Computer Vision (CSE3010)

Dr. Susant Kumar Panigrahi
Assistant Professor
School of Electrical & Electronics Engineering

VIT BHOPAL

Module-1 Syllabus

Digital Image Formation And Low Level Processing:

- Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, Fourier Transform,
- Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.

Module-2 Syllabus

Depth Estimation And Multi-Camera Views:

Depth Estimation and Multi-Camera Views: Perspective, Binocular Stereopsis: Camera and Epipolar Geometry; Homography, Rectification, DLT, RANSAC, 3-D reconstruction framework; Auto-calibration. apparel.

Module-3 Syllabus

Feature Extraction And Image Segmentation:

- **Feature Extraction**: Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT.
- **Image Segmentation:** Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection.

Module-4 Syllabus

Pattern Analysis And Motion Analysis:

- **Pattern Analysis**: Clustering: K-Means, K-Medoids, Mixture of Gaussians, Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Classifiers: Bayes, KNN, ANN models;
- **Dimensionality Reduction**: PCA, LDA, ICA; Non-parametric methods. Motion Analysis: Background Subtraction and Modelling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

Module-5 Syllabus

Shape From X:

Light at Surfaces; Phong Model; Reflectance Map;

Albedo estimation; Photometric Stereo; Use of Surface Smoothness

Constraint; Shape from Texture, color, motion and edges.

Guest Lecture on Contemporary Topics

Text Books

1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.
2. Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 2003.

Reference Book(s):

1. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison- Wesley, 1992.
2. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004.
3. K. Fukunaga; Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990.

Required Tools/Software/IDLE:

1. Python/jupyter-notebook/google-colab
2. OpenCV
3. MATLAB

Indicative List of Experiments:

1. Implement image preprocessing and Edge
2. Implement camera calibration methods
3. Implement Projection
4. Determine depth map from Stereo pair
5. Construct 3D model from Stereo pair
6. Implement Segmentation methods
7. Construct 3D model from defocus image
8. Construct 3D model from Images
9. Implement optical flow method
10. Implement object detection and tracking from video
11. Face detection and Recognition
12. Object detection from dynamic Background for Surveillance
13. Content based video retrieval
14. Construct 3D model from single image

Computer Vision

Unit – 02

Depth Estimation And Multi-Camera Views (Linear Camera Model and Camera Calibration)

Standing on the shoulder of Giants: Ref: Few Slides borrowed from:

1. Prof. Shree Nayar, *First Principles of Computer Vision is a lecture series*.
2. Prof. Mubarak Shah, *Computer Vision Video Lectures.*

Camera Model and Camera Calibration

- ✓ One of the key problems in computer vision is to recover the **3D model** from the **2D image**.
- ✓ Methods to find a camera's internal and external parameters are called **camera modeling**.

Topics:

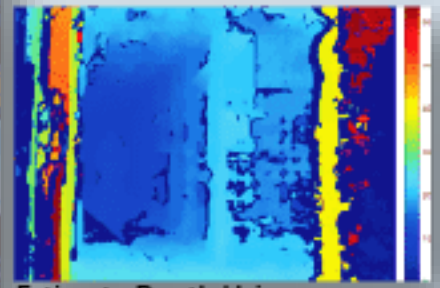
1. Linear Camera Model: (Forward Imaging Model)
   - ✓ It takes from point in 3d to its corresponding pixel in 2D image.
   - ✓ Needs to be linear model for ease in mathematical manipulation.
2. Camera Calibration
3. Extracting Intrinsic and Extrinsic Parameters
4. Shape of 3D Model: Multi-Camera View (Stereo) for Depth Estimation.

Camera Model and Camera Calibration

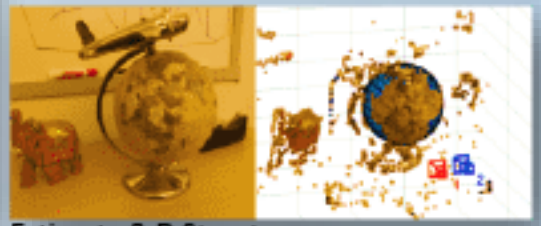
- ✓ Geometric camera calibration, also referred to as **camera resectioning**, estimates the parameters of a lens and image sensor of an image or video camera.
- ✓ You can use these parameters to correct for lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene.
- ✓ These tasks are used in applications such as machine vision to detect and measure objects. They are also used in robotics, for navigation systems, and 3-D scene reconstruction.

Examples of what you can do after calibrating your camera:

Remove Lens Distortion

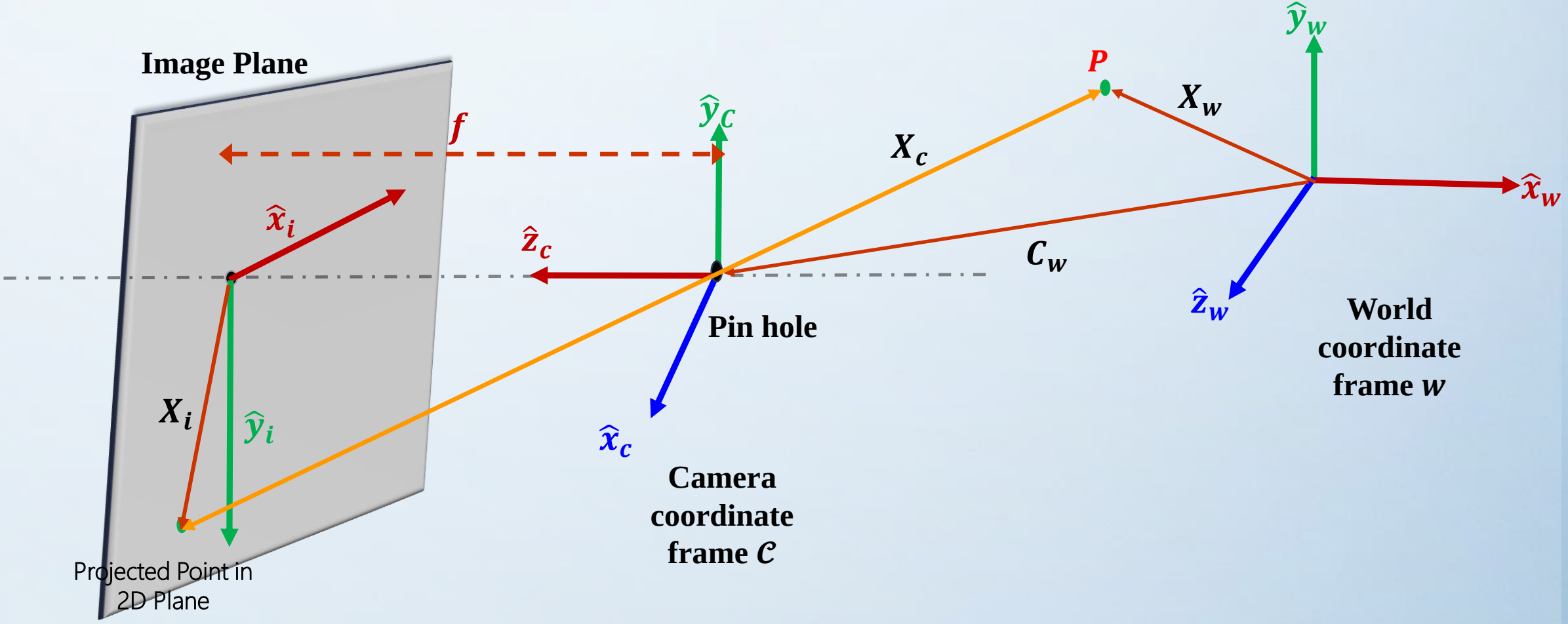

Estimate Depth Using a Stereo Camera

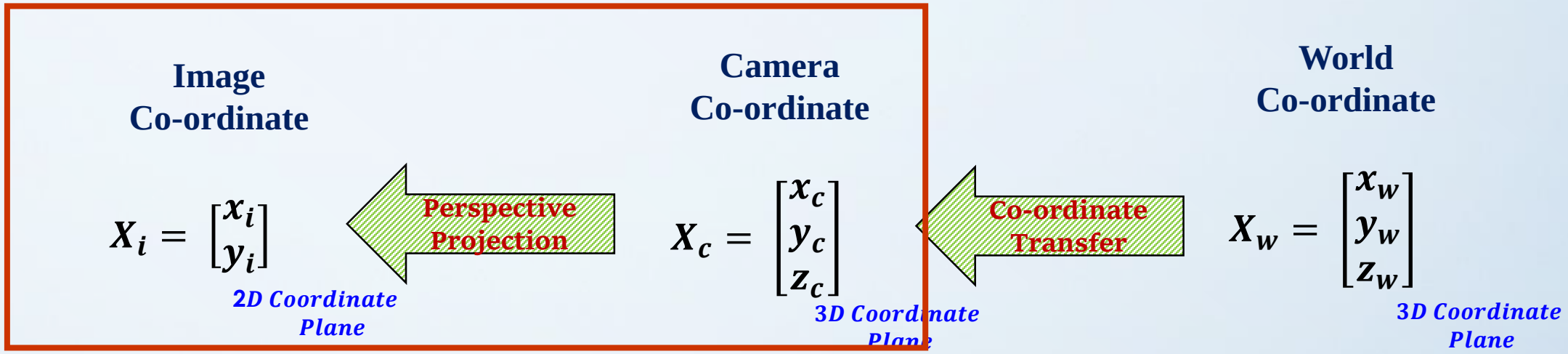
Measure Planar Objects


Estimate 3-D Structure from Camera Motion

Source: https://in.mathworks.com/help/vision/ug/camera-calibration.html

Forward Imaging (Camera) Modeling: 3D to2D





Perspective Projection:

Camera to Image Coordinate:

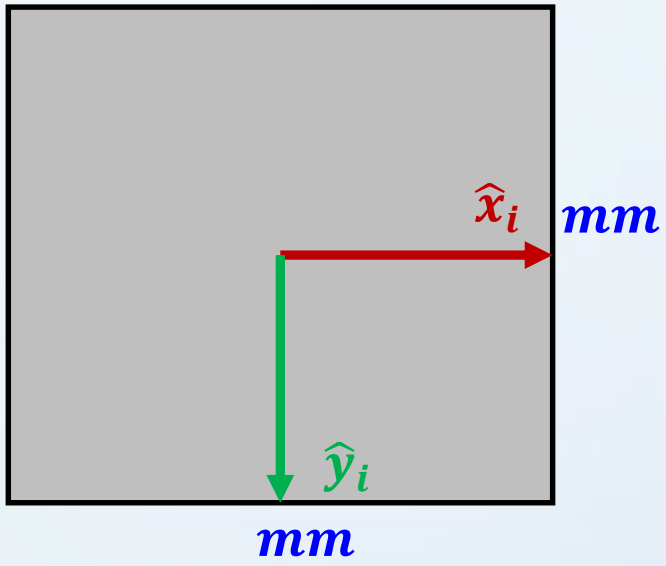
We know that:

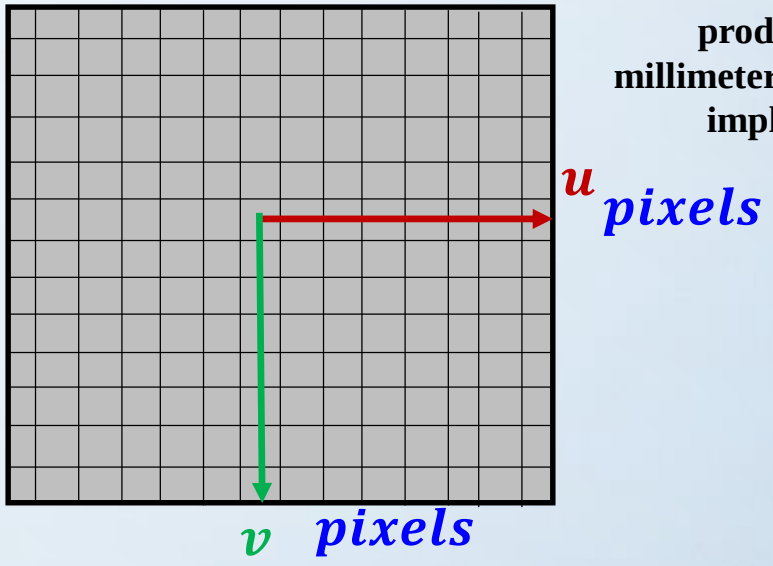
$$\frac{x_i}{f} = \frac{x_c}{z_c} \quad , \quad \frac{y_i}{f} = \frac{y_c}{z_c}$$

$$x_i = f\frac{x_o}{z_o} \quad , \quad y_i = f\frac{y_o}{z_o}$$

Image Plane to Image Sensor Mapping: (2D to 2D Mapping)

In image plane we measure a position in millimeters.





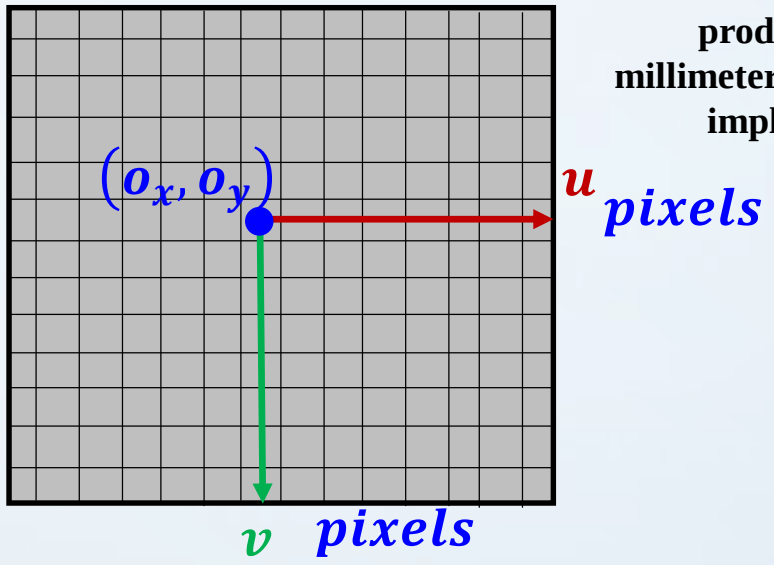
But sensors in the cameras produce pixels per millimeter. A single rectangle implies one pixel.

✓ If m_x and m_y are the pixel densities (pixels/mm) in the x- and y-directions, respectively, then the pixel coordinates can be written as:

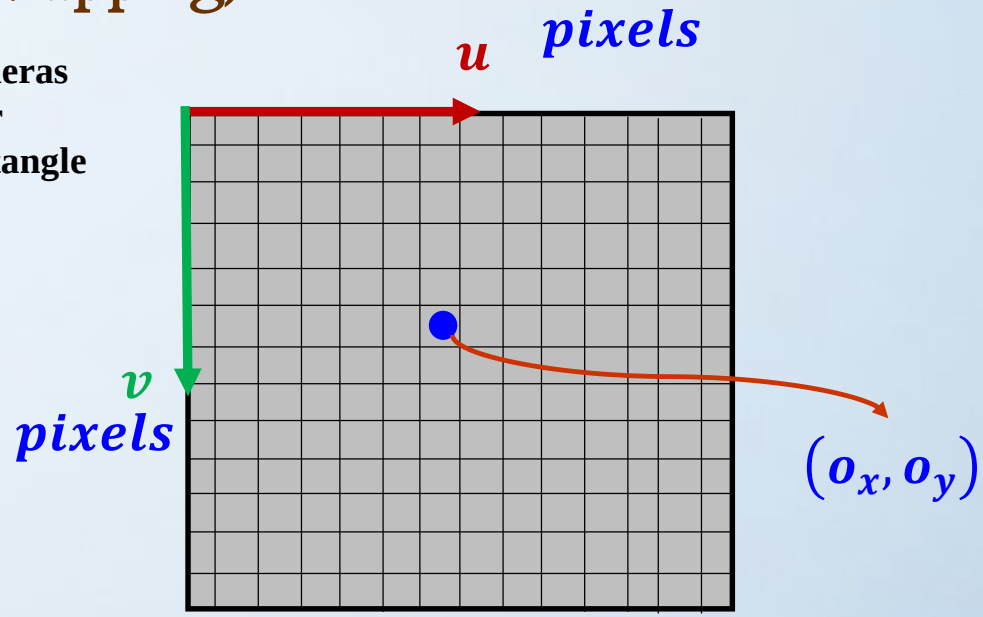
$$u = m_x x_i = m_x f \frac{x_x}{z_c}$$

$$\& \; v = m_y y_i = m_y f \frac{y_c}{z_c}$$

Image Plane to Image Sensor Mapping: (2D to 2D Mapping)



But sensors in the cameras produce pixels per millimeter. A single rectangle implies one pixel.



✓ If pixel (o_x, o_y) is the principal point where the optical axis pierces the sensor, then:

$$u = m_x x_i = m_x f \frac{x_x}{z_c} + o_x$$

$$\& \; v = m_y y_i = m_y f \frac{y_c}{z_c} + o_y \quad \text{assuming } f_x = m_x f \text{ and } f_y = m_y f$$

✓ Rewriting the equation:

$$u = f_x \frac{x_c}{z_c} + o_x \text{ and } v = f_y \frac{y_c}{z_c} + o_y$$

Image Plane to Image Sensor Mapping: (2D to 2D Mapping)

✓ The unknowns :

$(f_x, f_y ; o_x, o_y)$: Intrinsic parameters of camera. They represent camera's internal geometry.

Equations for perspective geometry (as derived in previous slide) are non-linear. It is convenient to express them as linear equation using Homogenous Coordinate System.

Homogenous Coordinate System

- ✓ The homogenous representation of 2D point $\mathcal{U} = (u, v)$ in a 3D point $\tilde{\mathcal{U}} = (\tilde{u}, \tilde{v}, \tilde{w})$, the third coordinate $\tilde{w} \neq 0$ is fictitious such that:

$$u = \frac{\tilde{u}}{\tilde{w}} \text{ and } v = \frac{\tilde{v}}{\tilde{w}}$$

$$\mathcal{U} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{w}\, u \\ \tilde{w}\, v \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \tilde{\mathcal{U}}$$

- ✓ The homogenous representation of 3D point $X = (x, y, z)$ in a 4D point $\tilde{X} = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w})$, the fourth coordinate $\tilde{w} \neq 0$ is fictitious such that:

$$x = \frac{x}{\tilde{w}}, y = \frac{y}{\tilde{w}} \text{ and } z = \frac{\tilde{z}}{\tilde{w}}$$

$$X = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{w}\, x \\ \tilde{w}\, y \\ \tilde{w}\, z \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{w} \end{bmatrix} = \tilde{X}$$

Perspective Projection (Homogenous Coordinate System):

✓ The perspective projection:

$$u = f_x \frac{x_c}{z_c} + o_x \qquad \& \qquad v = f_y \frac{y_c}{z_c} + o_y$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \equiv \tilde{u} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} z_c u \\ z_c v \\ z_c \end{bmatrix} = \begin{bmatrix} f_x x_c + o_x z_c \\ f_y x_c + o_y z_c \\ z_c \end{bmatrix}$$

Linear Model for Perspective Projection

$$\begin{bmatrix} z_c u \\ z_c v \\ z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

All internal parameters of camera

Calibration Matrix

$$k = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Maps camera 3D points to 2D points in image plane

The Intrinsic Parameter Matrix

$$M_{int} = [k|0]$$

Perspective Projection Equation

$$\tilde{u} = [k|0]\tilde{X}_c = M_{int}\tilde{X}_c$$

Extrinsic Parameters: [3D to 3D: Mapping from World Coordinate to Camera Coordinate]

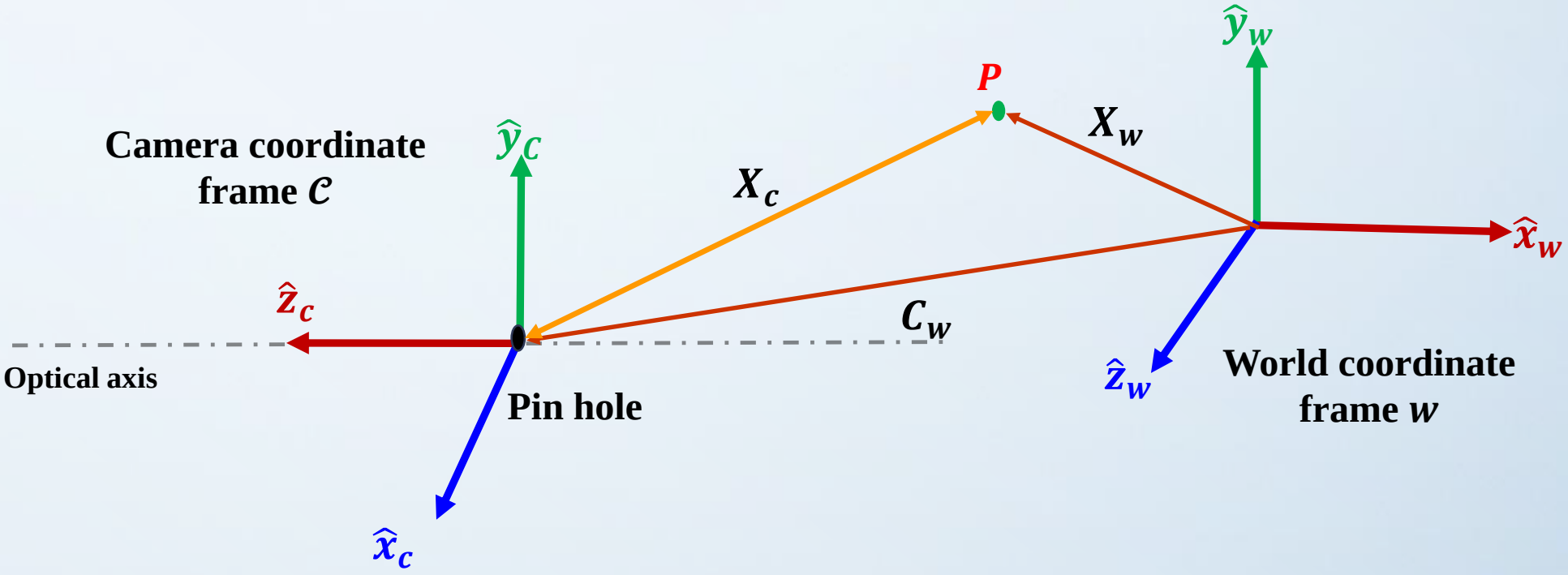


Image Co-ordinate

$$X_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

2D Coordinate Plane

← Perspective Projection

Camera Co-ordinate

$$X_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

3D Coordinate Plane

← Co-ordinate Transfer

World Co-ordinate

$$X_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

3D Coordinate Plane

Extrinsic Parameters: [3D to 3D: Mapping from World Coordinate to Camera Coordinate]

Position C_w and orientation R of the camera in the world coordinate frame **w** are the camera's extrinsic parameter.

✓ Rotation matrix:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Row 1: Direction of $\hat{x}_c$ in the world coordinate.
Row 2: Direction of $\hat{y}_c$ in the world coordinate.
Row 3: Direction of $\hat{z}_c$ in the world coordinate.

✓ Orientation / Rotation matrix R is orthonormal: [A property which will be useful in camera calibration]

$$R^T R = R R^T = I$$

$$\& \qquad R^{-1} = R^T$$

Given the extrinsic parameters ($\boldsymbol{C_w}$, $\boldsymbol{R}$) of camera the camera-centric location of the point $\boldsymbol{P}$ in the world coordinate frame.

$$X_c = R(X_w - C_w)$$

Extrinsic Parameters: [3D to 3D: Mapping from World Coordinate to Camera Coordinate]

✓ Mapping:

$$X_c = R(X_w - C_w) = RX_w - RC_w = RX_w + t$$

where: $\boldsymbol{t = -CR_w}$ **(Translation Matrix)**

Linear Model in Homogenous Coordinate

$$\tilde{X}_C = \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

The complete mapping (non-linear) in matrix representation

$$X_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

3D to 3D mapping equation

$$\tilde{X}_c = M_{ext}\tilde{X}_w$$

Extrinsic Matrix

$$M_{ext} = \begin{bmatrix} R_{3\times3} & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward Imaging (Camera) Modeling: (The Projection Matrix)

Linear Model for Perspective Projection

$$\begin{bmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{w}\end{bmatrix} = \begin{bmatrix}f_x & 0 & o_x & 0\\ 0 & f_y & o_y & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x_c\\ y_c\\ z_c\\ 1\end{bmatrix}$$

Perspective Projection Equation (Camera to Pixel)

$$\tilde{U} = [k|0]\tilde{X}_c = M_{int}\tilde{X}_c$$

Linear Model in Homogenous Coordinate

$$\tilde{X}_C = \begin{bmatrix}x_c\\ y_c\\ z_c\\ 1\end{bmatrix} = \begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\ r_{21} & r_{22} & r_{23} & t_y\\ r_{31} & r_{32} & r_{33} & t_z\\ 0 & 0 & 0 & 1\end{bmatrix}$$

3D to 3D mapping equation (World to camera)

$$\tilde{X}_c = M_{ext}\tilde{X}_w$$

✓ Combining above equations, we get the full projection matrix $\boldsymbol{P}$:

$$\tilde{U} = M_{int}\, M_{ext}\, \tilde{X}_w = \boldsymbol{P}\tilde{X}_w$$

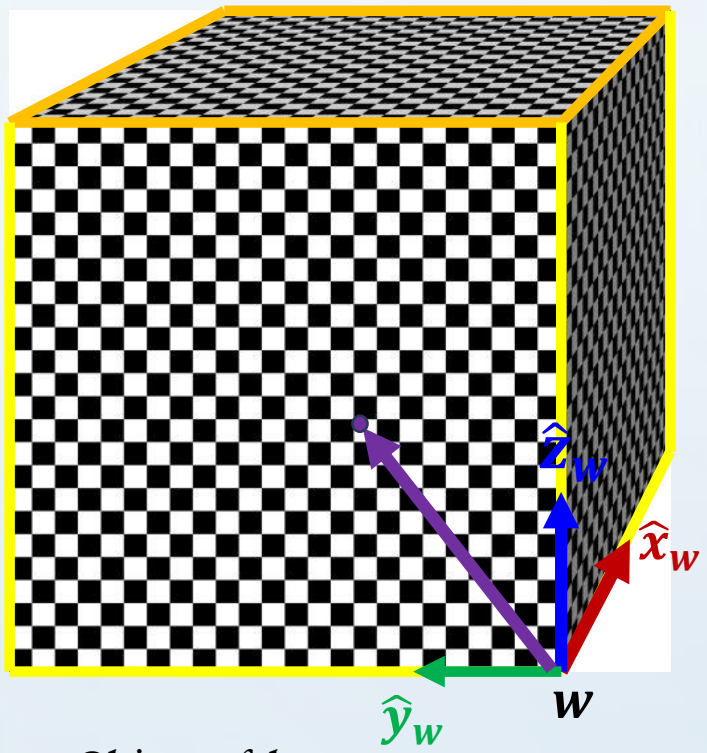
Forward Camera Model

$$\begin{bmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{w}\end{bmatrix} = \begin{bmatrix}P_{11} & P_{12} & P_{13} & P_{14}\\ P_{21} & P_{22} & P_{23} & P_{24}\\ P_{31} & P_{32} & P_{33} & P_{34}\end{bmatrix}\begin{bmatrix}x_w\\ y_w\\ z_w\\ 1\end{bmatrix}$$

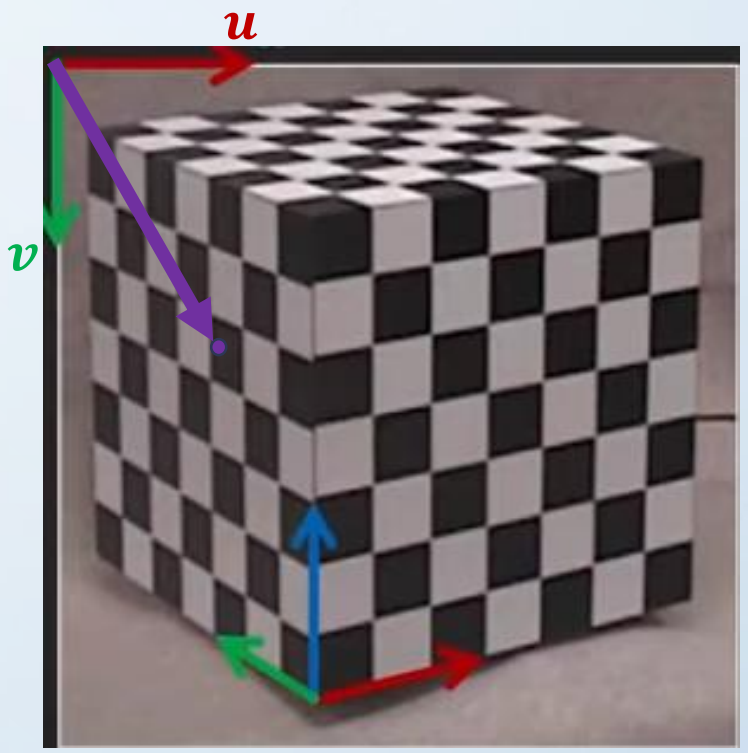

Camera Calibration

Camera Calibration Procedure

Step 1: Capture an image of an object with known geometry.



Object of known geometry



Captured Image in 2D Plane

- $X_w = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix}$ inches

- $\boldsymbol{u} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 56 \\ 115 \end{bmatrix}$ Pixels

Camera Calibration Procedure

Step 2: Similarly find few other correspondence points 2D image with world coordinate.

Step 3: For each corresponding points ***i*** in the scene and image.

Unknown

Known

$$\begin{bmatrix} u^{(i)} \\ v^{(i)} \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} x_w^{(i)} \\ y_w^{(i)} \\ z_w^{(i)} \\ 1 \end{bmatrix}$$

Known

Expanding the matrix as linear equations:

$$u^{(i)} = \frac{p_{11}x_w^{(i)} + p_{12}y_w^{(i)} + p_{13}z_w^{(i)} + p_{14}}{p_{31}x_w^{(i)} + p_{32}y_w^{(i)} + p_{33}z_w^{(i)} + p_{34}}$$

$$v^{(i)} = \frac{p_{21}x_w^{(i)} + p_{22}y_w^{(i)} + p_{23}z_w^{(i)} + p_{24}}{p_{31}x_w^{(i)} + p_{32}y_w^{(i)} + p_{33}z_w^{(i)} + p_{34}}$$

Camera Calibration Procedure

Step 4: Rearranging the terms.

$$\begin{bmatrix} x_w^{(1)} & y_w^{(1)} & z_w^{(1)} & 1 & 0 & 0 & 0 & 0 & -u_1 x_w^{(1)} & -u_1 y_w^{(1)} & -u_1 z_w^{(1)} & -u_1 \\ 0 & 0 & 0 & 0 & x_w^{(1)} & y_w^{(1)} & z_w^{(1)} & 1 & -v_1 x_w^{(1)} & -v_1 y_w^{(1)} & -v_1 z_w^{(1)} & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_w^{(i)} & y_w^{(i)} & z_w^{(i)} & 1 & 0 & 0 & 0 & 0 & -u_i x_w^{(i)} & -u_i y_w^{(i)} & -u_i z_w^{(i)} & -u_i \\ 0 & 0 & 0 & 0 & x_w^{(i)} & y_w^{(i)} & z_w^{(i)} & 1 & -v_i x_w^{(i)} & -v_i y_w^{(i)} & -v_i z_w^{(i)} & -v_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_w^{(n)} & y_w^{(n)} & z_w^{(n)} & 1 & 0 & 0 & 0 & 0 & -u_n x_w^{(n)} & -u_n y_w^{(n)} & -u_n z_w^{(n)} & -u_n \\ 0 & 0 & 0 & 0 & x_w^{(n)} & y_w^{(n)} & z_w^{(n)} & 1 & -v_n x_w^{(n)} & -v_n y_w^{(n)} & -v_n z_w^{(n)} & -v_n \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$\boldsymbol{A}$

Known

$\boldsymbol{P}$

Unknown

Step 5: Solve for $\boldsymbol{P}$

$$\boldsymbol{AP} = \mathbf{0}$$

Scale of a projection matrix

Projection matrix acts on homogenous coordinate:

We know that: $\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} \equiv k \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix}$ ($k \neq 0$ is any constant)

That is our projection matrix:

$$\begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \equiv k \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

Therefore the projection matrix ***P*** and ***kP*** produces same homogenous pixel coordinates.

Scaling the projection matrix, implies simultaneously scaling the world and the camera, which does not change the image

Least square solution for $\boldsymbol{P}$

- ✓ Set the scale so that: $\|\boldsymbol{P}\|^2 = 1$
- ✓ We want $\boldsymbol{AP}$ as close zero and $\|\boldsymbol{P}\|^2 = 1$.

$$\min_P \|\boldsymbol{AP}\|^2 \quad \text{such that } \|\boldsymbol{P}\|^2 = 1$$

$$\min_P (P^T A^T A P) \text{ such that } P^T P = 1$$

Define a loss function as $\mathcal{L}(P, \lambda)$:

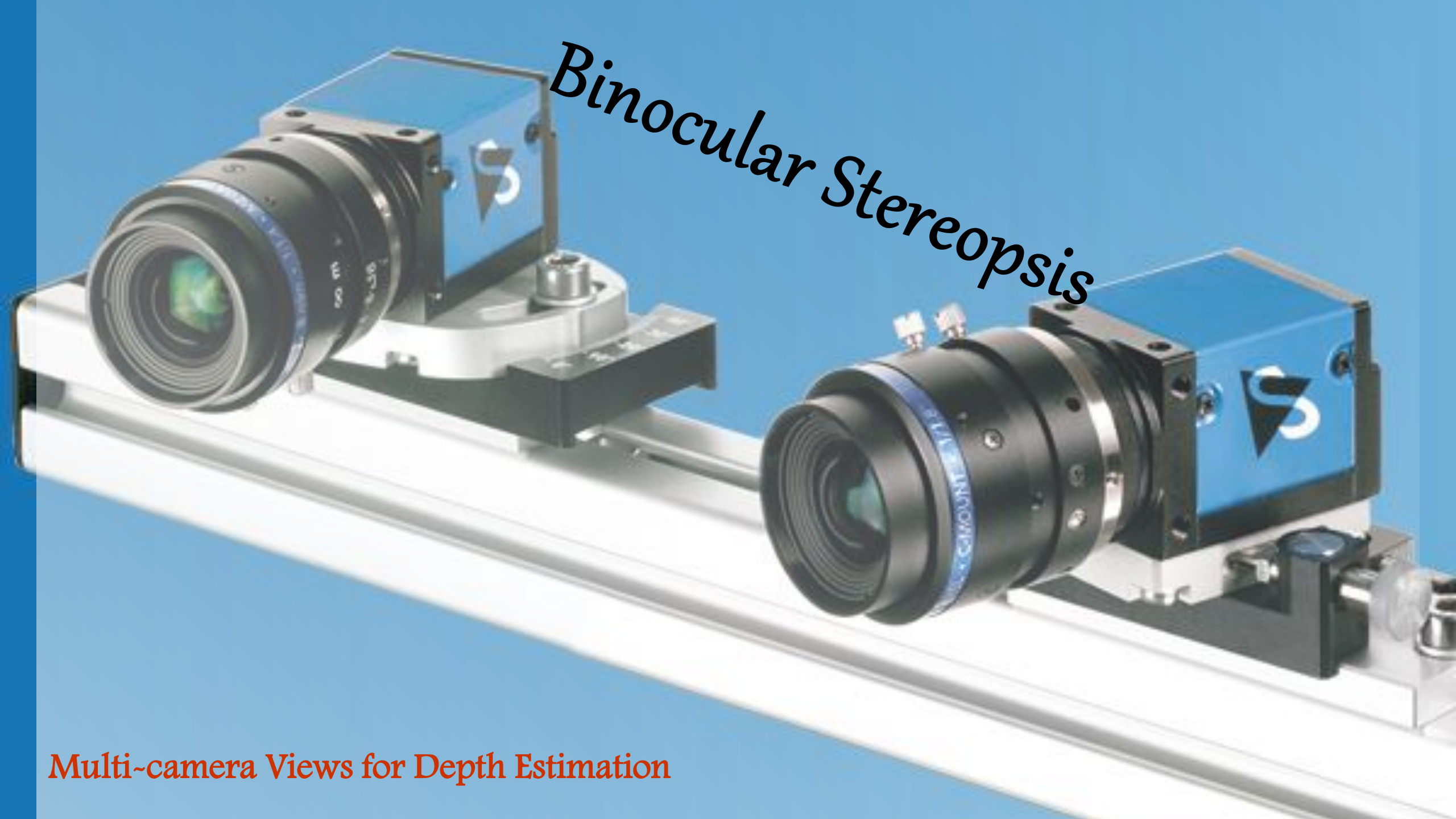
$$\mathcal{L}(P, \lambda) = P^T A^T A P - \lambda(P^T P - 1)$$

Taking derivative of $\mathcal{L}(P, \lambda)$ w.r.t. P:

$$2A^T A P - 2\lambda P = 0$$

Finally the solution for the $\boldsymbol{P}$ is the Eigen vector corresponds to the smallest Eigenvalue λ of the matrix $\boldsymbol{A^T A}$.

Binocular Stereopsis

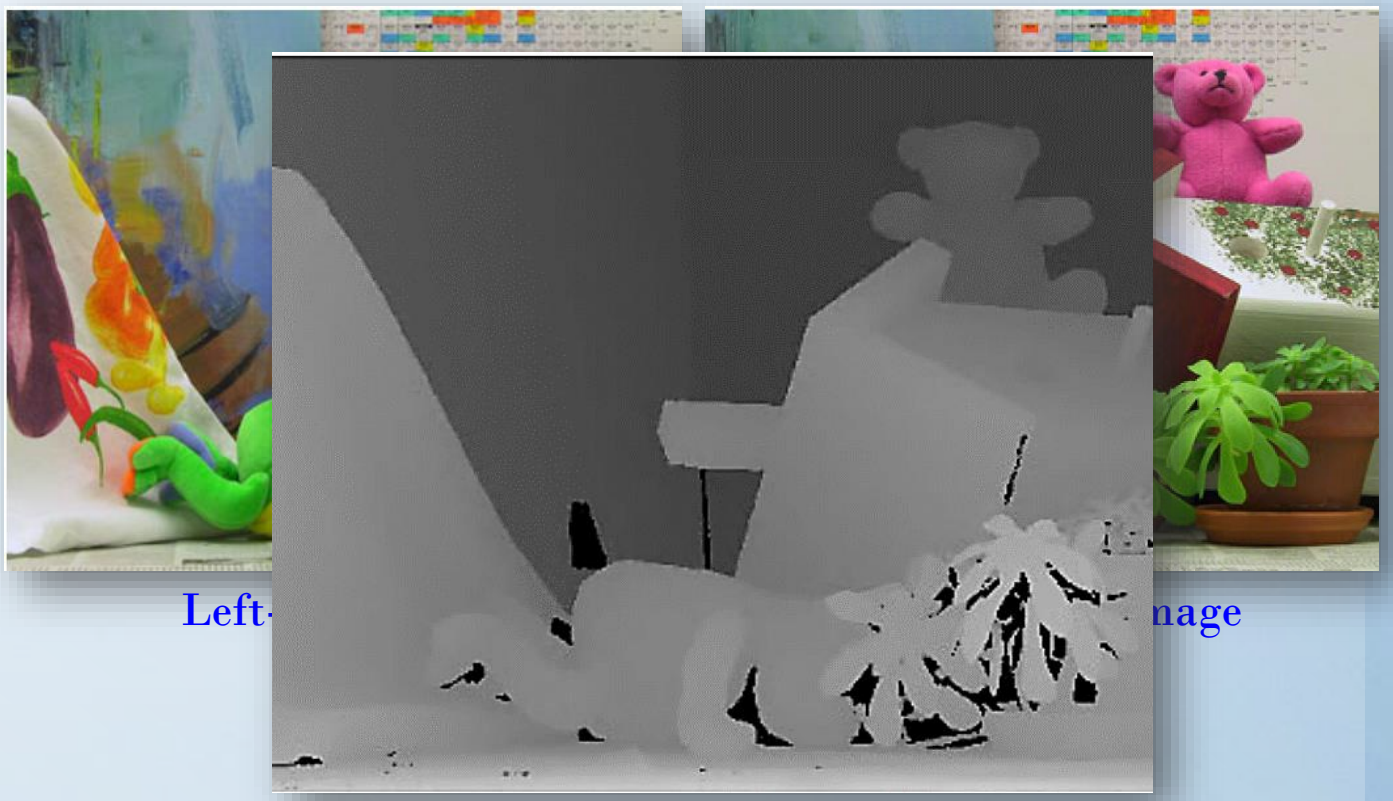


Multi-camera Views for Depth Estimation

Multi-views Stereo and Depth map

- ✓ Cameras Maps 3D objects to 2D (Loss of dimensionality): No Depth information.
- ✓ Stereo matching is the process of taking two or more images and estimating a 3D model of the scene by finding matching pixels in the images and converting their 2D positions into 3D depths.

- ✓ *it was known that we perceive depth based on the differences in appearance between the left and right eye.*
- ✓ *As a simple experiment, hold your finger vertically in front of your eyes and close each eye alternately. You will notice that the finger jumps left and right relative to the background of the scene.*



Left-

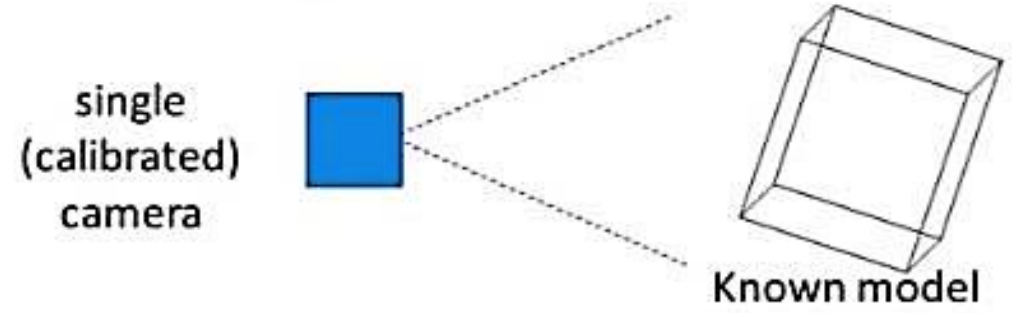
nage

Disparity Map

Multi-views Stereo and Depth map

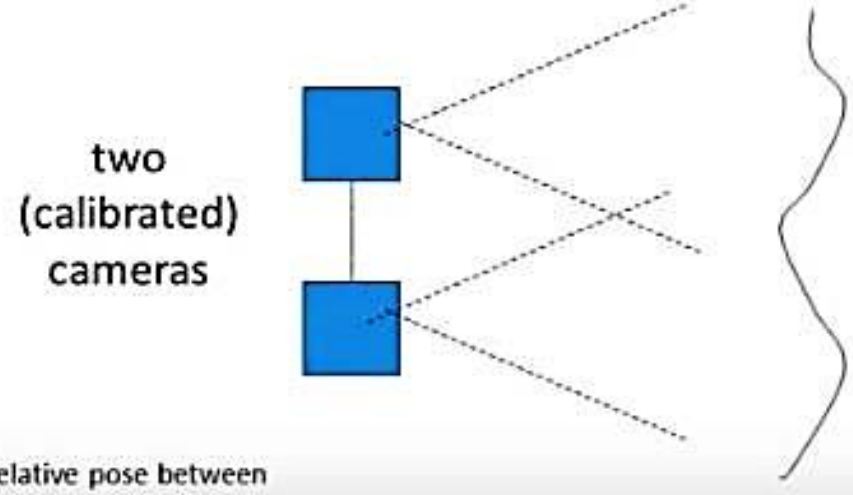
Inferring 3D from 2D

- Model based pose estimation



-> Can determine the pose of the model

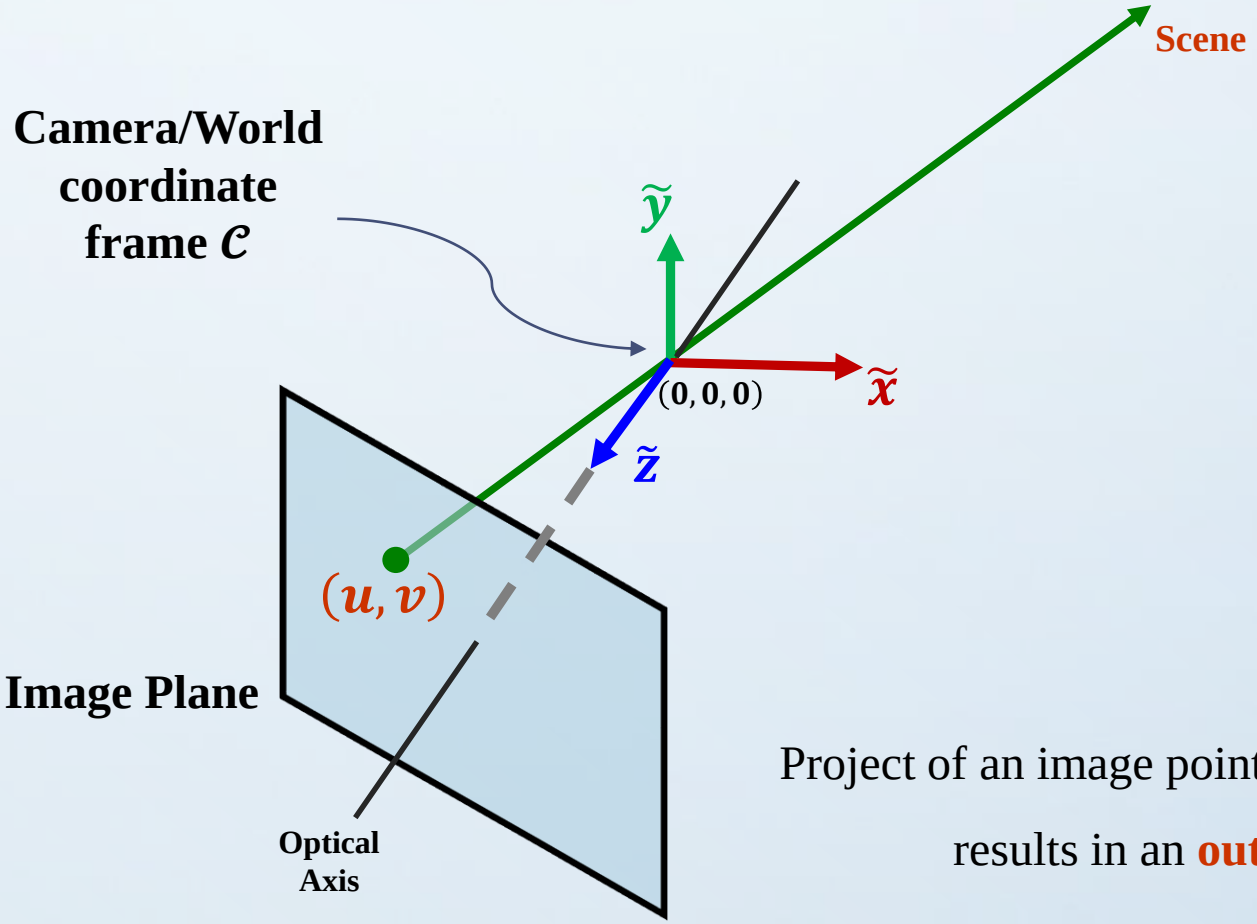
- Stereo vision



-> Can determine the positions of points in the scene

Backward Projection from 2D to 3D

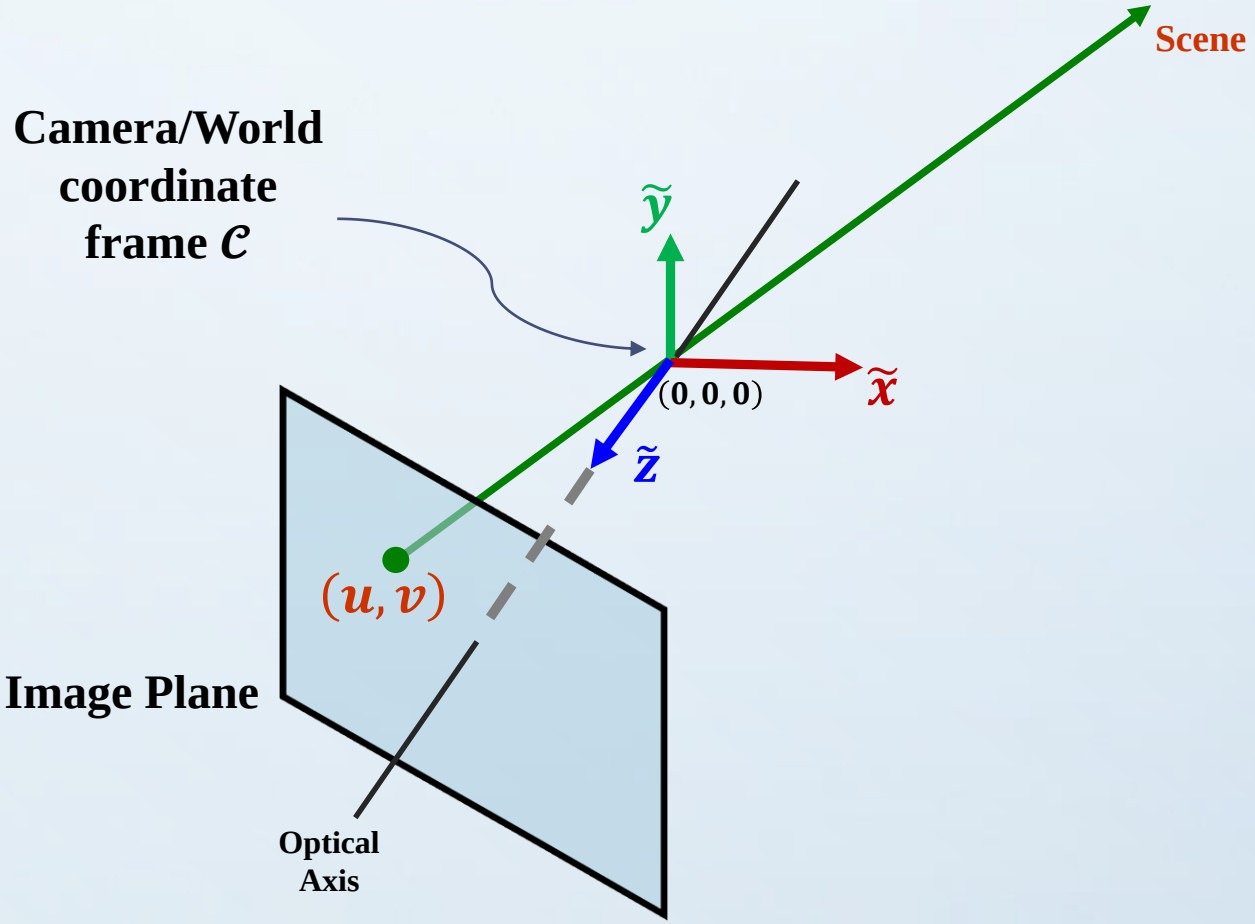
Given a calibrated camera, can we find out calibrated 3D scene from a single 2D image.



Project of an image point back into the scene results in an **outgoing ray**.

Backward Projection from 2D to 3D

Given a calibrated camera, can we find out calibrated 3D scene from a single 2D image.



3D to 2D (Point)

$$u = f_x \frac{x_c}{z_c} + o_x$$

$$v = f_y \frac{y_c}{z_c} + o_y$$

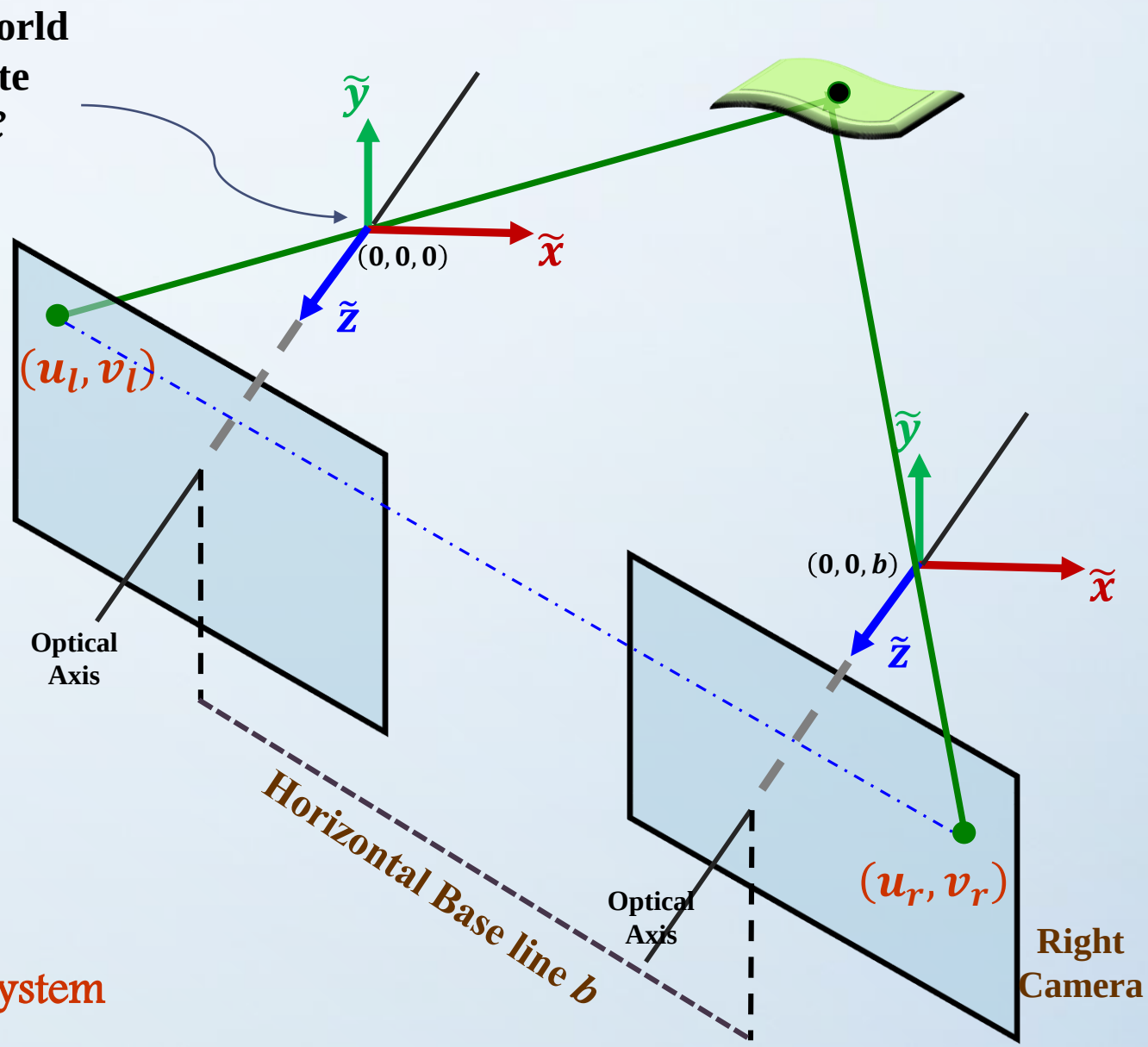
2D to 3D (Ray)

$$x = \frac{z}{f_x}(u - o_x)$$

$$y = \frac{z}{f_y}(v - o_y)$$

$$z > 0$$

Triangulation using Two Cameras



$$u_l = f_x \frac{x}{z} + o_x$$

$$v_l = f_y \frac{y}{z} + o_y$$

$$u_r = f_x \frac{x - b}{z} + o_x$$

$$v_r = f_y \frac{y}{z} + o_y$$

f_x, f_y, b, o_x, o_y are unknown

Stereo System

Binocular Vision

Simple Stereo Depth and Disparity

✓ Given a calibrated camera, can we find out calibrated 3D scene from a single 2D image.

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x,\ f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x,\ f_y \frac{y}{z} + o_y\right)$$

✓ Solving for x, y and z:

$$x = b\frac{u_l - o_x}{u_l - u_r} \qquad y = b\frac{f_x(v_l - o_y)}{f_x(u_l - u_r)} \qquad \boxed{z = b\frac{f_x}{u_l - u_r}}$$

Where, $(u_l - u_r)$ is called **Disparity**

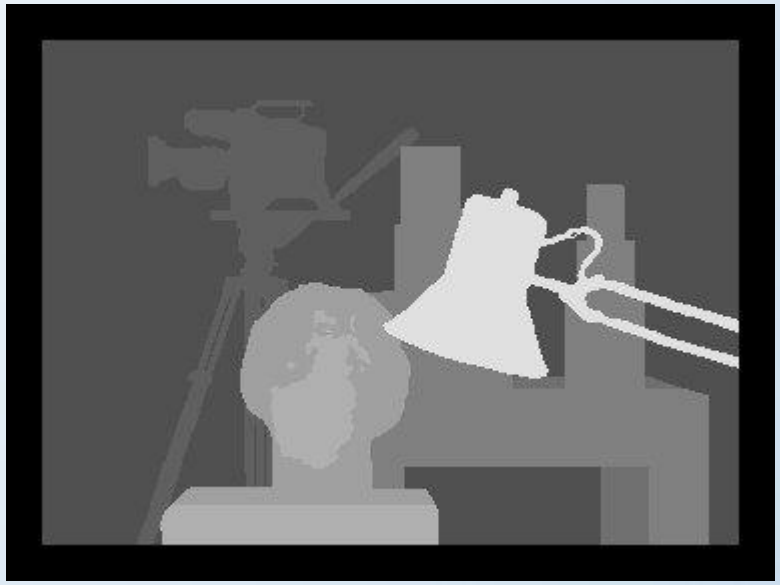
Depth *z* is inversely proportional to Disparity.

Disparity is proportional to Baseline.

Stereo Matching: Finding Disparity

✓ Goal: To find the disparity between left and right stereo pairs.

Left/Right Camera Images



Ground Truth Disparity Map

- From perspective geometry no disparity along y-direction:

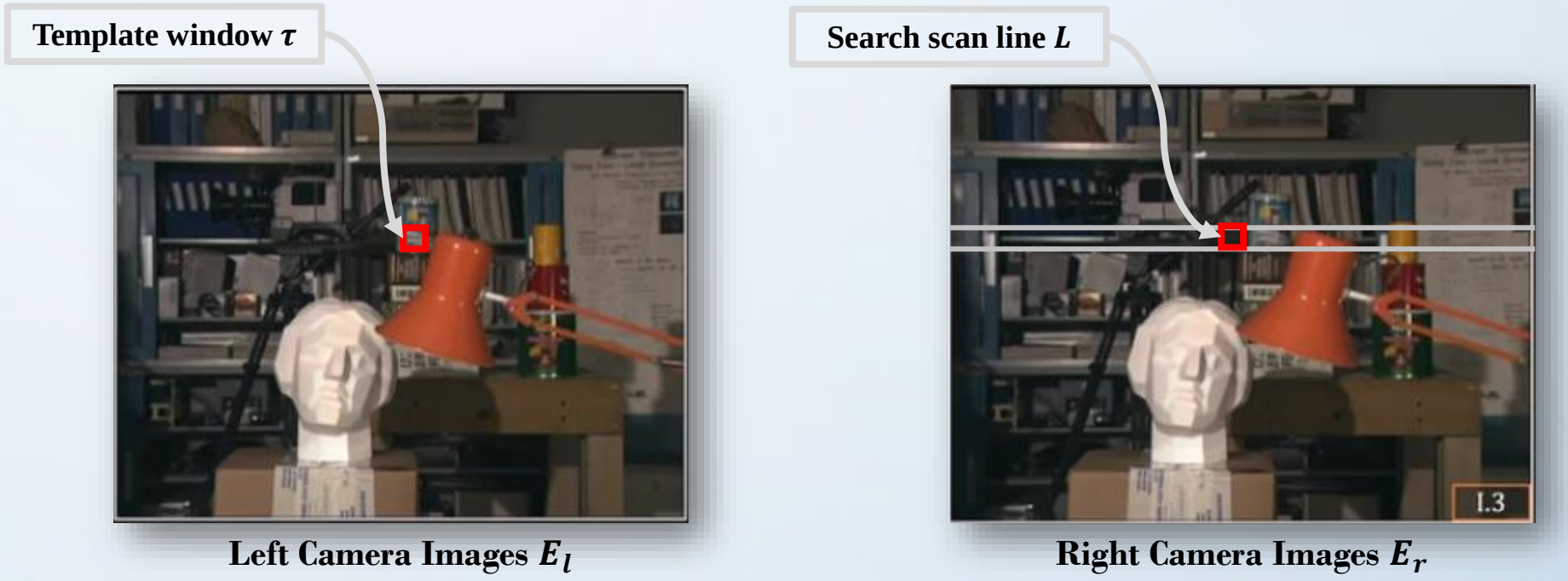
$$v_l, v_r = f_y \frac{y}{z} + o_y$$

Corresponding scene point must lie on the same horizontal scan line

Pic Courtesy: *2001 Stereo datasets with ground truth: https://vision.middlebury.edu/stereo/data/*

Window based Methods

Determine disparity using template matching.



Left Camera Images E_l Right Camera Images E_r

Disparity:

$$d = u_l - u_r$$

Depth:

$$z = b\frac{f_x}{u_l - u_r}$$

Template Matching: Similarity Measures

$$SAD(k,l) = \sum_{(i,j)\in T} |E_l(i,j) - E_r(i+k, j+l)|$$

What is the Ideal Window Size?

Window size 5 pixels
(Sensitive to noise)

Window size 30 pixels
(Poor Localization)

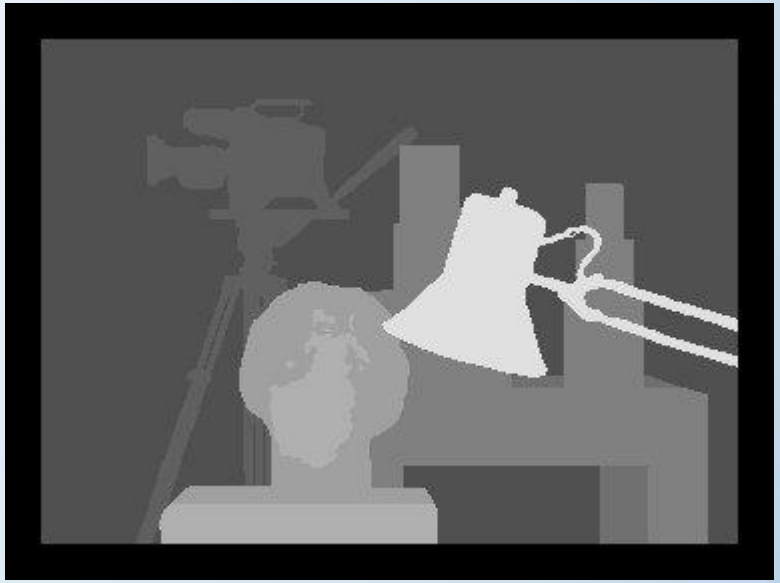
- *Adaptive Window Size*:

For each point, match using windows of multiple sizes and use the disparity that is a result of the best similarity measure (minimizing SAD per pixel)

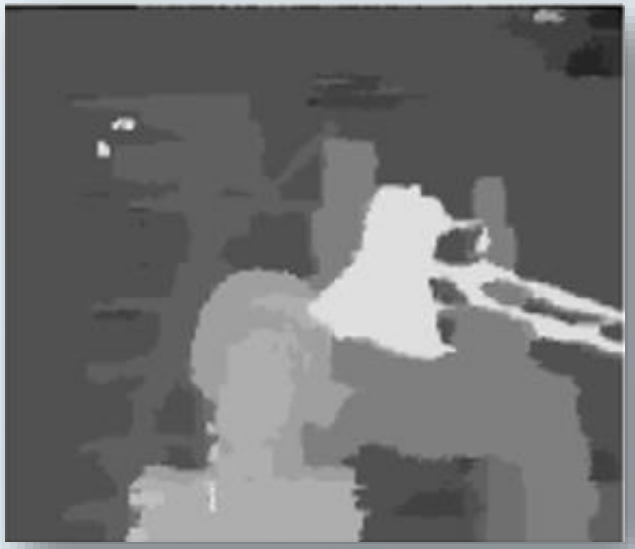
Stereo Matching Comparison

Left Camera Images E_l

Right Camera Images E_r



Ground Truth Disparity Map



SSD – Window Size 21

SSD – Adaptive Window

State of the art result

Problems with uncalbrated Stereo

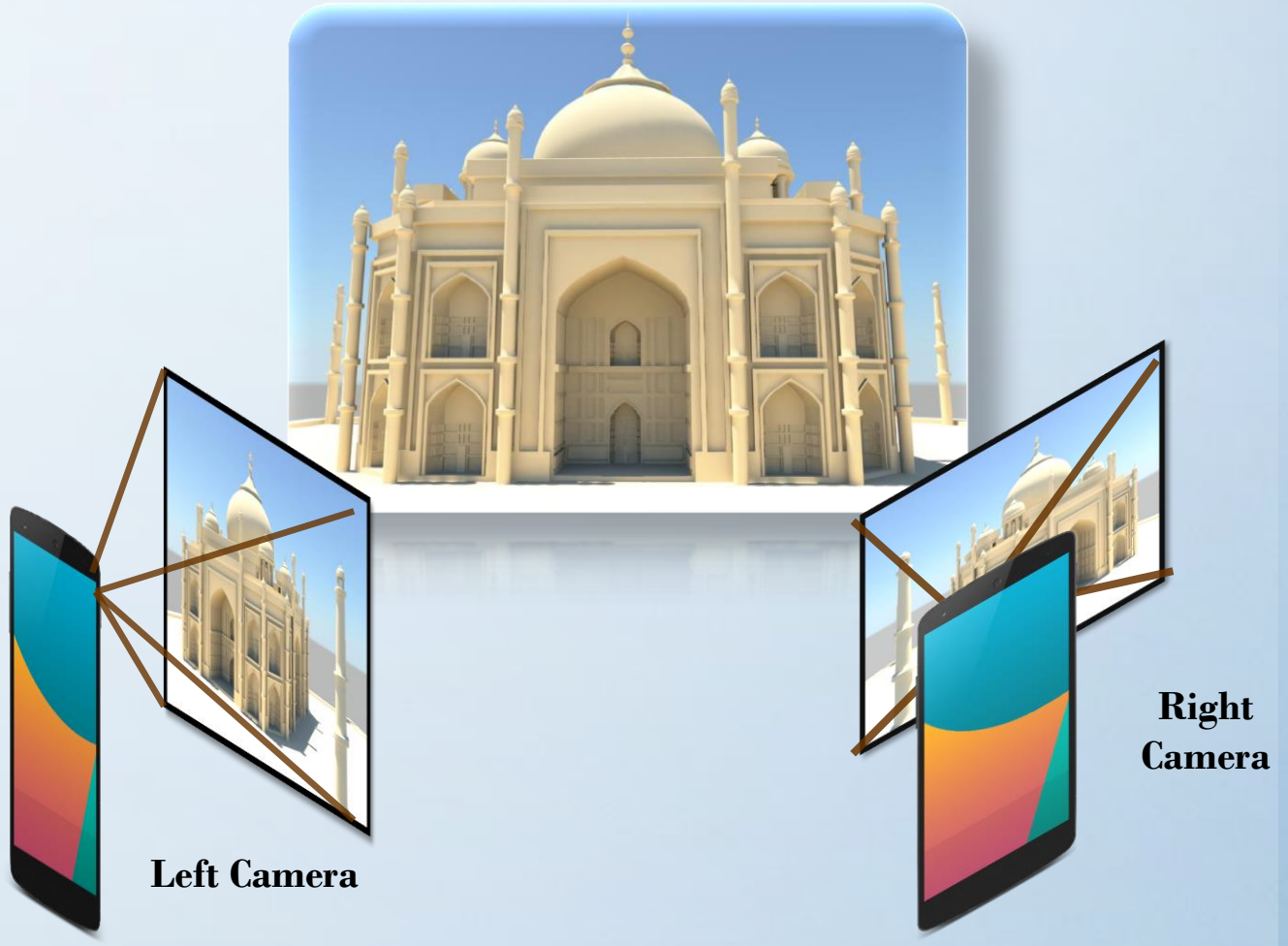
Computes 3D structure of a static scene from two arbitrary view points

- Known Intrinsic Parameters:

  (f_x, f_y, o_x, o_y) are known for both the views.

- Unknown Extrinsic Parameters:

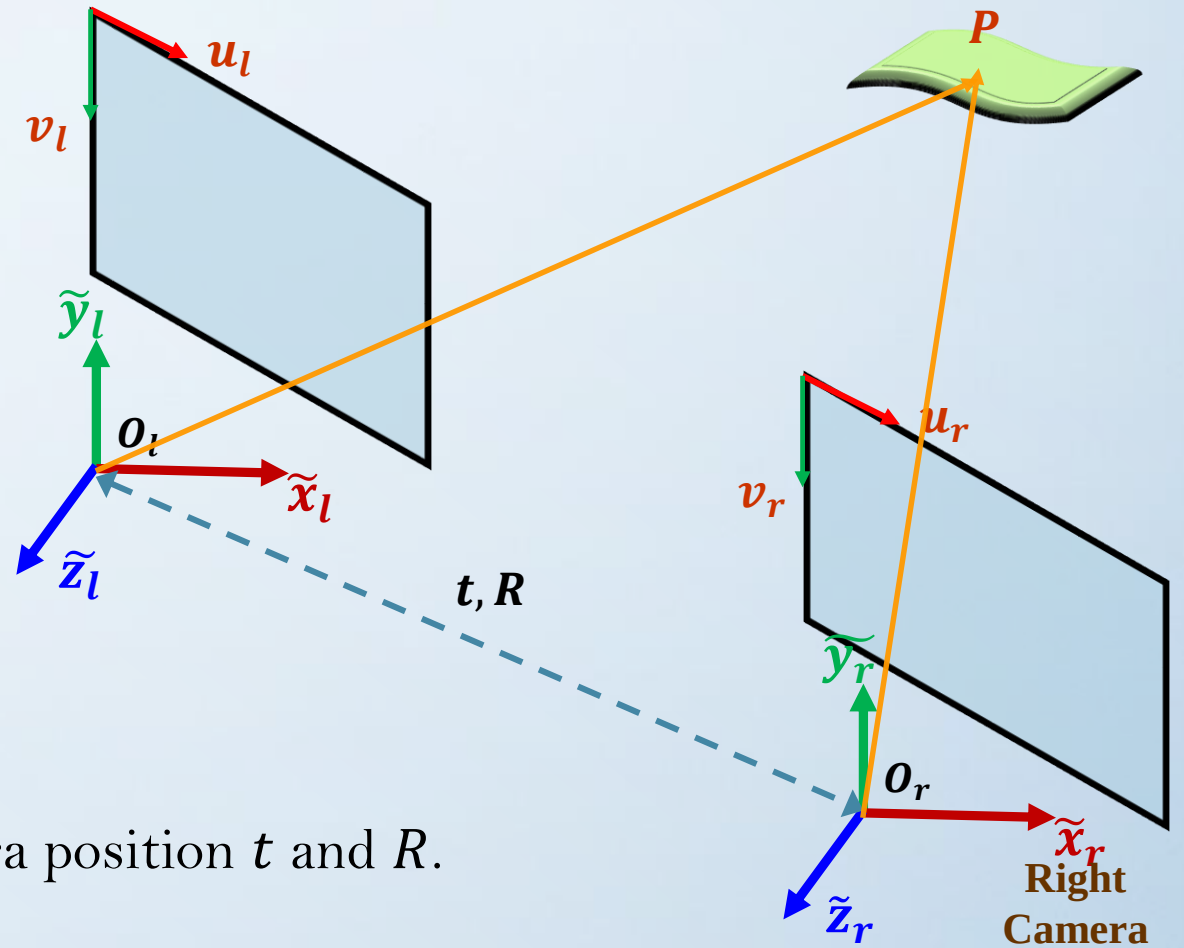
  Relative position and orientation of camera.



Problems with uncalbrated Stereo

1. Assuming the camera intrinsic parameter K is known for each camera.

2. Find a few reliable corresponding points in the images captured from left and right camera. This comes under finding interest points matching (which can be efficiently obtained using SIFT). Moreover, we exactly need 8 corresponding points in both the images.



3. Find the relative camera position t and R.

4. Find dense correspondence. [For each point in left image and finding its corresponding point in the right image]

5. Computer the depth using triangulation.

Epipolar Geometry *Calibrating the un-calibrated stereo*

- ✓ **Goal**: To find out the relative position and orientation of one camera with respect to the other camera.

- ✓ Given a point in the plane of the first image – Find the corresponding point in second image.

- ✓ Epipolar geometry is used to describe geometric relationship in image pairs.

- ✓ Enables efficient search for and prediction of corresponding point.

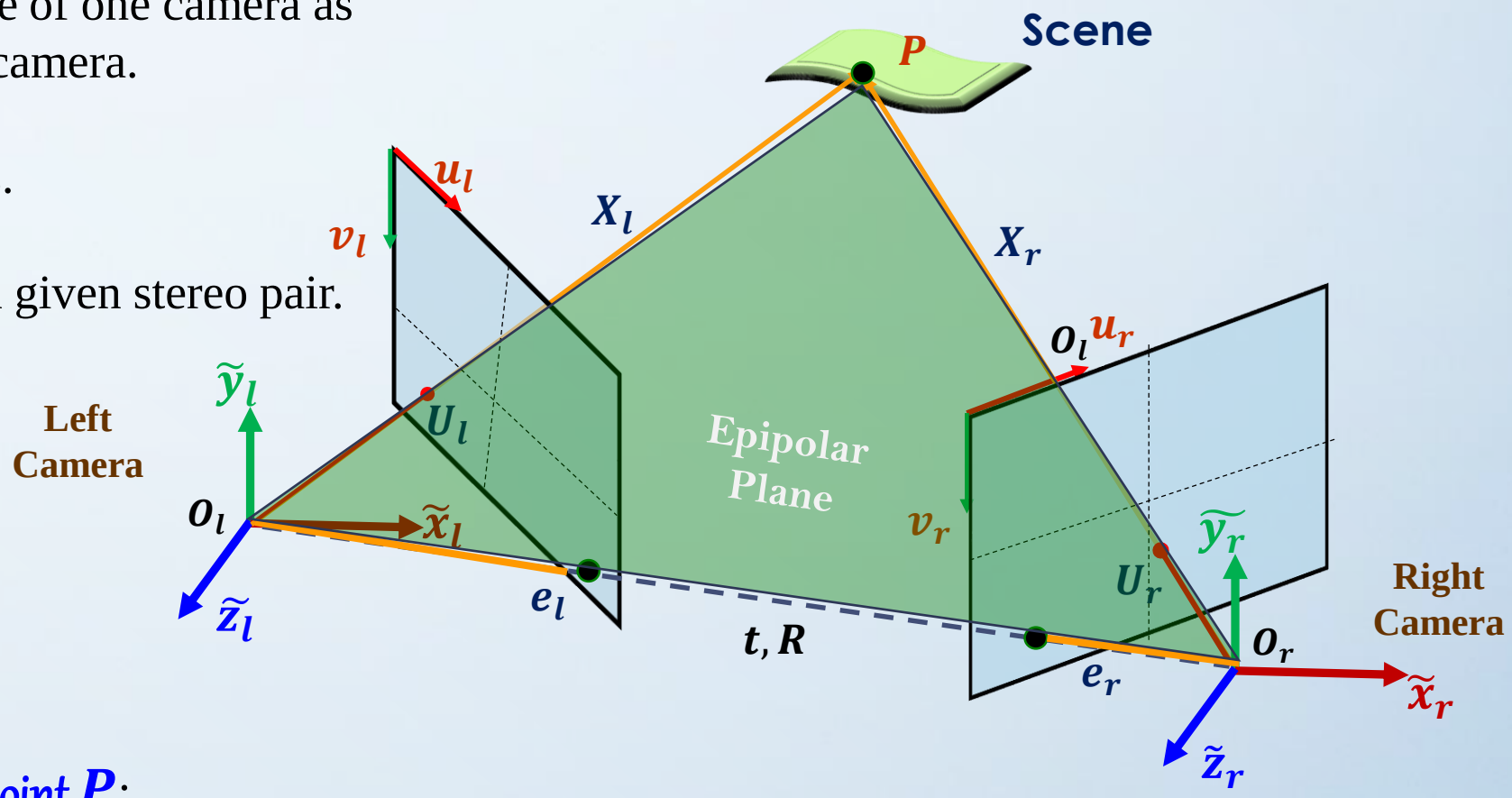
Epipolar Geometry: Epipoles

- **Epipole**:

Image point of origin/pinhole of one camera as viewed by other camera.

✓ e_l and e_r are the epipoles.

✓ e_l and e_r are unique for a given stereo pair.



- **Epipolar Plane of scene point *P***:

✓ The plane formed by the camera origins (O_l, O_r), epipoles (e_l, e_r) and the scene point P.

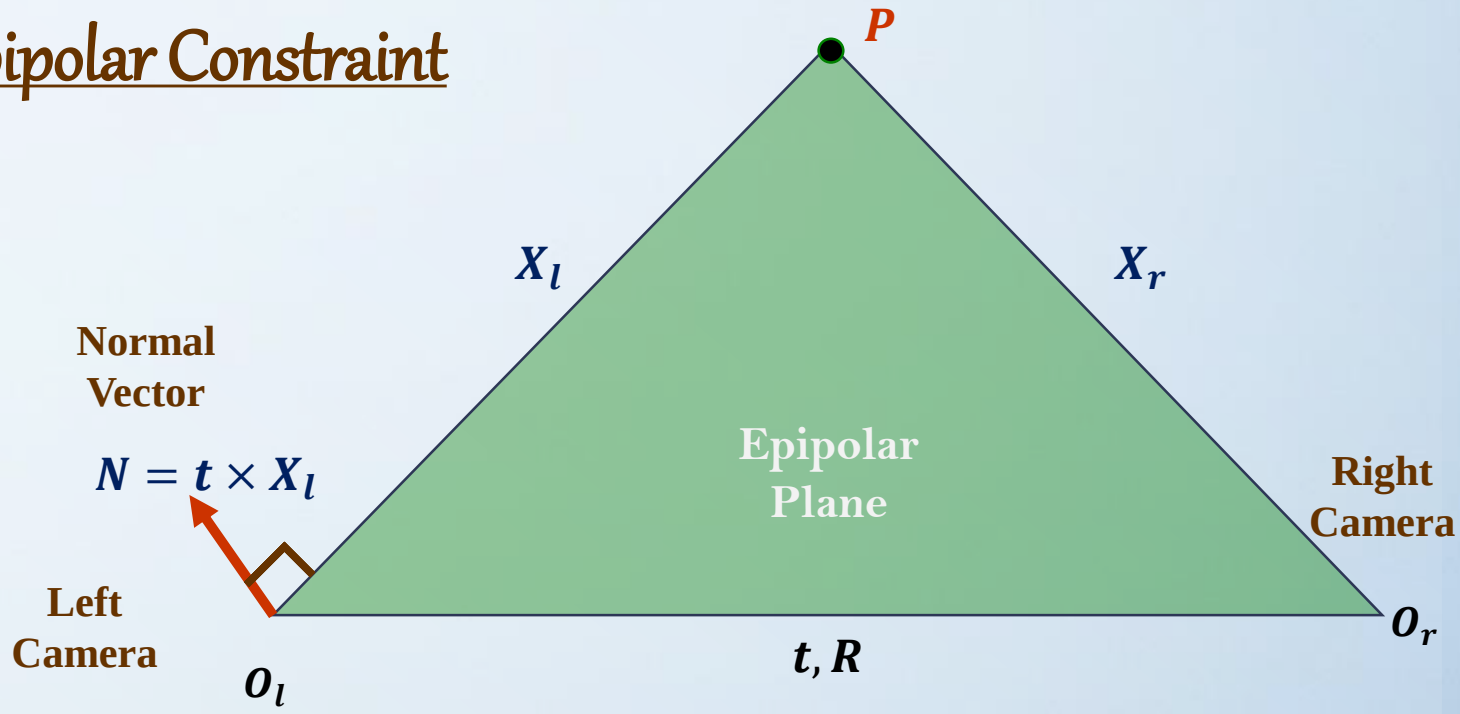
✓ Every scene point lie on a unique epipolar plane.

Epipolar Constraint

- *Vector Normal to the Epipolar Plane* :
  - Cross product between t and X_l.

$$N = t \times X_l$$

  - Note N and X_l are perpendicular to each other. The dot product between them is 0.



The epipolar constraint can be formulated as:

$$X_l \cdot (t \times X_l) = 0$$

- *Epipolar Constraint*:

$$[x_l \quad y_l \quad z_l] \begin{bmatrix} t_y z_l - t_z y_l \\ t_z x_l - t_x z_l \\ t_x y_l - t_y x_l \end{bmatrix} = 0$$

- *Epipolar Constraint* : In matrix-vector form:

$$[x_l \quad y_l \quad z_l] \underbrace{\begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}}_{T_\times} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = 0$$

Epipolar Constraint

- **Epipolar Constraint** : In matrix-vector form:

$$\begin{bmatrix} x_l & y_l & z_l \end{bmatrix} \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = 0$$

- $t_{3\times1}$: Position of right camera in left camera's coordinate frame.
- $R_{3\times3}$: Orientation of left camera in right camera's coordinate frame.
- Using these we can now relate the 3D coordinate of a point P in the left camera to the 3D coordinate of the same point to the right camera.

$$X_l = RX_r + t$$

$$\begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Epipolar Constraint

- Substituting in the epipolar constraint:

$$[x_l \quad y_l \quad z_l]\left(\begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} + \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}\right) = 0$$

$$\boldsymbol{t \times t = 0}$$

$$[x_l \quad y_l \quad z_l]\begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = 0$$

Essential Matrix $\boldsymbol{E}$

- Epipolar Matrix:

$$\boldsymbol{E = T_{\times} R}$$

Matrix $\boldsymbol{E}$ relates the 3D coordinate point of left camera X_l with the corresponding right camera X_r.

Decomposition of Essential Matrix E

- Essential Matrix:

$$\boldsymbol{E} = \boldsymbol{T}_{\times}\boldsymbol{R}$$

$$\begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

- Decomposition:

Given the $\boldsymbol{T}_{\times}$ a **skew symmetry** matrix ($\boldsymbol{a_{ij}} = -\boldsymbol{a_{ji}}$) and $\boldsymbol{R}$ is orthogonal matrix, It is possible to decouple $\boldsymbol{T}_{\times}$ and $\boldsymbol{R}$ from their product using **Singular Value Decomposition**.

- **If we know the essential matrix $\boldsymbol{E}$ then we can compute $\boldsymbol{T}_{\times}$ and $\boldsymbol{R}$**

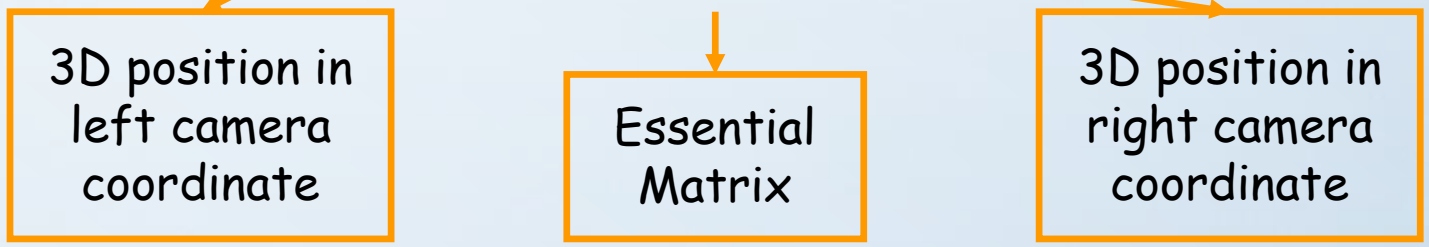
Computation of Essential Matrix E

- **Essential Matrix**:

Matrix $\boldsymbol{E}$ relates the 3D coordinate point of left camera X_l with the corresponding right camera X_r.

$$\boldsymbol{X_l^T E X_r = 0}$$

$$[x_l \quad y_l \quad z_l] \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = 0$$



- **Note**:
  - Both $\boldsymbol{X_l}$ and $\boldsymbol{X_r}$ are 3D coordinate points of left and right cameras respectively and both are unknown.
  - But we do know the corresponding 2D (projection) points in the image coordinate.

Perspective Projection for Left and Right Cameras

- Relating 3D coordinate to 2D coordinate:

$$u_l = f_x^{(l)} \frac{x_l}{z_l} + o_x^{(l)} \quad \text{and} \quad v_l = f_y^{(l)} \frac{y_l}{z_l} + o_y^{(l)}$$

$$u_l z_l = f_x^{(l)} x_l + z_l o_x^{(l)} \quad \text{and} \quad v_l z_l = f_y^{(l)} y_l + z_l o_y^{(l)}$$

- Representing in Matrix-Vector form:

$$z_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = \begin{bmatrix} z_l u_l \\ z_l v_l \\ z_l \end{bmatrix} = \begin{bmatrix} f_x^{(l)} x_l + z_l o_x^{(l)} \\ f_y^{(l)} y_l + z_l o_y^{(l)} \\ z_l \end{bmatrix} = \begin{bmatrix} f_x^{(l)} & o_x^{(l)} & 0 \\ 0 & f_y^{(l)} & o_y^{(l)} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix}$$

Known camera matrix K_l

- For both left and right camera:

$$z_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = \begin{bmatrix} f_x^{(l)} & o_x^{(l)} & 0 \\ 0 & f_y^{(l)} & o_y^{(l)} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix}$$

Known camera matrix K_l (left)

$$z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = \begin{bmatrix} f_x^{(r)} & o_x^{(r)} & 0 \\ 0 & f_y^{(r)} & o_y^{(r)} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix}$$

Known camera matrix K_r (right)

Perspective Projection for Left and Right Cameras

- Note: Both left and right camera matrix are indicating same 3D point in the scene.

$$[x_l \quad y_l \quad z_l] = X_l^T = [u_l \quad v_l \quad 1] z_l (K_l^{-1})^T$$

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = X_r = K_r^{-1} z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix}$$

- Rewriting epipolar constraint equation by substituting above equation:

$$[x_l \quad y_l \quad z_l] \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = 0$$

$$[u_l \quad v_l \quad 1] z_l \left(K_l^{-1}\right)^T \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} K_r^{-1} z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0$$

✓ z_l and z_r represent the depth of the scene, it is measured from the center of the pinhole. The depth can only be zero if the image lie on the pinhole: Thus,

$$\boldsymbol{z_l = z_r \neq 0}$$

Fundamental Matrix for Epipolar Geometry

- **Note**: Rewrinting by eliminating z_l and z_r, as:

$$[u_l \quad v_l \quad 1]\left(K_l^{-1}\right)^T \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} K_r^{-1} \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0$$

- Representing in Matrix-Vector form:

$$[u_l \quad v_l \quad 1] \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0$$

Fundamental Matrix ***F***

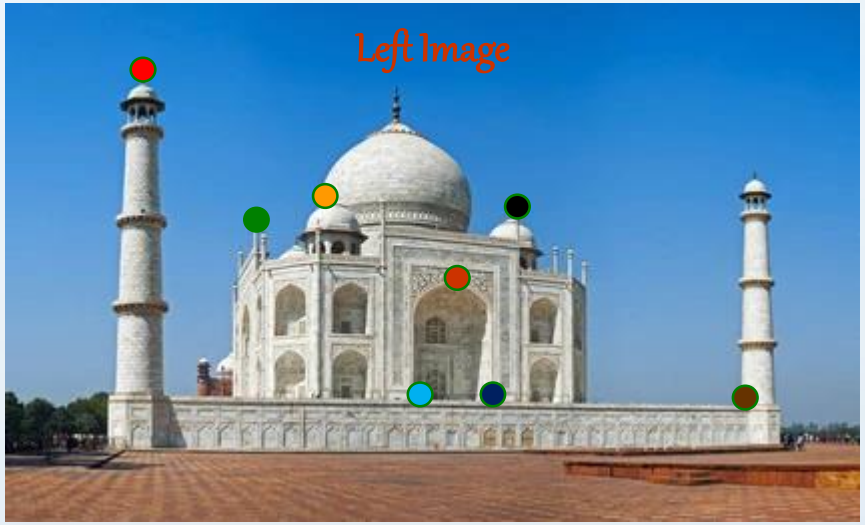
$$\boldsymbol{E = K_l^T F K_r}$$

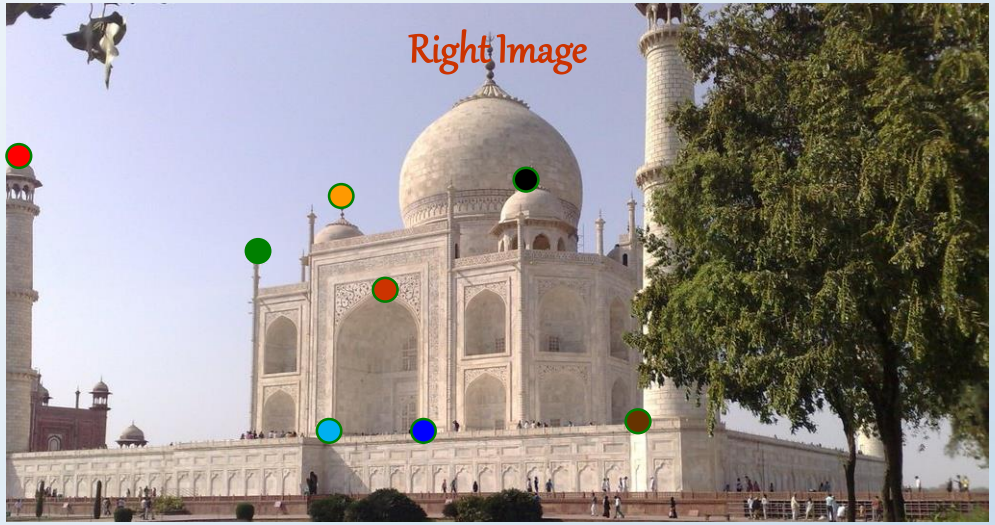
$$\boldsymbol{E = T_\times R}$$

Estimating Fundamental Matrix

Finding set of corresponding features in the left and right image (e.g. using SIFT or manually)

Note: For uncalibrated stereo these images are captured with different cameras





- (u_l^1, v_l^1)
- (u_l^2, v_l^2)
- (u_l^m, v_l^m)

- (u_r^1, v_r^1)
- (u_r^2, v_r^2)
- (u_r^m, v_r^m)

Stereo Calibration Procedure: *Calibrating the un-calibrated stereo*

Step A: For each corresponding point i the epipolar constraint is:

$$\underbrace{\begin{bmatrix} u_l^{(i)} & v_l^{(i)} & 1 \end{bmatrix}}_{\text{Known}} \underbrace{\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}}_{\text{Unknown}} \underbrace{\begin{bmatrix} u_r^{(i)} \\ v_r^{(i)} \\ 1 \end{bmatrix}}_{\text{Known}} = 0$$

Expanding the equation:

$$\left(f_{11}u_r^{(i)} + f_{12}v_r^{(i)} + f_{13}\right)u_l^{(i)} + \left(f_{21}u_r^{(i)} + f_{22}v_r^{(i)} + f_{23}\right)v_l^{(i)} + f_{31}u_r^{(i)} + f_{32}v_r^{(i)} + f_{33} = 0$$

We will get $\boldsymbol{m}$ such equations for $\boldsymbol{m}$ different points in left and right image.

Stereo Calibration Procedure: *Calibrating the un-calibrated stereo*

Step B: Rearranging terms to form a linear system:

$$\begin{bmatrix} u_l^{(1)}u_r^{(1)} & u_l^{(1)}v_r^{(1)} & u_l^{(1)} & v_l^{(1)}u_r^{(1)} & v_l^{(1)}v_r^{(1)} & v_l^{(1)} & u_r^{(1)} & v_r^{(1)} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_l^{(i)}u_r^{(i)} & u_l^{(i)}v_r^{(i)} & u_l^{(i)} & v_l^{(i)}u_r^{(i)} & v_l^{(i)}v_r^{(i)} & v_l^{(i)} & u_l^{(i)} & u_r^{(i)} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_l^{(m)}u_r^{(m)} & u_l^{(m)}v_r^{(m)} & u_l^{(m)} & v_l^{(m)}u_r^{(m)} & v_l^{(m)}v_r^{(m)} & v_l^{(m)} & u_l^{(m)} & u_r^{(m)} & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$\boldsymbol{A}$ — **Known**

$\boldsymbol{f}$ — **Unknown**

$$\boldsymbol{AP = 0}$$

The Tale of Scale

Step B: As discussed in single camera calibration; alike projection matrix $\boldsymbol{P}$ the fundamental matrix $\boldsymbol{f}$ is also acts on homogenous coordinate and invariant to scale as:

$$[u_l \quad v_l \quad 1]\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}\begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0 = [u_l \quad v_l \quad 1]\begin{bmatrix} \boldsymbol{k}f_{11} & \boldsymbol{k}f_{12} & \boldsymbol{k}f_{13} \\ \boldsymbol{k}f_{21} & \boldsymbol{k}f_{22} & \boldsymbol{k}f_{23} \\ \boldsymbol{k}f_{31} & \boldsymbol{k}f_{32} & \boldsymbol{k}f_{33} \end{bmatrix}\begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix}$$

The fundamental matrix $\boldsymbol{f}$ and $\boldsymbol{kf}$ defines the same epipolar geometry. That is F is defined up to a scale. Thus it preserves the magnitude:

Set the fundamental matrix to some arbitrary scale:

$$\|f\|^2 = 1$$

Solve for f

Step C: Find the least square solution for fundamental matrix $\boldsymbol{f}$:

- ✓ Set the scale so that: $\|\boldsymbol{f}\|^2 = 1$
- ✓ We want $\boldsymbol{Af}$ as close as to zero but $\|\boldsymbol{f}\|^2 = 1$.

$$\min_{f} \|\boldsymbol{Af}\|^2 \quad \text{such that } \|\boldsymbol{f}\|^2 = 1$$

This is a constrained least square problem

$$\min_{P} (f^T A^T A f) \text{ such that } f^T f = 1$$

Define a loss function as $\mathcal{L}(f, \lambda)$:

$$\mathcal{L}(f, \lambda) = f^T A^T A f - \lambda(f^T f - 1)$$

Finally the solution for the $\boldsymbol{f}$ is the Eigen vector corresponds to the smallest Eigenvalue λ of the matrix $\boldsymbol{A^T A}$.

Rearranging the vector $\boldsymbol{f}$ obtained from the previous solution in matrix form to obtain F.

Extracting Rotation and Translation Matrix

Step D: Compute the essential matrix $\boldsymbol{E}$ from known right and left intrinsic camera matrix and fundamental matrix $\boldsymbol{f}$:

$$\boldsymbol{E = K_l^T F K_r}$$

Step E: Extract $\boldsymbol{R}$ and $\boldsymbol{t}$ from $\boldsymbol{E}$:

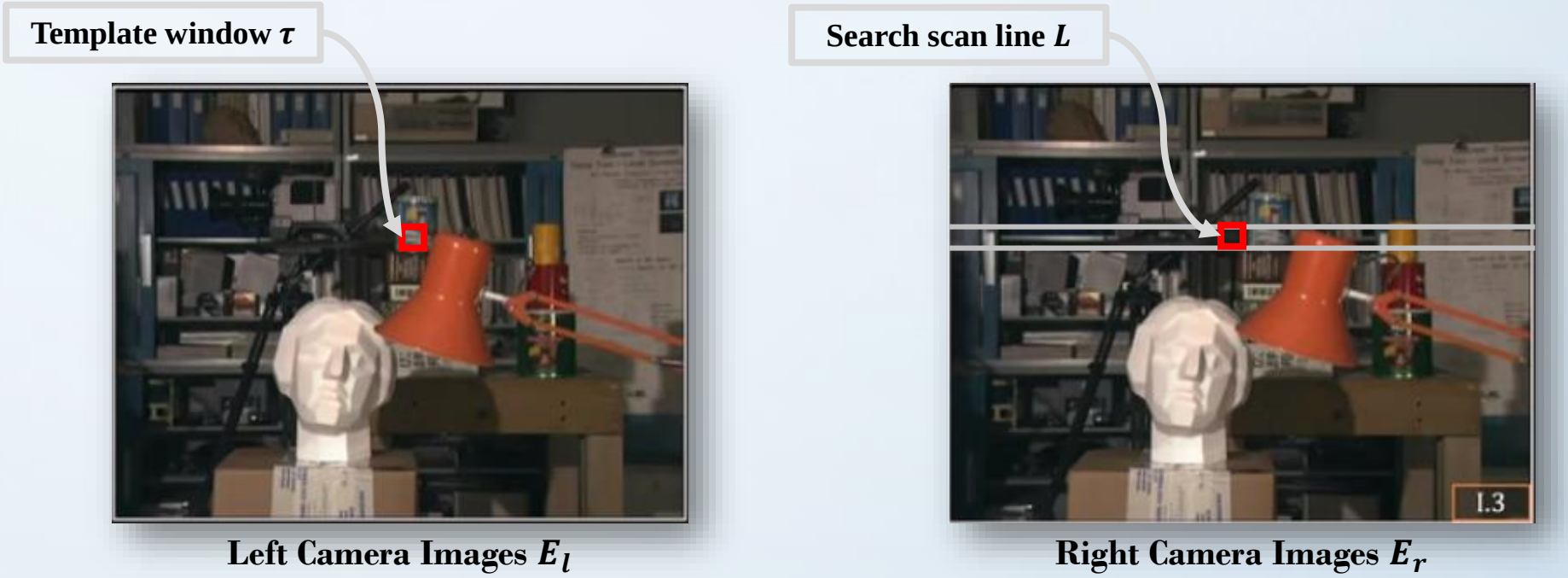
$$\boldsymbol{E = T_\times R}$$

(Using Singular value decomposition (SVD))

Finally the solution for the $\boldsymbol{f}$ is the Eigen vector corresponds to the smallest Eigenvalue λ of the matrix $\boldsymbol{A^T A}$.

Rearranging the vector $\boldsymbol{f}$ obtained from the previous solution in matrix form to obtain F.

Uncalibrated Stereo and Correspondence



Left Camera Images E_l

Right Camera Images E_r

- ✓ For calibrated stereo as we know the base line difference, simply the scan along the row or horizontal line (simple 1D search) would predict the depth and find the correspondence of left image compared to the right.

- ✓ Stereo matching reduces to 1D search.

Lets Observe for uncalibrated stereo with known fundamental matrix.

Uncalibrated Stereo and Correspondence: Finding Epipolar Line

- ✓ **Given**: Fundamental matrix f and the point on the left image (u_l, u_r):
- ✓ **To Find**: Equation of Epipolar line in right image.

$$\begin{bmatrix} u_l & v_l & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{u_r} \\ \boldsymbol{v_r} \\ \mathbf{1} \end{bmatrix} = 0$$

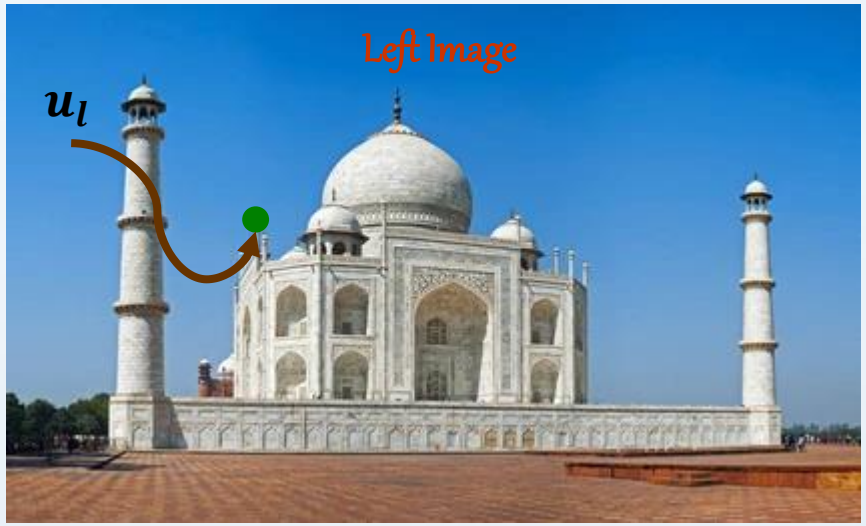
Expanding the equation:

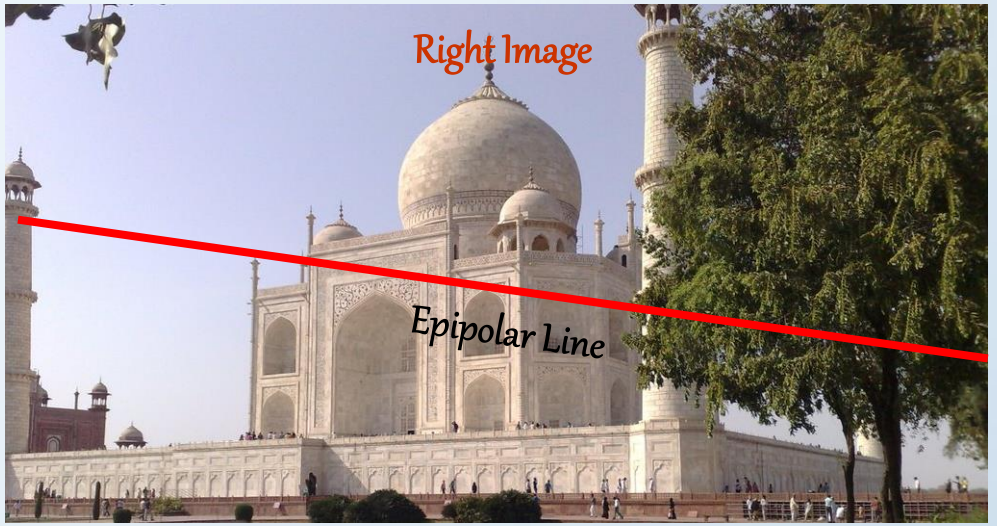
$$(f_{11}u_l + f_{21}v_l + f_{31})\boldsymbol{u_r} + (f_{12}u_l + f_{22}v_l + f_{32})\boldsymbol{v_l} + (f_{13}u_l + f_{23}v_l + f_{33}) = 0$$

Equation for right epipolar line:

$$a_l\boldsymbol{u_r} + b_l\boldsymbol{v_l} + c_l = 0$$

Uncalibrated Stereo and Correspondence: Finding Epipolar Line: Example





Given fundamental matrix:

$$F = \begin{bmatrix} -0.003 & -0.028 & 13.19 \\ -0.003 & -0.008 & -29.2 \\ 2.97 & 56.38 & -9999 \end{bmatrix}$$

The left image point:

$$u_l = \begin{bmatrix} 343 \\ 221 \\ 1 \end{bmatrix}$$

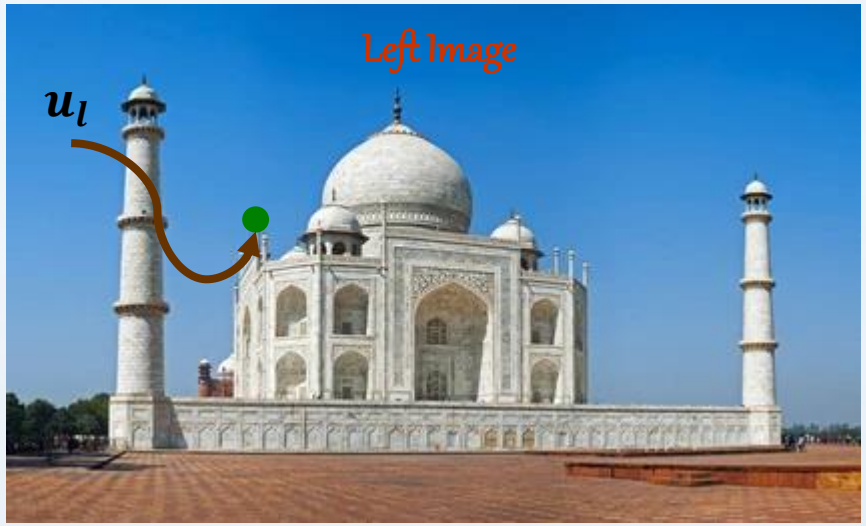
The equation for the epipolar line in the right image is

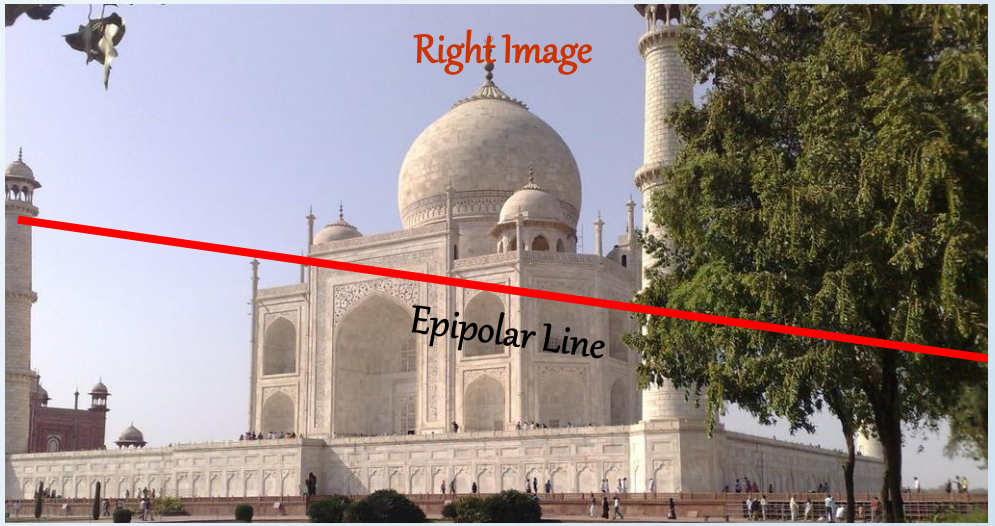
$$\begin{bmatrix} u_r & v_r & 1 \end{bmatrix} \begin{bmatrix} -.003 & -.003 & 2.97 \\ -.028 & -.008 & 56.38 \\ 13.19 & -29.2 & -9999 \end{bmatrix} \begin{bmatrix} 343 \\ 221 \\ 1 \end{bmatrix} = 0$$

Equation for the epipolar line in the right image:

$$0.03u_r + 0.99\, v_r - 265 = 0$$

Uncalibrated Stereo and Correspondence: Finding Epipolar Line: Example





Corresponding scene points lines on the epipolar line.

Finding correspondance is still a 1D search.