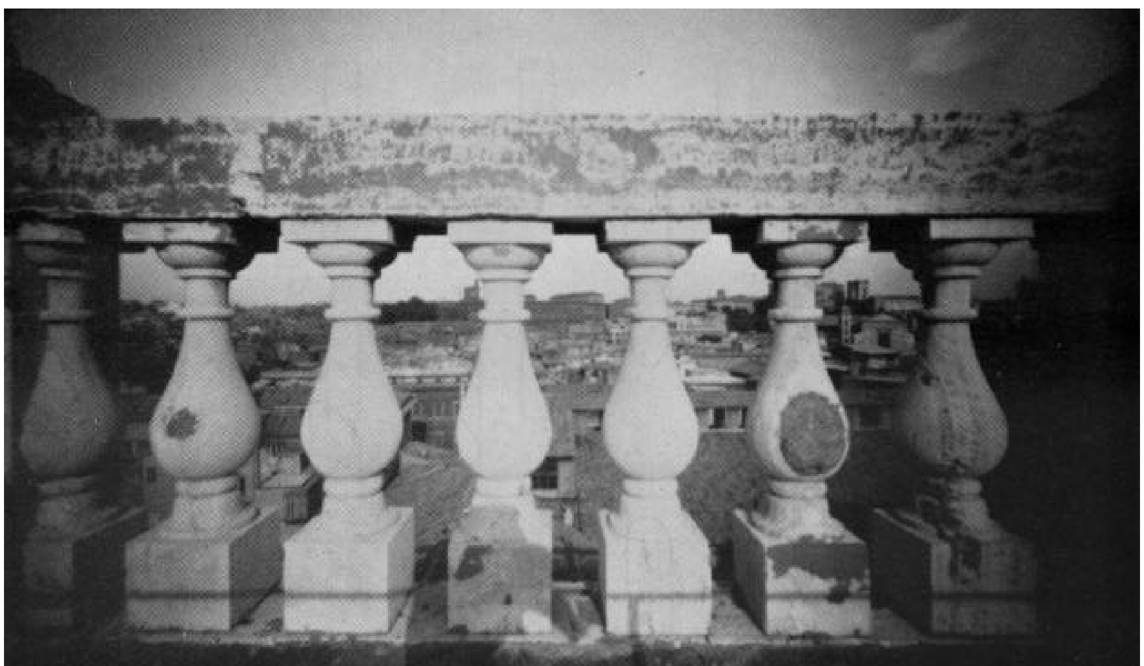
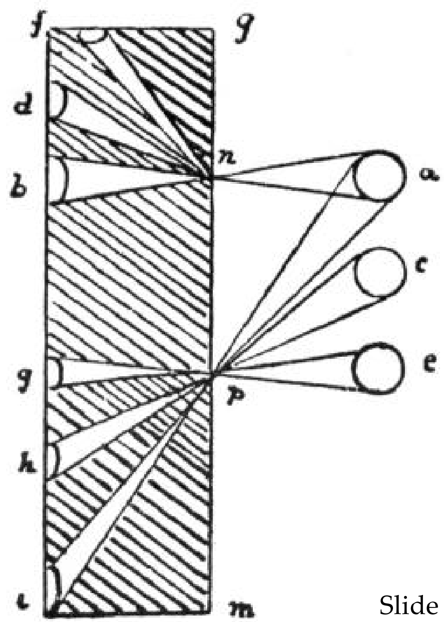


Image source: F. Durand

Perspective distortion

- The exterior columns appear bigger
- The distortion is not due to lens flaws
- Problem pointed out by Da Vinci



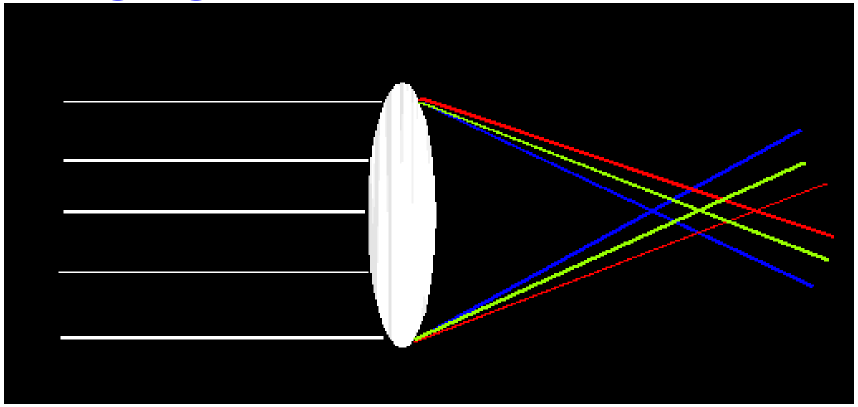


Slide by F. Durand

Perspective distortion: People

Lens Flaws: Chromatic Aberration

- Lens has different refractive indices for different wavelengths: causes color fringing

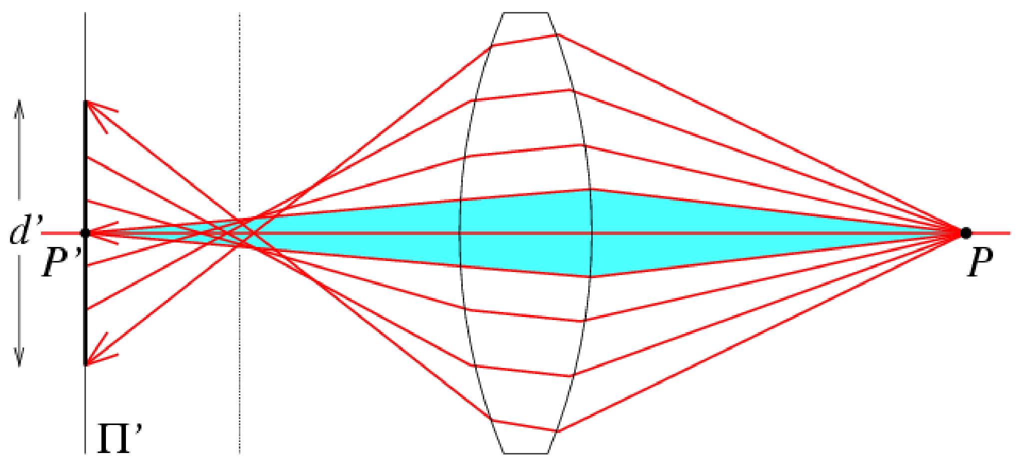


Near Lens Center

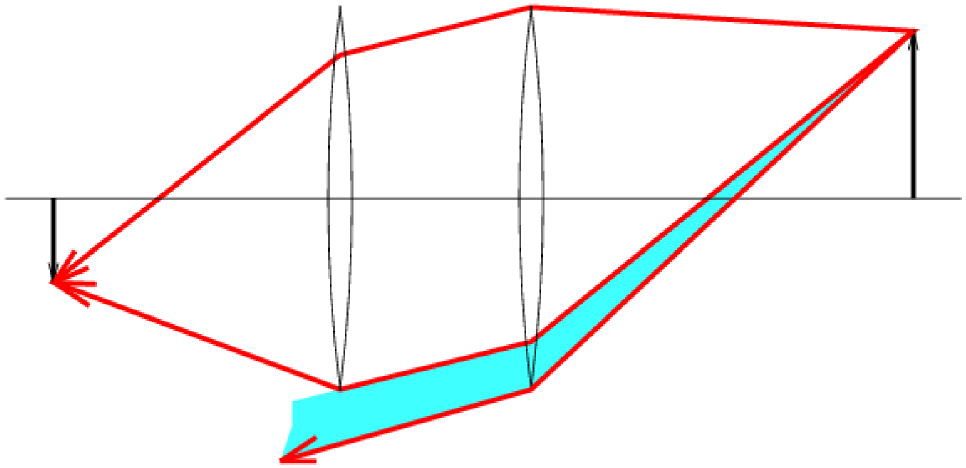
Near Lens Outer Edge

Lens flaws: Spherical aberration

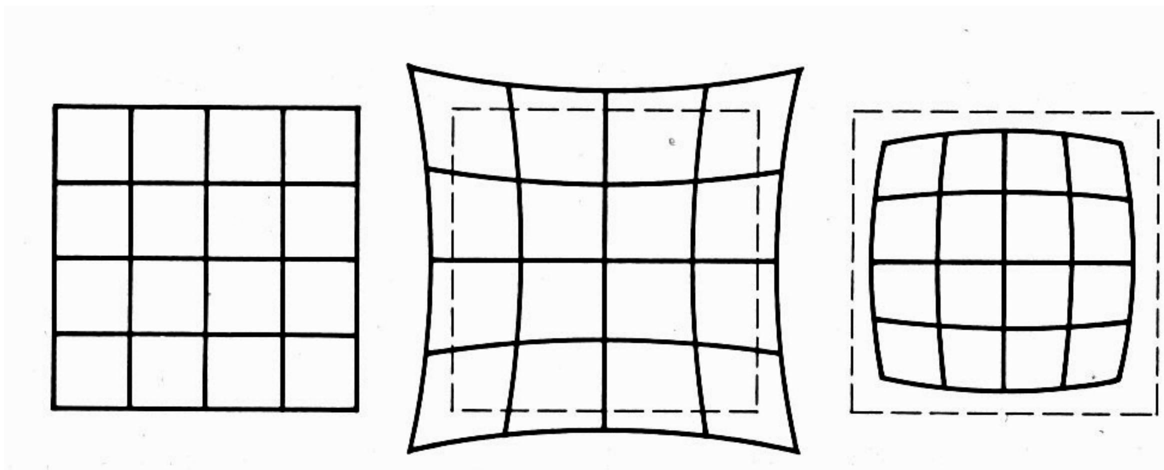
- Spherical lenses don’t focus light perfectly
- Rays farther from the optical axis focus closer



Lens flaws: Vignetting



Radial/Barrel Distortion



No distortion Pin cushion Barrel

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


©2004 Vincent Bockaert 123di.com


©2004 Vincent Bockaert 123di.com

Modeling distortion

Project $(\widehat{x}, \widehat{y}, \widehat{z})$ to "normalized" image coordinates

$$
\begin{aligned}
x'_n &= \widehat{x}/\widehat{z} \\
y'_n &= \widehat{y}/\widehat{z}
\end{aligned}
$$

Apply radial distortion

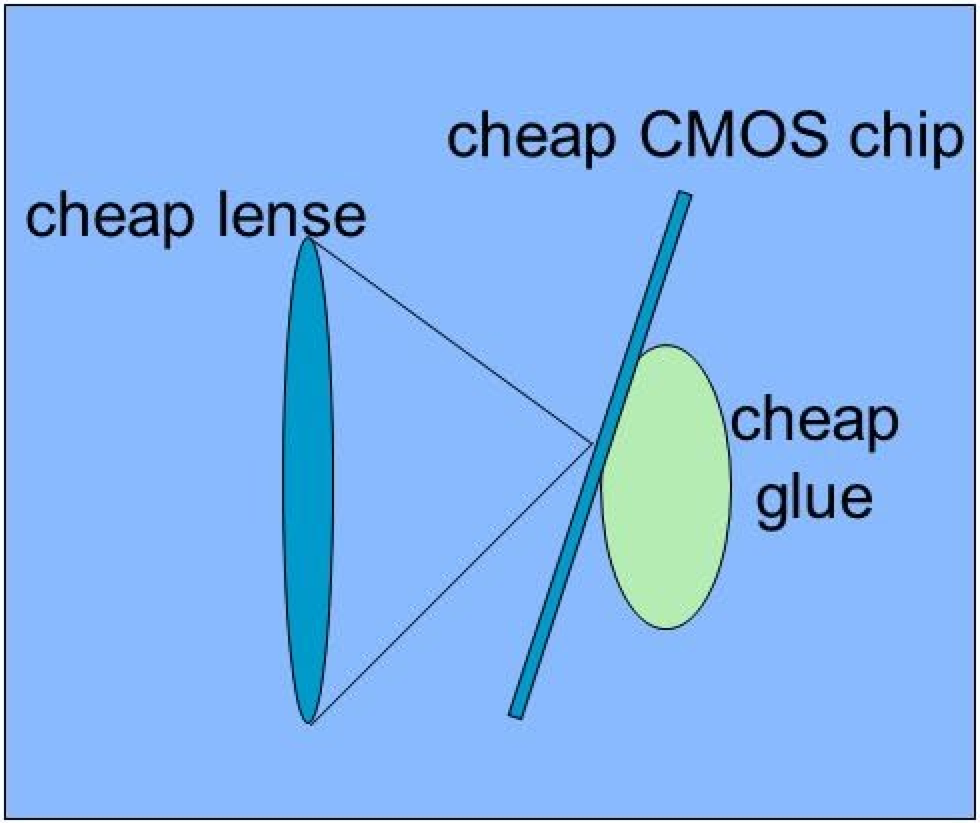
$$
\begin{aligned}
r^2 &= {x'_n}^2 + {y'_n}^2 \\
x'_d &= x'_n(1 + \kappa_1 r^2 + \kappa_2 r^4) \\
y'_d &= y'_n(1 + \kappa_1 r^2 + \kappa_2 r^4)
\end{aligned}
$$

Apply focal length translate image center

$$
\begin{aligned}
x' &= f x'_d + x_c \\
y' &= f y'_d + y_c
\end{aligned}
$$

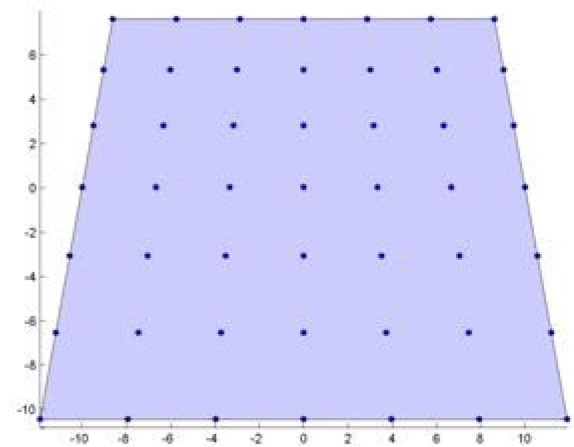
- To model lens distortion
  - Use above projection operation instead of standard projection matrix multiplication

Tangential Distortion



cheap camera

image

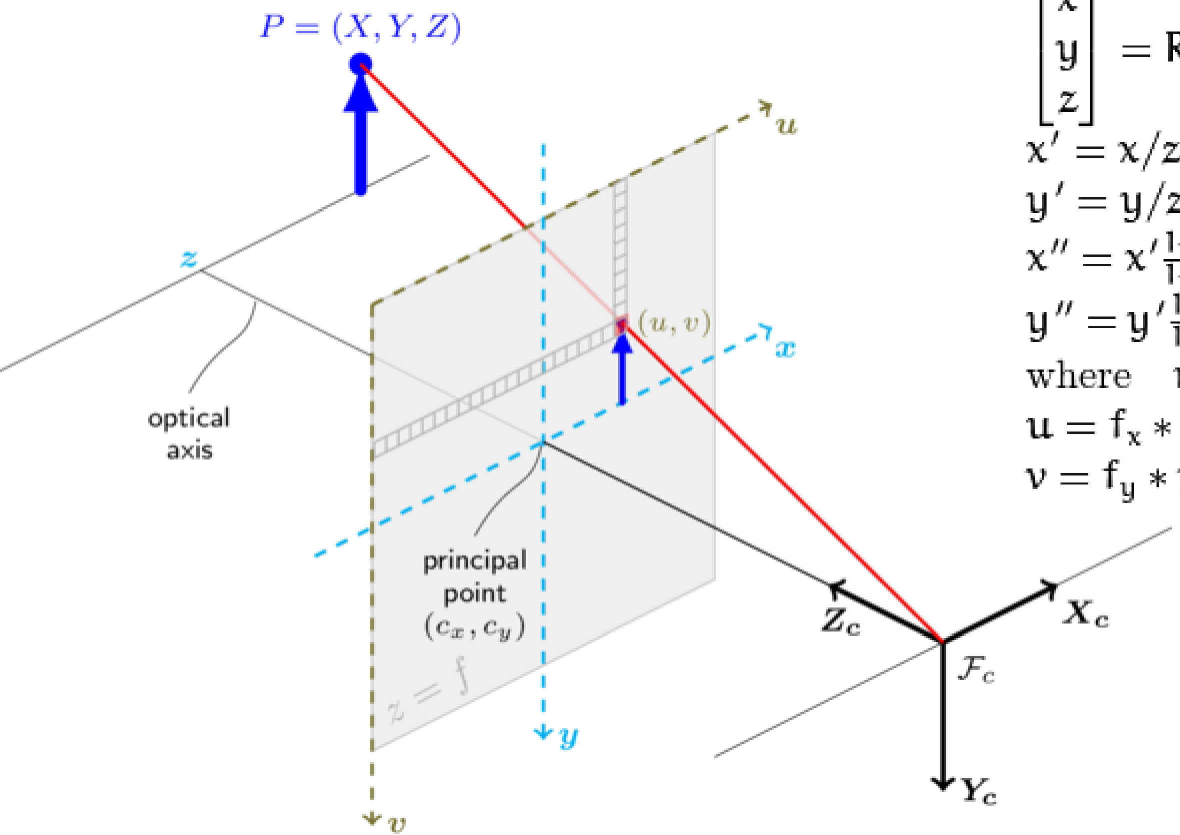


Tangential distortion correction:

$$x_{corrected} = x + [2p_1xy + p_2(r^2 + 1x^2)]$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2xy]$$

Distortion Correction Summary



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$
$$y' = y/z$$
$$x'' = x' \frac{1+k_1 r^2 + k_2 r^4 + k_3 r^6}{1+k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2(r^2 + 2x'^2)$$
$$y'' = y' \frac{1+k_1 r^2 + k_2 r^4 + k_3 r^6}{1+k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1(r^2 + 2y'^2) + 2p_2 x' y'$$
$$\text{where} \quad r^2 = x'^2 + y'^2$$
$$u = f_x * x'' + c_x$$
$$v = f_y * y'' + c_y$$

k_1, k_2, k_3, k_4, k_5, and k_6 are radial distortion coefficients. p_1 and p_2 are tangential distortion coefficients. Higher-order coefficients are not considered in OpenCV.

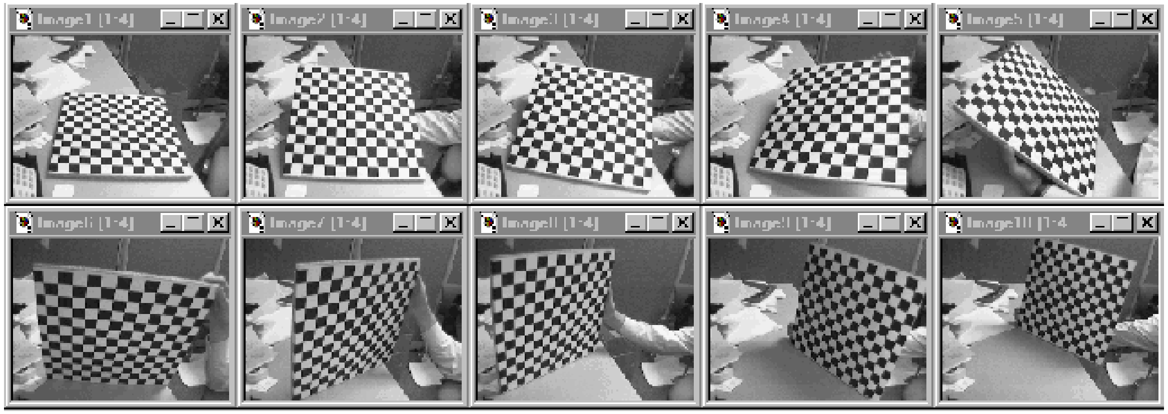
https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

Correcting radial distortion

from [Helmut Dersch]

Camera Calibration

Alternative: Multi-plane calibration

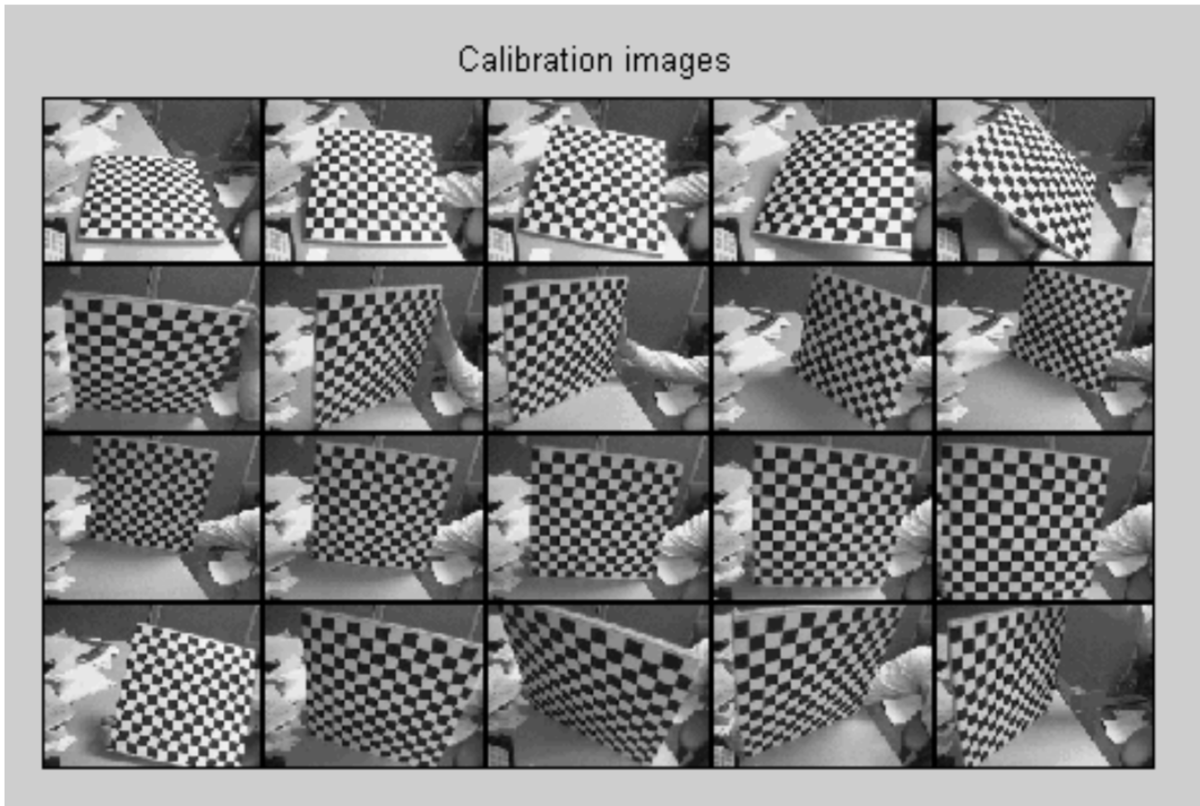


Advantages:

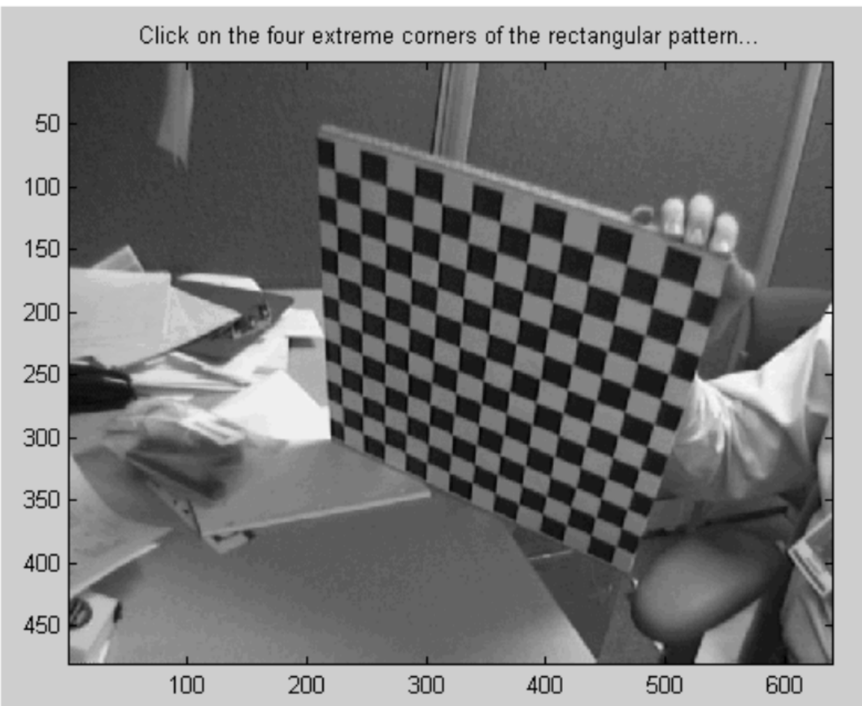
- Only requires a plane
- Don't have to know positions/orientations
- Great code available online!
  - Matlab version: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
  - Also available on OpenCV.

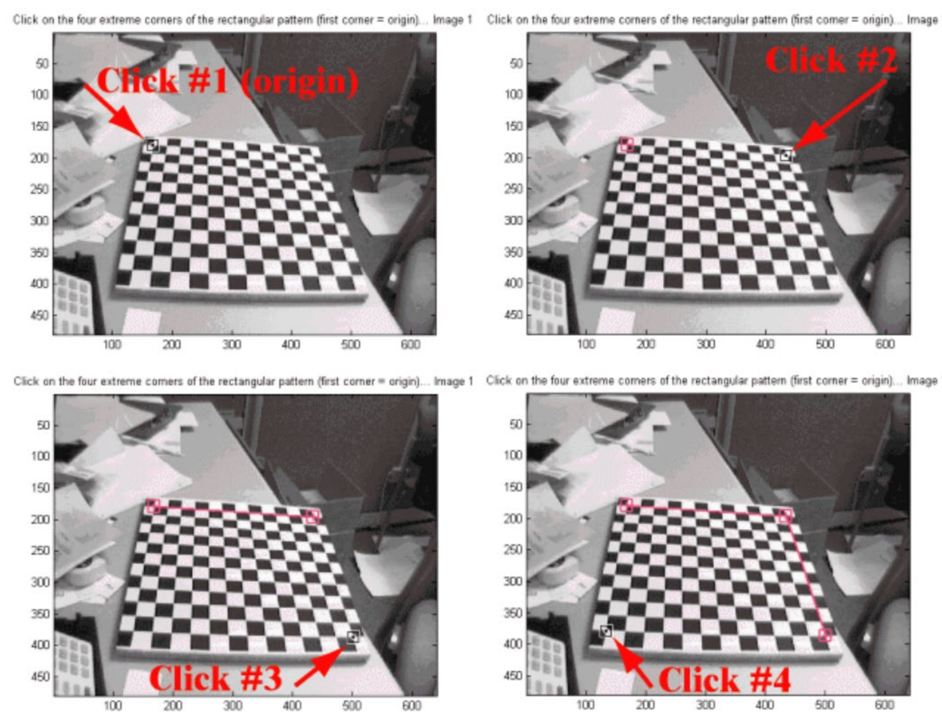
Disadvantage: Need to solve non-linear optimization problem.

Step-by-step demonstration

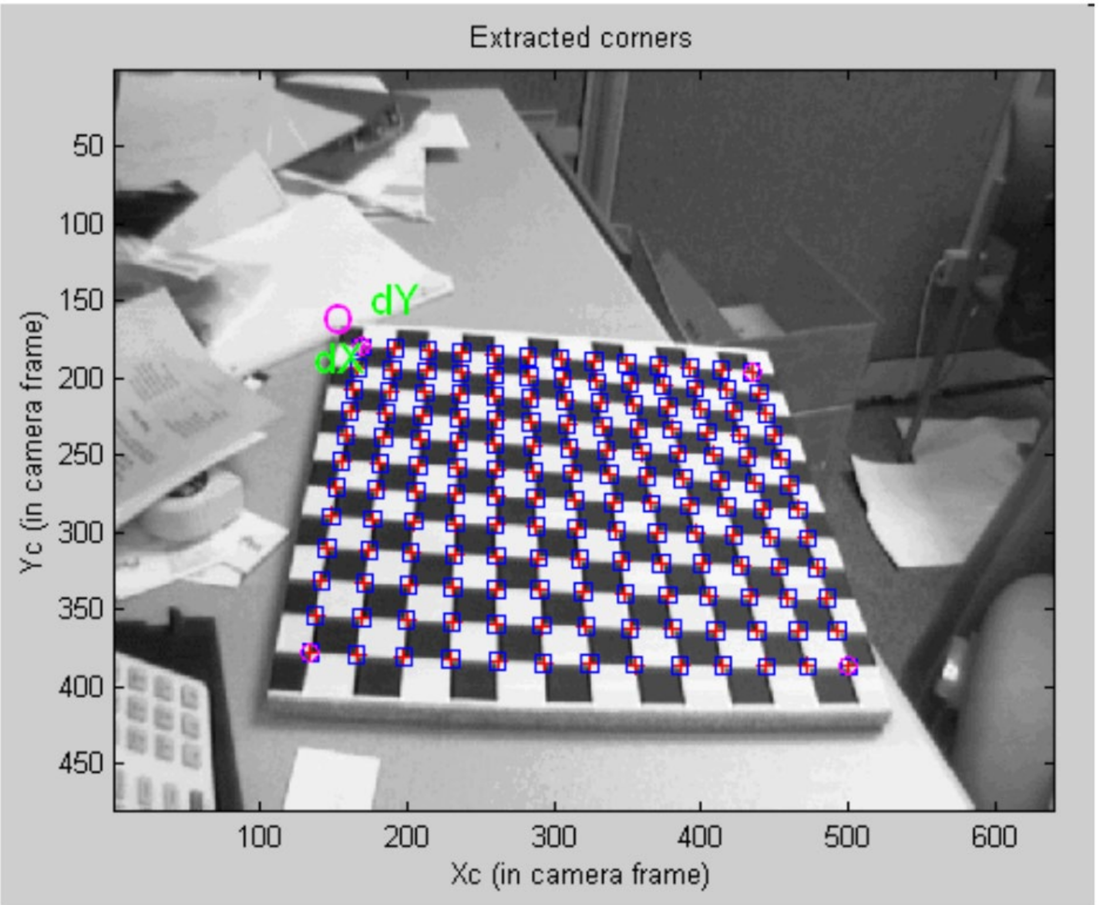


Step-by-step demonstration

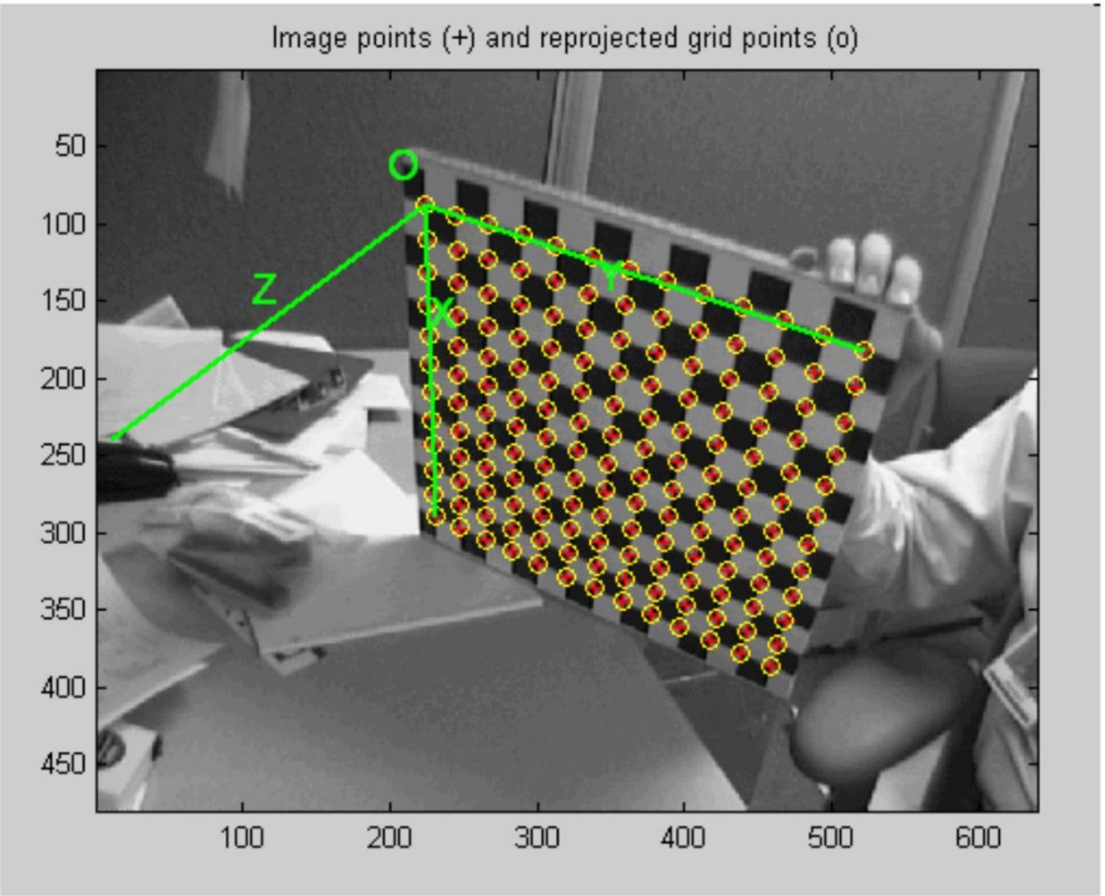




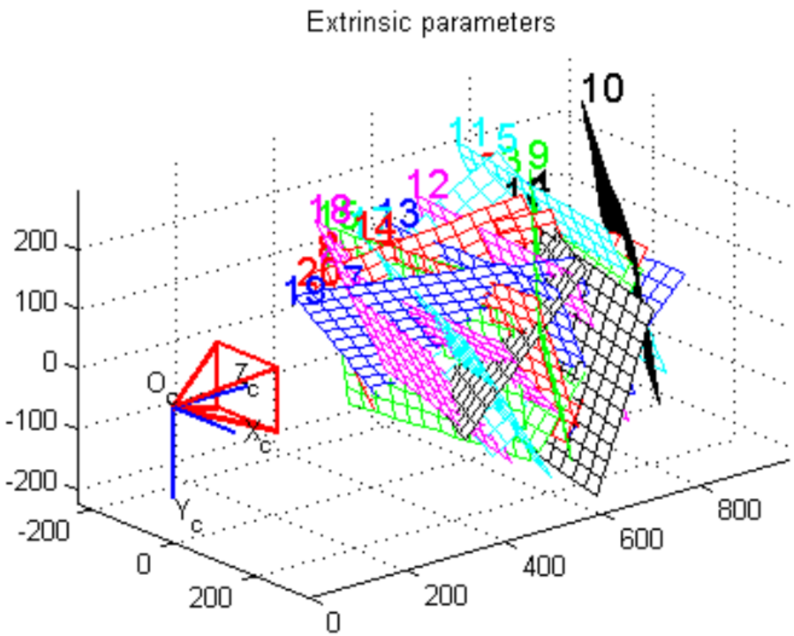
Step-by-step demonstration

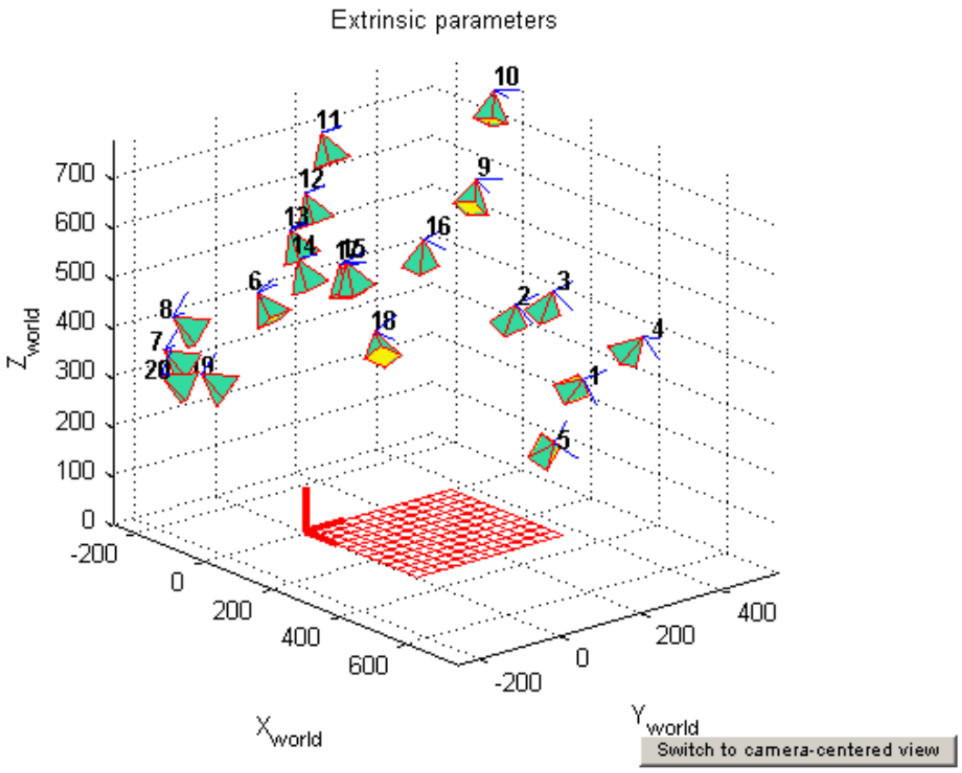


Step-by-step demonstration



Step-by-step demonstration





What does it mean to “calibrate a camera”?

Many different ways to calibrate a camera:

- Radiometric calibration.
- Color calibration.
- Geometric calibration.
- Noise calibration.
- Lens (or aberration) calibration.