Lecture 14

Camera Extrinsics & Intrinsics

Mon. Oct. 26, 2020

COMP 558 Overview

Part 1 : 2D Vision

RGB

Image filtering

Edge detection

Least Squares Estimation

Robust Estimation: Hough transform & RANSAC

Features 1: corners

Image Registration: the Lucas-Kanade method

Scale space

Histogram-based Tracking

Features 2: SIFT

Part 2 : 3D Vision

Perspective: projection, translation, vanishing points

Rotation, Homogeneous coordinates

Camera intrinsics and extrinsics

Least Squares methods (SVD)

Camera Calibration

Homographies

Stereo and Epipolar Geometry

Stereo correspondence

Features 3: CNN's

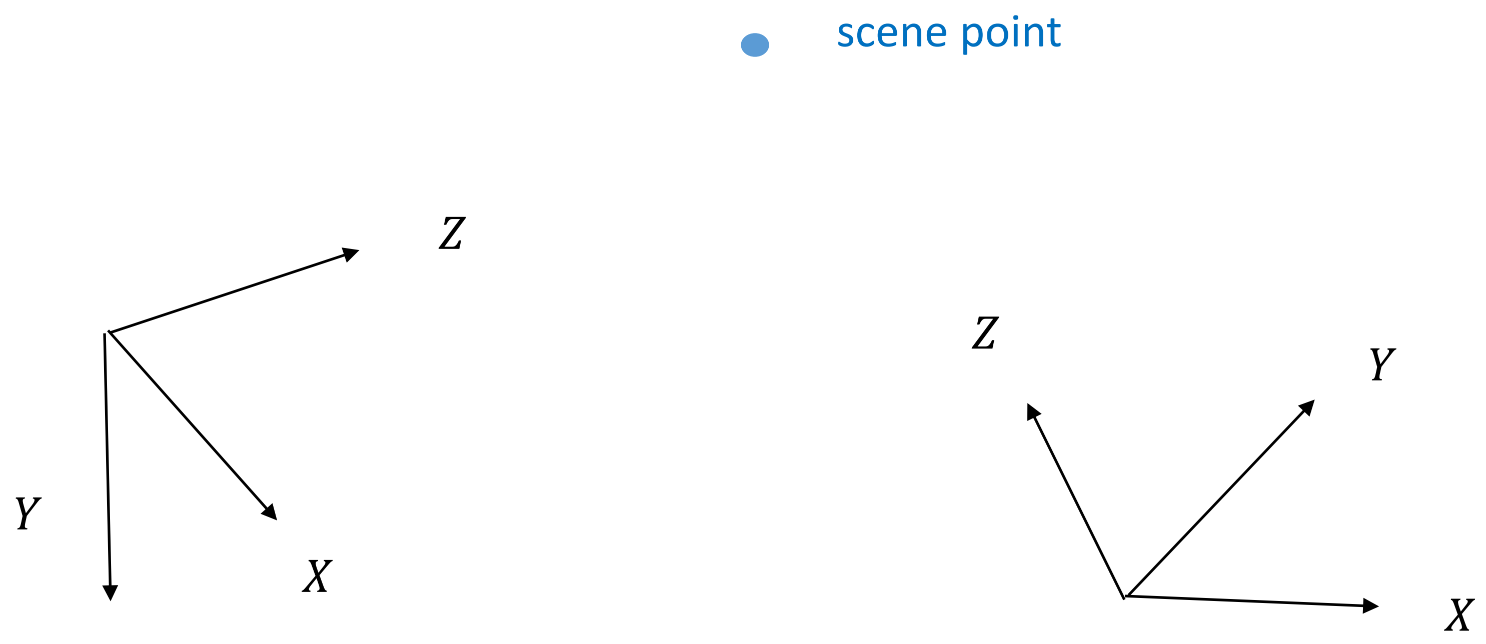
Object classification and detection

Segmentation (time permitting)

Cameras and Photography

RGBD Cameras

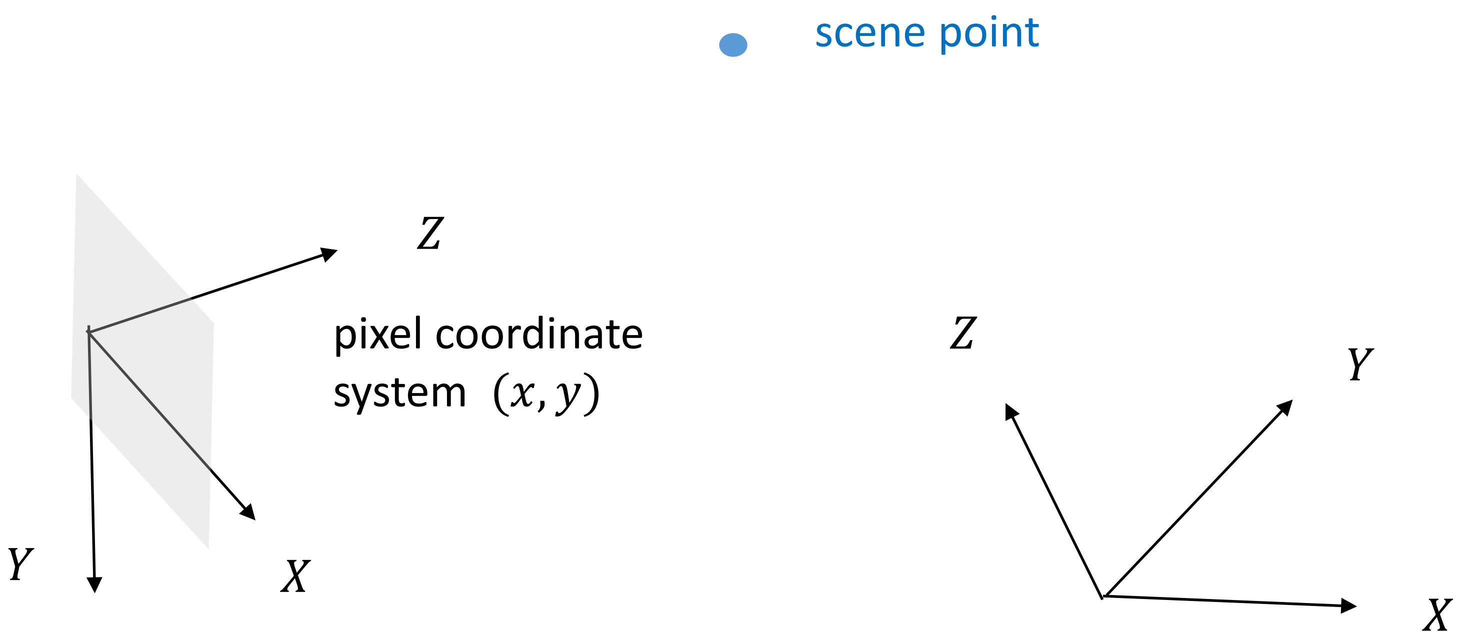
Camera vs. World Coordinates



camera coordinate system

world coordinate system

Pixel vs. Camera vs. World Coordinates



camera coordinate system

world coordinate system

Extrinsic (or external) means written in world coordinates.

Intrinsic (or internal) means written in camera pixel coordinates.

Goal for Today

How do 3D scene points in world coordinates map to pixel coordinates ?

1. Map from world coordinates to camera coordinates

2. Project onto the projection plane.

3. Map from projection plane to pixel coordinates.

Recall: Rotation

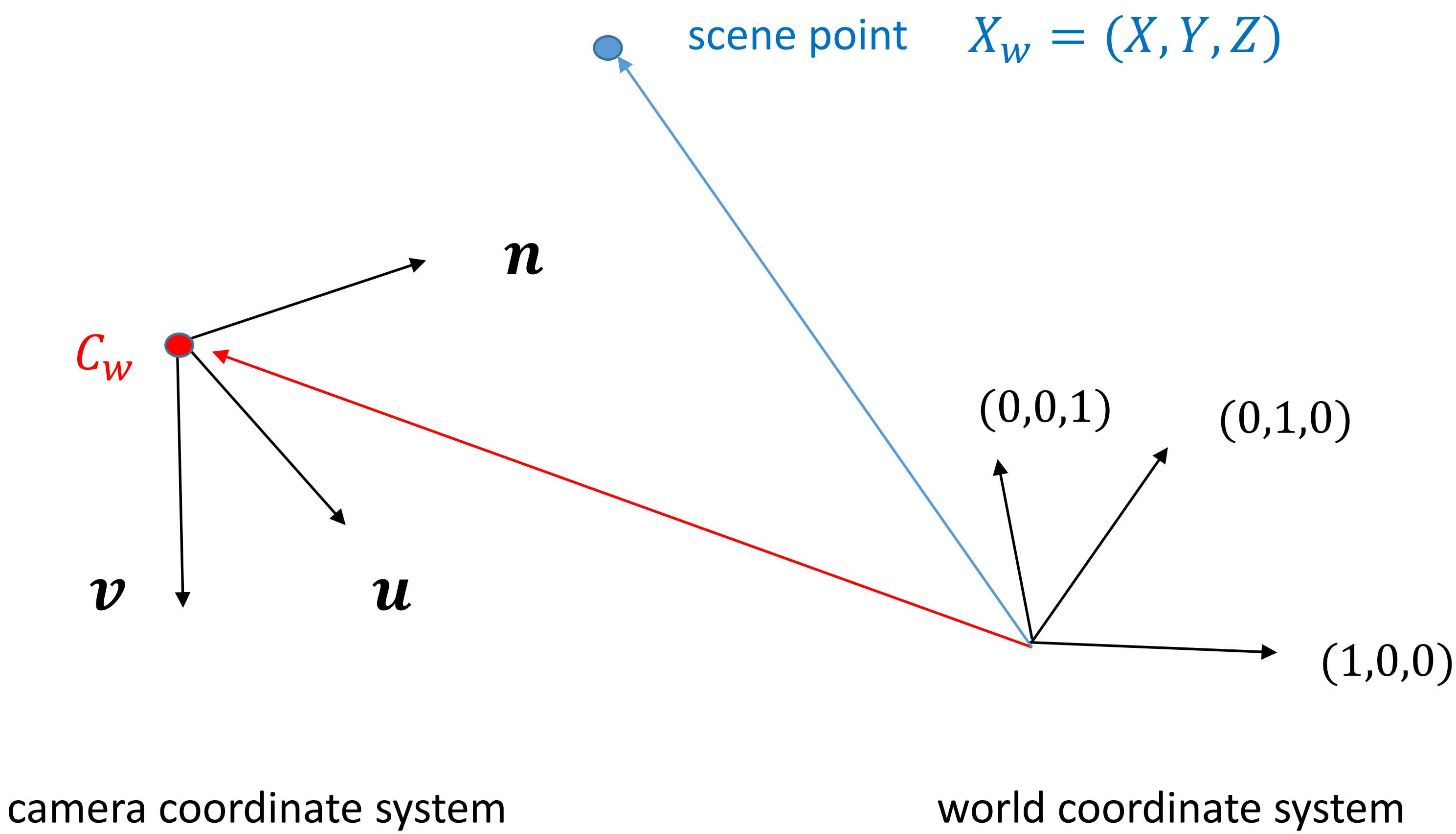
Let 3x3 matrix $\boldsymbol{R}$ have orthonormal rows and columns.

$$\boldsymbol{R} = \begin{bmatrix} \text{——} & \boldsymbol{u} & \text{——} \\ \text{——} & \boldsymbol{v} & \text{——} \\ \text{——} & \boldsymbol{n} & \text{——} \end{bmatrix}$$

$$\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$$

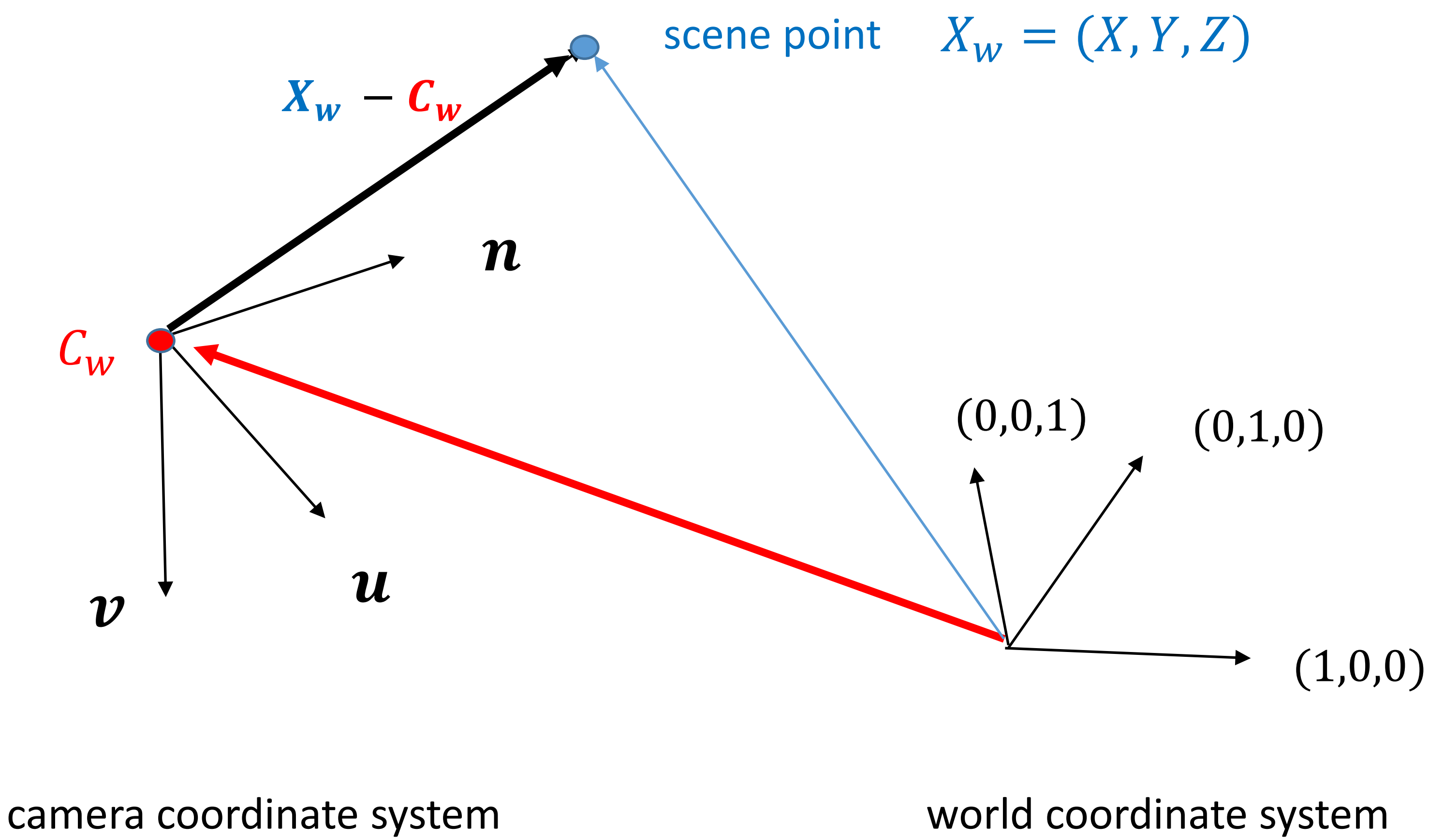
1. Map from world coordinates to camera coordinates

Let $\boldsymbol{X}_w$ be the position of a scene point *in world coordinates*.
Let C_w be the camera position *in world coordinates*.
Let $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{n}$ be the cameras "X, Y, Z" axes *in world coordinates*.

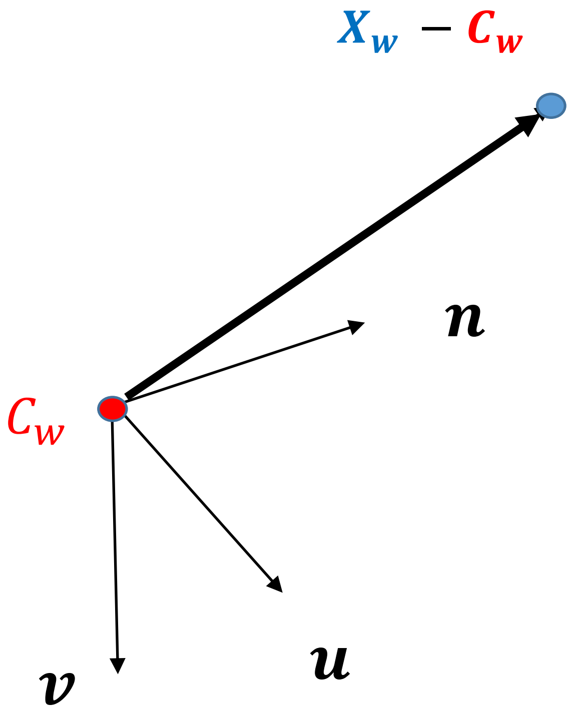


1. Map from world coordinates to camera coordinates

$\boldsymbol{X}_w - \boldsymbol{C}_w$ is the vector from $\boldsymbol{C}_w$ to $\boldsymbol{X}_w$.
This vector is expressed in world coordinates and we want to write it in camera coordinates. How?



We rotate $X_w - C_w$ from world coordinates to camera coordinates.

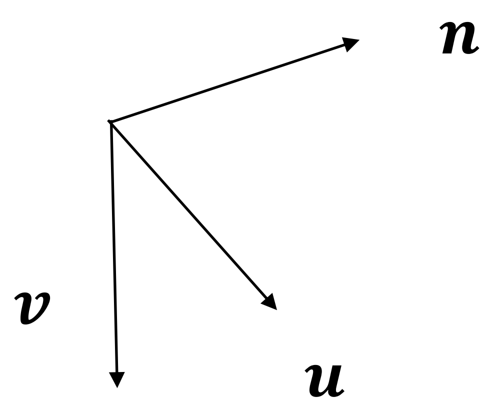


$$\mathbf{X}_c = \mathbf{R}(\mathbf{X}_w - \mathbf{C}_w)$$

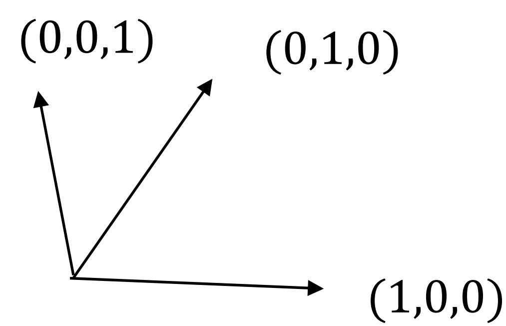
$$\mathbf{R}_{c \leftarrow w}$$

$$\mathbf{R}_{c\leftarrow w} = \begin{bmatrix} — \boldsymbol{u} — \\ — \boldsymbol{v} — \\ — \boldsymbol{n} — \end{bmatrix}$$

The rows of this rotation matrix are the camera coordinate axis unit vectors, $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{n}$ *, written in world coordinates.*



camera coordinate system



world coordinate system

Summary: transformation from world to camera coordinates

$$\mathbf{X}_c = \mathbf{R}(\mathbf{X}_w - \mathbf{C}_w)$$

$$= \mathbf{R}\mathbf{X}_w - \mathbf{R}\mathbf{C}_w$$

Using homogeneous coordinates, we write this as:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \overset{3\times3}{\mathbf{R}} & \overset{3\times1}{-\mathbf{R}\mathbf{C}_w} \\ 0 & 1 \end{bmatrix}}_{4\times4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\mathbf{X}_c = \mathbf{R}(\mathbf{X}_w - \mathbf{C}_w)$$

$$= \mathbf{R}\mathbf{X}_w - \mathbf{R}\mathbf{C}_w$$

Once we have the point in camera (intrinsic) coordinates, we drop the 4th coordinate. This means we don't actually need to use the 4th row of the transformation matrix.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ \cancel{1} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\mathbf{C}_w \\ \cancel{0} & \cancel{1} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

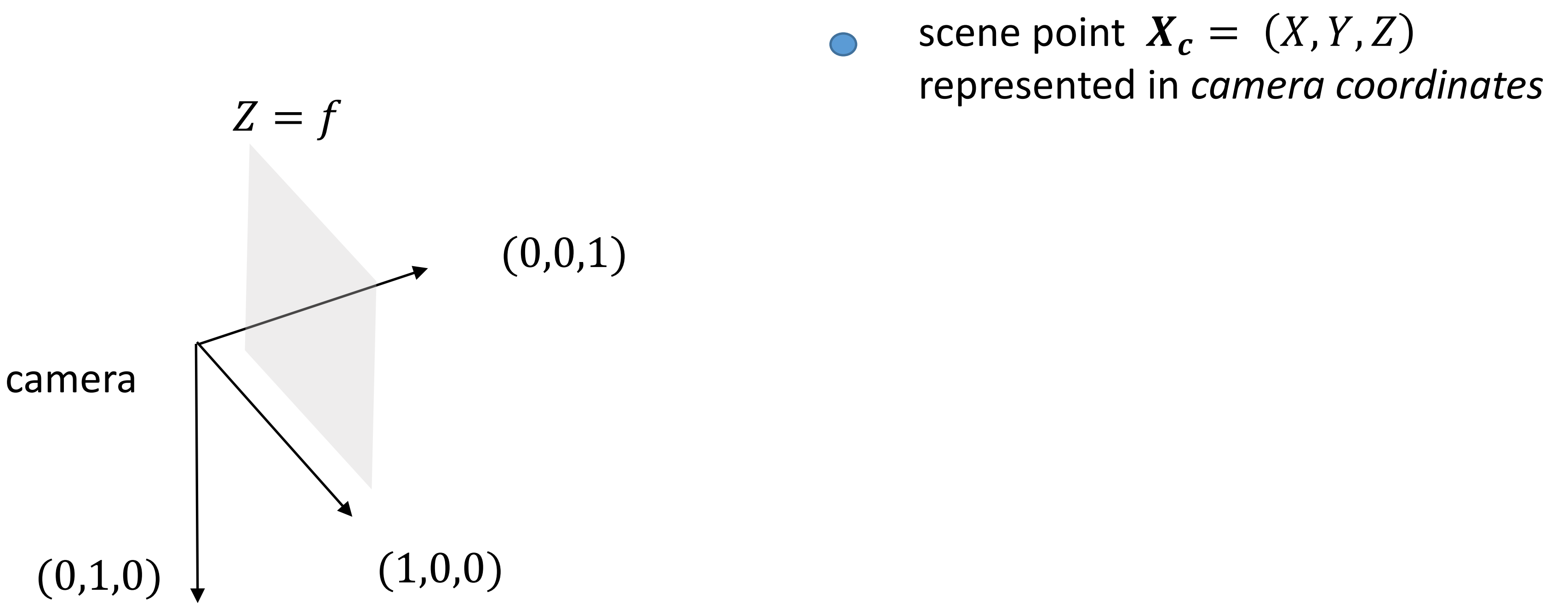
… becomes 3x4

How do 3D scene points (in world coordinates) map to pixel positions ?

1. Map from world coordinates to camera coordinates.
2. Project onto the projection plane.
3. Map from projection plane to pixel coordinates.

How to project $\boldsymbol{X_c} = (X, Y, Z)$ to projection plane $Z = f$?

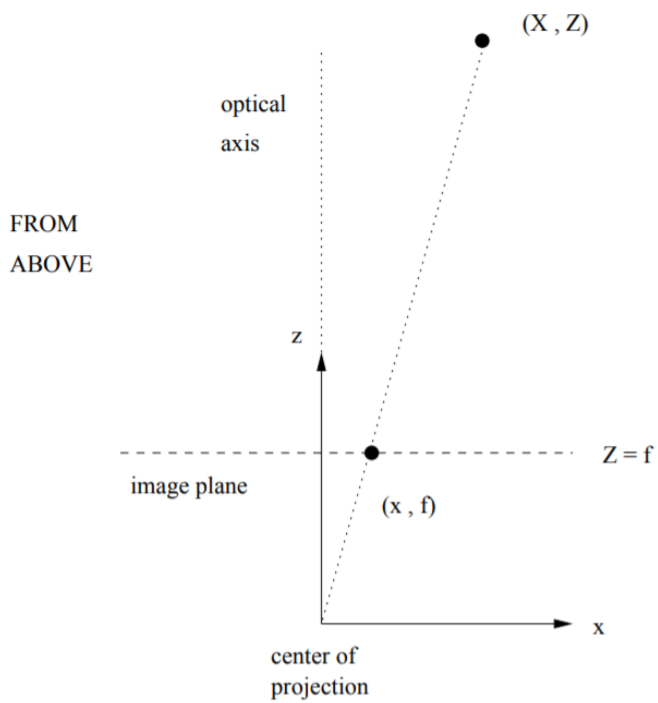
Note: *we use only camera coordinates now.*

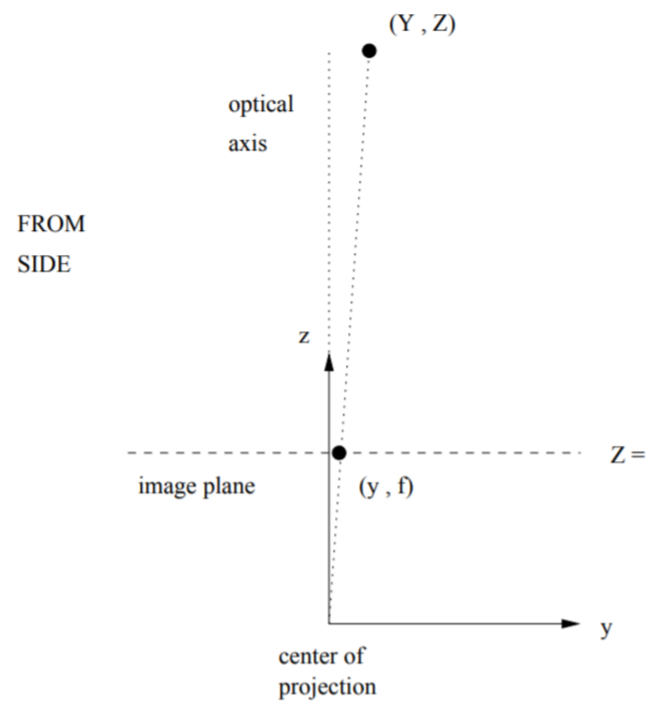


How to project $\boldsymbol{X_c} = (X, Y, Z)$ to projection plane $Z = f$?

Recall lecture 12:

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





How to project $\boldsymbol{X_c} = (X, Y, Z)$ to projection plane $Z = f$?

Let's rewrite this 2D point in homogeneous coordinates:

$$(x, y, 1) = (f\frac{X}{Z}, f\frac{Y}{Z}, 1)$$

If $Z \neq 0$ then

$$(f\frac{X}{Z}, f\frac{Y}{Z}, 1) \equiv (fX, fY, Z).$$

Next, I'll show how we can map $(X, Y, Z, 1)$ to (fX, fY, Z).

Map $(X, Y, Z, 1)$ to (fX, fY, Z).

$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2D homogeneous coordinate representation of our 3D point when it has been projected to the projection plane $Z = f$.

How do 3D scene points (in world coordinates) map to pixel positions ?

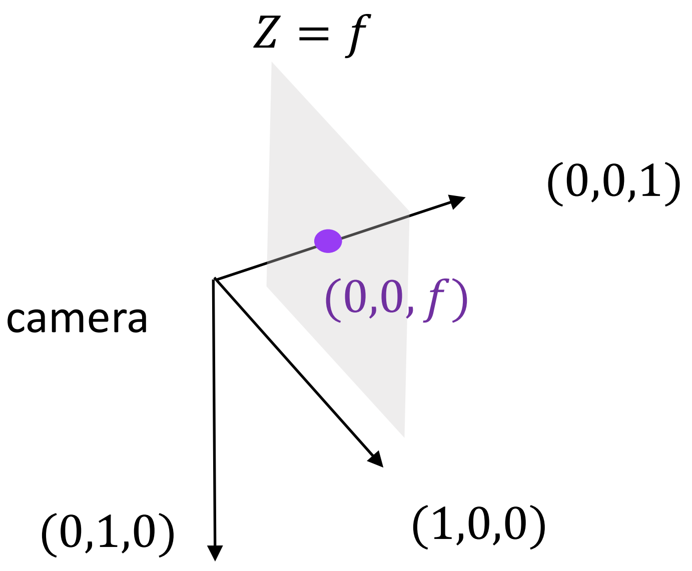
1. Map from world coordinates to camera coordinates.

2. Project onto the projection plane.

3. Map from projection plane coordinates to pixel coordinates.

The units of (x, y) are the same as the units of world and camera coordinates, e.g. millimetres.

We would like to transform them to pixel index coordinates.



projection plane (x, y)

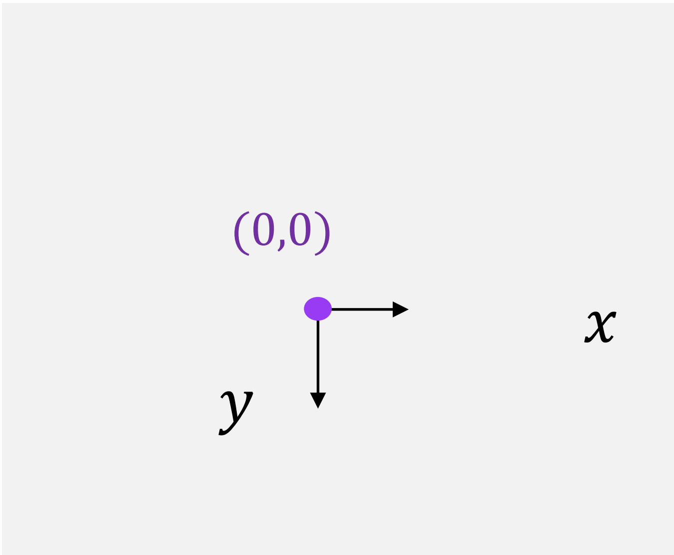
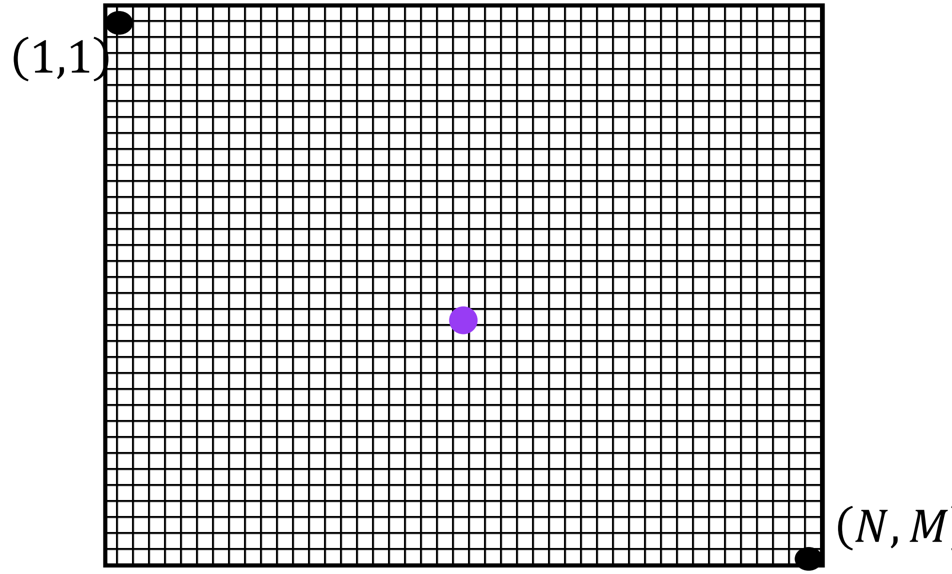


image pixels (x_p, y_p)



The image pixel positions define a rectangular grid in the projection plane.

We want to define a mapping from (x, y) to pixel positions (x_p, y_p).

We want to define a mapping from (x, y) to pixel positions (x_p, y_p).

Two issues:

- Units are different (mm versus pixel indices)

  Apply a scale transformation.

- The origins (0,0) are different.

  Apply a translation.

Units are different (mm versus pixel indices)

Apply a scale transformation.

$$(x, y) \rightarrow \left(m_x\, x,\ m_y\, y\right) \quad \text{where } m_* \text{ is pixels per mm.}$$

The scale factors m_x and m_y might not be exactly the same, i.e. rectangular rather than square lattice.

$(x, y) \rightarrow (m_x\, x,\ m_y\, y)$ where m_* is pixels per mm.

This **scale transformation** would be represented in 2D homogeneous coordinates as:

$$\begin{bmatrix} m_x\, x \\ m_y\, y \\ 1 \end{bmatrix} = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

We want to define a mapping from (x, y) to pixel positions (x_p, y_p).

Two issues:

- Units are different (mm versus pixel indices)

  Apply a scale transformation.

- The origins (0,0) are different.

  Apply a translation.

projection plane

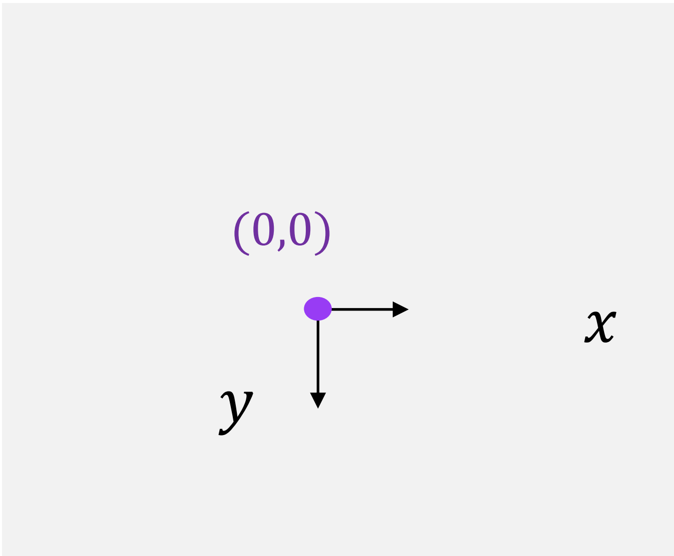
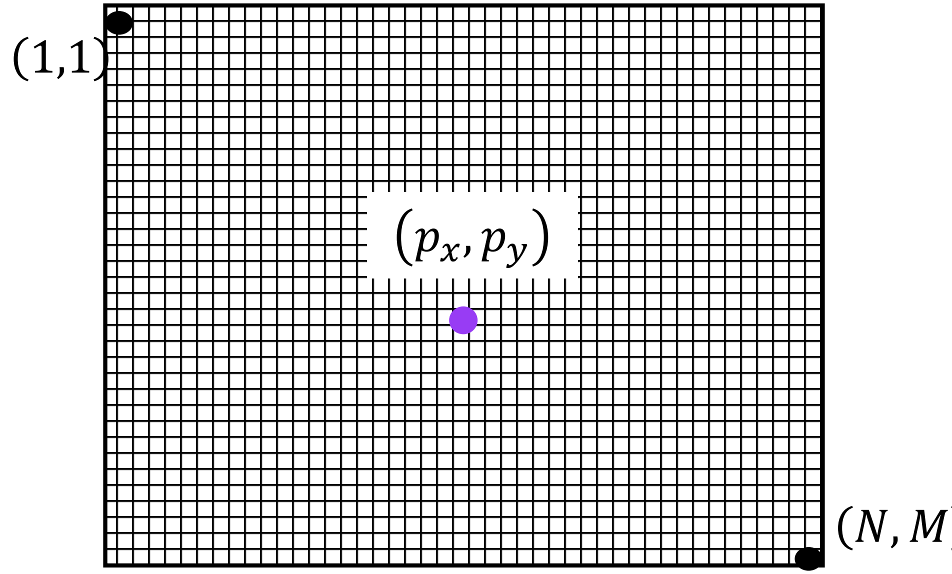


Image pixels



Let (p_x, p_y) be the pixel index that corresponds to $(x, y) = (0,0)$.

This position is called the "principal point". It might not correspond exactly to the center of the pixel grid.

Apply a translation to bring (0,0) to (p_x, p_y).

$$\begin{bmatrix} m_x\, x + p_x \\ m_y\, y + p_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(x_p, y_p)

How do 3D scene points (in world coordinates) map to pixel positions ?

1. Map from world coordinates to camera coordinates.

2. Project onto the projection plane.

3. Map from projection plane coordinates to pixel coordinates.

Let's put these together into one transformation.

Assume we are starting out with a point in camera coordinates, and written in homogeneous coordinates $(X, Y, Z, 1)^T$.

$$\begin{bmatrix} fm_x & 0 & p_x & 0 \\ 0 & fm_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \overset{\text{translation}}{\begin{bmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{bmatrix}} \overset{\text{scaling}}{\begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix}} \overset{\text{projection}}{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}$$

The 3x3 part is called the "camera calibration matrix" $\boldsymbol{K}$.
It invertible but the 3x4 matrix overall is not invertible since it includes projection.

Often one fits a slightly more general model:

Collapses the "scaling constants
$f\ m_x$ and $f\ m_y$ to constants
σ_x and σ_y , respectively.

allows for pixel shear
(small effect only)

$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

intrinsic

(what are the internal properties of the camera?)

extrinsic

(what are the external properties of the camera ie. position and orientation of camera in world coordinates?)

$$\mathbf{P} = \mathbf{K}\, \overbrace{\mathbf{R}\,[\,\mathbf{I} \mid -\mathbf{C}\,]}$$

3x4 3x3 3x3 3x4

The matrix $\mathbf{P}$ is called the “finite projective camera” model.

Exercises

If you are watching this recorded lecture, then I suggest you stop now and try to do the posted Exercises on your own.

(BTW, the Exercise PDF has more than the ones I'll cover here.)

Exercises

What is the null space of $\mathbf{P}$?

For what $\boldsymbol{X} = (X, Y, Z, 1)$, do we have $\boldsymbol{P}\boldsymbol{X} = \mathbf{0}$?

$$\mathbf{P} = \mathbf{K}\,\mathbf{R}\,[\,\mathbf{I} \mid -\mathbf{C}\,]$$

3x4 3x3 3x3 3x4

Exercises

What is the null space of $\mathbf{P}$?

For what $\boldsymbol{X} = (X, Y, Z, 1)$, do we have $\boldsymbol{P}\boldsymbol{X} = \mathbf{0}$?

$$\mathbf{P} \;=\; \mathbf{K}\,\mathbf{R}\,[\,\mathbf{I}\,|\; -\mathbf{C}\,]$$

3x4 3x3 3x3 3x4

Exercise

What is the first column of $\boldsymbol{P}$?

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{12} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Exercise

What is the last (4th) column of $\boldsymbol{P}$?

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{12} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Exercise

What is the first row of $\boldsymbol{P}$?

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{12} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Exercise

What is the last row of $\boldsymbol{P}$?

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{12} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Reminder

- Quiz 3 is on Wednesday (lectures 10-13).
- “Open book” means you can use the course PDF materials. It does not mean you can search the web.