Camera frame ↔ image plane transformation

Camera frame ⇒ image plane transformation

(*projection / normalization*) : $P = [X_C, Y_C, Z_C]^T$ [metric units] ⇒ $p = [u, v]^T$ [pixels]

1. Transform $P = [X_C, Y_C, Z_C]^T \Rightarrow p = [x, y, -f]^T$

Fundamental equations of the *perspective camera model* normalized with *1/Z*:

$$\begin{bmatrix} x \\ y \end{bmatrix} = f \begin{bmatrix} X_c / Z_c \\ Y_c / Z_c \end{bmatrix} = f \begin{bmatrix} x_N \\ y_N \end{bmatrix}$$

f – focal distance [metric units]

2. Transform **p** $[x, y]^T$ [metric units] ⇒ image coordinates $[u, v]^T$ [pixels]

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} D_u \cdot x \\ D_v \cdot y \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$

Du, Dv - coefficients needed to transform metric units to pixels: Du = 1 / dpx; Dv = 1 / dpy

1 + 2 ⇒ projection equation:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \cdot \begin{bmatrix} x_N \\ y_N \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} f_X & 0 & u_0 \\ 0 & f_Y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

A – is the camera matrix:

f_X – is the focal distance expressed in units of horizontal pixels:

$$f_X = f \cdot D_u = \frac{f}{dpx}$$

f_Y – is the focal distance expressed in units of vertical pixels:

$$f_Y = f \cdot D_v = \frac{f}{dpy}$$

Technical University of Cluj Napoca
Computer Science Department

Camera frame $\leftrightarrow$ image plane transformation

Image plane transformation $\Rightarrow$ camera frame

(*reconstruction*) : p = [u, v]^T [pixels] $\Rightarrow$ P = [X_C, Y_C, Z_C]^T [metric units]

$$\begin{bmatrix} x_N \\ y_N \\ 1 \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Notes:

With one camera we cannot measure depth (Z). We can determine only the projection equation / normalized coordinates:

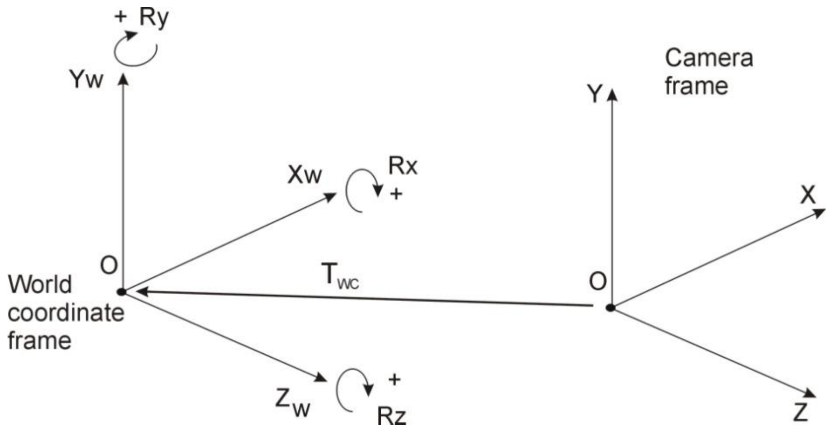
$$\begin{bmatrix} x_N \\ y_N \end{bmatrix} = \begin{bmatrix} X_C / Z_C \\ Yc / Z_C \end{bmatrix}$$

To measure the depth (Z) a stereo system (2 cameras) is needed

Camera frame ↔ world reference frame transformation

Direct mapping (world ⇒ camera)

XX$_W$ = [X_W , Y_W , Z_W]^T **(**world coordinate system - WRF) ⇒ **XX**$_C$ = [X_C , Y_C , Z_C]^T (camera coordinate system – CRF)



$$\mathbf{XX}_C = \mathbf{R}_{WC} \cdot \mathbf{XX}_W + \mathbf{T}_{WC}$$

where:

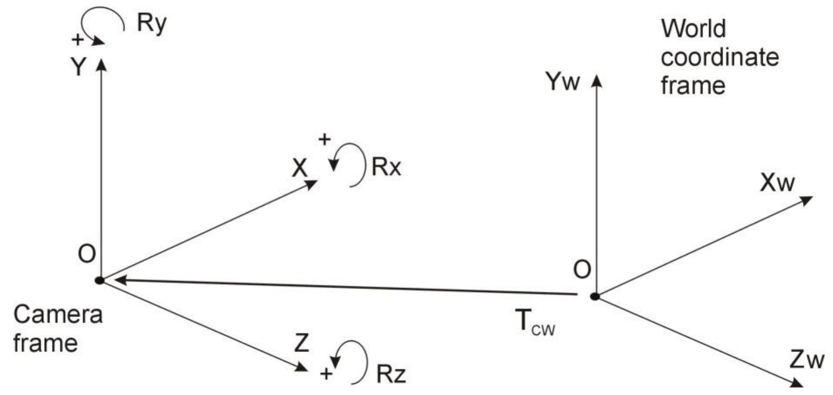
$\mathbf{T}_{WC}$ = [Tx, Ty, Tz]^T – world to camera translation vector;
$\mathbf{R}_{WC}$ – world to camera rotation matrix:

Camera frame ↔ world reference frame transformation

Inverse mapping (camera ⇒ world)

$\mathbf{XX}_C = [X_C, Y_C, Z_C]^T$ (camera coordinate system – CRF) ⇒ $\mathbf{XX}_W = [X_W, Y_W, Z_W]^T$ (world coordinate system - WRF)

$$\mathbf{XX}_W = \mathbf{R}_{WC}^{-1} \cdot (\mathbf{XX}_C - \mathbf{T}_{WC})$$



Rotation matrix is orthogonal [Trucco1998]:

$$\mathbf{R} \cdot \mathbf{R}^T = \mathbf{R}^T \cdot \mathbf{R} = 1 \Rightarrow \mathbf{R}^T = \mathbf{R}^{-1}$$

$$\mathbf{XX}_W = \mathbf{R}_{WC}^{T} \cdot (\mathbf{XX}_C - \mathbf{T}_{WC}) = \mathbf{R}_{CW} \cdot (\mathbf{XX}_C + \mathbf{T}_{CW})$$

where:

$\mathbf{T}_{CW} = [T_X\ T_Y\ T_Z]^T$ – camera to world translation vector $\quad T_{CW} = -T_{WC}$

$\mathbf{R}_{CW}$ – camera to world translation vector $\quad R_{CW} = R_{WC}^{T}$

Technical University of Cluj Napoca
Computer Science Department

Rotation Matrix

World-to-camera

$$\mathbf{R}_{WC} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{n}^{XW} & \mathbf{n}^{YW} & \mathbf{n}^{ZW} \end{bmatrix} = \begin{bmatrix} n_X^{XW} & n_X^{YW} & n_X^{ZW} \\ n_Y^{XW} & n_Y^{YW} & n_Y^{YW} \\ n_Z^{XW} & n_Z^{YW} & n_Z^{YW} \end{bmatrix}$$

$\mathbf{n}^{XW} = \begin{bmatrix} n_X^{XW} & n_Y^{XW} & n_Z^{XW} \end{bmatrix}^T$ – normal vector of **OX**$_W$ axis in the CRF

$\mathbf{n}^{YW} = \begin{bmatrix} n_X^{YW} & n_Y^{YW} & n_Z^{YW} \end{bmatrix}^T$ – normal vector of **OY**$_W$ axis in the CRF

$\mathbf{n}^{ZW} = \begin{bmatrix} n_X^{ZW} & n_Y^{ZW} & n_Z^{ZW} \end{bmatrix}^T$ – normal vector of **OZ**$_W$ axis in the CRF

Camera-to-world

$$\mathbf{R}_{CW} = \mathbf{R}_{WC}^{\ T} = \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{n}^{XC} & \mathbf{n}^{YC} & \mathbf{n}^{ZC} \end{bmatrix} = \begin{bmatrix} n_X^{XC} & n_X^{YC} & n_X^{ZC} \\ n_Y^{XC} & n_Y^{YC} & n_Y^{YC} \\ n_Z^{XC} & n_Z^{YC} & n_Z^{YC} \end{bmatrix}$$

$\mathbf{n}^{XC} = \begin{bmatrix} n_X^{XC} & n_Y^{XC} & n_Z^{XC} \end{bmatrix}^T$ – normal vector of **OX**$_C$ axis in the WRF

$\mathbf{n}^{YC} = \begin{bmatrix} n_X^{YC} & n_Y^{YC} & n_Z^{YC} \end{bmatrix}^{YT}$ – normal vector of **OY**$_C$ axis in the WRF

$\mathbf{n}^{ZC} = \begin{bmatrix} n_X^{ZC} & n_Y^{ZC} & n_Z^{ZC} \end{bmatrix}^T$ – normal vector of **OZ**$_C$ axis in the WRF

Rotation Matrix ↔ Rotation Vector

Rotation vector – Rotation matrix

$$\mathbf{r} = [\theta, \psi, \gamma]^T \qquad \mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\mathbf{R}x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \qquad \mathbf{R}y = \begin{pmatrix} \cos\psi & 0 & \sin\psi \\ 0 & 1 & 0 \\ \sin\psi & 0 & \cos\psi \end{pmatrix} \qquad \mathbf{R}z = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{R} = \mathbf{R}x\mathbf{R}y\mathbf{R}z$$

Technical University of Cluj Napoca
Computer Science Department

Rotation Matrix ↔ Rotation Vector

Rotation vector

$\mathbf{r}_{WC} = [R_X\ R_Y\ R_Z]^T$ (R_X - pitch, R_Y - yaw, R_Z - tilt / roll)

$r_{WC} \Rightarrow R_{WC}$ transform:

$$r_{11} = \cos(R_Y)\cos(R_Z)$$
$$r_{12} = \sin(R_X)\sin(R_Y)\cos(R_Z) - \cos(R_X)\sin(R_Z)$$
$$r_{13} = \cos(R_X)\sin(R_Y)\cos(R_Z) + \sin(R_X)\sin(R_Z)$$
$$r_{21} = \cos(R_Y)\sin(R_Z)$$
$$r_{22} = \sin(R_X)\sin(R_Y)\sin(R_Z) + \cos(R_X)\cos(R_Z)$$
$$r_{23} = \cos(R_X)\sin(R_Y)\sin(R_Z) - \sin(R_X)\cos(R_Z)$$
$$r_{31} = -\sin(R_Y)$$
$$r_{32} = \sin(R_X)\cos(R_Y)$$
$$r_{33} = \cos(R_X)\cos(R_Y)$$

$R_{WC} \Rightarrow r_{WC}$ transform:

$$R_Y = \arcsin(r_{31})$$

If $\cos(R_Y) \neq 0$:

$$R_X = \operatorname{atan2}\left(-\frac{r_{32}}{\cos(R_Y)}, \frac{r_{33}}{\cos(R_Y)}\right)$$
$$R_Z = -\operatorname{atan2}\left(-\frac{r_{21}}{\cos(R_Y)}, \frac{r_{11}}{\cos(R_Y)}\right)$$

If $\cos(R_Y) = 0$:

$$R_X = \operatorname{atan2}(r_{12}, r_{22})$$

$$R_Z = 0$$

3D (world) ⇒ 2D (image) mapping using the Projection Matrix

Projection matrix

$$\mathbf{P} = \mathbf{A} \cdot \left[\mathbf{R}_{WC} \mid \mathbf{T}_{WC}\right]$$

The projection equation of a 3D world point [X_W , Y_W , Z_W] expressed in normalized coordinates :

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} = \mathbf{P} \cdot \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

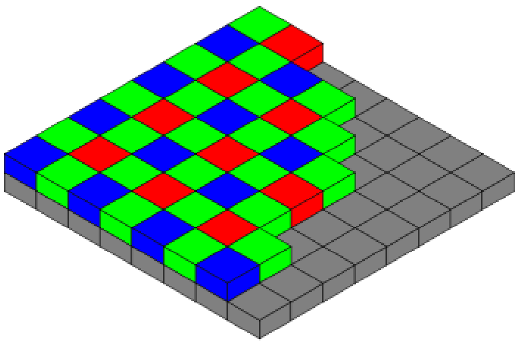
$s=z_S$ – scaling factor

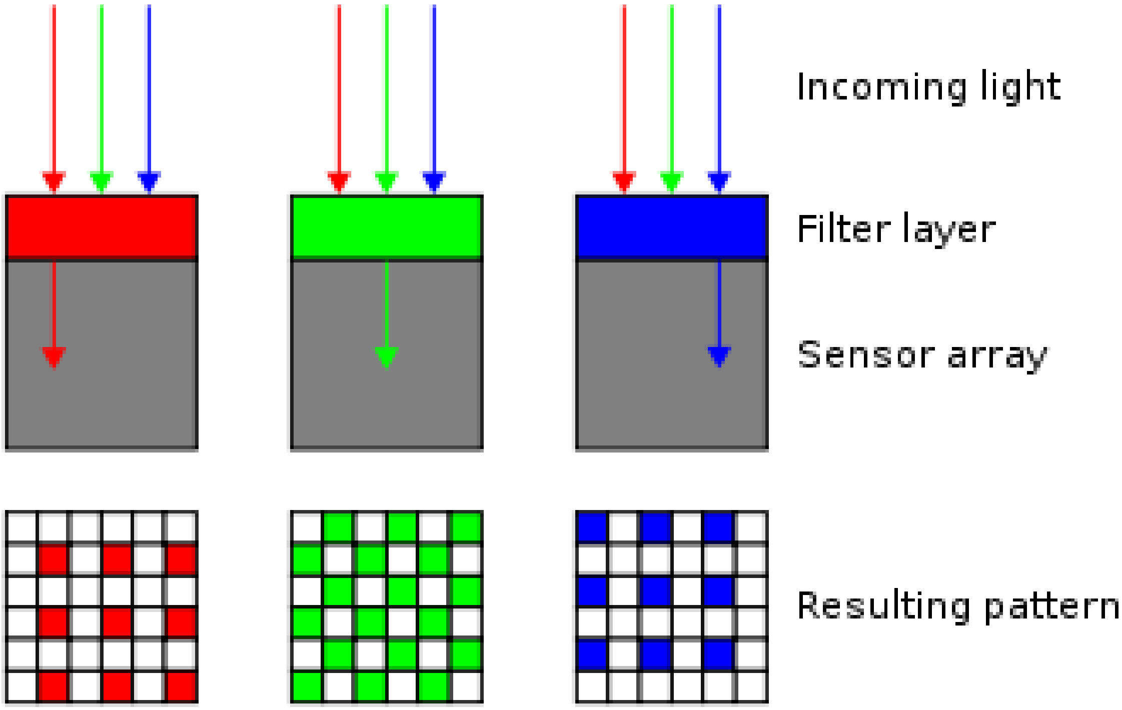
Obtaining the 2D image coordinates from normalized coordinate

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x_S / z_S \\ y_S / z_S \end{bmatrix}$$

Imaging sensors

Demosaicing Bayer pattern: Bilinear interpolation



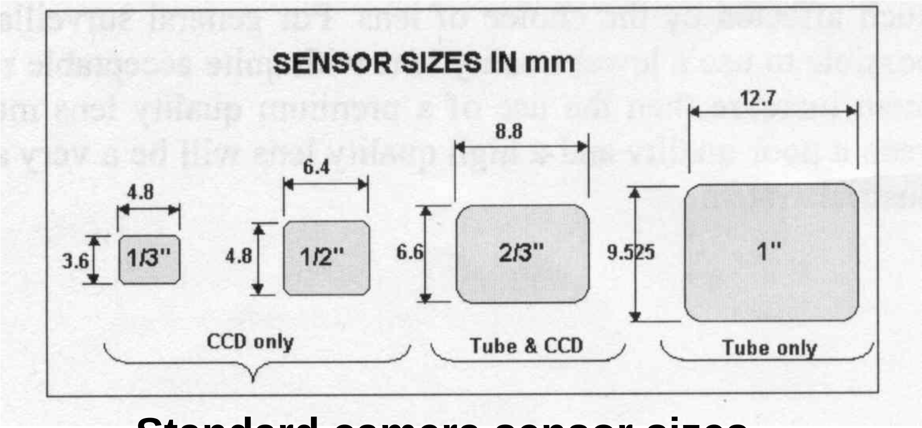


$$G = (G_n + G_w + G_e + G_s)/4$$
$$R_4 = (R_{nw} + R_{ne} + R_{se} + R_{sw})/4$$
$$R_{2c} = (R_n + R_s)/2$$
$$R_{2l} = (R_w + R_e)/2$$

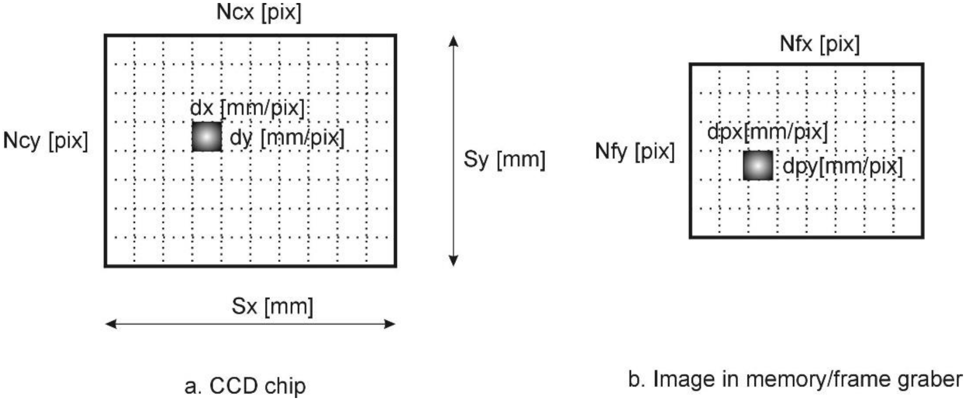
Technical University of Cluj Napoca
Computer Science Department

Imager parameters

Imager (sensor) parameters



Standard camera sensor sizes



Parameters of the imager and image in memory

Technical University of Cluj Napoca
Computer Science Department