Univerza v Ljubljani
Fakulteta za elektrotehniko

Computer Vision
02 – Image formation, part 1

doc. dr. Janez Perš
(with contributions by prof. Stanislav Kovačič)

Laboratory for Machine Intelligence
Faculty of Electrical Engineering
University of Ljubljana

Announcements (1/2)

- Everyone present?
- “Studis” student administration system
  - https://studij.fe.uni-lj.si
  - Exam enrollment, official message distribution system, etc.
  - If you have not yet done so, **set up your account**!

- E-classroom
  - http://e.fe.uni-lj.si
  - Lecture slides, lab tasks
  - Slides will be available before each lecture this year

Announcements (2/2)

- Timetable
  - “Urnik” in Slovenian
  - http://urnik.fe.uni-lj.si/
  - Lecture times
  - Faculty-wide (for all classes)
- In general
  - If I promise something, but forget, do not hesistate to remind me, in person or via email, or after/before classes
  - For example: If lecture slides are not up, etc.
  - Office hours by appointment (email)

Quick recap of the previous week

- What exactly is computer vision
  - The field which aims to provide sight to computers
  - The task: generate description of the scene from image/video
- Where/how did it start?
  - As part of the field of artificial intelligence, in 60s
- Where it is used?
  - Machine vision (industrial)
  - Robot vision (experimental-research)
  - Visual searches, photo database indexing, etc.

Outline

- Image formation (from 3D to 2D)
- Basics of camera modeling
- Direct Linear Transform (DLT)
- Camera calibration
- Reconstruction – back from 2D to 3D
- Camera modeling revisited
- Lens distortion

Image formation (1/5)

- Image formation is quite complex process
- There are many things to consider:
  - Spatial / geometric properties
    - Light, object, and camera themselves
    - Light-object-camera arrangement
  - Radiometric / Photometric properties of light sources, objects, and sensors.
    - Propagation of light (energy) and its effects
    - Remember: no light (energy), no image
    - In exceptional cases, objects radiate light, no external source needed (FLIR)
- The final result is an image

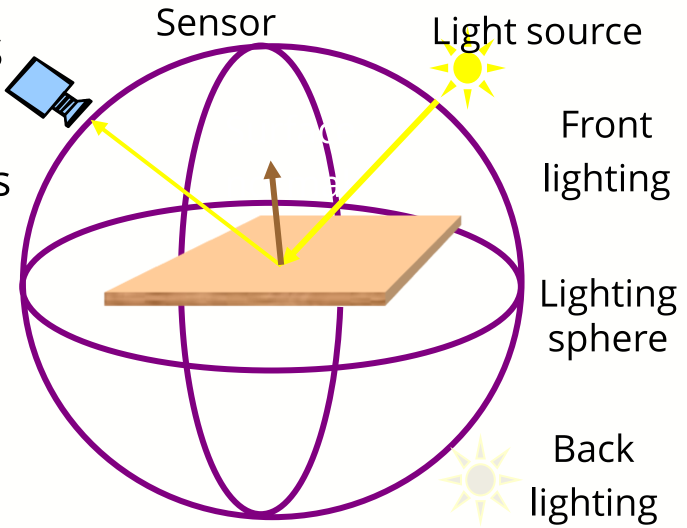
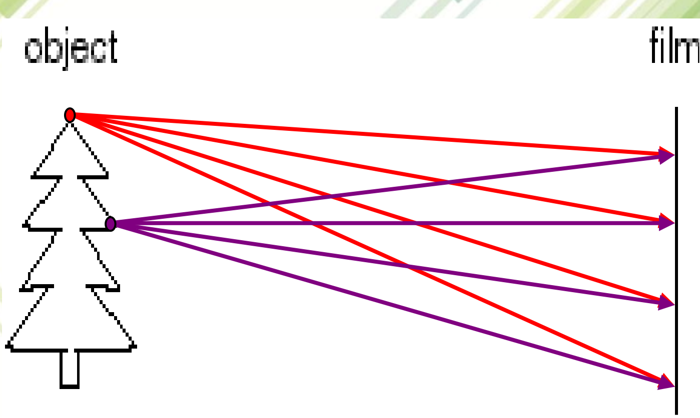
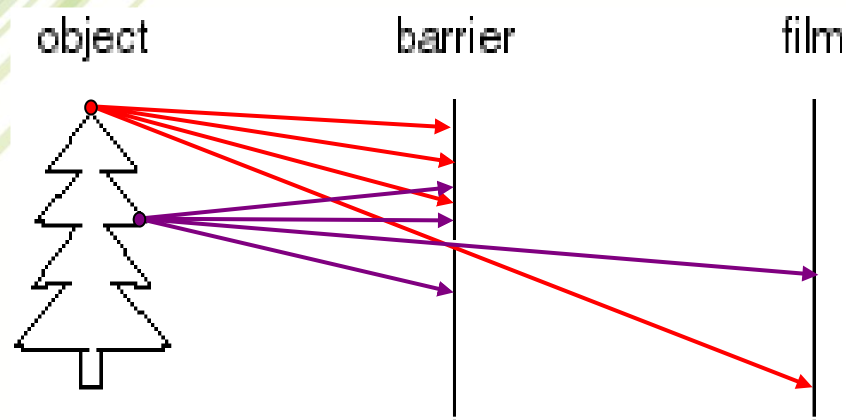


Image formation (2/5)

- We are interested in mapping from 3D world to 2D image plane: 3D → 2D





Slide source: Seitz

Wrong!

Right!

Image formation (3/5)

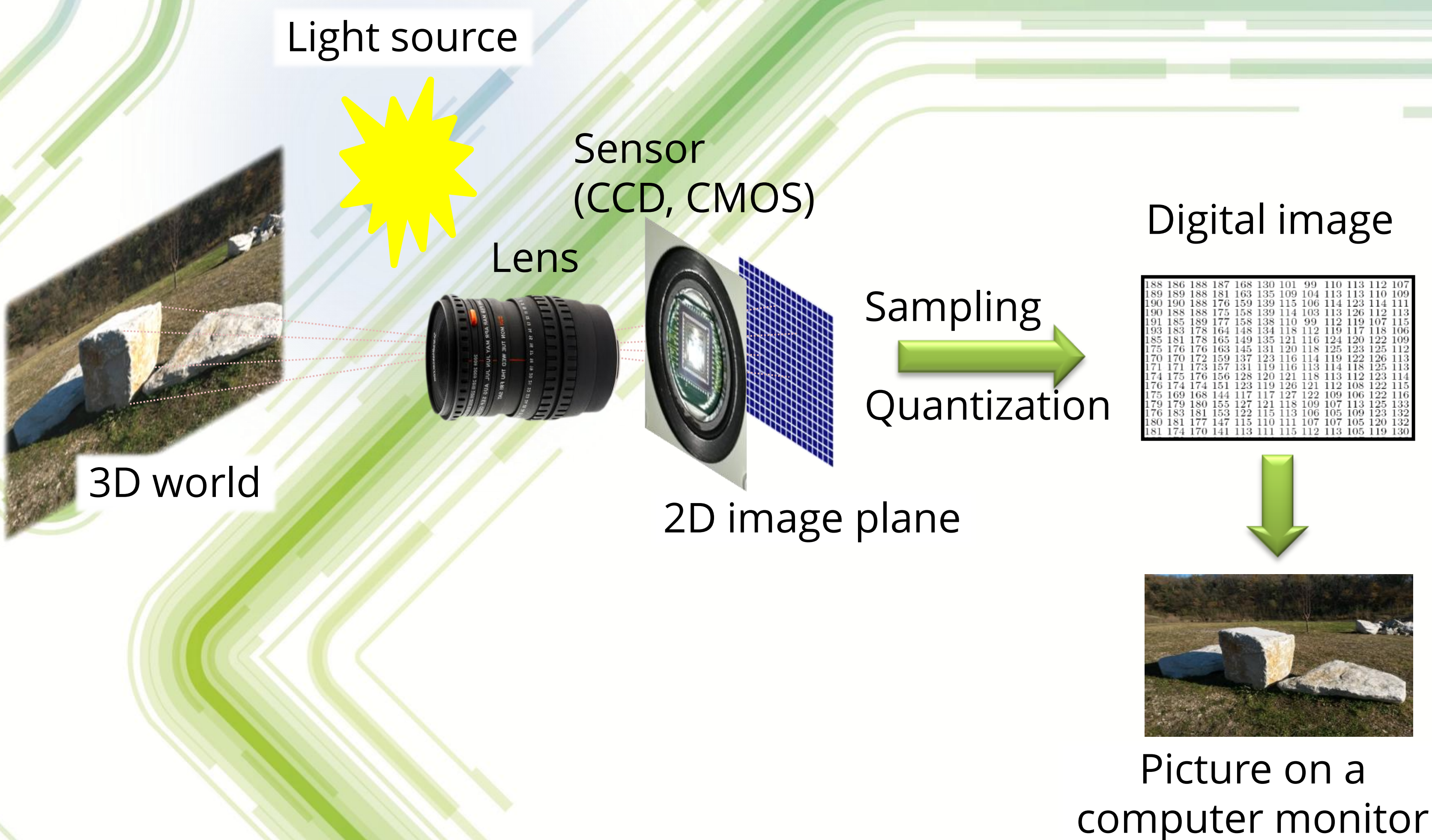


Image formation (4/5)

- Image:
  - a mapping of a 3D scene (world) onto a 2D plane.
- Digital image:
  - Sampled & quantized 2D image plane
  - Made of discrete picture elements – *pixels*
- Therefore, a digital image
  - Just an array (a matrix) of numbers.
  - 2 D matrix of brightness values for grayscale images
  - 3D array of R/G/B brightness values for RGB color images

Image formation (5/5)

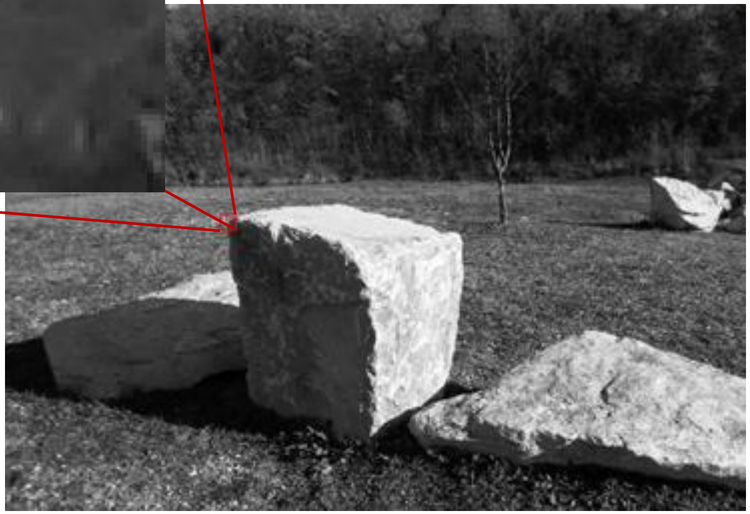
Patch (8 x 8 pixels)

92	125	145	147	148	140	137	129
121	152	159	157	147	135	130	140
139	150	152	184	144	106	114	130
147	154	171	222	195	170	115	112
167	201	197	198	186	143	88	87
172	195	169	158	175	130	95	120
180	153	155	164	177	127	133	171
154	136	145	145	145	123	149	173

Pixel value: 8 bits
0: completely black
255: completely white

Patch
(64 x 64 pixels)

Whole image
3264 x 2448 pixels



The CV task:

from an image or a set of 2D images derive (quantitative, qualitative) description of 3D world
e.g. *object detection, localization, recognition, 3D reconstruction*

Notation convention for images

- Mathematically: *f, f(x, y), f(i, j), I(i, j), ….*
  - 2D function, continuous or discrete.
  - Could be also a function of time, *f(x, y, t), f(i, j, k)*, …
- In the field of signal processing
  - image *f* is treated as 2D signal
  - Could be static (single image) *f(x,y)*, or time-varying, spatio-temporal signal, *f(x, y, t)*, ….
- Programmatically: *I, F,* …
  - 2D array/matrix of suitable type (logical, uint8, uint16, int16, single, double,…) and size (no. of rows x no. of colomns)

Basic image types

- What type of image is this?

By David Adam Kess (Own work) [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)], via Wikimedia Commons

Basic image types

- Black and white (or binary) image
  - an array of zeros and ones.
- Grayscale (or grayvalue, or intensity) image
  - an array of numbers of proper type (e.g. uint8)
- Color image - RGB, HSV, YUV …
  - RGB most commonly used in computer vision
  - R = Red, G = Green, B = Blue !
  - Could be treated as three 2D arrays for R, G, B color channel each.
  - Or alternatively, could be treated as 2D array of 3D vectors with R,G,B components .

Basic image types

- Conversion from RGB color to grayscale
  - Often used as first image processing step
  - When using algorithms, developed for grayscale images

C(x,y,c)

G = rgb2gray(C)

G(x,y)

Basic image concepts & properties

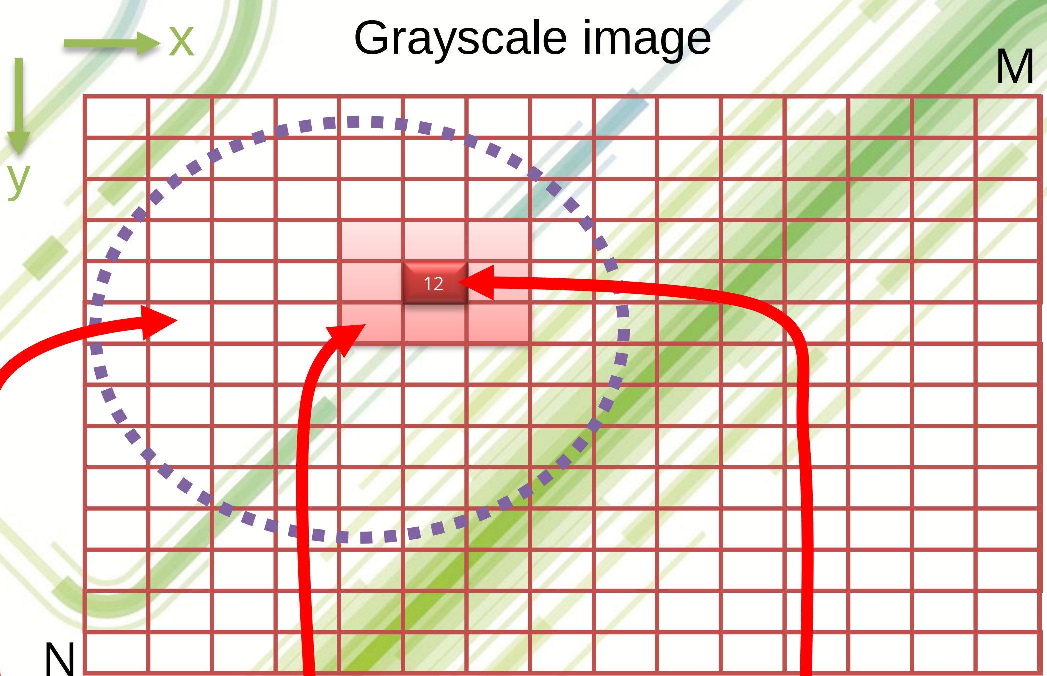


Image size

No. of rows x No. of columns
N x M, e.g. 1080 x 1920

OR

No. of pixels in X direction x
No. of pixels in Y direction
M x N, e.g. 1920 x 1080

Region
of interest
(ROI)

Neighboorhood
(or window)
3x3, 5x5, …

Pixel 12

Pixel
In the case of
color image

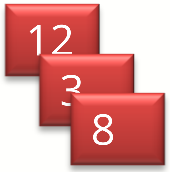


Image coordinates

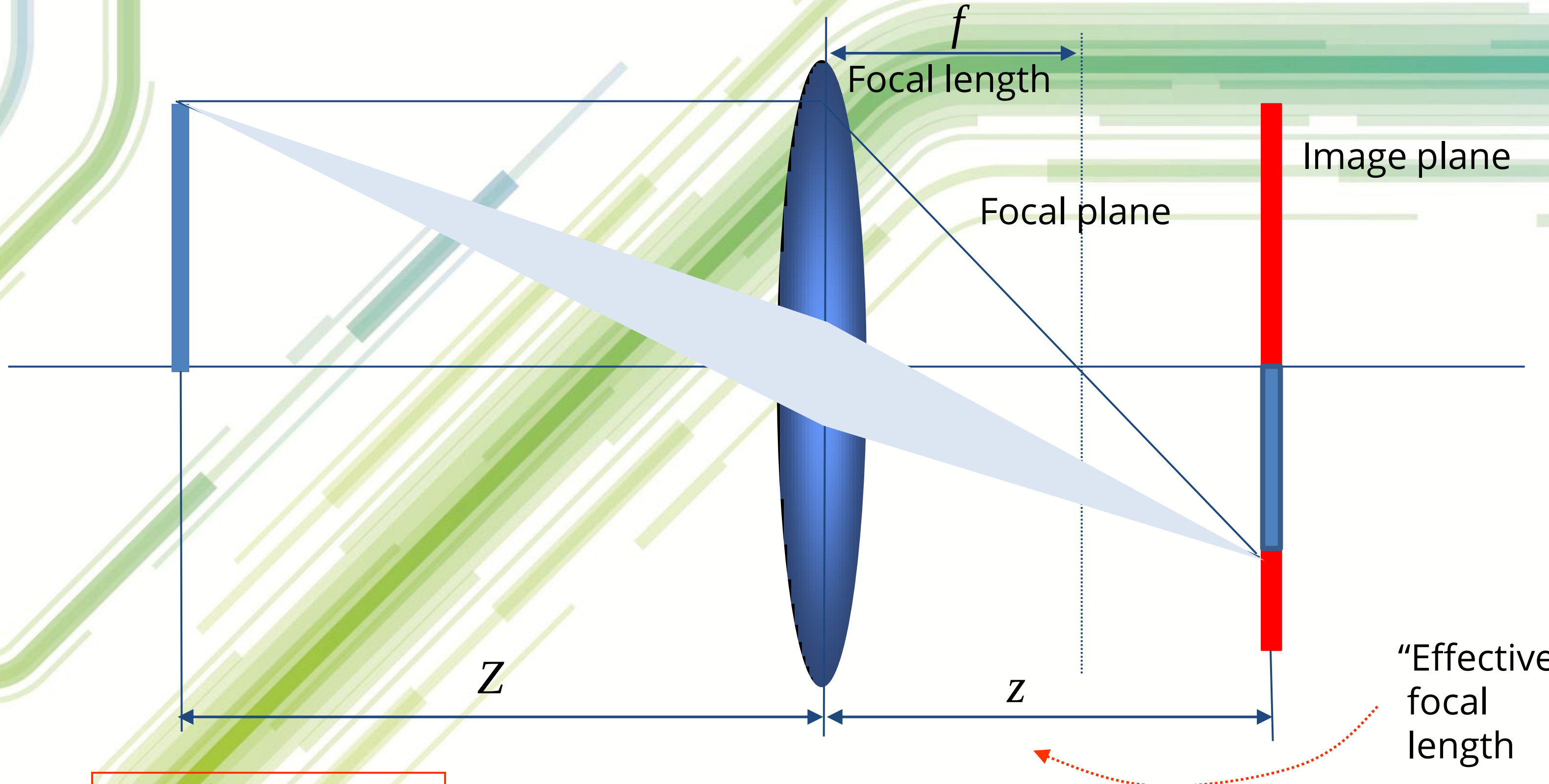
- Most often
  - we place the origin of the image coordinate system in the upper left corner
  - sometimes in the lower left corner.
- Sometimes we place the origin somewhere in the middle of the image.
- This often hidden in high levelfunctions that operate on images.
  - e.g. the example of *B=rgb2gray(A)*
  - But when you need to address individual pixels, it becomes important.

Image coordinates in Matlab

- We are used to think of images as functions of x and y
  - X is running horizontally
  - Y goes down (or sometimes up) vertically.
  - Therefore, *I(y,x),*
- But, specifically in Matlab
  - images are 2D arrays (or 3D arrays for color images, with 3rd dimension for color)
  - Arrays (matrices) are represented as rows and colums *I(row, column)*, row first, then column
  - Reason: Matlab was developed by mathematicians, uses mathematical matrix notation

Questions?

Thin lens equation



$$\boxed{\frac{1}{Z}+\frac{1}{z}=\frac{1}{f}}$$

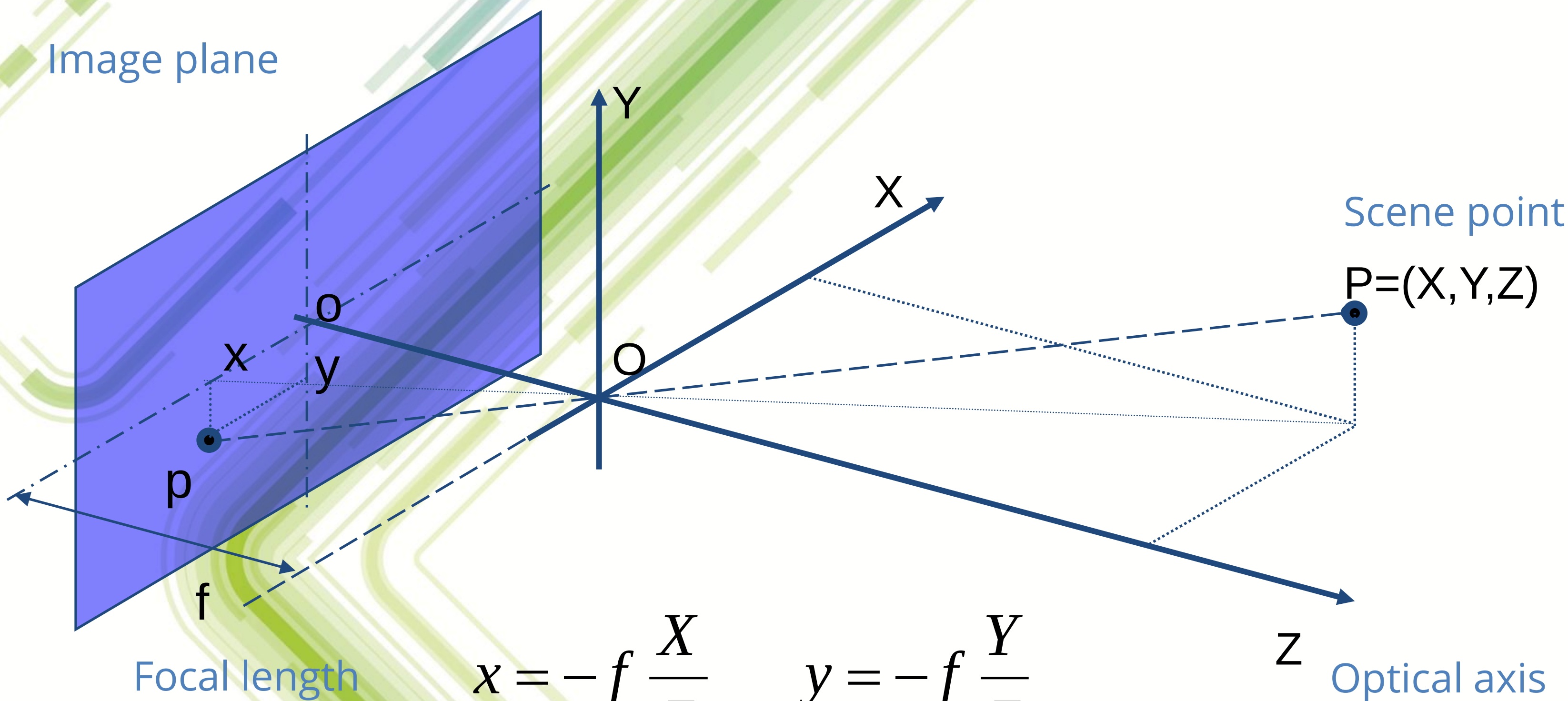
$$Z >> f \Rightarrow z \approx f$$

In all practical cases, image plane is in or very close to focal plane

For convex lens, *f* is always positive

Geometric camera model

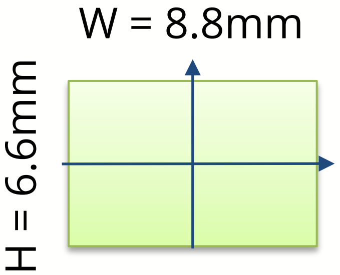
- Perspective projection
  - pin-hole camera model



$$x = -f\,\frac{X}{Z} \qquad y = -f\,\frac{Y}{Z}$$

A simple example

- Task: Calculate image point position
  - Camera lens: f = 16 mm (0.016m)
  - Scene point ("object") distance, Z = 1 m
  - X and Y positions of the scene point
    X = 20 cm (0.2 m), Y = 15 cm (0.15 m)



$$x = -f\frac{X}{Z} = -0.016\frac{0.2}{1} = -0.0032\,\mathrm{m} = -3.2\,\mathrm{mm}$$

$$y = -f\frac{Y}{Z} = -0.016\frac{0.15}{1} = -0.0024\,\mathrm{m} = -2.4\,\mathrm{mm}$$

Of course, image sensor should be large enough

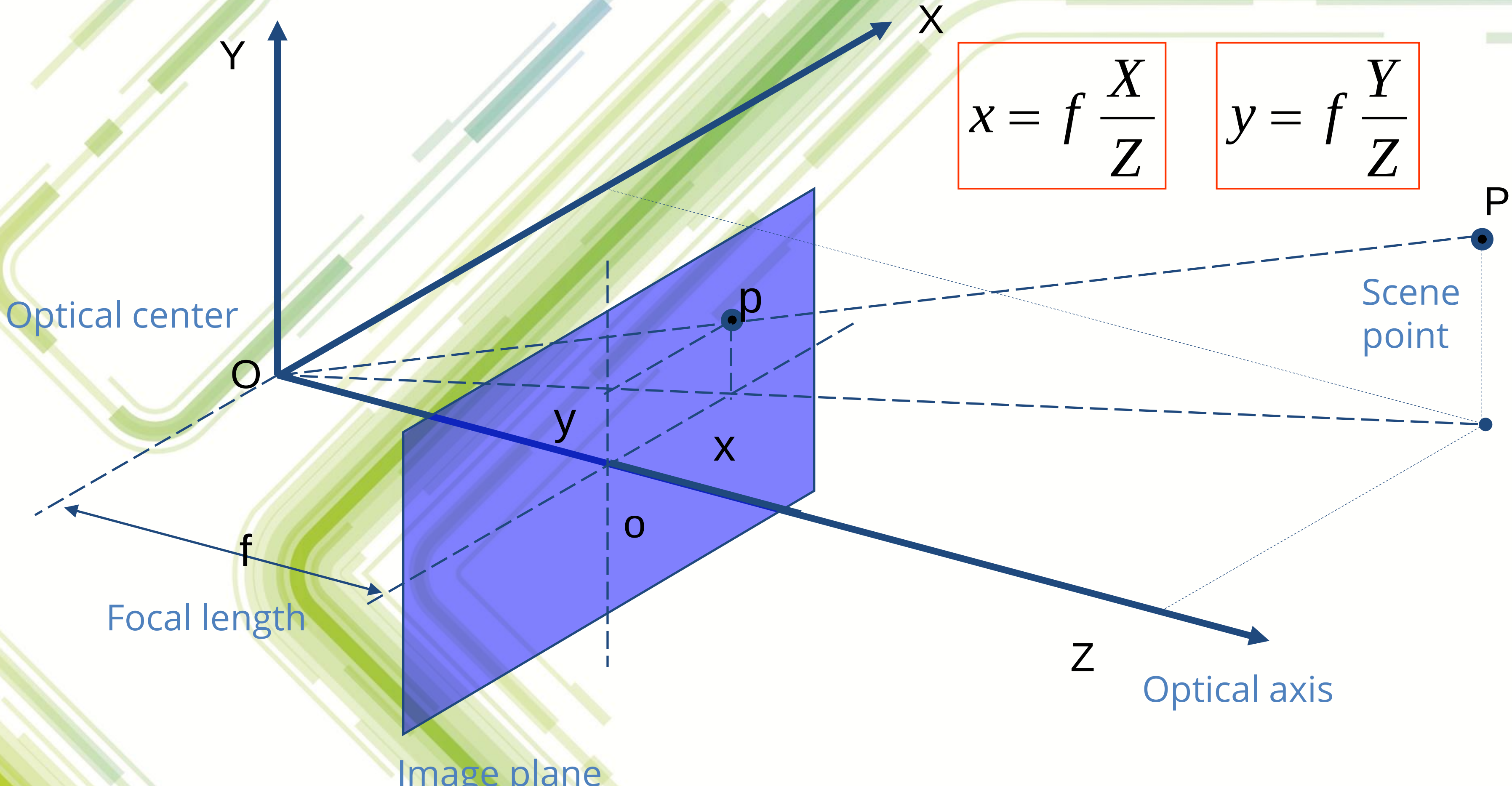
The same holds for the field of view (FOV) of the lens.

2/3" sensor would be appropriate, aspect ratio 3:4, diagonal 11mm

Geometric camera model

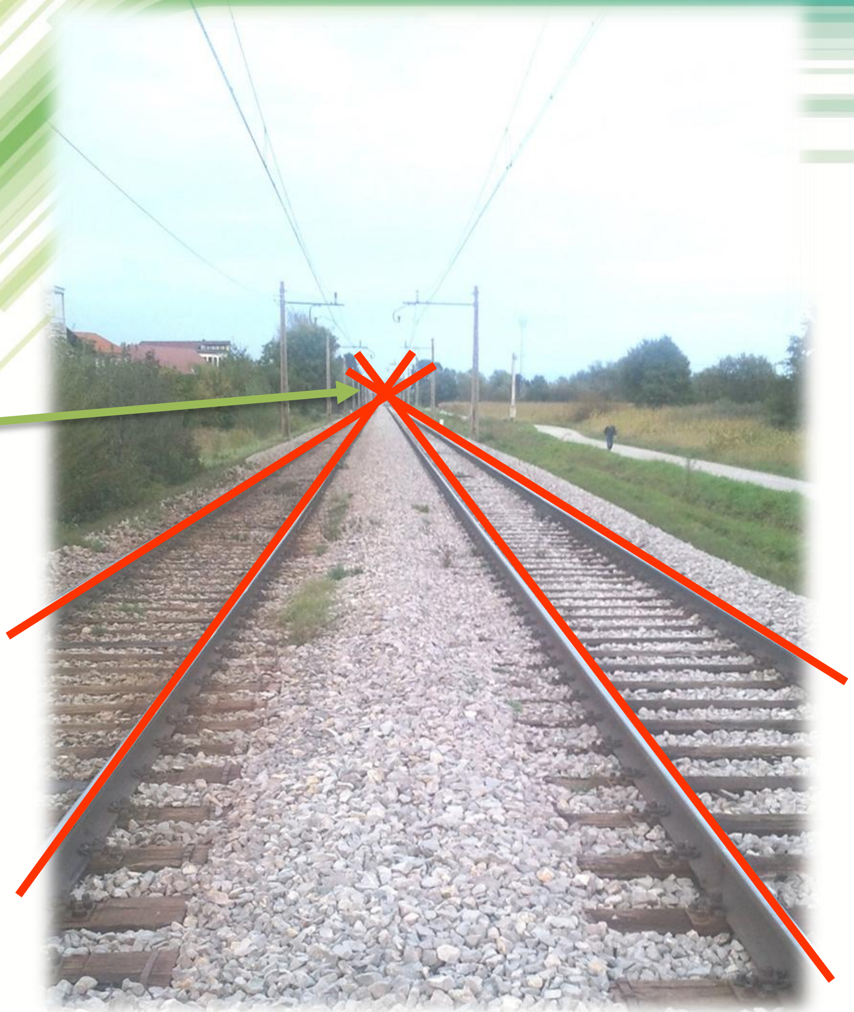
We move center of projection back, behind the image plane!

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



Perspective projection – basic properties

- Straight lines remain straight
- Parallel lines meet in the vanishing point

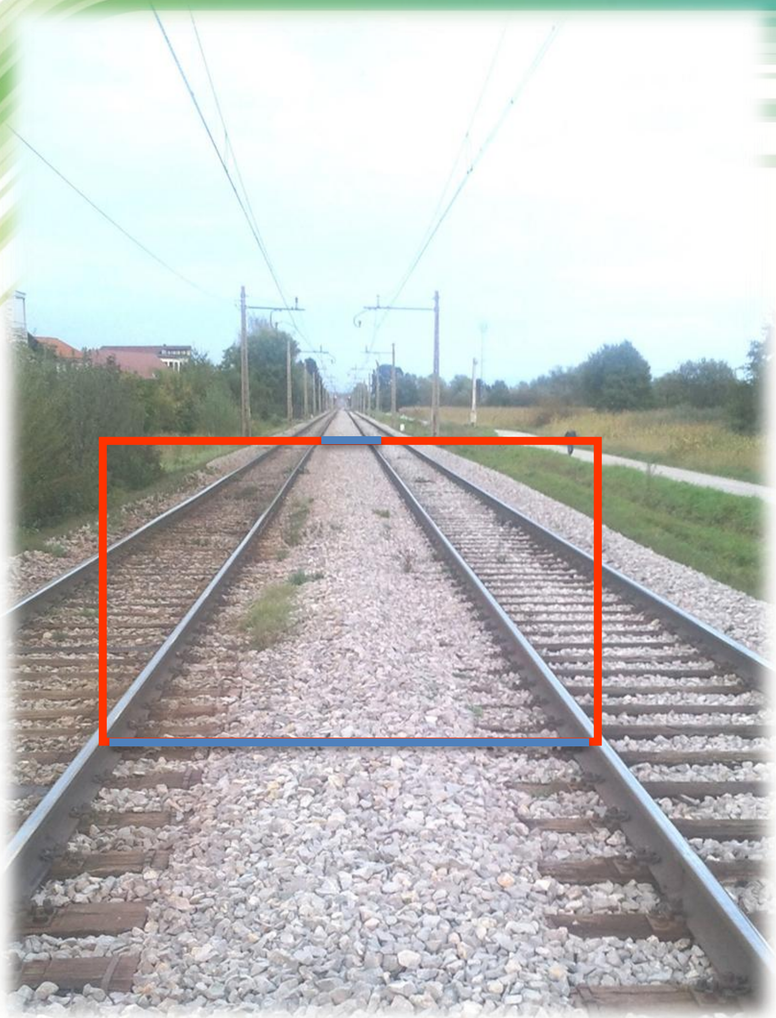


Perspective projection – basic properties

- Angles are NOT preserved!

Perspective projection – basic properties

- Lenghts / distances are NOT preserved.
- They (of course) depend on the distance from the camera.

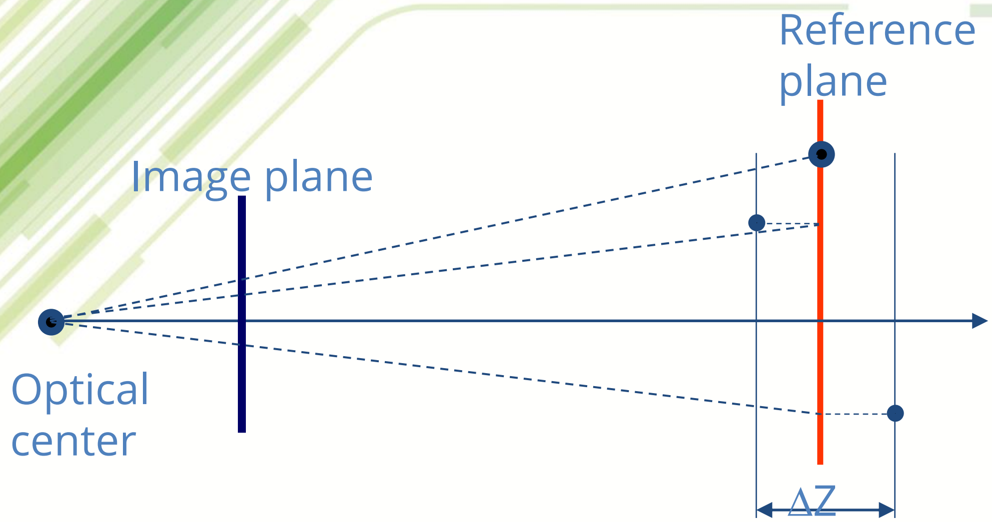


Geometric camera model

- Weak perspective model

$$x = \frac{f}{Z} X = mX$$

$$y = \frac{f}{Z} Y = mY$$



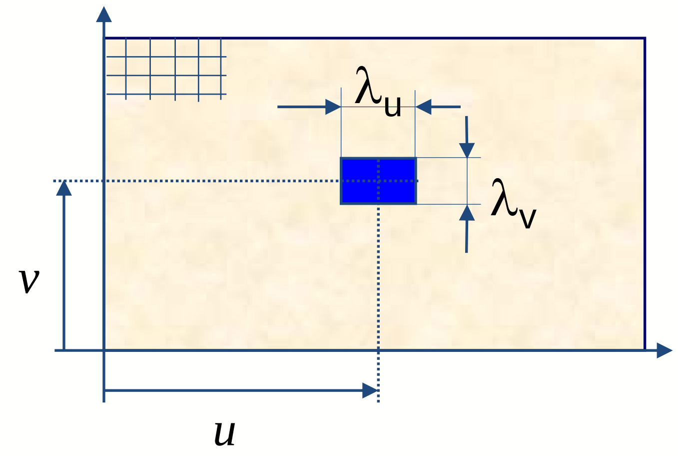
  - Object thickness much smaller than the distance to the object, $\Delta Z << Z$.
  - Valid for thin planar objects (essentially 2D scenes).
  - Orthographic projection (m=1) plus scaling (m < 1)

Geometric camera model

- Digital image / picture consists of array of picture elements – pixels.
  - Image (pixel) coordinates are given in pixels, (u, v).

$$u = \frac{x}{\lambda_u} = \frac{f}{\lambda_u}\frac{X}{Z} = f_u \frac{X}{Z} = \frac{f_u}{Z} X = m_u X$$

$$v = \frac{y}{\lambda_v} = \frac{f}{\lambda_v}\frac{Y}{Z} = f_v \frac{Y}{Z} = \frac{f_v}{Z} Y = m_v Y$$

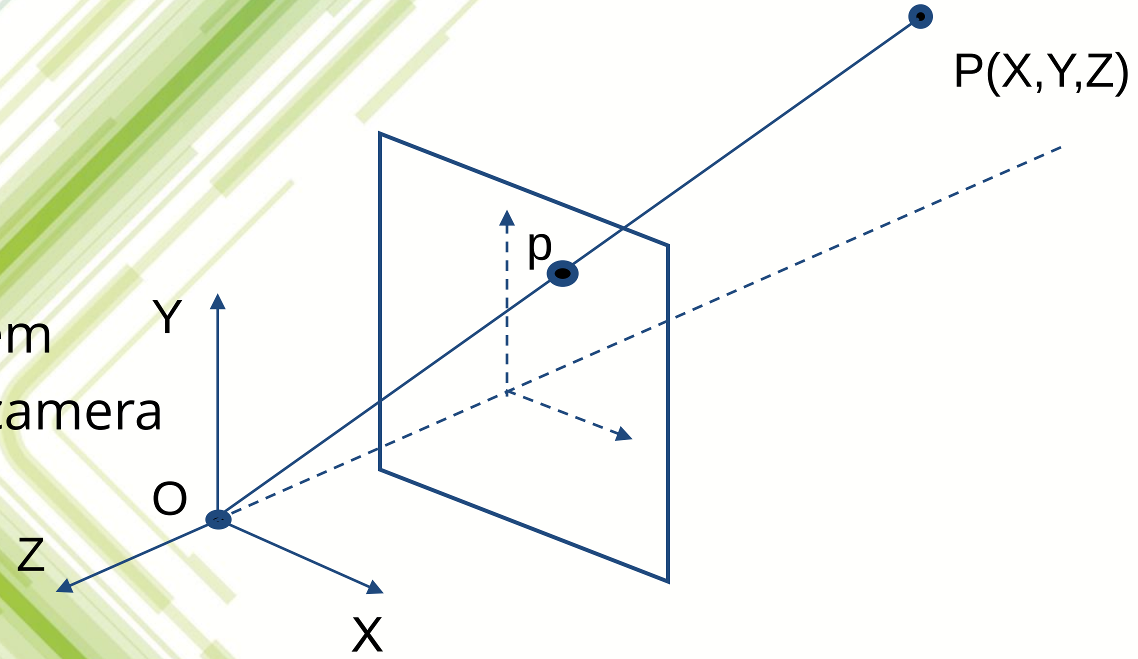


- In many practical machine vision situations it is sufficient to find λ_u , λ_v (or f_u , f_v or m_u , m_v)

Geometric camera model

- Position (X, Y, Z) of point P is given in camera coordinate system.
  - Camera coordinate system is not known or, directly accessible).

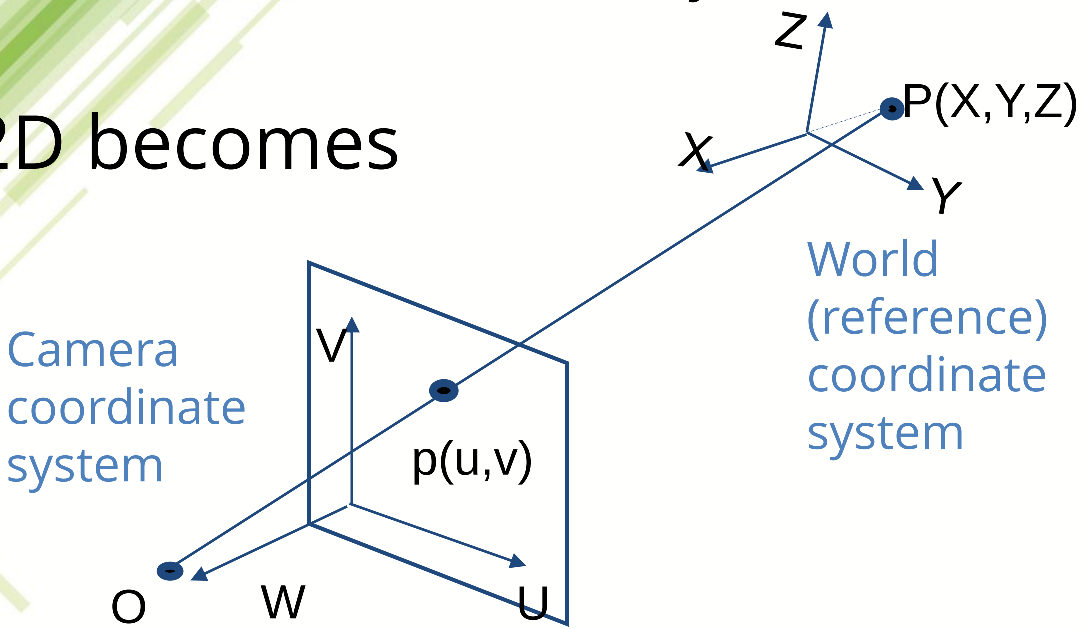
Coordinate system attached to the camera



Geometric camera model

- We prefer to define P in the world coordinate system instead
  - World coordinates could be defined by walls, object, robot…
  - Then, the coordinates can be measured directly

- Mapping 3D →2D becomes
  - Translate
  - Rotate
  - Project



Parameters of the camera model (1/2)

- Camera parameters
  - Extrinsic (“external”)
  - Intrinsic (“internal”)
- Extrinsic
  - Camera position
  - Camera orientation (pose)
  - …with respect to the world coordinate system!
- Intrinsic
  - parameters that relate camera coordinate system with image (pixel) coordinates
  - Depend on the whole optical system: camera + lens!

Parameters of the camera model (2/2)

- Altogether 11 parameters
  - Translation (3)
  - Rotation (3)
  - Focal length (1)
  - Image center (2)
  - Pixel size (2)
- So far we have only linear model!
- To model nonlinear lens distortions
  - We add distortion parameters
  - Usually 1-5 parameters, depending on the model

How do we obtain camera parameters?

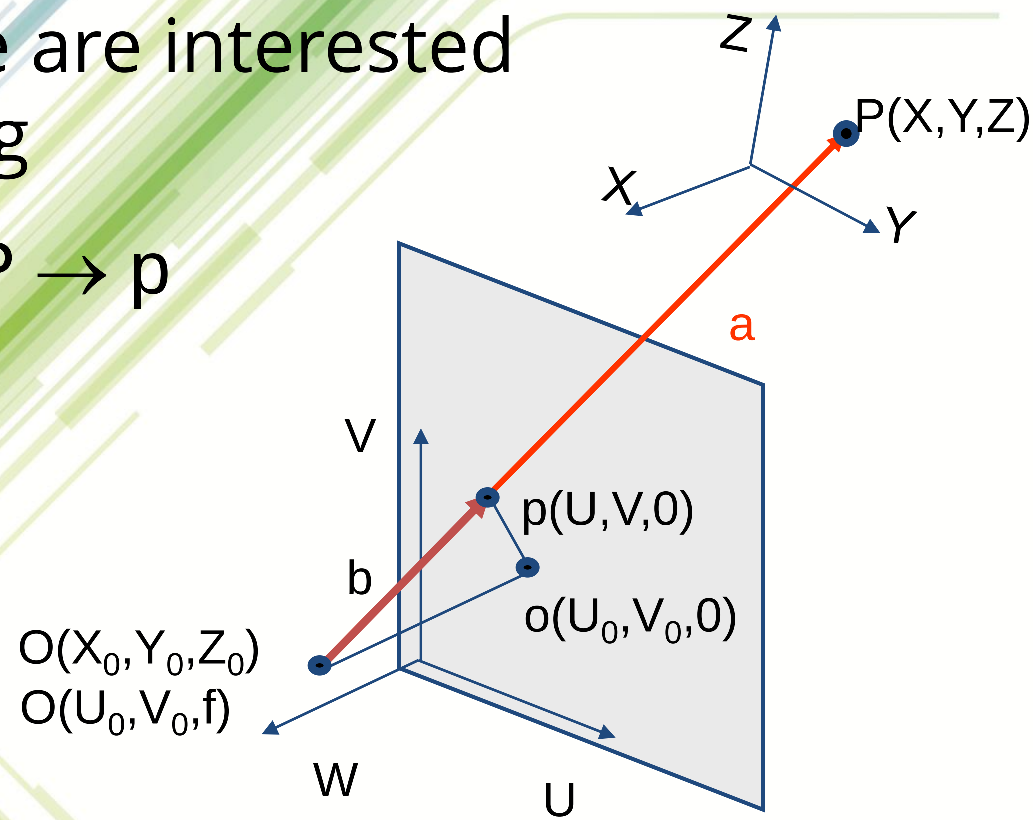
- Camera calibration
  - Target (“real world scene”) with known real world point coordinates
  - Image of the target with, where we can detect real world points i
- We have the model
  - So we only need to calculate the parameters
  - But it is not as simple as it seems
- In case of lens distortion
  - It gets even more difficult

Questions?

Direct linear transform (DLT)

- Abdel-Aziz & Carara, 1971
- In principle we are interested in the mapping

$T: \Re^3 \rightarrow \Re^2, \quad P \rightarrow p$



Direct linear transform (DLT)

$p = o + b$

$P = O + a$

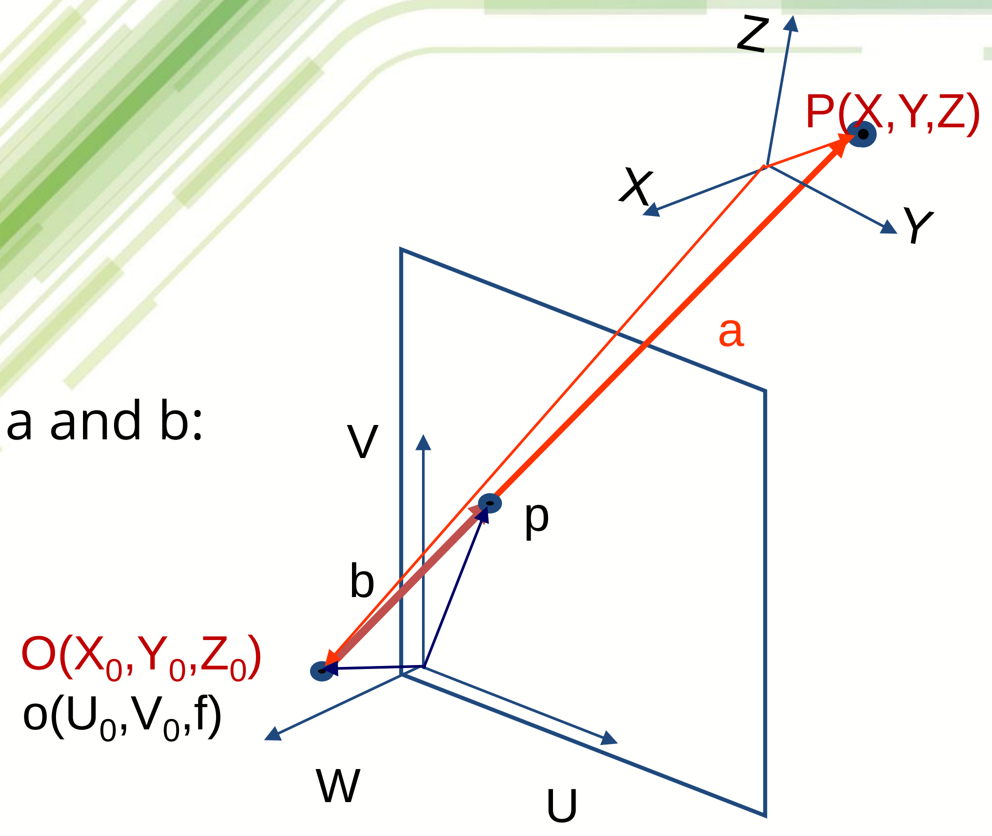
$b = p – o$

$a = P – O$

Due to colinearity of vectors a and b:

$b = c\,a$, $c>0$ (scalar!)

$p - o = c(P - O)$



Direct linear transform (DLT)

$b = c\,a$, $c > 0$ (skalar)

We express b in camera c.s. $b(U_b,V_b,W_b)$

We express a in world c.s. $a(X_a,Y_a,Z_a)$

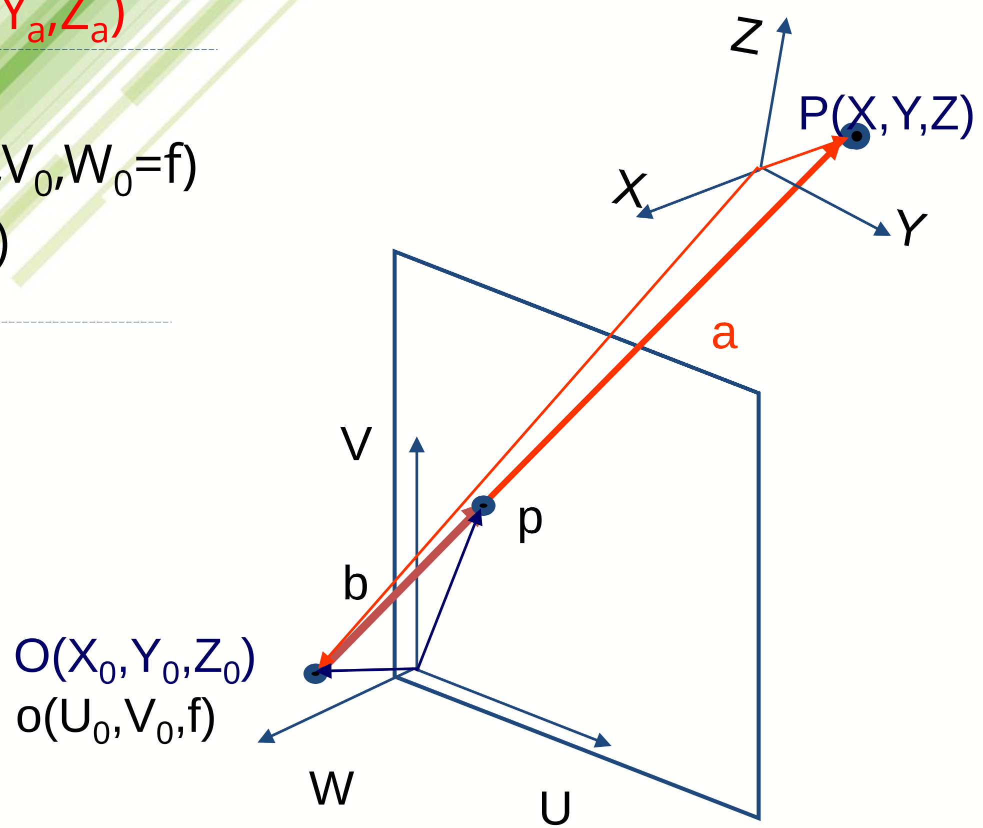
$b(U_b,V_b,W_b) = p(U,V,W=0) – o(U_0,V_0,W_0=f)$

$a(X_a,Y_a,Z_a) = P(X,Y,Z) – O(X_0,Y_0,Z_0)$

$b\,(U_b,V_b,W_b) = c\,R\,a(X_a,Y_a,Z_a)$

R = rotation matrix

$p–o = c\,R\,(P–O)$



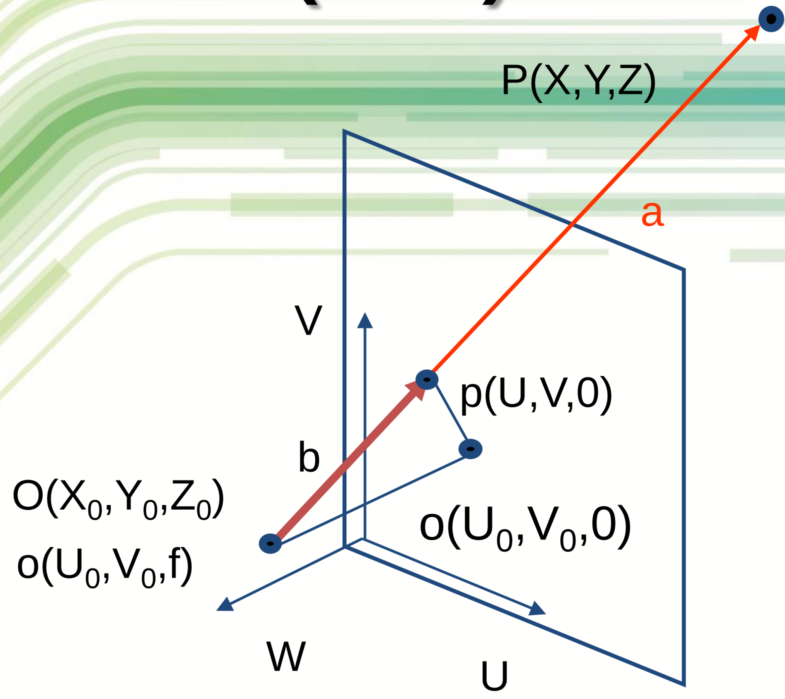
Direct linear transform (DLT)

- Let us write

  p–o = c R (P–O)

- in component form:

$$\begin{bmatrix} U-U_0 \\ V-V_0 \\ -f \end{bmatrix} = c \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X-X_0 \\ Y-Y_0 \\ Z-Z_0 \end{bmatrix}$$



$$\begin{aligned} U-U_0 &= c[r_{11}(X-X_0)+r_{12}(Y-Y_0)+r_{13}(Z-Z_0)] \\ V-V_0 &= c[r_{21}(X-X_0)+r_{22}(Y-Y_0)+r_{23}(Z-Z_0)] \\ -f &= c[r_{31}(X-X_0)+r_{32}(Y-Y_0)+r_{33}(Z-Z_0)] \end{aligned}$$

$$c = \frac{-f}{r_{31}(X-X_0)+r_{32}(Y-Y_0)+r_{33}(Z-Z_0)}$$

Direct linear transform (DLT)

- Insert *c* into first two equations and we obtain

$$U - U_0 = -f\,\frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)}$$

$$V - V_0 = -f\,\frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)}$$

- Apply discretization of image coordinates

$$U - U_0 = \lambda_u (u - u_0)$$

$$V - V_0 = \lambda_v (v - v_0)$$

(u_0, v_0) = point where optical axis goes through the image plane

(λ_u, λ_v) are pixel sizes (effectively: scaling factors) in u/v direction

In theory different, in practice (almost) the same

Direct linear transform (DLT)

- Let's expand

$$u - u_0 = -\frac{f}{\lambda_u}\frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)}$$

$$v - v_0 = -\frac{f}{\lambda_v}\frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)}$$

- Conventional, more concise notation:

$$\boxed{u = \frac{L_1 X + L_2 Y + L_3 Z + L_4}{L_9 X + L_{10} Y + L_{11} Z + 1}} \qquad \boxed{v = \frac{L_5 X + L_6 Y + L_7 Z + L_8}{L_9 X + L_{10} Y + L_{11} Z + 1}}$$

- L_i (i = 1,2,…,11) are so-called DLT parameters
  - They are not camera parameters, but related!

DLT parameters

$$L_1 = \frac{u_0 r_{31} - f_u r_{11}}{D}, \quad L_2 = \frac{u_0 r_{32} - f_u r_{12}}{D}, \quad L_3 = \frac{u_0 r_{33} - f_u r_{13}}{D}$$

$$L_4 = \frac{\left(f_u r_{11} - u_0 r_{31}\right) X_0 + \left(f_u r_{12} - u_0 r_{32}\right) Y_0 + \left(f_u r_{13} - u_0 r_{33}\right) Z_0}{D}$$

$$L_5 = \frac{v_0 r_{31} - f_v r_{21}}{D}, \quad L_6 = \frac{v_0 r_{32} - f_v r_{22}}{D}, \quad L_7 = \frac{v_0 r_{33} - f_v r_{23}}{D}$$

$$L_8 = \frac{\left(f_v r_{21} - v_0 r_{31}\right) X_0 + \left(f_v r_{22} - v_0 r_{32}\right) Y_0 + \left(f_v r_{23} - v_0 r_{33}\right) Z_0}{D}$$

$$L_9 = \frac{r_{31}}{D}, \quad L_{10} = \frac{r_{32}}{D}, \quad L_{11} = \frac{r_{33}}{D} \qquad f_u = \frac{f}{\lambda_u}, \quad f_v = \frac{f}{\lambda_v}$$

$$D = -(X_0 r_{31} + Y_0 r_{32} + Z_0 r_{33})$$

Example: deriving u

$$u - u_0 = -\frac{f_u r_{11} X + f_u r_{12} Y + f_u r_{13} Z - f_u r_{11} X_0 - f_u r_{12} Y_0 - f_u r_{13} Z_0}{r_{31} X + r_{32} Y + r_{33} Z - (r_{31} X_0 + r_{32} Y_0 + r_{33} Z_0)}$$

$$u - u_0 = -\frac{\frac{f_u r_{11}}{D} X + \frac{f_u r_{12}}{D} Y + \frac{f_u r_{13}}{D} Z - \frac{f_u r_{11} X_0 + f_u r_{12} Y_0 + f_u r_{13} Z_0}{D}}{\frac{r_{31}}{D} X + \frac{r_{32}}{D} Y + \frac{r_{33}}{D} Z + 1}$$

$$u = \frac{u_0\left(\frac{r_{31}}{D} X + \frac{r_{32}}{D} Y + \frac{r_{33}}{D} Z + 1\right) - \left(\frac{f_u r_{11}}{D} X + \frac{f_u r_{12}}{D} Y + \frac{f r_{13}}{D} Z\right) + \frac{f_u r_{11} X_0 + f_u r_{12} Y_0 + f_u r_{13} Z_0}{D}}{L_9 X + L_{10} Y + L_{11} Z + 1}$$

$$u = \frac{\frac{u_0 r_{31} - f_u r_{11}}{D} X + \frac{u_0 r_{32} - f_u r_{12}}{D} Y + \frac{u_0 r_{33} - f r_{13}}{D} Z + \frac{f_u r_{11} X_0 + f_u r_{12} Y_0 + f_u r_{13} Z_0 + u_0 D}{D}}{L_9 X + L_{10} Y + L_{11} Z + 1}$$

$$\boxed{u = \frac{L_1 X + L_2 Y + L_3 Z + L_4}{L_9 X + L_{10} Y + L_{11} Z + 1}}$$

DLT+lens distortion

- Δu,Δv: Lens distortion
- depends on u,v!
  $(\Delta u,\Delta v) = (\Delta u(u,v),\Delta v(u,v))$

$$u + \Delta u = \frac{L_1 X + L_2 Y + L_3 Z + L_4}{L_9 X + L_{10} Y + L_{11} Z + 1}$$

$$v + \Delta v = \frac{L_5 X + L_6 Y + L_7 Z + L_8}{L_9 X + L_{10} Y + L_{11} Z + 1}$$

- Distorsion adds 1,2, or more additional parameters plus nonlinearity

DLT: 2D case

- 3D case

$$\boxed{u = \frac{L_1 X + L_2 Y + L_3 Z + L_4}{L_9 X + L_{10} Y + L_{11} Z + 1}}$$

$$\boxed{v = \frac{L_5 X + L_6 Y + L_7 Z + L_8}{L_9 X + L_{10} Y + L_{11} Z + 1}}$$

- For 2D case, set Z = 0

$$\boxed{u = \frac{L_1 X + L_2 Y + L_4}{L_9 X + L_{10} Y + 1}}$$

$$\boxed{v = \frac{L_5 X + L_6 Y + L_8}{L_9 X + L_{10} Y + 1}}$$

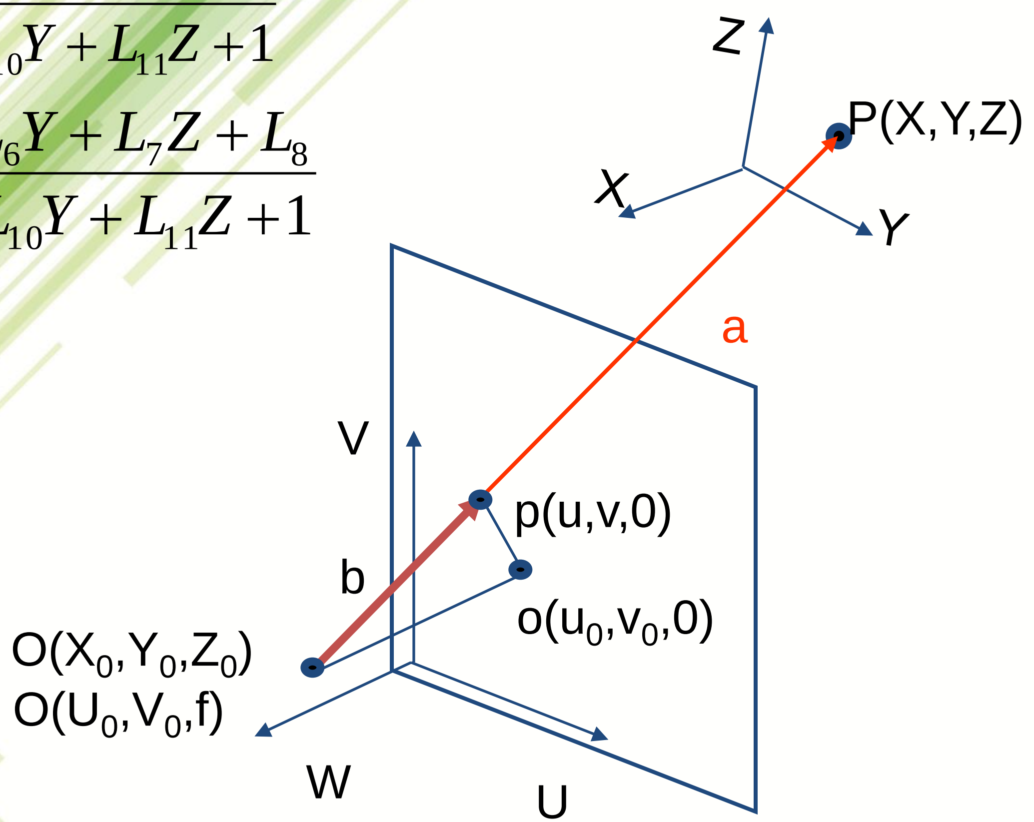
Questions?

Actual DLT calibration procedure

- We start from here

$$u = \frac{L_1 X + L_2 Y + L_3 Z + L_4}{L_9 X + L_{10} Y + L_{11} Z + 1}$$

$$v = \frac{L_5 X + L_6 Y + L_7 Z + L_8}{L_9 X + L_{10} Y + L_{11} Z + 1}$$



Actual DLT calibration procedure

$$\text{E1}: \quad L_1 = \frac{u_0 r_{31} - f_u r_{11}}{D}, \quad L_2 = \frac{u_0 r_{32} - f_u r_{12}}{D}, \quad L_3 = \frac{u_0 r_{33} - f_u r_{13}}{D}$$

$$\text{E2}: \quad L_4 = \frac{(f_u r_{11} - u_0 r_{31}) X_0 + (f_u r_{12} - u_0 r_{32}) Y_0 + (f_u r_{13} - u_0 r_{33}) Z_0}{D}$$

$$\text{E3}: \quad L_5 = \frac{v_0 r_{31} - f_v r_{21}}{D}, \quad L_6 = \frac{v_0 r_{32} - f_v r_{22}}{D}, \quad L_7 = \frac{v_0 r_{33} - f_v r_{23}}{D}$$

$$\text{E4}: \quad L_8 = \frac{(f_v r_{21} - v_0 r_{31}) X_0 + (f_v r_{22} - v_0 r_{32}) Y_0 + (f_v r_{23} - v_0 r_{33}) Z_0}{D}$$

$$\text{E5}: \quad L_9 = \frac{r_{31}}{D}, \quad L_{10} = \frac{r_{32}}{D}, \quad L_{11} = \frac{r_{33}}{D} \quad f_u = \frac{f}{\lambda_u}, \quad f_v = \frac{f}{\lambda_v}$$

$$\text{E6}: \quad D = -(X_0 r_{31} + Y_0 r_{32} + Z_0 r_{33})$$

Actual DLT calibration procedure

- We want to find unknowns L_i (i=1,2,....,11)

$$u = \frac{L_1 X + L_2 Y + L_3 Z + L_4}{L_9 X + L_{10} Y + L_{11} Z + 1} \qquad v = \frac{L_5 X + L_6 Y + L_7 Z + L_8}{L_9 X + L_{10} Y + L_{11} Z + 1}$$

$$u\left(L_9 X + L_{10} Y + L_{11} Z + 1\right) = L_1 X + L_2 Y + L_3 Z + L_4$$

$$v(L_9 X + L_{10} Y + L_{11} Z + 1) = L_5 X + L_6 Y + L_7 Z + L_8$$

$$L_1 X + L_2 Y + L_3 Z + L_4 - uXL_9 - uYL_{10} - uZL_{11} = u$$

$$L_5 X + L_6 Y + L_7 Z + L_8 - vXL_9 - vYL_{10} - vZL_{11} = v$$

Actual DLT calibration procedure

- Equation for ONE point P:
  - we know point coordinates in the scene
  - we know image coordinates of that point
  - we are interested in model parameters

$$\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -uX & -uY & -uZ \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -vX & -vY & -vZ \end{bmatrix} \times \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \\ L_9 \\ L_{10} \\ L_{11} \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$$

  - Each point gives two equations.
  - There are 11 unknowns (DLT parameters)
  - Thus, we need at least 6 „control" or so-called calibration points
  - In practice we take many more points (why?)

Vector b

Actual DLT calibration procedure

$$
\begin{bmatrix}
X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\
0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -u_1Z_1 \\
X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & -u_2X_2 & -u_2Y_2 & -u_2Z_2 \\
0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & -v_2X_2 & -v_2Y_2 & -v_2Z_2 \\
. & & & & & & & & & & . \\
. & & & & & & & & & & . \\
. & & & & & . & & & & & . \\
. & & & & & & & & & & . \\
. & & & & & & & . & & & . \\
. & & & & & & & & & & . \\
X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n \\
0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n
\end{bmatrix}
\times
\begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \\ L_9 \\ L_{10} \\ L_{11} \end{bmatrix}
=
\begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ . \\ . \\ . \\ . \\ . \\ . \\ u_n \\ v_n \end{bmatrix}
$$

Matrix A

Actual DLT calibration procedure

- We need to find the solution of an overdetermined system
  - The method of (linear) least squares (LS)!

$$\mathbf{A}_{2nx11}\mathrm{L} = \mathbf{b}$$

$$\left[\mathbf{A}^{\mathrm{T}}\mathbf{A}\right]\mathrm{L} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$$

$$\left[\mathbf{A}^{\mathrm{T}}\mathbf{A}\right]^{-1}\left[\mathbf{A}^{\mathrm{T}}\mathbf{A}\right]\mathrm{L} = \left[\mathbf{A}^{T}\mathbf{A}\right]^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

$$\mathrm{L} = \left[\mathbf{A}^{\mathrm{T}}\mathbf{A}\right]^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

Camera parameters from DLT

$$\text{E1}: \; L_1 = \frac{u_0 r_{31} - f_u r_{11}}{D}, \quad L_2 = \frac{u_0 r_{32} - f_u r_{12}}{D}, \quad L_3 = \frac{u_0 r_{33} - f_u r_{13}}{D}$$

$$\text{E2}: \; L_4 = \frac{(f_u r_{11} - u_0 r_{31}) X_0 + (f_u r_{12} - u_0 r_{32}) Y_0 + (f_u r_{13} - u_0 r_{33}) Z_0}{D}$$

$$L_4 = -\frac{(u_0 r_{31} - f_u r_{11})}{D} X_0 - \frac{(u_0 r_{32} - f_u r_{12})}{D} Y_0 - \frac{(u_0 r_{33} - f_u r_{13})}{D} Z_0$$

$$\boxed{L_1 X_0 + L_2 Y_0 + L_3 Z_0 = -L_4}$$

Camera parameters from DLT

$$\text{E3}: L_5 = \frac{v_0 r_{31} - f_v r_{21}}{D}, \quad L_6 = \frac{v_0 r_{32} - f_v r_{22}}{D}, \quad L_7 = \frac{v_0 r_{33} - f_v r_{23}}{D}$$

$$\text{E4}: L_8 = \frac{\left(f_v r_{21} - v_0 r_{31}\right) X_0 + \left(f_v r_{22} - v_0 r_{32}\right) Y_0 + \left(f_v r_{23} - v_0 r_{33}\right) Z_0}{D}$$

$$\downarrow$$

$$L_8 = -\frac{\left(v_0 r_{31} - f_v r_{21}\right)}{D} X_0 - \frac{\left(v_0 r_{32} - f_v r_{22}\right)}{D} Y_0 - \frac{\left(v_0 r_{33} - f_v r_{23}\right)}{D} Z_0$$

$$\boxed{L_5 X_0 + L_6 Y_0 + L_7 Z_0 = -L_8}$$

Camera parameters from DLT

$$\text{E5}: \; L_9 = \frac{r_{31}}{D}, \quad L_{10} = \frac{r_{32}}{D}, \quad L_{11} = \frac{r_{33}}{D}$$

$$\text{E6}: \; D = -(X_0 r_{31} + Y_0 r_{32} + Z_0 r_{33})$$

/ *divide by D*

$$-1 = X_0 \frac{r_{31}}{D} + Y_0 \frac{r_{32}}{D} + Z_0 \frac{r_{33}}{D}$$

$$\boxed{L_9 X_0 + L_{10} Y_0 + L_{11} Z_0 = -1}$$

Camera parameters from DLT

$$L_1 X_0 + L_2 Y_0 + L_3 Z_0 = -L_4$$
$$L_5 X_0 + L_6 Y_0 + L_7 Z_0 = -L_8$$
$$L_9 X_0 + L_{10} Y_0 + L_{11} Z_0 = -1$$

$$\begin{bmatrix} L_1 & L_2 & L_3 \\ L_5 & L_6 & L_7 \\ L_9 & L_{10} & L_{11} \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} = \begin{bmatrix} -L_4 \\ -L_8 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} = \begin{bmatrix} L_1 & L_2 & L_3 \\ L_5 & L_6 & L_7 \\ L_9 & L_{10} & L_{11} \end{bmatrix}^{-1} \begin{bmatrix} -L_4 \\ -L_8 \\ -1 \end{bmatrix}$$

Camera parameters from DLT

$$\text{E5}: L_9 = \frac{r_{31}}{D}, \quad L_{10} = \frac{r_{32}}{D}, \quad L_{11} = \frac{r_{33}}{D} \quad / \quad \textit{square, sum}$$

$$L_9^2 + L_{10}^2 + L_{11}^2 = \frac{1}{D^2}\left(\underbrace{r_{31}^2 + r_{32}^2 + r_{33}^2}_{1}\right) = \frac{1}{D^2}$$

- R is rotational matrix, not *just any* matrix!
- We assume orthogonality of R, $R^T R = I$
  - Preserves distances

Camera parameters from DLT

- Use orthogonality assumption again

$$\text{E5}: \; L_9 = \frac{r_{31}}{D}, \quad L_{10} = \frac{r_{32}}{D}, \quad L_{11} = \frac{r_{33}}{D} \quad / \; \textit{multiply by D}$$

$$\text{E1}: \; L_1 = \frac{u_0 r_{31} - f_u r_{11}}{D}, \quad L_2 = \frac{u_0 r_{32} - f_u r_{12}}{D}, \quad L_3 = \frac{u_0 r_{33} - f_u r_{13}}{D}$$

$$(DL_1)(DL_9) + (DL_2)(DL_{10}) + (DL_3)(DL_{11}) =$$

$$= u_0 (\underbrace{r_{31}^2 + r_{32}^2 + r_{33}^2}_{1}) - f_u (\underbrace{r_{11} r_{31} + r_{12} r_{32} + r_{13} r_{33}}_{0}) = u_0$$

Camera parameters from DLT

- And orthogonality again

$$\mathrm{E5:}\; L_9 = \frac{r_{31}}{D}, \quad L_{10} = \frac{r_{32}}{D}, \quad L_{11} = \frac{r_{33}}{D}$$

$$\mathrm{E3:}\; L_5 = \frac{v_0 r_{31} - f_v r_{21}}{D}, \quad L_6 = \frac{v_0 r_{32} - f_v r_{22}}{D}, \quad L_7 = \frac{v_0 r_{33} - f_v r_{23}}{D}$$

$$(DL_5)(DL_9) + (DL_6)(DL_{10}) + (DL_7)(DL_{11}) =$$

$$= v_0 \underbrace{(r_{31}^2 + r_{32}^2 + r_{33}^2)}_{1} - f_v \underbrace{(r_{21}r_{31} + r_{22}r_{32} + r_{23}r_{33})}_{0} = v_0$$

Camera parameters from DLT

$$(DL_1)(DL_9)+(DL_2)(DL_{10})+(DL_3)(DL_{11})=u_0$$
$$(DL_5)(DL_9)+(DL_6)(DL_{10})+(DL_7)(DL_{11})=v_0$$

$$L_9^2+L_{10}^2+L_{11}^2=\frac{1}{D^2}$$

$$u_0=D^2(L_1L_9+L_2L_{10}+L_3L_{11})=\frac{L_1L_9+L_2L_{10}+L_3L_{11}}{L_9^2+L_{10}^2+L_{11}^2}$$

$$v_0=D^2(L_5L_9+L_6L_{10}+L_7L_{11})=\frac{L_5L_9+L_6L_{10}+L_7L_{11}}{L_9^2+L_{10}^2+L_{11}^2}$$

Camera parameters from DLT

- We are searching for rotation matrix R now

$$\text{E5}: L_9 = \frac{r_{31}}{D}, \quad L_{10} = \frac{r_{32}}{D}, \quad L_{11} = \frac{r_{33}}{D}$$

$$r_{31} = DL_9, \quad r_{32} = DL_{10}, \quad r_{33} = DL_{11}$$

Camera parameters from DLT

$$\text{E1}: L_1 = \frac{u_0 r_{31} - f_u r_{11}}{D}, \quad L_2 = \frac{u_0 r_{32} - f_u r_{12}}{D}, \quad L_3 = \frac{u_0 r_{33} - f_u r_{13}}{D}$$

$$\text{E5}: L_9 = \frac{r_{31}}{D}, \quad L_{10} = \frac{r_{32}}{D}, \quad L_{11} = \frac{r_{33}}{D}$$

$$L_1 = u_0 L_9 - \frac{f_u r_{11}}{D}, \quad L_2 = u_0 L_{10} - \frac{f_u r_{12}}{D}, \quad L_3 = u_0 L_{11} - \frac{f_u r_{13}}{D}$$

$$r_{11} = \frac{D(u_0 L_9 - L_1)}{f_u}, \quad r_{12} = \frac{D(u_0 L_{10} - L_2)}{f_u}, \quad r_{13} = \frac{D(u_0 L_{11} - L_3)}{f_u}$$

Camera parameters from DLT

$$\mathrm{E3}: L_5 = \frac{v_0 r_{31} - f_v r_{21}}{D}, \quad L_6 = \frac{v_0 r_{32} - f_v r_{22}}{D}, \quad L_7 = \frac{v_0 r_{33} - f_v r_{23}}{D}$$

$$\mathrm{E5}: L_9 = \frac{r_{31}}{D}, \quad L_{10} = \frac{r_{32}}{D}, \quad L_{11} = \frac{r_{33}}{D}$$

$$L_5 = v_0 L_9 - \frac{f_v r_{21}}{D}, \quad L_6 = v_0 L_{10} - \frac{f_v r_{22}}{D}, \quad L_7 = v_0 L_{11} - \frac{f_v r_{23}}{D}$$

$$r_{21} = \frac{D(v_0 L_9 - L_5)}{f_v}, \quad r_{22} = \frac{D(v_0 L_{10} - L_6)}{f_v}, \quad r_{23} = \frac{D(v_0 L_{11} - L_7)}{f_v}$$

Camera parameters from DLT

$$r_{11} = \frac{D(u_0 L_9 - L_1)}{f_u}, \quad r_{12} = \frac{D(u_0 L_{10} - L_2)}{f_u}, \quad r_{13} = \frac{D(u_0 L_{11} - L_3)}{f_u}$$

$$r_{21} = \frac{D(v_0 L_9 - L_5)}{f_v}, \quad r_{22} = \frac{D(v_0 L_{10} - L_6)}{f_v}, \quad r_{23} = \frac{D(v_0 L_{11} - L_7)}{f_v}$$

$$r_{31} = DL_9, \quad r_{32} = DL_{10}, \quad r_{33} = DL_{11}$$

f_u and f_v are not known right now!

$$f_u = \frac{f}{\lambda_u}, \quad f_v = \frac{f}{\lambda_v}$$

Camera parameters from DLT

- Orthogonality again!

$$r_{11}^2 + r_{12}^2 + r_{13}^2 = \frac{D^2[(u_0 L_9 - L_1)^2 + (u_0 L_{10} - L_2)^2 + (u_0 L_{11} - L_3)^2]}{f_u^{\,2}} = 1$$

$$r_{21}^2 + r_{22}^2 + r_{23}^2 = \frac{D^2[(v_0 L_9 - L_5)^2 + (v_0 L_{10} - L_6)^2 + (v_0 L_{11} - L_7)^2]}{f_v^{\,2}} = 1$$

$$f_u^{\,2} = D^2[(u_0 L_9 - L_1)^2 + (u_0 L_{10} - L_2)^2 + (u_0 L_{11} - L_3)^2]$$

$$f_v^{\,2} = D^2[(v_0 L_9 - L_5)^2 + (v_0 L_{10} - L_6)^2 + (v_0 L_{11} - L_7)^2]$$

DLT calibration, recap

- Calibration “pattern” -> $X_i, Y_i, Z_i, (i=1,\ldots,N)$
  - Points should not be co-planar
  - Points should cover the working area well
- Calibration pattern image, processing -> $u_i, v_i, (i=1,\ldots,N)$
- X_i, Y_i, Z_i, u_i, v_i, LS method -> $L_1, \ldots, L_{11}$
- $L_1, \ldots, L_{11}$ -> X_0, Y_0, Z_0
- $L_1, \ldots, L_{11}$, ortogonalnost -> u_0, v_0
- $L_1, \ldots, L_{11}, u_0, v_0, \perp$ -> f_u, f_v
- $L_1, \ldots, L_{11}, u_0, v_0, f_u, f_v \perp$ -> $R=[r_{ij}]$

Reconstruction: obtaining X, Y, Z

- Camera is now calibrated
  - We are interested in object (point) position:
  - We know parameters L
  - We know image coordinates u,v
    (point detection algorithms)
  - We want to find X,Y,Z in space

- But, DLT model goes from 3D into 2D
  - Thus, we need inverse transformation
  - How?

Reconstruction: obtaining X, Y, Z

- We need multiple cameras (m>1)
  - Multiple images of the same scene (per camera) are beneficial

$$\begin{bmatrix} (L_1 - uL_9)_1 & (L_2 - uL_{10})_1 & (L_3 - uL_{11})_1 \\ (L_5 - vL_9)_1 & (L_6 - vL_{10})_1 & (L_7 - vL_{11})_1 \\ . & . & . \\ . & . & . \\ (L_1 - uL_9)_m & (L_2 - uL_{10})_m & (L_3 - uL_{11})_m \\ (L_5 - vL_9)_m & (L_6 - vL_{10})_m & (L_7 - vL_{11})_m \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} (L_4 - u)_1 \\ (L_8 - v)_1 \\ . \\ . \\ (L_4 - u)_m \\ (L_8 - v)_m \end{bmatrix}$$

- We again solve the system using LS

Reconstruction: obtaining X, Y, Z in practice

- We set up (calibrated) cameras
  - Need at least two (yes, that's called stereo system!)
- Capture images of the same scene
  - We don't NEED multiple images, but it helps with LS
- Analyze each image (each separately)
  - Find key points (good, distinctive points) in each image
- A key problem is:
  - Which points in different cameras correspond to same point in the scene (a.k.a. "correspondence problem")
- This is known as "stereo matching"

Literature

- http://kwon3d.com/theory/dlt/dlt.html

- E. Trucco, A. Verri, Introductory Techniques for 3D Computer Vision, Prentice Hall, 1998.

- M. Sonka, V. Hlavač, R. Boyle, Image Processing, Analysis, and Machine Vision, Thomson, 2008.

- Z. Zhang, “A flexible new technique for camera calibration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

Questions?