Università degli Studi di Padova

Projective geometry

Stefano Ghidoni

Agenda

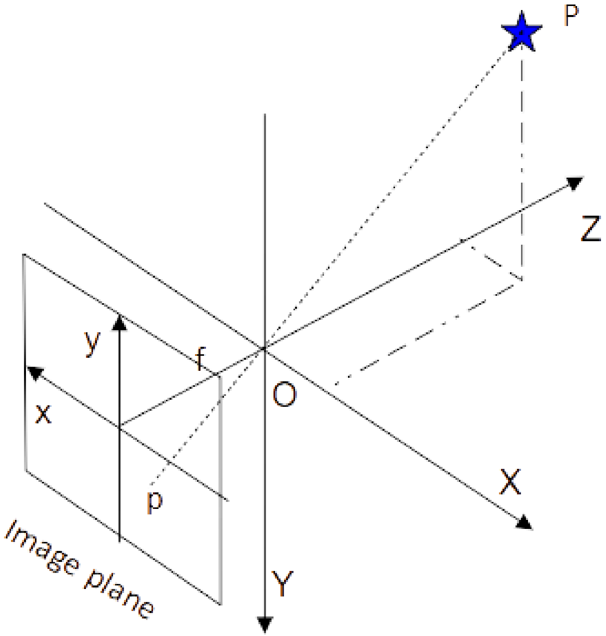
- Geometry of projection
- Reference systems and transformations for
  - Modeling the projection
  - Modeling the sensor
  - Modeling the camera orientation

Describing projection

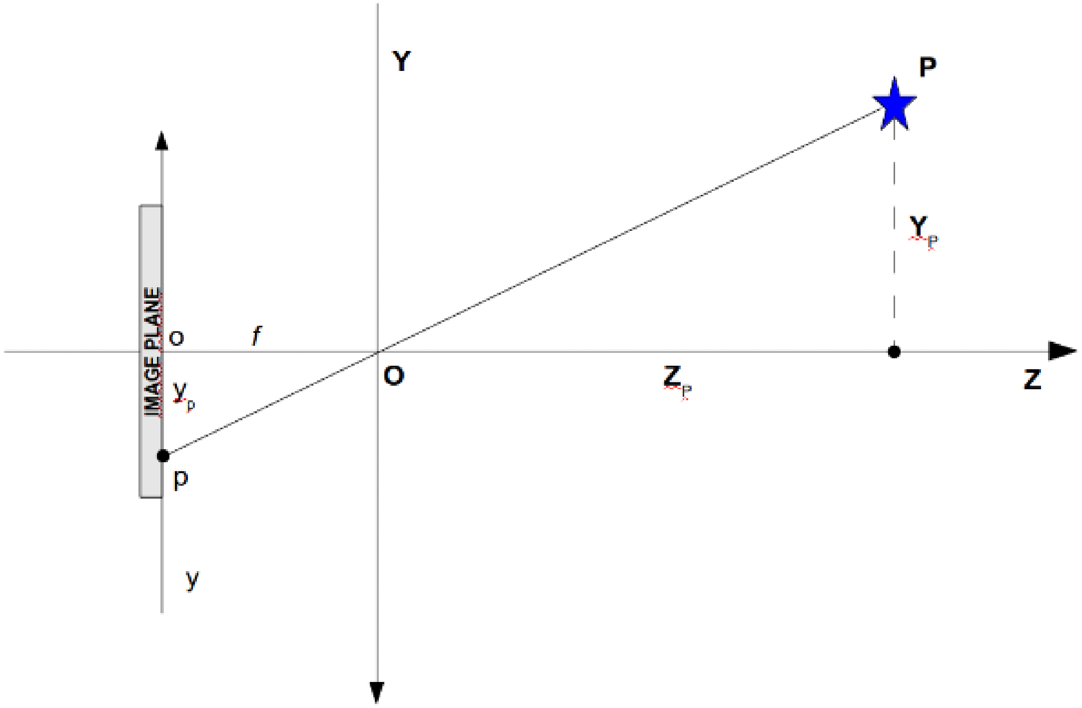
- We need to describe the geometry of projection quantitatively
- First element: relation between the two reference systems
  - 3D point in the world **seen from the camera**
  - 2D point on the image plane

Camera projection

Perspective view



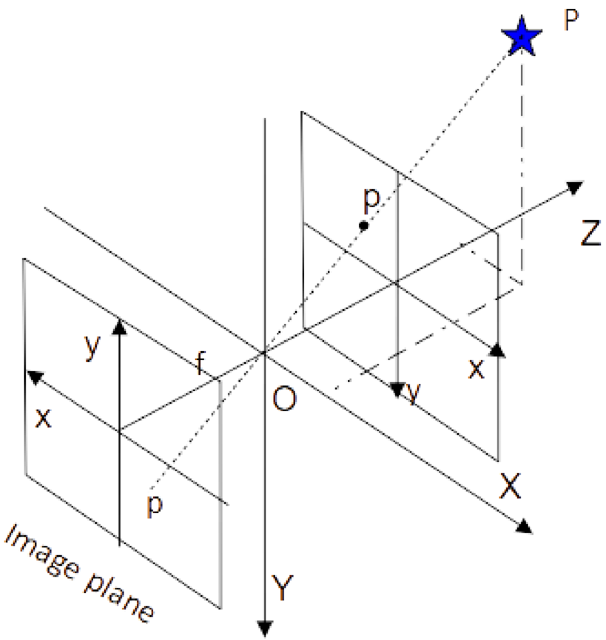
Side view



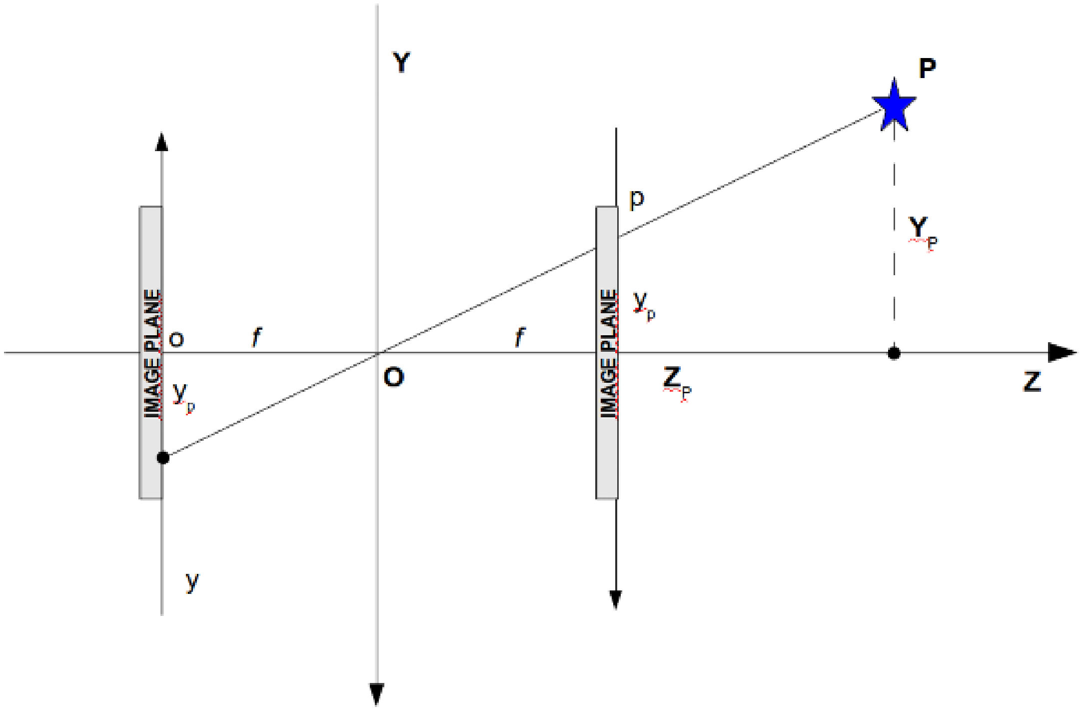
- Consider a point P and its projection p

Camera projection

Perspective view



Side view

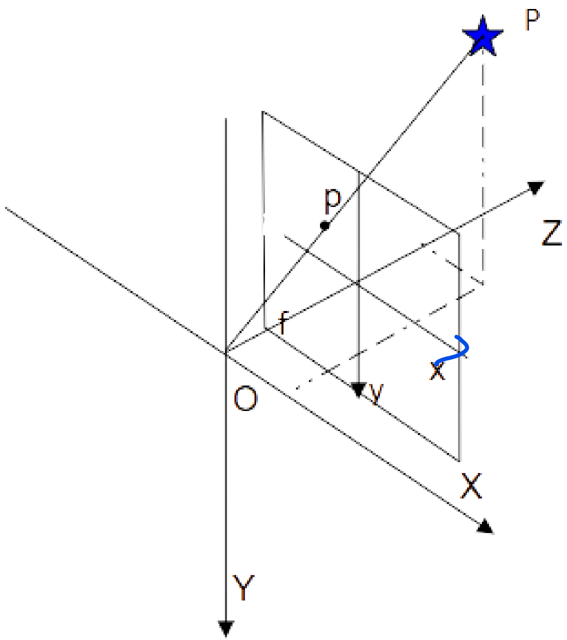


- Now consider a plane that is
  - Parallel to the image plane
  - In front of the optical center
  - At the same distance f from the optical center

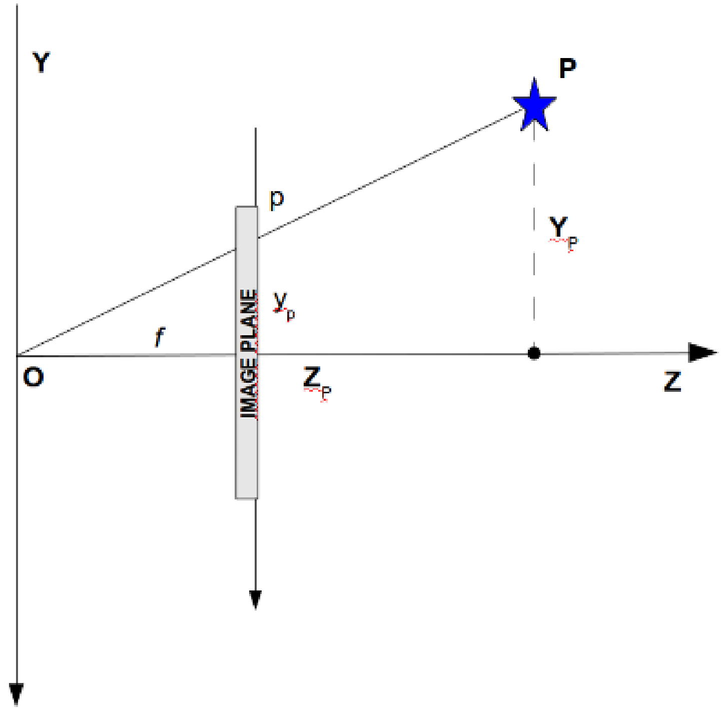
- Easier to work on this plane
  - Same geometrical relation
  - Avoid the upside-down effect

Camera projection

Perspective view



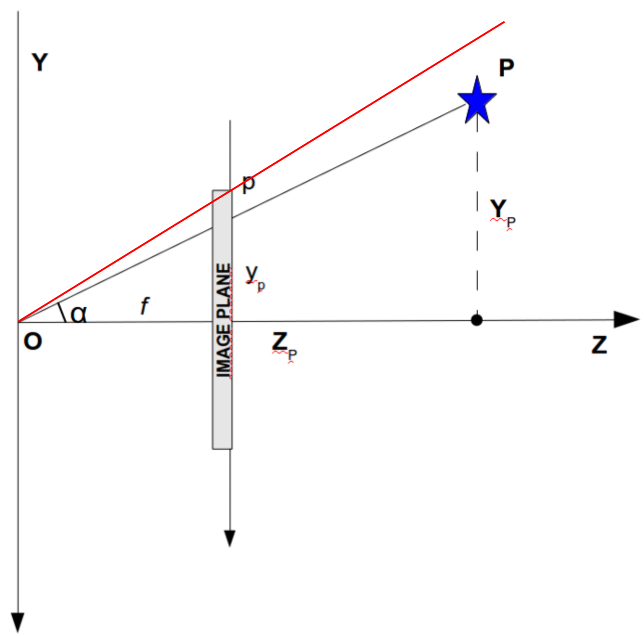
Side view



- We move to this plane for deriving the geometrical description

Field of view

- The Field of View (FoV) of a camera is the angle perceived by the camera
- Define α as the angle under which a point P is seen
- The maximum value for α is ½ of the FoV

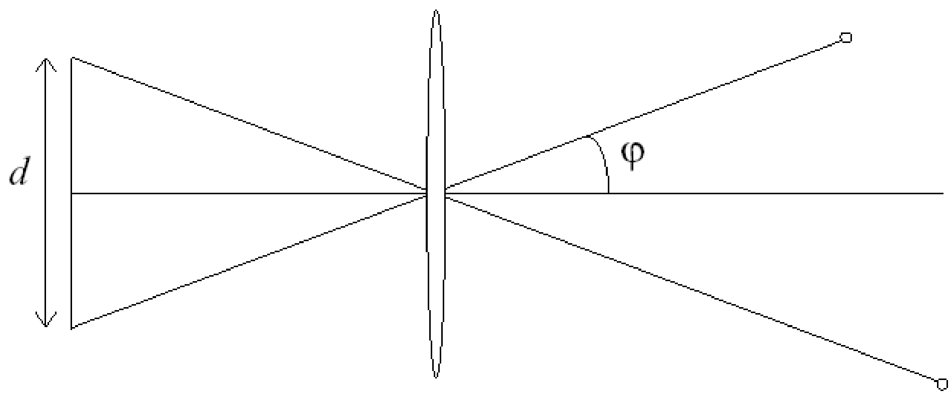


Field of view

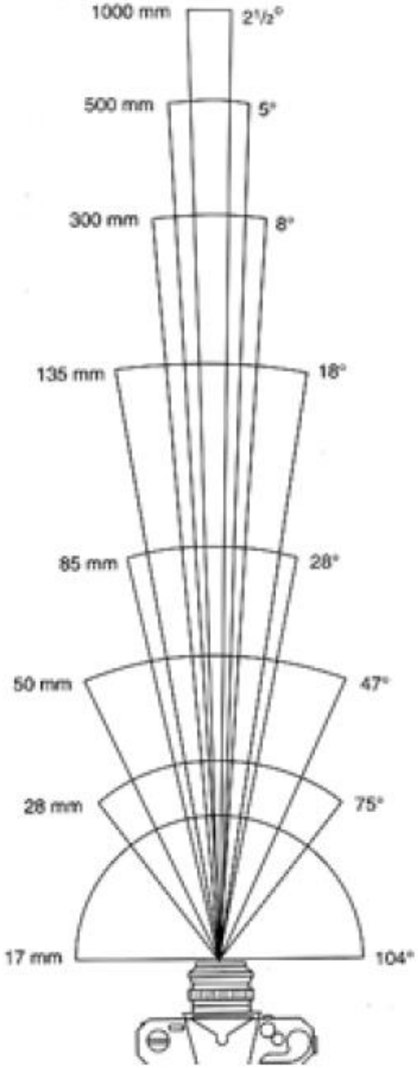
- The FoV depends on (FoV = 2φ)
  - The sensor size d
  - The focal length f

φ is the maximum value for α

$$\varphi = \arctan\left(\frac{d}{2f}\right)$$



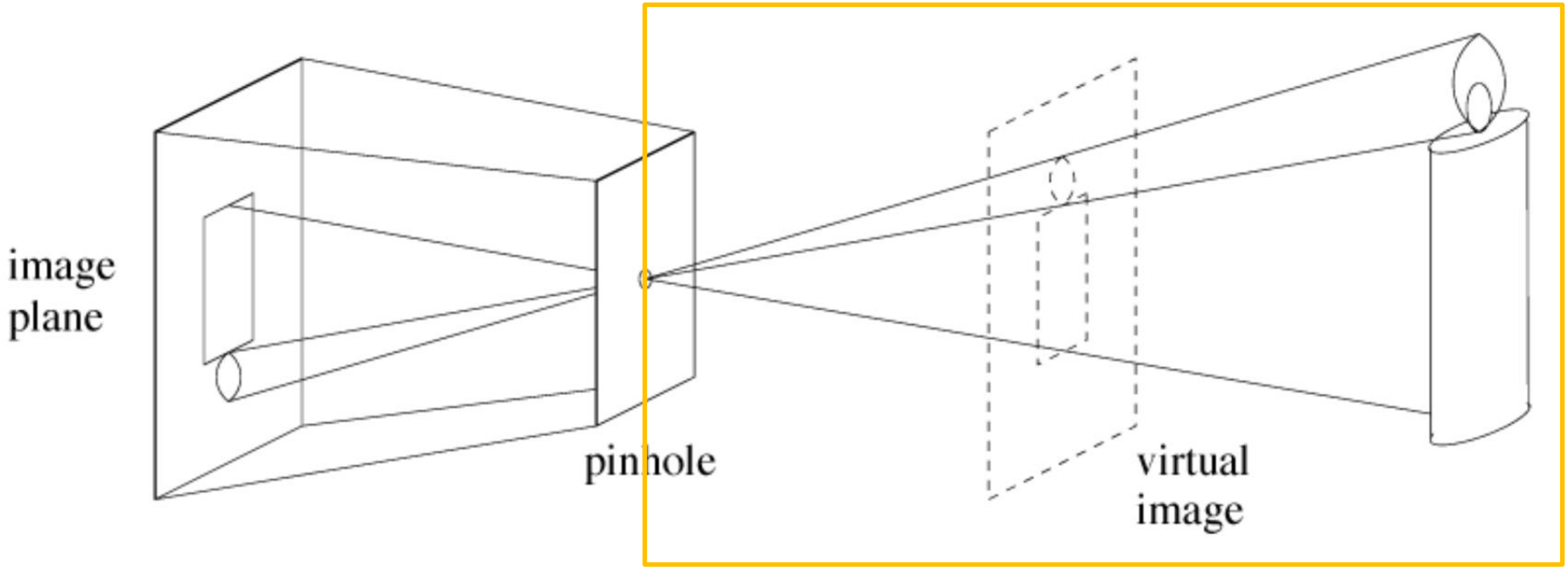
Field of view



From London and Upton

Camera projection

(recall)

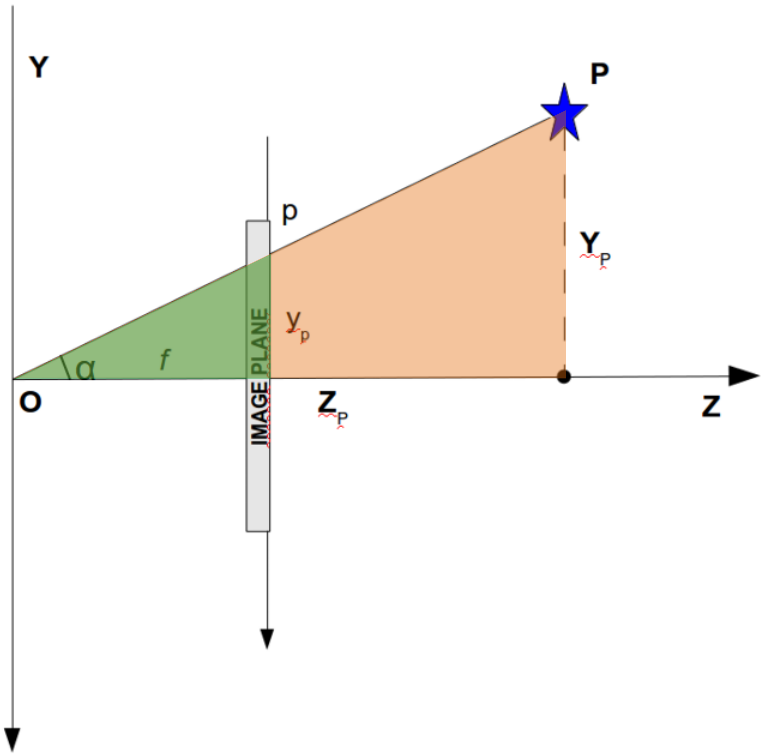


Camera projection

- Similar triangle rule

$$\frac{Y_p}{y_p} = \frac{Z_p}{f}$$

  - Analogous for x



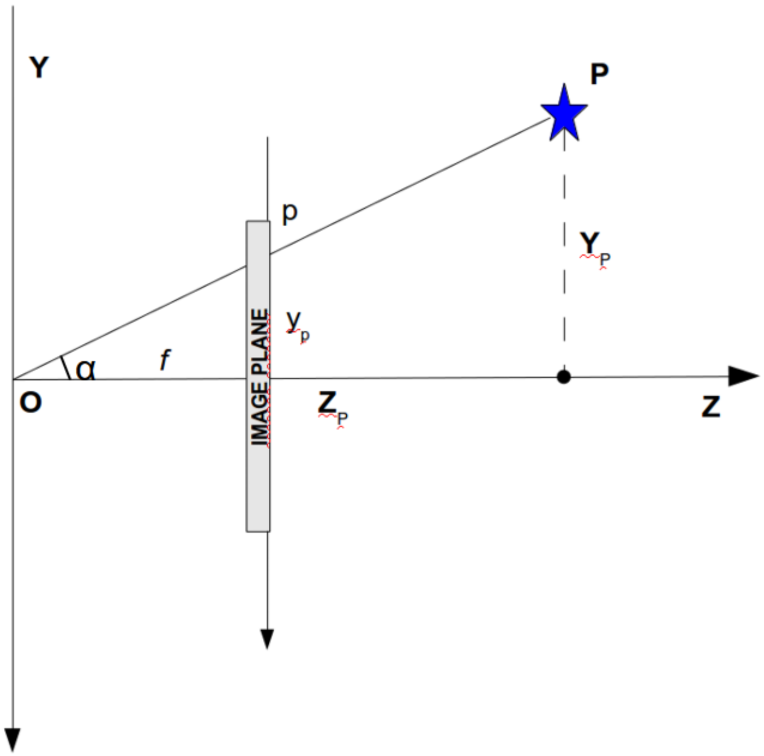
Camera projection

- Therefore:

$$x_p = f\,\frac{X_p}{Z_p}$$

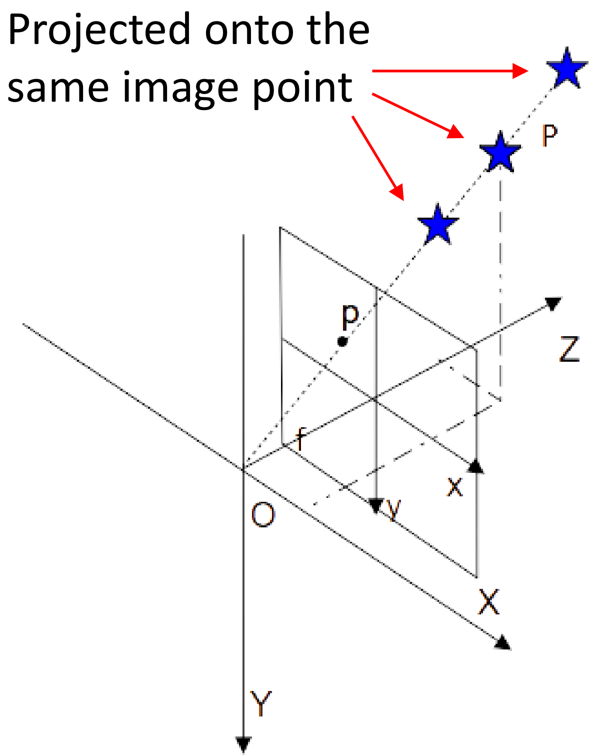
$$y_p = f\,\frac{Y_p}{Z_p}$$

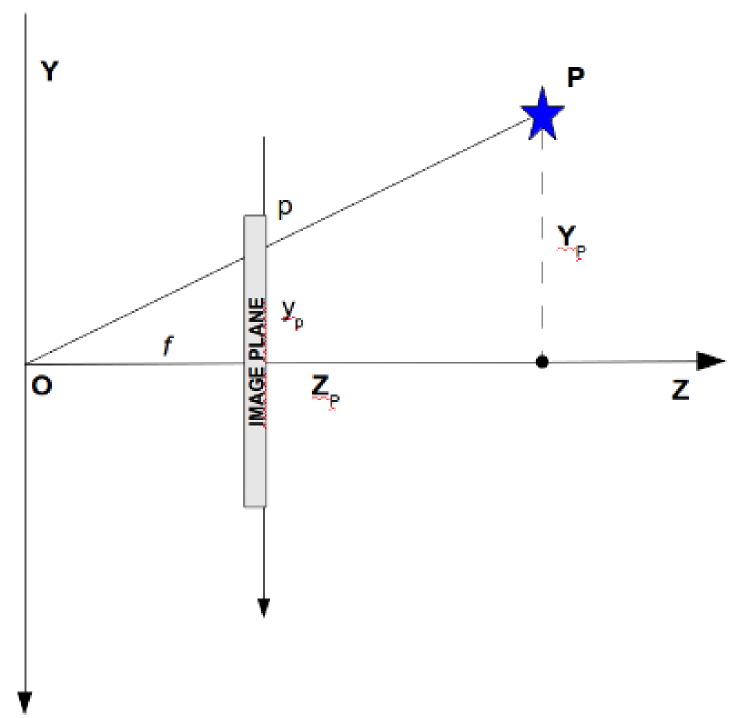
$$\tan(\alpha) = \frac{Y_p}{Z_p}$$



Loosing a dimension

- Projecting points on a 2D surface causes the loss of the distance information





Homogeneous coordinates

- The equations can be rearranged in matrix form using the homogeneous coordinates
- Points in 2D can be expressed in **homogeneous coordinates**
  - A "mathematical trick"

Homogeneous coordinates in 2D

- To homogeneous coordinates

$$\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} \tilde{w}x \\ \tilde{w}y \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{w} \end{bmatrix}$$

- From homogeneous coordinates

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{w} \end{bmatrix} \Rightarrow \begin{bmatrix} \tilde{x}/\tilde{w} \\ \tilde{y}/\tilde{w} \end{bmatrix}$$

Homogeneous coordinates

- Homogeneous coordinates can be extended to N dimensions
  - N-dimensional point transformed into (N+1) homogeneous coordinates
- We now want to exploit homogeneous coordinates to rewrite the equations:

$$x_p = f\frac{X_p}{Z_p}$$

$$y_p = f\frac{Y_p}{Z_p}$$

Projection matrix

- The equations can be rearranged as:

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

$$\Rightarrow \quad Z\boxed{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}} = Z\begin{bmatrix} \frac{fX}{Z} \\ \frac{fY}{Z} \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \boxed{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}} \boxed{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}$$

$$\boxed{\tilde{\boldsymbol{m}}} \simeq \boxed{P}\boxed{\tilde{\boldsymbol{M}}}$$

Equal to a scale factor (Z is removed)

P is the **projection matrix**

Projection matrix

- When f=1 we obtain the **essential perspective projection**

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = [I|\mathbf{0}]$$

- This represents the core of the projection process

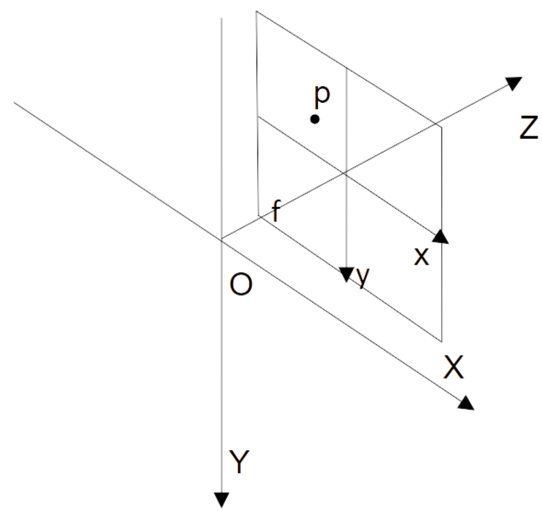
Reference systems and meas. units

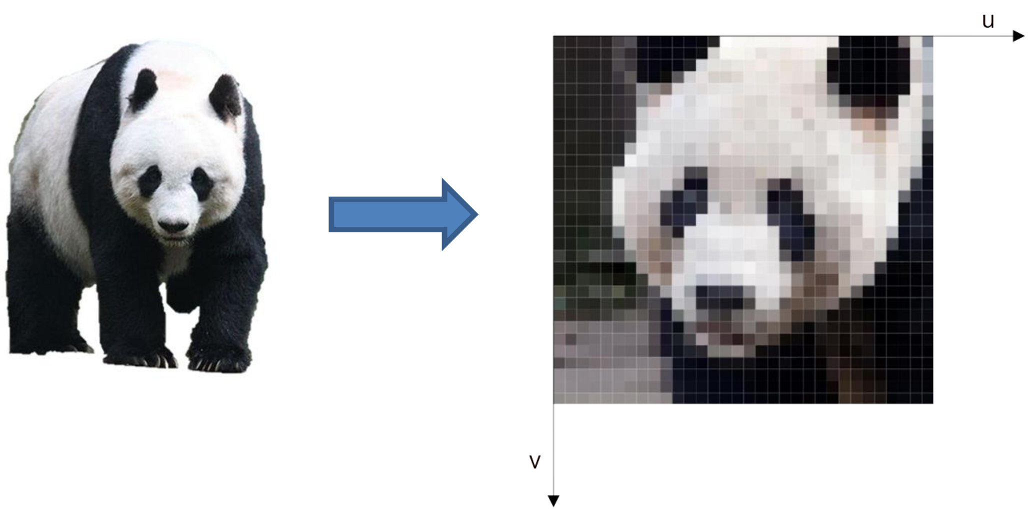
- So far: the projection matrix describes how the 3D world is mapped onto the image plane
- Now reflect:
  - What measurement unit is used for distances in the 3D world?
  - What measurement unit is used for distances on the projection plane?
  - What measurement unit do we commonly use for distances in a digital image?

- Anti-spoiler ☺

Mapping to image coordinates

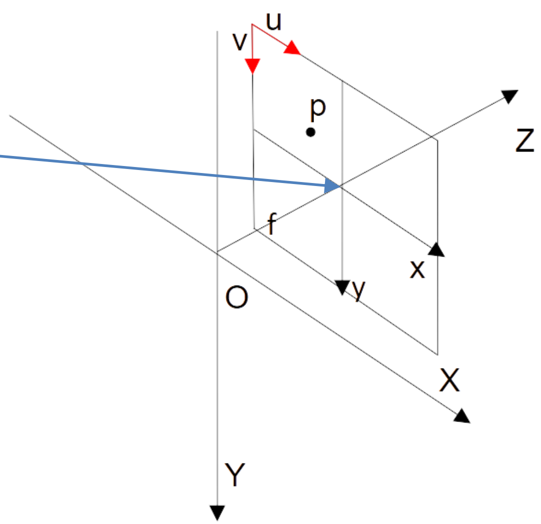
- We need to map points projected onto the image plane in the coordinates used for pixels
- From (x, y) to (u, v)





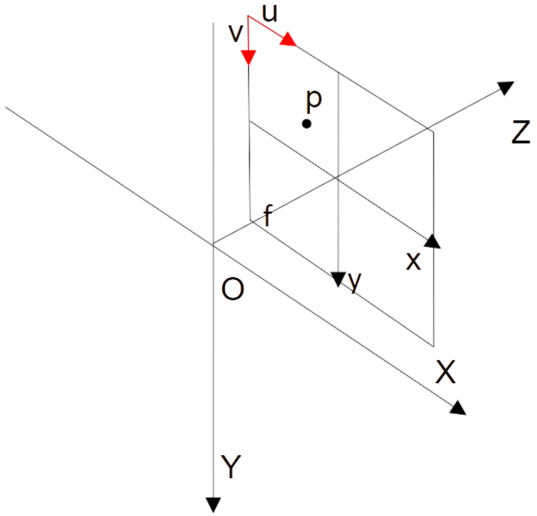
Mapping to image coordinates

- The transformation can be defined considering the coordinates of the principal point to be (u_0, v_0)
- The origin is in the top-left corner



Mapping to image coordinates

- Metric distances are converted to pixels using the pixel width w and h height
- Evaluate the coordinates of the principal point in pixel
- Conversion factors are usually defined as
  - $k_u = \frac{1}{w}$
  - $k_v = \frac{1}{h}$

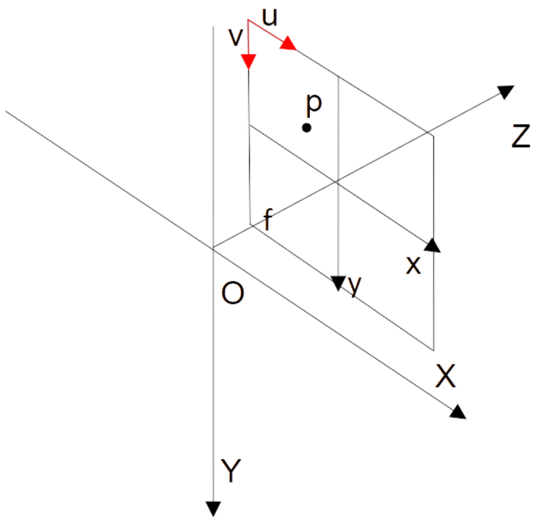


Mapping to image coordinates

- Mapping from (x, y) to (u, v) is obtained by translation and scaling:

$$u = u_0 + \frac{x_p}{w} = u_0 + k_u x_p$$

$$v = v_0 + \frac{y_p}{h} = v_0 + k_v y_p$$



From 3D to pixel coordinates

- We can combine the mappings:
  - From 3D to 2D image plane
  - From image plane to pixels

By substituting the last equations into the projection equation

From 3D to pixel coordinates

$$u = u_0 + k_u x_p = u_0 + k_u f \frac{X_p}{Z_p} = u_0 + f_u \frac{X_p}{Z_p}$$

Where $k_u f \triangleq f_u$ is the focal length **in pixels**

- Summarizing and applying a similar conversion for v yields:

$$u = u_0 + f_u \frac{X_p}{Z_p}$$

$$v = v_0 + f_v \frac{Y_p}{Z_p}$$

Camera matrix

- The projection is now expressed as

$$P = \begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \triangleq K[I|\mathbf{0}]$$

Where K is the **camera matrix**

- The previous equation $\widetilde{\boldsymbol{m}} \simeq P\widetilde{\boldsymbol{M}}$ still holds
  – Just a different formulation for P

Camera matrix

- We moved from:

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

To

$$P = \begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \triangleq K[I|\mathbf{0}]$$

- Compare the two matrices (concepts embedded in the matrix elements)

Camera matrix

- Consider the camera matrix

$$K = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

- How many parameters are involved?

- Anti-spoiler ☺

Camera matrix

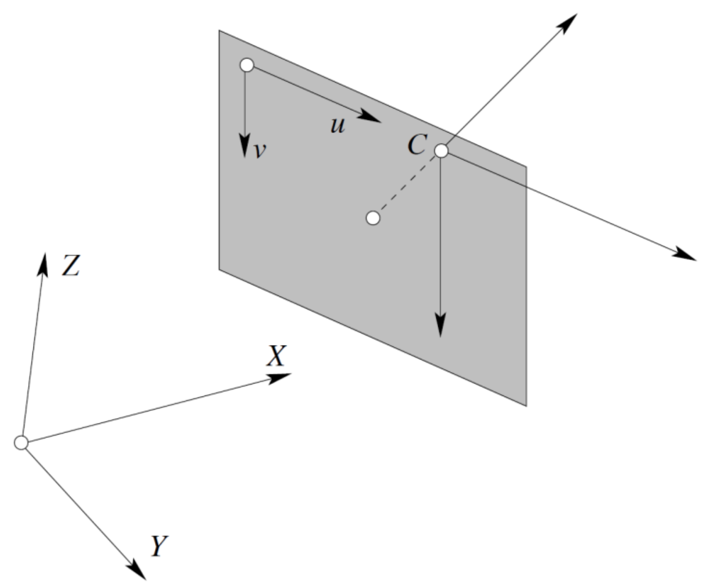
- Consider the camera matrix

$$K = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

- K depends on: k_u, k_v, u_0, v_0, f
  - They are called intrinsic parameters
    - Define the projection characteristics of the camera
  - Highlight: f_u, f_v embed three parameters

Camera vs real world

- So far, we mapped
  - World to image plane
  - Image plane to pixels
- However, a different reference frame can be defined on the world
- This is always defined as a **rototranslation**



Camera vs real world

- A rototranslation in 3D in homogeneous coordinates is expressed as

3x3

$$T = \begin{bmatrix} R & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

3x1

- The correspondence becomes

$$\widetilde{\boldsymbol{m}} \simeq PT\widetilde{\boldsymbol{M}}$$

Rototranslation params

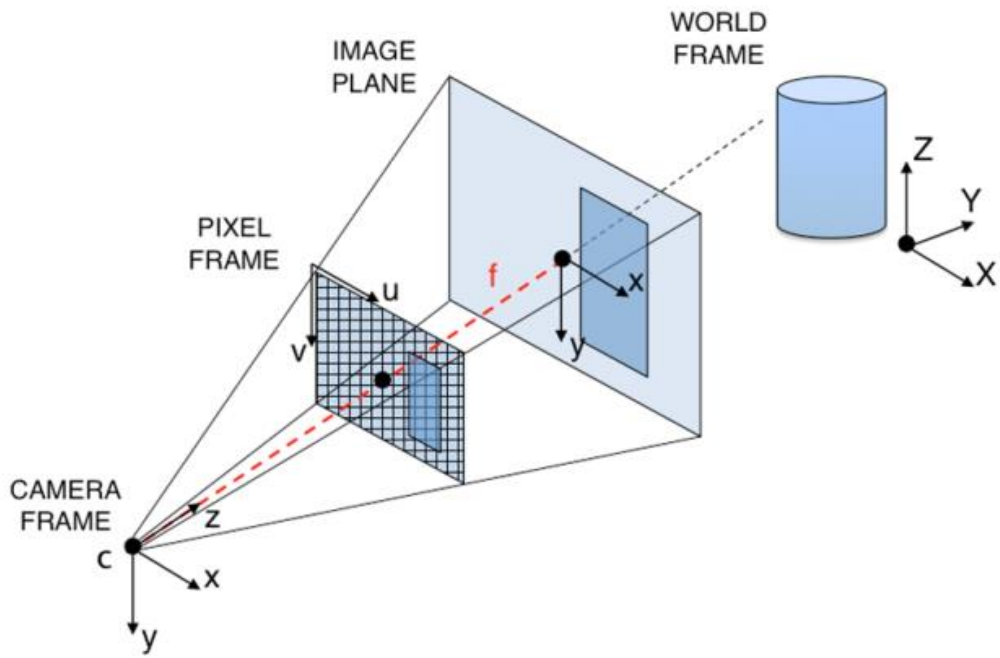
- Consider the rototranslation matrix T
- How many parameters are involved?

Rototranslation params

- Consider the rototranslation matrix T
- How many parameters are involved?
  - 3 for translations
  - 3 for rotations
- They are called extrinsic parameters
  - Define the relation between camera and world

Projection recap

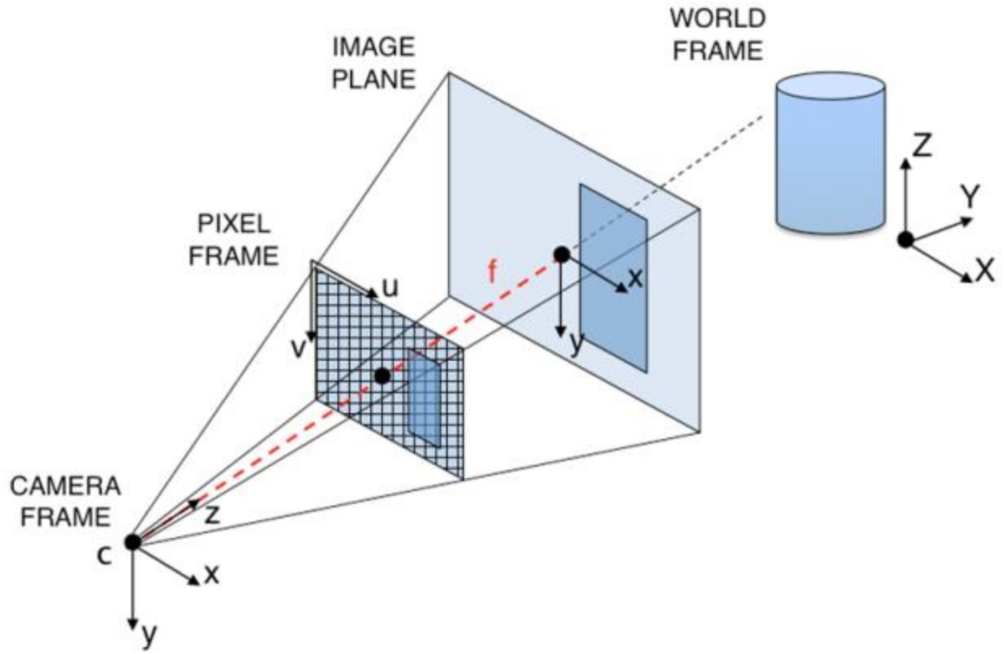
- The whole projection process involves
  - Four reference systems
  - Three transformations



Haoxuan Li et al., "An integration system of AI cars detection with enclosed photogrammetry for indoor parking lot", 2nd Int. Conference on Computing and Data Science

Projection recap

From world coordinates to camera coordinates	From camera coordinates to sensor plane	From sensor plane to sensor coordinates (pixels)
3D->3D	3D->2D	2D->2D discrete



Haoxuan Li et al., "An integration system of AI cars detection with enclosed photogrammetry for indoor parking lot", 2nd Int. Conference on Computing and Data Science

Inverse projection

- The projection process is described as:

$$\widetilde{\boldsymbol{m}} \simeq PT\widetilde{\boldsymbol{M}}$$

- Evaluates the projected point ($\widetilde{\boldsymbol{m}}$) given the 3D point ($\widetilde{\boldsymbol{M}}$)
- Is it possible to invert the transformation?

- Anti spoiler ☺

Inverse projection

- Two elements are not invertible:
  - Projection from 3D to 2D
  - Pixel quantization

Inverse projection

- Two elements are not invertible:
  - Projection from 3D to 2D
  - Pixel quantization
- We can invert the projection if
  - We accept as a result the direction of the object, not the 3D position, or
  - We have additional constraints providing the location on the line

Inverse projection

- Two elements are not invertible:
  - Projection from 3D to 2D
  - Pixel quantization
- We can invert the projection if
  - We neglect the quantization effect: the pixel location and the projected point are considered the same
    - Acceptable for high-resolution sensors

UNIVERSITÀ DEGLI STUDI DI PADOVA

Projective geometry

Stefano Ghidoni