TSBB21, Lecture 7
Camera calibration 2

- Camera calibration 2
  - Zhang's method for 3D camera calibration
  - Radial distortion
  - OpenCV:s extended version of Zhang's method
  - Perspective-n-Point (PnP) pose computation
- Literature
  - "A flexible new technique for camera calibration" by Zhengyou Zhang, Microsoft Research. *Available as short article or long report.*
  - "Short about camera geometry and camera calibration" by Maria Magnusson
- Literaturel, deepening
  - Parts of …
    "Introduction to Representations and Estimation in Geometry" (IREG) by Klas Nordberg
  - Parts of …
    "Mathematical Toolbox for Studies in Visual Computation at Linköping University" by Klas Nordberg

Camera calibration, general

- Photogrammetry
  - A 3D calibration object is manufactured with good precision.
    Disadvantage: expensive and complicated.
  - A 2D calibration object is manufactured with good precision. It can be a plane with squares. It is shown for the camera in different orientations. Zhang's approach. Advantage: cheap and simple. Lab task!
- Self-calibration
  - The camera is moving in a static scene.
    Advantage: Flexible.
    Disadvantage: The results are not always reliable.

See also Zhang, section 1: Motivations

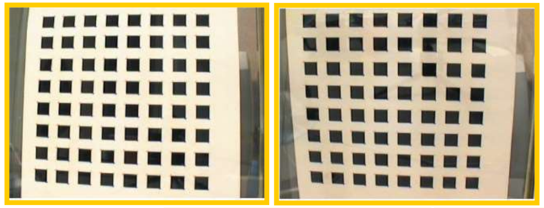
3D Camera calibration according to Zhang

A, **R**, and **t** in **C**=**A**[**Rt**] can be determined individually

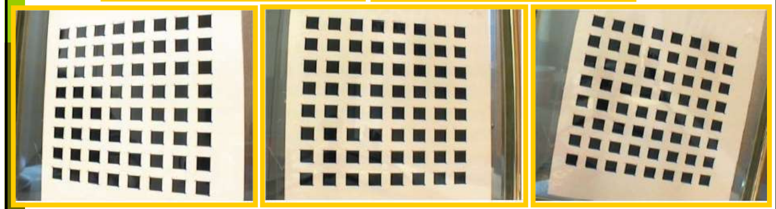
Calibration procedure, see Zhang: Section 3.3

- 1) Print a pattern and attach it to a planar surface.
- 2) Take a few images of the model plane under different orientations by moving the plane. Fig. 1.
- 3) Detect feature points in the images and relate them to points in the world.
- 4) Determine *n* C-matrices by calibrating *n* homographies. Determine **A** and [**Rt**] from the *n* C-matrices.
- 5) Estimate the coefficients of the lens radial distortion from the linear least square solution of an equation system.
- 6) Refine all parameters, including the lens radial distortion parameters in a non-linear minimization algorithm.
- 5) and 6) are not included in the lab "Camera Calibration 1", but in the lab "Camera Calibration 2".

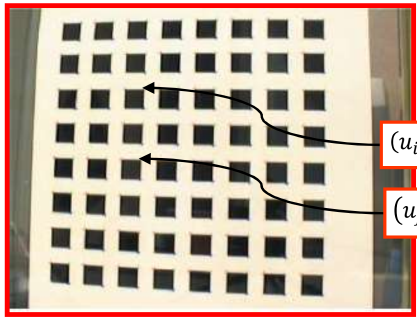
1,2) Hold the pattern in some different orientations and take images



Similar to Fig. 1



3) Detect interesting points in the images and relate them to points in the world



(u_i, v_i) corresponds to (X_i, Y_i)

(u_j, v_j) corresponds to (X_j, Y_j)

From *n* calibration planes we can determine *n* C-matrices by calibrating *n* homographies using the technique described in the previous lecture.

4) Determine A and [Rt] from the *n* C-matrices

Eq. (18) (Magnusson)

$$s(u,v,1)^T = \mathbf{A}[\mathbf{Rt}] \cdot (X,Y,Z,1)^T = \mathbf{A}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3 \quad \mathbf{t}] \cdot (X,Y,Z,1)^T$$

Note that: $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ are are orthonormal!

$$[\mathbf{Rt}] = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3 \quad \mathbf{t}] = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix}$$

For simplicity, assume that the planar pattern is at Z=0.

$$\begin{aligned} s(u,v,1)^T &= \mathbf{A}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3 \quad \mathbf{t}] \cdot (X,Y,0,1)^T \\ &= \mathbf{A}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}] \cdot (X,Y,1)^T = \mathbf{C} \cdot (X,Y,1)^T \end{aligned}$$

$$\mathbf{A}\,[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}] = \mathbf{A} \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix}$$

4) Determine A and [Rt] from the *n* C-matrices

C can only be determined up to a scale factor. Zhang set $C_{33} = 1$ and introduces λ as scale factor.

$$\lambda \cdot \mathbf{A} \cdot \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & 1 \end{pmatrix}$$

$$\lambda \cdot \mathbf{A} \cdot [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}] = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3]$$

Before Eq. (3)

Note that $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{t}$ are gone!

Two important constraints:

$$\mathbf{h}_1^T \; \mathbf{A}^{-T} \; \mathbf{A}^{-1} \; \mathbf{h}_2 = 0 \qquad \textit{Eq. } (3)$$

$$\mathbf{h}_1^T \; \mathbf{A}^{-T} \; \mathbf{A}^{-1} \; \mathbf{h}_1 = \mathbf{h}_2^T \; \mathbf{A}^{-T} \; \mathbf{A}^{-1} \; \mathbf{h}_2 \qquad \textit{Eq. } (4)$$

Proof on next slide!

Proof of the constraints (3) and (4)

Proof of 3 and 4

$$\begin{cases} h_1 = \lambda A r_1 \\ h_2 = \lambda A r_2 \end{cases} \Rightarrow \begin{cases} r_1 = \lambda^{-1} A^{-1} h_1 \\ r_2 = \lambda^{-1} A^{-1} h_2 \end{cases}$$

$$0 = r_1 \bullet r_2 = r_1^T r_2 = (\lambda^{-1} A^{-1} h_1)^T \lambda^{-1} A^{-1} h_2 =$$
$$= \lambda^{-2} h_1^T (A^{-1})^T A^{-1} h_2 \Rightarrow$$

$$\Rightarrow \underbrace{h_1^T A^{-T} A^{-1} h_2 = 0}\ \ 3$$

$$\left.\begin{aligned} 1 &= \|r_1\|^2 = r_1 \bullet r_1 = r_1^T r_1 = \lambda^{-2} h_1^T A^{-T} A^{-1} h_1 \\ 1 &= \|r_2\|^2 = r_2 \bullet r_2 = r_2^T r_2 = \lambda^{-2} h_2^T A^{-T} A^{-1} h_2 \end{aligned}\right\} \Rightarrow$$

$$\Rightarrow \underbrace{h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2}\ \ 4$$

4) Determine A and [Rt] from the *n* C-matrices, cont.

Form a **B**-matrix and a **b**-vector:

$$\mathbf{B} = \mathbf{A}^{-T}\ \mathbf{A}^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} = \{\text{insert and calculate}\} =$$

$$\begin{bmatrix} \dfrac{1}{\alpha^2} & -\dfrac{\gamma}{\alpha^2\beta} & \dfrac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\dfrac{\gamma}{\alpha^2\beta} & \dfrac{\gamma^2}{\alpha^2\beta^2} + \dfrac{1}{\beta^2} & -\dfrac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \dfrac{v_0}{\beta^2} \\ \dfrac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\dfrac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \dfrac{v_0}{\beta^2} & \dfrac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \dfrac{v_0^2}{\beta^2} + 1 \end{bmatrix}$$

Note that the B-matrix is symmetric and that we can solve α,β,… from it.

Eq. (5)

$$\mathbf{b} = [B_{11},\ B_{12},\ B_{22},\ B_{13}, B_{23},\ B_{33}]^T$$

Eq. (6)

4) Determine A and [Rt] from the *n* C-matrices, cont.

Note Zhang's different row/column notation

This is valid:

$$[\mathbf{h}_1\quad \mathbf{h}_2\quad \mathbf{h}_3] = \begin{bmatrix} h_{11} & h_{21} & h_{31} \\ h_{12} & h_{22} & h_{32} \\ h_{13} & h_{23} & h_{33} \end{bmatrix}$$

Set:

$$\mathbf{v}_{ij} = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i3}h_{j3} \end{bmatrix}$$

Then:

$$\mathbf{h}_i^T\ \mathbf{B}\ \mathbf{h}_j = \mathbf{v}_{ij}^T\ \mathbf{b}$$

Eq. (7)

This can be checked by inserting elements to the left and the right side.

Check on next slide!

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = 0$$

Eq. (8)

4) Determine A and [Rt] from the *n* C-matrices, cont.

Check of:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = 0$$

Eq. (8)

Checking Eq 8

3 and 4 and $A^{-T}A^{-1} \Rightarrow \begin{pmatrix} h_1^T B h_2 \\ h_1^T B h_1 - h_2^T B h_2 \end{pmatrix} = 0 \overset{7}{\Rightarrow} \begin{pmatrix} v_{12}^T b \\ (v_{11} - v_{22})^T b \end{pmatrix} = 0$ 8!

4) Determine A and [Rt] from the *n* C-matrices, cont.

2x6-matrix:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = 0$$

Eq. (8)

Pile *n* Eq. (8) on top of eachother

2nx6-matrix:

$$\mathbf{Vb} = 0$$

Eq. (9)

Remember: We have *n* C-matrices obtained from *n* calibration planes.

This is a homogenous equation system, which can be solved by using SVD-technique, see next lecture, "Short about camera geometry…" from previous lecture or "Mathematical Toolbox …"

p. 13

4) Determine A and [Rt] from the *n* C-matrices, cont.

When **b** is known, **B** is simply obtained.
The matrix **B**-matrix is estimated up to a scale factor:

$$\mathbf{B} = \lambda \mathbf{A}^{-T}\mathbf{A}^{-1}$$

The parameters α, β, γ, u_0, v_0 can be extracted from **B**:

$$
\begin{aligned}
v_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}{}^2)\\
\lambda &= B_{33} - \left[B_{13}{}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})\right]/B_{11}\\
\alpha &= \sqrt{\lambda/B_{11}}\\
\beta &= \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}{}^2)}\\
\gamma &= -B_{12}\alpha^2\beta/\lambda\\
u_0 &= \gamma v_0/\alpha - B_{13}\alpha^2/\lambda
\end{aligned}
$$

Below Eq. (9)

A is now determined!

p. 14

4) Determine A and [Rt] from the *n* C-matrices, cont.

Note: It is slightly better to solve **A** from **B** by using Cholesky decomposition (see Mathematical Toolbox). Then the parameters α, β, γ, u_0, v_0 can be directly obtained from **A** and they will probably be more accurate.

p. 15

How many calibration planes are needed?

- One calibration plane gives one calibration matrix **C**.
- One calibration matrix **C** gives one Eq.(8) with 2 equations.
- There are 5 unknowns in **A**.
- If the skew γ=0, there are 4 unknowns in **A**.
- How many calibration planes, at least, are needed to determine **A**?

3 planes are needed.
2 planes are needed if γ=0.

- **C=A[Rt]** is determined up to 8 parameters by 1 calibration plane. There are 6 degrees of freedom in **[Rt]**, 3 rotation angles and 3 translation directions. Consequently 8-6=2 equations are obtained for solving **A** from one calibration plane.

p. 16

4) Determine A and [Rt] from the *n* C-matrices, cont.

See before Eq. (3)

$$\lambda \cdot \mathbf{A} \cdot [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}] = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3]$$

When **A** is known, [**Rt**] is simply obtained as:

$$
\begin{cases}
\mathbf{r}_1 = \lambda \mathbf{A}^{-1}\mathbf{h}_1\\
\mathbf{r}_2 = \lambda \mathbf{A}^{-1}\mathbf{h}_2\\
\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2\\
\mathbf{t} = \lambda \mathbf{A}^{-1}\mathbf{h}_3
\end{cases}, \quad \text{where} \quad \lambda = \frac{1}{\|\mathbf{A}^{-1}\mathbf{h}_1\|} = \frac{1}{\|\mathbf{A}^{-1}\mathbf{h}_2\|}
$$

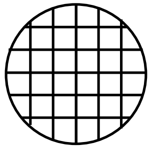
Observe that Zhang now changes λ to λ^{-1}

This is written a bit below Eq. (9)

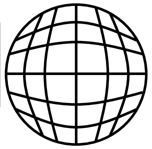
5) Radial distortion

- Radial distortion is the most common

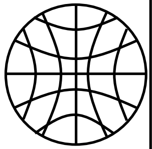
Undis-torted image:



Barrel distor-tion:



Pincush-ion dis-tortion:



- Other types of distortion: the human eye of an astigmatic person, fisheye-lenses, telescope

Radial distorsion can be included in the calibration procedure.

5) Radial distortion example: Extreme wide-angle lens gives barrel distortion

- Example from Aftonbladet: Image inside the "frimurar" room. (Anders Björck, Hasse Aro and the Swedish king are members.)

5) Radial distortion, equations

(u,v) are the real image coordinates, as before.
Let us call the normalized image coordinates (x,y) instead of (u_n,v_n):

$$(u,v,1)^T = \mathbf{A}\cdot\left(\frac{u_i}{f},\frac{v_i}{f},1\right)^T = \mathbf{A}\cdot(u_n,v_n,1)^T = \mathbf{A}\cdot(x,y,1)^T$$

inner parameters: $\alpha,\ \beta,\ \gamma,\ u_0,\ v_0$

$$\begin{cases} u = \alpha\cdot x + \gamma\cdot y + u_0 \\ v = \beta\cdot y + v_0 \end{cases}$$

$$\begin{cases} \breve{u} = \alpha\cdot \breve{x} + \gamma\cdot \breve{y} + u_0 \\ \breve{v} = \beta\cdot \breve{y} + v_0 \end{cases}$$

undistorted image coordinates: (u,v)
distorted image coordinates: $(\breve{u},\breve{v})$
undistorted normalized image coordinates: (x,y)
distorted normalized image coordinates: $(\breve{x},\breve{y})$

5) Radial distortion, equations

undistorted image coordinates: (u,v)
distorted image coordinates: $(\breve{u},\breve{v})$
undistorted normalized image coordinates: (x,y)
distorted normalized image coordinates: $(\breve{x},\breve{y})$

$$r^2 = x^2 + y^2$$

Model:

$$\begin{cases} \breve{x} = x + x\cdot(k_1 r^2 + k_2 r^4) \\ \breve{y} = y + y\cdot(k_1 r^2 + k_2 r^4) \end{cases}$$

k1 and k2 are the coefficients of radial distortion

Proof: See next slide.

$$\begin{cases} \breve{u} = u + (u-u_0)\cdot(k_1 r^2 + k_2 r^4) & Eq.\ (11) \\ \breve{v} = v + (v-v_0)\cdot(k_1 r^2 + k_2 r^4) & Eq.\ (12) \end{cases}$$

The center of the radial distortion is the same as the principal point.

5) Radial distortion, equations

$$\begin{cases} \breve{u} = u + (u - u_0) \cdot (k_1 r^2 + k_2 r^4) \\ \breve{v} = v + (v - v_0) \cdot (k_1 r^2 + k_2 r^4) \end{cases} \qquad \begin{cases} \breve{u} = \alpha \breve{x} + \gamma \breve{y} + u_0 \\ \breve{v} = \beta \breve{y} + v_0 \end{cases} \qquad \begin{cases} u = \alpha x + \gamma y + u_0 \\ v = \beta y + v_0 \end{cases}$$

$$\begin{cases} \alpha \breve{x} + \gamma \breve{y} + u_0 = \alpha x + \gamma y + u_0 + (\alpha x + \gamma y) \cdot (k_1 r^2 + k_2 r^4) \\ \beta \breve{y} + v_0 = \beta y + v_0 + (\beta y) \cdot (k_1 r^2 + k_2 r^4) \end{cases}$$

$$\begin{cases} \alpha \breve{x} + \gamma \breve{y} = \alpha x + \gamma y + (\alpha x + \gamma y) \cdot (k_1 r^2 + k_2 r^4) \\ \breve{y} = y + y \cdot (k_1 r^2 + k_2 r^4) \end{cases}$$

$$\begin{cases} \breve{x} = x + x \cdot (k_1 r^2 + k_2 r^4) \\ \breve{y} = y + y \cdot (k_1 r^2 + k_2 r^4) \end{cases}$$

5) Radial distortion, equations

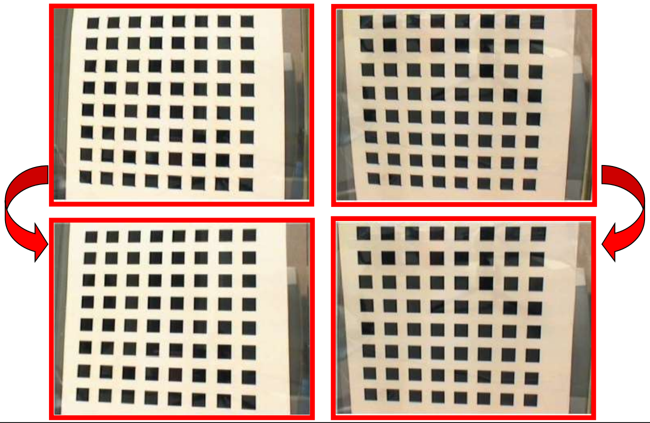
$$\begin{cases} \breve{u} = u + (u - u_0) \cdot (k_1 (x^2 + y^2) + k_2 (x^2 + y^2)^2) & Eq.\ (11) \\ \breve{v} = v + (v - v_0) \cdot (k_1 (x^2 + y^2) + k_2 (x^2 + y^2)^2) & Eq.\ (12) \end{cases}$$

$$\begin{bmatrix} (u - u_0) \cdot (x^2 + y^2) & (u - u_0) \cdot (x^2 + y^2)^2 \\ (v - v_0) \cdot (x^2 + y^2) & (v - v_0) \cdot (x^2 + y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \breve{u} - u \\ \breve{v} - v \end{bmatrix}$$

Given *m* points in *n* images, we can stack all equations together to obtain in total 2*mn* equations, or in matrix form as
Dk=**d**, where **k** =$[k_1,k_2]^T$.
The linear least-square solution is given by:

$$\mathbf{k} = (\mathbf{D}^T \mathbf{D})^{-1}\, \mathbf{D}^T \mathbf{d} \qquad Eq.\ (13)$$

Correction for radial distortion (in the report by Zhang)



6) Refine the parameter estimation in a non-linear minimization algorithm

Magnusson's notation: $s(u, v, 1)^T = \mathbf{A}[\mathbf{Rt}] \cdot (X, Y, Z, 1)^T$ $\quad Eq.\ (18)$

Zhang's notation: $s\tilde{m} = \mathbf{A}[\mathbf{Rt}] \cdot \tilde{M}$ $\quad Eq.\ (1)$

Point in the image

Point in the world

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \left\| \mathbf{m}_{ij} - \hat{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, M_j) \right\|^2 \qquad Eq.\ (14)$$

Projection of point M_j in image i

Can be solved by the Levenberg-Marquardt algorithm, lsqnonlin in Matlab

Degenerated configurations

- If the calibration plane at the second position is parallel with the first position, the 2:nd homography will not give any extra constraints

OpenCV:s extended version of Zhang's method

- Contains a more advanced model for radial distortion:

$$\boxed{\begin{cases} \breve{x} = x \cdot \dfrac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 xy + p_2(r^2 + 2x^2) \\ \breve{y} = y \cdot \dfrac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1(r^2 + 2y^2) + 2p_2 xy \end{cases}}$$

- k_1 and k_2 are Zhang's original coefficients for radial distortion
- p_1 and p_2 are tangential distortion
- For barrel distortion, typically $k_1 > 0$
- For pincushion distortion, typically $k_1 < 0$

Tangential distortion

- Tangential distortion occurs when the lens and the image plane are not parallel. The tangential distortion coefficients p_1 and p_2 model this type of distortion.

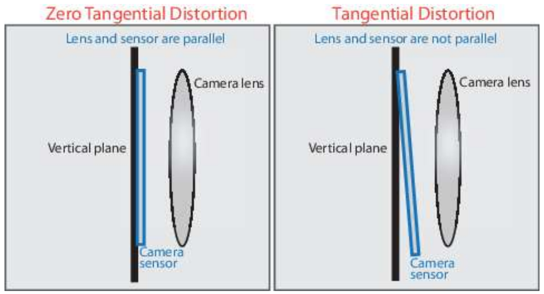
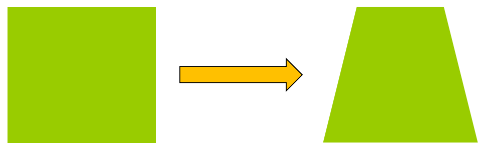


Figure from MathWorks Doc. of R2019b

Tangential distortion

- A simple example:



Alternative model for radial distortion: The arctan model

Used in Lab exercise E: Panorama stitching

Let the image be described in polar coordinates: (r, θ). Then

$$\boxed{r_{\text{out}} = \frac{\arctan(r_{\text{in}} \cdot \gamma)}{\gamma}}$$

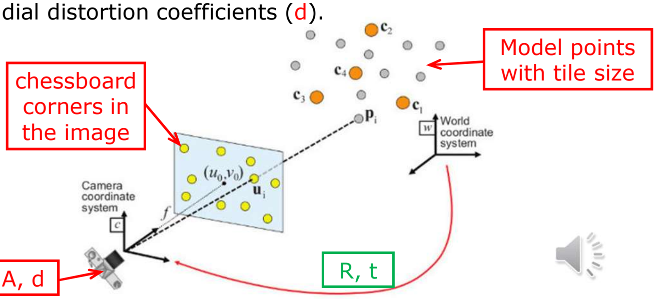
γ is small, e.g. γ=0.001

Perspective-n-Point (PnP) pose computation

- The pose computation problem consists in solving for the rotation and translation that minimizes the reprojection error from 3D-2D point correspondences.
- We used OpenCV:s solvePnP in the Camera calibration lab 2.

Perspective-n-Point (PnP) pose computation

- OpenCV:s solvePnP and related functions estimate the object pose (R, t) given a set of object points (For us: model points with tile size), their corresponding image projections (For us: chessboard corners detected in the image), as well as the camera intrinsic matrix (A) and the radial distortion coefficients (d).



Perspective-n-Point (PnP) pose computation

$$\underbrace{\begin{bmatrix}u\\v\\1\end{bmatrix}}_{\text{chessboard corners in the image}} = \underbrace{\underbrace{\begin{bmatrix}f_x&0&c_x\\0&f_y&c_y\\0&0&1\end{bmatrix}}_{A}\cancel{\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\end{bmatrix}}\underbrace{\begin{bmatrix}r_{11}&r_{12}&r_{13}&t_x\\r_{21}&r_{22}&r_{23}&t_y\\r_{31}&r_{32}&r_{33}&t_z\\\cancel{0}&\cancel{0}&\cancel{0}&\cancel{1}\end{bmatrix}}_{R,t}}_{C}\underbrace{\begin{bmatrix}X_w\\Y_w\\Z_w\\1\end{bmatrix}}_{\text{Model points with tile size}}$$

- We can solve C, given the model points with tile size and the corresponding chessboard corners in the image. This is similar to the calibration of a flat world that we did with the potato stick in Camera Calibration lab 1.
- We can then solve R and t from C and A.

Perspective-n-Point (PnP) pose computation

- The equation on the previous slide was simplified. The radial distortion d should be included also. However, OpenCV:s solvePnP can deal with this.
- Changing the size of the chessboard tiles will change the output translation vector t. However, this will not affect the projection of the model. The reprojection errors will not be affected. Also, R will be correct.