Image Formation and Camera Calibration

Ying Wu
Electrical Engineering & Computer Science
Northwestern University
Evanston, IL 60208
yingwu@ece.northwestern.edu

Contents

1 Pinhole Camera Model

1.1 Perspective Projection

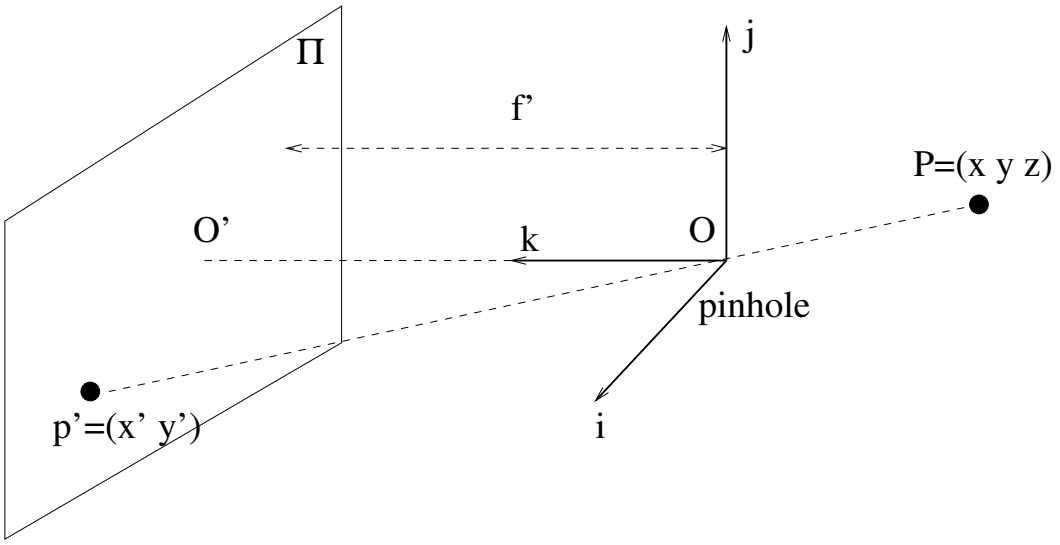

Figure 1: Illustration of perspective projection of a pinhole camera

From Figure 1, we can easily figure out:

$$\begin{cases} x' &= f'x/z \\ y' &= f'y/z \end{cases} \tag{1}$$

Note: Thin lenses cameras has the same geometry as pinhole cameras. Thin lenses camera geometry:

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

1.2 Weak-Perspective Projection

$$\begin{cases} x' &= (f'/z_0)x \\ y' &= (f'/z_0)y \end{cases} \tag{2}$$

When the scene depth is small comparing to the average distance from the camera, i.e., the depth of the scene is "flat", weak-perspective projection is a good approximation of perspective projection. It is also called *scaled-orthographic projection*.

1.3 Orthographic Projection

$$\begin{cases} x' &= x \\ y' &= y \end{cases} \tag{3}$$

When camera always remains at a roughly constant distance from the scene, and scene centers the optic axis, orthographic projection could be used as an approximation. Obviously, orthographic projection is not real.

2 Homogeneous Coordinates

A 3D point is

$$\mathbf{P} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

and a plane can be written by $ax + by + cz - d = 0$. We use homogeneous coordinates to unify the representation for points and lines by adding one dimensionality, i.e., we use

$$\mathbf{P} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

for points and

$$\mathbf{\Pi} = \begin{pmatrix} a \\ b \\ c \\ -d \end{pmatrix}$$

for planes, where the plane $\mathbf{\Pi}$ is defined up to a scale. We have

$$\mathbf{\Pi} \cdot \mathbf{P} = 0$$

3 Coordinate System Changes and Rigid Transformations

For convenience, We use the "Craig notation". ${}^{F}\mathbf{P}$ means the coordinates of point P in frame F.

3.1 Translation

$$ {}^{B}\mathbf{P} = {}^{A}\mathbf{P} + {}^{B}\mathbf{O}_{A} \tag{4}$$

where ${}^{B}\mathbf{O}_{A}$ is the coordinate of the origin $\mathbf{O}_{A}$ of frame A in the new coordinate system B.

3.2 Rotation

$$ {}^{B}_{A}\mathbf{R} = ({}^{B}\mathbf{i}_A \ {}^{B}\mathbf{j}_A \ {}^{B}\mathbf{k}_A) = \begin{pmatrix} {}^{A}\mathbf{i}_B^T \\ {}^{A}\mathbf{j}_B^T \\ {}^{A}\mathbf{k}_B^T \end{pmatrix} \tag{5} $$

where ${}^{B}\mathbf{i}_A$ is the coordinate of the axis $\mathbf{i}_A$ of frame A in the new coordinate system B. Then we have,

$$ {}^{B}\mathbf{P} = {}^{B}_{A}\mathbf{R}\,{}^{A}\mathbf{P} $$

3.3 Rigid Transformation

$$ {}^{B}\mathbf{P} = {}^{B}_{A}\mathbf{R}\,{}^{A}\mathbf{P} + {}^{B}\mathbf{O}_A \tag{6} $$

Let's see here an advantage of the homogeneous coordinates. If we make two consecutive rigid transformation, i.e., from A$\rightarrow$ B $\rightarrow$ C, the coordinate of point P in frame C will be written by:

$$ {}^{C}\mathbf{P} = {}^{C}_{B}\mathbf{R}({}^{B}_{A}\mathbf{R}\,{}^{A}\mathbf{P} + {}^{B}\mathbf{O}_A) + {}^{C}\mathbf{O}_B = {}^{C}_{B}\mathbf{R}\,{}^{B}_{A}\mathbf{R}\,{}^{A}\mathbf{P} + ({}^{C}_{B}\mathbf{R}\,{}^{B}\mathbf{O}_A + {}^{C}\mathbf{O}_B) $$

It looks very awkward. If we present it by homogeneous coordinates, it looks very concise.

$$ \begin{bmatrix} {}^{B}\mathbf{P} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{B}_{A}\mathbf{R} & {}^{B}\mathbf{O}_A \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} {}^{A}\mathbf{P} \\ 1 \end{bmatrix} $$

and

$$ \begin{bmatrix} {}^{C}\mathbf{P} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{C}_{B}\mathbf{R} & {}^{C}\mathbf{O}_B \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} {}^{B}\mathbf{P} \\ 1 \end{bmatrix} $$

So,

$$ \begin{bmatrix} {}^{C}\mathbf{P} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{C}_{B}\mathbf{R} & {}^{C}\mathbf{O}_B \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} {}^{B}_{A}\mathbf{R} & {}^{B}\mathbf{O}_A \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} {}^{A}\mathbf{P} \\ 1 \end{bmatrix} $$

3.4 Summary

We could write a transformation by:

$$ T = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} $$

We call it a *projective transformation*. If we can write:

$$ T = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} $$

It becomes an *affine transformation*. If $\mathbf{A} = \mathbf{R}$, i.e., a rotation matrix ($\mathbf{R}^T\mathbf{R} = \mathbf{I}$),

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

it becomes a *Euclidean transformation* or a *rigid transformation*. Obviously, Euclidean transformation preserves both parallel lines and angles, but affine preserves parallel lines but not angles.

4 Image Formation (Geometrical)

In this section, we discuss the process of image formation in terms of geometry. For a 3D point $\mathbf{p}^w = [x^w, y^w, z^w]^T$ in the world coordinate system, its will be mapped to a camera coordinate system (C) from the world coordinate system (F), then map to the physical retina, i.e., the physical image plane, and get image coordinates $[u, v]^T$. We shall ask how this 3D point is mapped to its image coordinate.

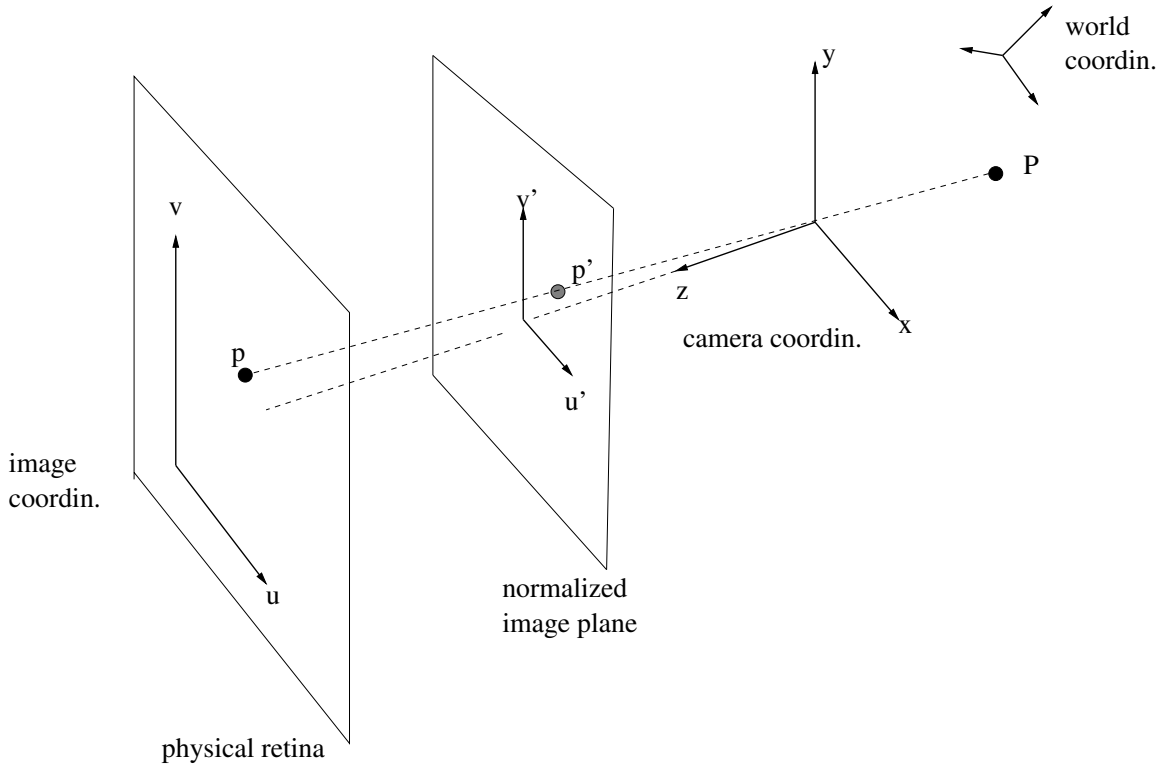

Figure 2: Illustration of the geometry of the image formation process under perspective projection of a pinhole camera

For convenience, we introduce a normalized image plane located at the focal length $f = 1$. In such a normalized image plane, the pinhole (c) is mapped to the origin of the image plane

($\hat{c}$), and $\mathbf{p}$ is mapped to $\hat{\mathbf{p}} = [\hat{u}, \hat{v}]^T$.

$$\hat{\mathbf{p}} = \begin{bmatrix} \hat{u} \\ \hat{v} \\ 1 \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p}^c \\ 1 \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_c \\ y^c \\ z^c \\ 1 \end{bmatrix}$$

And we also have

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} kf & 0 & u_0 \\ 0 & lf & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} kf & 0 & u_0 \\ 0 & lf & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \\ 1 \end{bmatrix}$$

Let $\alpha = kf$ and $\beta = lf$. We call these parameters α,β,u_0 and v_0 *intrinsic parameters*, which present the inner camera imaging parameters. We can write

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} \alpha & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \\ 1 \end{bmatrix} = \frac{1}{z^c} \begin{bmatrix} \alpha & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ z^w \\ 1 \end{bmatrix}$$

We call $\mathbf{R}$ and $\mathbf{t}$ *extrinsic parameters*, which represent the coordinate transformation between the camera coordinate system and the world coordinate system. So, we can write,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z^c} \mathbf{M_1} \mathbf{M_2} \mathbf{p}^w = \frac{1}{z^c} \mathbf{M} \mathbf{p}^w \tag{7}$$

We call $\mathbf{M}$ the *projection matrix*.

5 Camera Calibration – Inferring Camera Parameters

5.1 The Setting of the Problem

We are given (1) a calibration rig, i.e., a reference object, to provide the world coordinate system, and (2) an image of the reference object. The problem is to solve (a) the projection matrix, and (b) the intrinsic and extrinsic parameters. Mathematically, given $[x_i^w, y_i^w, z_i^w]^T, i = 1, \ldots, n$, and $[u_i, v_i]^t, i = 1, \ldots, n$, we want to solve $\mathbf{M}_1$ and $\mathbf{M}_2$, s.t.,

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \frac{1}{z_i^c} \mathbf{M}_1 \mathbf{M}_2 \begin{bmatrix} x_i^w \\ y_i^w \\ z_i^w \\ 1 \end{bmatrix} = \frac{1}{z_i^c} \mathbf{M} \begin{bmatrix} x_i^w \\ y_i^w \\ z_i^w \\ 1 \end{bmatrix}, \quad \forall i$$

5.2 Computing the Projection Matrix

$$z_i^c \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x_i^w \\ y_i^w \\ z_i^w \\ 1 \end{bmatrix}$$

We can write

$$\begin{aligned} z_i^c u_i &= m_{11}x_i^w + m_{12}y_i^w + m_{13}z_i^w + m_{14} \\ z_i^c v_i &= m_{21}x_i^w + m_{22}y_i^w + m_{23}z_i^w + m_{24} \\ z_i^c &= m_{31}x_i^w + m_{32}y_i^w + m_{33}z_i^w + m_{34} \end{aligned}$$

Then

$$\begin{aligned} x_i^w m_{11} + y_i^w m_{12} + z_i^w m_{13} + m_{14} - u_i x_i^w m_{31} - u_i y_i^w m_{32} - u_i z_i^w m_{33} &= u_i m_{34} \\ x_i^w m_{21} + y_i^w m_{22} + z_i^w m_{23} + m_{24} - v_i x_i^w m_{31} - v_i y_i^w m_{32} - v_i z_i^w m_{33} &= v_i m_{34} \end{aligned}$$

Then,

$$\begin{bmatrix} x_1^w & y_1^w & z_1^w & 1 & 0 & 0 & 0 & 0 & -u_1x_1^w & -u_1y_1^w & -u_1z_1^w \\ 0 & 0 & 0 & 0 & x_1^w & y_1^w & z_1^w & 1 & -v_1x_1^w & -v_1y_1^w & -v_1z_1^w \\ \vdots & & & \vdots & & & \vdots & & & \vdots & \vdots \\ x_n^w & y_n^w & z_n^w & 1 & 0 & 0 & 0 & 0 & -u_nx_n^w & -u_ny_n^w & -u_nz_n^w \\ 0 & 0 & 0 & 0 & x_n^w & y_n^w & z_n^w & 1 & -v_nx_n^w & -v_ny_n^w & -v_nz_n^w \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ \vdots \\ m_{32} \\ m_{33} \end{bmatrix} = \begin{bmatrix} u_1m_{34} \\ v_1m_{34} \\ u_2m_{34} \\ v_2m_{34} \\ \vdots \\ u_nm_{34} \\ v_nm_{34} \end{bmatrix} \quad (8)$$

Obviously, we can let $m_{34} = 1$, i.e., the projection matrix is scaled by m_{34}. We have:

$$\mathbf{Km} = \mathbf{U} \quad (9)$$

Where, $\mathbf{K}$ is a $2n \times 11$ matrix, $\mathbf{m}$ is a 11-D vector, and $\mathbf{U}$ is a 2n-D vector. The least squares solution of Equation 9 is obtainted by:

$$\mathbf{m} = \mathbf{K}^{\dagger}\mathbf{U} = (\mathbf{K^T K})^{-1}\mathbf{K}^T\mathbf{U} \quad (10)$$

where $\mathbf{K}^{\dagger}$ is the pseudoinverse of $\mathbf{K}$. Here, $\mathbf{m}$ and $m_{34} = 1$ constitute the projection matrix $\mathbf{M}$. This is a linear solution to the project matrix.

5.3 Computing Intrinsic and Extrinsic Parameters

After computing the projection matrix $\mathbf{M}$, we can compute the intrinsic and extrinsic parameters from $\mathbf{M}$. Please notice that the projection matrix $\mathbf{M}$ obtained from Equation 9 is a scaled version of the true $\mathbf{M}$, since we have let $m_{34} = 1$. To decompose the projection

matrix in order to solve $\mathbf{M}_1$ and $\mathbf{M}_2$, we need to take into account of m_{34}, i.e., the true m_{34} needs to be figured out. Apparently, we have:

$$m_{34}\mathbf{M} = m_{34}\begin{bmatrix}\mathbf{m}_1^T & m_{14}\\ \mathbf{m}_2^T & m_{24}\\ \mathbf{m}_3^T & 1\end{bmatrix} = \begin{bmatrix}\alpha & 0 & u_0 & 0\\ 0 & \beta & v_0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\mathbf{r}_1^T & t_x\\ \mathbf{r}_2^T & t_y\\ \mathbf{r}_3^T & t_z\\ \mathbf{0}^T & 1\end{bmatrix} = \begin{bmatrix}\alpha\mathbf{r}_1^T + u_0\mathbf{r}_3^T & \alpha t_x + u_0 t_x\\ \beta\mathbf{r}_2^T + v_0\mathbf{r}_3^T & \beta t_y + v_0 t_x\\ \mathbf{r}_3^T & t_x\end{bmatrix} \quad (11)$$

where $\mathbf{m}_i^T = [m_{i1}, m_{i2}, m_{i3}]$, $\mathbf{M} = \{m_{ij}\}$ is computed from Equation 9, and $\mathbf{r}_i^T = [r_{i1}, r_{i2}, r_{i3}]$, $\mathbf{R} = \{r_{ij}\}$ is the rotation matrix.

To see it clearly, we have:

$$\begin{bmatrix}\alpha\mathbf{r}_1^T + u_0\mathbf{r}_3^T & \alpha t_x + u_0 t_x\\ \beta\mathbf{r}_2^T + v_0\mathbf{r}_3^T & \beta t_y + v_0 t_x\\ \mathbf{r}_3^T & t_x\end{bmatrix} = \begin{bmatrix}m_{34}\mathbf{m}_1^T & m_{34}m_{14}\\ m_{34}\mathbf{m}_2^T & m_{34}m_{24}\\ m_{34}\mathbf{m}_3^T & m_{34}\end{bmatrix} \quad (12)$$

Obviously, by comparing these two matrices, it is easy to see $m_{34}\mathbf{m}_3 = \mathbf{r}_3$. In addition, since $\mathbf{R}$ is a rotation matrix, we have $|\mathbf{r}_i| = 1$. Then, we have

$$m_{34} = \frac{1}{|\mathbf{m}_3|}$$

Then, it is easy to figure out all the other parameters:

$$\begin{aligned}
\mathbf{r}_3 &= m_{34}\mathbf{m}_3 && (13)\\
u_0 &= (\alpha\mathbf{r}_1^T + u_0\mathbf{r}_3^T)\mathbf{r}^3 = m_{34}^2\mathbf{m}_1^T\mathbf{m}_3 && (14)\\
v_0 &= (\beta\mathbf{r}_2^T + v_0\mathbf{r}_3^T)\mathbf{r}^3 = m_{34}^2\mathbf{m}_2^T\mathbf{m}_3 && (15)\\
\alpha &= m_{34}^2|\mathbf{m}_1 \times \mathbf{m}_3| && (16)\\
\beta &= m_{34}^2|\mathbf{m}_2 \times \mathbf{m}_3| && (17)
\end{aligned}$$

After that, it is also easy to get:

$$\begin{aligned}
\mathbf{r}_1 &= \frac{m_{34}}{\alpha}(\mathbf{m}_1 - u_0\mathbf{m}_3) && (18)\\
\mathbf{r}_2 &= \frac{m_{34}}{\beta}(\mathbf{m}_2 - v_0\mathbf{m}_3) && (19)\\
t_z &= m_{34} && (20)\\
t_x &= \frac{m_{34}}{\alpha}(m_{14} - u_0) && (21)\\
t_y &= \frac{m_{34}}{\beta}(m_{24} - v_0) && (22)
\end{aligned}$$

Now, we have obtained all the intrinsic and extrinsic parameters. For the analysis of skewed camera models (i.e., $\theta \neq 90^o$), please read chapter 6 of Forsyth&Ponce.

5.4 Questions to Think Over

We'v introduced a linear approach for camera calibration. Let's think some questions over:

- The matrix $\mathbf{M}_1$, representing the intrinsic characteristics of cameras, has 4 independent variables; and the matrix $\mathbf{M}_2$, representing the extrinsic characteristics, has 6 independent variables. So the projection matrix $\mathbf{M}$ has 10 independent variables. When letting $m_{34} = 1$, we still have 11 parameters to determine. However, please note, these 11 parameters are not independent! But our solution by Equation 9 did not address such dependency or constraints among these 11 parameters. Consequently, computation errors are not avoidable. The question is that, if we can find an approach to take into account of that to enhance the accuracy?

- The method described above assumes that we know the 2D image coordinates. How can we get these 2D image points?

- If the detection of these 2D image points contains noise, how does the noise affect the accuracy of the result?

- If we are not using point-corespondences, can we use lines?