Master in Computer Vision Barcelona

Module: 3D Vision
Lecture 4: Camera calibration. Pose estimation.
Lecturer: Gloria Haro

Outline

- Calibration
  - Camera resectioning
  - Calibration from a planar pattern
  - Image of the absolute conic
  - Camera distortion
- Pose estimation

Camera model

Remember:

$$\mathbf{x} \equiv P\mathbf{X}, \qquad \mathbf{X} \in \mathbb{P}^3, \mathbf{x} \in \mathbb{P}^2,$$

where P is a 3×4 matrix, the **camera matrix**,

$$P = K\,[R\,|\,\mathbf{t}]$$

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \end{bmatrix}$$

- K is the **calibration matrix** containing the **internal parameters** of the camera.
- R and $\mathbf{t}$ contain the **external parameters** of the camera: position and orientation of the camera in the reference system of the world **(camera pose)**.

Calibration and pose estimation

- Calibration

$$P = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \qquad \longrightarrow \qquad K$$

- Pose estimation

$$P = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

Outline

- Calibration
    - Camera resectioning
    - Calibration from a planar pattern
    - Camera distortion
    - Image of the absolute conic
    - Radial distortion
- Pose estimation

Camera resectioning

It uses a calibration object

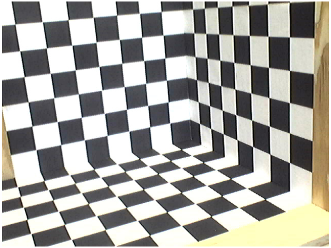


Image source: [B. Thibault]

for establishing a set of **3D to 2D point correspondences**

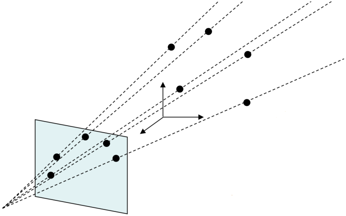


Image source: [V. Lepetit]

Camera resectioning

It uses a calibration object

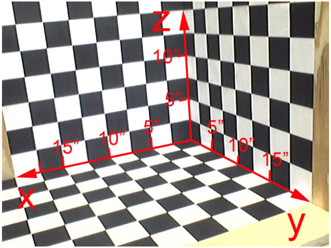


Image source: [B. Thibault]

for establishing a set of **3D to 2D point correspondences**

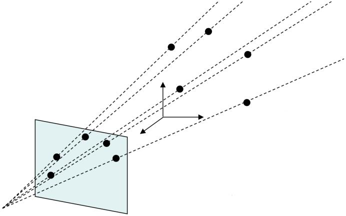


Image source: [V. Lepetit]

Camera resectioning

From $N \geq 6$ **known 3D to 2D point correspondences** $\mathbf{X}_i \longleftrightarrow \mathbf{x}_i$

$$\mathbf{x}_i \equiv P\mathbf{X}_i, \qquad i = 1, 2, \ldots, N \quad \text{where } \mathbf{x}_i \in \mathbb{P}^2, \mathbf{X}_i \in \mathbb{P}^3 \text{are known.}$$

REMARK: As in DLT algorithm is important to previously normalize data, $\mathbf{x}_i$ and $\mathbf{X}_i$, $i = 1, 2, \ldots, N$.

By the same steps as for the DLT algorithm:

$$\begin{pmatrix} \mathbf{0}^T & -w_i\mathbf{X}_i^T & y_i\mathbf{X}_i^T \\ w_i\mathbf{X}_i^T & \mathbf{0}^T & -x_i\mathbf{X}_i^T \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

where $P = \begin{pmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{pmatrix}_{3\times 4}$. Or, in short,

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

($\mathbf{p}$ is the 12×1 vector of unknowns).

Camera resectioning

Each correspondence $\mathbf{X}_i \longleftrightarrow \mathbf{x}_i$
produces 2 equations for the 12 unknowns: $\mathbf{A}_i\mathbf{p} = \mathbf{0}$

As $P_{3\times 4}$ has 11 **degrees of freedom** ($3 \times 4 - 1$ scale factor),
we need $N \geq 6$ correspondences $\mathbf{X}_i \longleftrightarrow \mathbf{x}_i$.

If $\mathbf{A}$ denotes the matrix obtained by

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \dots \\ \mathbf{A}_N \end{pmatrix}_{2N\times 12}$$

with A_i the matrix of each correspondence $\mathbf{X}_i \longleftrightarrow \mathbf{x}_i$,
then the system of equations is

$$\mathbf{A}\mathbf{p} = \mathbf{0} \in \mathbb{R}^{2N}$$

The vector $\mathbf{p}$ is in the null space of $\mathbf{A}$.

Camera resectioning

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

If the correspondences have noise, the identity is not satisfied and in practice we look for **p** such that

$$\begin{array}{l} \min \ \|\mathbf{A}\mathbf{p}\| \\ \text{such that } \|\mathbf{p}\|_2 = 1 \end{array}$$

We compute the solution by SVD :
p ≡ last column of V in the SVD decomposition of $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}}$.

Camera resectioning

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

If the correspondences have noise, the identity is not satisfied and in practice we look for $\mathbf{p}$ such that

$$\begin{array}{l} \min \ \|\mathbf{A}\mathbf{p}\| \\ \text{such that } \|\mathbf{p}\|_2 = 1 \end{array}$$

We compute the solution by SVD :
$\mathbf{p} \equiv$ **last column of V in the SVD decomposition of** $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}}$.

In presence of **outlier correspondences**
we may use the **RANSAC version of the DLT algorithm**

Camera resectioning

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

If the correspondences have noise, the identity is not satisfied and in practice we look for **p** such that

$$\min \ \|\mathbf{A}\mathbf{p}\| \\ \text{such that } \|\mathbf{p}\|_2 = 1$$

We compute the solution by SVD :
$\mathbf{p} \equiv$ last column of V in the SVD decomposition of $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathbf{T}}$.

In presence of **outlier correspondences**
we may use the **RANSAC version of the DLT algorithm**

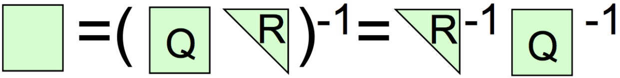
This estimate my be refined by minimizing the geometric reprojection error: $\min \sum_i d(\mathbf{x}_i, P\mathbf{X}_i)^2$

Complete algorithm: Algorithm 7.1, Hartley-Zisserman book, page 181

Camera resectioning

Once P has been estimated, K, R, and $\mathbf{t}$ may be extracted:

1. *QR factorization:* decomposition of the matrix A into a product $A = QR$, where Q is orthogonal, and R is an upper-triangular matrix.



$P = K[R \,|\, \mathbf{t}]$ where K is upper-triangular and R is orthogonal

QR decomposition of $P_{3\times3}$ gives K and R.

The ambiguity in the decomposition is removed by requiring that K have positive diagonal entries.

REMARK: RQ factorization may be also used.

Camera resectioning

Once P has been estimated, K, R, and $\mathbf{t}$ may be extracted:

2. Case of zero skew:

$$A = P_{3\times3}P_{3\times3}^T = (KR)(KR)^T = KRR^TK = KK^T$$

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ x_0 & y_0 & 1 \end{bmatrix} = \begin{bmatrix} f_x^2 + x_0^2 & x_0\, y_0 & x_0 \\ x_0\, y_0 & f_y^2 + y_0^2 & y_0 \\ x_0 & y_0 & 1 \end{bmatrix}$$

- Normalize $(P_{3\times3}P_{3\times3}^T)_{3\times3} = A_{33} = 1$
- Identify parameters of K:

$$x_0 = A_{13}, \quad y_0 = A_{23}$$

$$f_x = \sqrt{A_{11} - x_0^2}, \quad f_y = \sqrt{A_{22} - y_0^2}$$

- Extract extrinsics (pose):

$$[R|\mathbf{t}] = K^{-1}P$$

Remark: R may not be necessarily orthogonal, thus we further impose it.

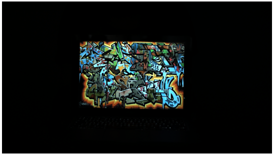
Outline

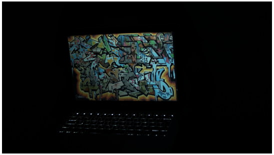
- Calibration
  - Camera resectioning
  - Calibration from a planar pattern
  - Image of the absolute conic
  - Radial distortion
- Pose estimation

Calibration from a planar pattern (Zhang's method)

Z. Zhang. A Flexible New Technique for Camera Calibration. IEEE Trans. on Pattern Analysis and Machine Intelligence, 22 (11), 2000.

Calibration from a planar pattern (Zhang's method)

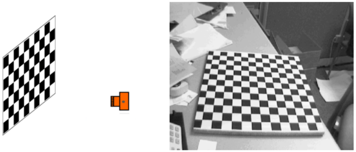




Calibration from a planar pattern (Zhang's method)

1. Relation of a flat object and its image projection: H

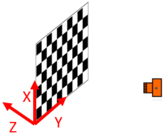
$$\begin{bmatrix} \lambda\, u \\ \lambda\, v \\ \lambda \end{bmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [\mathbf{h}_1\, \mathbf{h}_2\, \mathbf{h}_3] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Calibration from a planar pattern (Zhang's method)

1. Relation of a flat object and its image projection: H

$$\begin{bmatrix} \lambda\, u \\ \lambda\, v \\ \lambda \end{bmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [\mathbf{h}_1\, \mathbf{h}_2\, \mathbf{h}_3] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



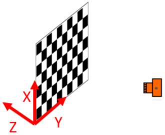
2. Flat object at plane $Z = 0$ projected to the image

$$\begin{bmatrix} \alpha\, u \\ \alpha\, v \\ \alpha \end{bmatrix} = K \left[\underbrace{\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{r}_3}_{R}\ \ \mathbf{t} \right] \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix}$$

Calibration from a planar pattern (Zhang's method)

1. Relation of a flat object and its image projection: H

$$\begin{bmatrix} \lambda\, u \\ \lambda\, v \\ \lambda \end{bmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [\mathbf{h}_1\, \mathbf{h}_2\, \mathbf{h}_3] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



2. Flat object at plane $Z = 0$ projected to the image

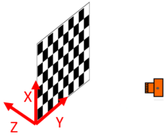
$$\begin{bmatrix} \alpha\, u \\ \alpha\, v \\ \alpha \end{bmatrix} = K \left[\underbrace{\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{r}_3}_{R}\ \mathbf{t}\right] \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = [K\mathbf{r}_1\, K\mathbf{r}_2\, K\mathbf{t}] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Z. Zhang. A Flexible New Technique for Camera Calibration. IEEE Trans. on Pattern Analysis and Machine Intelligence, 22 (11), 2000.

Calibration from a planar pattern (Zhang's method)

1. Relation of a flat object and its image projection: H

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [\mathbf{h}_1\, \mathbf{h}_2\, \mathbf{h}_3] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



2. Flat object at plane $Z = 0$ projected to the image

$$\begin{bmatrix} \alpha u \\ \alpha v \\ \alpha \end{bmatrix} = K \left[\underbrace{\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{r}_3}_{R}\ \mathbf{t} \right] \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = \underbrace{[K\mathbf{r}_1\; K\mathbf{r}_2\; K\mathbf{t}]}_{\sim[\mathbf{h}_1\, \mathbf{h}_2\, \mathbf{h}_3]} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Z. Zhang. A Flexible New Technique for Camera Calibration. IEEE Trans. on Pattern Analysis and Machine Intelligence, 22 (11), 2000.

Calibration from a planar pattern

$$[K\mathbf{r}_1 \; K\mathbf{r}_2 \; K\mathbf{t}] \sim [\mathbf{h}_1 \; \mathbf{h}_2 \; \mathbf{h}_3]$$

Calibration from a planar pattern

$$[K\mathbf{r}_1 \; K\mathbf{r}_2 \; K\mathbf{t}] \sim [\mathbf{h}_1 \; \mathbf{h}_2 \; \mathbf{h}_3]$$

H can be estimated,

Calibration from a planar pattern

$$[K\mathbf{r}_1\ K\mathbf{r}_2\ K\mathbf{t}] \sim [\mathbf{h}_1\ \mathbf{h}_2\ \mathbf{h}_3]$$

H can be estimated, $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{t}$ are unknowns

Calibration from a planar pattern

$$[K\mathbf{r}_1 \; K\mathbf{r}_2 \; K\mathbf{t}] \sim [\mathbf{h}_1 \; \mathbf{h}_2 \; \mathbf{h}_3]$$

H can be estimated, $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{t}$ are unknowns
BUT, we know that $\mathbf{r}_1$ and $\mathbf{r}_2$ are columns of an orthogonal matrix

$$\left.\begin{array}{r} \mathbf{r}_1^T\mathbf{r}_2 = 0 \\ \mathbf{r}_1^T\mathbf{r}_1 = \mathbf{r}_2^T\mathbf{r}_2 = 1 \end{array}\right\} \tag{1}$$

Calibration from a planar pattern

$$[K\mathbf{r}_1\; K\mathbf{r}_2\; K\mathbf{t}] \sim [\mathbf{h}_1\; \mathbf{h}_2\; \mathbf{h}_3]$$

H can be estimated, $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{t}$ are unknowns
BUT, we know that $\mathbf{r}_1$ and $\mathbf{r}_2$ are columns of an orthogonal matrix

$$\left.\begin{array}{c} \mathbf{r}_1^T\mathbf{r}_2 = 0 \\ \mathbf{r}_1^T\mathbf{r}_1 = \mathbf{r}_2^T\mathbf{r}_2 = 1 \end{array}\right\} \tag{1}$$

Isolating

$$[\mathbf{r}_1\; \mathbf{r}_2\; \mathbf{t}] \sim \left[K^{-1}\mathbf{h}_1\; K^{-1}\mathbf{h}_2\; K^{-1}\mathbf{h}_3\right]$$

Calibration from a planar pattern

$$[K\mathbf{r}_1\; K\mathbf{r}_2\; K\mathbf{t}] \sim [\mathbf{h}_1\; \mathbf{h}_2\; \mathbf{h}_3]$$

H can be estimated, $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{t}$ are unknowns
BUT, we know that $\mathbf{r}_1$ and $\mathbf{r}_2$ are columns of an orthogonal matrix

$$\left.\begin{array}{c} \mathbf{r}_1^T\mathbf{r}_2 = 0 \\ \mathbf{r}_1^T\mathbf{r}_1 = \mathbf{r}_2^T\mathbf{r}_2 = 1 \end{array}\right\} \qquad (1)$$

Isolating

$$[\mathbf{r}_1\; \mathbf{r}_2\; \mathbf{t}] \sim \left[K^{-1}\mathbf{h}_1\; K^{-1}\mathbf{h}_2\; K^{-1}\mathbf{h}_3\right]$$

Applying (1)

$$\left.\begin{array}{c} \mathbf{h}_1^T K^{-T} K^{-1}\mathbf{h}_2 = 0 \\ \mathbf{h}_1^T K^{-T} K^{-1}\mathbf{h}_1 = \mathbf{h}_2^T K^{-T} K^{-1}\mathbf{h}_2 \end{array}\right\}$$

Calibration from a planar pattern

$$[K\mathbf{r}_1 \; K\mathbf{r}_2 \; K\mathbf{t}] \sim [\mathbf{h}_1 \; \mathbf{h}_2 \; \mathbf{h}_3]$$

H can be estimated, $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{t}$ are unknowns
BUT, we know that $\mathbf{r}_1$ and $\mathbf{r}_2$ are columns of an orthogonal matrix

$$\left.\begin{array}{c} \mathbf{r}_1^T\mathbf{r}_2 = 0 \\ \mathbf{r}_1^T\mathbf{r}_1 = \mathbf{r}_2^T\mathbf{r}_2 = 1 \end{array}\right\} \qquad (1)$$

Isolating

$$[\mathbf{r}_1 \; \mathbf{r}_2 \; \mathbf{t}] \sim \left[K^{-1}\mathbf{h}_1 \; K^{-1}\mathbf{h}_2 \; K^{-1}\mathbf{h}_3\right]$$

Applying (1)

$$\left.\begin{array}{c} \mathbf{h}_1^T K^{-T} K^{-1}\mathbf{h}_2 = 0 \\ \mathbf{h}_1^T K^{-T} K^{-1}\mathbf{h}_1 = \mathbf{h}_2^T K^{-T} K^{-1}\mathbf{h}_2 \end{array}\right\}$$

where $\omega = (KK^T)^{-1} = K^{-T}K^{-1}$ is the **Image of the Absolute Conic**

Calibration from a planar pattern

We have

$$\left.\begin{array}{r} \mathbf{h}_1^T \omega\, \mathbf{h}_2 = 0 \\ \mathbf{h}_1^T \omega\, \mathbf{h}_1 = \mathbf{h}_2^T \omega\, \mathbf{h}_2 \end{array}\right\}$$

$\omega = (KK^T)^{-1} = K^{-T}K^{-1}$ is a symmetric matrix

$$\omega = \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{12} & \omega_{22} & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{pmatrix}$$

Calibration from a planar pattern

We have

$$\left.\begin{aligned} \mathbf{h}_1^T \omega\, \mathbf{h}_2 &= 0 \\ \mathbf{h}_1^T \omega\, \mathbf{h}_1 &= \mathbf{h}_2^T \omega\, \mathbf{h}_2 \end{aligned}\right\}$$

$\omega = (KK^T)^{-1} = K^{-T}K^{-1}$ is a symmetric matrix

$$\omega = \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{12} & \omega_{22} & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{pmatrix} \longrightarrow \Omega = (\omega_{11}\ \omega_{12}\ \omega_{13}\ \omega_{22}\ \omega_{23}\ \omega_{33})^T$$

Calibration from a planar pattern

We have

$$\left.\begin{array}{r} \mathbf{h}_1^T \omega\, \mathbf{h}_2 = 0 \\ \mathbf{h}_1^T \omega\, \mathbf{h}_1 = \mathbf{h}_2^T \omega\, \mathbf{h}_2 \end{array}\right\} \longrightarrow \text{two constraints on } \omega$$

$\omega = (KK^T)^{-1} = K^{-T}K^{-1}$ is a symmetric matrix

$$\omega = \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{12} & \omega_{22} & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{pmatrix} \longrightarrow \Omega = (\omega_{11}\ \omega_{12}\ \omega_{13}\ \omega_{22}\ \omega_{23}\ \omega_{33})^T$$

Calibration from a planar pattern

$$\mathbf{h}_i^T \omega \mathbf{h}_j^T = (h_{1i} \;\; h_{2i} \;\; h_{3i}) \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{12} & \omega_{22} & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{pmatrix} \begin{pmatrix} h_{1j} \\ h_{2j} \\ h_{3j} \end{pmatrix} = \mathbf{V}_{ij}^T \Omega$$

where $\mathbf{V}_{ij}^T =$
$(h_{1i}\, h_{1j},\; h_{1i}\, h_{2j} + h_{2i}\, h_{1j},\; h_{1i} h_{3j} + h_{3i}\, h_{1j},\; h_{2i} h_{2j},\; h_{2i} h_{3j} + h_{3i}\, h_{2j},\; h_{3i} h_{3j})$

Calibration from a planar pattern

$$\mathbf{h}_i^T \omega \mathbf{h}_j^T = (h_{1i} \ h_{2i} \ h_{3i}) \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{12} & \omega_{22} & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{pmatrix} \begin{pmatrix} h_{1j} \\ h_{2j} \\ h_{3j} \end{pmatrix} = \mathbf{V}_{ij}^T \Omega$$

where $\mathbf{V}_{ij}^T =$
$(h_{1i}\,h_{1j},\ h_{1i}\,h_{2j} + h_{2i}\,h_{1j},\ h_{1i}h_{3j} + h_{3i}\,h_{1j},\ h_{2i}h_{2j},\ h_{2i}h_{3j} + h_{3i}\,h_{2j},\ h_{3i}h_{3j})$

Then

$$\left.\begin{array}{c} \mathbf{V}_{12}^T \Omega = 0 \\ (\mathbf{V}_{11}^T - \mathbf{V}_{22}^T)\,\Omega = 0 \end{array}\right\}$$

Calibration from a planar pattern

$$\mathbf{h}_i^T \omega \mathbf{h}_j^T = (h_{1i}\ \ h_{2i}\ \ h_{3i}) \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{12} & \omega_{22} & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{pmatrix} \begin{pmatrix} h_{1j} \\ h_{2j} \\ h_{3j} \end{pmatrix} = \mathbf{V}_{ij}^T \Omega$$

where $\mathbf{V}_{ij}^T =$
$(h_{1i}\, h_{1j},\ h_{1i}\, h_{2j} + h_{2i}\, h_{1j},\ h_{1i} h_{3j} + h_{3i}\, h_{1j},\ h_{2i} h_{2j},\ h_{2i} h_{3j} + h_{3i}\, h_{2j},\ h_{3i} h_{3j})$

Then

$$\left.\begin{array}{r} \mathbf{V}_{12}^T\,\Omega = 0 \\ (\mathbf{V}_{11}^T - \mathbf{V}_{22}^T)\,\Omega = 0 \end{array}\right\}$$

2 equations for 6 unknowns $\longrightarrow$ we need $n = 3$ views (homographies) to find ω
We have to solve

$$V_{2n\times 6}\,\Omega = \mathbf{0}$$

Calibration from a planar pattern

$$\mathbf{h}_i^T \omega \mathbf{h}_j^T = (h_{1i}\ h_{2i}\ h_{3i}) \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{12} & \omega_{22} & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{pmatrix} \begin{pmatrix} h_{1j} \\ h_{2j} \\ h_{3j} \end{pmatrix} = \mathbf{V}_{ij}^T \Omega$$

where $\mathbf{V}_{ij}^T =$
$(h_{1i}\, h_{1j},\ h_{1i}\, h_{2j} + h_{2i}\, h_{1j},\ h_{1i} h_{3j} + h_{3i}\, h_{1j},\ h_{2i} h_{2j},\ h_{2i} h_{3j} + h_{3i}\, h_{2j},\ h_{3i} h_{3j})$

Then

$$\left.\begin{array}{r} \mathbf{V}_{12}^T\, \Omega = 0 \\ (\mathbf{V}_{11}^T - \mathbf{V}_{22}^T)\, \Omega = 0 \end{array}\right\}$$

2 equations for 6 unknowns $\longrightarrow$ we need $n = 3$ views (homographies) to find ω
We have to solve

$$V_{2n \times 6}\, \Omega = \mathbf{0}$$

Remark: if we use additional constraints on K

- zero skew $\longrightarrow$ we need $n = 2$ views
- zero skew and known principal point $\longrightarrow$ we need $n = 1$ view

Calibration from a planar pattern

We have to solve

$$V\,\Omega = \mathbf{0}$$

In practice we solve:

$$\min_{\Omega} \|V\,\Omega\|_2$$

$$\text{such that } \|\Omega\|_2 = 1$$

SOLUTION: Ω is the singular vector associated to the smallest singular value (last column of $\bar{U}$, where $V \underset{SVD}{=} UD\bar{U}^T$)

Calibration from a planar pattern

We have to solve

$$V\Omega = \mathbf{0}$$

In practice we solve:

$$\min_{\Omega} \|V\Omega\|_2$$

$$\text{such that } \|\Omega\|_2 = 1$$

SOLUTION: Ω is the singular vector associated to the smallest singular value (last column of $\bar{U}$, where $V \underset{SVD}{=} UD\bar{U}^T$)

$$V \xrightarrow{\text{SVD}} \Omega \xrightarrow{\text{rearrangement}} \omega = K^{-T}K^{-1} \xrightarrow{\text{Cholesky}} K$$

Calibration from a planar pattern

How to find R and t (pose) from K?

Calibration from a planar pattern

How to find R and t (pose) from K?

We have

$$[\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{t}] \sim \left[K^{-1}\mathbf{h}_1\, K^{-1}\mathbf{h}_2\, K^{-1}\mathbf{h}_3\right]$$

and since $\|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1$,
then,

$$\mathbf{r}_1 = \frac{K^{-1}\mathbf{h}_1}{\|K^{-1}\mathbf{h}_1\|}, \quad \mathbf{r}_2 = \frac{K^{-1}\mathbf{h}_2}{\|K^{-1}\mathbf{h}_2\|}, \quad \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

$$\mathbf{t} = \frac{K^{-1}\mathbf{h}_3}{\|K^{-1}\mathbf{h}_1\|}$$

Calibration from a planar pattern

How to find R and t (pose) from K?

We have

$$[\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{t}] \sim \left[K^{-1}\mathbf{h}_1\, K^{-1}\mathbf{h}_2\, K^{-1}\mathbf{h}_3\right]$$

and since $\|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1$,
then,

$$\mathbf{r}_1 = \frac{K^{-1}\mathbf{h}_1}{\|K^{-1}\mathbf{h}_1\|}, \quad \mathbf{r}_2 = \frac{K^{-1}\mathbf{h}_2}{\|K^{-1}\mathbf{h}_2\|}, \quad \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

$$\mathbf{t} = \frac{K^{-1}\mathbf{h}_3}{\|K^{-1}\mathbf{h}_1\|}$$

BUT, not guarantee (noise) that $R = (\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{r}_3)$ is an orthogonal matrix.

Calibration from a planar pattern

How to find R and t (pose) from K?

We have

$$[\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{t}] \sim \left[K^{-1}\mathbf{h}_1\, K^{-1}\mathbf{h}_2\, K^{-1}\mathbf{h}_3\right]$$

and since $\|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1$,
then,

$$\mathbf{r}_1 = \frac{K^{-1}\mathbf{h}_1}{\|K^{-1}\mathbf{h}_1\|}, \quad \mathbf{r}_2 = \frac{K^{-1}\mathbf{h}_2}{\|K^{-1}\mathbf{h}_2\|}, \quad \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

$$\mathbf{t} = \frac{K^{-1}\mathbf{h}_3}{\|K^{-1}\mathbf{h}_1\|}$$

BUT, not guarantee (noise) that $R = (\mathbf{r}_1\, \mathbf{r}_2\, \mathbf{r}_3)$ is an orthogonal matrix.
THEN, we impose it:

$$\tilde{R} = U\bar{U}^T$$

where $R \underset{SVD}{=} UD\bar{U}^T$.

Calibration from a planar pattern

Steps of the calibration algorithm

- Estimate the n homographies that relate the planar pattern and the n images.
- Build matrix V.
- Find ω by solving the SVD of V.
- Extract calibration parameters, K, by Cholesky factorization of ω.
- (Optional) Extract pose, R and $\mathbf{t}$.
- (Optional) Refinement by minimizing the geometric error

Outline

- Calibration
  - Camera resectioning
  - Calibration from a planar pattern
  - Image of the absolute conic
  - Camera distortion
- Pose estimation

Image of the absolute conic

The **absolute conic**, Ω_∞, is a conic on the plane at infinity, Π_∞.

It consists of points $X = (\underbrace{x_1, x_2, x_3}_{\mathbf{d}}, 0)^T$ such that $x_1^2 + x_2^2 + x_3^2 = 0$.

In matricial form: $\mathbf{d}^T I \, \mathbf{d} = 0$, then

$$\Omega_\infty = I_{3\times 3}$$

PROPERTY: The absolute conic remains fixed by a similarity.

Image of the absolute conic

REMARK: Ω_∞ is defined in an Euclidean frame, then if we change coordinates to a projective frame the image of Ω_∞ will be, in general, a finite conic represented by matrix ω.

Image of the absolute conic

REMARK: Ω_∞ is defined in an Euclidean frame, then if we change coordinates to a projective frame the image of Ω_∞ will be, in general, a finite conic represented by matrix ω.

Let us see it:

Image of the absolute conic

REMARK: Ω_∞ is defined in an Euclidean frame, then if we change coordinates to a projective frame the image of Ω_∞ will be, in general, a finite conic represented by matrix ω.

Let us see it:

Points on Ω_∞: $\begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix}$

Image of the absolute conic

REMARK: Ω_∞ is defined in an Euclidean frame, then if we change coordinates to a projective frame the image of Ω_∞ will be, in general, a finite conic represented by matrix ω.

Let us see it:

Points on Ω_∞: $\begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix}$

Their image (camera projection) is:

$$P \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} = K[R \ \mathbf{t}] \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} = KR\,\mathbf{d} = \mathbf{x}$$

Image of the absolute conic

REMARK: Ω_∞ is defined in an Euclidean frame, then if we change coordinates to a projective frame the image of Ω_∞ will be, in general, a finite conic represented by matrix ω.

Let us see it:

Points on Ω_∞: $\begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix}$

Their image (camera projection) is:

$$P \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} = K[R \ \mathbf{t}] \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} = \underbrace{KR}_{H} \mathbf{d} = \mathbf{x}$$

$H = KR$ is the (2D) homography that relates Π_∞ and the image plane.

Image of the absolute conic

REMARK: Ω_∞ is defined in an Euclidean frame, then if we change coordinates to a projective frame the image of Ω_∞ will be, in general, a finite conic represented by matrix ω.

Let us see it:

Points on Ω_∞: $\begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix}$

Their image (camera projection) is:

$$P \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} = K[R\ \mathbf{t}] \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} = \underbrace{KR}_{H} \mathbf{d} = \mathbf{x}$$

$H = KR$ is the (2D) homography that relates Π_∞ and the image plane.

Remember how a conic is transformed by a homography: $C' = H^{-T} C H^{-1}$
Then,

$$(KR)^{-T} I (KR)^{-1} = K^{-T} \underbrace{R R^{-1}}_{I} K^{-1} = K^{-T} K^{-1} = (KK^T)^{-1} = \omega$$

Image of the absolute conic

REMARKS:

- ω depends only on the internal parameters of the camera, K.
- ω allows to meausre angles in 3D world from image points.

Image of the absolute conic

REMARKS:

- ω depends only on the internal parameters of the camera, K.
- ω allows to meausre angles in 3D world from image points.

  If $\mathbf{d}_1$ and $\mathbf{d}_2$ are the directions of two lines in $\mathbb{R}^3$, the angle θ they form satisfies:

$$\cos\theta = \frac{\mathbf{d}_1^T \mathbf{d}_2}{\sqrt{\mathbf{d}_1^T \mathbf{d}_1}\sqrt{\mathbf{d}_2^T \mathbf{d}_2}} = \frac{\mathbf{d}_1^T \Omega_\infty \mathbf{d}_2}{\sqrt{\mathbf{d}_1^T \Omega_\infty \mathbf{d}_1}\sqrt{\mathbf{d}_2^T \Omega_\infty \mathbf{d}_2}}$$

Image of the absolute conic

REMARKS:

- ω depends only on the internal parameters of the camera, K.
- ω allows to meausre angles in 3D world from image points.

  If $\mathbf{d}_1$ and $\mathbf{d}_2$ are the directions of two lines in $\mathbb{R}^3$, the angle θ they form satisfies:

$$\cos\theta = \frac{\mathbf{d}_1^T \mathbf{d}_2}{\sqrt{\mathbf{d}_1^T \mathbf{d}_1}\sqrt{\mathbf{d}_2^T \mathbf{d}_2}} = \frac{\mathbf{d}_1^T \Omega_\infty \mathbf{d}_2}{\sqrt{\mathbf{d}_1^T \Omega_\infty \mathbf{d}_1}\sqrt{\mathbf{d}_2^T \Omega_\infty \mathbf{d}_2}}$$

  Since their points at infinity are $(\mathbf{d}_1^T, 0)^T$ and $(\mathbf{d}_2^T, 0)^T$, their image projections (vanishing points) are:

$$\mathbf{x}_1 = KR\mathbf{d}_1 \text{ and } \mathbf{x}_2 = KR\mathbf{d}_2$$

Image of the absolute conic

REMARKS:

- ω depends only on the internal parameters of the camera, K.
- ω allows to meausre angles in 3D world from image points.

  If $\mathbf{d}_1$ and $\mathbf{d}_2$ are the directions of two lines in $\mathbb{R}^3$, the angle θ they form satisfies:

$$\cos\theta = \frac{\mathbf{d}_1^T \mathbf{d}_2}{\sqrt{\mathbf{d}_1^T \mathbf{d}_1}\sqrt{\mathbf{d}_2^T \mathbf{d}_2}} = \frac{\mathbf{d}_1^T \Omega_\infty \mathbf{d}_2}{\sqrt{\mathbf{d}_1^T \Omega_\infty \mathbf{d}_1}\sqrt{\mathbf{d}_2^T \Omega_\infty \mathbf{d}_2}}$$

  Since their points at infinity are $(\mathbf{d}_1^T, 0)^T$ and $(\mathbf{d}_2^T, 0)^T$, their image projections (vanishing points) are:

$$\mathbf{x}_1 = KR\mathbf{d}_1 \text{ and } \mathbf{x}_2 = KR\mathbf{d}_2$$

  We have that:

$$\mathbf{d}_i^T \Omega_\infty \mathbf{d}_j = \mathbf{x}_i^T (KR)^{-T} \Omega_\infty (KR)^{-1} \mathbf{x}_j = \mathbf{x}_i^T \omega\, \mathbf{x}_j, \text{ for } i,j \in \{1,2\}$$

Image of the absolute conic

Then,

$$\cos\theta = \frac{\mathbf{x}_1^T \omega \mathbf{x}_2}{\sqrt{\mathbf{x}_1^T \omega \mathbf{x}_1}\sqrt{\mathbf{x}_2^T \omega \mathbf{x}_2}}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are two vanishing points.

Image of the absolute conic

Then,

$$\cos\theta = \frac{\mathbf{x}_1^T \omega \mathbf{x}_2}{\sqrt{\mathbf{x}_1^T \omega \mathbf{x}_1}\sqrt{\mathbf{x}_2^T \omega \mathbf{x}_2}}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are two vanishing points.

In particular, for orthogonal directions

$$\mathbf{x}_1^T \omega \, \mathbf{x}_2 = 0$$

It provides a constraint on ω (K)

Image of the absolute conic

Then,

$$\cos\theta = \frac{\mathbf{x}_1^T \omega \mathbf{x}_2}{\sqrt{\mathbf{x}_1^T \omega \mathbf{x}_1}\sqrt{\mathbf{x}_2^T \omega \mathbf{x}_2}}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are two vanishing points.

In particular, for orthogonal directions

$$\mathbf{x}_1^T \omega \, \mathbf{x}_2 = 0$$

It provides a constraint on ω (K) $\longrightarrow$ **Useful for self-calibration!**

Outline

- Calibration
    - Camera resectioning
    - Calibration from a planar pattern
    - Image of the absolute conic
    - Camera distortion
- Pose estimation

Camera distortion

Pinhole cameras: linear model of the imaging process
Real (non-pinhole) cameras: lense distortion (non-linear), usually radial distortion

short vs long focal length

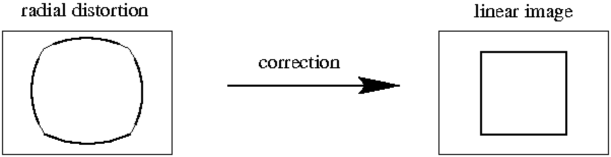
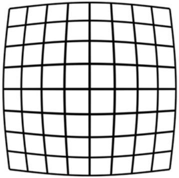


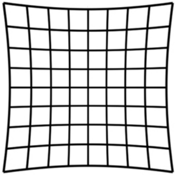
Image source: [Hartley and Zisserman 2004]

Camera distortion

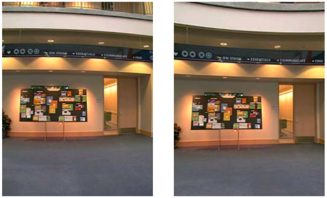
Radial distortion



Barrel Distortion



Pincushion Distortion



Barrel and Pincushion examples

Coordinates in the observed image are displaced away, **barrel distortion**, or towards, **pincushion distortion**, the image center by an amount proportional to their radial distance.

$$x_d = L(r)\,x = (1 + k_1 r + k_2 r^2 + k_3 r^3 + \ldots)\,x$$

$$y_d = L(r)\,x = (1 + k_1 r + k_2 r^2 + k_3 r^3 + \ldots)\,y$$

where $r = \sqrt{x^2 + y^2}$

Image source: [Szeliski 2010]

Camera distortion

How to estimate the distortion parameters:

- **Plumb-line method**, adjust parameters k_i so that images of straight lines become straight in the image.

- Parameters k_i are estimated jointly with those of P by **iterative minimization of a geometric error.**

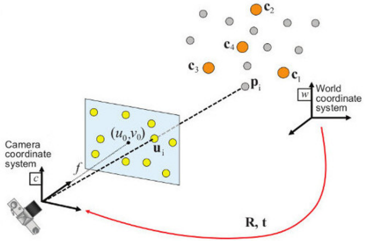
$$\min \sum_i \sum_j \|\mathbf{x}_{ij} - f(k_1, k_2, ..., k_n, K, R_i, \mathbf{t}_i, \mathbf{X}_j)\|_2^2$$

Outline

- Calibration
  - Camera resectioning
  - Calibration from a planar pattern
  - Image of the absolute conic
  - Camera distortion
- Pose estimation

Pose estimation

Estimate camera pose from n 3D-to-2D point correspondences.



If we assume known calibration parameters, K, the problem is called **Perspective-*n*-Point (P*n*P)** problem

$$P = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

Only R and $\mathbf{t}$ (6 dof) need to be determined.

Image source: [https://docs.opencv.org]

PnP methods

PnP problem: Estimate the absolute pose of a calibrated camera
Data: n 3D-2D point correspondences + camera internal parameters K
Goal: estimate camera pose, R and $\mathbf{t}$

Applications:
localization of robots and object manipulation, augmented reality, ...

Collet et al. Object Recognition and Full Pose Registration from a Single Image for Robotic Manipulation. In Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA), 2009.

Müller et al. Mobile augmented reality for computer-assisted percutaneous nephrolithotomy. International journal of computer assisted radiology and surgery, 2013.

PnP methods

Different methods (according to n):

- P3P [1,2]
- P4P [3,4]
- P5P [4]

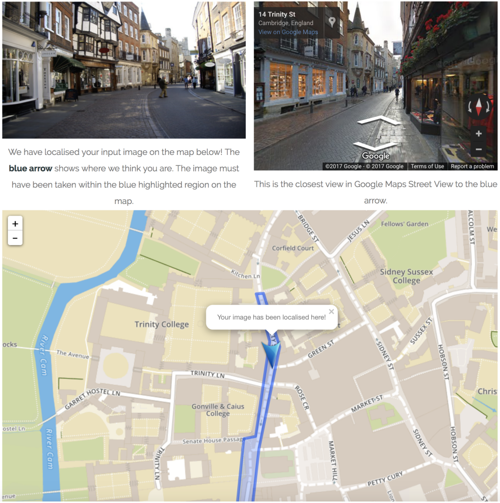
Closed-form solutions, very efficient but very sensitive to noise

- PnP methods
  - iterative (geometric or algebraic errors) $\longrightarrow$ accurate but sensitive to local minima
  - non-iterative $\longrightarrow$ no local minima
    - EPnP [5] (1st efficient non-iterative solution for large n)
    - RPnP [6]
    - ASPnP [7]
    - OPnP [8]
  - Robustified EPnP [9] $\longrightarrow$ fast, accurate, robust to outliers

[1] D. DeMenthon, L. Davis. Exact and approximate solutions of the perspective-three-point problem. PAMI 14(11), 1992.
[2] L. Kneip, D. Scaramuzza, R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. CVPR, 2011.
[3] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACMA, 1981.
[4] B. Triggs. Camera pose and calibration from 4 or 5 known 3D points. ICCV, 1999.
[5] F. Moreno-Noguer, V. Lepetit, P. Fua. Accurate non-iterative o(n) solution to the PnP problem. ICCV 2007.
[6] S. Li, C. Xu, M. Xie. A robust o(n) solution to the perspective-n-point problem. PAMI 34(7), 2012.
[7] Y. Zheng, S. Sugimoto, M. Okutomi. AsPnP: An accurate and scalable solution to the perspective-n-point problem. Trans. on Information and Systems, 96(7), 2013.
[8] Y. Zheng, Y. Kuang, S. Sugimoto, K. Aström, M. Okutomi. Revisiting the PnP problem: A fast, general and optimal solution. ICCV, 2013.
[9] L. Ferraz, X. Binefa, F. Moreno-Noguer. Very fast solution to the PnP problem with algebraic outlier rejection. CVPR 2014.

Pose estimation with deep learning

http://mi.eng.cam.ac.uk/projects/relocalisation



Posenet: estimates 3D position and orientation of the camera given a single monocular image taken from a large previously explored scene.

References

[Hartley and Zisserman 2004] R.I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.

[Szeliski 2010] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010.