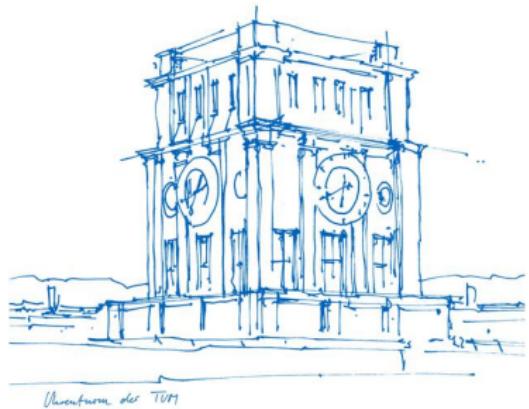


# Computer Vision II: Multiple View Geometry (IN2228)

Chapter 03 Image Formation  
(Part 1 Perspective Projection)

Dr. Haoang Li

03 May 2023 12:00-13:30



# Announcement before Class

Today, we will have the **exercise session** about Mathematical Background

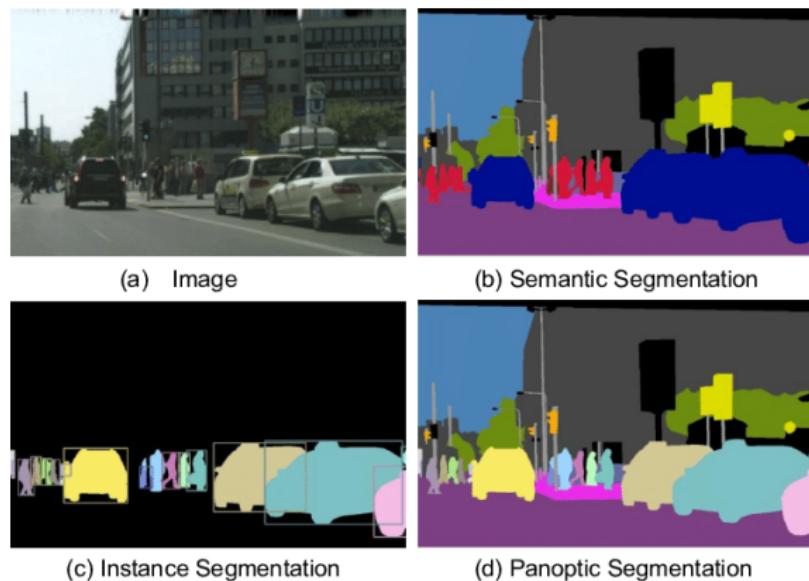
- ✓ Time: from 16:00 to 18:00
- ✓ Room: 102, Hörsaal 2, "Interims I" (5620.01.102)

# Explanations before Class

- Clarification of labels in semantic segmentation



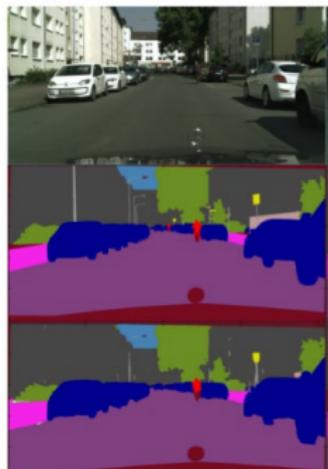
Images presented in our previous class



# Explanations before Class

- Clarification of labels in semantic segmentation

The prediction results depend on the type of the ground truth.

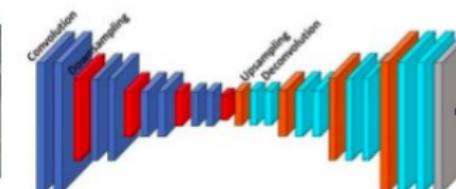


Input RGB  
image

Ground truth

Prediction

An example



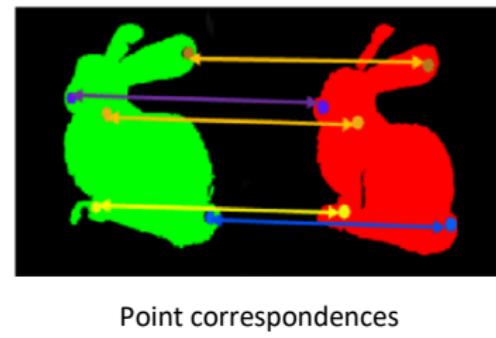
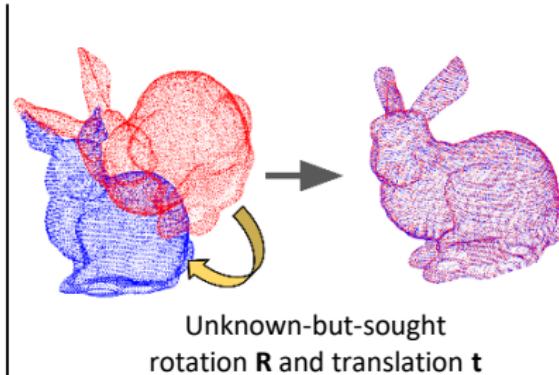
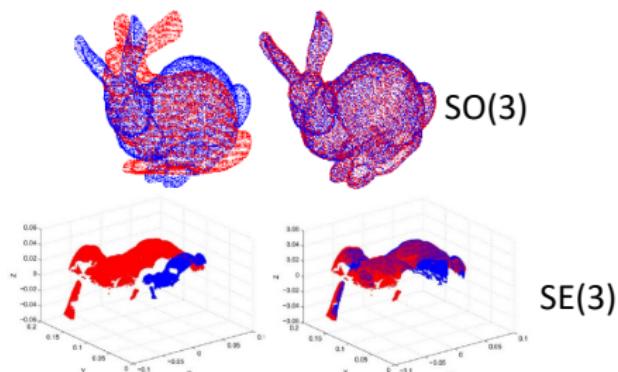
Neural network (a “fitter”) for label prediction

For more knowledge, please attend the course “Computer Vision III: Detection, Segmentation and Tracking” provided by Dr. Nikita Araslanov

# Explanations before Class

- Clarification of General Pipeline

Model selection -> Data fitting (parameter estimation)

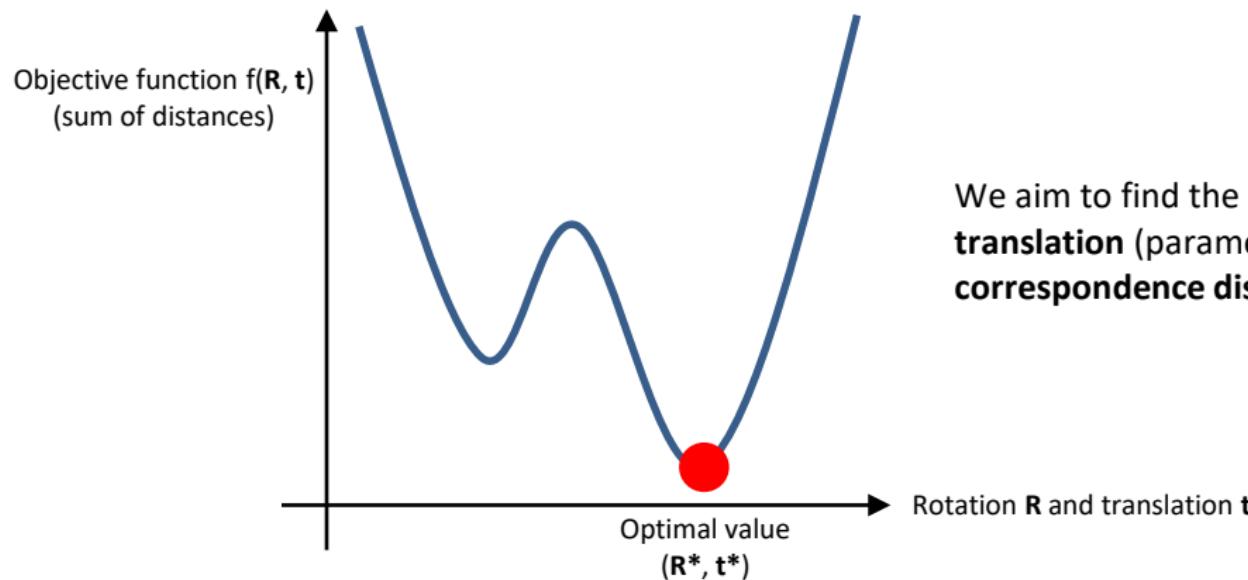




# Explanations before Class

- Clarification of General Pipeline

Model selection -> Data fitting (parameter estimation)



We aim to find the optimal **rotation and translation** (parameters) to minimize the sum of correspondence distances (objective function).

# Today's Outline

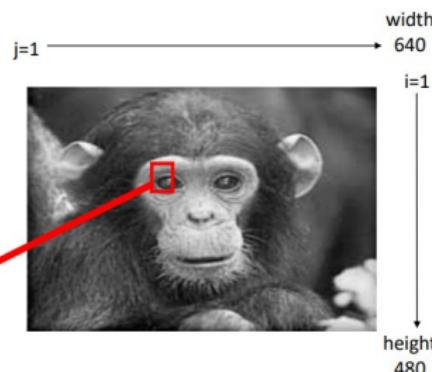
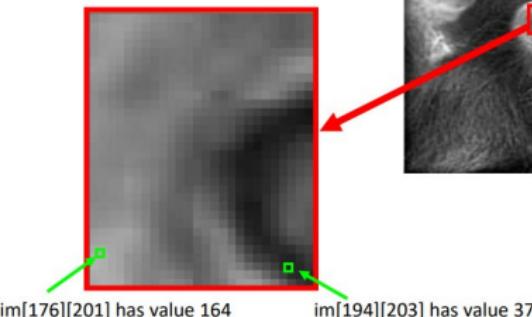
- Recap on Digital Images
- Pinhole Camera
- Perspective Projection



# Recap on Digital Images

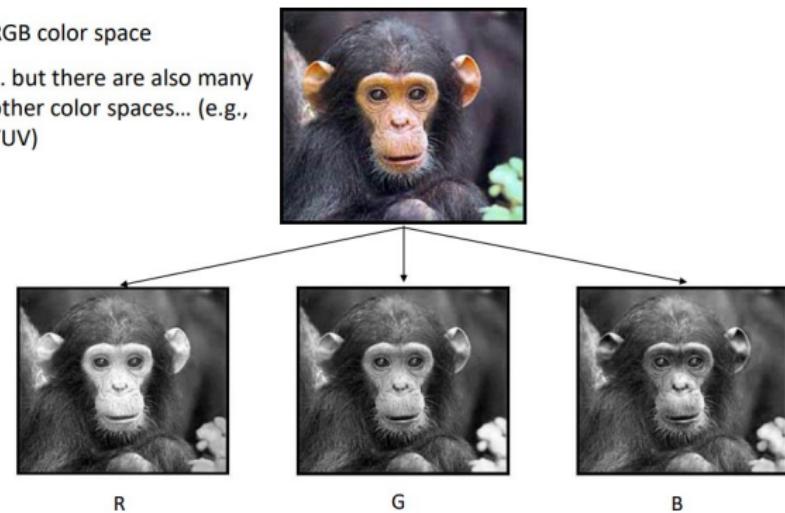
- Pixel Intensity and RGB channels

Pixel Intensity with 8 bits  
ranges between [0,255]



Pixel intensity

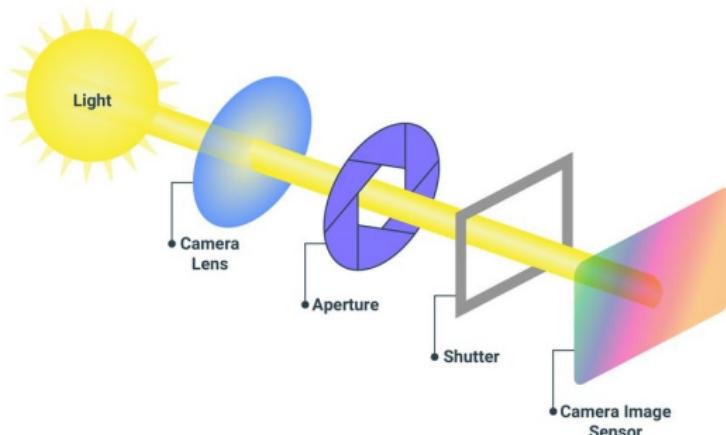
RGB color space  
... but there are also many  
other color spaces... (e.g.,  
YUV)



RGB Channels

# Recap on Digital Images

- From Light Signal to Electrical Signal
- ✓ Basic configuration



- **Aperture** controls the area over which light can enter your camera
- **Shutter speed** controls the duration of the exposure

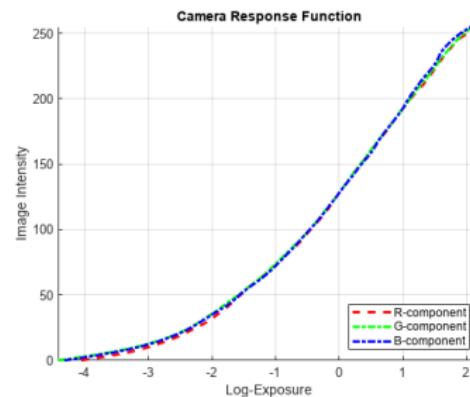
Exposure can be explained as the amount of light collected by a camera center.

# Recap on Digital Images

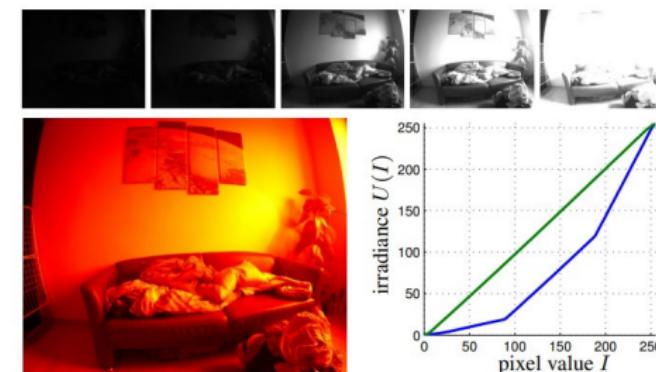
- From Light Signal to Electrical Signal

- ✓ Response function

The camera response function maps the log-exposure value (scene radiance) to the intensity levels in the input images.



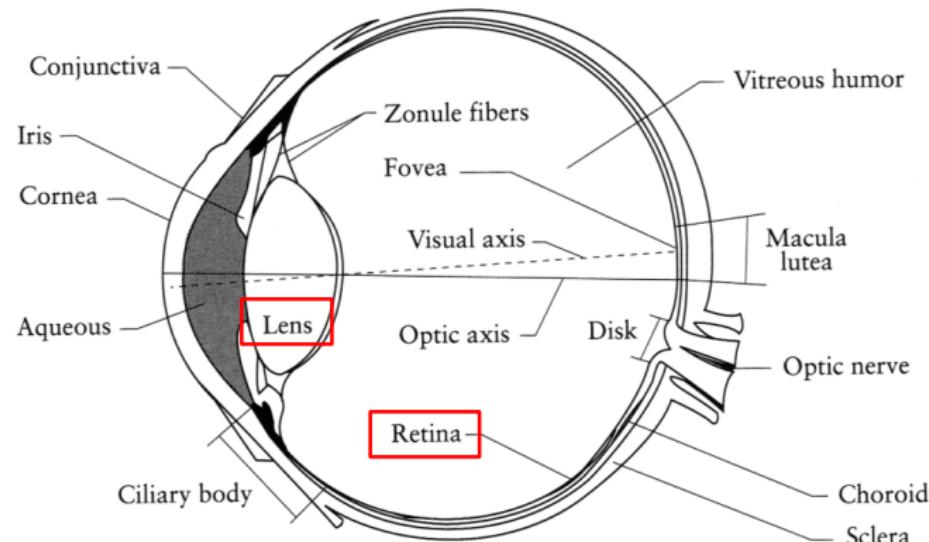
Response function



Photometric calibration

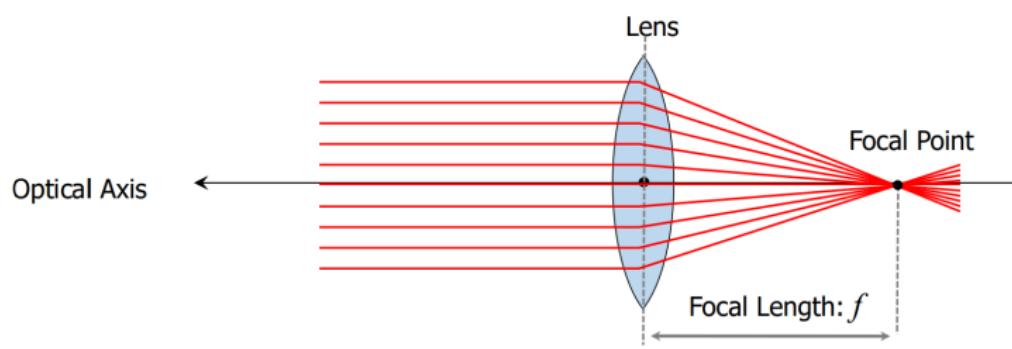
# Pinhole Camera

- Human Eye
- ✓ The human eye is a camera
  - Pupil corresponds to the “**aperture**” whose size is controlled by the iris
  - Photoreceptor cells in the **retina** correspond to the “**film**”

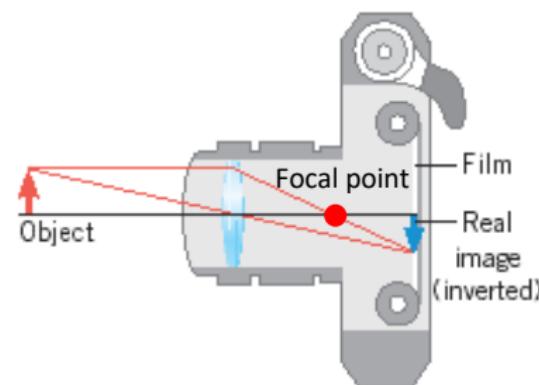


# Pinhole Camera

- Converging Lens
- ✓ All rays parallel to the optical axis converge at the focal point



A thin converging lens focuses light onto the film



Camera and converging lens

# Pinhole Camera

- Pinhole Camera Model
- ✓ The relationship between the image and object

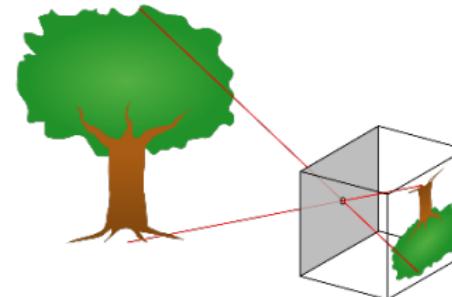


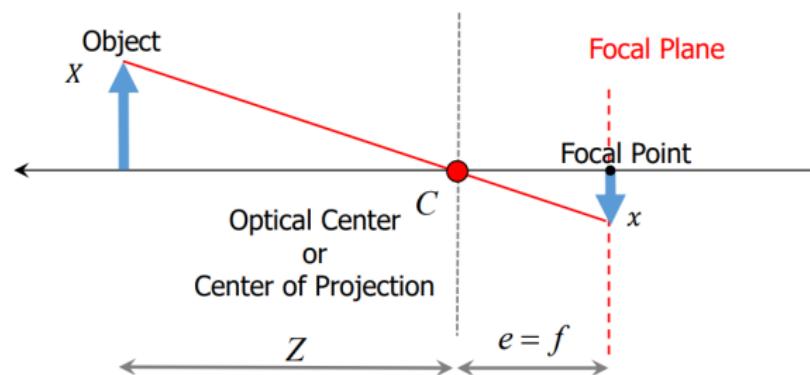
Image coordinates

$$-\frac{x}{X} = \frac{f}{Z} \Rightarrow x = -f \frac{X}{Z}$$

3D coordinates

Similar triangle

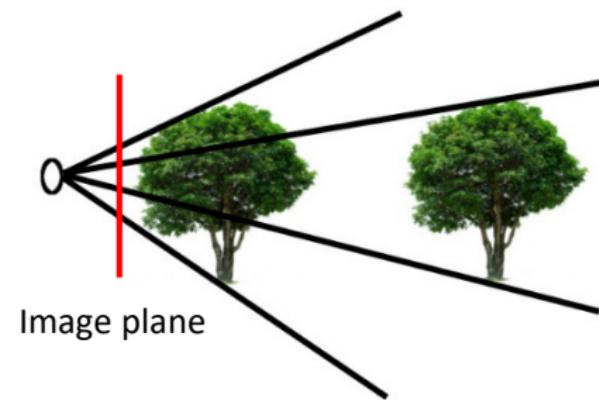
The equation 
$$-\frac{x}{X} = \frac{f}{Z} \Rightarrow x = -f \frac{X}{Z}$$
 is highlighted with a blue rounded rectangle.





# Pinhole Camera

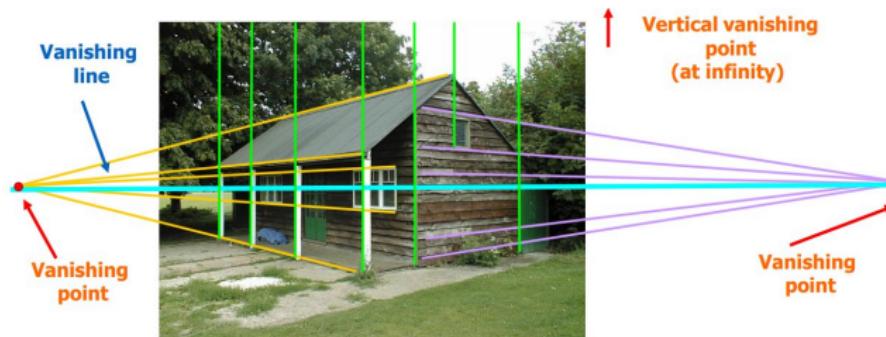
- Perspective Effects
- ✓ Far away objects appear smaller, with **size inversely proportional to distance.**



# Pinhole Camera

## ➤ Perspective Effects

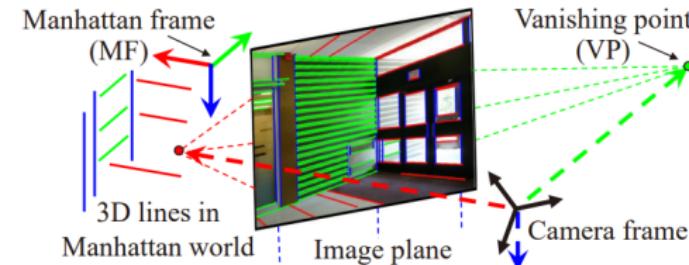
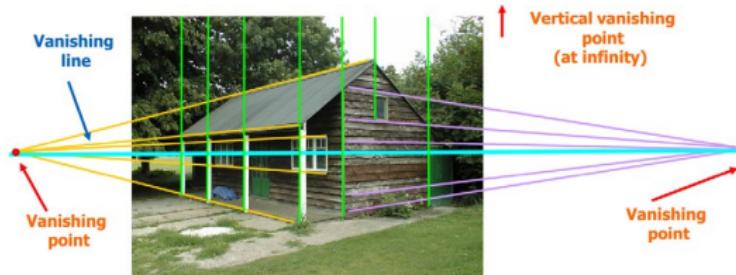
- ✓ Intersection of parallel lines in 2D
  - Parallel lines intersect at a “**vanishing point**” in the image
  - Vanishing points can fall both inside or outside the image
  - The connection between two horizontal vanishing points is the horizon



# Pinhole Camera

## ➤ Perspective Effects

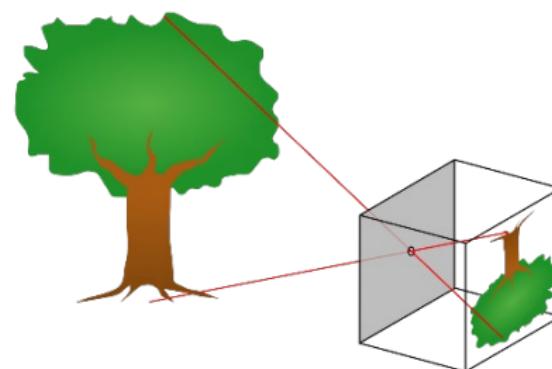
- ✓ Vanishing directions
  - A vanishing direction is defined by the connection between a vanishing point and camera center.
  - Vanishing direction is parallel to a 3D dominant direction.
  - Vanishing direction in 3D correspond to vanishing line in 2D.



# Pinhole Camera

- “Front” Image Plane

For convenience, the image plane is usually represented **in front of the lens**, such that the image preserves the same orientation (i.e. not flipped)



**Flipped image** in the pinhole camera model

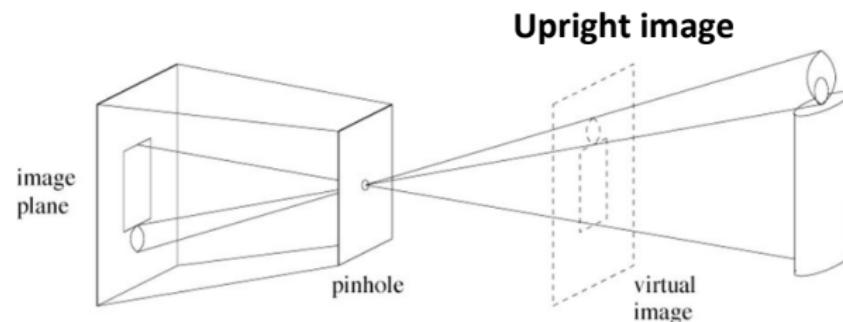


Illustration of virtual (upright) image plane

# Pinhole Camera

- “Front” Image Plane

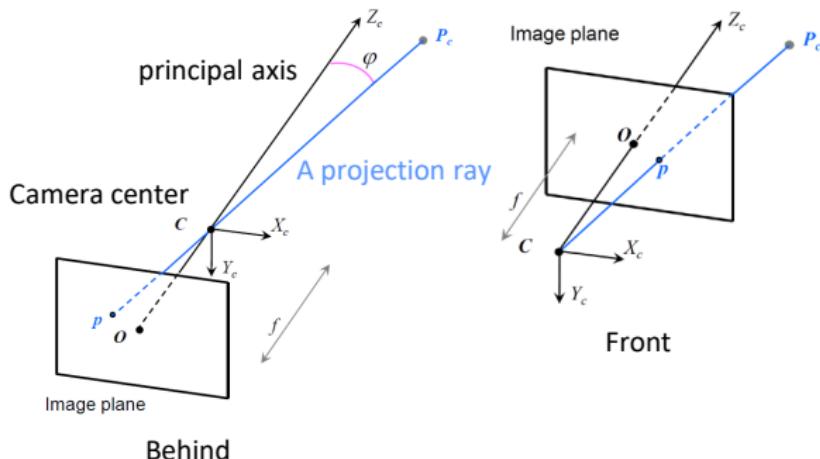
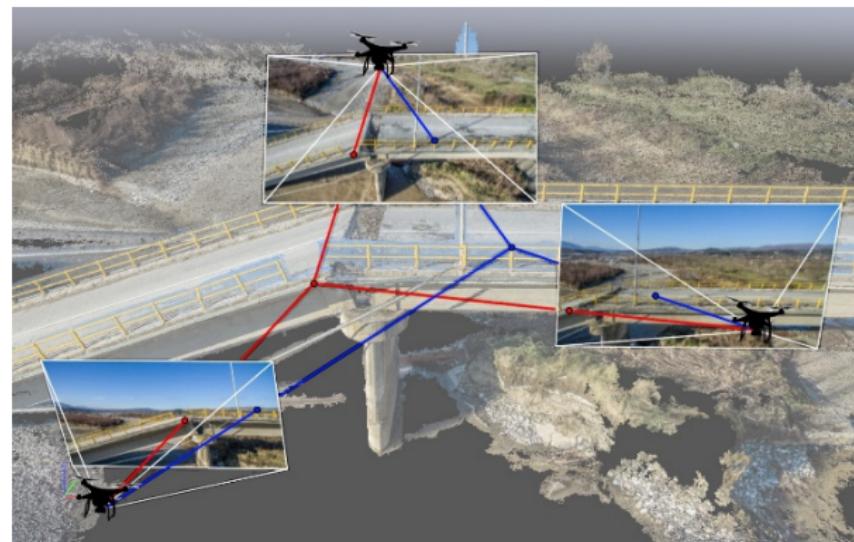


Illustration of image planes behind or in front of lens



Application to structure from motion  
(non-flipped images)

# Pinhole Camera

- Field of View (FOV)
- ✓ FOV is the **angular portion** of 3D scene seen by the camera

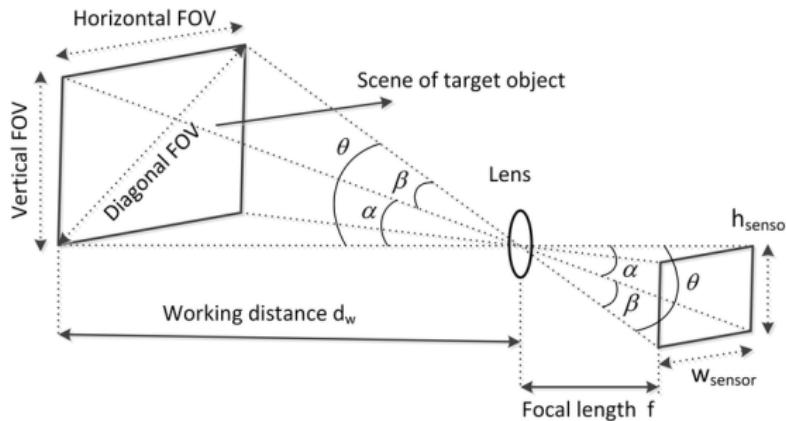
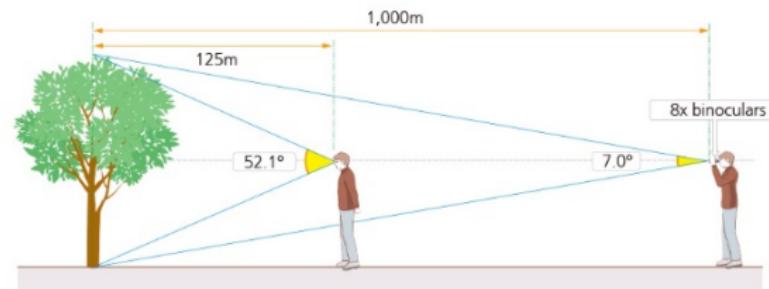
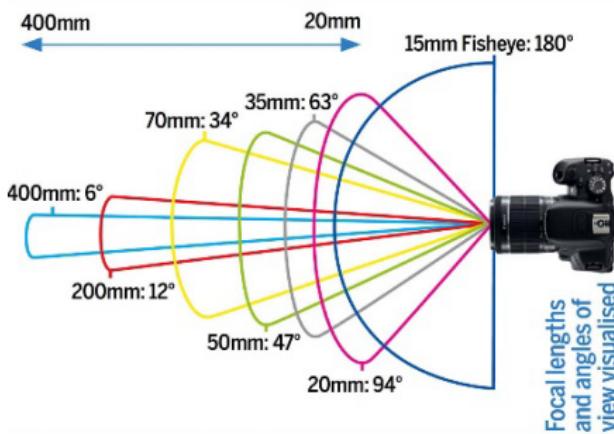


Illustration of FOV



# Pinhole Camera

- Field of View (FOV)
- ✓ FOV is inversely proportional to the focal length



Relationship between FOV and focal length

Short focal length & large FOV

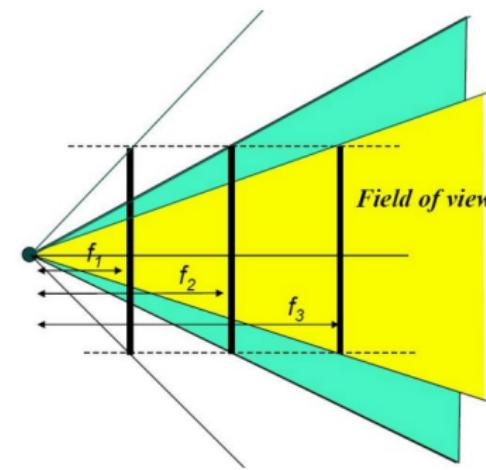
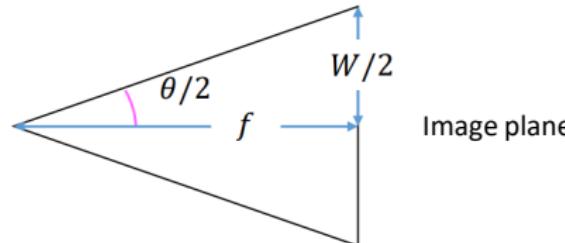


Long focal length & small FOV

# Pinhole Camera

- Field of View (FOV)
- ✓ Mathematical relation between **field of view**  $\theta$ , image width  $W$ , and focal length  $f$ :

$$\tan \frac{\theta}{2} = \frac{W}{2f} \rightarrow f = \frac{W}{2} \left[ \tan \frac{\theta}{2} \right]^{-1}$$



- ✓ We can also define the FOV angle by image height.

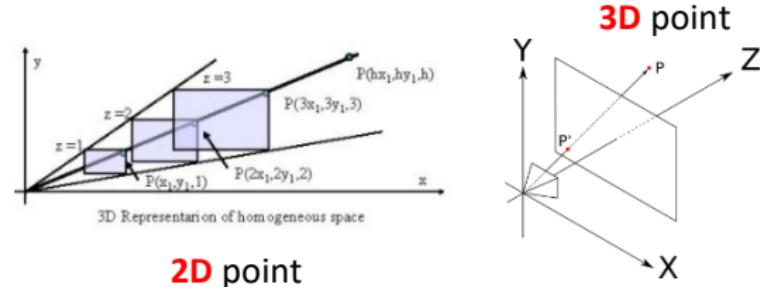
# Perspective Projection

## ➤ Recap on Homogeneous Coordinates

- ✓ For ease of computation/representation

- **3D Point**

Homogeneous	Cartesian
$(1, 2, 3) \Rightarrow \left(\frac{1}{3}, \frac{2}{3}\right)$	
$(2, 4, 6) \Rightarrow \left(\frac{2}{6}, \frac{4}{6}\right) = \left(\frac{1}{3}, \frac{2}{3}\right)$	
$(4, 8, 12) \Rightarrow \left(\frac{4}{12}, \frac{8}{12}\right) = \left(\frac{1}{3}, \frac{2}{3}\right)$	
$\vdots$	$\vdots$
$(1a, 2a, 3a) \Rightarrow \left(\frac{1a}{3a}, \frac{2a}{3a}\right) = \left(\frac{1}{3}, \frac{2}{3}\right)$	



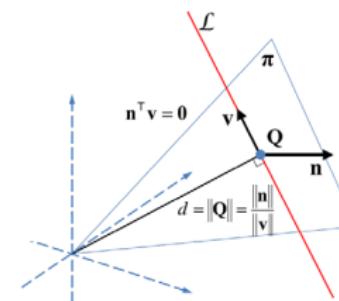
- **3D Line (Plucker Coordinates)**

$v$ : direction of 3D line (typically a unit vector)

$n$ : normal of projection plane

$$n = Q \times v$$

$$\|n\| = d * \|v\|$$



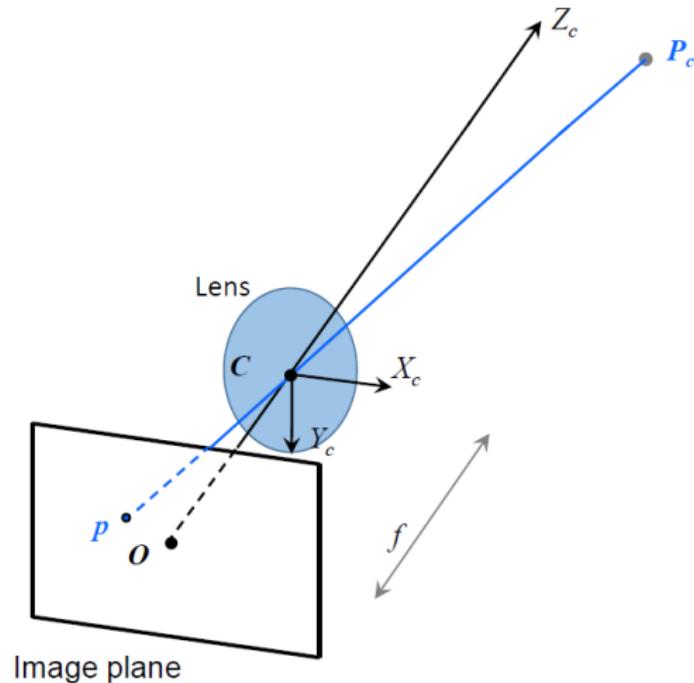


# Perspective Projection

## ➤ Basic Knowledge

- $C$ : optical center, i.e., center of the lens, i.e., center of projection
- $X_c, Y_c, Z_c$ : axes of the camera frame
- $Z_c$ : optical axis (principal axis)
- $O$ : principal point, i.e., intersection of optical axis and image plane

Note: principal point is not exactly the image center (will be introduced later)

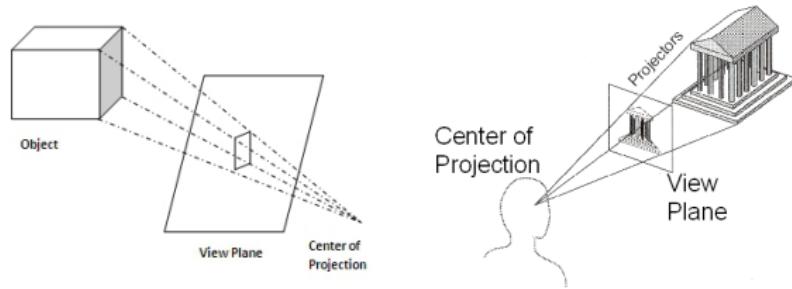


# Perspective Projection

## ➤ Perspective Projection vs. Parallel Projection

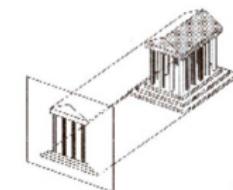
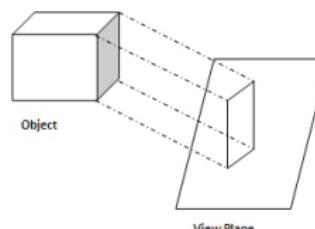
### ✓ Perspective Projection

- Size varies inversely with distance – looks realistic
- Parallel lines do not (in general) remain parallel



### ✓ Parallel Projection

- Good for exact measurements
- Parallel lines remain parallel
- Less realistic looking





# Perspective Projection

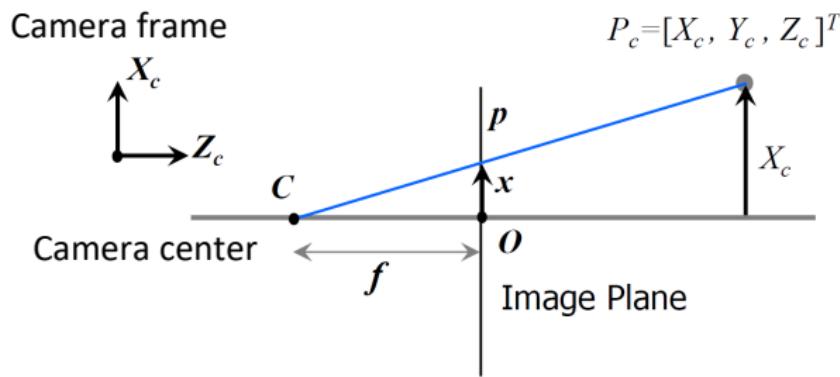
- From Camera Frame to Image Coordinates

A 3D point  $P_c = [X_c, Y_c, Z_c]^T$  in the camera frame is projected to  $p = (x, y)$  onto the image plane.

Based on similar triangles

Image coordinates

$$\frac{x}{f} = \frac{X_c}{Z_c} \Rightarrow x = \frac{f X_c}{Z_c}$$
$$\frac{y}{f} = \frac{Y_c}{Z_c} \Rightarrow y = \frac{f Y_c}{Z_c}$$



Side view of a scene



# Perspective Projection

## ➤ From Image Coordinates to Pixel Coordinates

- ✓ Let  $O = (u_0, v_0)$  be the pixel coordinates of the camera optical center
- ✓ Let  $k_u, k_v$  be the pixel conversion factors (conversion between mm and pixels)

$$x = \frac{fX_c}{Z_c}$$
$$y = \frac{fY_c}{Z_c}$$

Given image Coordinates  $(x, y)$ , we compute the Pixel Coordinates  $(u, v)$  as

$$u = u_0 + k_u x \rightarrow u = u_0 + \frac{k_u f X_c}{Z_c}$$

$$v = v_0 + k_v y \rightarrow v = v_0 + \frac{k_v f Y_c}{Z_c}$$

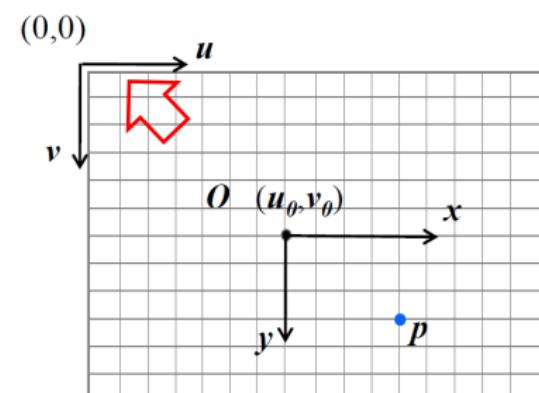


Image plane



# Perspective Projection

- From Image Coordinates to Pixel Coordinates

$$u = u_0 + k_u x \rightarrow u = u_0 + \frac{k_u f X_C}{Z_C}$$

Expressed by mm

$$v = v_0 + k_v y \rightarrow v = v_0 + \frac{k_v f Y_C}{Z_C}$$



$$u = u_0 + k_u x \rightarrow u = u_0 + \frac{\alpha_u X_C}{Z_C}$$

Focal lengths  
(expressed in pixels)

$$v = v_0 + k_v y \rightarrow v = v_0 + \frac{\alpha_v Y_C}{Z_C}$$

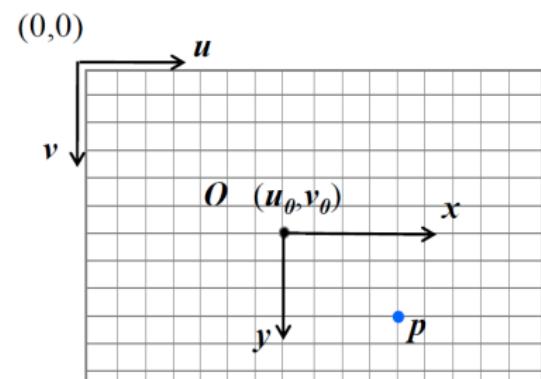


Image plane

# Perspective Projection

## ➤ Intrinsic/Calibration Matrix

Homogeneous coordinates

$$p = \begin{pmatrix} u \\ v \end{pmatrix} \quad \Rightarrow \quad \tilde{p} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Matrix form of perspective projection

- Focal length  $a_u$   $a_v$
- Principal points  $u_0$   $v_0$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Intrinsic/Calibration matrix

Not equal due to conversion factor

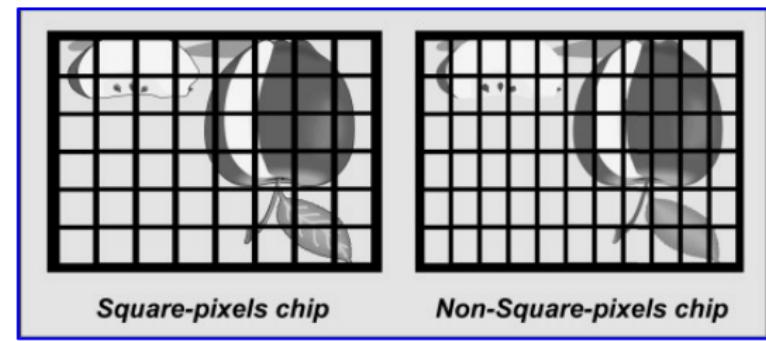
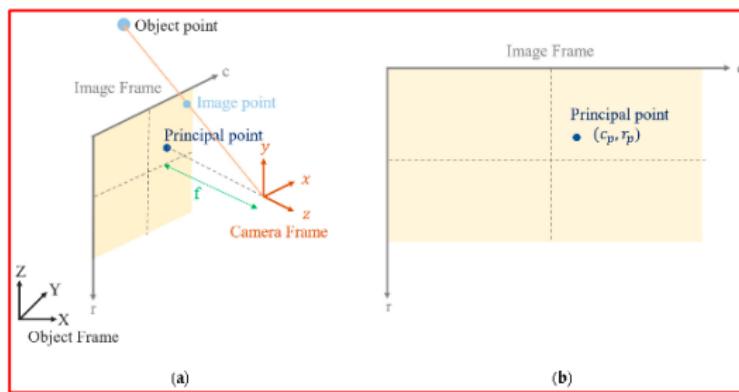
$$u = u_0 + k_u x \rightarrow u = u_0 + \frac{\alpha_u X_c}{Z_c}$$
$$v = v_0 + k_v y \rightarrow v = v_0 + \frac{\alpha_v Y_c}{Z_c}$$

# Perspective Projection

## ➤ Intrinsic/Calibration Matrix



- ✓ In the past it was common to assume a **skew factor** in the pixel manufacturing process.
- ✓ However, the camera manufacturing process today is so good that we can safely assume **skew factor = 0** and  $\alpha_u = \alpha_v$  (i.e., square pixels).



# Perspective Projection

## ➤ An Example of Intrinsic Parameters

Most widely-used SLAM datasets, e.g., TUM RGBD dataset provide intrinsic parameters calibrated beforehand (calibration will be introduced later).

### CALIBRATION OF THE COLOR CAMERA

We computed the intrinsic parameters of the RGB camera from the rgbd\_dataset\_freiburg1/2\_rgb\_calibration.bag.

Camera	fx	cx	cy	d0	d1	d2	d3	d4	
(ROS default)	525.0	525.0	319.5	239.5	0.0	0.0	0.0	0.0	0.0
 Freiburg 1 RGB	517.3	516.5	318.6	255.3	0.2624	-0.9531	-0.0054	0.0026	1.1633
 Freiburg 2 RGB	520.9	521.0	325.1	249.7	0.2312	-0.7849	-0.0033	-0.0001	0.9172
 Freiburg 3 RGB	535.4	539.2	320.1	247.6	0	0	0	0	0

Note that both the color and IR images of the Freiburg 3 sequences have already been undistorted, therefore the distortion parameters are all zero. The original distortion values can be found in the tgz file.

**Note:** We recommend to use the ROS default parameter set (i.e., without undistortion), as undistortion of the pre-registered depth images is not trivial.

Image resolution: **640\*480 pixels**

[https://cvg.cit.tum.de/data/datasets/rgbd-dataset/file\\_formats](https://cvg.cit.tum.de/data/datasets/rgbd-dataset/file_formats)

# Perspective Projection

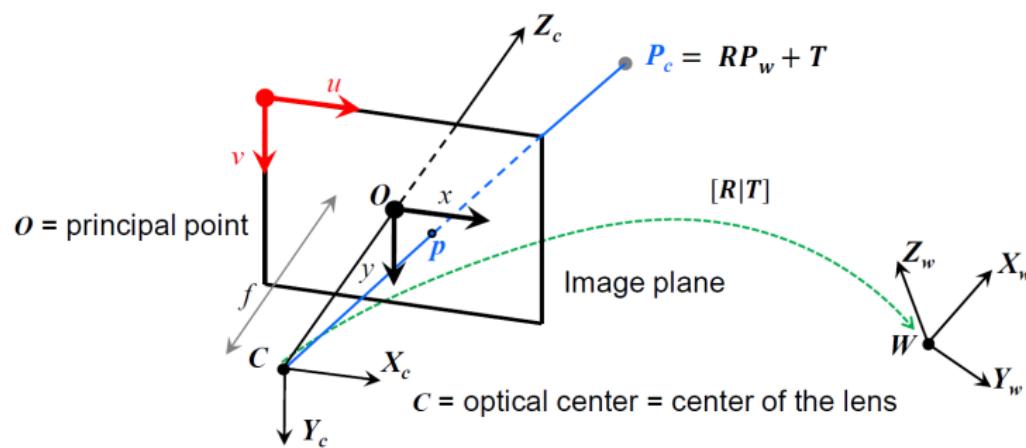
- From World Frame to Pixel Coordinates

## Coordinate systems

- Camera frame
- Image coordinates
- Pixel coordinates
- **World frame**

## Camera parameters

- Intrinsic parameters
- **Extrinsic parameters**



# Perspective Projection

## ➤ Projection Matrix

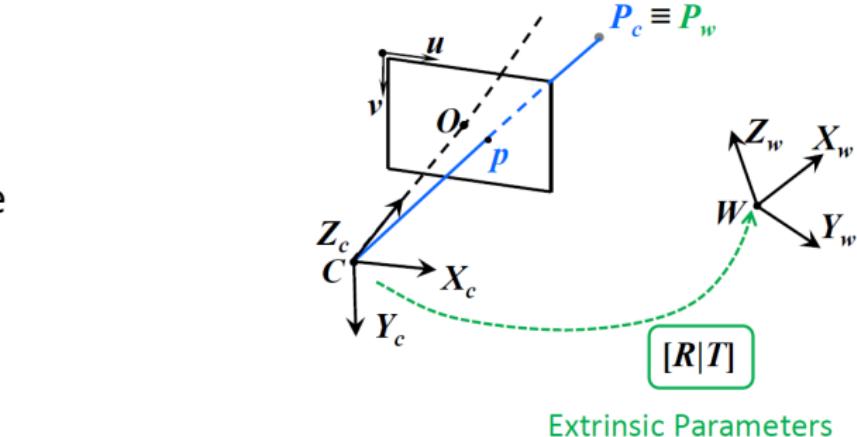
From the world frame to the camera frame

$$\mathbf{X}_c = \mathbf{R}\mathbf{X}_w + \mathbf{t}$$

Rigid transformation (extrinsic parameters)

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

Matrix form



Extrinsic Parameters

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

More compact form

# Perspective Projection

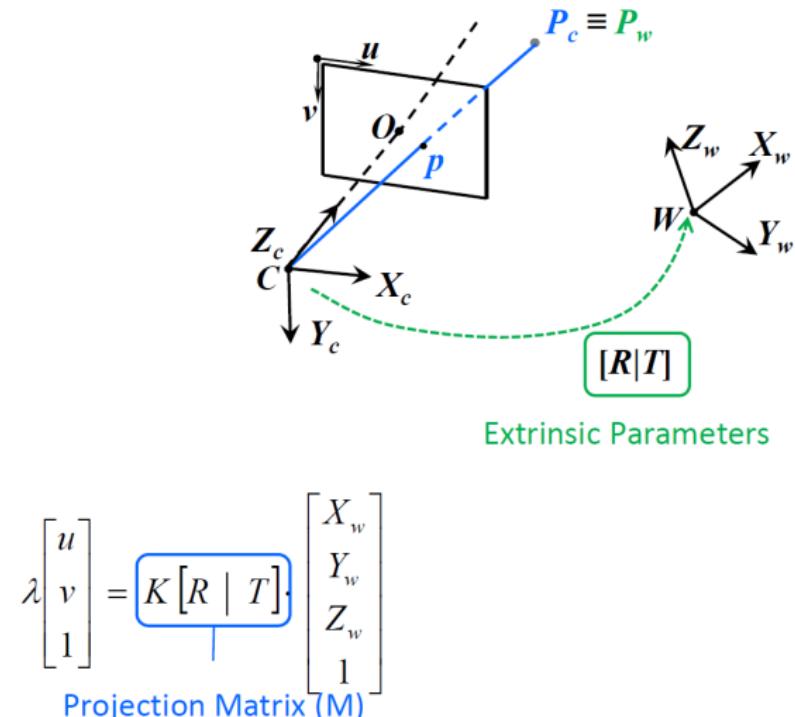
- Projection Matrix

- ✓ Rigid transformation

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \left[ \begin{array}{c|c} R & T \end{array} \right] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

- ✓ Perspective projection (camera frame)

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$



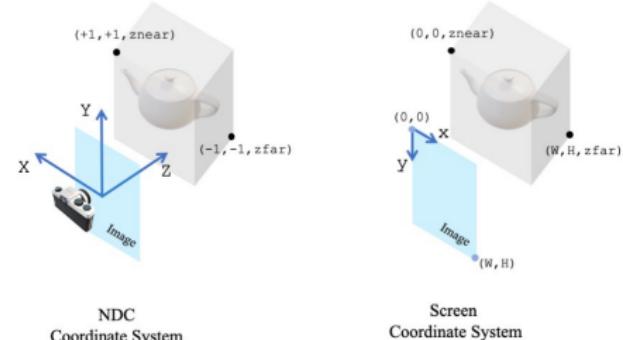
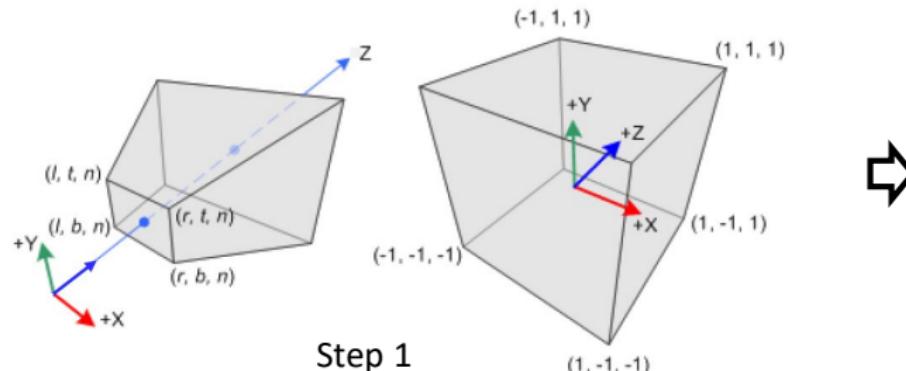


# Perspective Projection

## ➤ Computer Vision vs Computer Graphics

### ✓ Pipeline

- Transforms the view volume, i.e., the pyramidal frustum to the canonical view volume, i.e., normalized device coordinates (NDC).
- Linearly expand the XOY plane of NDC to screen/image plane.



Step 2

# Perspective Projection

➤ Computer Vision vs Computer Graphics

✓ Step 1: Convert perspective frustum to NDC space

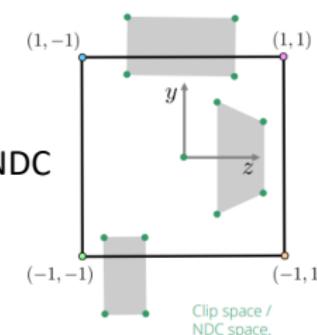
$$\begin{bmatrix} x_{clip} \\ y_{clip} \\ z_{clip} \\ w_{clip} \end{bmatrix} = \boxed{P} \begin{bmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ w_{eye} \end{bmatrix} \quad \text{where} \quad P = \begin{bmatrix} \frac{2n}{r-l} & 0 & -\frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & -\frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Perspective matrix      Homogeneous coordinates of a 3D point

NDC coordinates

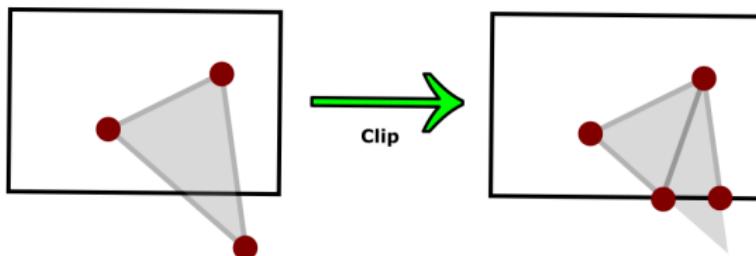
$$\boxed{\begin{bmatrix} x_{ndc} \\ y_{ndc} \\ z_{ndc} \end{bmatrix}} = \begin{bmatrix} x_{clip} / w_{clip} \\ y_{clip} / w_{clip} \\ z_{clip} / w_{clip} \end{bmatrix} \rightarrow$$

XOY plane of NDC

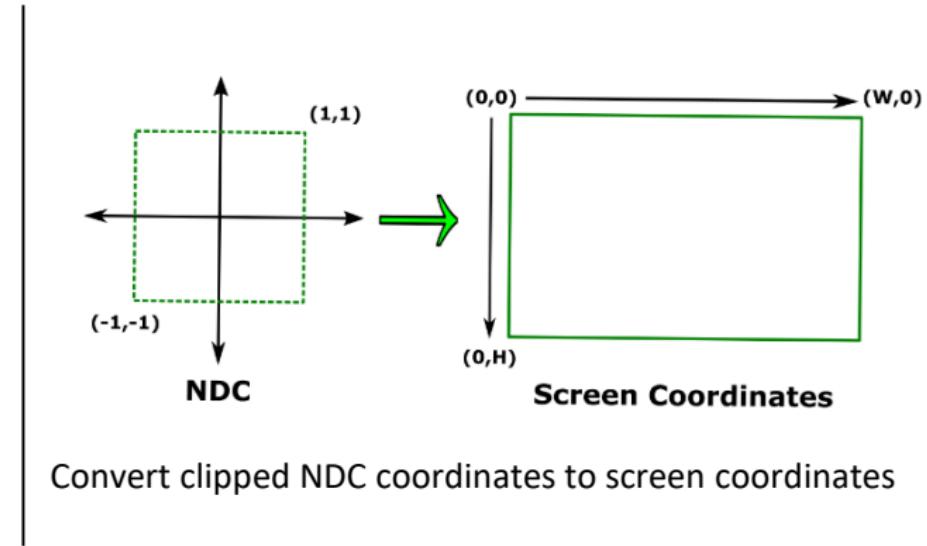


# Perspective Projection

- Computer Vision vs Computer Graphics
- ✓ Step 2: From NDC to screen space



Clip the content outside the NDC coordinates



Convert clipped NDC coordinates to screen coordinates



# Perspective Projection

## ➤ Line Projection

- ✓ Two-step computation method
  - Coordinates of 2D endpoints (homogeneous)

$$\mathbf{A}' = \mathbf{KA}$$

$$\mathbf{B}' = \mathbf{KB}$$



Intrinsic matrix

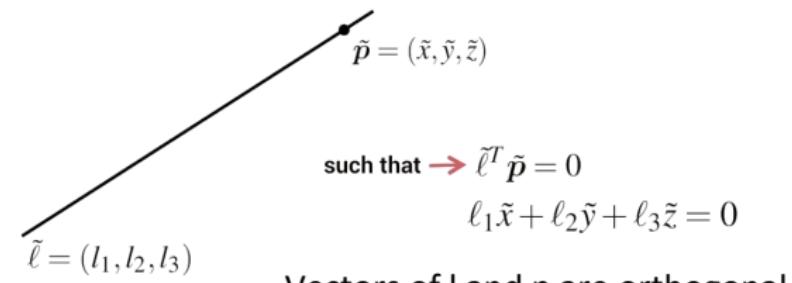
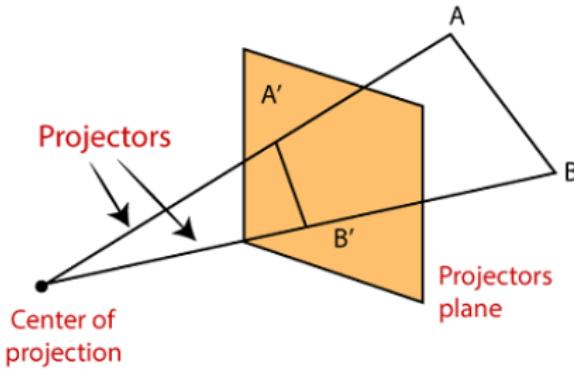
- Coordinates of 2D line (homogeneous)

$$\mathbf{l} = \mathbf{A}' \times \mathbf{B}'$$



3D vector

## Perspective Projection



Vectors of  $\mathbf{l}$  and  $\mathbf{p}$  are orthogonal



# Perspective Projection

## ➤ Line Projection

- ✓ One-step computation method
- Coordinates of line (homogeneous)

$$\mathbf{l} = \mathcal{K}\mathbf{n}$$

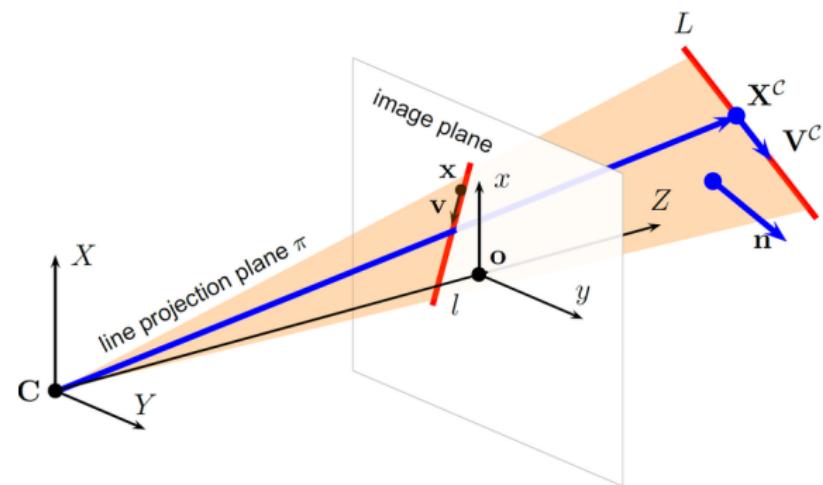
↑

3D vector

$$\mathcal{K} = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ -f_y x_0 & -f_x y_0 & f_x f_y \end{bmatrix}$$

Intrinsic matrix for line projection

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$



# Perspective Projection

- Relationship between Points, Lines, and Planes

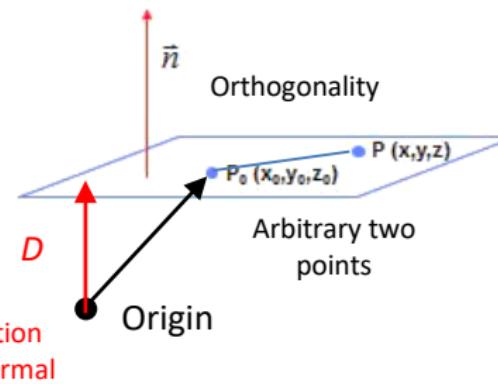
- ✓ Some important conclusions

- Homogenous coordinates of a plane

A four-dimensional vector  $(A, B, C, D)$

Normal of  
plane

Projection  
onto normal



- A point  $P$  in homogenous coordinates  $(X, Y, Z, 1)$  lies on a plane

$$P^T n = 0$$

$$\vec{n} = [A, B, C] \quad \text{Unit vector}$$

$$\vec{n} \cdot \vec{P_0P} = 0$$

$$[A, B, C] \cdot [x - x_0, y - y_0, z - z_0] = 0$$

$$A(x - x_0) + B(y - y_0) + C(z - z_0) = 0$$

$$Ax + By + Cz = Ax_0 + By_0 + Cz_0$$

$$Ax + By + Cz = D$$

Dot product

# Perspective Projection

- Relationship between Points, Lines, and Planes
- ✓ Some important conclusions

- Projection plane computed by image line

$$\underline{\pi}_L = P^\top \underline{l}_L \in \mathbb{R}^4$$

Projection matrix (3\*4)

- Intersection between a 3D line and a 3D plane

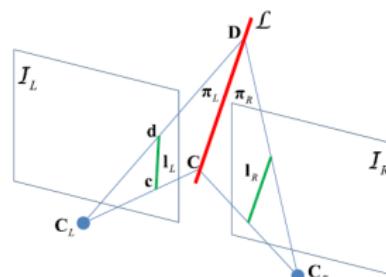
$$\underline{D} = L\underline{\pi}$$

Homogeneous coordinates

$$L = \begin{bmatrix} [\mathbf{n}]_\times \mathbf{v} \\ -\mathbf{v}^\top & 0 \end{bmatrix} \Leftrightarrow \mathcal{L} = (\mathbf{n}^\top, \mathbf{v}^\top)^\top$$

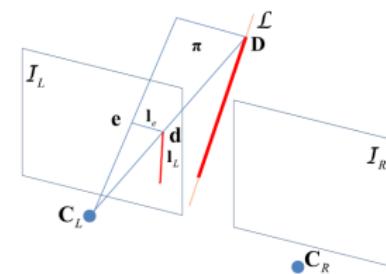
Plucker matrix

Plucker coordinates



$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K [R \mid T] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Projection Matrix (M)

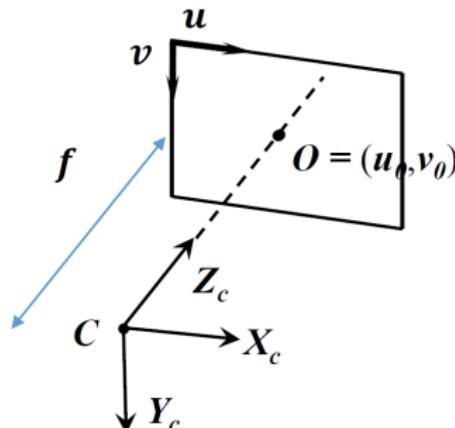




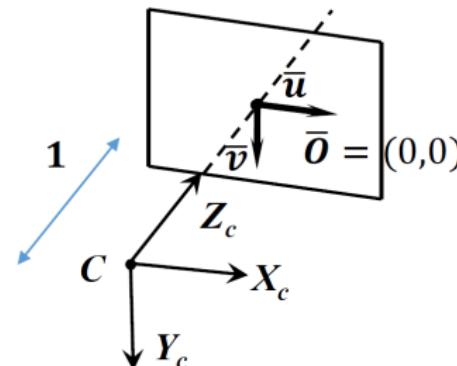
# Perspective Projection

- Normalized Image

A virtual image plane with **focal length equal to 1 unit** and **origin of the pixel coordinates at the principal point**.



Ordinary image plane



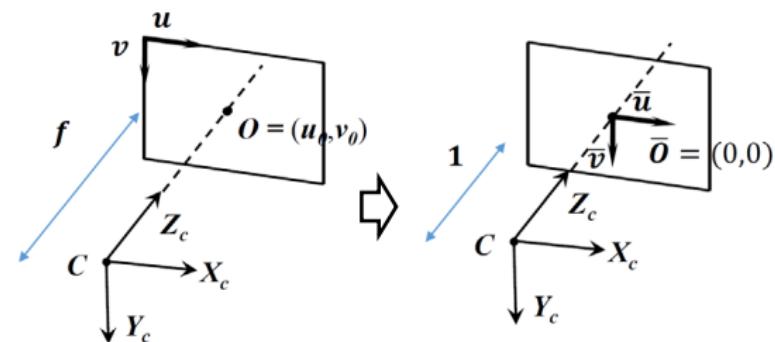
Normalized image plane



# Perspective Projection

- Normalized Image
- ✓ Computation of normalized coordinates

$$\begin{bmatrix} \bar{u} \\ \bar{v} \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha} & 0 & -\frac{u_0}{\alpha} \\ 0 & \frac{1}{\alpha} & -\frac{v_0}{\alpha} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{u-u_0}{\alpha} \\ \frac{v-v_0}{\alpha} \\ 1 \end{bmatrix}$$



# Perspective Projection

## ➤ Normalized Image

Multiply both terms of the perspective projection equation in camera frame coordinates by  $K^{-1}$

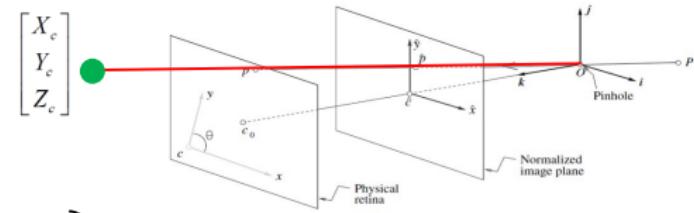
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \Rightarrow \lambda K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \cancel{K} \cdot \cancel{K^{-1}} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \Rightarrow \lambda K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Basic projection

Normalized image coordinates

$$\begin{bmatrix} \bar{u} \\ \bar{v} \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha} & 0 & -\frac{u_0}{\alpha} \\ 0 & \frac{1}{\alpha} & -\frac{v_0}{\alpha} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{u-u_0}{\alpha} \\ \frac{v-v_0}{\alpha} \\ 1 \end{bmatrix}$$

Geometric meaning?



Camera frame

$$\lambda \begin{bmatrix} \bar{u} \\ \bar{v} \\ 1 \end{bmatrix} = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

Collinearity between 3D vectors in camera frame



# Perspective Projection

- Geometric constraints of points
- ✓ Parallelism of ray directions

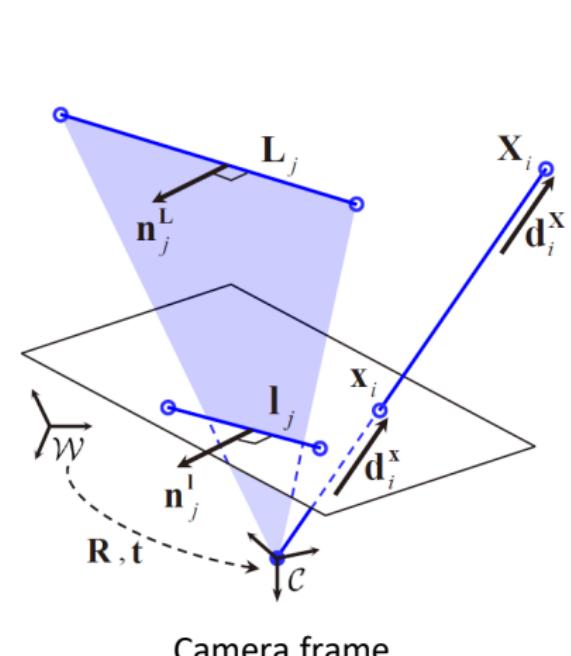
$$\mathbf{d}_i^x \propto \mathbf{d}_i^X \Rightarrow \mathbf{K}^{-1}\mathbf{x}_i \propto [\mathbf{R} \quad \mathbf{t}] \mathbf{X}_i$$

3D vector      3D vector

“ $\propto$ ” represents equality regardless of scale, i.e., two vectors are parallel, which leads to the cross product of 0.

- A  $3 \times 3$  skew-symmetric matrix has the rank of 2, so each 3D-2D point correspondence provide two constraints.

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_{\times} \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$



# Perspective Projection

- Geometric constraints of lines
- ✓ Parallelism of normals of projection plane

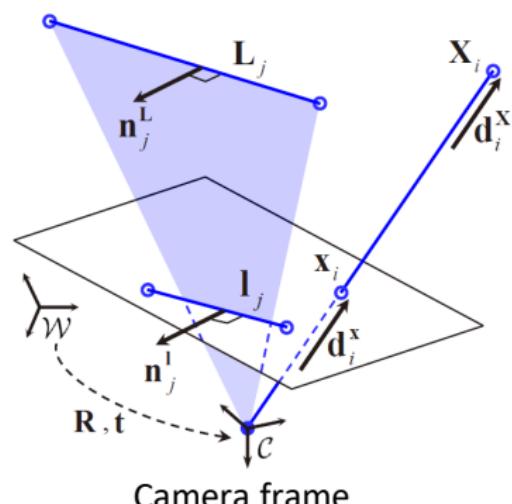
$$\mathbf{n}_j^1 \propto \mathbf{n}_j^L \Rightarrow \tilde{\mathbf{K}}^{-1} \mathbf{l}_j \propto [\mathbf{R} \quad [\mathbf{t}]_{\times} \mathbf{R}] \mathbf{L}_j$$

3D vector                    3D vector

$\mathbf{L}_j$       First 3 elements of Plucker coordinates

$$\mathcal{K} = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ -f_y x_0 & -f_x y_0 & f_x f_y \end{bmatrix}$$

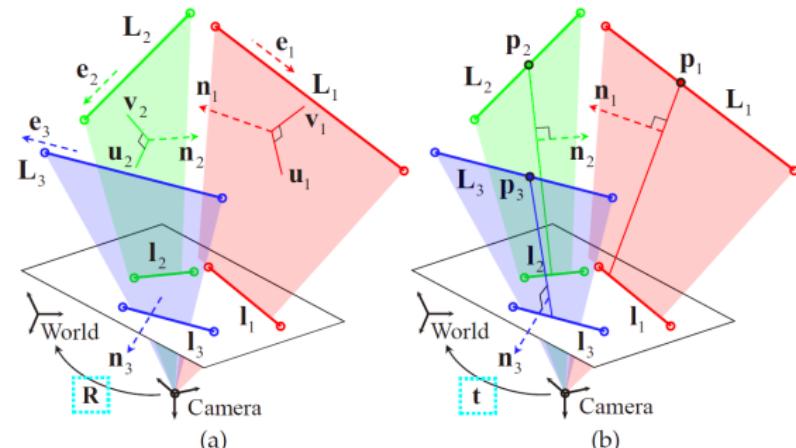
$$\begin{bmatrix} \mathbf{n}_j \\ \mathbf{v}_j \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{ji} & [\mathbf{t}_{ji}]_{\times} \mathbf{R}_{ji} \\ \mathbf{0} & \mathbf{R}_{ji} \end{bmatrix} \begin{bmatrix} \mathbf{n}_i \\ \mathbf{v}_i \end{bmatrix}$$



- Similarly,  $3 \times 3$  skew-symmetric matrix has the rank of 2, so each 3D-2D line correspondence provide two constraints.
- How many points and/or lines should we use to compute 6-DOF camera pose?

# Perspective Projection

- Geometric constraints of lines
- ✓ An alternative expression of line constraint
- 3D line direction is orthogonal to the normal of projection plane. (one constraint)
- The direction defined by a 3D point lying on the 3D line and the origin is orthogonal to the normal of projection plane. (one constraint)



3D-2D line correspondences  $\{(\mathbf{L}_k, \mathbf{l}_k)\}_{k=1}^3$

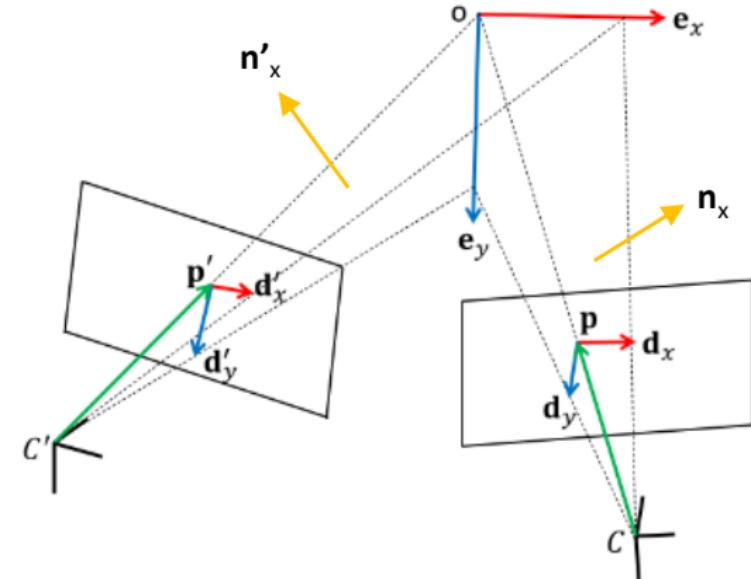
# Perspective Projection

- Normalized Image
- ✓ Applications to geometric constraints of lines
- Point and Line (Ray-Point-Ray Structure)

$$\begin{cases} \mathbf{n}_x = \mathbf{p} \times \mathbf{d}_x \\ \mathbf{n}_y = \mathbf{p} \times \mathbf{d}_y \end{cases} \quad \begin{cases} \mathbf{n}'_x = \mathbf{p}' \times \mathbf{d}'_x \\ \mathbf{n}'_y = \mathbf{p}' \times \mathbf{d}'_y \end{cases}$$



$$\begin{cases} \mathbf{e}_x \propto \mathbf{n}'_x \times \mathbf{R}\mathbf{n}_x \\ \mathbf{e}_y \propto \mathbf{n}'_y \times \mathbf{R}\mathbf{n}_y \end{cases}$$



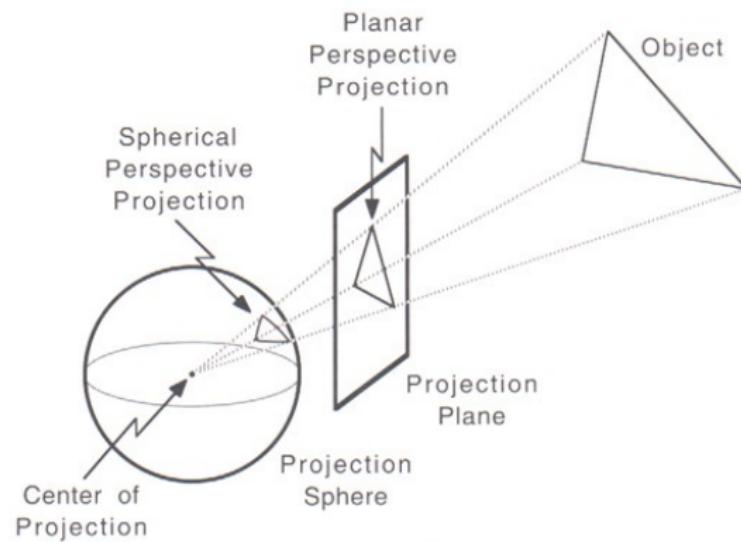
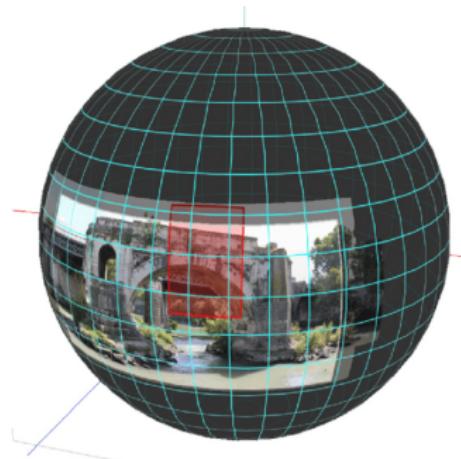
Two-view configuration of ray-point-ray structure

$$\mathbf{e}_x^\top \mathbf{e}_y = \cos \alpha \cdot \|\mathbf{e}_x\| \cdot \|\mathbf{e}_y\|$$

Angle between two lines

# Perspective Projection

- Spherical Projection
- ✓ Planar projection vs. spherical projection



Spherical projection has a larger FOV than planar projection

# Perspective Projection

- Spherical Projection
- ✓ Obtaining a panorama with a 360 degree field of view



Omnicamera



Equirectangular panorama in a spherical projection.

# Perspective Projection

- Spherical Projection
- ✓ Pipeline of spherical image generation
- Map 3D point (X,Y,Z) onto sphere

$$(\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2+Y^2+Z^2}}(X, Y, Z)$$

- Convert to spherical coordinates

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = \tan^{-1} \left( \frac{y}{x} \right)$$

$$\phi = \cos^{-1} \left( \frac{z}{r} \right),$$

$$x = r \cos \theta \sin \phi$$

$$y = r \sin \theta \sin \phi$$

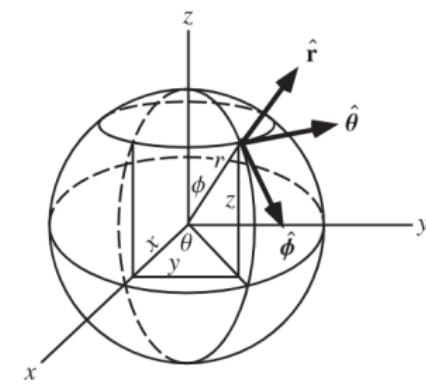
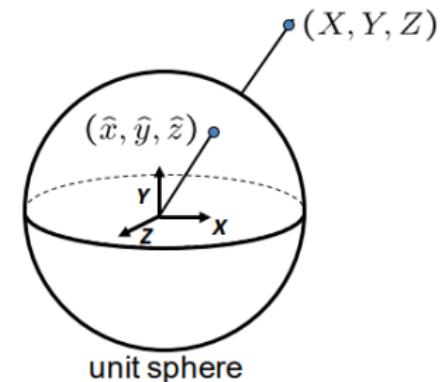
$$z = r \cos \phi.$$

Azimuth

$$\theta \in [0, 2\pi)$$

Polar angle

$$\phi \in [0, \pi]$$

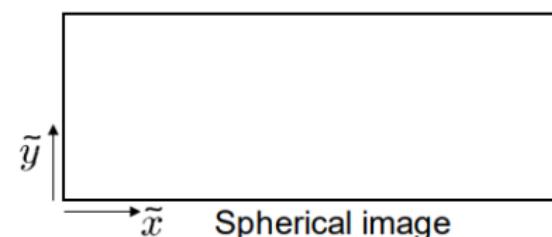
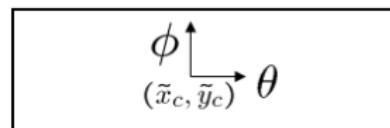


# Perspective Projection

- Spherical Projection
- ✓ Pipeline of spherical image generation
  - $s$  defines size of the final image  
(often convenient to set  $s = \text{camera focal length}$ )

$$(\tilde{x}, \tilde{y}) = (s\theta, s\phi) + (\tilde{x}_c, \tilde{y}_c) \quad \text{Displacement of origin}$$

Linear mapping

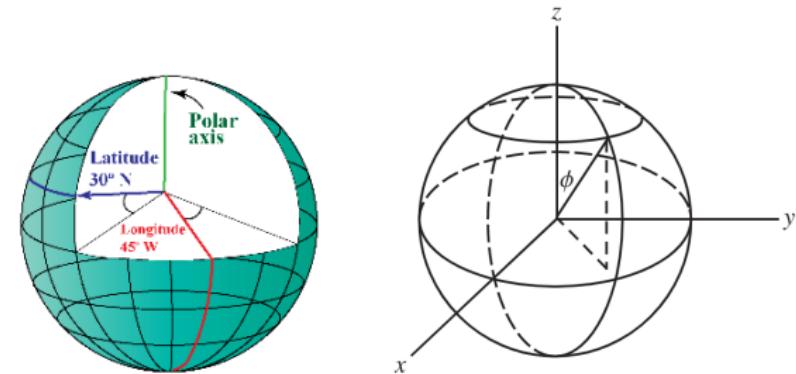


unwrapped sphere

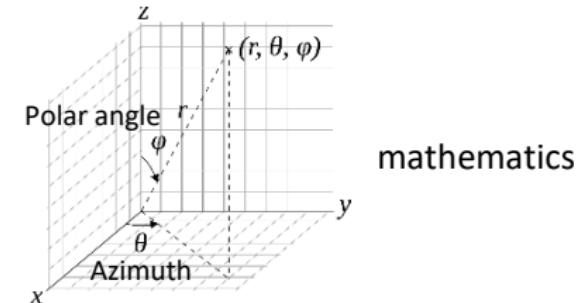
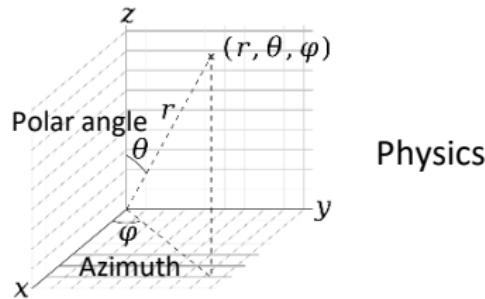
Spherical image

# Perspective Projection

- Spherical Projection
- ✓ Difference between latitude and polar angle

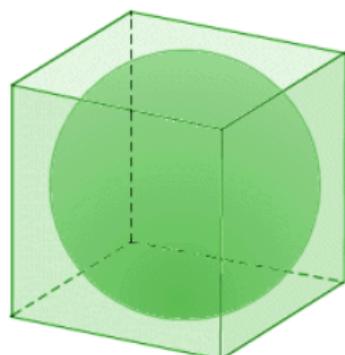


- ✓ Difference between mathematical and physical representation

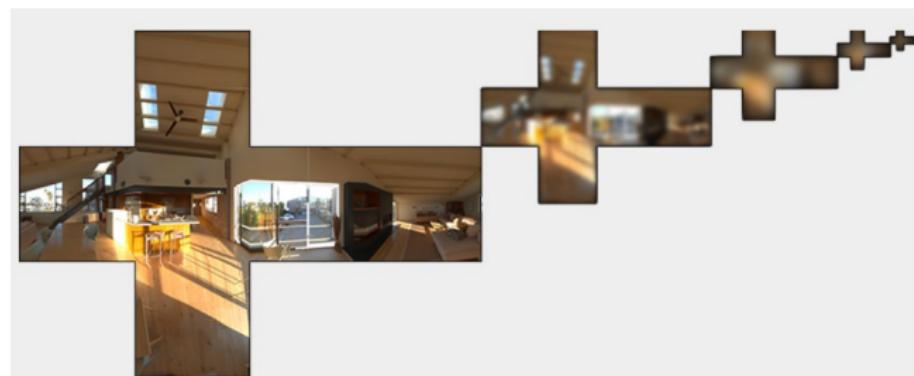


# Perspective Projection

- Spherical Projection
- ✓ Cube-based representation



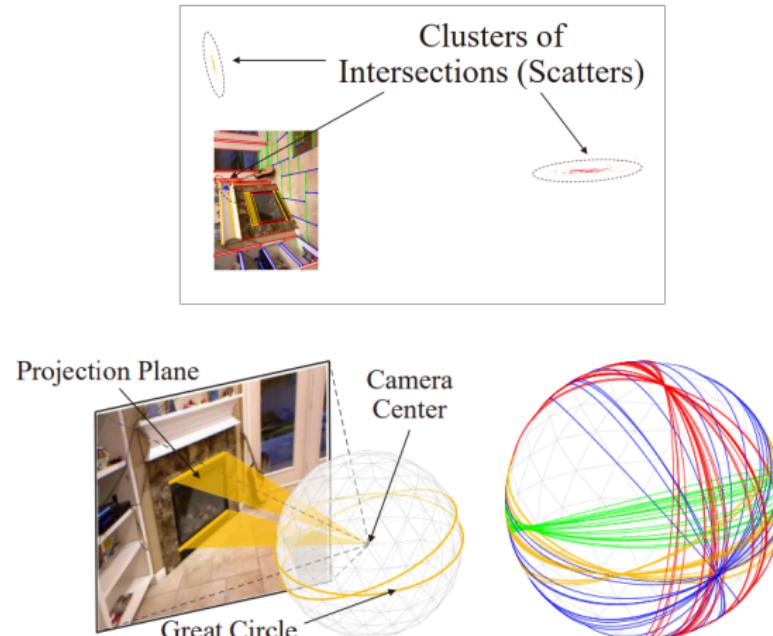
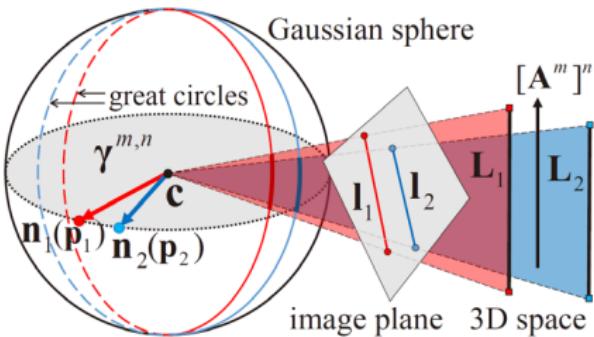
Inscribed sphere



Cross-shaped expansion (more commonly used in computer graphics)

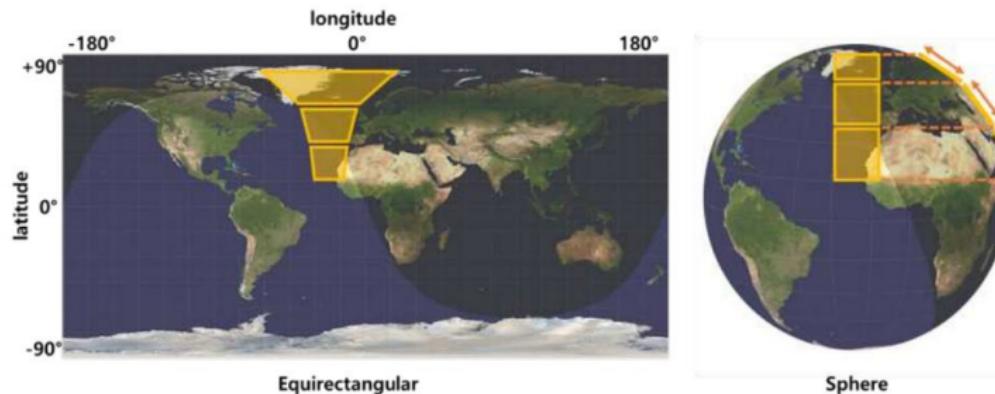
# Perspective Projection

- Spherical Projection
- ✓ Lines and vanishing points



# Perspective Projection

- Other Expressions of Sphere
- ✓ Spatial distortion due to equi-rectangular representation



- Yellow squares on both sides represent the same surface areas on the sphere.
- The area of Antarctica seems large.



Actual area comparison between Antarctica and Australia plus New Zealand

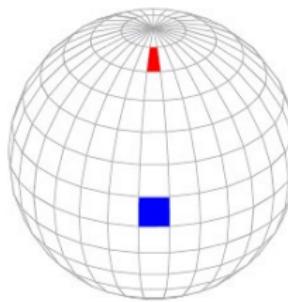


# Perspective Projection

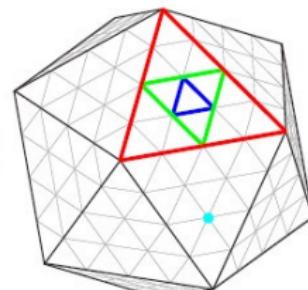
## ➤ Other Expressions of Sphere

### ✓ Icosahedral representation

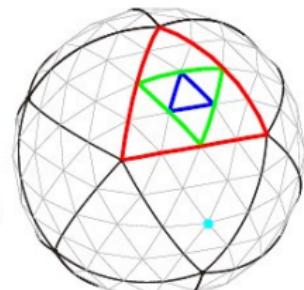
We extrude all the vertices of icosahedron sub-faces to the unit sphere, obtaining the icosahedral spherical representation



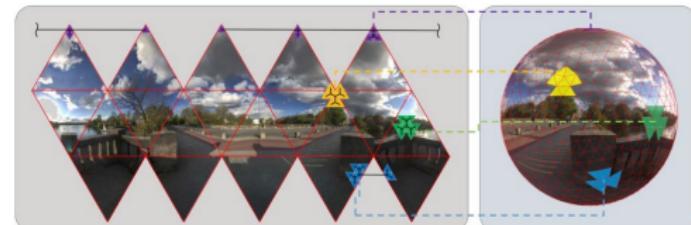
(a)  
Equi-angular  
discretization



(b)  
Icosahedron



(c)  
icosahedral spherical  
representation



Expansion of icosahedral sphere

# Summary

- Recap on Image Processing
- Pinhole Camera
- Perspective Projection



Thank you for your listening!  
If you have any questions, please come to me :-)