Computer Vision: Lecture 3

Carl Olsson

2020-01-28
Camera Calibration

- Repetition: The camera equations.
- Repetition: Structure from motion.
- Projective vs. Euclidean Reconstruction.
- The inner parameters - $K$.
- Finding the camera matrix.
- DLT - Direct Linear Transformation.
- Normalization of uncalibrated cameras.
- Radial distortion
The camera equations:

\[ \lambda \mathbf{x} = K \begin{bmatrix} R & t \end{bmatrix} \mathbf{X} = P \]

- **K** - intrinsic parameters
- **R, t** - extrinsic parameters

**Diagram:**
- Coordinate systems: \( \mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z \) and \( \mathbf{e}_x', \mathbf{e}_y', \mathbf{e}_z' \)
- Camera projection matrix \( P \)
- Points \( \mathbf{x}, \mathbf{X}, \mathbf{x}' \)
The structure from motion problem:

Given Images

Compute 3D Model

4 images out of a sequence with 435 images.
Repetition

The structure from motion problem (main goal of the course):
Solve the camera equations:

\[ \lambda_{ij} \mathbf{x}_{ij} = P_i \mathbf{x}_j, \quad \forall i, j. \]

Find both camera matrices \( P_i \), and 3D points \( \mathbf{x}_i \)!

Two versions:
- Projective reconstruction: Nothing is known about \( P_i \).
- Euclidean reconstruction: \( P_i = K_i \begin{bmatrix} R_i & t_i \end{bmatrix} \), where \( K_i \) is known!
The reconstruction is determined up to a projective transformation. If $\lambda x = PX$, then for any projective transformation

$$\tilde{X} = H^{-1}X$$

we have

$$\lambda x = PHH^{-1}X = PH\tilde{X}.$$ 

$PH$ is also a valid camera.
A camera

\[ P = K [R \ t], \]

where the inner parameters \( K \) are known is called calibrated. If we change coordinates in the image using

\[ \tilde{x} = K^{-1}x, \]

we get a so called normalized (calibrated) camera

\[ \tilde{x} = K^{-1}K [R \ t]X = [R \ t]X. \]
Euclidean

The reconstruction is determined up to a similarity transformation. If \( \lambda \mathbf{x} = [R \ t] \mathbf{X} \), then for any similarity transformation

\[
\tilde{\mathbf{X}} = H^{-1} \mathbf{X} = \begin{bmatrix} sQ & v \\ 0 & 1 \end{bmatrix}^{-1} \mathbf{X}
\]

we have

\[
\frac{\lambda}{s} \mathbf{x} = [R \ t] \begin{bmatrix} Q & v \\ 0 & \frac{1}{s} \end{bmatrix} \tilde{\mathbf{X}} = [RQ \ Rv + \frac{t}{s}] \tilde{\mathbf{X}}.
\]

Since \( RQ \) is a rotation this is a normalized camera.
Arch of triumph, Paris. The reconstructions have exactly the same reprojection error. But the projective coordinate system makes things look strange.
Demo.
The matrix $K$ is the upper triangular matrix:

$$K = \begin{bmatrix} \gamma f & sf & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}.$$ 

- $f$ - focal length
- $\gamma$ - aspect ratio
- $s$ - skew
- $(x_0, y_0)$ - principal points
The Inner Parameters \(-K\)

The focal length \( f \)

\[
\begin{pmatrix}
fx \\
fy \\
1
\end{pmatrix} = \begin{bmatrix}
f & 0 & 0 \\
0 & f & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{pmatrix}
x \\
y \\
1
\end{pmatrix}
\]

Re scales the images (e.g. meters \(\rightarrow\) pixels).
The Inner Parameters - $K$

The principal point $(x_0, y_0)$

\[
\begin{pmatrix}
fx + x_0 \\
fy + y_0 \\
1
\end{pmatrix} =
\begin{bmatrix}
f & 0 & x_0 \\
0 & f & y_0 \\
0 & 0 & 1
\end{bmatrix}
\begin{pmatrix}
x \\
y \\
1
\end{pmatrix}
\]

Re centers the image. Typically transforms the point $(0, 0, 1)$ to the middle of the image.
The Inner Parameters - $K$

**Aspect ratio**

\[
\begin{pmatrix}
\gamma f x + x_0 \\
fy + y_0 \\
1
\end{pmatrix}
= 
\begin{bmatrix}
\gamma f & 0 & x_0 \\
0 & f & y_0 \\
0 & 0 & 1
\end{bmatrix}
\begin{pmatrix}
x \\
y \\
1
\end{pmatrix}
\]

Pixels are not always squares but can be rectangular. In such cases the scaling in the $x$-direction should be different from the $y$-direction.

**Skew**

\[
\begin{bmatrix}
\gamma f & sf & x_0 \\
0 & f & y_0 \\
0 & 0 & 1
\end{bmatrix}
\]

Corrects for tilted pixels. Typically zero.
Finding K

1. Solve the **resection** problem: Find $P$ from the camera equations

   $$\lambda_i x_i = PX_i,$$

   when both $x_i$ and $X_i$ are known (for all $i$).
   (Structure form motion with known 3D points.)

2. Use **RQ-factorization** to extract $K$ from $P$. 
Theorem

If $A$ is an $n \times n$ matrix then there is an orthogonal matrix $Q$ and a right triangular matrix $R$ such that $A = RQ$.

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

Note: In our case we will use $K$ for the triangular matrix and $R$ for the rotation.
Finding K

See lecture notes.
Finding K

Exercise 1: If

\[
P = \begin{pmatrix}
3000 & 0 & -1000 & 1 \\
1000 & 2000\sqrt{2} & 1000 & 2 \\
2 & 0 & 2 & 3
\end{pmatrix}
\]

find \( f \) and \( R_3 \).

Exercise 2: Determine \( e \), \( R_2 \) and \( d \) for \( P \) in Ex 1.

Exercise 3: Determine \( a, b, c \) and \( R_1 \) for \( P \) in Ex 1.
Finding the camera matrix (The Resection Problem)

Use images of a known object to eliminate the projective ambiguity. If $X_i$ are 3d-points of a known object, and $x_i$ corresponding projections we have

$$\lambda_1 x_1 = P X_1 \quad \lambda_2 x_2 = P X_2 \quad \vdots \quad \lambda_N x_N = P X_N.$$ 

There are $3N$ equations and $11 + N$ unknowns. We need $3N \geq 11 + N \Rightarrow N \geq 6$ points to solve the problem.
Matrix Formulation

\[ P = \begin{bmatrix} p_1^T \\ p_2^T \\ p_3^T \end{bmatrix} \]

where \( p_i \) are the rows of \( P \). The first equality is

\[
\begin{align*}
X_1^T p_1 - \lambda_1 x_1 &= 0 \\
X_1^T p_2 - \lambda_1 y_1 &= 0 \\
X_1^T p_3 - \lambda_1 &= 0,
\end{align*}
\]

where \( x_1 = (x_1, y_1, 1) \). In matrix form

\[
\begin{bmatrix}
X_1^T & 0 & 0 & -x_1 \\
0 & X_1^T & 0 & -y_1 \\
0 & 0 & X_1^T & -1
\end{bmatrix}
\begin{pmatrix}
p_1 \\
p_2 \\
p_3 \\
\lambda_1
\end{pmatrix}
= \begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}
\]
Matrix Formulation

More equations:

\[
\begin{bmatrix}
X_1^T & 0 & 0 & -x_1 & 0 & 0 & \ldots \\
0 & X_1^T & 0 & -y_1 & 0 & 0 & \ldots \\
0 & 0 & X_1^T & -1 & 0 & 0 & \ldots \\
X_2^T & 0 & 0 & 0 & -x_2 & 0 & \ldots \\
0 & X_2^T & 0 & 0 & -y_2 & 0 & \ldots \\
0 & 0 & X_2^T & 0 & -1 & 0 & \ldots \\
X_3^T & 0 & 0 & 0 & 0 & -x_3 & \ldots \\
0 & X_3^T & 0 & 0 & 0 & -y_3 & \ldots \\
0 & 0 & X_3^T & 0 & 0 & -1 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\begin{bmatrix}
p_1 \\
p_2 \\
p_3 \\
\lambda_1 \\
\lambda_2 \\
\lambda_3 \\
\vdots
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]
Homogeneous Least Squares

See lecture notes...
Singular values decomposition

**Theorem**

Each $m \times n$ matrix $M$ (with real coefficients) can be factorized into

$$M = USV^T,$$

where $U$ and $V$ are orthogonal ($m \times m$ and $n \times n$ respectively),

$$S = \begin{bmatrix} \text{diag}(\sigma_1, \sigma_2, ..., \sigma_r) & 0 \\ 0 & 0 \end{bmatrix},$$

$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0$ and $r$ is the rank of the matrix.

Very useful tool in this course!
Homogeneous Least Squares

See lecture notes...
Exercise 4

If $M = USV^T$ where

$$
\begin{pmatrix}
3 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 1
\end{pmatrix}
\quad \text{and} \quad
V =
\begin{pmatrix}
1 & 0 & 0 \\
0 & \frac{1}{2} & -\frac{\sqrt{3}}{2} \\
0 & \frac{\sqrt{3}}{2} & \frac{1}{2}
\end{pmatrix}
$$

find a vector $v$ of length 1 that minimizes $\|Mv\|$ and the minimal value.
Algorithm for minimizing $\|Mv\|^2$ with $\|v\| = 1$:

1. Compute the factorization

\[
M = USV^T
\]

(in Matlab).

2. Select the solution

\[v = \text{last column of } V.\]
Improving the Numerics (Normalization of uncalibrated cameras)

The matrix contains entries \(x_i, y_j\) and ones. Since \(x_i\) and \(y_i\) can be about a thousand, the numerics are often greatly improved by translating the coordinates such that their “center of mass” is zero and then rescaling the coordinates to be roughly 1.

- Change coordinates according to

\[
\tilde{x} = \begin{bmatrix}
    s & 0 & -s\bar{x} \\
    0 & s & -s\bar{y} \\
    0 & 0 & 1
\end{bmatrix}x.
\]

- Solve the homogeneous linear least squares system and transform back to the original coordinate system.

- Similar transformations for the 3D-points \(X_i\) may also improve the results.
Pose estimation using DLT

3D points measured using scanning arm.
14 points used for computing the camera matrix.
14 points used for computing the camera matrix.
Texturing the chair

Project the rest of the points into the image.
Texturing the chair

Form triangles. Use the texture from the image.
Textured chair
Radial Distortion

Not modeled by the $K$-matrix. Cannot be removed by a projective mapping since lines are not mapped onto lines (see Szeliski).
Todo

- Finish Assignment 1.
- Start working on Assignment 2.
  Theory for E1, CE1, E2, E3, E4, E5, CE2 is done.