**Camera calibration**

*Goal: 3D to 2D camera projection:* Given a set of points $X_i$ in 3D space, and a set of corresponding points $x_i$ in an image, find the 3D to 2D projective mapping that maps $X_i$ to $x_i$.

Once you know the camera projection, you can use it to correct distortions (for example lens distortion), to measure the size of an object in world units, or determine the location of the camera in the scene!

**Camera projection matrix**

$$ x = MX $$

where $X$ is a point in the 3D space, and $x$ is the projection on the image plane.

We will next see how the matrix $M$ is

**Intrinsic parameters - camera models**

The pinhole model

a. homogeneous coordinates:

![Homogeneous Coordinates Diagram]

$$ P = \text{diag}(f, f, 1) [I \mid 0] $$

b. Principal point offset:

$$ \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & p_x & 0 \\ f & p_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. $$

Therefore:
and thus:

\[
x = K [1 \ 0] X
\]

K is called the *camera calibration matrix*.

**The skew:**

The skew coefficient is non-zero if the image axes are not perpendicular:

\[
k \sim \tan(\alpha)
\]

\[
K = \begin{bmatrix}
f & s & p_x \\
0 & f_y & p_y \\
0 & 0 & 1
\end{bmatrix}
\]

K can be decomposed by:

\[
K \cong \begin{bmatrix}
1 & 0 & p_x \\
0 & 1 & p_y \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
1 & s/f_x & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
f_x & 0 & 0 \\
0 & f_y & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

**Demo:** [http://ksimek.github.io/perspective_camera_toy.html](http://ksimek.github.io/perspective_camera_toy.html)
Extrinsic parameters: Camera rotation and translation

Moving to the camera coordinate system:
It is often convenient not to make the camera centre explicit, and instead to represent the world to image transformation as

$$x_{\text{cam}} = RX + t$$

In this case the camera matrix is simply

$$M = K[R | t]$$

Demo: [http://ksimek.github.io/perspective_camera_toy.html](http://ksimek.github.io/perspective_camera_toy.html)
Calibration:

Goal: estimate the intrinsic and extrinsic parameters from one or multiple images

Input: \( n \) points in the real world \( X_1, \ldots, X_n \) with known coordinates and known positions in the image \( x_1, \ldots, x_n \)

Main Steps:

1. **Calculating the Matrix** \( P \) s.t. for all the given points:
   \[
   x = MX
   \]

2. **Extract the intrinsic and extrinsic parameters** from \( P \) by decomposition:
   \[
   M = K[R | t]
   \]

Solution:

1. **Calculating the Matrix** \( P \) is done by the Direct linear Transform (DLT) algorithm:
   a. We have the linear system:
      \[
      \begin{pmatrix}
      \tilde{x} \\
      \tilde{y} \\
      \tilde{w}
      \end{pmatrix} = K \begin{pmatrix}
      R & t
      \end{pmatrix} \begin{pmatrix}
      X \\
      Y \\
      Z \\
      1
      \end{pmatrix}
      
      \text{with } M \in \mathbb{R}^{3 \times 4}
      
      M = \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}
      
      \]
   b. Let’s write the 3 equations:
      \[
      \tilde{x} = m_{11}X + m_{12}Y + m_{13}Z + m_{14} \quad \tilde{y} = m_{21}X + m_{22}Y + m_{23}Z + m_{24} \quad \tilde{w} = m_{31}X + m_{32}Y + m_{33}Z + m_{34}
      \]
   c. Getting 2D point:
      \[
      x = \frac{\tilde{x}}{\tilde{w}} = \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \quad y = \frac{\tilde{y}}{\tilde{w}} = \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}
      \]
      \[
      (m_{31}X + m_{32}Y + m_{33}Z + m_{34})x = m_{11}X + m_{12}Y + m_{13}Z + m_{14}
      \]
      \[
      (m_{31}X + m_{32}Y + m_{33}Z + m_{34})y = m_{21}X + m_{22}Y + m_{23}Z + m_{24}
      \]
   d. Matrix form:
      \[
      \begin{bmatrix}
      X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ & -x \\
      0 & 0 & 0 & 0 & X & Y & Z & 1 & -yX & -yY & -yZ & -y
      \end{bmatrix} m = 0
      
      m = \begin{pmatrix}
      m_{11} \\
      m_{12} \\
      m_{13} \\
      \vdots \\
      m_{34}
      \end{pmatrix} \in \mathbb{R}^{12}
      
      \]
We got that each 2D/3D correspondence point generates two equations.

**Concatenate** equations for n correspondences:

\[ Am = 0 \]

How many pairs of corresponding points do we need? 11 unknowns \( \Rightarrow \) 11 equations \( \Rightarrow \) 6 points

In practice, we will take more. Why?

- We have homogenous equations, solution is in the null space of A (i.e., eigenvector corresponding to zero eigenvalue of A)
- In practice – observations are noisy: solution using SVD: the right singular vector with the lowest singular value (on board)

2. **Decompose M into K,R,t**
   a. Construct M from m:

\[
\begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ \vdots \\ m_{34} \end{pmatrix} \in \mathbb{R}^{12} \quad \Rightarrow \quad M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}
\]

b. Key observation: the first 3x3 matrix, , is a product of upper triangular (K) and orthogonal (R) matrices Therefore, we can use QR factorization ([https://en.wikipedia.org/wiki/QR_decomposition](https://en.wikipedia.org/wiki/QR_decomposition)) (decomposition of a matrix A into a product \( A = QT \) of an orthogonal matrix Q and an upper triangular matrix T).

c. Now we can construct t as follows:

\[
t = K^{-1} \begin{pmatrix} m_{14} \\ m_{24} \\ m_{34} \end{pmatrix}
\]
**Theorem 2.** If $A$ is an $n \times n$ matrix then there is an orthogonal matrix $Q$ and a right triangular matrix $R$ such that

$$A = RQ.$$  \hspace{1cm} (35)

(If $A$ is invertible and the diagonal elements are chosen positive then the factorization is unique.)

In order to be consistent with the notation in the rest of the lecture we will use $K$ for the right triangular matrix and $R$ for the orthogonal matrix. Given a camera matrix $P = [A \ a]$ we want to use $RQ$-factorization to find $K$ and $R$ such that $A = KR$. If

$$K = \begin{pmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{pmatrix}, \quad A = \begin{bmatrix} A_1^T \\ A_2^T \\ A_3^T \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} R_1^T \\ R_2^T \\ R_3^T \end{bmatrix},$$  \hspace{1cm} (36)

that is, $R_1, R_2, R_3$ and $A_1, A_2, A_3$ are $3 \times 1$ vectors containing the rows of $R$ and $A$ respectively, then we get

$$\begin{bmatrix} A_1^T \\ A_2^T \\ A_3^T \end{bmatrix} = \begin{pmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{pmatrix} \begin{bmatrix} R_1^T \\ R_2^T \\ R_3^T \end{bmatrix} = \begin{bmatrix} aR_1^T + bR_2^T + cR_3^T \\ dR_2^T + eR_3^T \\ fR_3^T \end{bmatrix}.$$  \hspace{1cm} (37)

From the third row of (37) we see that $A_3 = fR_3$. Since the matrix $R$ is orthogonal $R_3$ has to have the length 1. We therefore see that need to select

$$f = \|A_3\| \quad \text{and} \quad R_3 = \frac{1}{\|A_3\|}A_3.$$  \hspace{1cm} (38)

to get a positive coefficient $f$. When $R_3$ is known we can proceed to the second row of (37). The equation $A_2 = dR_2 + eR_3$ tells us that $A_2$ is a linear combination of two orthogonal vectors (both of length one). Hence, the coefficient $e$ can be computed from the scalar product

$$e = A_2^T R_3.$$  \hspace{1cm} (39)

When $e$ is known we can compute $R_2$ and $d$ from

$$dR_2 = A_2 - eR_3,$$  \hspace{1cm} (40)

similar to what we did for $f$ and $R_3$ in (38). When $R_2$ and $R_3$ is known we use the first row of (37)

$$A_1 = aR_1 + bR_2 + cR_3$$  \hspace{1cm} (41)

to compute $b$ and $c$. Finally we can compute $a$ and $R_1$ from

$$A_1 - bR_2 - cR_3 = aR_1.$$  \hspace{1cm} (42)

The resulting matrix $K$ is not necessarily of the form $[A \ a]$ since element $(3, 3)$ might not be one. To determine the individual parameters, focal length, principal point etc. we therefore need to divide the matrix with element $(3, 3)$. Note however, that this does not modify the camera in any way since the scale is arbitrary.
Question 1 – Camera Calibration: (Exam A - Winter 2017)

The relationship between a 3D point at world coordinates (X,Y,Z) and its corresponding 2D pixel at image coordinates (u,v) can be defined as a projective transformation using a $3 \times 4$ camera projection matrix $P$.

a. Give the main steps of an algorithm for computing the matrix $P$ from a single image of a known 3D “calibration object.”

**Note:** Please write down the entire mathematical operation. (6 points)

**Answer:**

Let $P_1, P_2, P_3$ be the three rows of the projection matrix and let $P_{123}$ be a 12-vector of their concatenation. Every image coordinate, e.g. $u$, and the corresponding 3D coordinates $X$ (4 vector) specify a constraint $P_1^T X u = P_3^T X$, which is linear in the components of $P_{123}$. The constraints for $m$ points may be written as a matrix equation $A P_{123} = 0$ where every line of the $(2m \times 12)$ matrix $A$ contains the linear constraints coefficient associated with one image coordinate. With reasonable large $m$, the homogeneous system does not have a solution and a good approximation is the right singular vector with the lowest singular value (found using SVD). For more details, see to the camera calibration tutorial.

b. Consider the option of down-sampling each dimension of the image by 4. (6 points)

1. Does it affect the intrinsic camera matrix? If so explain how.
2. Can this operation be replaced by a zoom operation? If so explain why.

**Answer:**

1. It changes the intrinsic camera matrix by scaling $Sx, Sy$ by 4 and also by changing the principal point (note that the sensor location relative to the camera does not change, but the unit by which we measure the principal point, do.
2. Multiplying $f$ by .25 compensates for the scaling but not for the principal point changes. Therefore, zoom operation cannot replace the scaling.

c. Consider a point $(u', v')$ in the image plane of a camera characterized by known intrinsic and extrinsic projection matrices. Give an expression (not necessarily explicit) for the set of 3D points projected to $(u', v')$ in world coordinate system and in the camera coordinate system. (7 points)

**Answer:**

First, consider the case of the world coordinate system. Let $K = M_{int}$, $M_{ext}$ be the intrinsic and extrinsic projection matrices respectively. The full projection matrix from world coordinates to image coordinates is $P_w = K M_{ext}$. Denote the rows of this projection matrix $P1, P2, P3$. All of them are 4-vectors. We know that $u' = P1^T X / P3^T X$, $v' = P2^T X / P3^T X$. This are two linear constraints on the homogeneous coordinates vector $X$, which (together) limit it to a line.

Consider now the case of camera coordinate system. The extrinsic projection matrix in the camera coordinate system simply $M_{ext} = [I \mid 0]$. The rest is the same.