How reconstruct a three-dimensional space from images

> A camera calibration course by *First Principles of Computer Vision*

When we take a picture, how know where each point are in world coordinate system? We only have images that measure points in terms of pixels

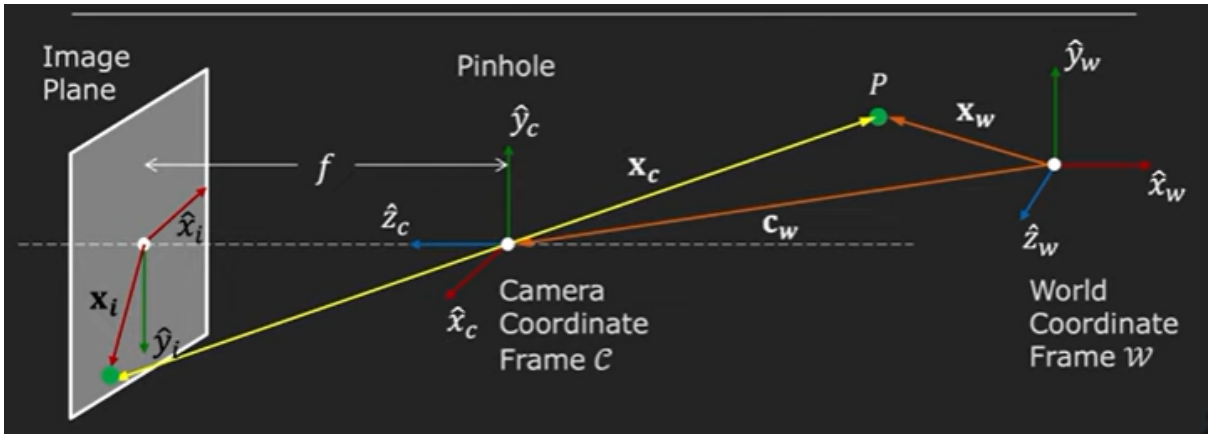
To do a full metric reconstruction, we need to know:

- The position and orientation of the camera in the world system, what we call by camera external parameters
- How the camera maps the perspective projetion points in the world onto its image, what we call by camera internal parameters

The process that determines the external and internal parameters are call **camera calibration.** To procede,it's need the **linear camera model.** Let's start to understand that process!

Finding a linear camera model

First, having a point in the world's coordinate system (X_w, Y_w, Z_w), convert to the camera's coordinate system (X_c, Y_c, Z_c) using a 3D to 3D transformation,what is called Coordinate Transformation.Now, having the point in the camera's coordinate frame, apply the Perspective Projection, a 3D to 2D transformation, that go from camera's coordinate system (X_c, Y_c, Z_c) to image's plane (X_i, Y_i)



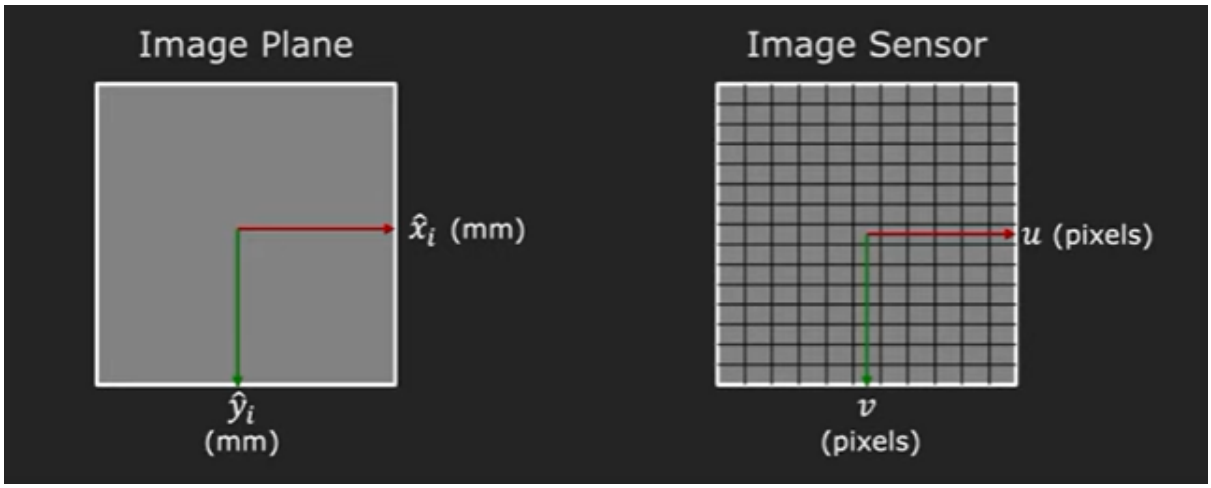
Representation of the coordinates transformations

The coordinates of the point P in the exemple above onto the image is:

$$X_i = f \cdot \frac{X_c}{Z_c} \qquad Y_i = f \cdot \frac{Y_c}{Z_c}$$

Where f is the focal distance, a distance between the effective central projection and the image plane

Image plane to Image sensor mapping:



Assuming that m_x and m_y are the pixel density in $\hat{x}$ and $\hat{y}$ directions, the pixel coordinates can be calculate by:

$$u = m_x \cdot x_i = m_x \cdot f \cdot \frac{X_c}{Z_c}$$

$$v = m_y \cdot y_i = m_y \cdot f \cdot \frac{Y_c}{Z_c}$$

The top-left corner of the image is usually treated as the origin. So, the point (O_x, O_y) is the principle point, where the optical axis pierces the sensor. Adding the principal point, the pixels coordinates now are:

$$u = m_x \cdot f \cdot \frac{X_c}{Z_c} + O_x$$

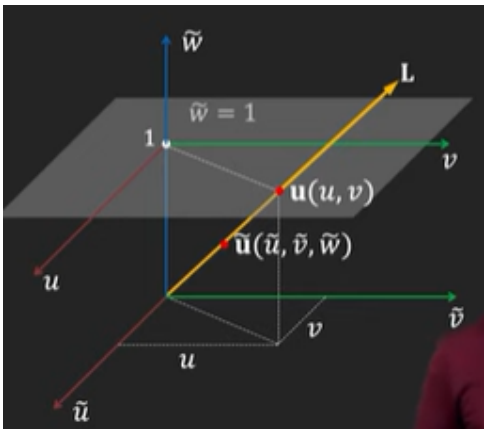
$$v = m_y \cdot f \cdot \frac{Y_c}{Z_c} + O_y$$

The product $m_x \cdot f$ and $m_y \cdot f$ is substituted by (f_x, f_y), the focal distance in pixels in the $\hat{x}$ and $\hat{y}$ directions. These unknown variables (f_x, f_y, O_x, O_y) are the intrisic parameters of the camera and they coordinate the camera internal geometry.

The equations above are non-linear, needing a **Homogenous Coordinates** transformation to linearize these equations. The homogenous representation of a 2D point $u = (u, v)$ in 3D space is a point $\tilde{u} = (\tilde{u}, \tilde{v}, \tilde{w})$. The z-coordinate $(\tilde{w} \neq 0)$ is fictitious, used just for sealing and normalization purpose.

Using Homogenous Coordinates, the pixels coordinates are:

$$u = \frac{\tilde{u}}{\tilde{w}} \qquad v = \frac{\tilde{v}}{\tilde{w}} \qquad \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix}$$



Every point on line L represente the homogenous coordinate of u(u,v)

Replacing these equations on perspective projection:

$$\begin{bmatrix} Z_c \cdot u \\ Z_c \cdot v \\ Z_c \end{bmatrix} = \begin{bmatrix} f_x \cdot X_c + Z_c \cdot O_x \\ f_y \cdot Y_c + Z_c \cdot O_y \\ Z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & O_x & 0 \\ 0 & f_y & O_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

This last matrix are called the intrisic matrix, that contains all the internal camera parameters. There is multiplied by the homogenous coordinates of the three dimensional point defined on the camera coordinate frame.

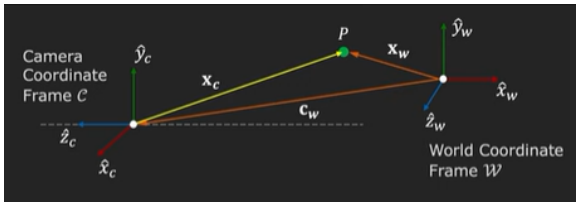
Excluding the last collumn, the remain matrix is called Calibration Matrix (K), a upper-triangular matrix.

$$\begin{bmatrix} f_x & 0 & O_x \\ 0 & f_y & O_y \\ 0 & 0 & 1 \end{bmatrix}$$

That can be represented by:

$$\tilde{u} = [\ K \mid 0\ ] \cdot \tilde{X_c} = M_{int} \cdot \tilde{X_c}$$

How map a point from the world to the camera coordinates?



The position c_w and the orientation R of the camera in the world coordinate frame are the camera's **extrinsic parameters**

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

Where the first line is the direction of $\tilde{X_c}$ in w, the second is the direction of $\tilde{Y_c}$ in w and the last one is the direction of $\tilde{Z_c}$ in w. R is a orthonormal matrix, so your inverse is equals to your transpose, what is the same to say that the product of R by R^T is equals to I, a identity matrix.

With the extrinsic parameters (R, c_w) of the camera and a point P in the world coordinate system, your camera-centric location is calculated by:

$$X_c = R(X_w - c_w) = RX_w - R_{c_w} = RX_w + t\ , \text{where t is the translation vector}$$

Rewriting the equation above:

$$X_c = \begin{bmatrix} \hat{X}_c \\ \hat{Y}_c \\ \hat{Z}_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Using homogeneous coordinates:

$$\tilde{X}_c = \begin{bmatrix} \hat{X}_c \\ \hat{Y}_c \\ \hat{Z}_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

That matrix $\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$ is the extrinsic matrix (M_{ext}), containing all external camera parameters.

So the world to camera transformation is

$$\begin{bmatrix} \hat{X}_c \\ \hat{Y}_c \\ \hat{Z}_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \qquad \tilde{X}_c = M_{ext} \cdot \tilde{X}_w$$

And the camera to pixel transformation:

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} f_x & 0 & O_x & 0 \\ 0 & f_y & O_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \qquad \tilde{u} = M_{int} \cdot \tilde{X}_c$$

Uning these two transformations, the result is **one transformation that goes to world's coordinate frame to image's coordinate frame**

$$\tilde{u} = M_{ext} \cdot M_{int} \cdot \tilde{X}_w = P \cdot \tilde{X}_w$$

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

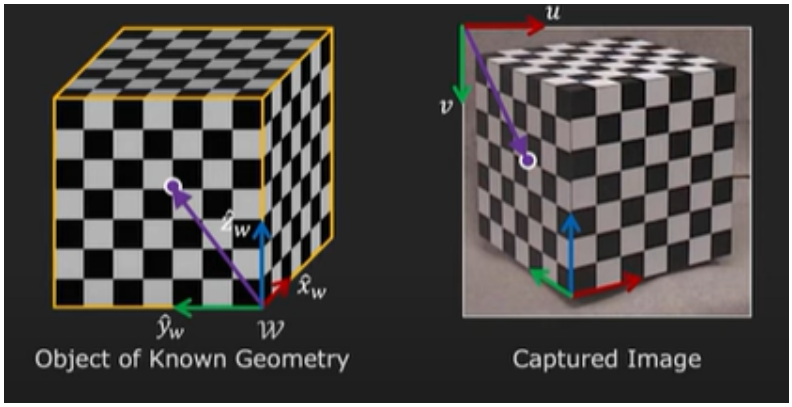
A linear camera model are found! To finally calibrate your camera all needed is the projection matrix P

Camera calibration process - Step-by-step

(**1)** Capture an image of an object with know geometry

- The location of each point in the world coordinate system is already determinate

(2) Find the correspondeces between 3D scene points and image points



(3) For each corresponding point i in scene :

$$\begin{bmatrix} \tilde{u_i} \\ \tilde{v_i} \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} \cdot \begin{bmatrix} Xi_w \\ Yi_w \\ Zi_w \\ 1 \end{bmatrix}$$

The images points and world points are know. The projection matrix is the last part that is needed.

(4) Rearrenge the projection matrix to find then - Decompose into intrinsic matrix and extrinsic matrix

$$p = M_{int} \cdot M_{ext}$$

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} = \begin{bmatrix} f_x & 0 & O_x \\ 0 & f_y & O_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = K \cdot R$$

$$\begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix} = \begin{bmatrix} f_x & 0 & O_x \\ 0 & f_y & O_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = K \cdot t$$

Where K is the calibration matrix, R the rotation matrix and t the translation vector. K is a upper right triangle matrix and R is a orthonormal. To solve the product $K \cdot R$ use QR factorization

(5) Now, the projection matrix is already know!

Other intrisic parameters: Distortions

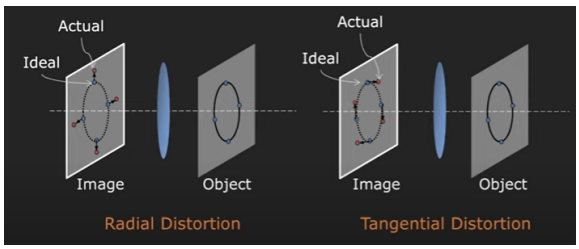
Pinholes don’t exhibit image distortion, but lenses do! There are two tips of distortions

Radial Distortions

Points are throw out away from center

Tangencial Distortions

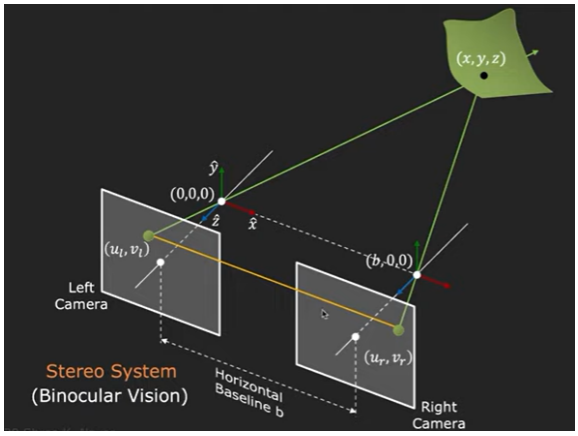
Points are twisted in the image



The intrinsic model of the camera will need to include the distortions coefficients

Stereo Vision - Recovery the three dimensional structure of a scene from two images

To reconstructed a 3D scene, it’s needed more information that just one camera can provides. It’s needed a stereo system



Find the correspondet image point on the right camera by tracing the rays. Where the rays intercepts, it’s the position of the object

From perspective projection, the coordinates of image points of each camera:

$$(u_l, v_l) = (f_x \frac{x}{z} + O_x, f_y \frac{y}{z} + O_y)$$
$$(u_r, v_r) = (f_x \frac{x - b}{z} + O_x, f_y \frac{y}{z} + O_y)$$

Solving for (x, y, z) :

$$x = \frac{b(u_l - O_x)}{u_l - u_r}$$

$$y = \frac{b f_x (v_l - O_y)}{f_y (u_l - u_r)}$$

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

Where z represents the depth of the point and $(u_l - u_r)$ the disparity

💡 **Observation:**

The depth z is inversely proportional to disparity:

- If the point is too close to the system, the depth decrease and the disparity increase. The inverse is also true. If the point goes to infinity, the disparity goes to zero

The disparity is proportional to the baseline b

Stereo Matching: Find the disparity between cameras

Remeber: The corresponding scene points lies on the same horizontal scan line. Because of this, the disparity is determinate by using template matching.

💡 Template Matching

- Look for corresponding points in the same horizontal line

To procede with stereo matching, first take some area to macth with other. This area is called window and their size is adaptive. Match points using multiple sizes and use the disparity to see the best similarity measure. Some issus with this procedure are the surface of the object, which must have texture and the foreshortenning effect.