Omni-Geometry

Luiz Velho
IMPA

Outlook

- Omnidirectional Calibration
- Omnidirectional Geometry
- Omnidirectional 3D Reconstruction

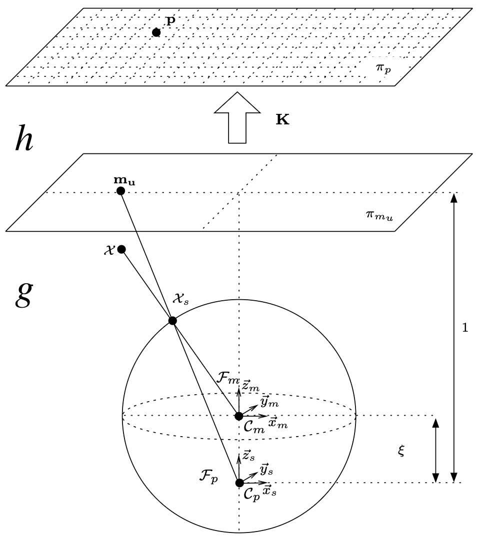
Omnidirectional Calibration

Based on (Micusik, 2004)

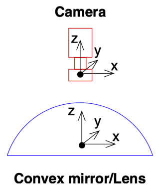
Calibration Fundamentals

General Unified Projection Model

- Image Formation

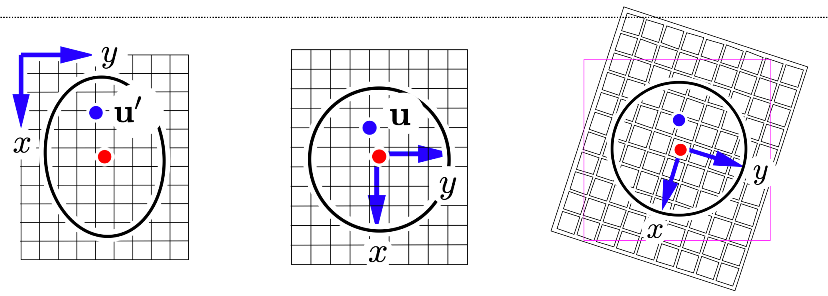


Axis Convention

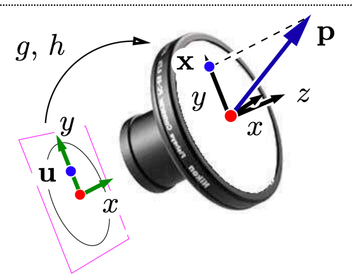


Overview

- Mapping from Digital Image to 3D Rays (*inverse of previous lecture*)



Digital Image *Normalized Image* *Sensor*



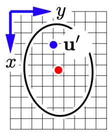
3D Rays

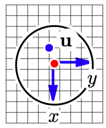
- Two Steps:
  1. Mapping from Image to Sensor (*digitization*)
  2. Mapping form Sensor to Scene Rays (*optics*)

Step 1 (pre-calibration)

- Map View Ellipse to Circle

$$\mathbf{u} = \mathtt{A}\,\mathbf{u}' + \mathbf{t}$$



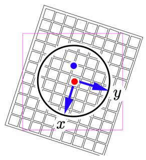


Ambiguities:

- Radius
- Rotation

- Map Circle to Sensor

$$\mathbf{t} = \frac{1}{\rho}\mathtt{R}^{-1}\mathbf{t}',$$
$$\mathtt{A} = \frac{1}{\rho}\mathtt{R}^{-1}\mathtt{A}'.$$



Unknowns:

Scale Factor: ρ

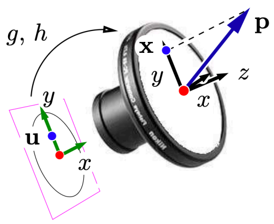
Rotation: R

$$\begin{matrix}\mathbf{u}'' = \mathtt{A}'\,\mathbf{u}' + \mathbf{t}' \\ \mathbf{u} = \mathtt{A}\,\mathbf{u}' + \mathbf{t}\end{matrix} \quad \Rightarrow \quad \mathbf{u}'' = \rho\,\mathtt{R}\,\mathbf{u}.$$

Step 2 (calibration)

Maps point u in pre-calibrated image to vector p

$$\exists m\colon \mathbb{R}\times\mathbb{R}^N \to \mathbb{R}^N, \exists i\colon \mathbb{R}\to\mathbb{R}, \forall \mathbf{u}\in\mathbb{R}^2, \forall\rho\in\mathbb{R}, \forall \mathbf{a}''\in\mathbb{R}^N:$$
$$h(\|\rho\mathbf{u}\|, \mathbf{a}'') = i(\rho)h\Big(\|\mathbf{u}\|, m(\rho,\mathbf{a}'')\Big),$$
$$g(\|\rho\mathbf{u}\|, \mathbf{a}'') = \rho\, i(\rho) f\Big(\|\mathbf{u}\|, m(\rho,\mathbf{a}'')\Big),$$



$u \to p$

- functions g, h model a concrete lens or a mirror
- function $m(\,\cdot\,)$ absorbs the scale ρ into vector a with the same number of elements as a''
- function $i(\,\cdot\,)$ absorbs the scale ρ arised by taking it out from functions g, h

The calibration method finds $m(\,\cdot\,)$, $i(\,\cdot\,)$ and estimates the parameters a

A metrically calibrated camera is obtained

Visualization

- Calibration on a Real Image

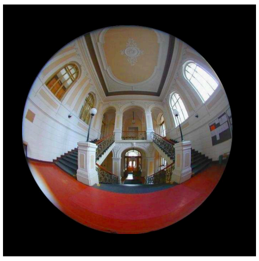
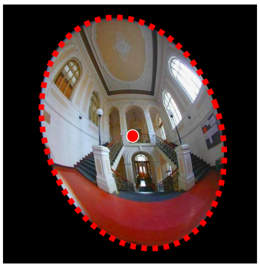
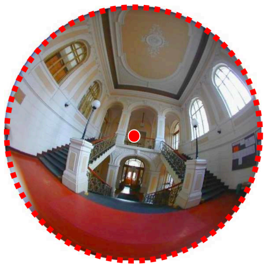


image on sensor plane



digital image with ellipse fitted on view field



transformed circular pre-calibrated image

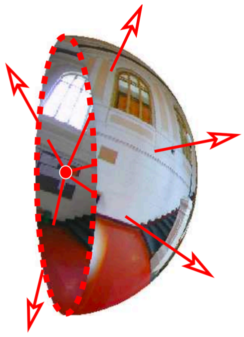


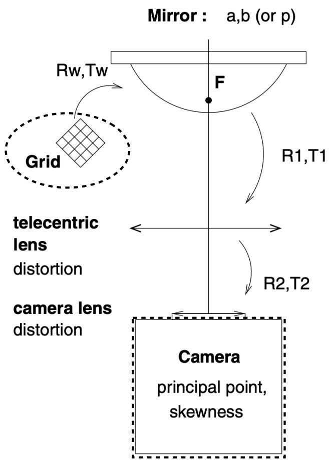
image on a sphere

Omni Camera Calibration from Planar Grids

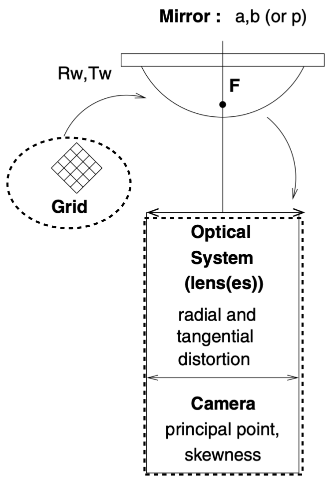
(Mei and Rives, 2007)

Calibration Parameters

- Complete Parameters



- Simplified Parameters

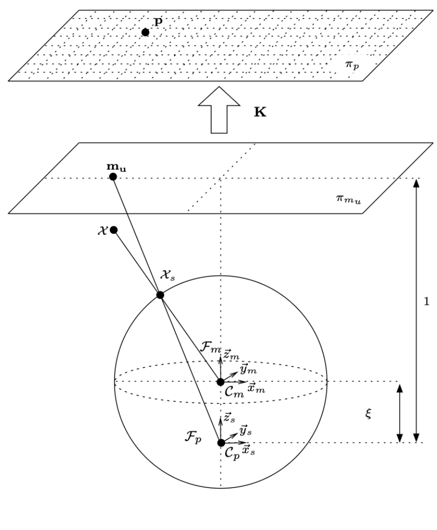


Universal Camera Model

General Projection Model

- Projection of 3D Points

1) world points in the mirror frame are projected onto the unit sphere, $(\boldsymbol{\mathcal{X}})_{\mathcal{F}_m} \longrightarrow (\boldsymbol{\mathcal{X}}_s)_{\mathcal{F}_m} = \frac{\boldsymbol{\mathcal{X}}}{\|\boldsymbol{\mathcal{X}}\|} = (X_s, Y_s, Z_s)$
2) the points are then changed to a new reference frame centered in $\boldsymbol{\mathcal{C}}_p = (0, 0, \xi)$, $(\boldsymbol{\mathcal{X}}_s)_{\mathcal{F}_m} \longrightarrow (\boldsymbol{\mathcal{X}}_s)_{\mathcal{F}_p} = (X_s, Y_s, Z_s + \xi)$
3) we then project the point onto the normalised plane, $\mathbf{m} = (\frac{X_s}{Z_s+\xi}, \frac{Y_s}{Z_s+\xi}, 1) = \hbar(\boldsymbol{\mathcal{X}}_s)$
4) the final projection involves a *generalised* camera projection matrix $\mathbf{K}$ (with $[f_1, f_2]^\top$ the focal length, (u_0, v_0) the principal point and α the skew)



Calibration Process

- Lifting the Projection

$$\mathbf{m} = (\frac{X_s}{Z_s+\xi}, \frac{Y_s}{Z_s+\xi}, 1) = \hbar(\boldsymbol{\mathcal{X}}_s)$$

$$\mathbf{p} = \mathbf{K}\mathbf{m} = \begin{bmatrix} f_1\eta & f_1\eta\alpha & u_0 \\ 0 & f_2\eta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{m} = k(\mathbf{m}) \quad (1)$$

$$\hbar^{-1}(\mathbf{m}) = \begin{bmatrix} \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}x \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}y \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{bmatrix} \quad (2)$$

Unified Model Parameters

- Catadioptric Systems

	ξ	η
Parabola	1	$-2p$
Hyperbola	$\frac{d}{\sqrt{d^2+4p^2}}$	$\frac{-2p}{\sqrt{d^2+4p^2}}$
Ellipse	$\frac{d}{\sqrt{d^2+4p^2}}$	$\frac{2p}{\sqrt{d^2+4p^2}}$
Planar	0	-1
d : distance between focal points $4p$: latus rectum		

Model Parameters

Parabola	$\sqrt{x^2+y^2+z^2} = z + 2p$
Hyperbola	$\frac{(z+\frac{d}{2})^2}{a^2} - \frac{x^2}{b^2} - \frac{y^2}{b^2} = 1$
Ellipse	$\frac{(z+\frac{d}{2})^2}{a^2} + \frac{x^2}{b^2} + \frac{y^2}{b^2} = 1$
Plane	$z = -\frac{d}{2}$
With '$-$' for a hyperbola and '$+$' for an ellipse : $a = 1/2(\sqrt{d^2+4p^2} \pm 2p)$ $b = \sqrt{p(\sqrt{d^2+4p^2} \pm 2p)}$	

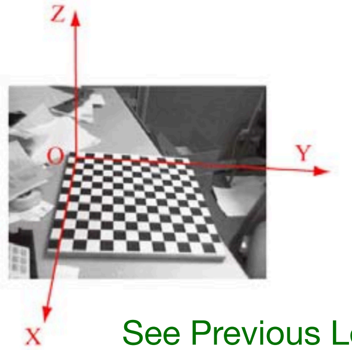
Mirror Equations

- *PS: also valid for fisheye and spherical sensors*

Camera Calibration

(Based on Dynamic Vision by T. Schon)

Camera Calibration - Idea



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

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

See Previous Lectures

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

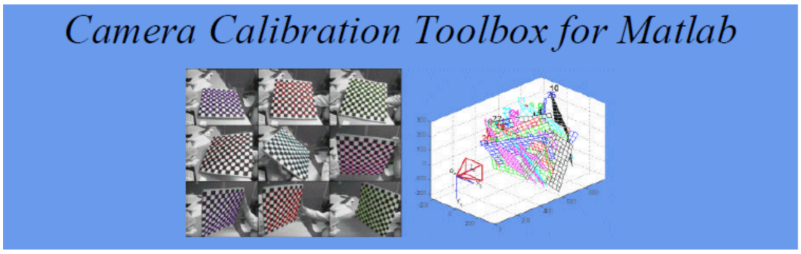
1. Print a checkerboard pattern and attach it to a planar surface.
2. Acquire a few images of the checkerboard pattern under different poses, either by moving the camera or the pattern.
3. Detect the corners in the images. This provides a set of 2D/3D correspondences p_p^{ij}, P_w^i for each image j.
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.

Omni Calibration Tools

Camera Calibration Software

There is very good software freely available on the Internet!

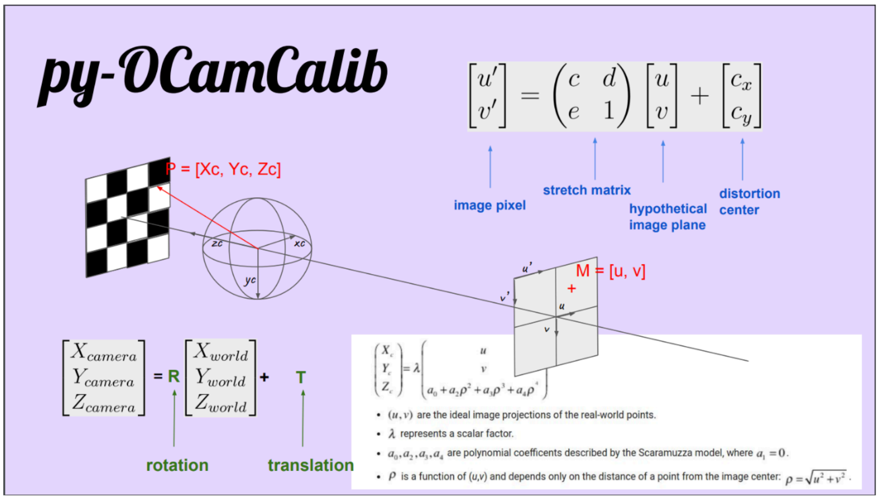
1. Caltech camera calibration toolbox



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

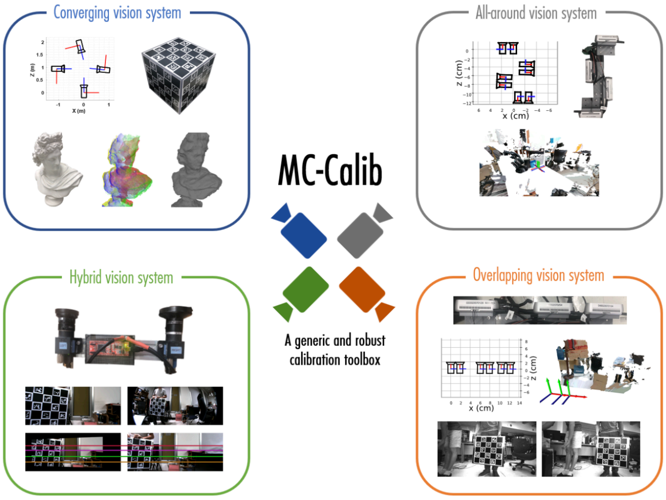
2. OpenCV is a computer vision library originally developed by Intel, now available on sourceforge.net. Free for commercial and research use under BSD license. Contains much more than calibration!

Omnidirectional Camera Calibration



Py-OCamCalib is a pure Python/Numpy implementation of Scaramuzzas OcamCalib Toolbox.

Omnidirectional Rig Calibration



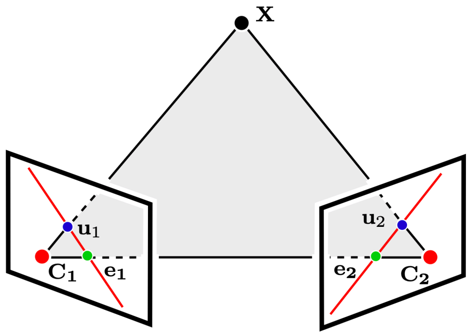
Omnidirectional Geometry

Recap

- Perspective Projective Geometry
- Fundamental Matrix

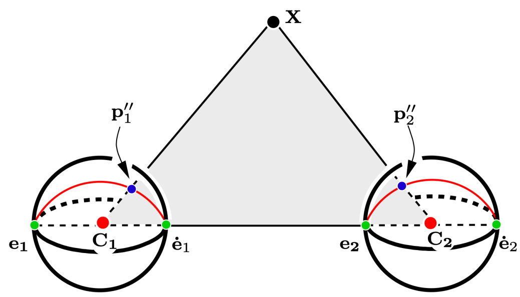
$$\begin{pmatrix} \mathbf{u}_2^\top & 1 \end{pmatrix} \mathtt{F} \begin{pmatrix} \mathbf{u}_1 \\ 1 \end{pmatrix} = 0$$

- Epipolar Constraint
  - Maps points in I_1 to lines in I_2



Omnidirectional Epipolar Geometry

- Epipolar Planes intersect spherical retinas in *circles*
- Epipolar Constraint
  - Maps points in C_1 to *curves* in C_2
- Two epipoles: e and $\dot{e}$
- Need to distinguish
  - Ray orientation
  - Lines and Half-Lines



$$\mathbf{p}_2''^\top \mathbf{F}'' \mathbf{p}_1'' = 0$$

Working with Epipolar Geometry

- Fundamental Matrix

  $\mathbf{p}_2''^\top \, \mathtt{F}'' \, \mathbf{p}_1'' = 0$

- Epipolar Constraint
  - Cannot be applied to image points
  - Applies to <u>3D vectors</u> computed from image points using functions $\boldsymbol{g}$, $\boldsymbol{h}$.

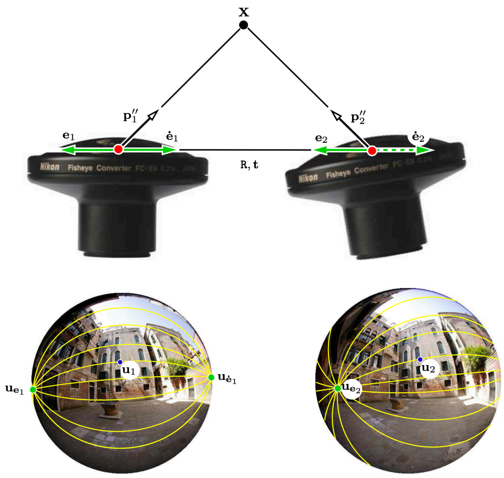
Theorem 2 *The epipolar constraint holds for the vectors* $\mathbf{p}_1$, $\mathbf{p}_2$ *obtained from a calibration process if and only if it holds for the vectors* $\mathbf{p}_1''$, $\mathbf{p}_2''$.

$$\mathbf{p}_2^\top \underbrace{\begin{pmatrix} \mathtt{R}_2 & \\ & 1 \end{pmatrix}^\top \mathtt{F}'' \begin{pmatrix} \mathtt{R}_1 & \\ & 1 \end{pmatrix}}_{\mathtt{F}} \mathbf{p}_1 = 0$$

$$\mathbf{p}_2^\top \mathtt{F} \, \mathbf{p}_1 = 0\,.$$

A Real Example

- Epipolar geometry for a pair of omnidirectional cameras



Omnidirectional 3D Reconstruction

Based on (Micusik, 2004)

3D Reconstruction (ii)

- Projection equation for omnidirectional cameras

$$\alpha''\mathbf{p}'' = \mathtt{P}''\mathbf{X}$$

- Relationship btw calibrated vector p and the "real" vector p''

$$\alpha\,\mathbf{p} = \underbrace{\begin{pmatrix} \mathtt{R}^\top & \\ & 1 \end{pmatrix} \mathtt{P}''}_{\mathtt{P}}\,\mathbf{X} = \mathtt{P}\,\mathbf{X}$$

- Assuming calibrated cameras
  - Compute 3D vectors p for all image points u such that it holds in both images

$$\begin{aligned} \alpha_1\mathbf{p}_1 &= \mathtt{P}_1\mathbf{X} \\ \alpha_2\mathbf{p}_2 &= \mathtt{P}_2\mathbf{X} \end{aligned}$$

3D Reconstruction (i)

- Since matrices P_i are known, the equations (A) can be combined into: $\mathtt{A}\mathbf{X} = \mathbf{0}$

$$\mathtt{A} = \begin{bmatrix} x_1\mathbf{r}_1^{3\top} - z_1\mathbf{r}_1^{1\top} \\ x_1\mathbf{r}_1^{2\top} - y_1\mathbf{r}_1^{1\top} \\ y_1\mathbf{r}_1^{3\top} - z_1\mathbf{r}_1^{2\top} \\ x_2\mathbf{r}_2^{3\top} - z_2\mathbf{r}_2^{1\top} \\ x_2\mathbf{r}_2^{2\top} - y_2\mathbf{r}_2^{1\top} \\ y_2\mathbf{r}_2^{3\top} - z_2\mathbf{r}_2^{2\top} \end{bmatrix} \quad \text{where } \mathbf{r}_1^{i\top} \text{ are the rows of the } \mathtt{P}_1$$

- The linear estimate X should be further used as a starting point in a nonlinear bundle adjustment minimizing reprojection errors.