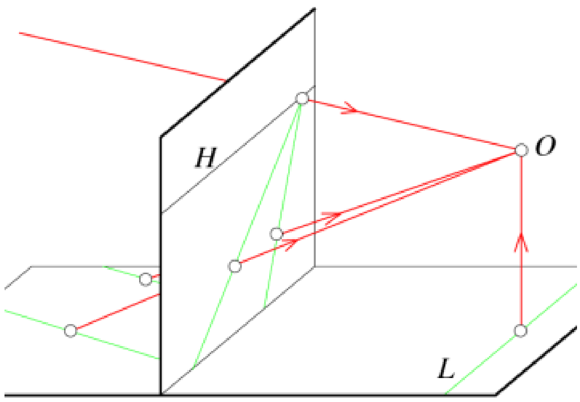
Camera Model

Structure from motion

Geometric Properties of Projection

- Points go to points
- Lines go to lines
- Planes go to whole image or half-plane
- Polygons go to polygons
- Degenerate cases:
  - line through focal point yields point
  - plane through focal point yields line



To present an image on screen, we need a way to project 3D objects onto a 2D plane. Usually we use a "camera" to do this. In reality, complicated lens are needed to converge rays to a single point. In computer world, we could use an ideal camera, i.e. pinhole camera.

Pinhole Camera

Notation

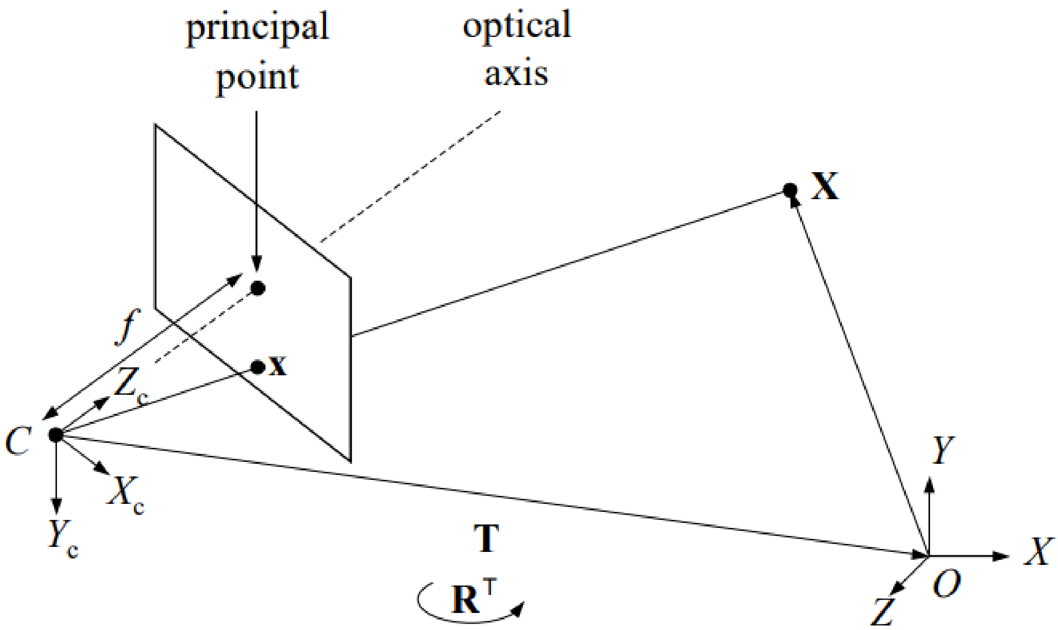
In Projective Geometry we introduced homogeneous coordinates. In this chapter, we denote a homogeneous 3D point $\tilde{\mathbf{X}} \sim [\tilde{X}, \tilde{Y}, \tilde{Z}, \tilde{W}]^\top$(where $\sim$ means equality up to scale). Provided $\tilde{W}$ is non-zero, $\tilde{\mathbf{X}}$ is related to its Euclidean equivalent $\mathbf{X} = [X, Y, Z]^\top$ by the following equations:

$$\mathbf{X} = [\tilde{X}/\tilde{W}, \tilde{Y}/\tilde{W}, \tilde{Z}/\tilde{W}]^\top \qquad \tilde{\mathbf{X}} \sim [X, Y, Z, 1]^\top$$

Similarly, a homogeneous 2D point $\tilde{\mathbf{x}} \sim [\tilde{x}, \tilde{y}, \tilde{z}]^\top$ is related to its Euclidean equivalent $\mathbf{x} = [x, y]^\top$

$$\mathbf{x} = [\tilde{x}/\tilde{w}, \tilde{y}/\tilde{w}]^\top \qquad \tilde{\mathbf{x}} \sim [x, y, 1]^\top$$

The Projective Matrix



The relationship between a 3D point and its corresponding 2D image point has three components, which are described below:

1. The first component is the rigid body transformation that relates points $\tilde{\mathbf{X}} \sim [X, Y, Z, 1]^\top$ in the world coordinate system to points $\tilde{\mathbf{X}}_c \sim [X_c, Y_c, Z_c, 1]^\top$ in the camera coordinate system:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \sim \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2. The second component is the 3D to 2D transformation that relates 3D points $\tilde{\mathbf{X}}_c \sim [X_c, Y_c, Z_c, 1]^\top$(in camera coordinate system) to 2D points $\tilde{\mathbf{x}} \sim [x, y, 1]^\top$ on the camera image plane. By using similar triangles, we obtain the following relation ship

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

where f is the focal length. Since changing the value of f corresponds simply to scaling the image, we can set $f = 1$ and account for the missing scale factor within the camera calibration matrix (below). Then, using homogenous coordinates, the relationship can be expressed by the following matrix equation

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}.$$

3. The final component is the 2D to 2D transformation that relates points $\tilde{\mathbf{x}}$ on the camera image plane to pixel coordinates $\tilde{\mathbf{u}} = [u, v, 1]^\top$. This is written as follows

$$\tilde{\mathbf{u}} = \mathbf{K}\tilde{\mathbf{x}}$$

where $\mathbf{K}$ is an upper triangular camera calibration matrix of the form:

$$K = \begin{bmatrix} \alpha_u & s & u_0 \\ & \alpha_v & v_0 \\ & & 1 \end{bmatrix}$$

and α_u and α_v are scale factors, s is *skewness*, and $\mathbf{u}_0 = [u_0, v_0]^\top$ is the principal point. These are camera intrinsic parameters. Usually, pixels are assumed to be square in which case $\alpha_u = \alpha_v = \alpha$ and $s = 0$. Hence, α can be considered to be the focal length of the lens expressed in units of the pixel dimension. We often say that an image is *skewed* when the camera coordinate system is skewed, meaning that the angle between the two axes is slightly larger or smaller than 90 degrees. Most cameras have zero-skew, but some degree of skewness may occur because of sensor manufacturing errors.

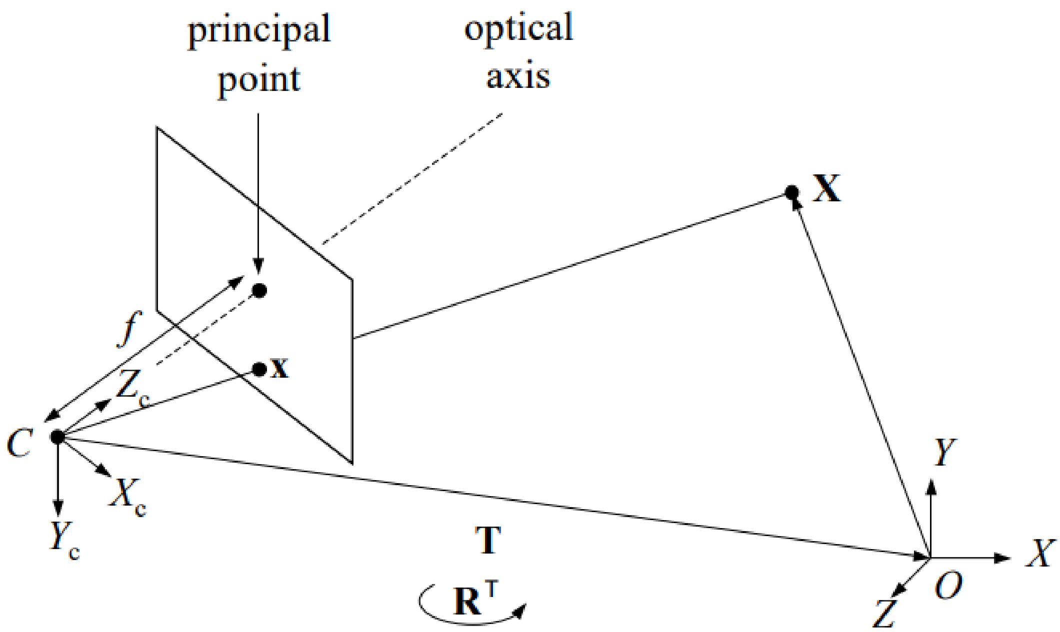
It's convenient to combine theses three components into a single linear transformation. Using homogeneous coordinates, a 3D point $\tilde{\mathbf{X}}$ is related to its pixel position $\tilde{\mathbf{u}}$ in a 2D image array by the following relationship:

$$\tilde{\mathbf{u}} \sim \mathbf{P}\tilde{\mathbf{X}}$$

where $\mathbf{P} \sim \mathbf{K}[\mathbf{R}|\mathbf{T}]$ is a 3×4 projection matrix.

These parameters $\mathbf{R}$ and $\mathbf{T}$ are known as the *extrinsic parameters* because they are external to and do not depend on the camera. All parameters contained in the camera matrix $\mathbf{K}$ are the *intrinsic parameters*, which change as the type of camera changes. In total, we have 11 Dof. We have in total 11 Dof. (5 in $\mathbf{K}$ and 6 in $\mathbf{R}|\mathbf{T}$)

Camera Calibration



Camera intrinsic and extrinsic parameters can be determined for a particular camera and lens combination by photographing a controlled scene.

Let $\tilde{\mathbf{u}}_i = [u_i, v_i, 1]^\top$ be the measured image position of $3D$ point $\tilde{\mathbf{X}}_i = [X_i, Y_i, Z_i, 1]^\top$. Write the projection matrix $\mathbf{P}$ as

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{bmatrix}$$

Then $\tilde{\mathbf{u}}_i \sim \mathbf{P}\tilde{\mathbf{X}}_i$ could be written as

$$\begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \sim \begin{pmatrix} \mathbf{p}_1^\top \tilde{\mathbf{X}}_i \\ \mathbf{p}_2^\top \tilde{\mathbf{X}}_i \\ \mathbf{p}_3^\top \tilde{\mathbf{X}}_i \end{pmatrix} \implies u_i = \frac{\mathbf{p}_1^\top \tilde{\mathbf{X}}_i}{\mathbf{p}_3^\top \tilde{\mathbf{X}}_i}, \quad v_i = \frac{\mathbf{p}_2^\top \tilde{\mathbf{X}}_i}{\mathbf{p}_3^\top \tilde{\mathbf{X}}_i}$$

which could be rearranged as

$$\begin{pmatrix} \tilde{\mathbf{X}}_i^\top & \mathbf{0}_{1\times 4} & -u_i \tilde{\mathbf{X}}_i^\top \\ \mathbf{0}_{1\times 4} & \tilde{\mathbf{X}}_i^\top & -v_i \tilde{\mathbf{X}}_i^\top \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \mathbf{0}_{12\times 1}$$

Denote

$$A_i = \begin{pmatrix} \tilde{\mathbf{X}}_i^\top & \mathbf{0}_{1\times 4} & -u_i \tilde{\mathbf{X}}_i^\top \\ \mathbf{0}_{1\times 4} & \tilde{\mathbf{X}}_i^\top & -v_i \tilde{\mathbf{X}}_i^\top \end{pmatrix}$$

and stack A_is together, we have a linear system:

$$\mathbf{A}\mathbf{p} = \mathbf{0}$$

where

$$\mathbf{A} = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \vdots \\ A_n \end{bmatrix} \in \mathbb{R}^{2n\times 12}$$

Explicitly, the equation is

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix} = \mathbf{0}$$

Since there are 11 unknowns (scale is arbitrary), we need to observe at least 6 3D points to recover the projection matrix and calibrate the camera.

The equations can be solved using orthogonal least squares. The linear least squares solution minimizes $\|\mathbf{A}\mathbf{p}\|$ subject to $\|\mathbf{p}\| = 1$ and is given by the unit eigenvector corresponding to the smallest eigenvalue of $\mathbf{A}^\top\mathbf{A}$. Numerically, this computation is performed via the Singular Value Decomposition of the matrix:

$$\mathbf{A} = \mathbf{U}\Lambda\mathbf{V}^\top$$

where $\Lambda = \mathrm{diag}(\sigma_1, \sigma_2, \dots, \sigma_{12})$ is the diagonal matrix of singular values and the matrices $\mathbf{U}$ and $\mathbf{V}$ are orthonormal. The columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{A}^\top\mathbf{A}$ and the required solution is the column of $\mathbf{V}$ corresponding the smallest singular value σ_{12}.

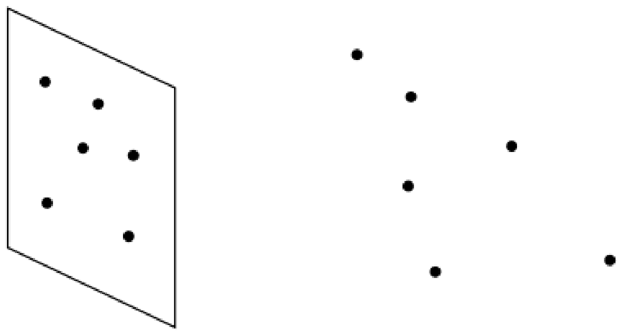
However, the least squares solution is only approximate and should be used as the starting point for non-linear optimization(not explained here).

Once the projection matrix has been estimated, the first 3×3 sub-matrix can be decomposed (by QR decomposition) into an upper triangular camera calibration matrix $\mathbf{K}$ and an orthonormal rotation matrix $\mathbf{R}$.

DLT

The above method is equivalent to direct linear transformation.

Given a set of correspondences $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$, we want to determine the projective matrix P.



Using DLT, we have

$$\mathbf{x}_i = \mathbf{P}\mathbf{X}_i$$

and denote

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{pmatrix}$$

Following the same way as before

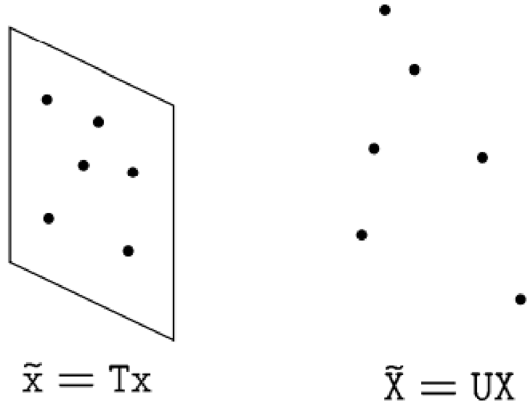
$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \begin{bmatrix} \mathbf{0}^\top & -w_i\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ w_i\mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \\ -y_i\mathbf{X}_i^\top & x_i\mathbf{X}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \mathbf{0}$$

Let

$$A_i = \begin{bmatrix} \mathbf{0}^\top & -w_i\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ w_i\mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \end{bmatrix}$$

and stack A_is to get A. Solve the system using the same way as in DLT.

Data Normalization



Here

$$\mathrm{T}=\begin{bmatrix}\sigma_{2D} & 0 & \bar{x}\\ 0 & \sigma_{2D} & \bar{y}\\ 0 & 0 & 1\end{bmatrix}^{-1}$$

and

$$\mathrm{U}=\begin{bmatrix}\sigma_{3D} & 0 & 0 & \bar{X}\\ 0 & \sigma_{3D} & 0 & \bar{Y}\\ 0 & 0 & \sigma_{3D} & \bar{Z}\\ 0 & 0 & 0 & 1\end{bmatrix}^{-1}$$

1. move center of mass to origin
2. scale to yield order 1 values

Gold Standard Algorithm
Objective:
Given $n \geq 6$ 2D to 3D point correspondences $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$, determine the Maximum Likelihood Estimation of P
Algorithm:
1. Linear solution
a. Normalization: $\tilde{\mathbf{X}}_i = U\mathbf{X}_I$, $\tilde{\mathbf{x}}_i = T\mathbf{x}_i$
b. DLT
2. Minimization of geometric error: using the linear estimate as a starting point minimize the geometric error.

$$\min_P \sum_i d(\tilde{\mathbf{x}}_i, \tilde{P}\tilde{\mathbf{X}}_i)$$

3. Denormalization: $P = T^{-1}\tilde{P}U$