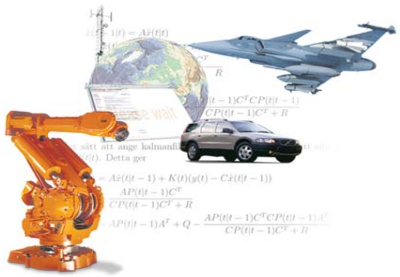
Lecture 2 – Camera Models and Calibration



Thomas Schön,
Division of Automatic Control,
Department of Electrical Engineering,
Linköping University.

Email: schon@isy.liu.se

Camera – A device that provides 2D projections of the 3D world $x = \frac{X}{Z}, \; y = \frac{X}{Z}$

Content – Lecture 2

1. Summary of Lecture 1
2. Image representation
3. **Geometric camera models**
   $$p_p = (\mathcal{K} \circ \mathcal{D} \circ \mathcal{P}_n \circ \mathcal{R})\,(P_w)$$
   a. Extrinsic camera parameters $P_c = \mathcal{R}(P_w)$
   b. Normalized pinhole model $p_n = \mathcal{P}_n(P_c)$
   c. Lens distortion $p_d = \mathcal{D}(P_n)$
   d. Intrinsic camera parameters $p_p = \mathcal{K}(p_d)$
4. **Camera calibration** (gray-box sys.id. problem)
   a. Initial parameters
   b. Maximum likelihood

Summary – Lecture 1 (Rotations SO(3))

The Special Orthogonal group:

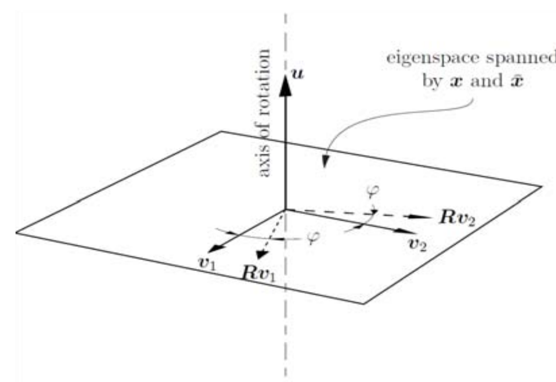
$$SO(3) \triangleq \left\{ R \in \mathbb{R}^{3\times 3} \,|\, R^T R = I, \det(R) = +1 \right\}$$

Commonly used parameterizations of SO(3):

1. Rotation matrices
2. Unit quaternions
3. Euler angles
4. Exponential coordinates
5. Axis/angle

Summary – Lecture 1 (Rotations SO(3))

Theorem: (Euler) Any orientation $R \in SO(3)$ is equivalent to a rotation about a fixed axis $u \in \mathbb{R}^3$ through an angle $\varphi \in [0, 2\pi)$

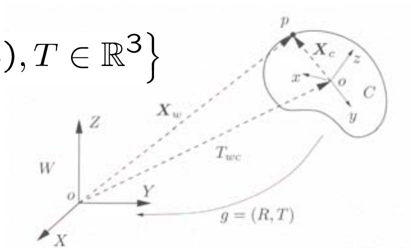


Summary – Lecture 1 (SE(3))

Definition: (rigid body motion / special Euclidean transformation) A mapping $g : \mathbb{R}^3 \to \mathbb{R}^3$ is a rigid body motion / special Euclidean transformation if it satisfies the following properties:

1. Length is preserved: $\|g(p) - g(q)\| = \|p - q\|$ for all points $p, q \in \mathbb{R}^3$
2. The cross product is preserved: $g(v \times w) = g(v) \times g(w)$ for all vectors $v, w \in \mathbb{R}^3$

$$SE(3) \triangleq \left\{ g = (R, T) | R \in SO(3), T \in \mathbb{R}^3 \right\}$$



Summary – Lecture 1 (SE(3) and homogeneous coord.)

Homogeneous coordinates are obtained by augmenting the Euclidean coordinates with an additional 1.

$$X^w = R^{wc} X^c + T^{wc}$$

$$\bar{X}^w = \begin{pmatrix} X^w \\ 1 \end{pmatrix} = \begin{pmatrix} R^{wc} & T^{wc} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X^c \\ 1 \end{pmatrix} = \bar{g}^{wc} \bar{X}^c$$

$$\boldsymbol{SE(3) \triangleq \left\{ \bar{g} = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \middle| R \in SO(3), T \in \mathbb{R}^3 \right\}}$$

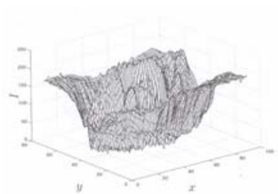
Theorem: (Chasles) Every rigid body motion can be realized by a rotation about an axis combined with a translation about that axis.

Image Representation

Common example for illustration

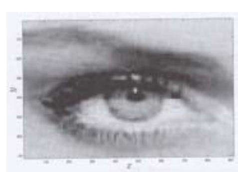
$$I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}_+ \qquad \Omega = [1, 640] \times [1, 480] \subset \mathbb{Z}^2$$
$$\mathbb{R}_+ \approx [0, 255] \subset \mathbb{Z}_+$$

Three different representations



1. The graph of I

2. Matrix of integers



3. A "picture" of the image

Geometric Camera Models

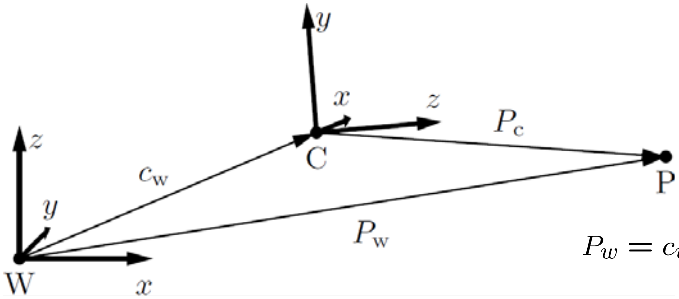
"A camera is a device that produce 2D projections of the 3D world"

Geometric Camera Models

Coordinate frames

World (w): This is considered an inertial frame and it is typically attached to a real object in the scene (hence another name is object frame).

Camera (c): The camera frame is fixed to the moving camera.



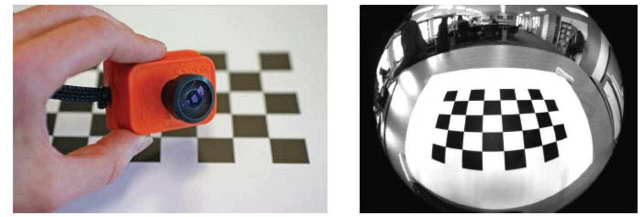
$$P_w = c_w + R_{wc} P_c$$

Geometric Camera Models – Different Lenses

Standard perspective lens

(a) (b)

Fish-eye lens

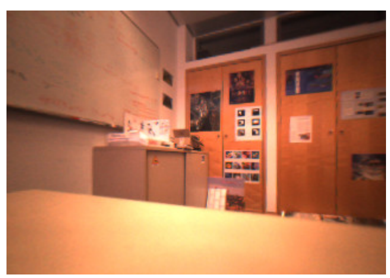


(c) (d)

Geometric Camera Models – Radial Lens Distortion

Distorted image

Undistorted image



Distorted image (obtained directly from the camera)

Compensate for the radial distortion

Undistorted image (as if it was generated by a pinhole camera)

Geometric Camera Models – Radial and Tangential Distortion

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = \underbrace{(1 + a_1 r^2 + a_2 r^4 + a_3 r^6)\begin{pmatrix} x_n \\ y_n \end{pmatrix}}_{\text{Radial distortion}} + \underbrace{\begin{pmatrix} 2a_4 x_n y_n + a_5(r^2 + 2x_n^2) \\ a_4(r^2 + 2y_n^2) + 2a_5 x_n y_n \end{pmatrix}}_{\text{Tangential distortion}}$$

The tangential distortion is due to imperfect centering (“decentering”) of the lens components and other manufacturing defects in a compound lens.

Historical Notes,

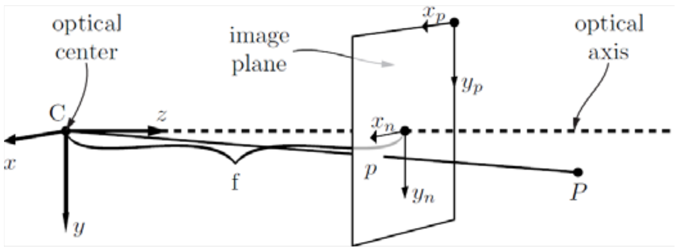
D. C. Brown, **Decentering distortion of lenses**, *Photometric Engineering*, 32(3): 444-462, 1966.

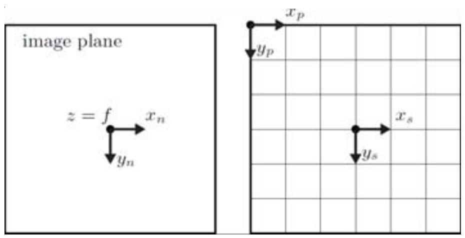
One of the first introduction of the tangential distortion model. This distortion model is also known as the "Brown-Conrady model".

A. Conrady, **Decentering lens systems**, *Monthly notices of the Royal Astronomical Society*, 79:384-390, 1919.

The very first introduction of the decentering distortion model.

Gemoetric Camera Models – Intrinsic Parameters





Lecture 2

Geometric Camera Models

Using homogeneous coordinates and normalized pinhole projection $\mathcal{D} = I$ we have

$$\lambda \underbrace{\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}}_{p_p} = \underbrace{\begin{pmatrix} f s_x & f s_\theta & o_x \\ 0 & f s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R_{cw} & w_c \\ 0 & 1 \end{pmatrix}}_{\Pi} \underbrace{\begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}}_{P_w}$$

$$\lambda p_p = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{pmatrix} P_w$$

Note that the model is nonlinear in Euclidean space

$$x_p = \frac{\pi_1 P_w}{\pi_3 P_w}, \qquad y_p = \frac{\pi_2 P_w}{\pi_3 P_w}$$

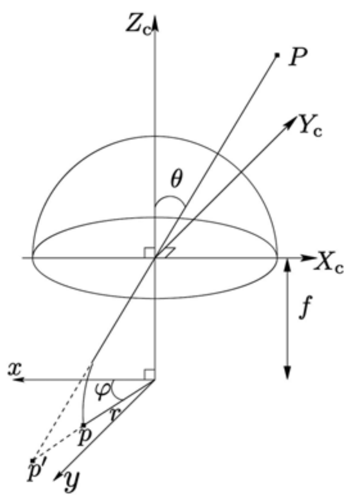
Lecture 2

Geometric Camera Models – Fish-Eye Lens

A fish-eye lens covers the whole hemispherical field in front of the camera and the angle of view is very large, about 180.

The spherical projection model is different from the pinhole model, for a good introduction, see

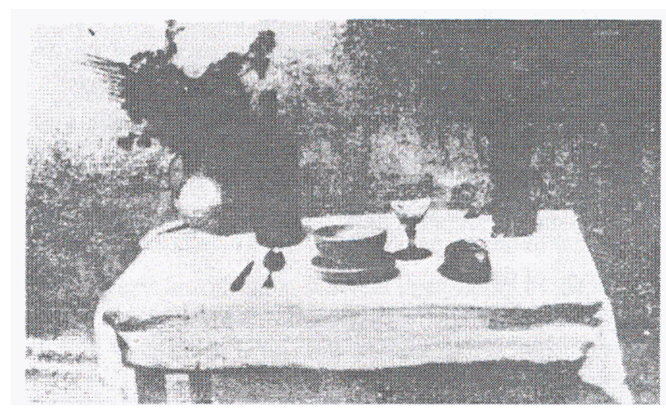
J. Kannala, S. S. Brandt, **A generic camera model and calibration for conventional, wide-angle and fish-eye lenses**, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8): 1335-1340, Aug. 2006.



Lecture 2

History – First Photograph on record

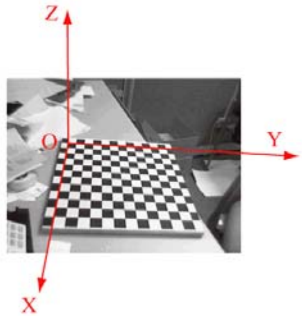
By Nicéphore Niepce in 1822



The set table (la table service)

Lecture 2

Camera Calibration – Idea



Without loss of generality we can choose the world reference frame to be aligned with checkerboard,

$$P_w = \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix}$$

Part of the course literature

Calibration for standard perspective lenses:

Z. Zhang, **A flexible new technique for camera calibration**, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11): 1330-1334, Nov. 2000.

Also taking care of wide-angle and fish-eye lenses:

J. Kannala, S. S. Brandt, **A generic camera model and calibration for conventional, wide-angle and fish-eye lenses**, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8): 1335-1340, Aug. 2006.

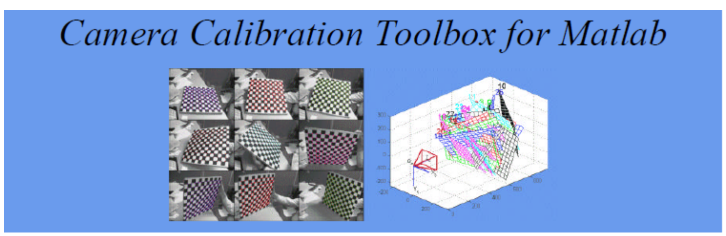
Camera Calibration – Procedure

1. Print a checkerboard pattern and attach it to a planar surface.
2. Acquire a few images of the checkerboard pattern under different poses, either by moving the camera or the pattern.
3. Detect the corners in the images. This provides a set of 2D/3D correspondences p_p^{ij}, P_w^i for each image j.
4. Obtain an initial estimate of the intrinsic parameters and all the extrinsic parameters.
5. Solve a maximum likelihood problem to obtain the intrinsic parameters, all the extrinsic parameters and the lens distortion parameters.

Camera Calibration - Software

There is very good software freely available on the Internet!

1. Caltech camera calibration toolbox



Just google "camera calibration toolbox" or use
http://www.vision.caltech.edu/bouguetj/calib_doc/

2. OpenCV is a computer vision library originally developed by Intel, now available on sourceforge.net. Free for commercial and research use under BSD license. Contains much more than calibration!

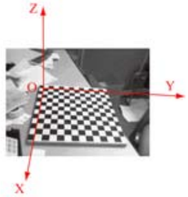
Camera Calibration

When a camera is calibrated it is convenient to preprocess the images and work with the normalized image coordinates p_n instead.

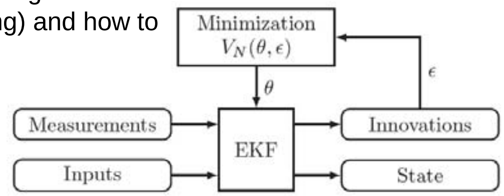
This implies that the camera measurements are decoupled from the intrinsic and the distortion parameters.

Useful for assembling estimation problems including images as we will see later in the course.

$$\hat{\theta} = \arg\min_{\theta} \sum_{t=1}^{N} (y_t - \hat{y}_t(\theta))^T \Lambda_t^{-1} (y_t - \hat{y}_t(\theta)),$$



- Use a movie as input, not a couple of images.
- The parameters only include the intrinsic parameters and the lens distortion, NOT the pose.
- The pose is obtained by solving a filtering problem.
- This project is a very good way of getting used to how cameras work (mathematically speaking) and how to formulate estimation problems.



More details are available on the course web site.