The pinhole camera model is a mathematical representation that describes the relationship between a 3D point and its projection into the image plane. This representation is idealistic since no lenses are used to focus the light. The image plane is the 2D plane that corresponds to the sensing array. Figure 1 shows a pinhole camera where

- Point $C$: center of projection, it is the camera center.
- Plane $(x_c, y_c)$ is parallel to the image plane.
- $z_c$ is perpendicular to the image plane and therefore it is along the optical axis.
- $\lambda$ is the focal length.
- The point at which the optical axis and the image plane intersect is called the principal point.
- Point $P$ is a 3D point with coordinates $(x, y, z)$ in the camera reference frame.
- Point $e$ coordinates are $e = [u, v, \lambda]$.

The pinhole assumption

In the pinhole camera model, points $P$, $e$ and $C$ are collinear. In this case, there is a positive number $K$ such that:

$$K \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ \lambda \end{bmatrix}$$

(1)

Which can be written as

$$Kx = u$$

(2)

$$Ky = v$$

(3)

$$Kz = \lambda$$

(4)

$$K = \frac{\lambda}{z}$$

(5)

and therefore

![Fig. 1. Pinhole camera model](image-url)
Fig. 2. Illustration of the offset between image plane and pixel coordinates

Equation (6) is called the perspective projection. It is possible to write

\[
\begin{bmatrix}
u \\
v \\
1
\end{bmatrix} = \begin{bmatrix}
\frac{\lambda x}{z} \\
\frac{\lambda y}{z} \\
0
\end{bmatrix}
\]

(6)

Equation (6) is called the perspective projection. It is possible to write

\[
\begin{bmatrix}
u \\
v \\
1
\end{bmatrix} = A_0 \begin{bmatrix}
x \\
y \\
z
\end{bmatrix}
\]

(7)

with

\[
A_0 = \begin{bmatrix}
\frac{\lambda}{z} & 0 & 0 \\
0 & \frac{\lambda}{z} & 0 \\
0 & 0 & \frac{1}{z}
\end{bmatrix}
\]

(8)

In addition to the image plane coordinate system \((u, v)\), it is possible to define the pixel coordinate system \((r, c)\) as shown...
in figure 3. The relationship between the two frames is described as follows:

\[
\begin{align*}
    r &= u_0 + u \\
    c &= v_0 + v \\
    r &= u_0 + \frac{\lambda x}{z} \\
    c &= v_0 + \frac{\lambda y}{z}
\end{align*}
\]

where \((u_0, v_0)\) are the coordinates of the origin of the image plane in the \((r, c)\) reference frame. It is possible to write

\[
\begin{bmatrix}
    r \\
    c \\
    1
\end{bmatrix} =
\begin{bmatrix}
    \frac{\lambda}{z} & 0 & \frac{u_0}{z} \\
    0 & \frac{\lambda}{z} & \frac{v_0}{z} \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    z
\end{bmatrix}
\]

This model assumes that pixels have unit size. The sensing elements are not of unit size. Let \(s_x, s_y\) be the horizontal and vertical dimension of pixels. We have

\[
\begin{align*}
    r &= u_0 + \frac{u}{s_x} \\
    c &= v_0 + \frac{v}{s_y} \\
    s_x(r - u_0) &= \frac{\lambda x}{z} \\
    s_y(c - v_0) &= \frac{\lambda x}{z} \\
    r &= u_0 + \frac{\lambda x}{zs_x} \\
    c &= v_0 + \frac{\lambda y}{zs_y}
\end{align*}
\]

**Example**

Let

\[
\begin{align*}
    s_x &= 0.05mm \\
    s_y &= 0.05mm
\end{align*}
\]

Assuming that

\[
(u_0, v_0) = (6, 4)
\]

Find \((r, c)\) if \((u, v) = (0.3mm, 0.25mm)\).

\[
\begin{align*}
    r &= 6 + 0.3/0.05 = 12 \\
    c &= 4 + 0.25/0.05 = 9
\end{align*}
\]

It is possible to write under matrix form

\[
\begin{bmatrix}
    r \\
    c \\
    z
\end{bmatrix} =
\begin{bmatrix}
    \frac{\lambda}{zs_x} & 0 & \frac{u_0}{zs_x} \\
    0 & \frac{\lambda}{zs_y} & \frac{v_0}{zs_y} \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    z
\end{bmatrix}
\]

or

\[
\begin{bmatrix}
    zr \\
    zc \\
    z
\end{bmatrix} =
\begin{bmatrix}
    \frac{\lambda}{s_x} & 0 & u_0 \\
    0 & \frac{\lambda}{s_y} & v_0 \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    z
\end{bmatrix}
\]
We put:

\[ \alpha_u = \frac{\lambda}{s_x} \]  
\[ \alpha_v = \frac{\lambda}{s_y} \]  \hspace{1cm} (27)

The 3 by 3 matrix given by

\[
A = \begin{bmatrix}
\alpha_u & 0 & u_0 \\
0 & \alpha_v & v_0 \\
0 & 0 & 1
\end{bmatrix}
\]  \hspace{1cm} (29)

is called the intrinsic parameter matrix. Axes \( r \) and \( c \) of the pixel coordinate system are not necessarily orthogonal. They are inclined by an angle \( \theta \). In this case, matrix \( A \) becomes:

\[
A = \begin{bmatrix}
\alpha_u & \gamma & u_0 \\
0 & \alpha_v & v_0 \\
0 & 0 & 1
\end{bmatrix}
\]  \hspace{1cm} (30)

where \( \gamma \) is a parameter describing the skew of the two image axes. \( \alpha_u, \alpha_v, \gamma, u_0, v_0 \) are called the intrinsic camera parameters.

The relationship between the coordinates of point \( P \) in the camera reference frame and the world reference frame is given by

\[
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} = R \begin{bmatrix}
x_w \\
y_w \\
z_w
\end{bmatrix} + T
\]  \hspace{1cm} (31)

where \( R \) is a 3 \( \times \) 3 rotation matrix and \( T \) is a 3 \( \times \) 1 translation matrix. Note that \( (x, y, z) \) represents the coordinates of point \( P \) in the camera reference frame and \( (x_w, y_w, z_w) \) represents the coordinates of the same point in the world reference frame.

The relationship between the pixel and the world can be decomposed as follows

\[ \text{Translation} \rightarrow \text{Rotation} \rightarrow \text{Projection} \rightarrow A \]  \hspace{1cm} (32)

\[
H = A \begin{bmatrix}
R \\
T
\end{bmatrix}
\]  \hspace{1cm} (33)

The rotation matrix and the translation matrix together form the extrinsic parameters of the camera. Camera calibration refers to the determination of the intrinsic and the extrinsic parameters. Different approaches are used to achieve this goal including experimental methods and optimization techniques.

\[ \text{CAMERA CALIBRATION} \]

The goal from camera calibration is to determine the intrinsic and extrinsic parameters of the camera. These parameters play an important role when mapping a point to its image coordinates. Recall that, in the camera calibration process, it is possible to define four different reference frames:

- Scene coordinate system
- Camera coordinate system
- Image coordinate system
- Pixel coordinate system

The pixel coordinates and the corresponding 3D coordinates in the world reference frame of point \( P \) are denoted by

\[
m = \begin{bmatrix}
r \\
c
\end{bmatrix}
\]

\[
P_w = \begin{bmatrix}
x_w \\
y_w \\
z_w
\end{bmatrix}
\]  \hspace{1cm} (34)

We define the augmented vectors:

\[
\tilde{m} = \begin{bmatrix}
r \\
c \\
1
\end{bmatrix}
\]

\[
\tilde{P}_w = \begin{bmatrix}
x_w \\
y_w \\
z_w \\
1
\end{bmatrix}
\]  \hspace{1cm} (35)
It is possible to write the calibration equation in terms of the augmented parameters as follows

\[
\tilde{m} = A \begin{bmatrix} R & T \end{bmatrix} \tilde{P}_w
\]

\[
= H \tilde{P}_w
\]

- \( A \) is the intrinsic parameter matrix. It represents the internal characteristics of the camera: \( \alpha_u, \alpha_v, u_0, v_0, \gamma \)
- Matrices \( R \) and \( T \) represent the orientation (rotation) and location (translation) of the camera. Together they form the extrinsic parameters.

There exist several ways to represent and solve the problem. This summary discusses two different methods.

**Method 1**

Recall that

\[
r = u_0 + \frac{x}{x_w} = u_0 + \alpha_u x_w
\]

\[
c = v_0 + \frac{y}{y_w} = v_0 + \alpha_v y_w
\]

and

\[
\begin{bmatrix}
x & y & z \\
\end{bmatrix} =
\begin{bmatrix}
r_{11} & r_{12} & r_{13} \\
r_{21} & r_{22} & r_{23} \\
r_{31} & r_{32} & r_{33} \\
\end{bmatrix}
\begin{bmatrix}
x_w & y_w & z_w \\
\end{bmatrix} +
\begin{bmatrix}
T_x \\
T_y \\
T_z \\
\end{bmatrix}
\]

(39)

Combining (38) and (39), we get

\[
\begin{align*}
r - u_0 &= \alpha_u (r_{11}x_w + r_{12}y_w + r_{13}z_w + T_x) \\
c - v_0 &= \alpha_v (r_{21}x_w + r_{22}y_w + r_{23}z_w + T_y)
\end{align*}
\]

(40)

(41)

We make the following change of variable

\[
r \leftarrow r - u_0
\]

\[
c \leftarrow c - v_0
\]

(42)

(43)

where \( u_0 \) and \( v_0 \) are determined separately from the other variables. Now, we can write equations (40) and (41) with the new \( r, c \) as follows

\[
r(r_{31}x_w + r_{32}y_w + r_{33}z_w + T_z) = \alpha_u (r_{11}x_w + r_{12}y_w + r_{13}z_w + T_x)
\]

\[
c(r_{31}x_w + r_{32}y_w + r_{33}z_w + T_z) = \alpha_v (r_{21}x_w + r_{22}y_w + r_{23}z_w + T_y)
\]

\[
r\alpha_u (r_{21}x_w + r_{22}y_w + r_{23}z_w + T_y) = c\alpha_u (r_{11}x_w + r_{12}y_w + r_{13}z_w + T_x)
\]

(44)

(45)

(46)

If we take \( N \) points \((x_{wi}, y_{wi}, z_{wi}, r_i, c_i)\) with \( i = 1, ..., N \), we can write \( N \) equations:

\[
r_1\alpha_u (r_{21}x_{w1} + r_{22}y_{w1} + r_{23}z_{w1} + T_y) = c_1\alpha_u (r_{11}x_{w1} + r_{12}y_{w1} + r_{13}z_{w1} + T_x)
\]

\[
r_2\alpha_u (r_{21}x_{w2} + r_{22}y_{w2} + r_{23}z_{w2} + T_y) = c_2\alpha_u (r_{11}x_{w2} + r_{12}y_{w2} + r_{13}z_{w2} + T_x)
\]

\[
\vdots
\]

\[
r_N\alpha_u (r_{21}x_{wN} + r_{22}y_{wN} + r_{23}z_{wN} + T_y) = c_N\alpha_u (r_{11}x_{wN} + r_{12}y_{wN} + r_{13}z_{wN} + T_x)
\]

(47)

We can combine the \( N \) equations into matrix form:

\[
Ax = 0
\]

with

\[
A =
\begin{bmatrix}
c_1 x_{w1} & c_1 y_{w1} & c_1 z_{w1} & c_1 & -r_{11} x_{w1} & -r_{11} y_{w1} & -r_{11} z_{w1} & -r_1 \\
c_2 x_{w2} & c_2 y_{w2} & c_2 z_{w2} & c_2 & -r_{11} x_{w2} & -r_{11} y_{w2} & -r_{11} z_{w2} & -r_2 \\
\vdots \\
c_N x_{wN} & c_N y_{wN} & c_N z_{wN} & c_N & -r_{11} x_{wN} & -r_{11} y_{wN} & -r_{11} z_{wN} & -r_N
\end{bmatrix}
\]

(48)

(49)
This equation can be solved using singular value decomposition SVD. The solution $x$ is the eigenvector corresponding to the only zero eigenvalue of $A^T A$.

$$[U,D,V] = \text{svd}(A);$$

$$x = V(:, \text{end});$$

**Method 2**

It is possible to write

$$\tilde{m} = \tilde{H}\tilde{P}_w$$

with

$$\tilde{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \end{bmatrix}$$

Matrix $\tilde{H}$ has 12 entries but only 11 are independent. We can write it under the following form (by dividing by one element such as $h_{34}$ for example):

$$\tilde{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & 1 \end{bmatrix}$$

Now, have

$$r = \frac{h_{11}x_w + h_{12}y_w + h_{13}z_w + h_{14}}{h_{31}x_w + h_{32}y_w + h_{33}z_w + 1}$$

$$c = \frac{h_{21}x_w + h_{22}y_w + h_{23}z_w + h_{24}}{h_{31}x_w + h_{32}y_w + h_{33}z_w + 1}$$

which can be written as

$$\begin{bmatrix} h_{31}x_w + h_{32}y_w + h_{33}z_w + 1 \end{bmatrix} r = h_{11}x_w + h_{12}y_w + h_{13}z_w + h_{14}$$

$$\begin{bmatrix} h_{31}x_w + h_{32}y_w + h_{33}z_w + 1 \end{bmatrix} c = h_{21}x_w + h_{22}y_w + h_{23}z_w + h_{24}$$

Or under matrix form:

$$M = \begin{bmatrix} x_w & y_w & z_w & 1 & 0 & 0 & 0 & 0 & -rx_w & -ry_w & -rz_w \\ 0 & 0 & 0 & 0 & x_w & y_w & z_w & 1 & -cx_w & -cy_w & -cz_w \end{bmatrix}$$
We need at least six world-image point correspondences to solve. Assuming we take \( N \) points, we have

\[
M_N = \begin{bmatrix}
    x_w1 & y_w1 & z_w1 & 1 & 0 & 0 & 0 & 0 & -r_xw1 & -r_yw1 & -r_zw1 \\
    0 & 0 & 0 & 0 & x_w1 & y_w1 & z_w1 & 1 & -c_xw1 & -c_yw1 & -c_zw1 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    x_wN & y_wN & z_wN & 1 & 0 & 0 & 0 & 0 & -r_xwN & -r_ywN & -r_zwN \\
    0 & 0 & 0 & 0 & x_wN & y_wN & z_wN & 1 & -c_xwN & -c_ywN & -c_zwN
\end{bmatrix}
\]

Under matrix form:

\[
M_N H_v = m_N
\]  

(62)

with

\[
m_N = \begin{bmatrix}
    r_1 & c_1 & \ldots & r_N & c_N
\end{bmatrix}^T
\]

\[
H_v = \begin{bmatrix}
    h_{11} & h_{12} & h_{13} & h_{14} & h_{21} & h_{22} & h_{23} & h_{34} & h_{31} & h_{32} & h_{33}
\end{bmatrix}^T
\]

System (62) is an overdetermined system with solution given by:

\[
M_N^T M_N H_v = M_N^T m_N
\]  

(63)

\[
H_v = (M_N^T M_N)^{-1} M_N^T m_N
\]  

(64)

In general for overdetermined systems is possible to solve

- using \( L_1 \) norm optimization: by minimizing the following cost function

\[
\text{minimize } ||M_N H_v - m_N||
\]  

(65)

- using nonlinear least square: by minimizing the following cost function

\[
\text{minimize } ||M_N H_v - m_N||^2
\]  

(66)

**RADIAL DISTORTION**

The projection model assumes a linear projection between the camera and the real world. This is not always the case. Straight line in the world are not always straight lines in the image. This is called radial distortion. It is common with wide angle lenses. The coordinates in the distorted image are displaced away or towards the image center. We use the following notation

- Distorted image \((r_d, c_d)\)
- Undistorted image \((r, c)\)

The relationship between the distorted image and the undistorted image is given by

\[
\begin{bmatrix}
    r_d \\
    c_d
\end{bmatrix} = (1 + k_1 R^2) \begin{bmatrix}
    r - u_0 \\
    c - v_0
\end{bmatrix} \begin{bmatrix}
    u_0 \\
    v_0
\end{bmatrix}
\]  

(67)

where \( k_1 \) is the radial distortion parameter. with

\[
R^2 = (r - u_0)^2 + (c - v_0)^2
\]  

(68)

- Nokia 6103: \( k_1 = -2 \times 10^{-7} \)
- Sony Ericsson 760i: \( k_1 = 6 \times 10^{-8} \)

**TSAI CALIBRATION METHOD (1987)**

The implementation of this method requires the knowledge of the 3D coordinates and their corresponding image coordinates. The method solves first for the position and orientation of the camera and then solves for the internal parameters.

Instead of using 3D coordinates, Zhang suggested to use a planar grid with chessboard like pattern where the corners are visible. This approach is called calibration from a planar grid and is very common. The method can be summarized as follows:

- Take several pictures of the pattern shown at different orientations and positions.
- By knowing the 2D position of the corners on the real pattern and their corresponding pixel coordinates on the on each image, solve for the intrinsic and extrinsic parameters.
- The method uses least squares followed by nonlinear refinement such as Gauss Newton.
- The accuracy improves with the number of images.