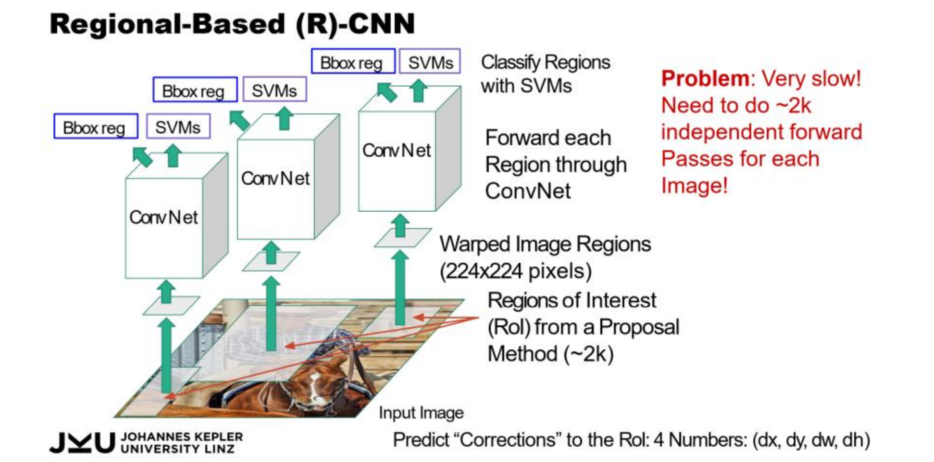
Computer Vision

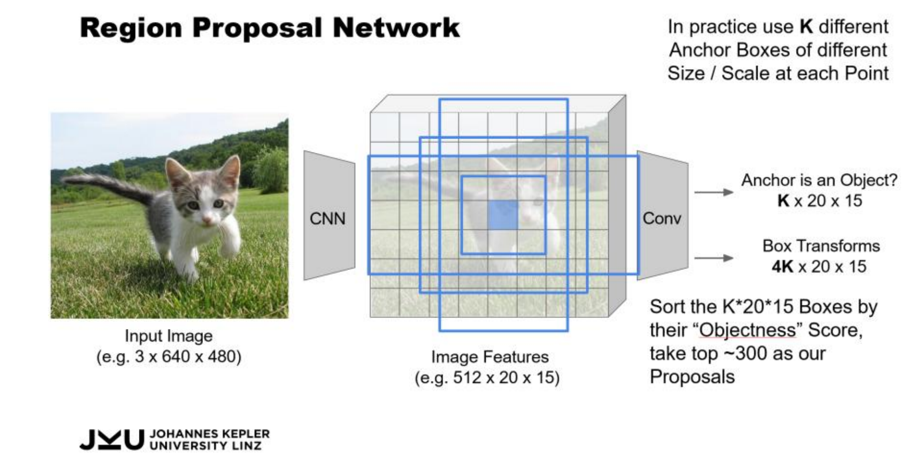
Lecture 9: Multi-View Geometry

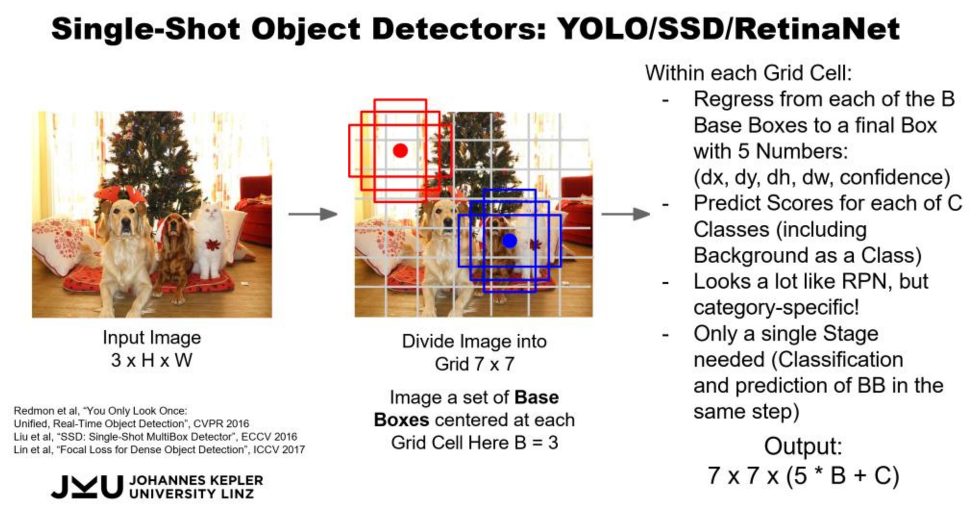
Oliver Bimber

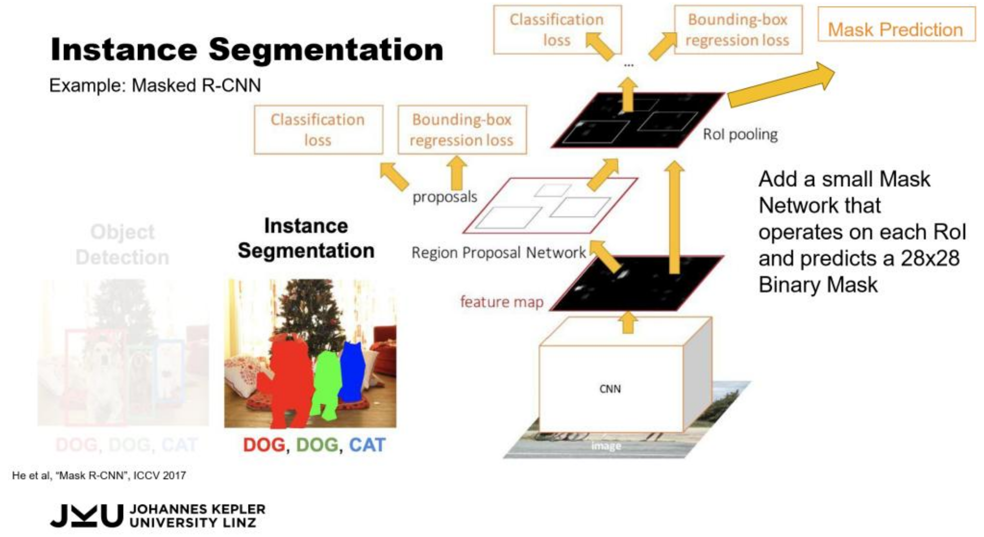
JOHANNES KEPLER UNIVERSITY LINZ

Last Week: Object Detection









Course Overview

CW	Topic	Date	Place	Lab
41	Introduction and Course Overview	07.10.2025	Zoom	Lab 1
42	Capturing Digital Images	14.10.2025	Zoom	Lab 2
43	Digital Image Processing	21.10.2025	Zoom	Assignment 1
44	Machine Learning	28.10.2025	Zoom	
45	Feature Extraction	04.11.2025	Zoom	Open Lab 1
46	Segmentation	11.11.2025	Zoom	Assignment 2
47	Optical Flow	18.11.2025	Zoom	Open Lab 2
48	Object Detection	25.11.2025	Zoom	Assignment 3
49	Multi-View Geometry	02.12.2025	Zoom	Open Lab 3
50	3D Vision	10.12.2025	Zoom	Assignment 4
3	Trends in Computer Vision	13.01.2026	Zoom	
4	Q&A	20.01.2026	Zoom	Open Lab 4
5	Exam	27.01.2026	HS1 (Linz), S1/S3 (Vienna), S5 (Bregenz)	
9	Retry Exam	24.02.2026	tba	

Goal: Depth-Estimation

Depth from Stereo(-scopic Images) = two Views

Two Problems:

1.) Finding matching Features
2.) Computing Depth of corresponding Scenepoint (relative to Cameras)

Recap: Simplified Mathematical Camera Model

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \underbrace{\overset{[3x3]}{\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{Intrinsic Parameters (K)}} \cdot \underbrace{\overset{[3x3]}{\begin{bmatrix} \mathbf{R} \end{bmatrix}} \cdot \overset{[3x4]}{\begin{bmatrix} \mathbf{I} & \mathbf{-T} \end{bmatrix}}}_{\text{Extrinsic Parameters}} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Note, that in Practice there are more intrinsic Parameters: e.g. Principle Point and Skew Angle describing Misalignment of Sensor on Optical Axis.

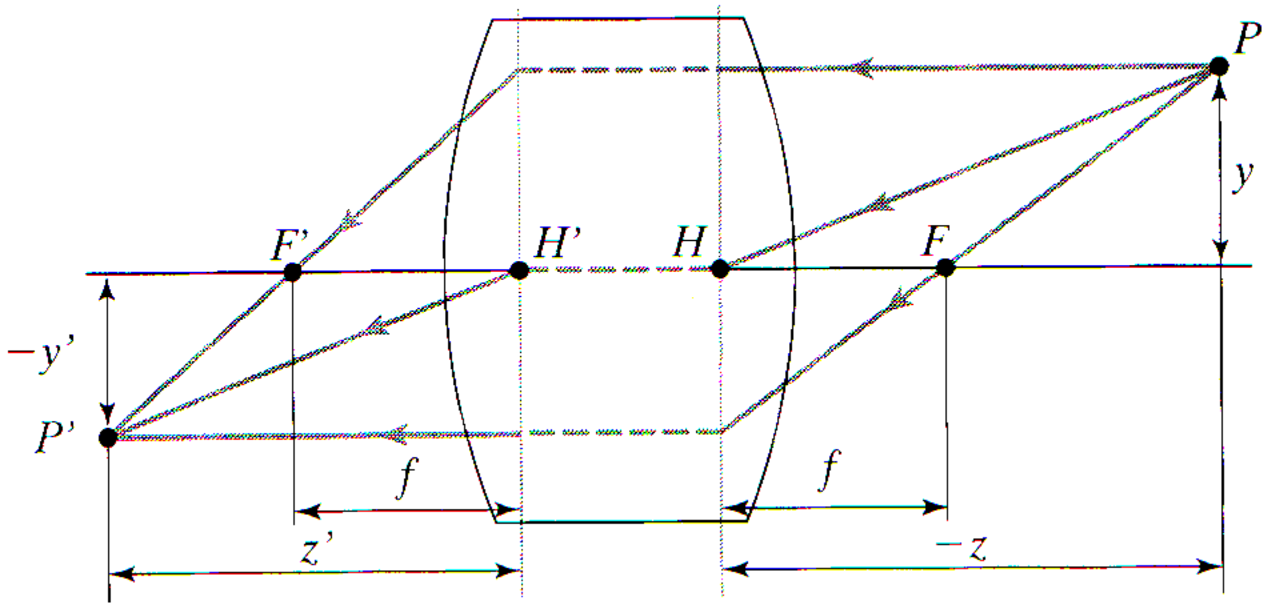
Recap: Simplified Mathematical Camera Model

$$p = MP \longrightarrow p = \frac{1}{z} MP \qquad M = K(\, R \quad t \,)$$

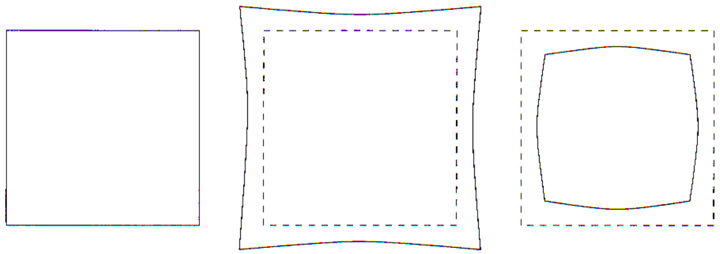
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \underbrace{\overset{[3x3]}{\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{Intrinsic Parameters (K)}} \cdot \underbrace{\overset{[3x3]}{\begin{bmatrix} R \end{bmatrix}} \cdot \overset{[3x4]}{\begin{bmatrix} I & -T \end{bmatrix}}}_{\text{Extrinsic Parameters}} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Note, that in Practice there are more intrinsic Parameters: e.g. Principle Point and Skew Angle describing Misalignment of Sensor on Optical Axis.

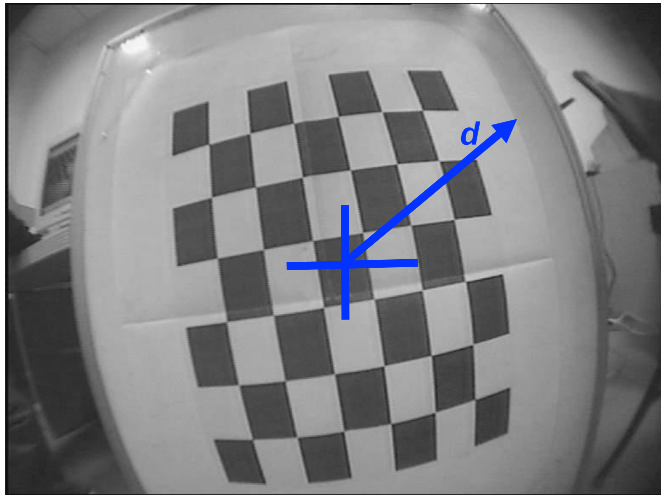
Recap: Radial Distortion



Thick Lens



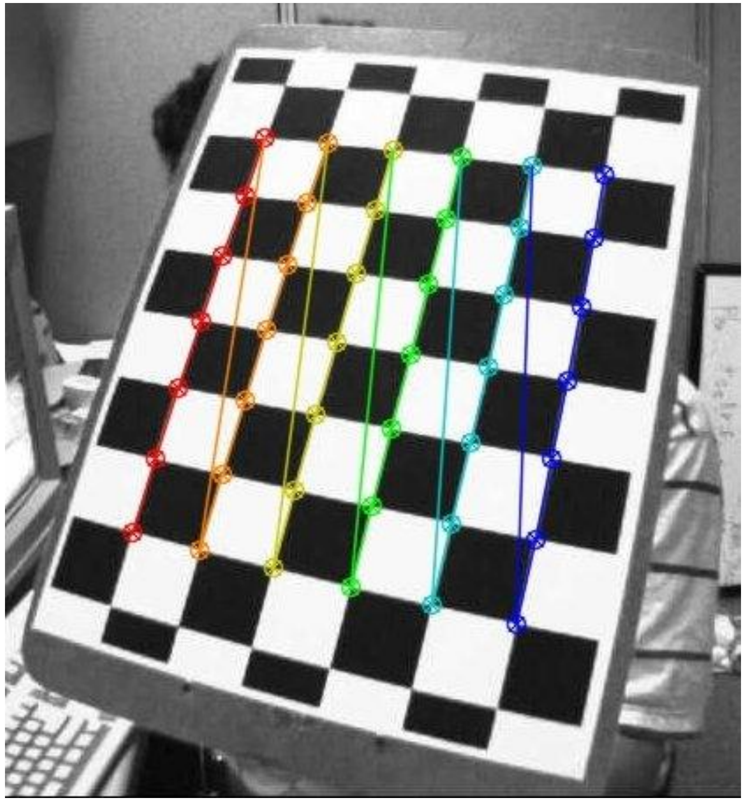
Pincushion Barrel



Radial Distortion

If Lens Distortion is considered, our Mathematical Model becomes non-linear (usually a Polynomial Function)!

Recap: Camera Calibration

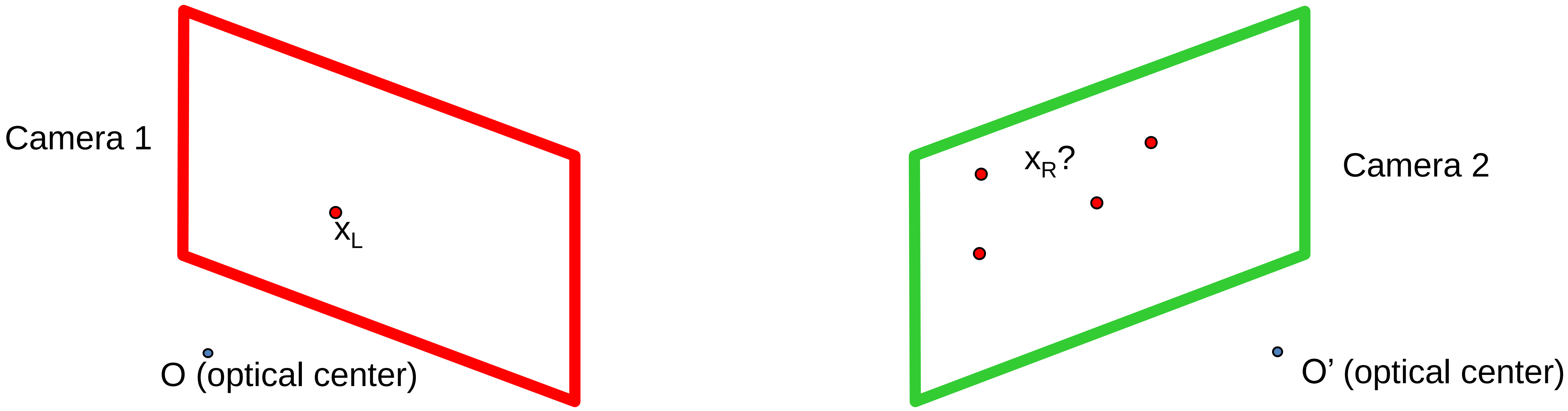


Images of known Calibration Pattern

cv.calibrateCamera() returns the camera matrix, distortion coefficients, rotation and translation vectors etc.

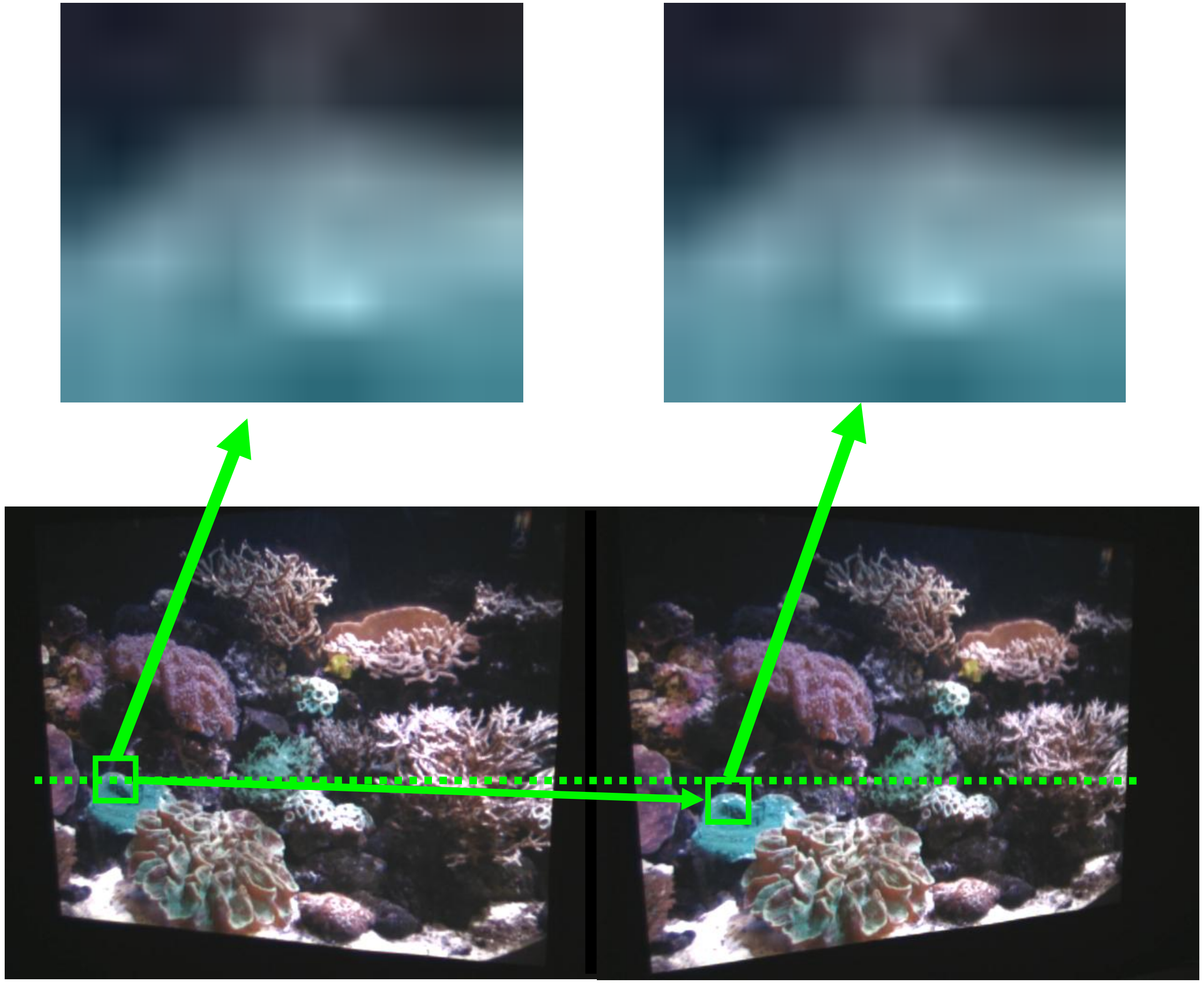
In OpenCV: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

Two Calibrated Cameras



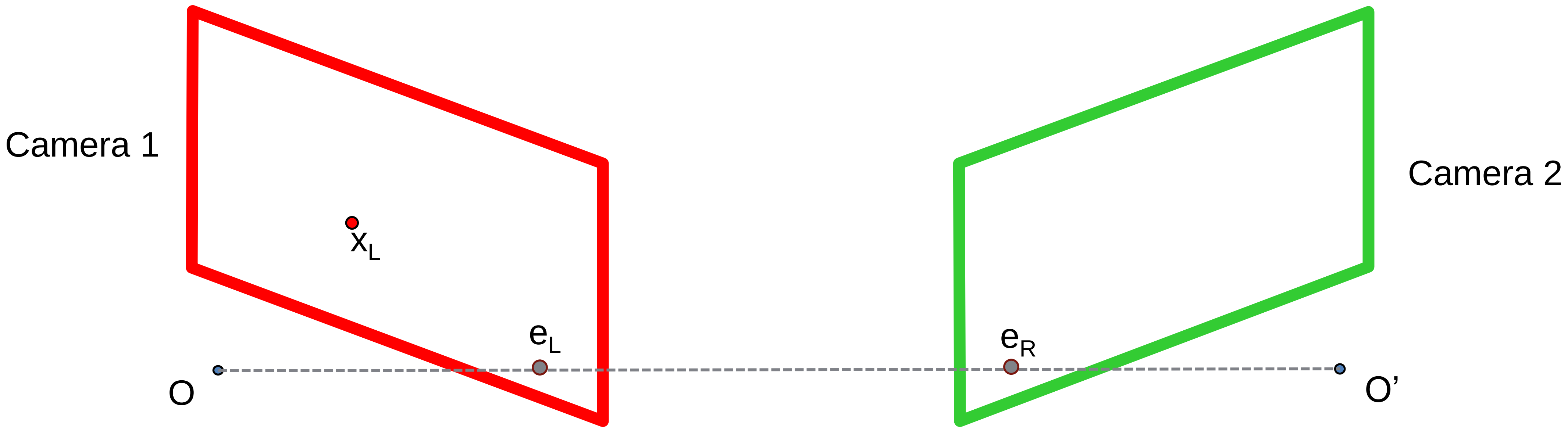
If we see a Point in Cam 1, are there any Constraints on where we will find it in Cam 2?

Feature Matching Problem



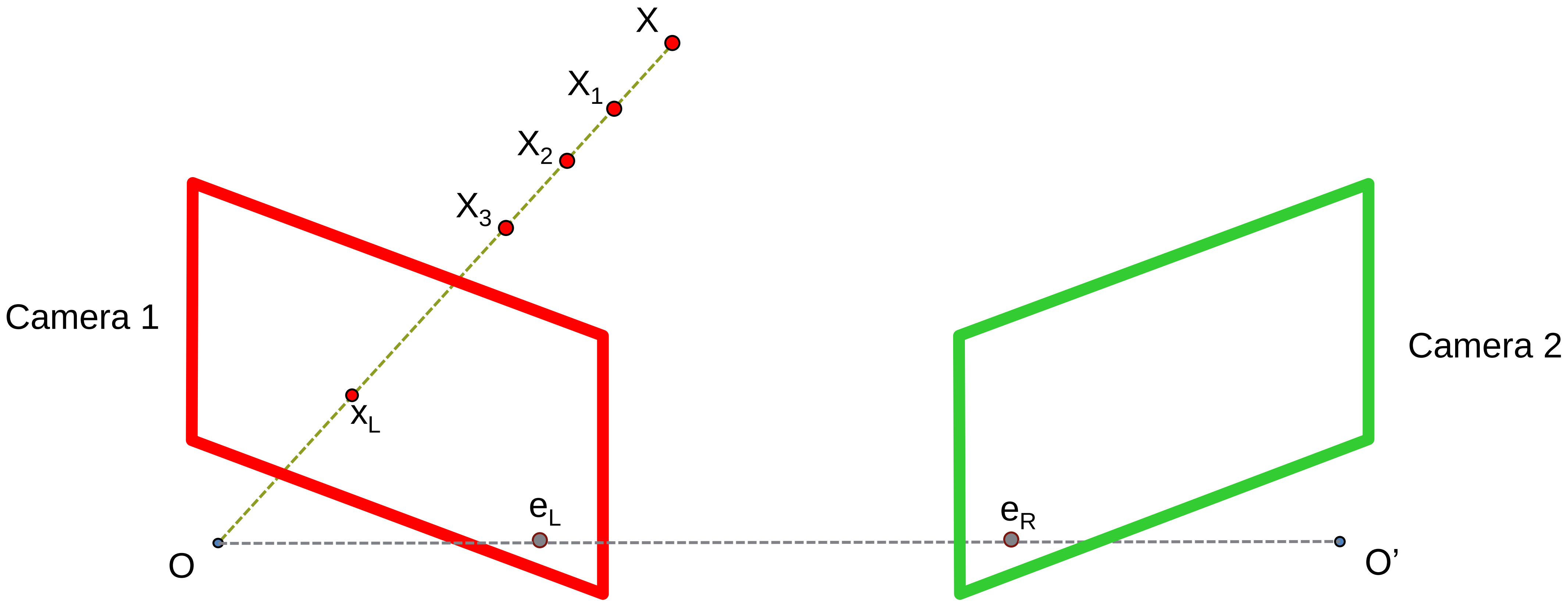
Is it really a 2D Search Problem?

Epipolar Constraints

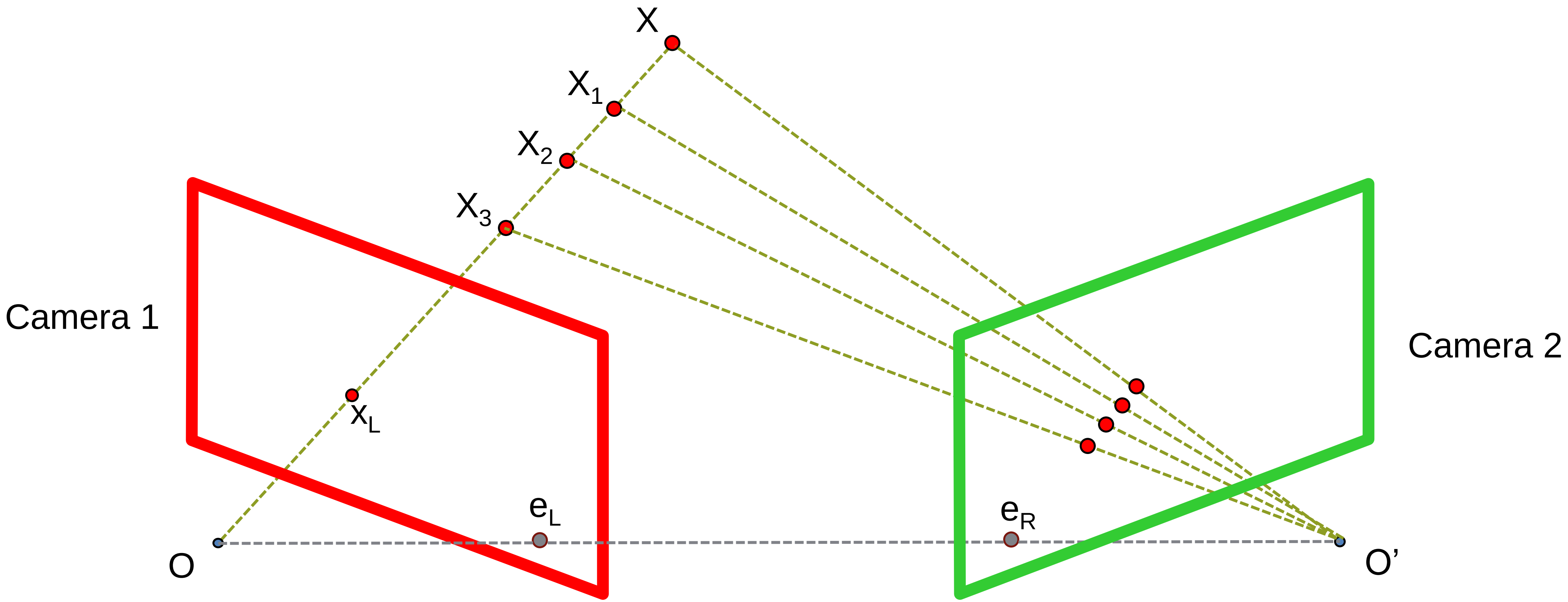


Epipoles: project each Optical Center onto other Image Plane
(Distance between Optical Centers is called Baseline)

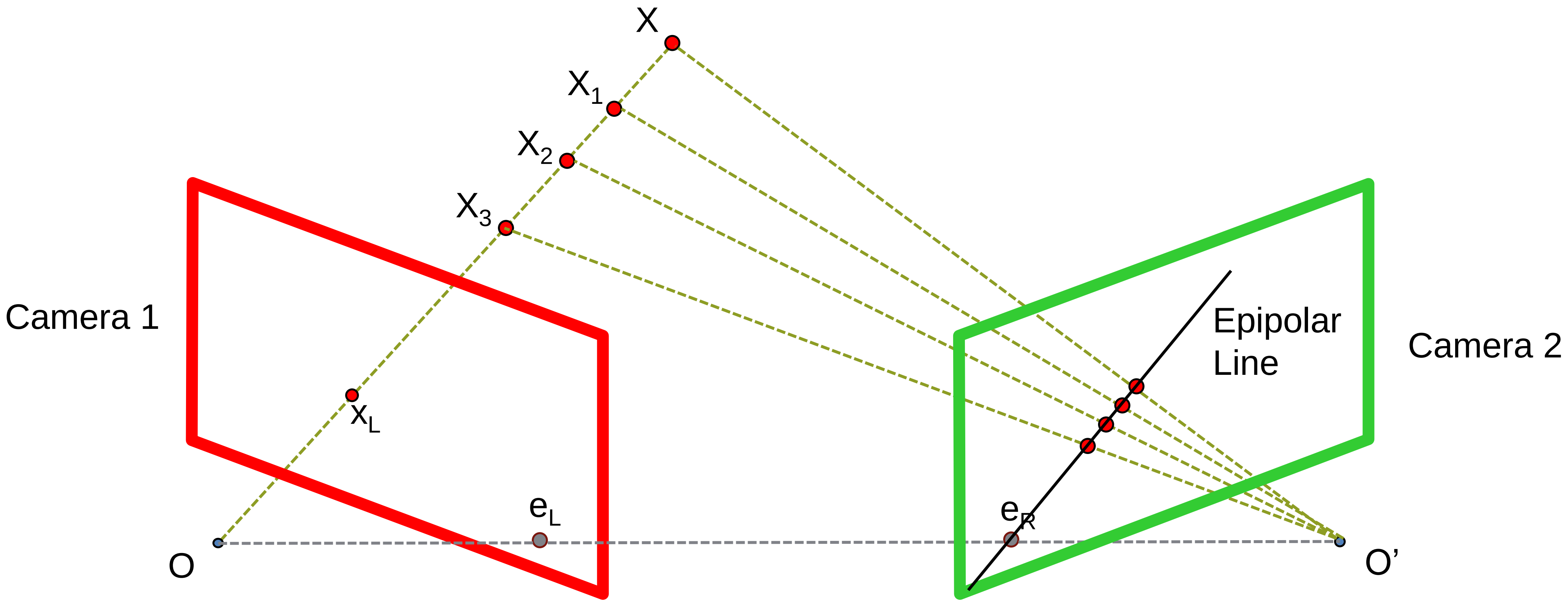
Epipolar Constraints



Epipolar Constraints



Epipolar Constraints



It's a 1D Search Problem!
But how do we get the Epipolar Line for a given Point?

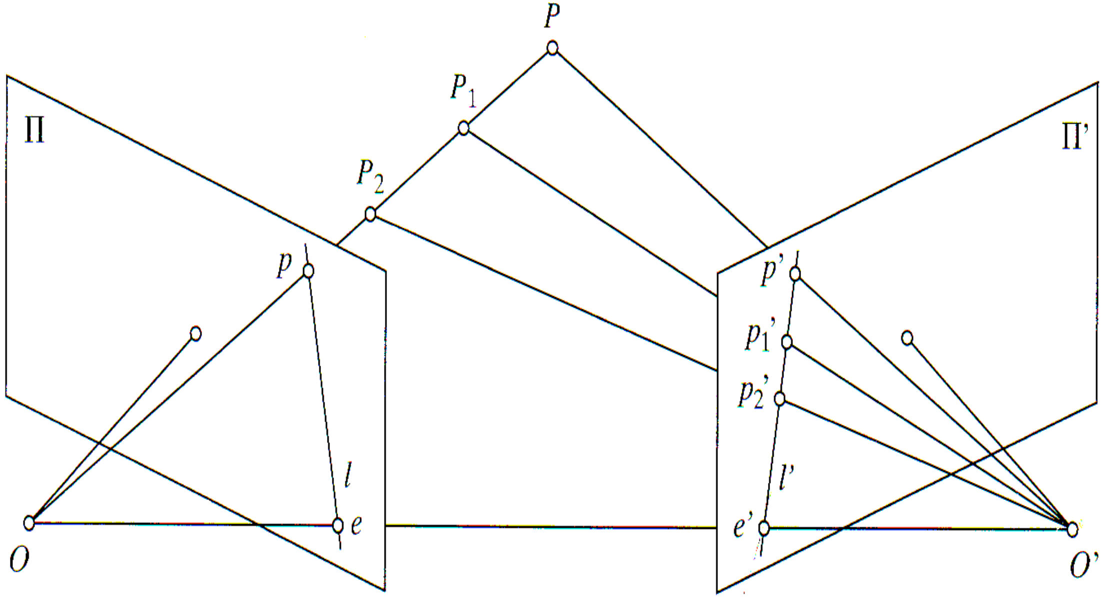
Epipolar Constraints

Calibrated Case
p,p’ on normalized Image Plane, Intrinsic K known:

$$p^T \varepsilon p' = 0$$

$l = \varepsilon p'$, $l' = \varepsilon^T p$

Line Parameters of Epipolar Line on normalized Image Plane, ***ε*** → 3x3 Essential Matrix



Uncalibrated Case
p,p’ on physical Image Plane, Intrinsic K unknown:

$$p^T F p' = 0$$

$l = Fp'$, $l' = F^T p$

Line Parameters of Epipolar Line on physical Image Plane, ***F*** → 3x3 Fundamental Matrix

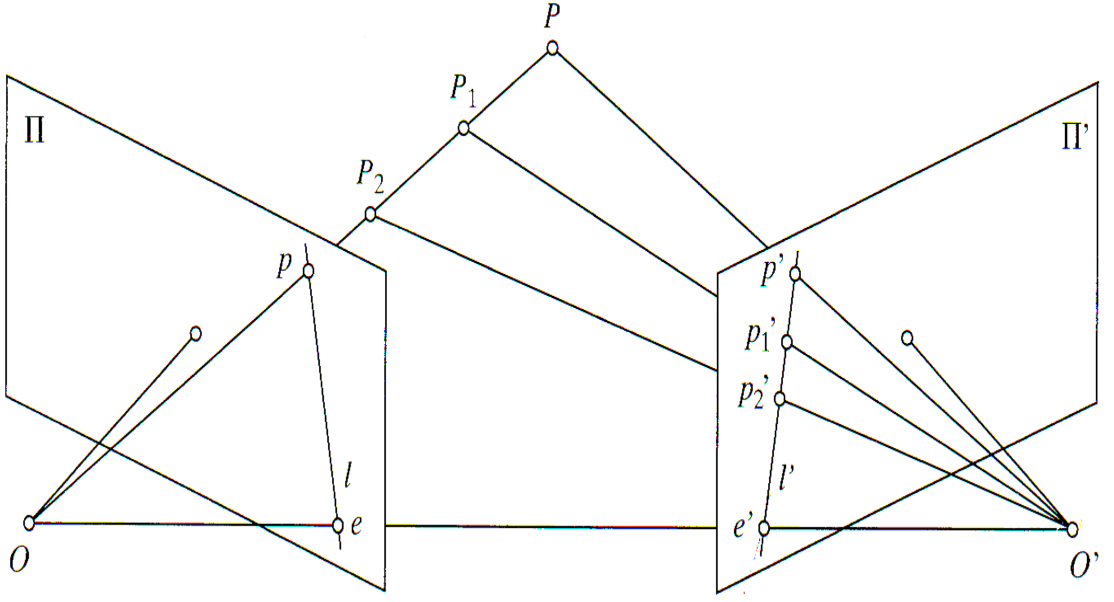
Epipolar Constraints

Calibrated Case
p,p' on normalized Image Plane, Intrinsic K known:

$$p^T \varepsilon p' = 0$$

$l = \varepsilon p'$, $l' = \varepsilon^T p$

Line Parameters of Epipolar Line on normalized Image Plane, $\boldsymbol{\varepsilon}$ → 3x3 Essential Matrix



How do we get the Matrices?

Uncalibrated Case
p,p' on physical Image Plane, Intrinsic K unknown:

$$p^T F p' = 0$$

$l = Fp'$, $l' = F^T p$

Line Parameters of Epipolar Line on physical Image Plane, ***F*** → 3x3 Fundamental Matrix

Example: The uncalibrated Case

$$p^T F p' = 0$$

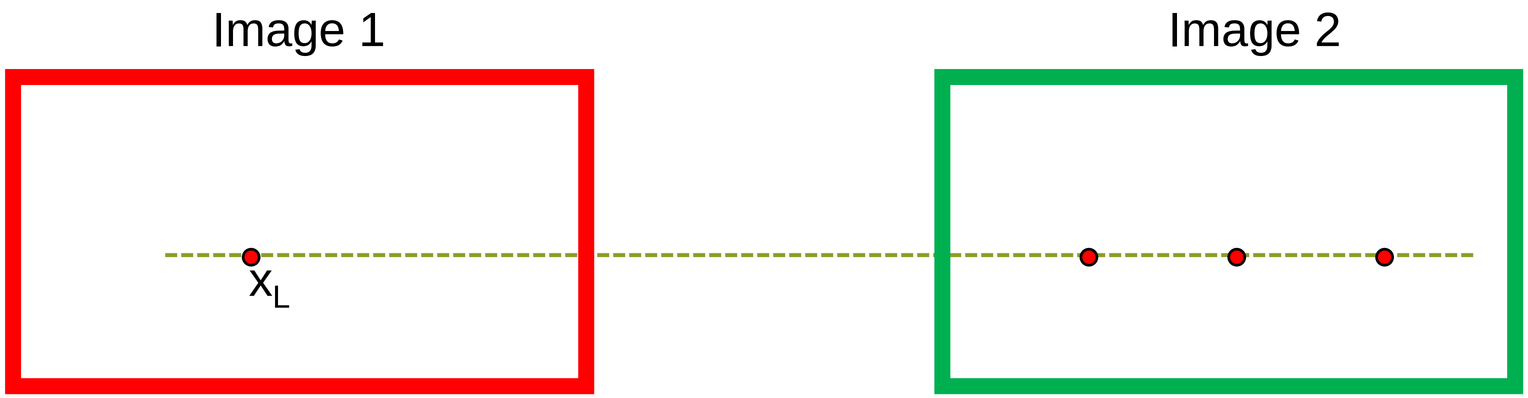
$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}^T \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

$$E = \sum_{i=1}^{n} (p_i^T F p'_i)^2$$

To get the Matrix-Coefficients, you need to have sufficient amount of known corresponding point-pairs (>8)

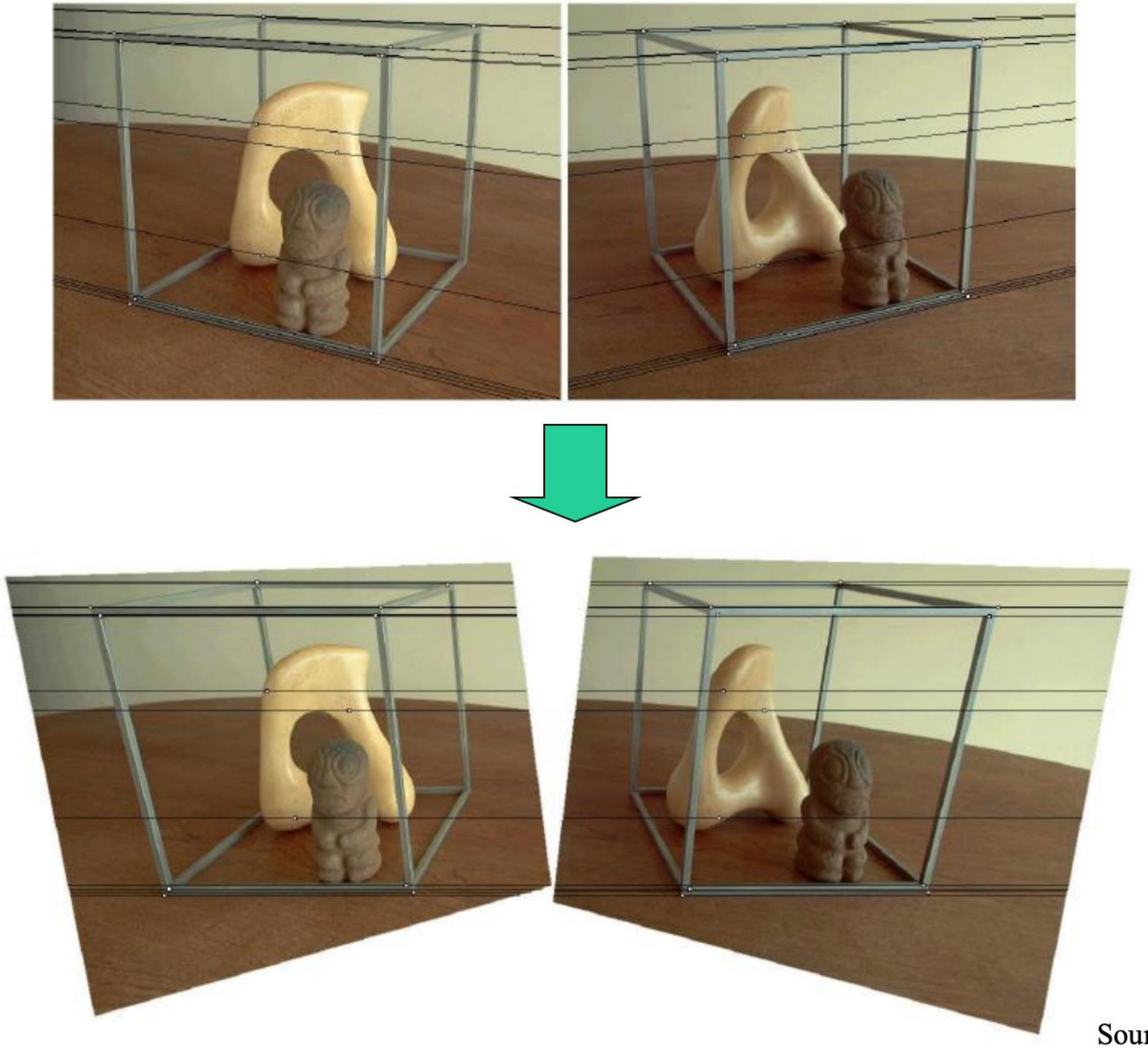
Image Rectification

If cameras are only displaced horizontally, Epipolar Lines are horizontal!



Matching Point must lie on this Line!

Image Rectification

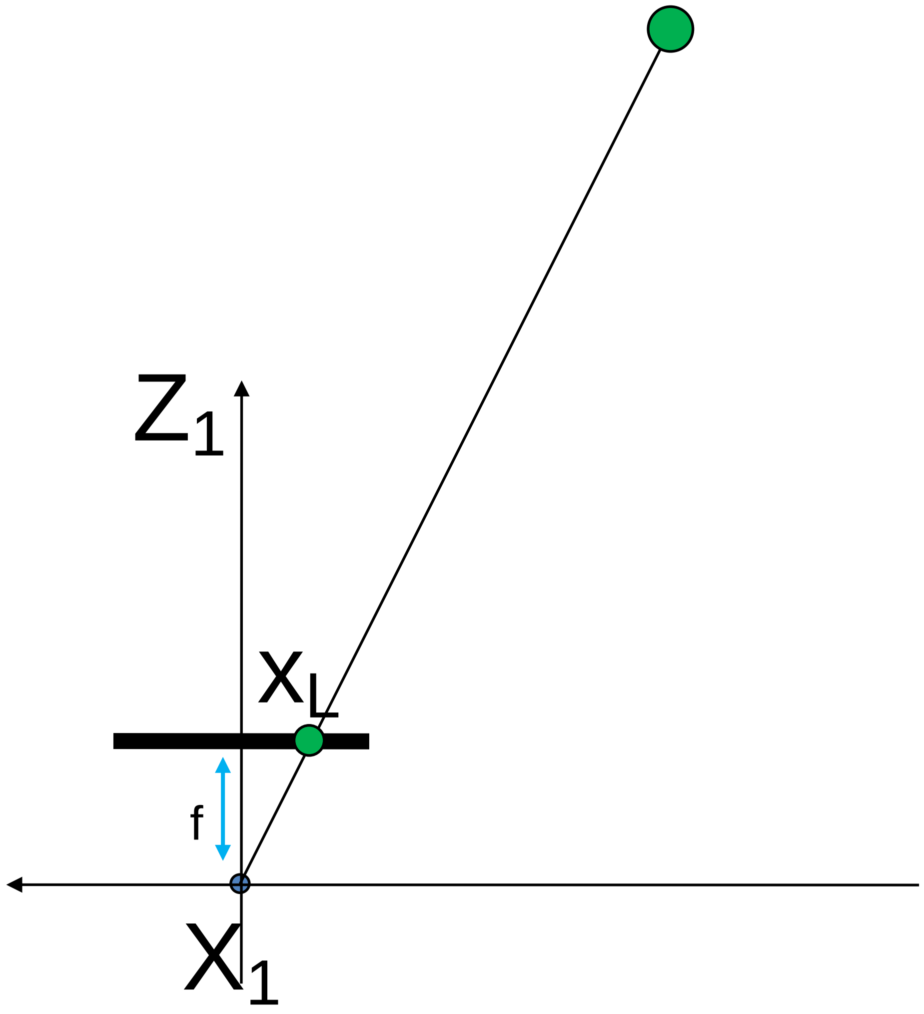


Source: Alyosha Efros

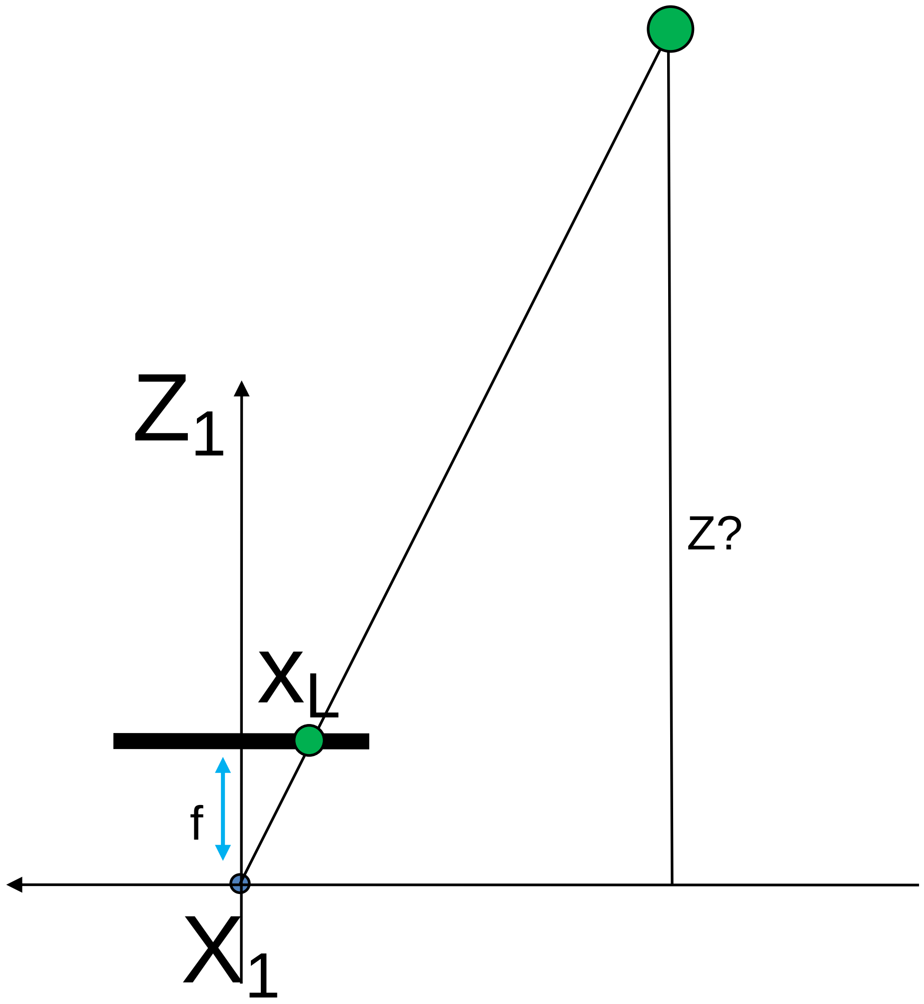
Correspondence Search

Assuming Images are rectified, we only need to search for Matches along horizontal Scanlines.

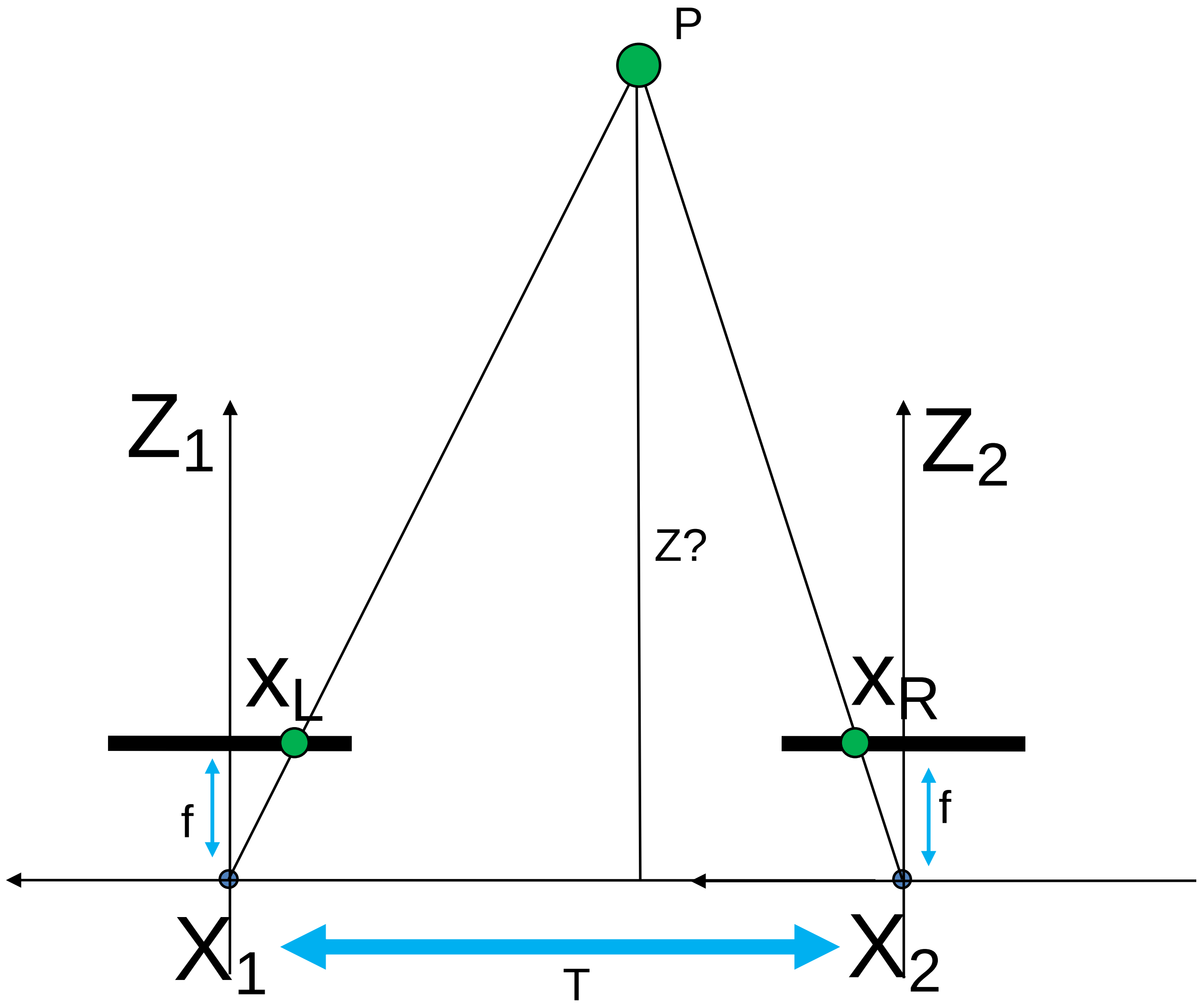
Stereoscopic Depth-Reconstruction



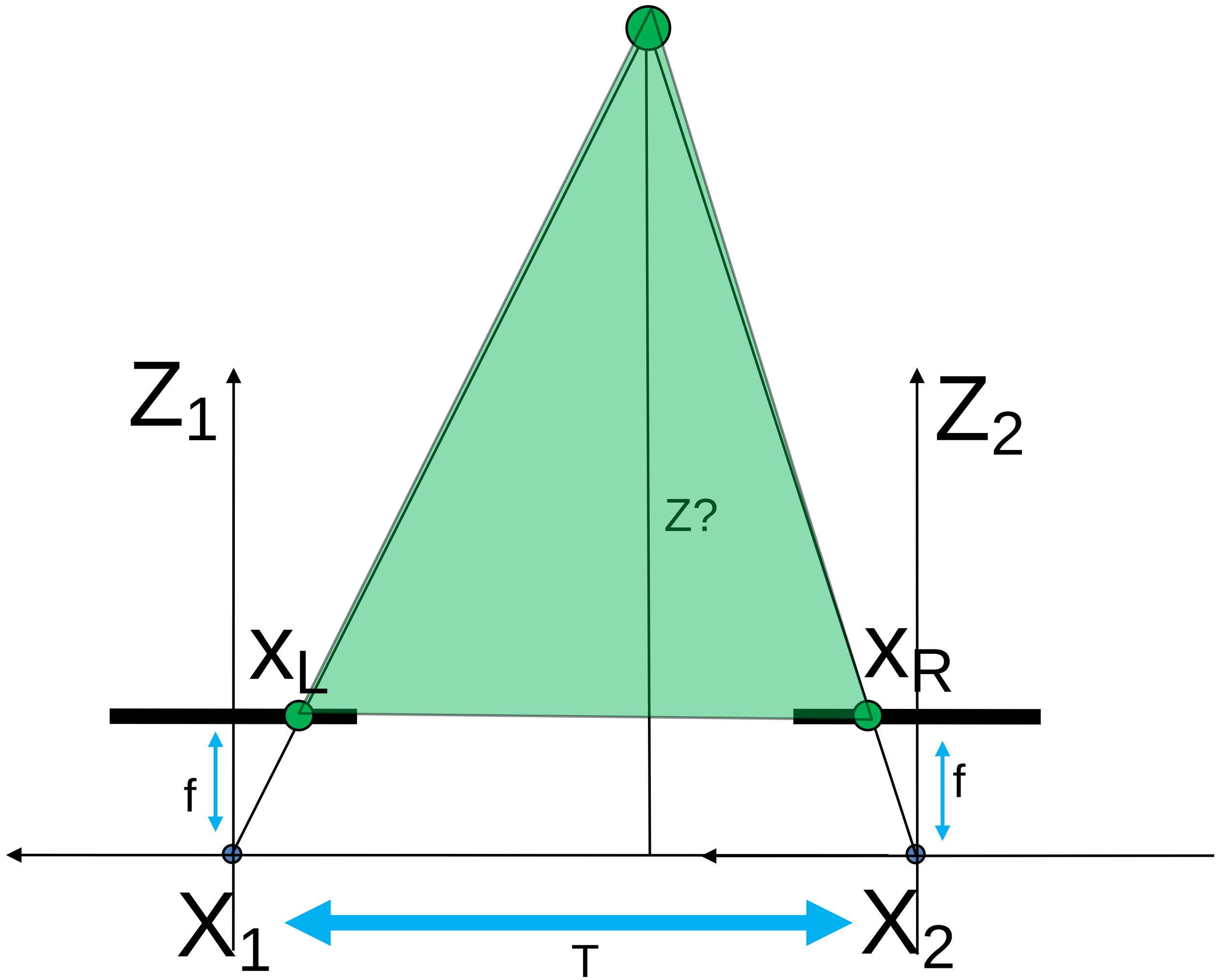
Stereoscopic Depth-Reconstruction



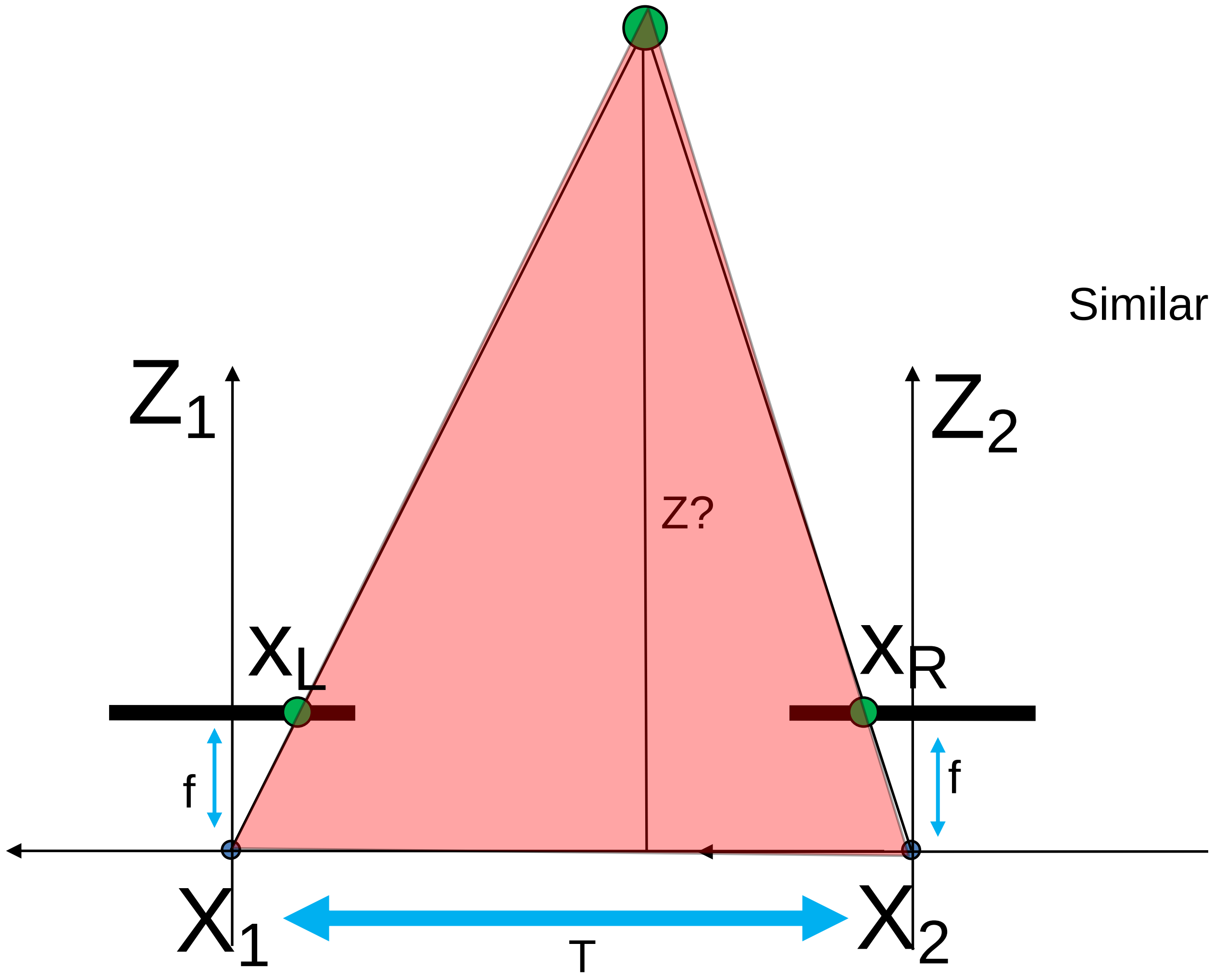
Stereoscopic Depth-Reconstruction



Stereoscopic Depth-Reconstruction

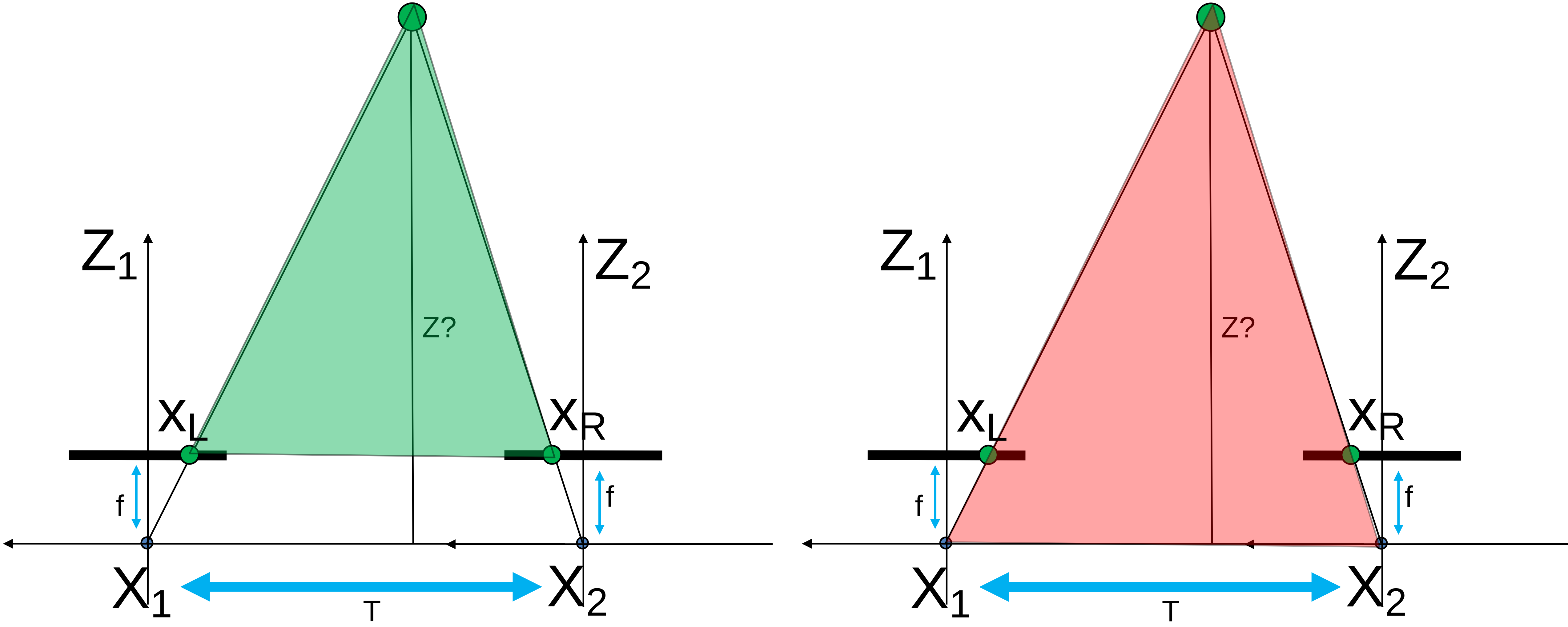


Stereoscopic Depth-Reconstruction

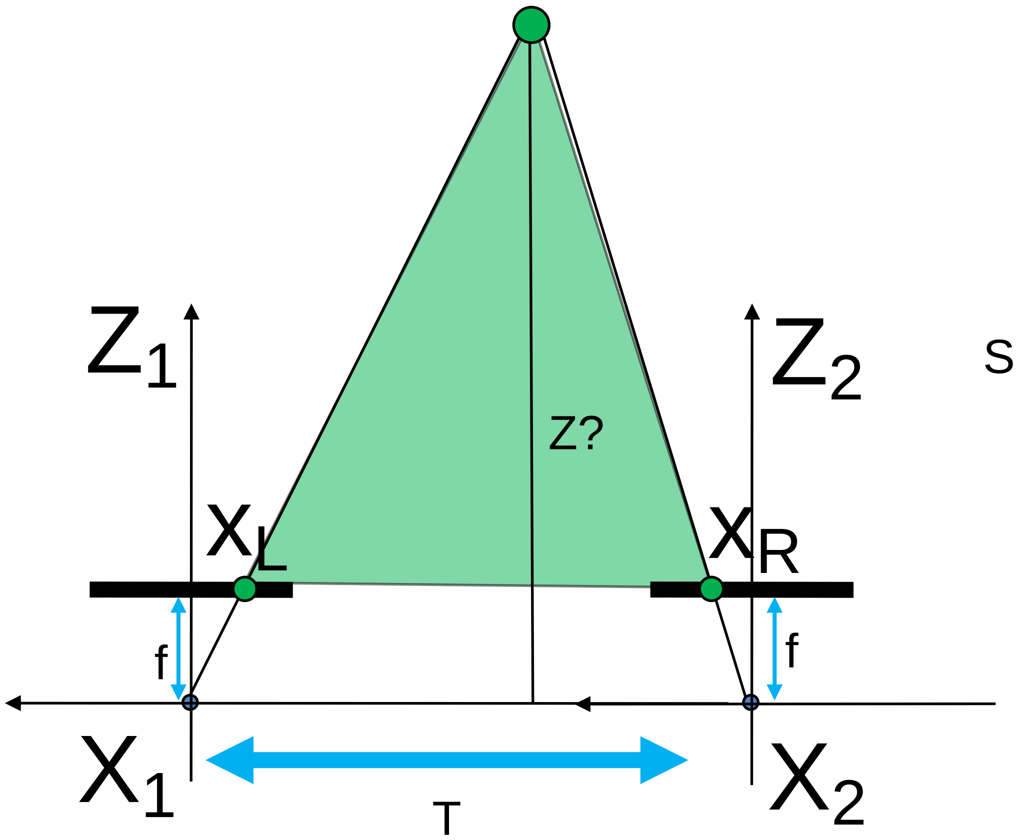


Similar Triangles

Stereoscopic Depth-Reconstruction



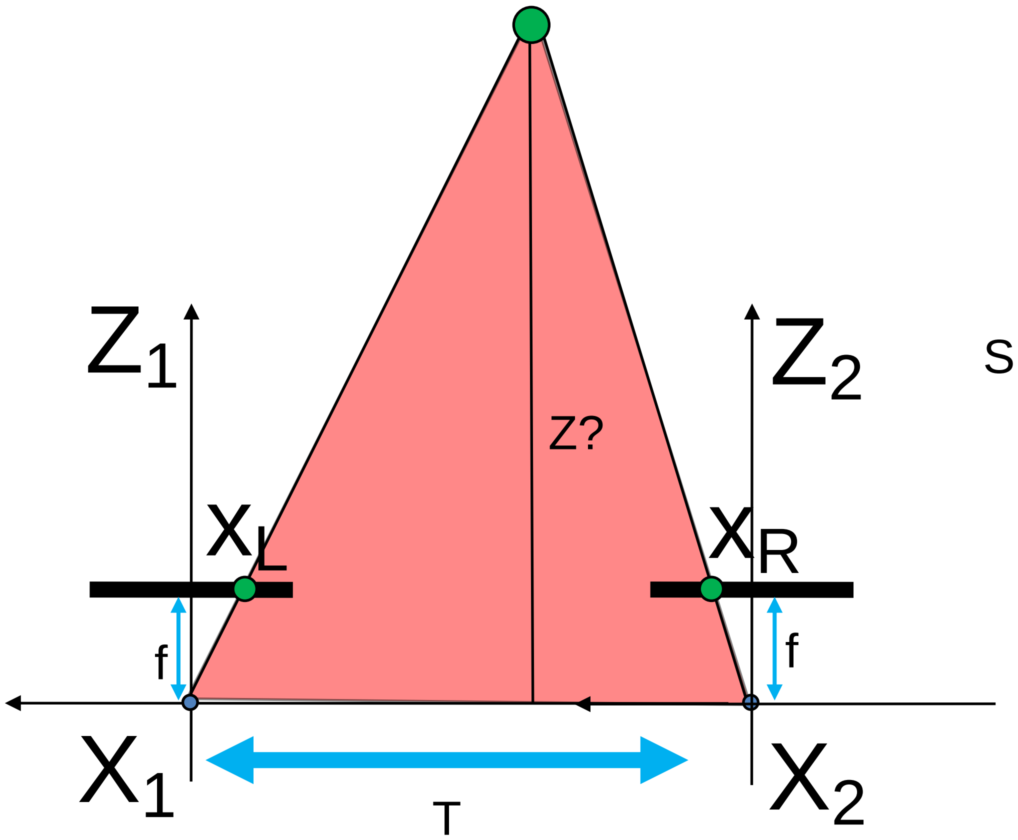
Stereoscopic Depth-Reconstruction



Similar Triangles:

$$\frac{T+X_R-X_L}{Z-f} =$$

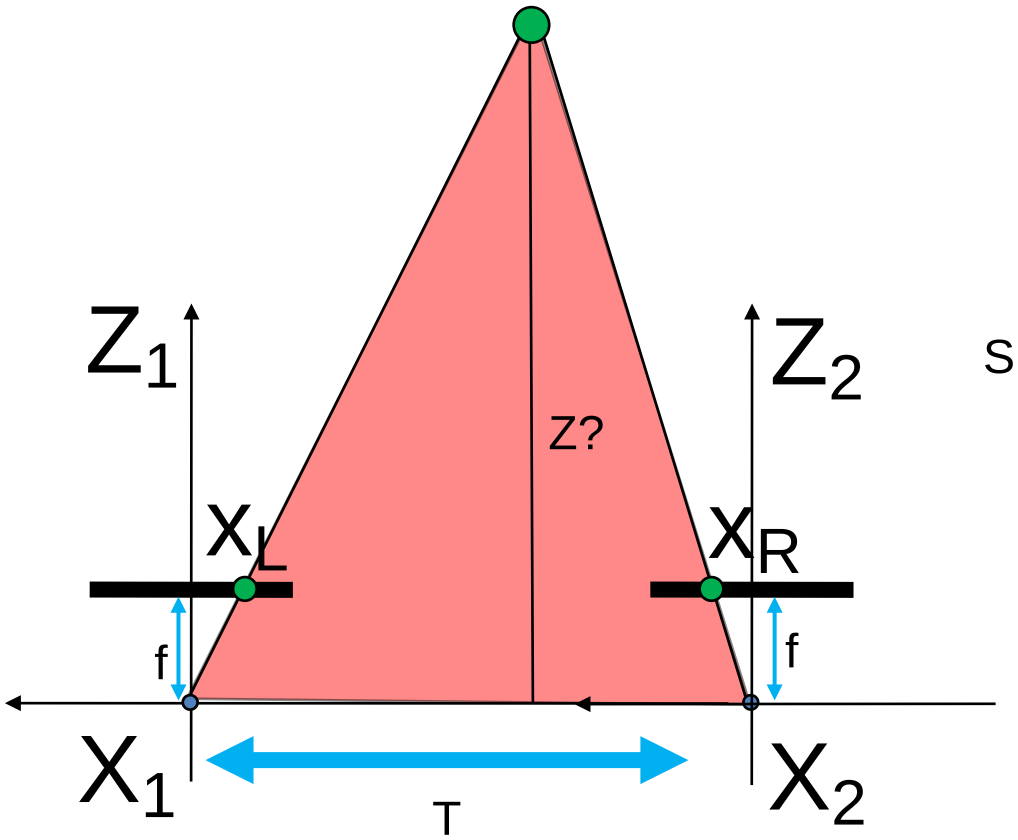
Stereoscopic Depth-Reconstruction



Similar Triangles:

$$\frac{T + X_R - X_L}{Z - f} = \frac{T}{Z}$$

Stereoscopic Depth-Reconstruction



Similar Triangles:

$$\frac{T+X_R-X_L}{Z-f} = \frac{T}{Z}$$

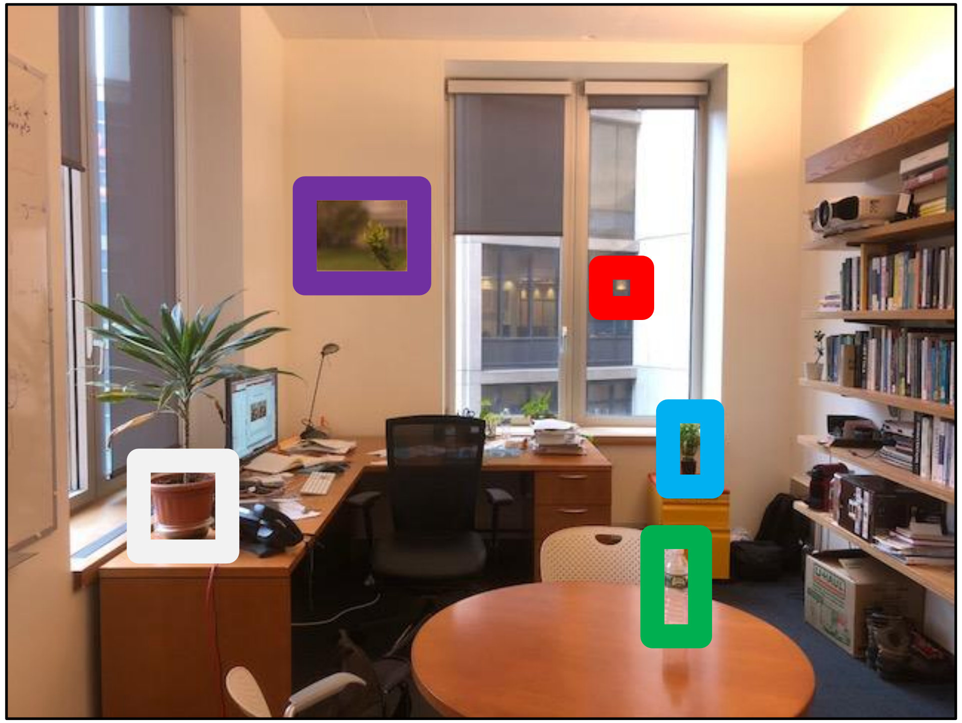
Solving for Z:

$$Z = f\,\frac{T}{X_L - X_R}$$

Disparity

So Z (depth) is proportional to Disparity.

Measuring Disparity

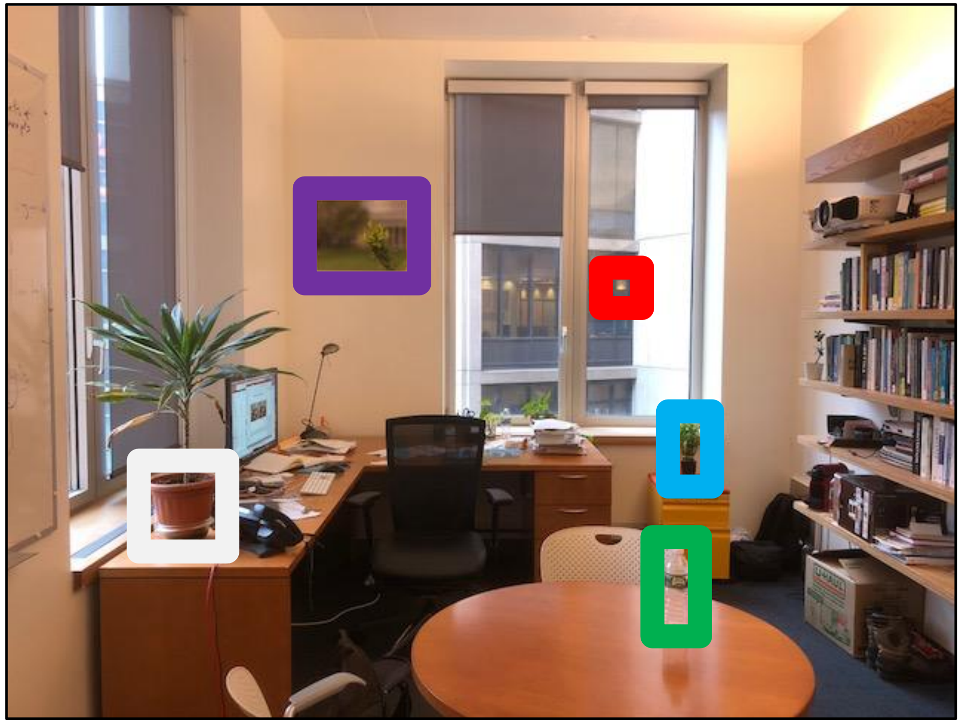


Left Image

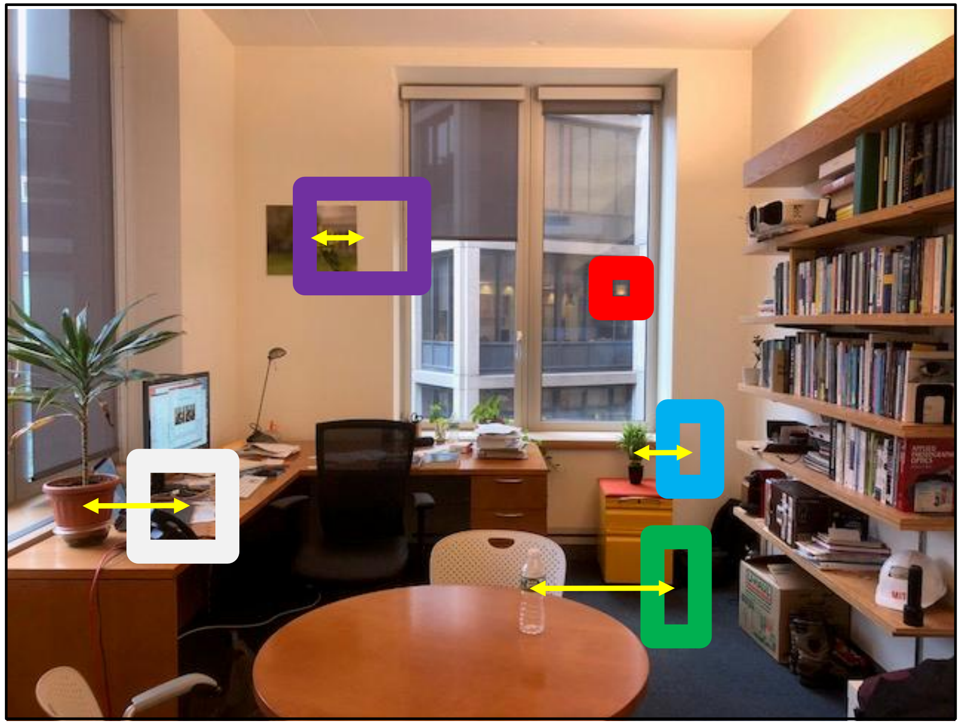
Right Image

(Taken by moving ~1m horizontally to the right)

Measuring Disparity



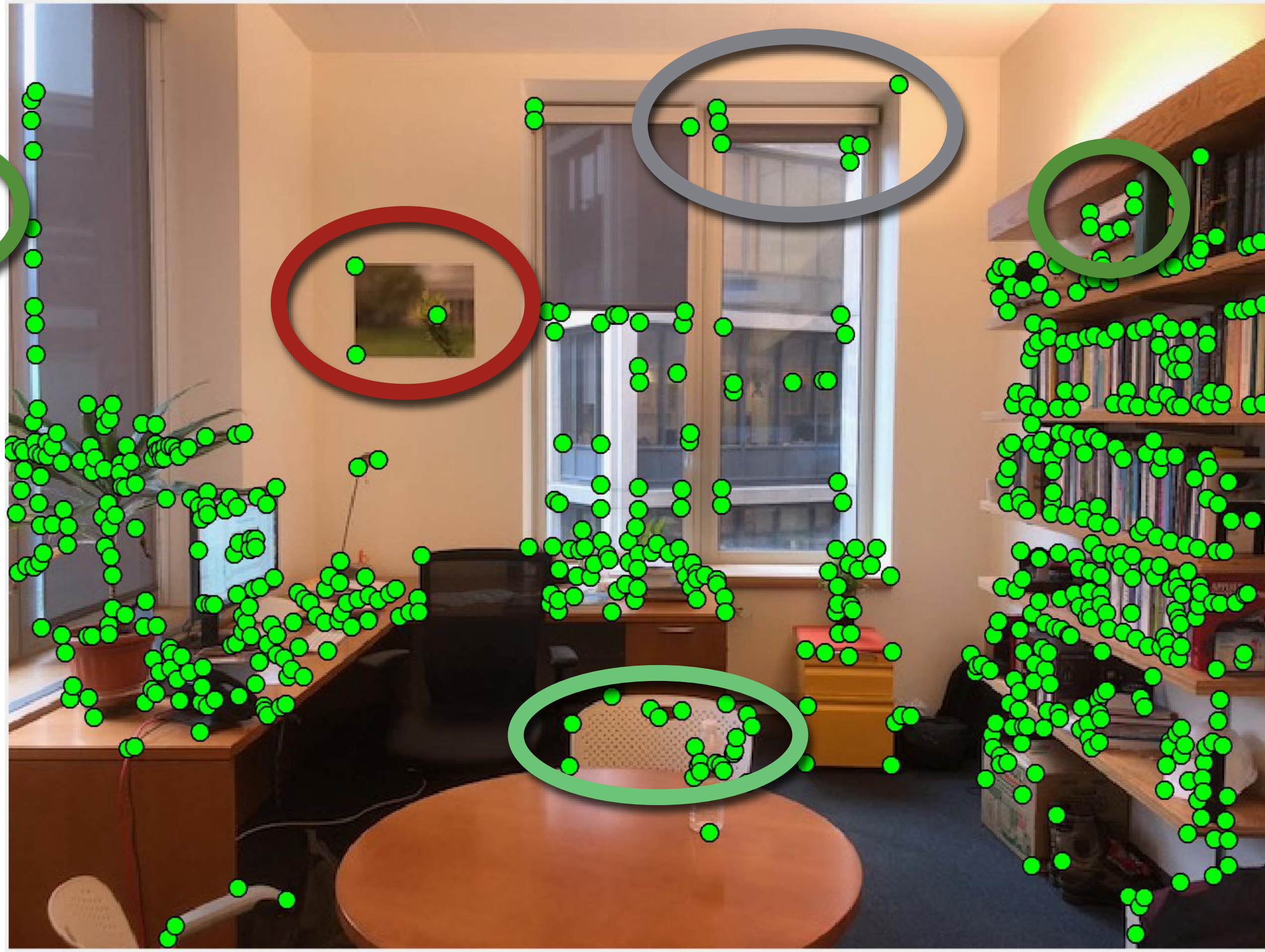
Left Image



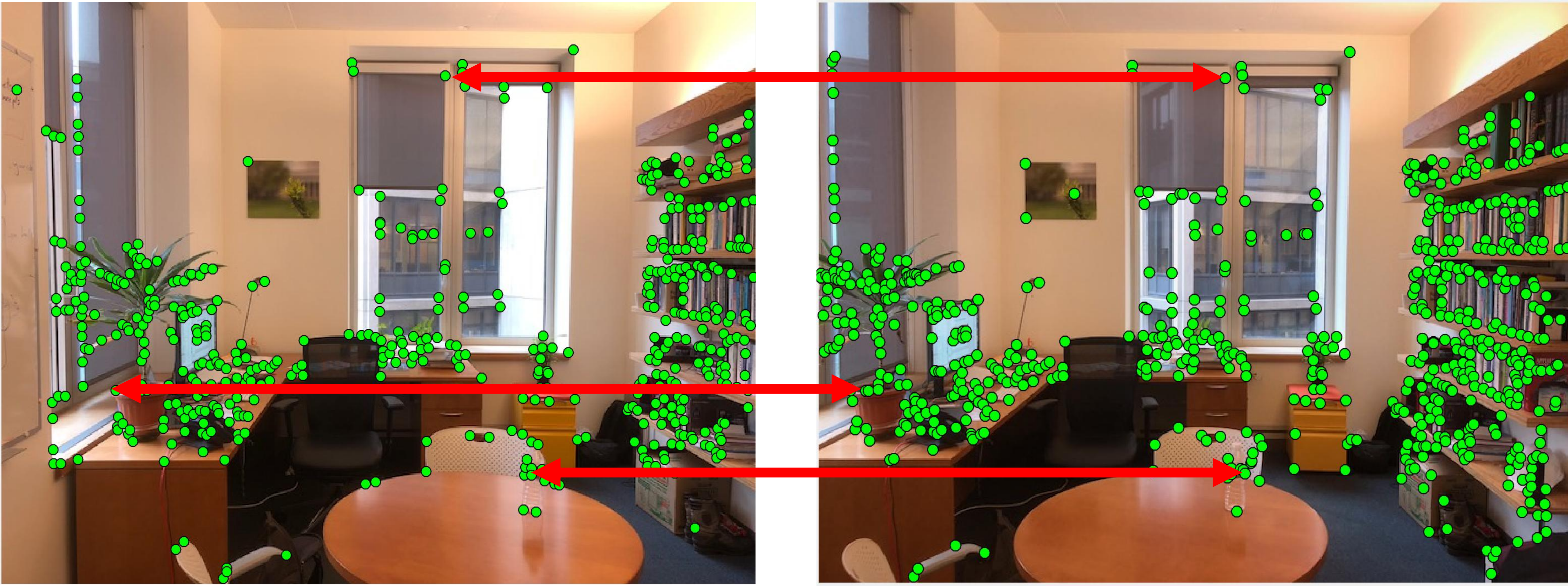
Right Image

JOHANNES KEPLER
UNIVERSITY LINZ

Example: SIFT Features



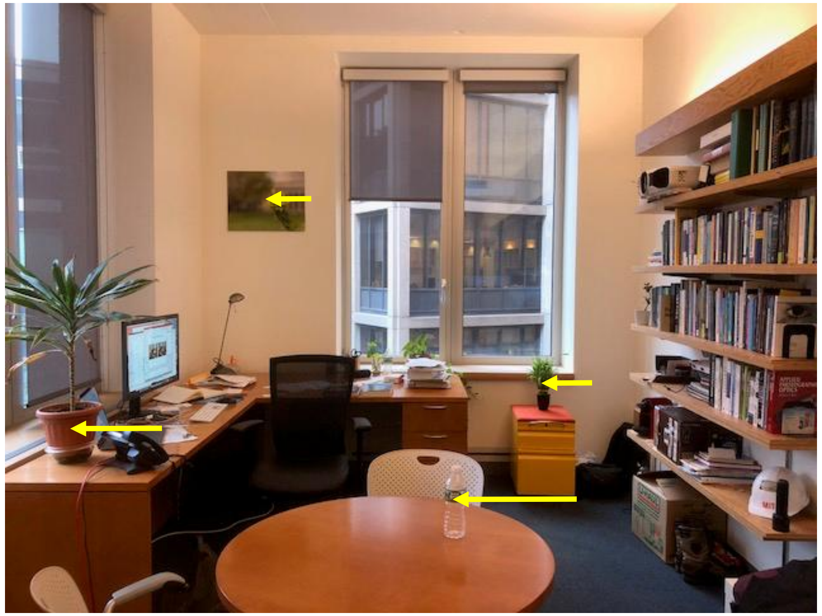
Example: SIFT Features



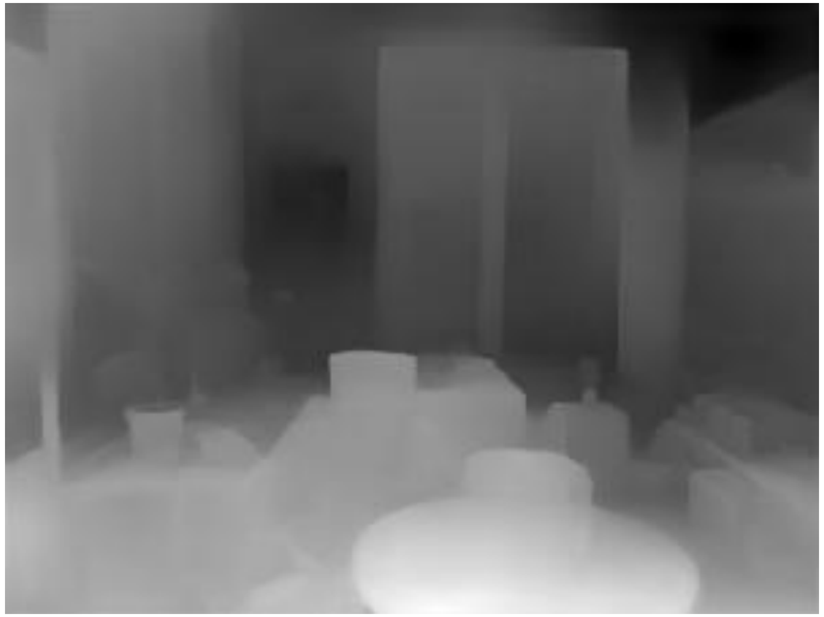
Disparity Maps

$I(x,y)$

$I'(x,y) = I(x+D(x,y), y)$



$D(x,y)$



$$Z(x,y) \propto \frac{1}{D(x,y)}$$

Structure-from-Motion

Can we estimate Depth (= Structure) and Camera Pose (= Motion) simultaneously from multiple Images?

Structure-from-Motion in a Nutshell (and Simplified)

$P_j(j = 1,...,n)$ 3D Points

$i = 1,..,m$ Images (Camera View)

$j = 1,...,n$ Points

This is what is unkown

$$p_{ij} = M_i \begin{pmatrix} P_j \\ 1 \end{pmatrix}$$

This is what is known

Structure-from-Motion in a Nutshell (and Simplified)

This is what is unkown

$P_j (j = 1, \ldots, n)$ 3D Points

$i = 1, .., m$ Images (Camera View)

$j = 1, \ldots, n$ Points

$$p_{ij} = M_i \begin{pmatrix} P_j \\ 1 \end{pmatrix}$$

This is what is known

This is an Equation-System: How many Unkowns vs. how many Equations, and when can it be solved?

Structure-from-Motion in a Nutshell (and Simplified)

$P_j (j = 1,...,n)$ 3D Points

$i = 1,..,m$ Images (Camera View)

$j = 1,...,n$ Points

This is what is unkown

$$p_{ij} = M_i \begin{pmatrix} P_j \\ 1 \end{pmatrix}$$

This is what is known

This is an Equation-System: How many Unkowns vs. how many Equations, and when can it be solved?

Unknowns: 12m Coefficients in all m M + 3n Coefficients in all n P , **Equations**: 2 Equations for all nm p

Structure-from-Motion in a Nutshell (and Simplified)

This is what is unkown

$P_j (j = 1,...,n)$ 3D Points

$i = 1,..,m$ Images (Camera View)

$j = 1,...,n$ Points

$$p_{ij} = M_i \begin{pmatrix} P_j \\ 1 \end{pmatrix}$$

This is what is known

This is an Equation-System: How many Unkowns vs. how many Equations, and when can it be solved?

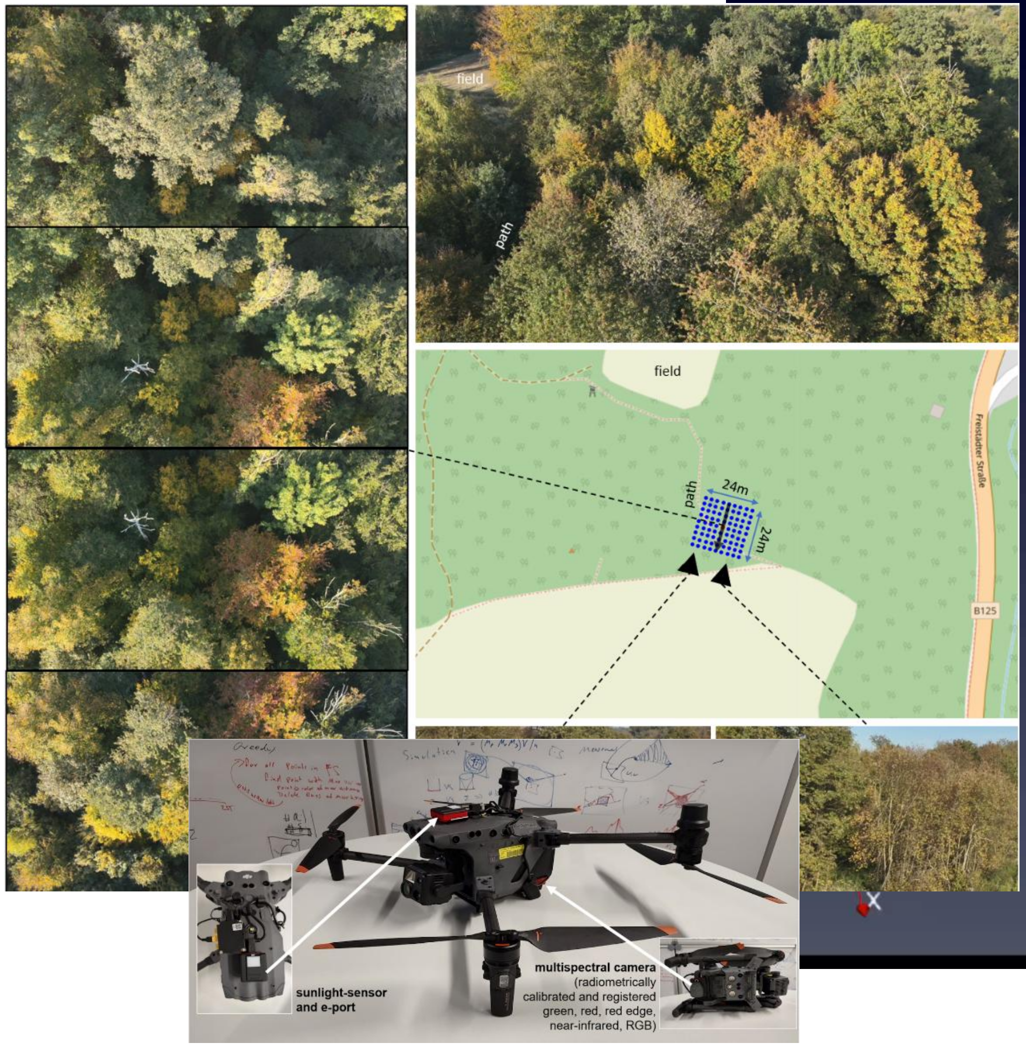
Unknowns: 12m Coefficients in all m M + 3n Coefficients in all n P , **Equations**: 2 Equations for all nm p

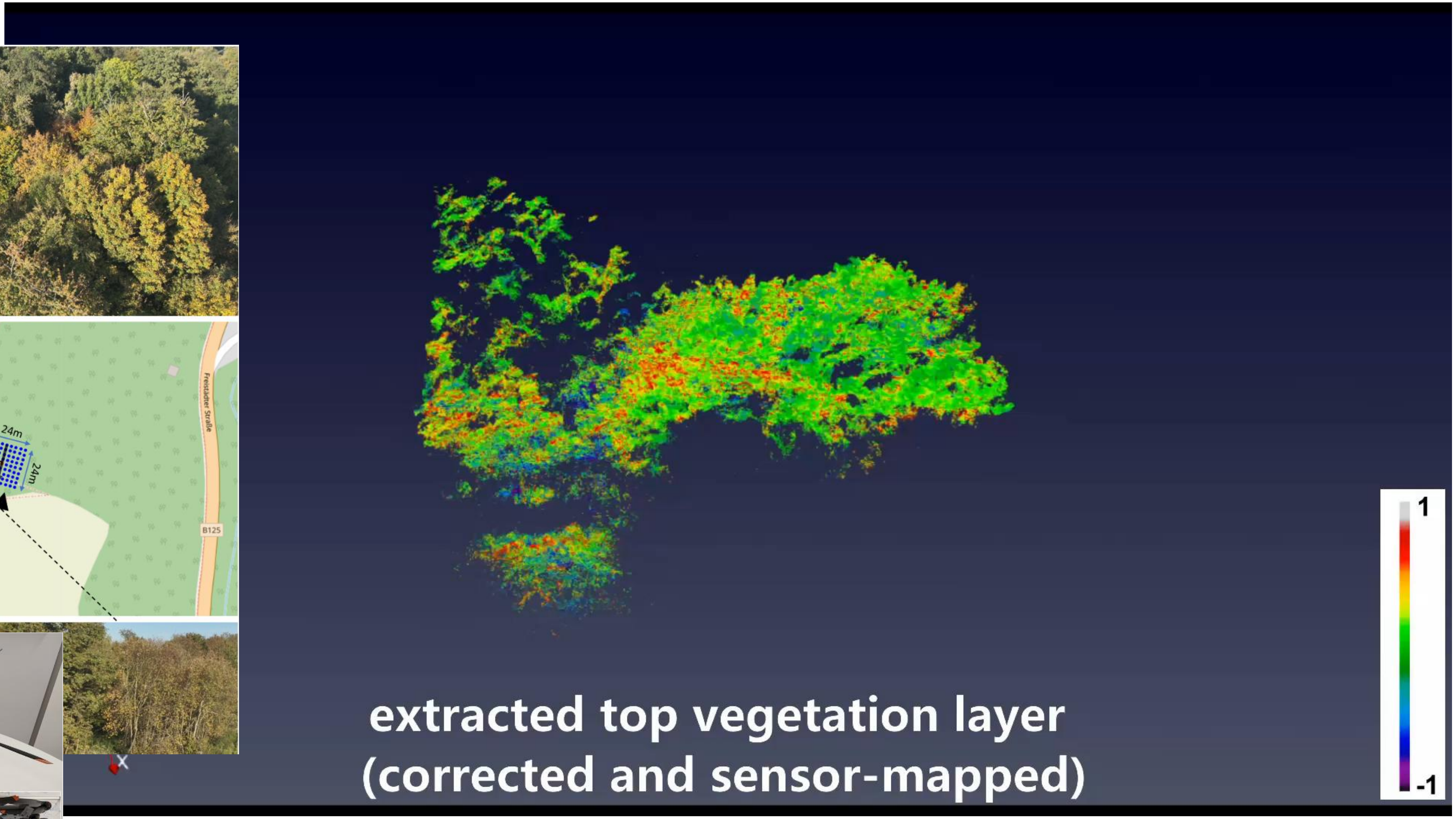
Solvable (simplified)**:** 2nm > 12m+3n (e.g., m=2 and n=25)

Structure-from-Motion

https://colmap.github.io/

Research Example: Reconstructing 3D Forest Health





JOHANNES KEPLER
UNIVERSITY LINZ

UFZ HELMHOLTZ
Zentrum für Umweltforschung

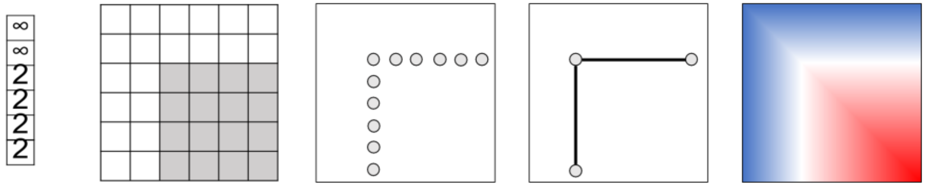
Journal of Remote Sensing
A SCIENCE PARTNER JOURNAL

Course Overview

CW	Topic	Date	Place	Lab
41	Introduction and Course Overview	07.10.2025	Zoom	Lab 1
42	Capturing Digital Images	14.10.2025	Zoom	Lab 2
43	Digital Image Processing	21.10.2025	Zoom	Assignment 1
44	Machine Learning	28.10.2025	Zoom	
45	Feature Extraction	04.11.2025	Zoom	Open Lab 1
46	Segmentation	11.11.2025	Zoom	Assignment 2
47	Optical Flow	18.11.2025	Zoom	Open Lab 2
48	Object Detection	25.11.2025	Zoom	Assignment 3
49	Multi-View Geometry	02.12.2025	Zoom	Open Lab 3
50	3D Vision	10.12.2025	Zoom	Assignment 4
3	Trends in Computer Vision	13.01.2026	Zoom	
4	Q&A	20.01.2026	Zoom	Open Lab 4
5	Exam	27.01.2026	HS1 (Linz), S1/S3 (Vienna), S5 (Bregenz)	
9	Retry Exam	24.02.2026	tba	

Next Week: 3D Vision

How to represent Depth

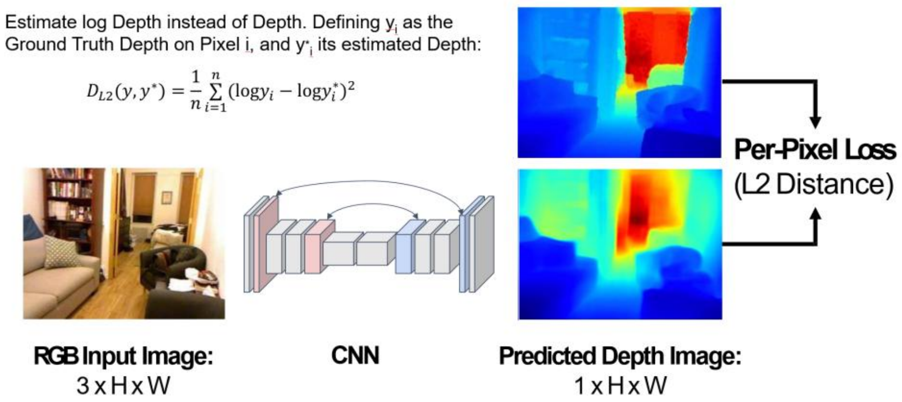


Depth Map | Voxel Grid | Point Cloud | Mesh | Implicit Surface

Predicting Depth Maps

Estimate log Depth instead of Depth. Defining y_i as the Ground Truth Depth on Pixel i, and y^*_i its estimated Depth:

$$D_{L2}(y, y^*) = \frac{1}{n}\sum_{i=1}^{n}(\log y_i - \log y_i^*)^2$$



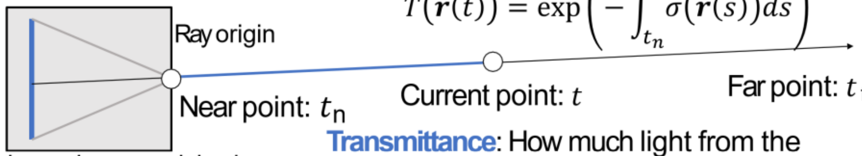
JKU JOHANNES KEPLER UNIVERSITY LINZ

Volume Rendering

Color observed by the camera given by volume rendering equation:

$$C(\boldsymbol{r}) = \int_{t_n}^{t_f} T(\boldsymbol{r}(t))\sigma(\boldsymbol{r}(t))\boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{d})dt$$

$$T(\boldsymbol{r}(t)) = \exp\left(-\int_{t_n}^{t}\sigma(\boldsymbol{r}(s))ds\right)$$



Parameterize each ray as origin plus direction: $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$
Volume Density is $\sigma(\boldsymbol{p}) \in [0,1]$
Color that a point **p** emits in direction **d** is $c(\boldsymbol{p}, \boldsymbol{d}) \in [0,1]^3$

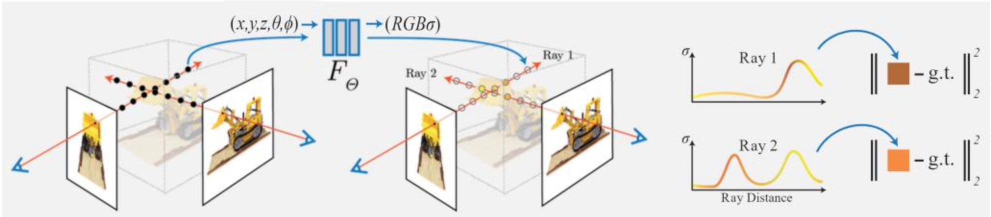
Transmittance: How much light from the current point will reach the camera?

Compute transmittance by accumulating volume density up to current point

JKU JOHANNES KEPLER UNIVERSITY LINZ

Neural Radiance Fields (NeRFs)

Fully-connected Network: Input Position p=x,y,z and Direction d=θ,Φ, and output Volume Density (σ) and RGB color



Mildenhall et al, "Representing Scenes as Neural Radiance Fields for View Synthesis", ECCV 2020

JKU JOHANNES KEPLER UNIVERSITY LINZ

JKU JOHANNES KEPLER UNIVERSITY LINZ

Thank You