Robot Vision

TTK4255

Lecture 02 – Image Formation

Annette Stahl

(Annette.Stahl@ntnu.no)

Department of Engineering Cybernetics – ITK

NTNU, Trondheim

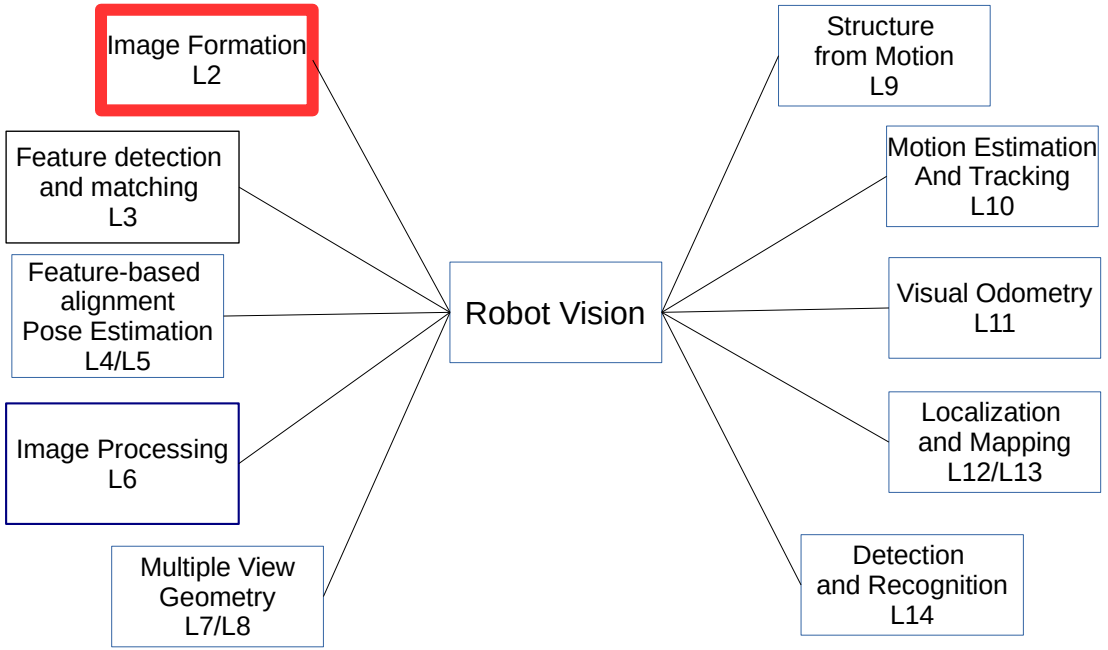
Spring Semester

13. January 2020

Lecture 02 – Image Formation

Annette Stahl (Annette.Stahl@ntnu.no)

Simen Haugo (Simen.Haugo@ntnu.no)

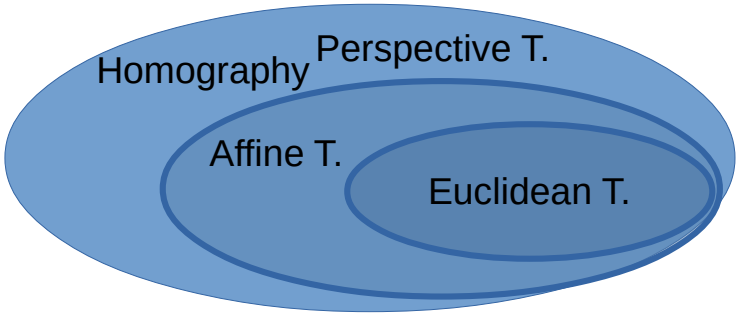


Outline of the second lecture:

- Motivation to Image Formation
- Representation of Images
- Image Formation
- Perspective Projection
- Camera Parameters
- Lens Distortion
- Camera Calibration
- Calibration Software

Recap L01

- Projective Space, important ingredient: Homogeneous Coordinates
- Projective Space $\supseteq$ Affine Space $\supseteq$ Euclidean Space
- Euclidean Geometry: Parallel Lines do not intersect!
- Perspective Geometry: Parallel Lines intersect at infinity (Vanishing Points → Horizon)
- Groups of linear projective transformations:



- Special Orthogonal Group SO(3) – Rotational Group

$$SO(3) = \left\{ R \in \mathbb{R}^{3\times 3} \mid R^\top R = I, \det R = +1 \right\}$$

- Special Euclidean Group SE(3) – Rigid Body Motion

$$SE(3) = \left\{ g = (R, T) \mid R \in SO(3), T \in \mathbb{R}^3 \right\}$$

- Composition: Chain together consecutive rotations (orientations)

$$R_{ac} = R_{ab} R_{bc}$$

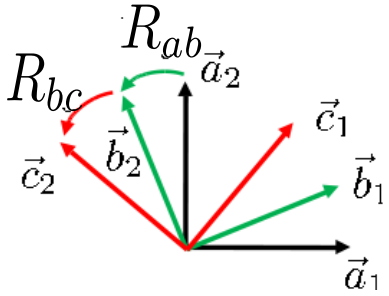


Image Projection

The mathematics of image projection allow us to answer two questions:

- Given a 3D scene, how does it project to the 2D camera images? (“forward” model)

  Image Formation/Projection - study of how objects (and surroundings) are mapped onto an image

- Given several images, what 3D scene does create them? (“inverse” model)

  3D reconstruction - inverse problem of image formation

  Attempt to use images to recover a description of objects in space. This is also about guessing the story behind the seen scene. The latter is the (largely ignored) holy grail of computer vision.

Image courtesy: triAyaam Vision Labs Pvt. Ltd

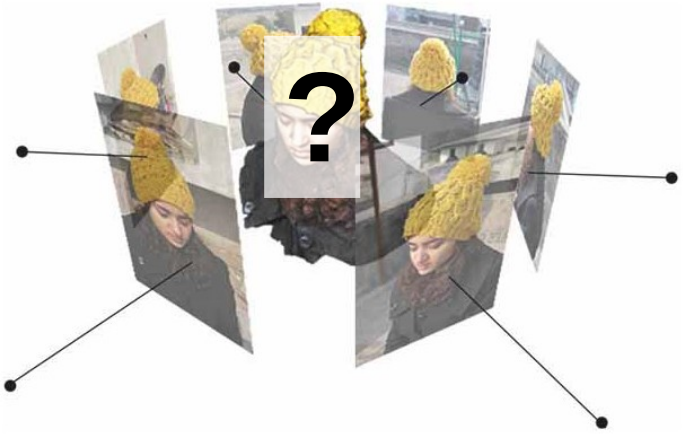


Image courtesy: triAyaam Vision Labs Pvt. Ltd

Image Projection

The understanding of projections is also exploited by artists:
The 3D scene (observed from a certain point) is projected onto the pavement

Visual Perception of a Robot

How does a robot perceive the world around it by using cameras?

We can equip an robot with cameras, but in order to process the recorded data we need an understanding of how a 3D scene projects to a 2D image.

→ Process of **Image formation**

Courtesy: NTNU Cyborg

Describing the camera model and the image formation mathematically, enables us to understand

- how a point in 3D corresponds to a point in the image and
- how an image will change as we move a camera in a 3D environment.

Application: use this information to perform complex perception tasks such as reconstructing 3D scenes from video

Representation of Images

Recall that a gray value image is usually represented by a two-dimensional brightness array.

It is a **map/function I** defined on a compact region of a two-dim surface, taking values in the positive real numbers

$$I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}_+$$
$$(x_1, x_2) \mapsto I(x_1, x_2)$$

The domain Ω and the range $\mathbb{R}_+$ are discretized.

Example: Digital Image:

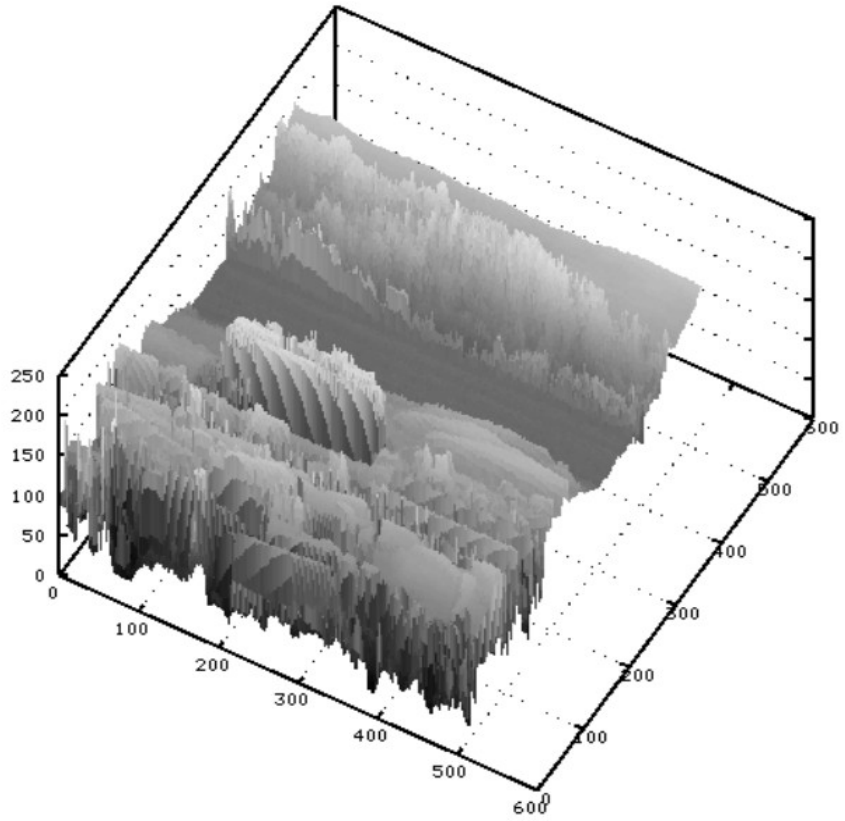
The domain $\Omega = [1, 512]x[1, 512] \subset \mathbb{Z}^2$ and the range $\mathbb{R}_+$ are discretized.

$\mathbb{R}_+$ is approximated by an interval of integers [0,255] $\in \mathbb{N}$.

Note: “At this stage a robot has no understanding of what is seen in an image”

Representation of Images

A different representation of an image, easily makes it unrecognizable for a human observer.



Octave/Matlab Code

Show the height field with Octave/MATLAB:

```
myimg=imread('imagename.png');
imshow(myimg)
[dimx,dimy]=size(myimg);
tx= linspace (1, dimx, dimx)';
ty= linspace (1, dimy, dimy)';
surf (tx, ty, myimg);
shading interp
```

Try:

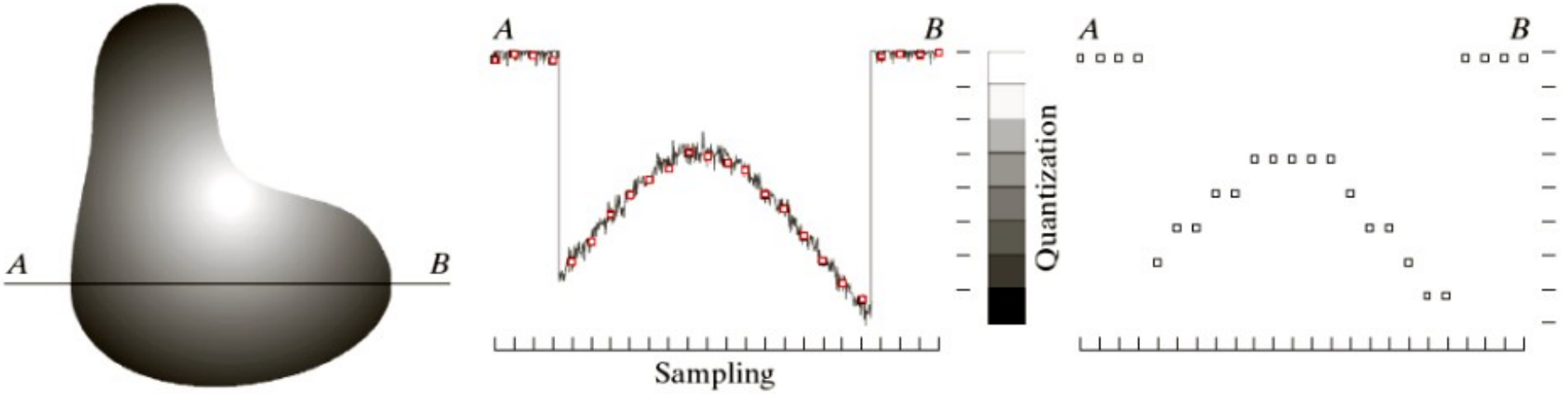
```
view(0,90)
view(30,65)
```

Sampling and Quantization

Continuous sensed data is converted into digital form by two processes:

Sampling: Digitize/Scan at discrete coordinates
→ For images most often performed at regularly spaced intervals

Quantization: Digitize the amplitude values



Sampling and quantization of a scan line between A and B.

Representation of Color Images

An image is a quantization in space and brightness of the light energy that is focused on the surface of a sensor.

Color images can be seen as a vector valued representation.

$$c = \begin{pmatrix} r \\ g \\ b \end{pmatrix}$$

red

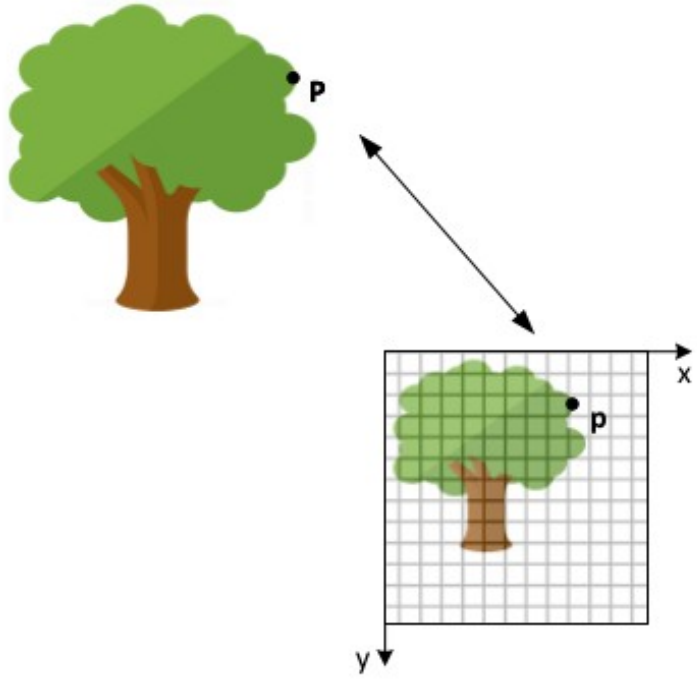
green

blue

Here the image is represented in the RGB-space, but several other useful color-spaces exist (HSV, HSL, HSB, CMYK, etc.)

Image Formation

Relation between a 3D scene point and its corresponding 2D point in the image:

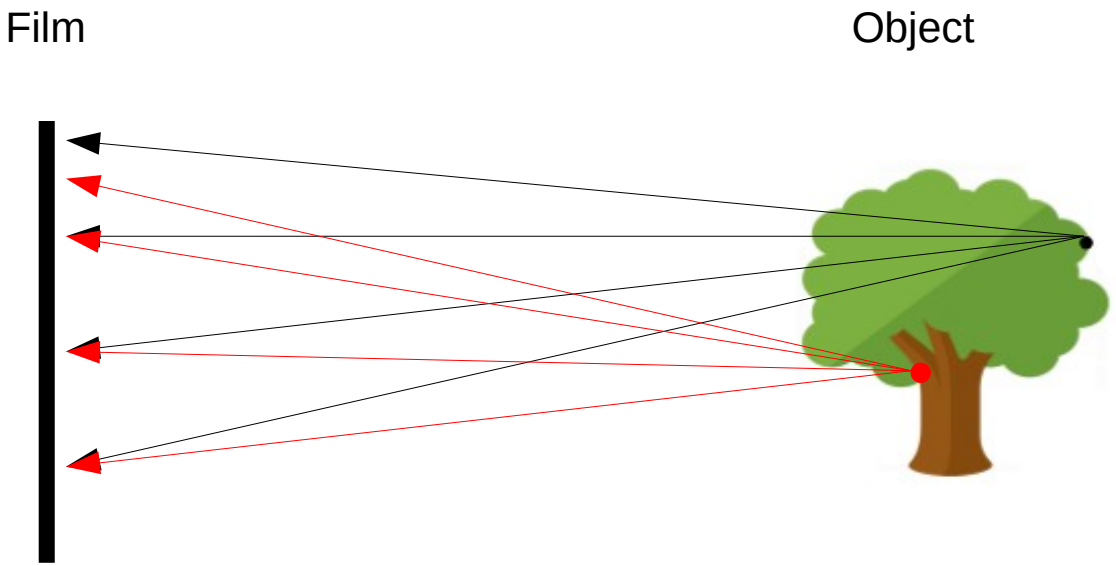


Formulate a precise correspondence between

- points in 3D space (in world coordinates) and
- their projections into a 2D image plane (in camera coordinates)

How to form an Image?

Place a film in front of an object.

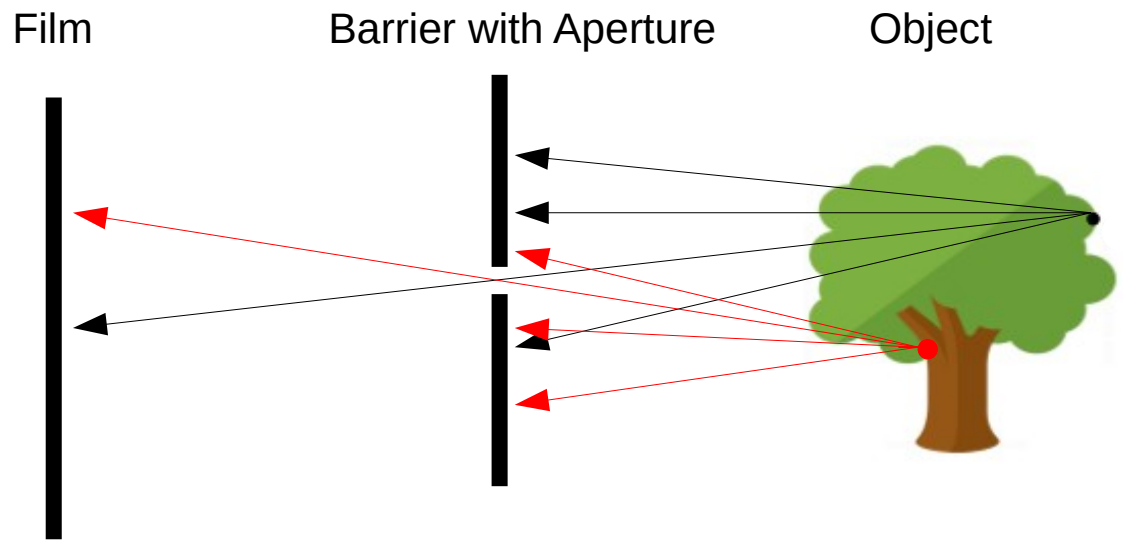


Would we receive an reasonable image?

How to form an Image?

Add a barrier with a small opening (aperture) in front of the film to block off most of the rays and

One has created a pinhole camera.

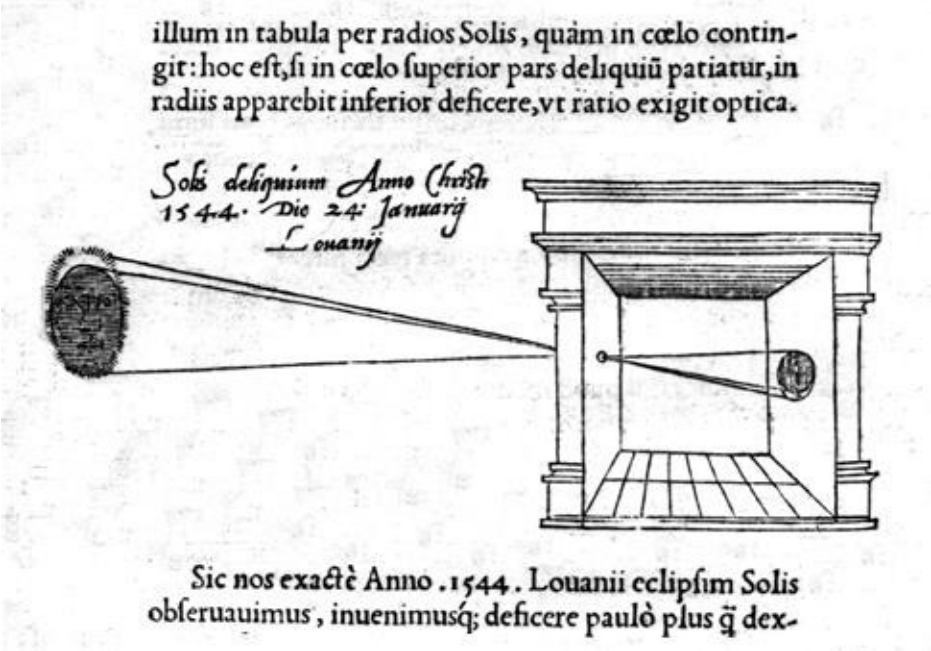


→ This reduces the blurring

Which orientation does the tree have on the film?

Camera Obscura

"Reinerus Gemma-Frisius, observed an eclipse of the sun at Louvain on January 24, 1544, and later he used this illustration of the event in his book De Radio Astronomica et Geometrica, 1545. It is thought to be the first published illustration of a camera obscura…". Hammond, John H., The Camera Obscura, A Chronicle.



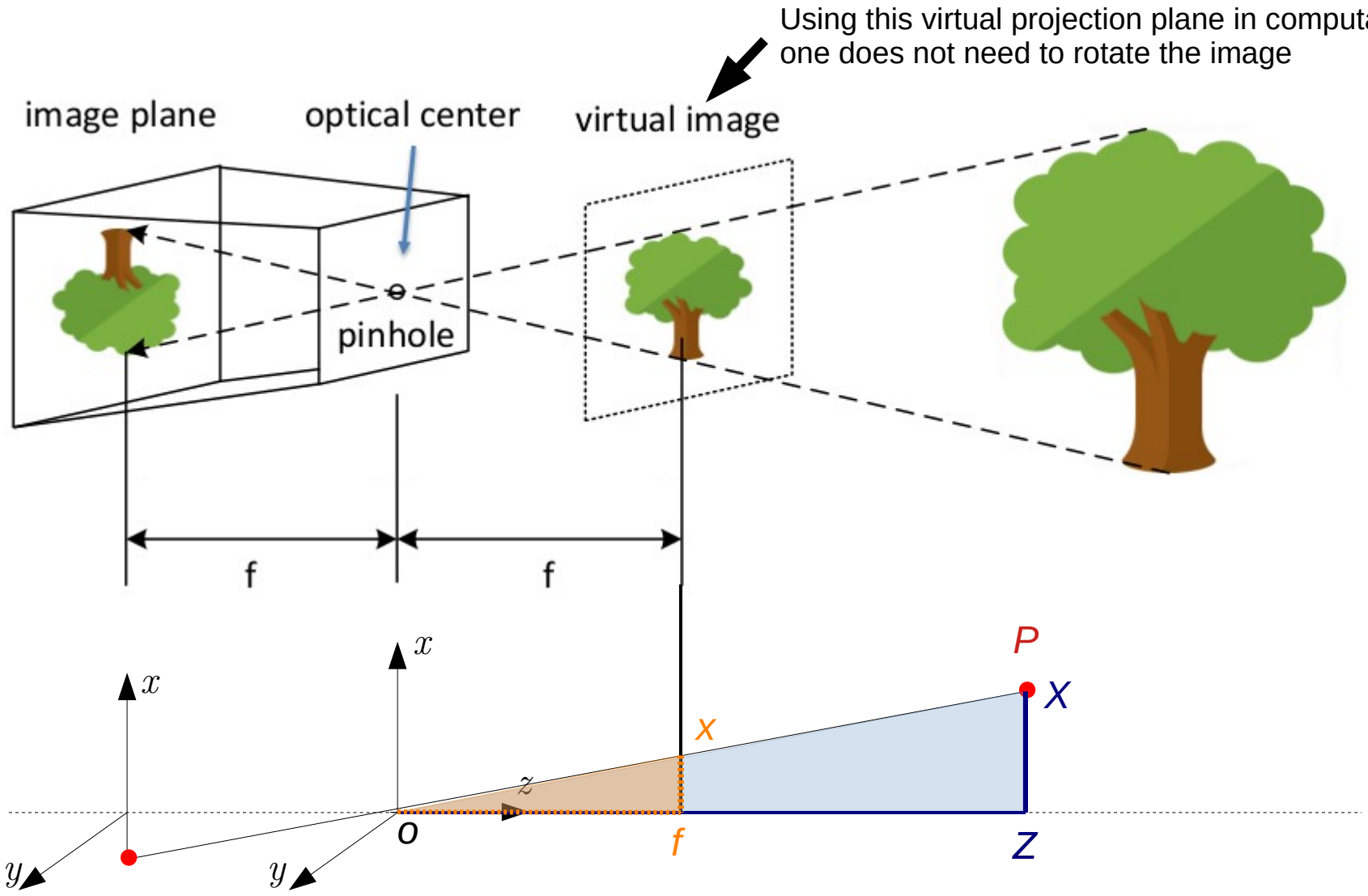
Real world example

Camera obscura! 2011

half alive - soo zzzz | CC A2.0

Image courtesy: P. Corke

Pinhole Camera Model



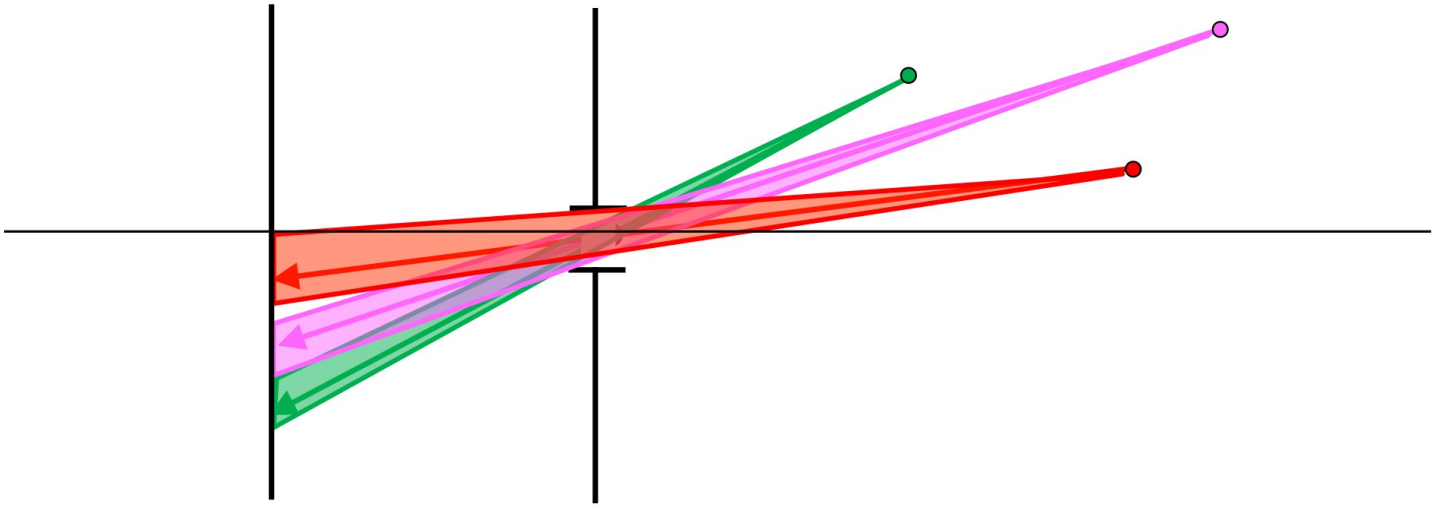
Using similar triangles:

$$\frac{x}{f} = \frac{X}{Z},\ \frac{y}{f} = \frac{Y}{Z} \quad \Rightarrow \quad x = f\frac{X}{Z},\ y = f\frac{Y}{Z}$$

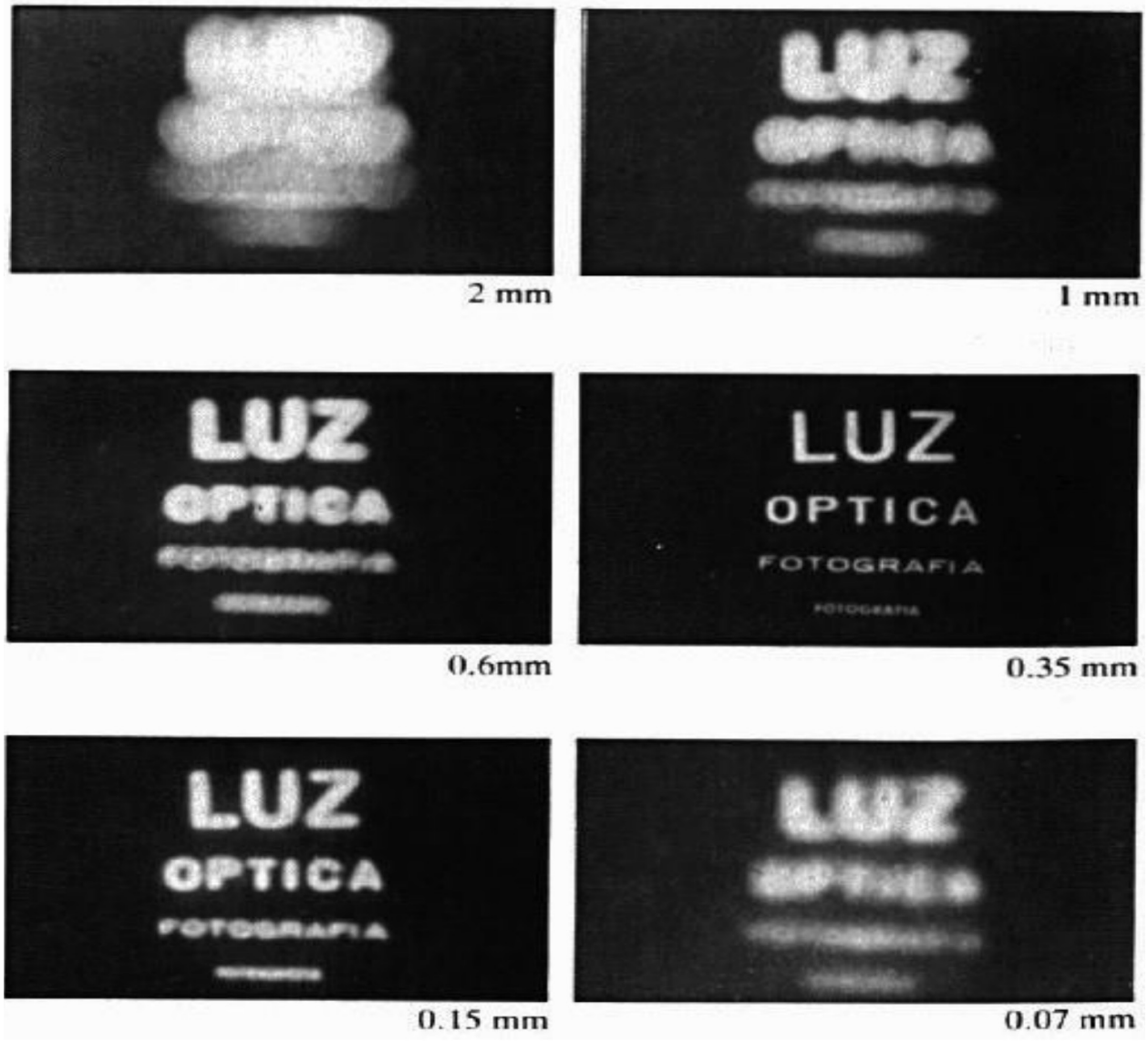
Size of the Aperture

In an ideal pinhole camera, only one ray of light reaches each point on the film ➔ the image can be very dim

- Making the aperture bigger makes the image blurry



Reduce the Size of the Aperture



Photos courtesy Dr. N. Joel, UNESCO

Make the aperture as small as possible

Less light gets through the opening

→ sharper but very dim

→ exposure time must be increased

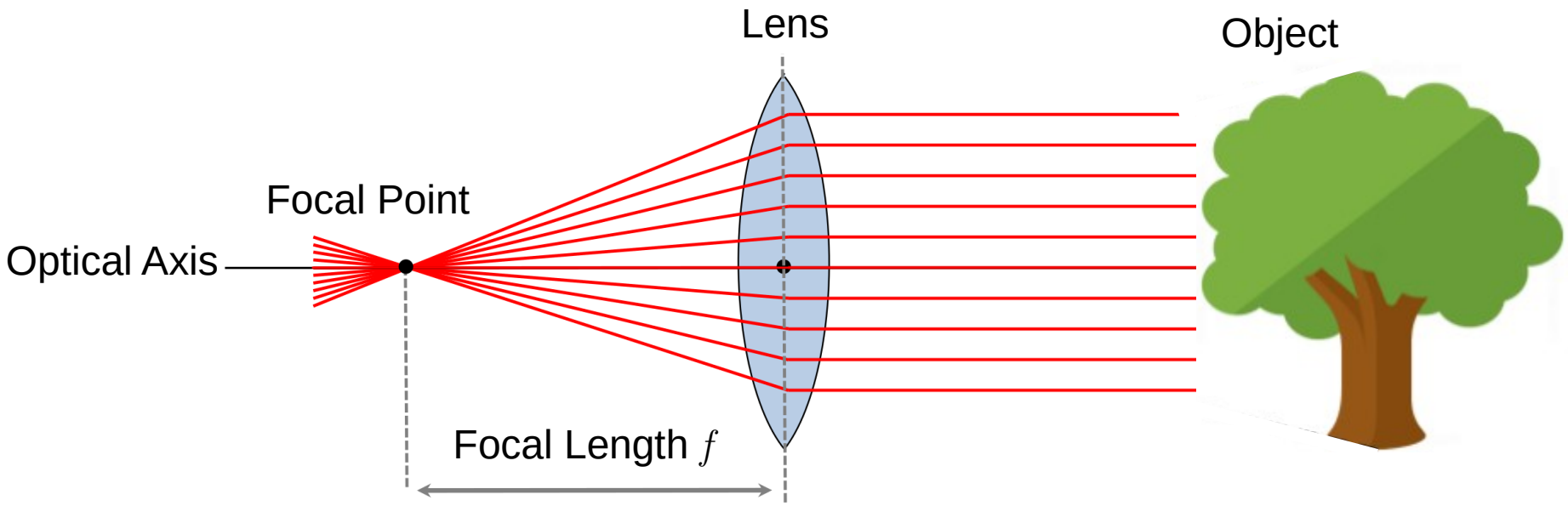
→ Diffraction effects occur

Lens

A lens focuses light onto the film

Rays passing through the Optical Center are not deviated

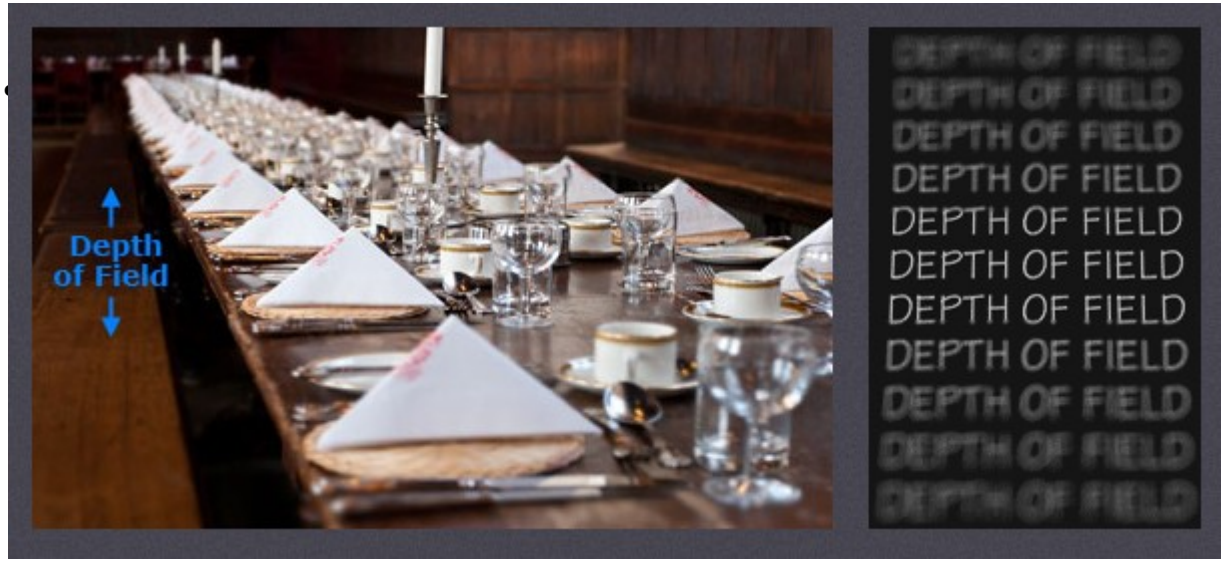
All rays parallel to the Optical Axis converge at the Focal Point



Focus and Depth of Field

Depth of Field (DOF) is the distance between the nearest and farthest objects in a scene that appear acceptably sharp in an image.

Although a lens can precisely focus at only one distance at a time, the decrease in sharpness is gradual on each side of the focused distance, so that within the DOF, the unsharpness is imperceptible under normal viewing conditions.

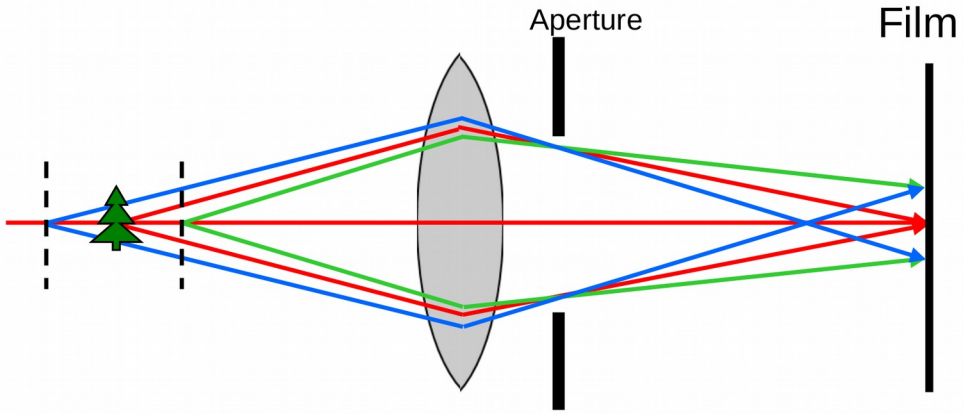


https://www.cambridgeincolour.com/tutorials/depth-of-field.htm

Focus and Depth of Field

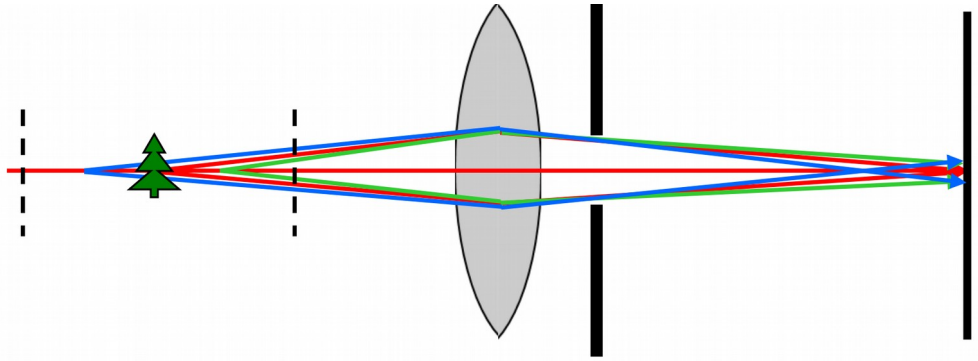
Aperture size effects the depth of field:

A larger aperture decreases the depth of field → need to decrease exposure time



f / 5.6

A smaller aperture increases the range in which an object is approximately in focus, it increases the DOF but reduces the amount of light into the camera
→ need to increase exposure time



f / 32

Image courtesy J. Bedros

Depth of Field

Depth of field refers to the range of distance that appears acceptably sharp.

It varies depending on

- camera type
- aperture
- focusing distance
- print size
- viewing distance

Field of View (FoV)

Angular measure of portion of 3D space seen by the camera

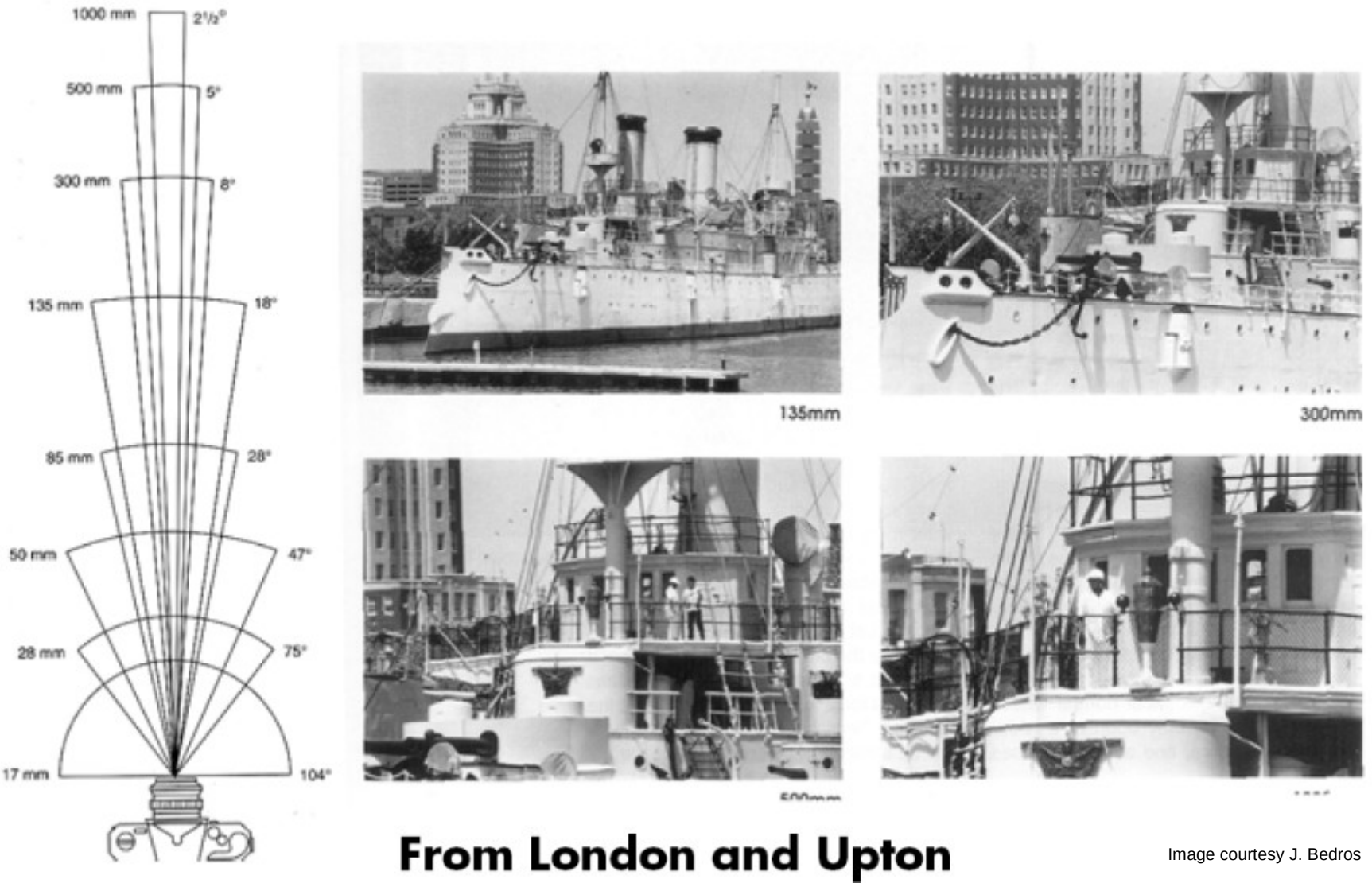


Image courtesy J. Bedros

Field of View

Field of view depends on focal length:

- As f gets smaller, image becomes more wide angle
  - more world points project onto the finite image plane

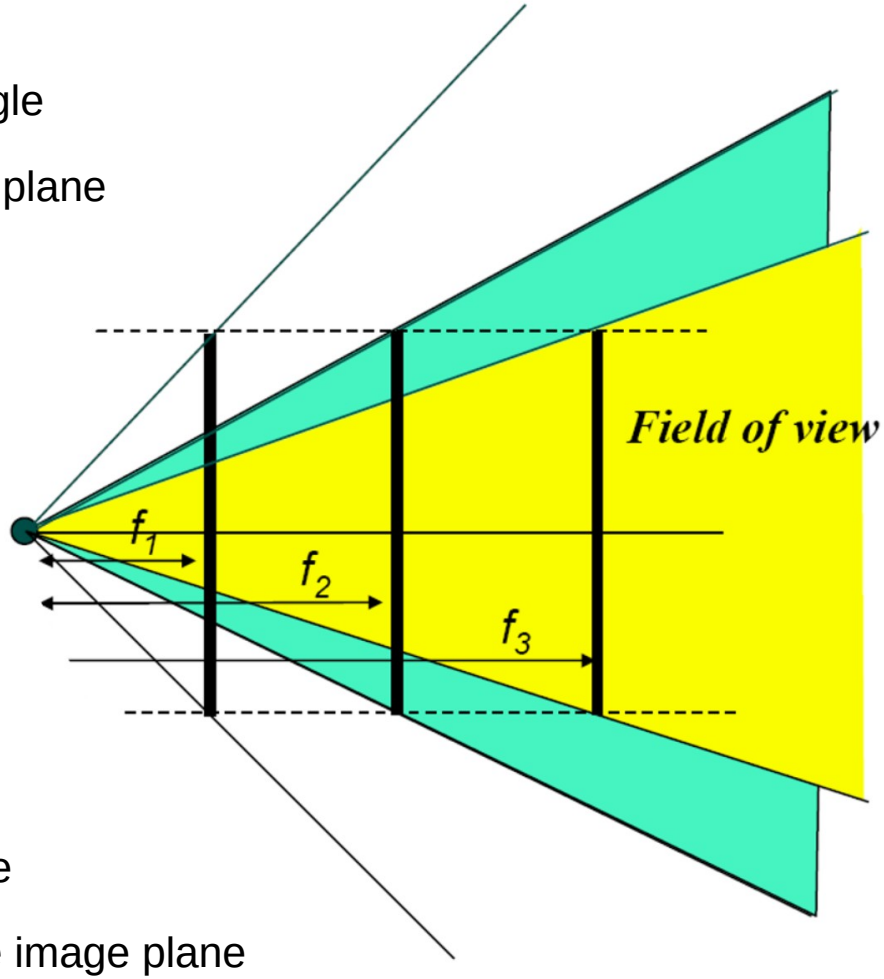


Image courtesy: D. Scaramuzza

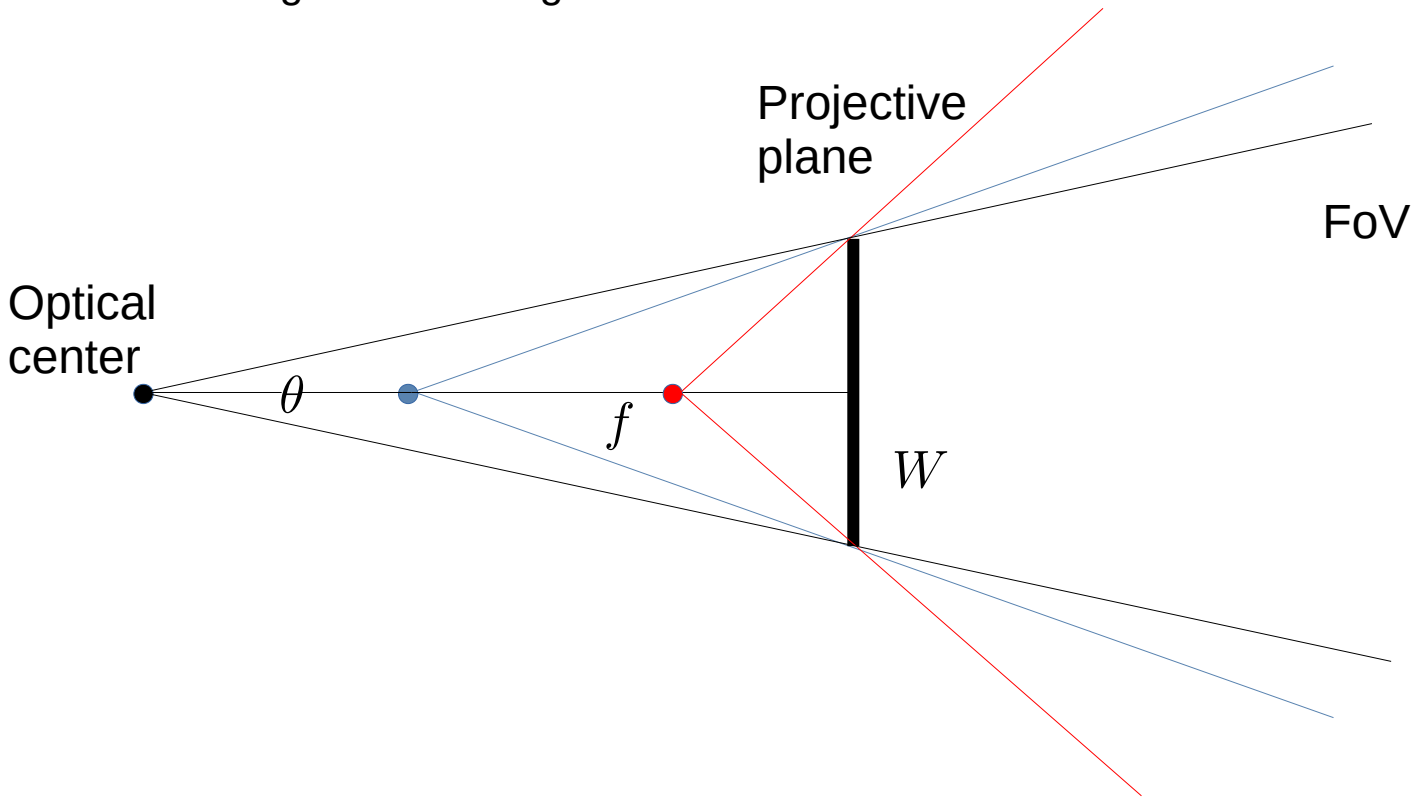
As f gets larger, image becomes more narrow angle

- smaller part of the world projects onto the finite image plane

Field of View

Relation between field of view and focal length

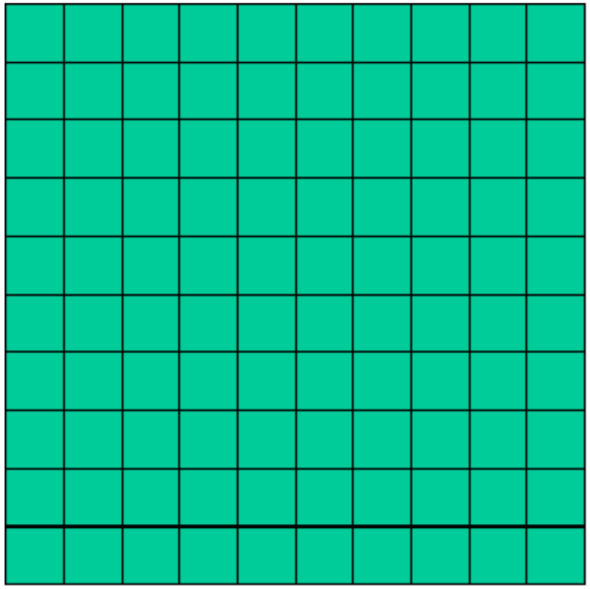
Smaller FOV = larger Focal Length



3 cases (black,blue,red) for different focal length are illustrated

$$\tan\frac{\theta}{2} = \frac{W}{2f} \quad \text{or} \quad f = \frac{W}{2}\left[\tan\frac{\theta}{2}\right]^{-1}$$

Rolling and Global Shutter



courtesy: D. Scaramuzza

Rolling Shutter

- Rows of pixels are exposed at different times, one after the other
- No problem for still or slow objects
- But may distort image for moving objects

Global Shutter

- All pixels are exposed simultaneously
- Good for moving objects
- No image distortion

Rolling and Global Shutter

Rolling shutter

Image courtesy: D. Scaramuzza

Global shutter

https://www.youtube.com/watch?v=17PSgsRlO9Q
https://www.youtube.com/watch?v=iXg_7Ckv_io

From World to Pixel Coordinates

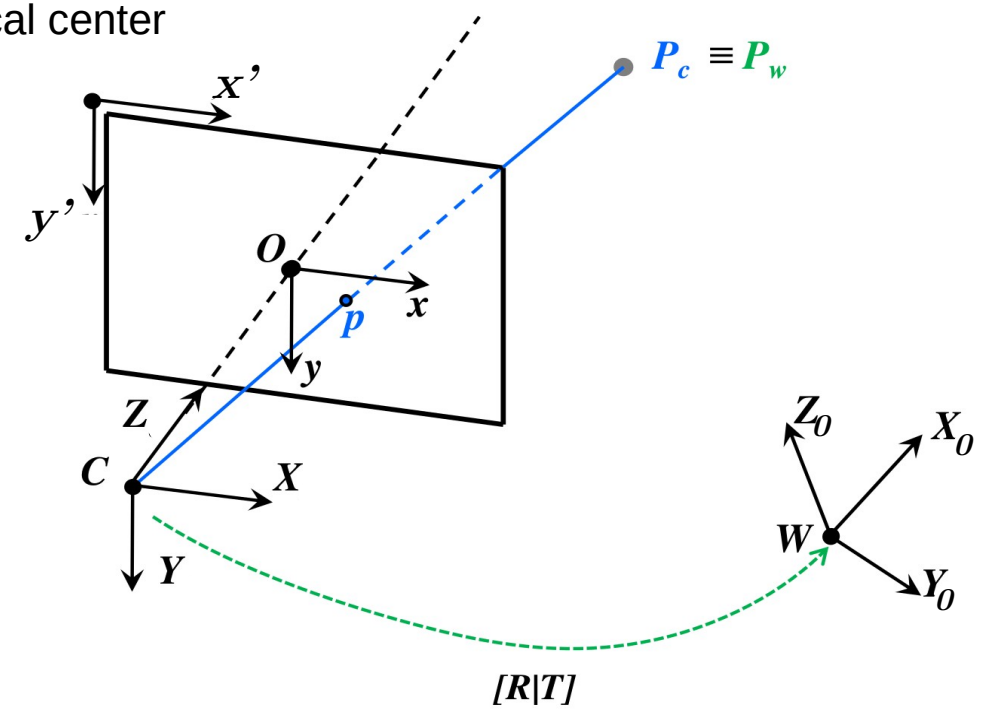
Involved Coordinate Systems:

I we wish to project a point in the world onto a camera plane we have to consider the following coordinate systems:

W – World coordinate system - origin arbitrary defined

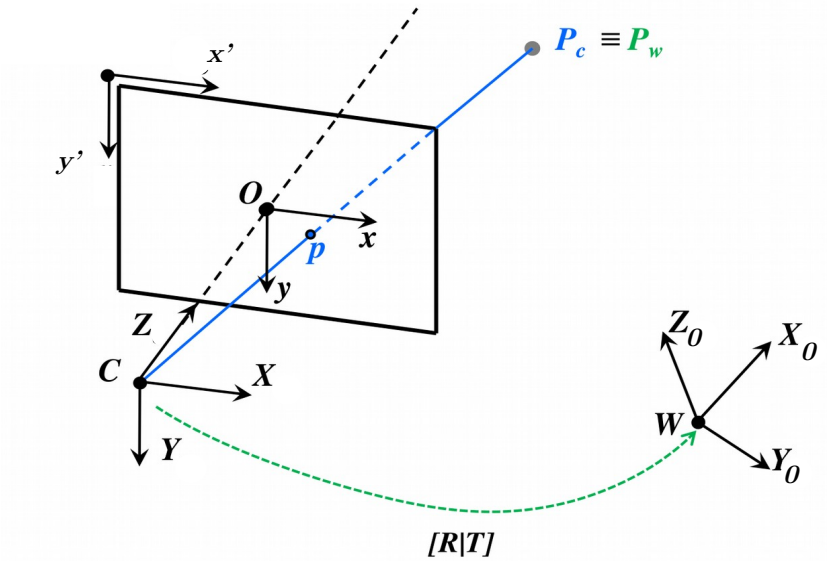
C – Camera coordinate system - origin in optical center

O – Image coordinate system.



From World to Pixel Coordinates

Aim: Find pixel coordinates (x',y') of a point $X_w = [X_0, Y_0, Z_0]^\top$ in the world coordinate frame.



A mathematical model must perform the following transformations:

1) Convert world point X_w to point $X_c = [X, Y, Z]$ in camera coordinates through a rigid body transformation $g(R,T)$

   Rigid-body motion: coordinate transformation between the camera frame and the world frame

2) Project point X_c to image-plane coordinates (x,y)

   Perspective projection/pinhole camera model: projection of 3D camera onto 2D image coordinates

3) Convert (x,y) to (discretized) pixel coordinates (x',y')

   Image transformation: coordinate transformation between possible choices of image coordinate

From World to Pixel Coordinates

Transformation from world coordinates X_w to camera coordinates X_c (rigid- body motion):

$X_c = R_{wc} X_w + T$, in homogeneous coordinates:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}$$

Using the pinhole camera model, a point X_c is projected onto the image plane at the point

$$x = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}$$

Using homogeneous coordinates we can express the projection as:

$$Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Note: One needs to scale the homogeneous weight to 1 before obtaining $x,\ y \rightarrow$ factor $\lambda = \frac{1}{Z}$

We can decompose the above matrix into two matrices:

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

Scaling projection matrix

Ideal Perspective Camera

Exploiting X_c in homogeneous coordinates $\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}$

one obtains the following expression for the projection of X_w to ideal image coordinates $[x, y, 1]^\top$

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}$$

3D rotation and translation

Scaling

Projection matrix

This is the overall geometric model for an ideal perspective camera.

Note:

- Here the origin of the image coordinates is at the intersection point with the optical axis (principle point)!
- “Ideal” refers to images with no distortion (straight lines, appear as straight lines in the image) in the scene

Image Coordinates

The previous model is specified for a reference coordinate system, centered at the optical center (the origin is aligned with the optical axis)

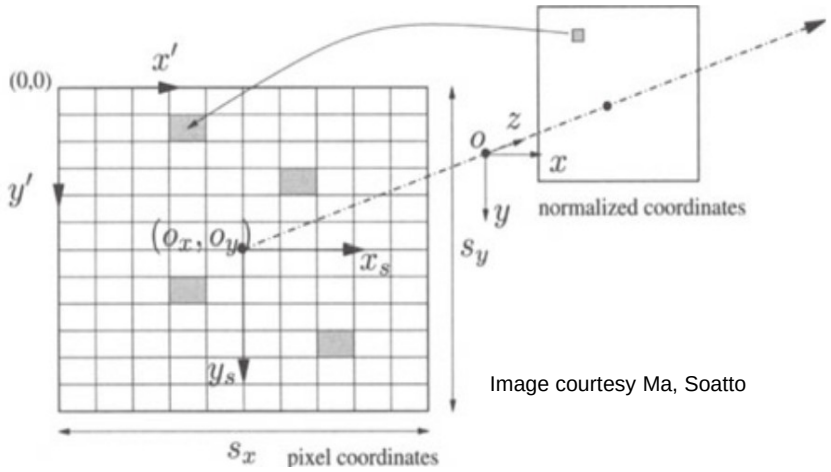


Image courtesy Ma, Soatto

Usually, using pixel coordinates the origin is not centered at the optical axis → top-left corner = (0,0)

Specify the relation between ideal coordinate frame and pixel array (Image transformation)!

Render the frame with respect to the pixel array = Specify scaling matrix (if pixels are notperpendicular → skew factor s_θ is non-zero)

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} s_x & s_\theta \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Place origin at the corner of the reference frame

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x_s + o_x \\ s_y + o_y \end{bmatrix}$$

Camera Matrix

The Image coordinates (x',y') can then be computed (in homogeneous coordinates) by

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

Combine ideal projection model with scaling and translation

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Multiplying of the coordinate transformation matrix with the scaling matrix leads to

$$K = K_s \cdot K_f = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Transformation - 3D point and 2D pixel

Finally, this results in the following projection equation:

Perspective projection matrix PI

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Intrinsic parameter matrix
Camera matrix K

- o_x: x-coordinate of the principal point in pixels,
- o_y: y-coordinate of the principal point in pixels,
- $fs_x = \alpha_x$: size of unit length in horizontal pixels,
- $fs_y = \alpha_y$: size of unit length in vertical pixels,
- α_x/α_y: aspect ratio σ,
- fs_θ: skew of the pixel, often close to zero.

Note: “size of unit length” refers here to pixel size.

Geometric Model of a Perspective Camera

To summarize, the geometric relationship between a point in world coordinates and its corresponding image coordinates in pixels depends on

1) the rigid-body motion (R,T) between the world and the camera frame also referred to as the **extrinsic** calibration parameters
2) the ideal projection
3) the **intrinsic** camera parameters

and can be expressed by the following equation

Perspective projection matrix Π

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}$$

Intrinsic parameter matrix
Camera matrix K

Extrinsic parameter matrix
3D rotation and translation

Extrinsic Parameters: Parameters that define the location and orientation of the camera reference frame with respect to a known world ref reference frame – rigid-body motion g(R,T)

Intrinsic parameters: Parameters necessary to link the pixel coordinates of an image point with the corresponding coordinates in the camera reference frame (focal length + principal point)

Radial Distortion

In simple/cheap cameras or cameras with a wide field of view, one can often observe significant distortions.

Distorted

Corrected

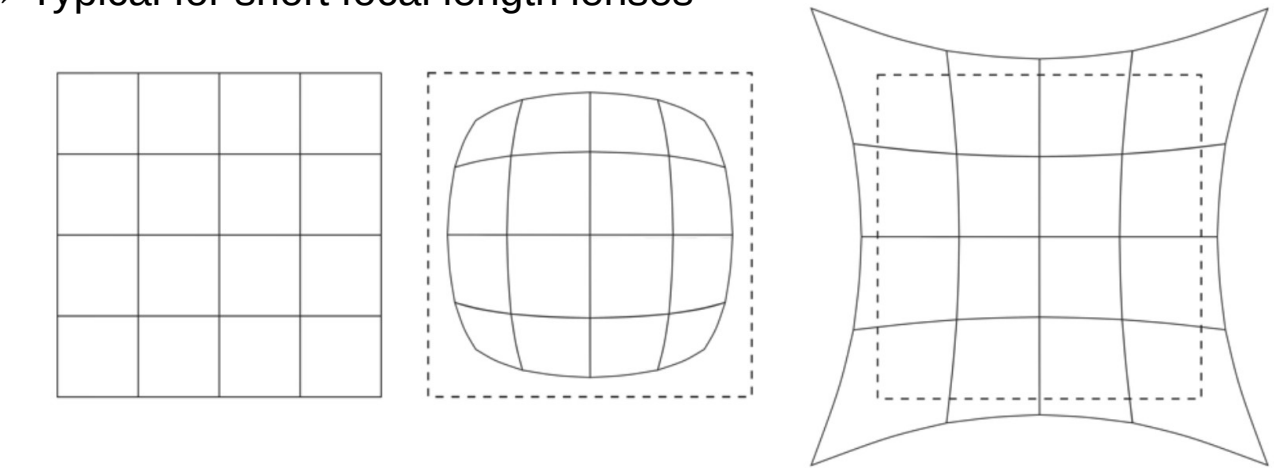
Image courtesy Ma, Soatto

“Straight lines do not appear as straight lines”

Radial Distortion

Radial distortion (positive and negative):

Radial distortion occurs when light rays bend more near the edges of a lens than they do at its optical center → Typical for short focal length lenses



No distortion

Positive radial distortion
(Barrel distortion)

Negative radial distortion
(Pincushion distortion)

Image courtesy: OpenCV Documentation

Fisheye Lens

Extreme example for radial distortion: Fisheye - lens

Tanner Helland (dot) com

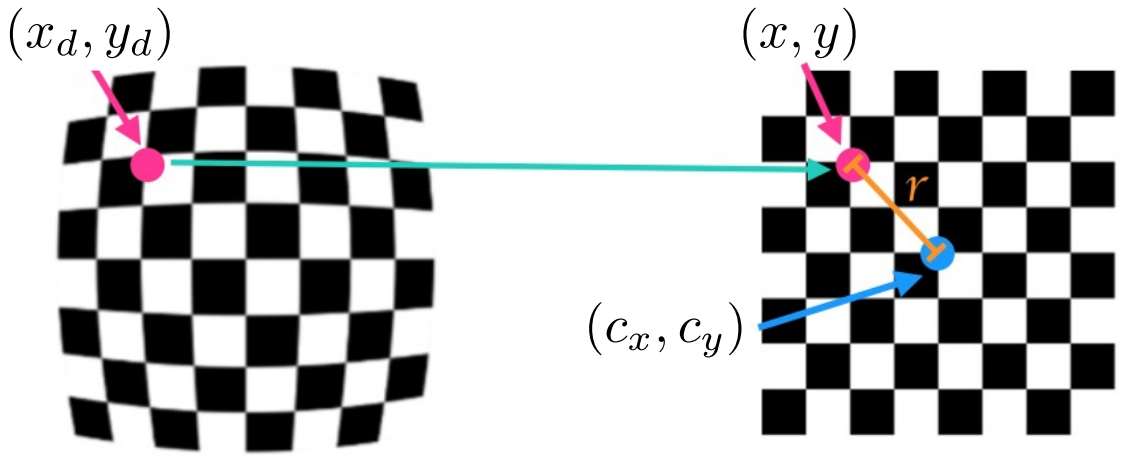
Radial Distortion

The standard model of radial distortion is a transformation from the ideal coordinates (x,y) (i.e. non-distored) to the real, observed coordinates (distorted) (x_d, y_d)

For a given non-distorted image point (x,y), the amount of distortion is a nonlinear function of its distance r from the principal point. For most lenses, a simple **quadratic model** of distortion produces good results

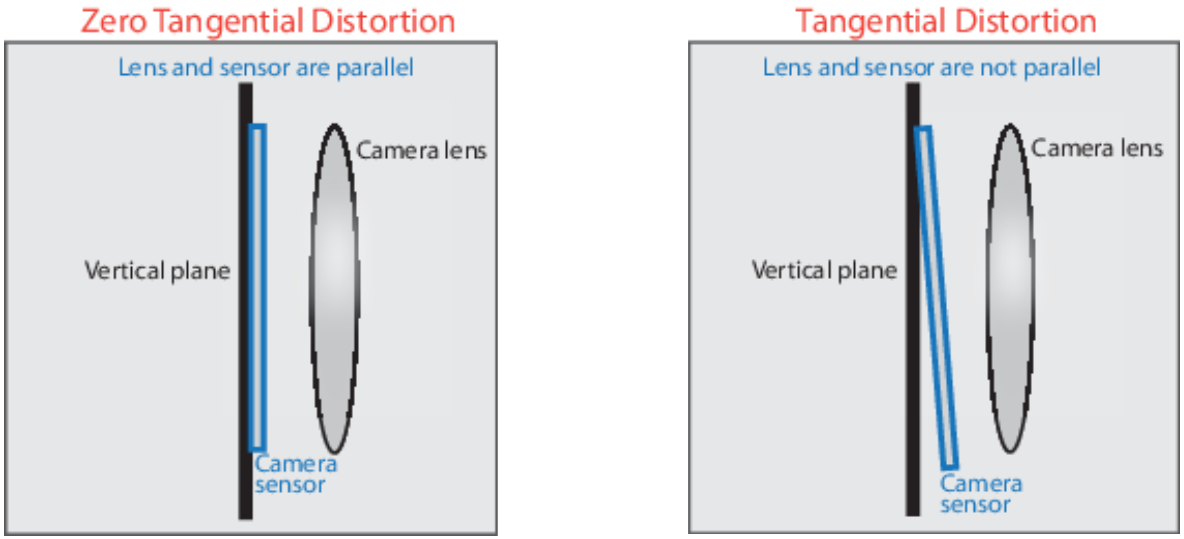
$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = (1 + k_1 r^2) \begin{bmatrix} x - c_x \\ y - c_y \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}$$

Where $r^2 = (x - c_x)^2 + (y - c_y)^2$ measured from the principle point or distortion center (c_x, c_y)



Tangential Distortion

Another form of distortion one might have to correct for is known as tangential distortion:



Courtesy: Matlab.com

If the lens is misaligned (not perfectly orthogonal to the image sensor), a non-radial distortion is introduced

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} 2k_4(x-c_x)(y-c_y) + k_5(r^2 + 2(x-c_x)^2) \\ k_4(r^2 + 2(y-c_y)^2 + 2k_5(x-c_x)(y-c_y) \end{bmatrix}$$

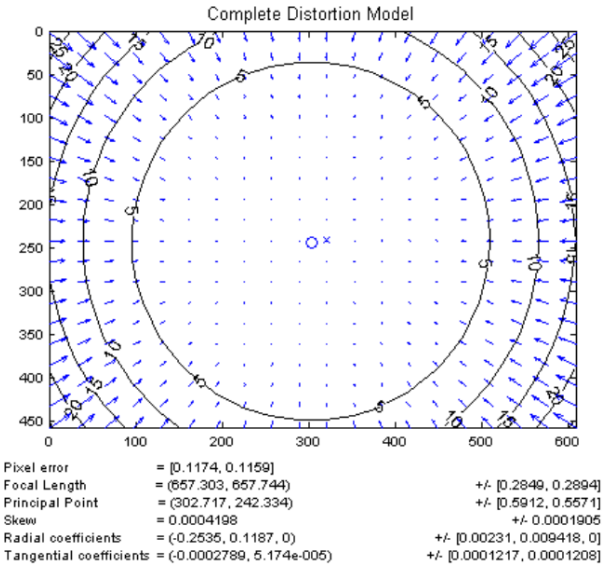
Radial and Tangential Distortion

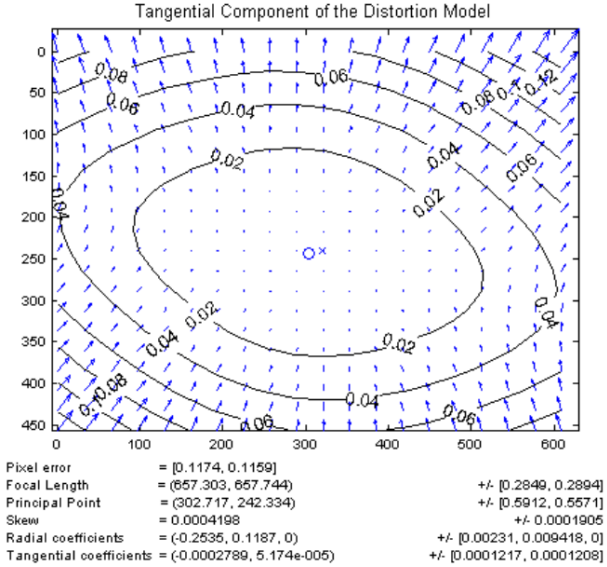
Depending on the distortion effect (camera field of view), **higher order terms** can be introduced for the radial distortion.

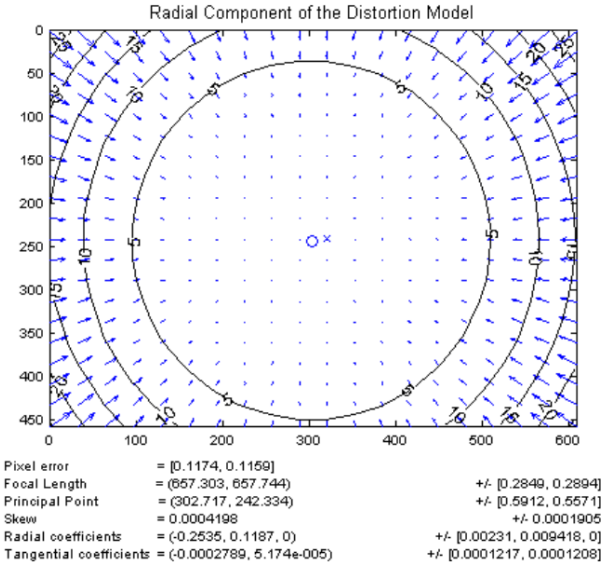
Radial distortion

Tangential distortion

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{bmatrix} x - c_x \\ y - c_y \end{bmatrix} + \begin{bmatrix} 2k_4(x - c_x)(y - c_y) + k_5(r^2 + 2(x - c_x)^2) \\ k_4(r^2 + 2(y - c_y)^2 + 2k_5(x - c_x)(y - c_y) \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}$$







The **left figure** shows the impact of the **complete distortion model (radial + tangential)** on each pixel of the image. Each arrow represents the effective displacement of a pixel induced by the lens distortion. Observe that points at the corners of the image are displaced by as much as **25 pixels**. The **center figure** shows the impact of the **tangential component of distortion**. On this plot, the **maximum induced displacement is 0.14 pixel** (at the upper left corner of the image). Finally, the **right figure** shows the impact of the **radial component of distortion**. This plot is very similar to the full distortion plot, showing that the tangential component could very well be discarded in the complete distortion model. On the three figures, the cross

http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html

Practical Tools

- OpenCV documentation:

  http://docs.opencv.org/2.4.13.3/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

  cv::undistort(img, undist_img, P, distCoeffs);

  cv::undistortPoints(pts,undist_pts,P,distCoeffs);

- Matlab documentation:

  https://se.mathworks.com/help/vision/ref/undistortimage.html

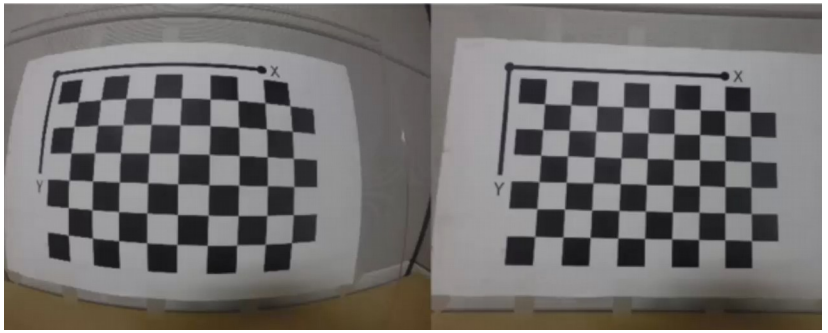
  [undist_img,newOrigin] = undistortImage(img,cameraParams);

  undistortedPoints = undistortPoints(points,cameraParams);

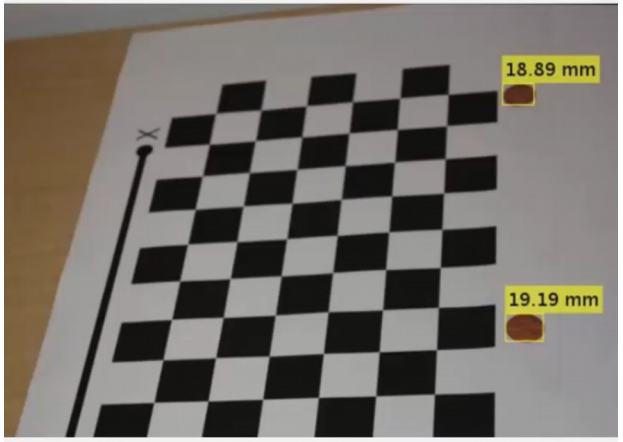
Camera Calibration - Motivation

When to perform a Camera Calibration

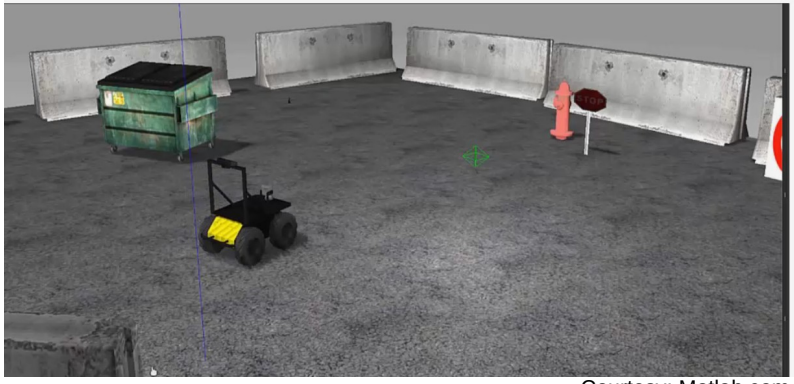
- correct for lens distortion



- measure the size of an object in world units



- determine the location of the camera in the scene (navigation)



Courtesy: Matlab.com

Camera Calibration

Camera calibration for a mono camera is the process to determine the intrinsic parameters of the camera model. It can be coupled with estimating the distortion parameters.

Correct for distortion
→ satisfy the perspective camera model

Estimate the camera calibration matrix K
→ correspondences between points in the world and point in a undistorted image.

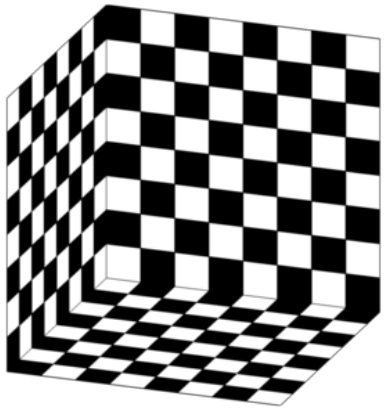
Intrinsic parameters: camera's internal characteristics

- focal length,
- coordinates of the principle point,

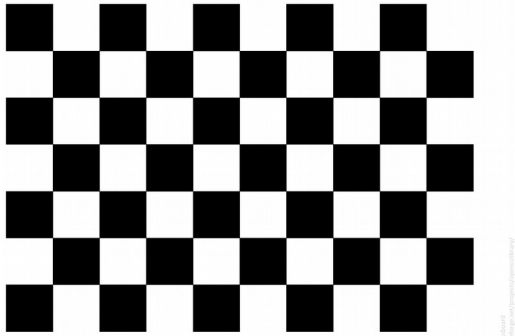
Parameters which are intrinsic as well but not always mentioned:
- radial and/or tangential distortion coordinates,
- skew parameters,
- dimensions of the sensor matrix,
- sensor cell size or aspect ratio of sensor height to width,
- scaling factor

Calibration Pattern

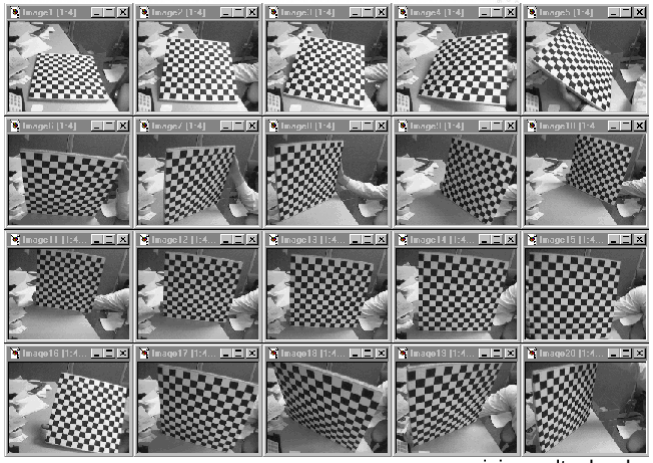
Early methods did rely on calibration pattern/objects, where points had to lie on different planes.



Today's standard camera calibration method uses a planar grid (e.g., a chessboard),



and several images of it, recorded at different orientations (covering at the end the entire image plane).

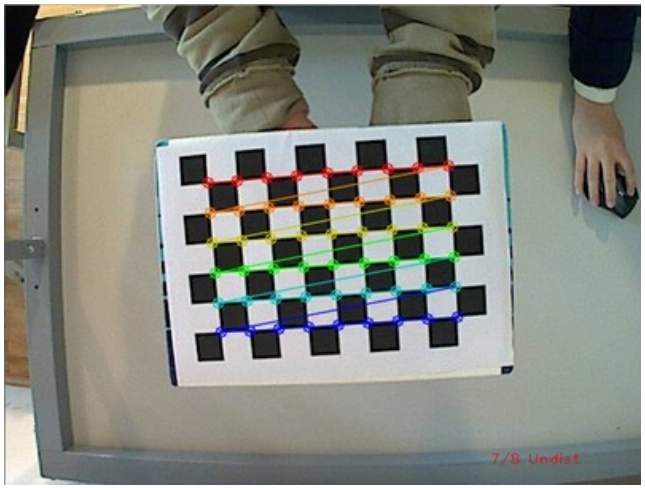


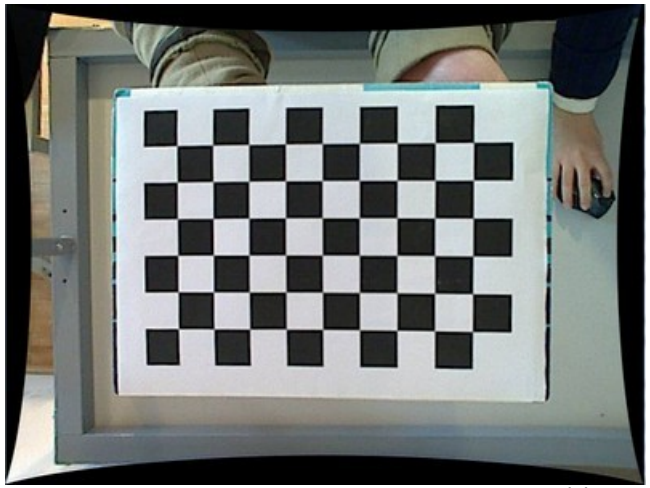
www.vision.caltech.edu

Recommendation:
Accuracy of the calibration depends on the accuracy of the measurements of the calibration pattern - its construction tolerances.
The calibration pattern should be built with tolerances one or two order of magnitudes smaller than the desired accuracy of calibration if desired accuracy of calibration is 0.1mm the calibration pattern should be built with tolerances smaller than 0.01 mm.

Zhang's Method

- This method was invented by Zhegyou Zhang (1999) @Microsoft Research
- Z. Zhang, "A flexible new technique for camera calibration," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, Nov. 2000.'' Zhang2000.pdf
- implemented in
  - OpenCV C/C++ library → calibrateCamera()
  - Matlab → estimateCameraParameters()
    https://www.mathworks.com/help/vision/ref/undistortimage.html





Courtesy: Matlab.com

Calibration Tools

Matlab computer vision system toolbox

https://se.mathworks.com/products/computer-vision/apps.html

https://se.mathworks.com/help/vision/ug/fisheye-calibration-basics.html

OCamCalib: Omnidirectional camera calibration toolbox

https://sites.google.com/site/scarabotix/ocamcalib-toolbox

GML

http://graphics.cs.msu.ru/en/node/909

Jean-Yves Bouguet's toolbox

http://www.vision.caltech.edu/bouguetj/calib_doc/

Kalibr

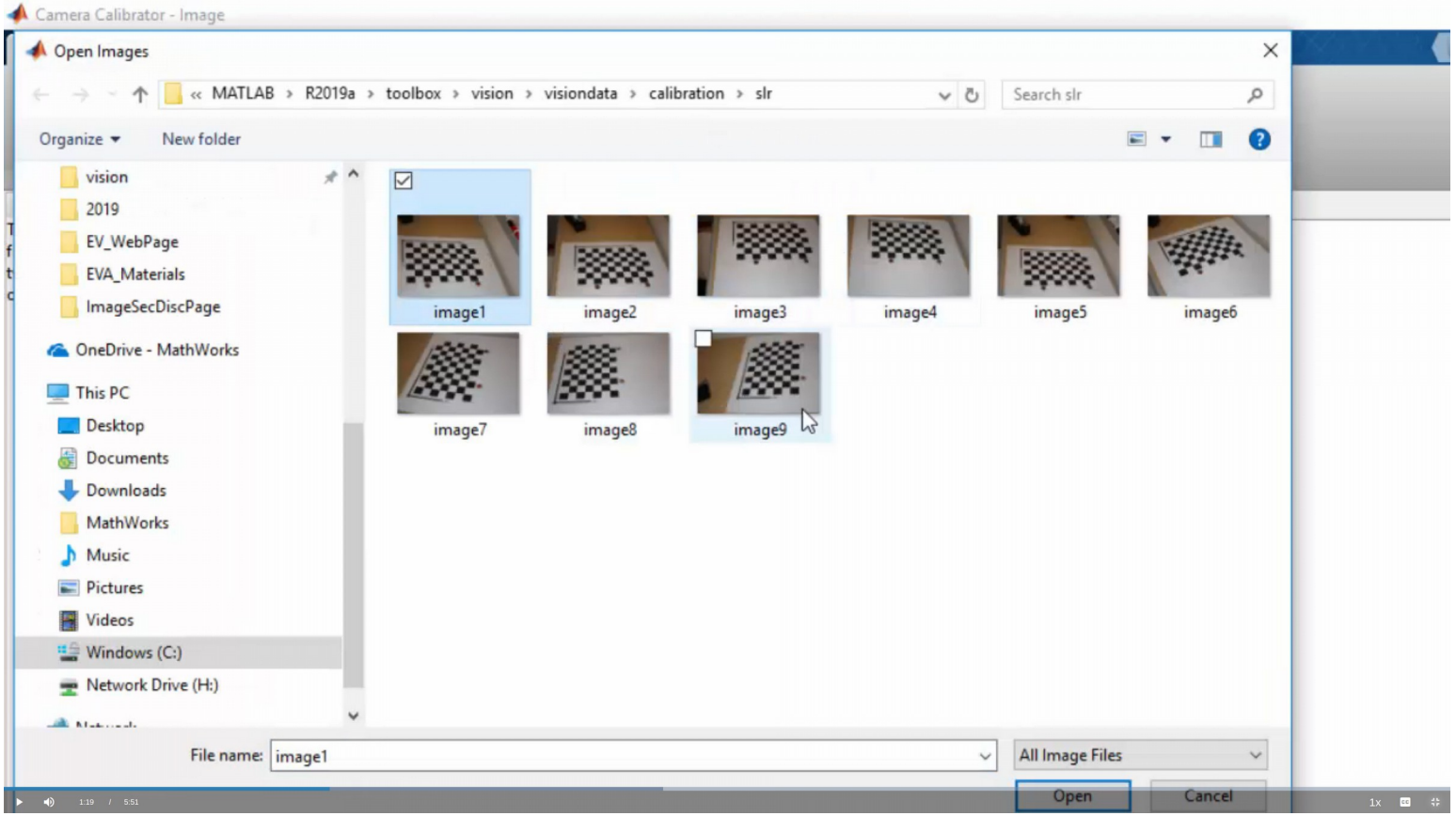
https://github.com/ethz-asl/kalibr/wiki/supported-models

Camera Calibration - Practical Steps

https://se.mathworks.com/videos/camera-calibration-with-matlab-81233.html

1. Take images of the checkerboard calibration pattern

2. Add the images into the calibration app



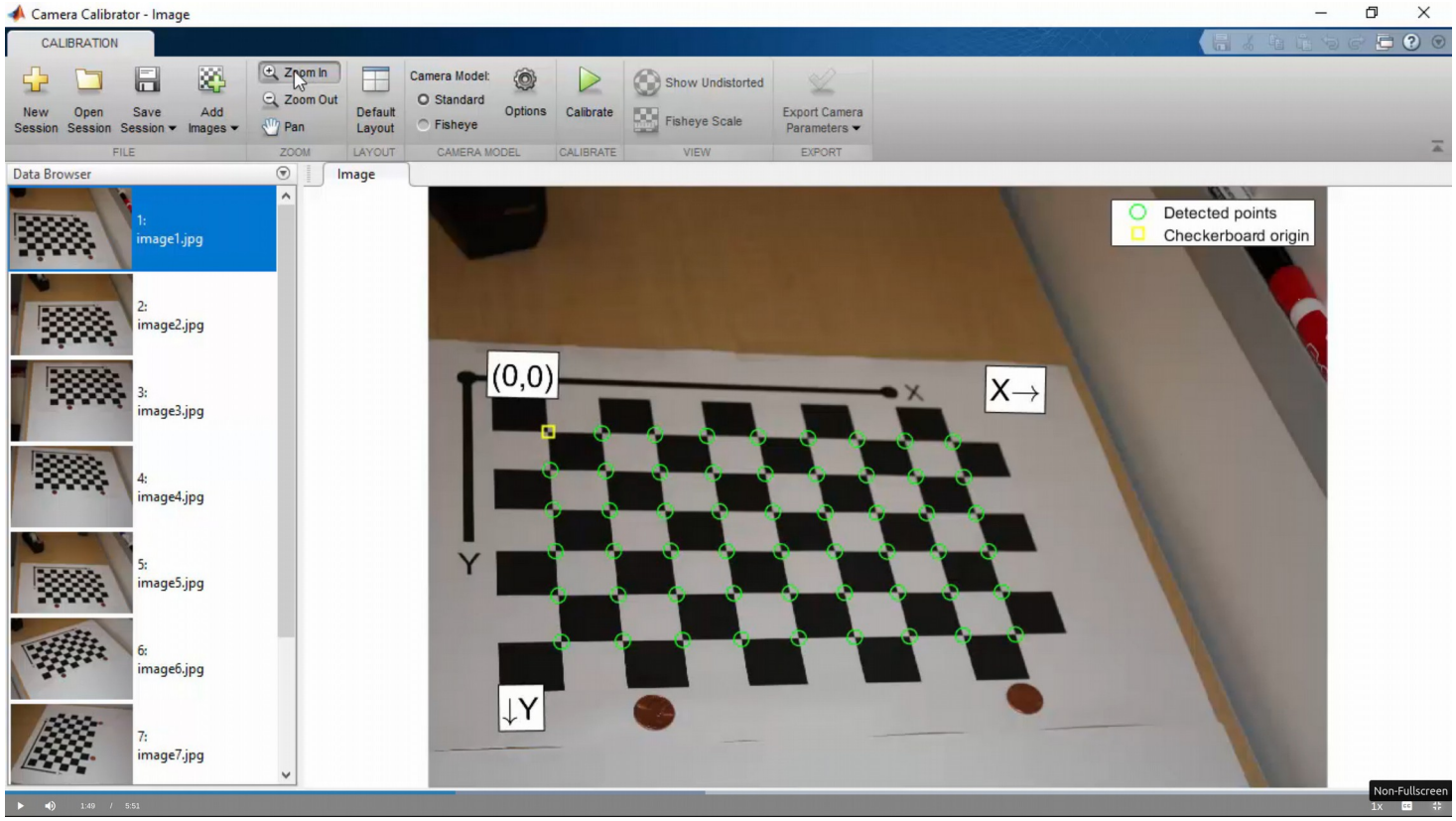
Camera Calibration - Practical Steps

https://se.mathworks.com/videos/camera-calibration-with-matlab-81233.html

3. Enter the size of the checkerboard square in world units (mm)

   → matching between world and image pixels

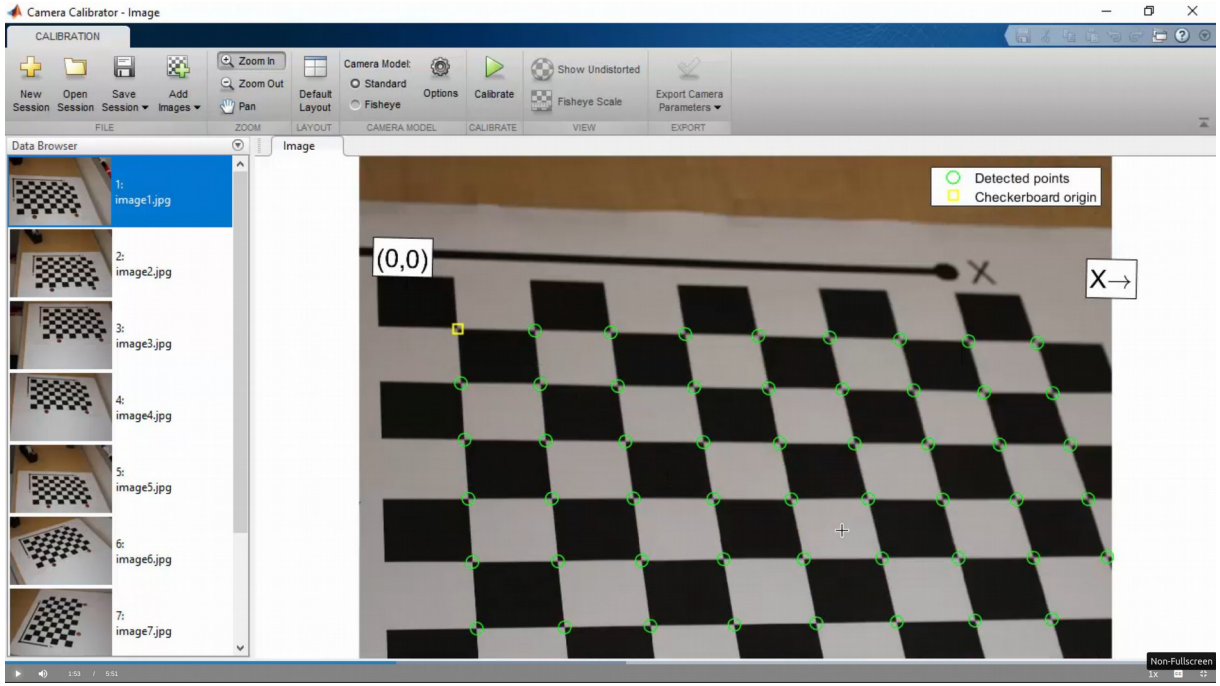
4. Chessboard detection: Detects corners of the checkerboard squares



Camera Calibration - Practical Steps

https://se.mathworks.com/videos/camera-calibration-with-matlab-81233.html

5. Inspect the correctness of the detection by zooming into the detection results



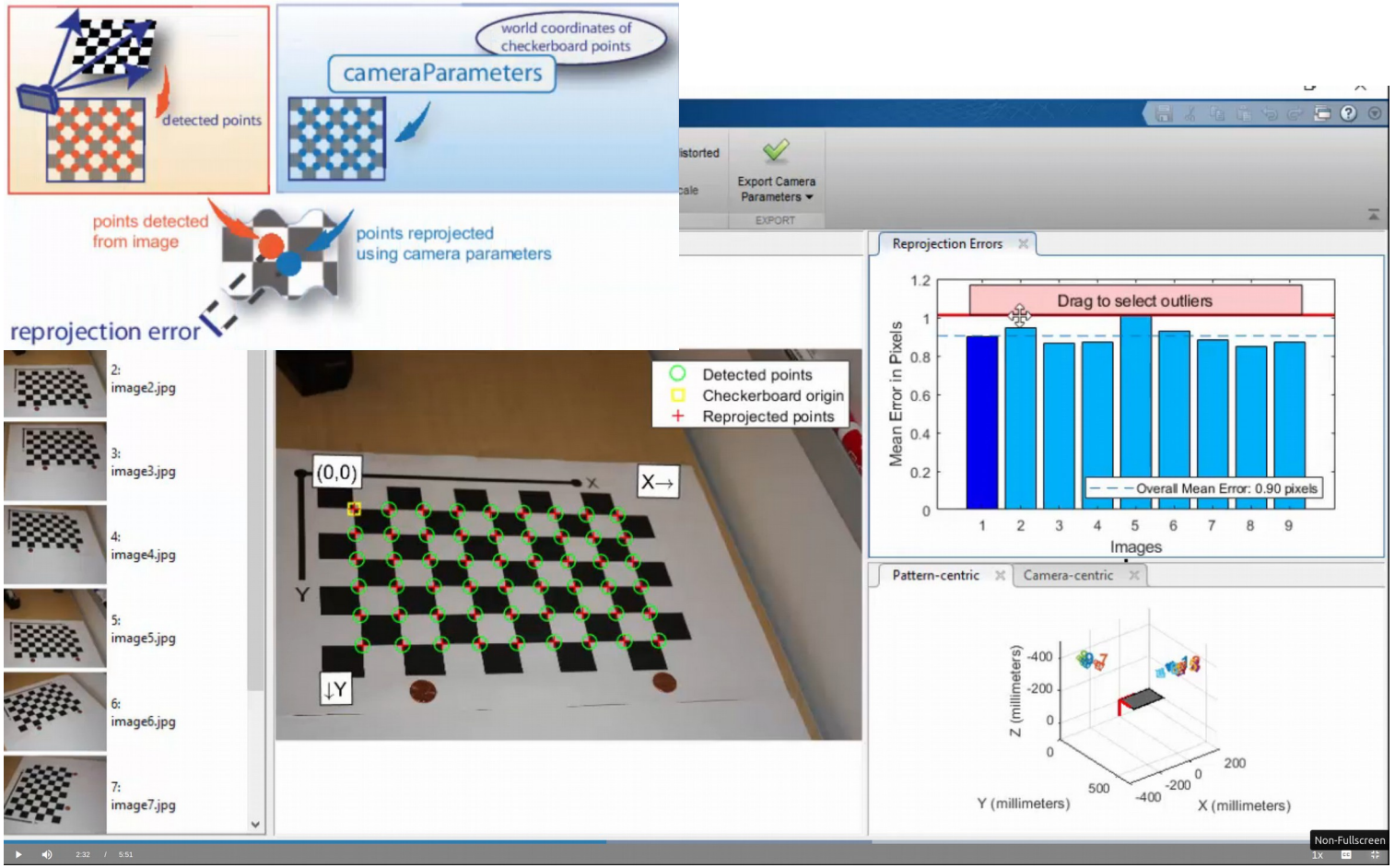
6. Choose what to estimate:

- Radial distortion
- Tangential distortion
- Skrew

Camera Calibration - Practical Steps

https://se.mathworks.com/videos/camera-calibration-with-matlab-81233.html

7. Calibrate and evaluate calibration results (evaluate the reprojection error)



8. Export camera parameters: Calibrated camera parameters are now computed.

Lecture 2 - Summary

- Camera Parameters
- Perspective Projection
- Calibration

  - Zhangs method using planar 3D-points

  - OpenCV demo

- Undistortion

Additional reading

- Soatto: Chapter 3
- Forsyth, D., and Ponce, J.: Computer Vision: A Modern Approach

  – chapter 1 Geometric Camera Models

  – chapter 22 Optimization Techniques (section 1-2)
- Szeliski Chapter 6.3
- Z. Zhang, "A flexible new technique for camera calibration," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, Nov. 2000.” Zhang2000.pdf