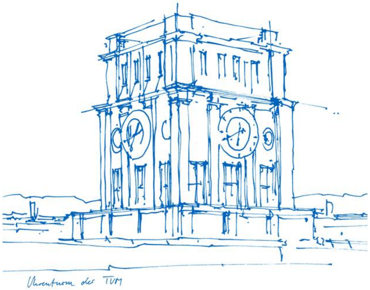
Computer Vision II: Multiple View Geometry (IN2228)

Chapter 04 Camera Calibration

Dr. Haoang Li

10 May 2023 12:00-13:30



Announcement before Class

Today, we will have the **exercise session** about “Representing a Moving Scene” (Chapter 02)

- ✓ Time: from 16:00 to 18:00
- ✓ Room: 102, Hörsaal 2, "Interims I" (5620.01.102)

Explanations before Class

Normal of Projection Plane

- Basic Configuration of 2D Line Detection and 3D Line Projection

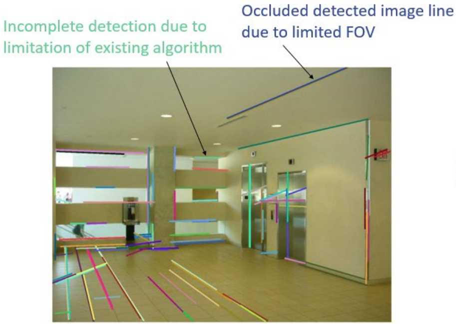
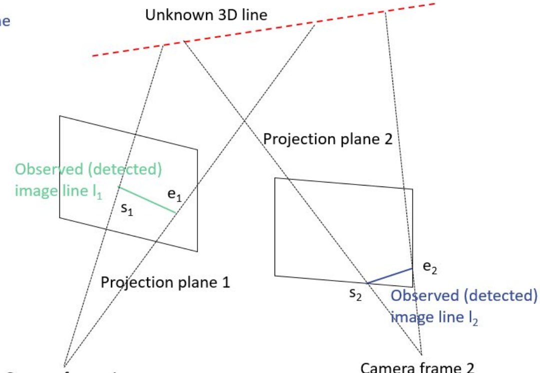


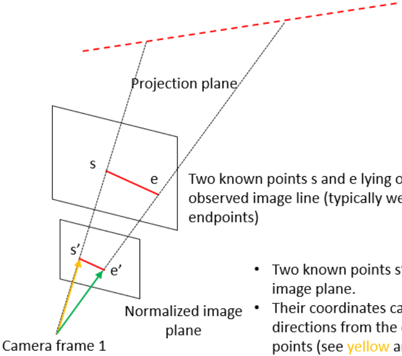
Image line detection



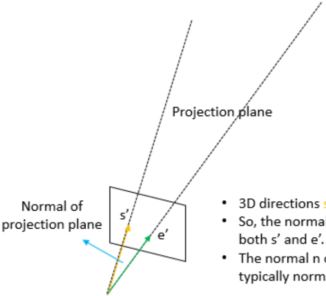
Explanations before Class

Normal of Projection Plane

➢ Computation Based on the Normalized Image Plane



- Two known points s' and e' in the normalized image plane.
- Their coordinates can be also treated as the directions from the camera center to these points (see yellow and green arrows)

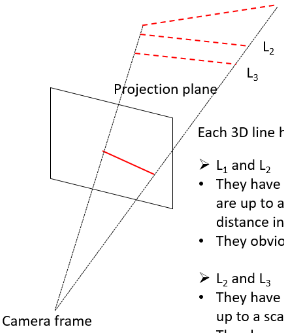


- 3D directions s' and e' span the projection plane.
- So, the normal of projection plane is orthogonal to both s' and e'.
- The normal n can be computed by $n = s' \times e'$ (we typically normalize it in practice)

Explanations before Class

Normal of Projection Plane

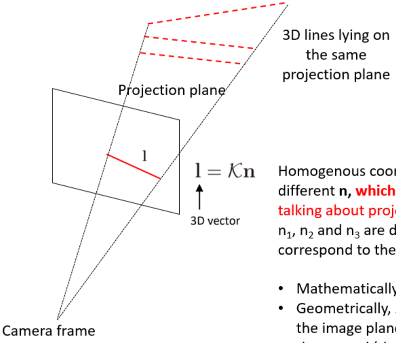
➢ Projection Ambiguity Problem



Each 3D line has unique Plucker Coordinates

➢ L_1 and L_2
- They have the different **n** (two vectors are up to a scale). **n** encodes the distance information.
- They obviously have different **v.**

➢ L_2 and L_3
- They have different **n** (two vectors are up to a scale).
- They have the same **v**.



Homogenous coordinates do not consider scale. So, different **n, which are up to a scale**, are equivalent when talking about projection.
n_1, n_2 and n_3 are different, but only up to scale. So, they correspond to the same projection.

- Mathematically, 2D projection is only determined by n.
- Geometrically, 2D projection is the intersection between the image plane and projection plane parametrized by the normal (the fourth dimension of projection plane equals 0 since it passes through the origin).

Explanations before Class

Normal of Projection Plane

- Reference papers

✓ For conclusions:

[1] Guoxuan Zhang, Jin Han Lee, Jongwoo Lim, and Il Hong Suh, "Building a 3-D Line-Based Map Using Stereo SLAM", IEEE TRO, 2015.

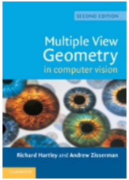
✓ For derivations:

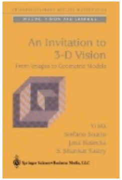
[2] A. Bartoli and P. Sturm, "The 3D line motion matrix and alignment of line reconstructions," in IEEE CVPR, 2001.

✓ For applications:

[3] A. Bartoli and P. Sturm, "Structure from motion using lines: Representation, triangulation and bundle adjustment," *CVIU*, 2005.

Clarification before Class





Reference Materials of this course

- Course “Computer Vision II” provided by Prof. Daniel Cremers

Materials: https://cvg.cit.tum.de/teaching/ss2022/mvg2022
Video: https://www.youtube.com/playlist?list=PLTBdjV_4f-EJn6udZ34tht9EVIW7lbeo4

- Course “Vision Algorithms for Mobile Robotics” provided by Prof. Davide Scaramuzza

Materials: https://rpg.ifi.uzh.ch/teaching.html

- Book “Multiple View Geometry in Computer Vision”: R. Hartley and A. Zisserman

Link: https://www.robots.ox.ac.uk/~vgg/hzbook/

- Book “An Invitation to 3D Vision”: Y. Ma, S. Soatto, J. Kosecka, S.S. Sastry

Link: https://www.eecis.udel.edu/~cer/arv/readings/old_mkss.pdf

- Academic papers in computer vision, robotics, and computer graphics

Dominant venues: ICCV, CVPR, ECCV, TPAMI, IJCV, TIP + IJRR, TRO, ICRA, IROS, RSS + SIGGRAPH

Today's Outline

- Overview of Calibration
- Tsai's Method: From 3D Objects
- Zhang's Method: From Planar Grids
- Image Undistortion

Overview of Calibration

➢ Definition

✓ Calibration is the process to determine

- The extrinsic parameters (R, T) of a camera.
- The intrinsic parameters (K plus lens distortion)

Point in the camera frame

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R \mid T] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Extrinsic matrix

Point in the camera frame

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

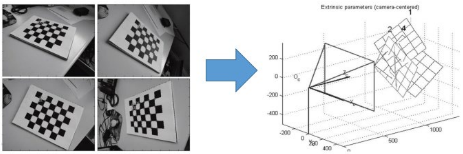
K: Intrinsic/Calibration matrix

Overview of Calibration

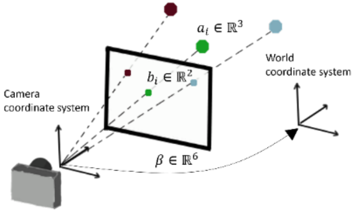
➢ Organization

✓ In this chapter, we will focus on "simultaneous" calibration of extrinsic and intrinsic parameters.

✓ Estimation of extrinsic parameters with "known" intrinsic parameters (camera localization) will be introduced in the Chapter 07 "3D-2D Geometry".



Camera calibration



Camera localization

Overview of Calibration

➢ Organization

✓ We will temporarily neglect the lens distortion and see later how it can be determined.

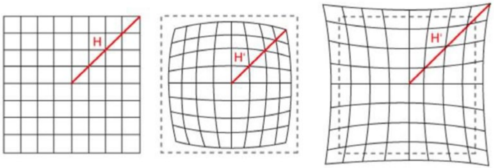


Image distortion

Barrel distortion

Corrected

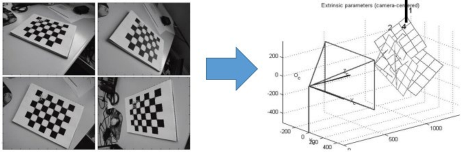
Image Undistortion

Overview of Calibration

➢ Practical Application Scenario

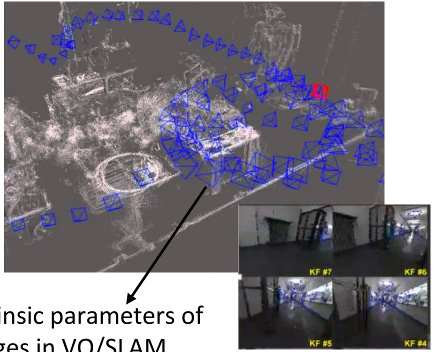
✓ We first calibrate a camera and only **save its intrinsic parameters**. Then we use this camera to run VO/SLAM.

Extrinsic parameters of image(s) of chessboard



Camera calibration

Intrinsic parameters



Extrinsic parameters of images in VO/SLAM

Overview of Calibration

➢ An Example of Intrinsic and Distortion Parameters

✓ Most widely-used SLAM datasets, e.g., TUM RGBD dataset provide intrinsic parameters calibrated beforehand (calibration will be introduced later).

CALIBRATION OF THE COLOR CAMERA

We computed the intrinsic parameters of the RGB camera from the rgbd_dataset_freiburg1/2_rgb_calibration.bag.

Camera	fx	fy	cx	cy	d0	d1	d2	d3	d4
(ROS default)	525.0	525.0	319.5	239.5	0.0	0.0	0.0	0.0	0.0
Freiburg 1 RGB	517.3	516.5	318.6	255.3	0.2624	-0.9531	-0.0054	0.0026	1.1633
Freiburg 2 RGB	520.9	521.0	325.1	249.7	0.2312	-0.7849	-0.0033	-0.0001	0.9172
Freiburg 3 RGB	535.4	539.2	320.1	247.6	0	0	0	0	0

Note that both the color and IR images of the Freiburg 3 sequences have already been undistorted, therefore the distortion parameters are all zero. The original distortion values can be found in the tgz file.

Note: We recommend to use the ROS default parameter set (i.e., without undistortion), as undistortion of the pre-registered depth images is not trivial.

Image resolution: 640*480 pixels

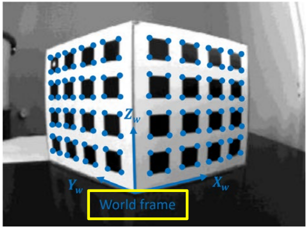
https://cvg.cit.tum.de/data/datasets/rgbd-dataset/file_formats

Tsai's Method: From 3D Objects

➢ Overview

✓ Tsai's method [1] consists of **measuring** the 3D position of $\boldsymbol{n} \geq \mathbf{6}$ 3D control points on a 3D calibration target and the 2D coordinates of their projections in the image.

✓ Tsai's method is based on only a single image.



Through the prior knowledge about the size of each square (e.g., 5 cm), we can obtain the coordinates of each 3D point.

[1] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE Journal of Robotics and Automation, 3(4):323–344, 1987.

Tsai's Method: From 3D Objects

➢ Solving Problem Based on DLT

✓ Direct linear transform (DLT) rewrites the perspective projection equation below as a homogeneous linear equation and solves it by standard methods.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R \mid T] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Known (measured) 2D projection coordinates

Known (measured) 3D point coordinates in the world frame

Intrinsic matrix (unknown)

Extrinsic matrix (unknown)

Tsai's Method: From 3D Objects

➢ Solving Problem Based on DLT

✓ Rewrite the perspective equation for a generic 3D-2D point correspondence

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u r_{11} + u_0 r_{31} & \alpha_u r_{12} + u_0 r_{32} & \alpha_u r_{13} + u_0 r_{33} & \alpha_u t_1 + u_0 t_3 \\ \alpha_v r_{21} + v_0 r_{31} & \alpha_v r_{22} + v_0 r_{32} & \alpha_v r_{23} + v_0 r_{33} & \alpha_v t_2 + v_0 t_3 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

We first compute this matrix as a whole and decompose it back into intrinsic and extrinsic matrices later

Tsai's Method: From 3D Objects

➢ Solving Problem Based on DLT

✓ Rewrite the perspective equation for a generic 3D-2D point correspondence

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

where m_i^T is the i-th row of M

Tsai's Method: From 3D Objects

- Solving Problem Based on DLT

✓ Conversion back from homogeneous coordinates to pixel coordinates leads to

$$\Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \rightarrow P$$

$$u = \frac{\lambda u}{\lambda} = \frac{m_1^T \cdot P}{m_3^T \cdot P}$$

$$v = \frac{\lambda v}{\lambda} = \frac{m_2^T \cdot P}{m_3^T \cdot P}$$

$$\Rightarrow \begin{aligned} (m_1^T - u_i m_3^T) \cdot P &= 0 \\ (m_2^T - v_i m_3^T) \cdot P &= 0 \end{aligned}$$

Divided by scale λ

Tsai's Method: From 3D Objects

➢ Solving Problem Based on DLT

Linear system w.r.t. the elements of unknown M matrix

Known coefficient matrix

✓ By re-arranging the terms, we obtain

$$\begin{aligned}(m_1^{\mathrm{T}} - u_i m_3^{\mathrm{T}}) \cdot P = 0 \\ (m_2^{\mathrm{T}} - v_i m_3^{\mathrm{T}}) \cdot P = 0\end{aligned} \Rightarrow \begin{pmatrix} P_1^{\mathrm{T}} & 0^{\mathrm{T}} & -u_1 P^{\mathrm{T}} \\ 0^{\mathrm{T}} & P_1^{\mathrm{T}} & -v_1 P^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Column vector

✓ For n points, we can stack all these equations into a big matrix

$$\begin{pmatrix} P_1^{\mathrm{T}} & 0^{\mathrm{T}} & -u_1 P_1^{\mathrm{T}} \\ 0^{\mathrm{T}} & P_1^{\mathrm{T}} & -v_1 P_1^{\mathrm{T}} \\ & \vdots & \\ P_n^{\mathrm{T}} & 0^{\mathrm{T}} & -u_n P_n^{\mathrm{T}} \\ 0^{\mathrm{T}} & P_n^{\mathrm{T}} & -v_n P_n^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

Tsai's Method: From 3D Objects

- Solving Problem Based on DLT

✓ Final homogenous linear system

$$\underbrace{\begin{pmatrix} X_w^1 & Y_w^1 & Z_w^1 & 1 & 0 & 0 & 0 & 0 & -u_1X_w^1 & -u_1Y_w^1 & -u_1Z_w^1 & -u_1 \\ 0 & 0 & 0 & 0 & X_w^1 & Y_w^1 & Z_w^1 & 1 & -v_1X_w^1 & -v_1Y_w^1 & -v_1Z_w^1 & -v_1 \\ & & & & & & & \vdots & & & & \\ X_w^n & Y_w^n & Z_w^n & 1 & 0 & 0 & 0 & 0 & -u_nX_w^n & -u_nY_w^n & -u_nZ_w^n & -u_n \\ 0 & 0 & 0 & 0 & X_w^n & Y_w^n & Z_w^n & 1 & -v_nX_w^n & -v_nY_w^n & -v_nZ_w^n & -v_n \end{pmatrix}}_{\text{Q (this matrix is known)}} \underbrace{\begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix}}_{\text{M (this matrix is unknown)}} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \Rightarrow \mathrm{Q} \cdot \mathrm{M} = 0$$

Q (this matrix is **known**)

M (this matrix is **unknown**)

Tsai's Method: From 3D Objects

➢ Solving Problem Based on DLT

$$\Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

✓ Solving the linear system

$$\mathrm{Q} \cdot \mathrm{M} = 0$$

Scale of M does not matter for homogenous linear system. Scale can be recovered based on the constraint of last element of K (i.e., 1).

2n*12 matrix (known)

12-dimensional vector (unknown)

Minimal solution

- Q ($2n \times 12$) should have rank 11 to have a unique (up-to-scale) non-zero solution of vector M.
- Dimension of null space is 1. Vector M can be expressed by a basis vector multiplied by an arbitrary scalar.
- Because each 3D-to-2D point correspondence provides 2 independent equations, then 6 (5.5 in theory) point correspondences are needed.

Tsai's Method: From 3D Objects

➢ Solving Problem Based on DLT

✓ Solving the linear system

$$Q \cdot M = 0$$

2n*12 matrix (known) → Q

12-dimensional vector (unknown) → M

Over-determined solution

- For $n \geq 6$ points, a solution is the Least-Squares solution, which minimizes the sum of squared residuals, $||QM||^2$, subject to the constraint $||M||^2 = 1$ (explain this constraint later).
- It can be solved through Singular Value Decomposition (SVD).

Tsai's Method: From 3D Objects

➢ Solving Problem Based on DLT

✓ Solving the linear system

$$Q \cdot M = \boxed{0}$$

- Why do we need to add the constraint $||M||^2 = 1$? **Zero vector** is an obvious solution.
- How can we apply SVD to computing least-squares solution?

$$\begin{aligned} \arg\min_{b} \quad & \|Ab\|_2^2 \\ \text{subject to} \quad & \|b\|_2 = 1 \end{aligned}$$

$$\mathbf{A} = \mathbf{U\Sigma V}$$

Optimal solution b* is the column of V corresponding to the smallest singular value.

Tsai's Method: From 3D Objects

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

- ➢ Camera Parameter Recovery

✓ Recover the intrinsic and extrinsic parameters

Recap on definition of M matrix

known $\boxed{M} = \boxed{K(R \mid T)}$ Unknown

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} = \begin{bmatrix} \alpha r_{11} + u_0 r_{31} & \alpha r_{12} + u_0 r_{32} & \alpha r_{13} + u_0 r_{33} & \alpha t_1 + u_0 t_3 \\ \alpha r_{21} + v_0 r_{31} & \alpha r_{22} + v_0 r_{32} & \alpha r_{23} + v_0 r_{33} & \alpha t_2 + v_0 t_3 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$

Tsai's Method: From 3D Objects

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$

➢ Camera Parameter Recovery

✓ Enforcing the orthogonality constraint

- We are not enforcing the constraint that $\boldsymbol{R}$ is orthogonal, i.e., $\boldsymbol{R} \cdot \boldsymbol{R}^T = \boldsymbol{I}$
- We can use the so-called QR factorization of $\boldsymbol{M}$, which decomposes $\boldsymbol{M}$ into a R (orthogonal), T, and an upper triangular matrix (i.e., K)
- Orthogonality is **inherently** satisfied

$$Q^\mathsf{T} = Q^{-1}$$

A = QR

Case of square matrix

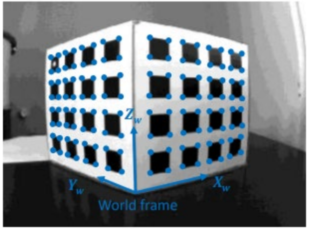
$$\begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,n} \\ & u_{2,2} & u_{2,3} & \cdots & u_{2,n} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & u_{n-1,n} \\ 0 & & & & u_{n,n} \end{bmatrix}$$

⇨ Case of non-square matrix

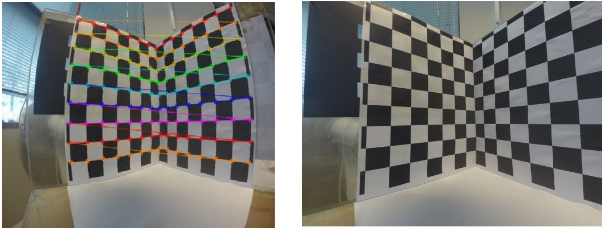
Tsai's Method: From 3D Objects

➢ Practical Setup

- ✓ Use many more than 6 points (ideally more than 20) and non coplanar.
- ✓ Corners can be detected with accuracy < 0.1 pixels (will be introduced in Chapter 05 "Correspondence Estimation").



Keypoint detection



Distortion can be also considered

Zhang's Method: From Planar Grids

➢ A Simpler Setup

✓ Zhang's method [2] relies on 3D coplanar points.

A single image

Multiview images

Tsai calibration object (left), Zhang calibration object (right)

[2] Z. Zhang. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330–1334, 2000.

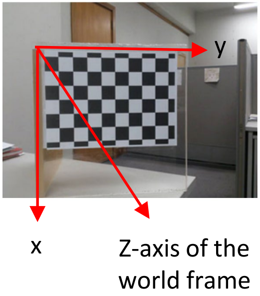
Zhang's Method: From Planar Grids

➢ Solving Problem based on DLT

✓ As in Tsai's method, we start by neglecting the radial distortion.
✓ Zhang's method the points are all coplanar, i.e., $\boldsymbol{Z_w}$ = 0.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R \mid T] \cdot \begin{bmatrix} X_w \\ Y_w \\ \boxed{0} \\ 1 \end{bmatrix} \Rightarrow$$

$$\Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix}$$



Zhang's Method: From Planar Grids

- Solving Problem based on DLT

✓ Rewriting Equations

$$\Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix}$$

$$\Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \qquad \Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

Zhang's Method: From Planar Grids

- Solving Problem based on DLT

✓ Rewriting Equations

$$\Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \cdot \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

This matrix is called Homography

$$\Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \qquad \Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix} \cdot \underbrace{\begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}}_{P}$$

where h_i^T is the i-*th* row of H

Zhang's Method: From Planar Grids

- Solving Problem based on DLT

✓ Conversion back from homogeneous coordinates to pixel coordinates

$$\Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_1^{\mathrm{T}} \\ h_2^{\mathrm{T}} \\ h_3^{\mathrm{T}} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \rightarrow P$$

Homogeneous coordinates

$$u = \frac{\lambda u}{\lambda} = \frac{h_1^{\mathrm{T}} \cdot P}{h_3^{\mathrm{T}} \cdot P} \qquad v = \frac{\lambda v}{\lambda} = \frac{h_2^{\mathrm{T}} \cdot P}{h_3^{\mathrm{T}} \cdot P}$$

Pixel coordinates

$$\Rightarrow \begin{aligned} (h_1^{\mathrm{T}} - u_i h_3^{\mathrm{T}}) \cdot P_i &= 0 \\ (h_2^{\mathrm{T}} - v_i h_3^{\mathrm{T}}) \cdot P_i &= 0 \end{aligned}$$

The i-th observed 3D-2D correspondence

Zhang's Method: From Planar Grids

- Solving Problem based on DLT

✓ Re-arranging the terms

$$\begin{aligned}(h_1^T - u_i h_3^T) \cdot P_i = 0\\(h_2^T - v_i h_3^T) \cdot P_i = 0\end{aligned} \quad \Rightarrow \quad \begin{aligned}P_i^T \cdot h_1 + 0 \cdot h_2^T - u_i P_i^T \cdot h_3^T = 0\\0 \cdot h_1^T + P_i^T \cdot h_2^T - v_i P_i^T \cdot h_3^T = 0\end{aligned} \quad \Rightarrow \begin{pmatrix} P_i^T & 0^T & -u_1 P_i^T \\ 0^T & P_i^T & -v_1 P_i^T \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Linear system w.r.t.
elements of homography

Zhang's Method: From Planar Grids

➢ Solving Problem based on DLT

✓ For n points (from a single view), we can stack all these equations into a big matrix

$$\begin{pmatrix} P_1^T & 0^T & -u_1 P_1^T \\ 0^T & P_1^T & -v_1 P_1^T \\ \cdots & \cdots & \cdots \\ P_n^T & 0^T & -u_n P_n^T \\ 0^T & P_n^T & -v_n P_n^T \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \Rightarrow \mathrm{Q} \cdot \mathrm{H} = 0$$

Q (this matrix is **known**) H (this matrix is **unknown**)

Zhang’s Method: From Planar Grids

➢ Solving Problem based on DLT

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \cdot \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

This matrix is called Homography

Scale does not matter

✓ Solving the linear system

$$\mathrm{Q} \cdot \mathrm{H} = 0$$

Minimal solution

- Q $(2n \times 9)$ should have rank 8 to have a unique (up to a scale) non-trivial solution H (properties of Homography will be introduced in the future)
- Each point correspondence provides 2 independent equations
- Thus, a minimum of 4 non-collinear points is required

Zhang's Method: From Planar Grids

➢ Solving Problem based on DLT

✓ Solving the linear system

$$Q \cdot H = 0$$

Solution for $n \geq 4$ points

- It can be solved through Singular Value Decomposition (SVD) (same considerations as before)

$$\underset{b}{\arg\min} \quad \|Ab\|_2^2 \qquad \text{subject to} \quad \|b\|_2 = 1$$

$$\mathbf{A} = \mathbf{U\Sigma V}$$

Optimal solution b* is the column of V corresponding to the smallest singular value.

Zhang's Method: From Planar Grids

➢ Camera Parameter Recovery: Overview

✓ K, R, T can be recovered by decomposition of H

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}$$

Up to scale

Can we still use QR decomposition? No.
Upper triangular matrix * orthogonal matrix

- Different from Tsai's method, the decomposition of H into K, R, T requires multiple views (introduced later).
- In practice the more views the better, e.g., 20-50 views spanning the entire field of view of the camera for the best calibration results.

Zhang's Method: From Planar Grids

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}$$

➢ Camera Parameter Recovery: Overview

✓ Each view j has a different homography H^j (and so a different R^j and T^j). However, $\boldsymbol{K}$ is the same for all views*.

✓ Estimate the homography H_i for each i-th view using the DLT algorithm.

$$\begin{bmatrix} h^j_{11} & h^j_{12} & h^j_{13} \\ h^j_{21} & h^j_{22} & h^j_{23} \\ h^j_{31} & h^j_{33} & h^j_{33} \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r^j_{11} & r^j_{12} & t^j_1 \\ r^j_{21} & r^j_{22} & t^j_2 \\ r^j_{31} & r^j_{32} & t^j_3 \end{bmatrix}$$

Each view corresponds to a homography

* In our slides, we also denote intrinsic matrix by *M*.

Zhang’s Method: From Planar Grids

Known

Intrinsic matrix

➢ Camera Parameter Recovery: Details

$$H = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] = sM[r_1 \quad r_2 \quad t]$$

✓ First step: Determine **intrinsic matrix** *M* of camera from **a set of known homographies**.

- Idea: Use the prior constraints of rotation to derive **formulas w.r.t. only unknown intrinsic parameters**.

- We first express columns of rotation by **unknown intrinsic parameters**

$$r_1 = \lambda M^{-1} h_1 \quad r_2 = \lambda M^{-1} h_2 \qquad t = \lambda M^{-1} h_3 \qquad \lambda = s^{-1}$$

Zhang's Method: From Planar Grids

Intrinsic matrix

Known

- ➢ Camera Parameter Recovery: Details

$$H = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] = sM[r_1 \quad r_2 \quad t]$$

✓ First step: Determine **intrinsic matrix** *M* of camera from **a set of known homographies**.

- We then enforce the constraints of columns w.r.t. rotation

First constraint w.r.t. only M

$$r_1^T r_2 = 0 \xrightarrow[r_2 = \lambda M^{-1} h_2]{r_1 = \lambda M^{-1} h_1} h_1^T \boxed{(M^{-1})^T M^{-1}} h_2 = 0$$

Second constraintw.r.t. only M

$$\|r_1\| = \|r_2\| = 1 \longrightarrow r_1^T r_1 = r_2^T r_2 \xrightarrow[r_2 = \lambda M^{-1} h_2]{r_1 = \lambda M^{-1} h_1} h_1^T \boxed{(M^{-1})^T M^{-1}} h_1 = h_2^T \boxed{(M^{-1})^T M^{-1}} h_2$$

Zhang's Method: From Planar Grids

➢ Camera Parameter Recovery: Details

✓ First step: Determine **intrinsic matrix** *M* of camera from **a set of known homographies**.

- We define a matrix B w.r.t. the unknown intrinsic parameters of M
- Instead of directly solving M, we firsts estimate B

$$h_1^T (M^{-1})^T M^{-1} h_2 = 0$$

$$h_1^T (M^{-1})^T M^{-1} h_1 = h_2^T (M^{-1})^T M^{-1} h_2$$

⇨

$$B = (M^{-1})^T M^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$$

A symmetric matrix
(6 elements to estimate)

$$\frac{1}{f_x^2}$$

Example: B11 is w.r.t. the focal length

- If we solved matrix B based on Homography, we can extract intrinsic parameters from B

Zhang's Method: From Planar Grids

➢ Camera Parameter Recovery: Details

✓ First step: Determine **intrinsic matrix** *M* of camera from **a set of known homographies**.

- Each homography $H_i \sim K * [\, \boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{t} \,]$ provides two linear equations in the 6 entries of the matrix

$$B = (M^{-1})^T M^{-1}$$

$$h_1^T (M^{-1})^T M^{-1} h_2 = 0 \qquad h_1^T (M^{-1})^T M^{-1} h_1 = h_2^T (M^{-1})^T M^{-1} h_2$$

$$h_i^T B h_j = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i3}h_{j3} \end{bmatrix}^T \begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix}$$

Vectors h1 and h2 are known

- Stack 2N equations from N views, to yield a linear system $A\boldsymbol{b} = \boldsymbol{0}$. Solve for b (i.e., B) using the Singular Value Decomposition (SVD).
- Typically, we need more than 3 views (each view provides two constraints).

Zhang's Method: From Planar Grids

➢ Camera Parameter Recovery: Details

Known — Intrinsic matrix

$$H = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] = sM[r_1 \quad r_2 \quad t]$$

✓ Second step: The **extrinsic parameters** for each view can be computed using M:

- Compute each column

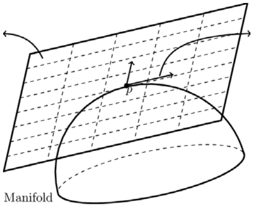
$$r_1 = \lambda M^{-1} h_1 \qquad r_2 = \lambda M^{-1} h_2$$

$$r_3 = r_1 \times r_2 \qquad t = \lambda M^{-1} h_3$$

s.t. $\|r_1\| = \|\lambda M^{-1} h_1\| = 1$

Constraint on scale

- Finally, build $R_i = (\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3)$ and enforce rotation matrix constraints.



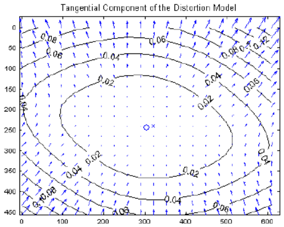
Projecting the result from the **matrix space** onto the **SO(3) manifold**

Image Undistortion

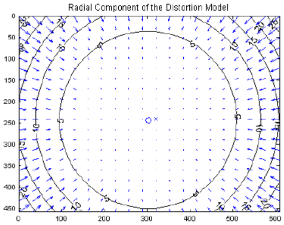
➢ Recap on Type of Distortion

✓ Radial Distortion occurs when light rays **bend more** near the edges of a lens than they do at its optical center.

✓ Tangential Distortion: if the lens is misaligned (not perfectly parallel to the image sensor), a tangential distortion occurs.



Tangential distortion



Radial distortion

Image Undistortion

➢ Introducing Distortion Model into Perspective Projection

✓ From world frame to camera frame

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

✓ From camera frame to image (distortion-free case)

$$x' = x/z$$
$$y' = y/z$$
$$u = f_x * x' + c_x$$
$$v = f_y * y' + c_y$$

(Non-homogenous coordinates)

Image Undistortion

➢ Introducing Distortion Model into Perspective Projection

✓ Adding the distortion coefficients

- k_n coefficients will describe radial distortion
- p_n coefficients will describe tangential distortion

$$x' = x/z$$
$$y' = y/z$$
$$u = f_x * x' + c_x$$
$$v = f_y * y' + c_y$$

Distortion-free model

This expression is not unique

$$x'' = x' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2$$
$$y'' = y' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1 (r^2 + 2y'^2) + 2p_2 x' y'$$
$$\text{where} \quad r^2 = x'^2 + y'^2$$

Distortion model

$$\Rightarrow \quad u = f_x * x'' + c_x$$
$$v = f_y * y'' + c_y$$

Image Undistortion

➢ Joint Estimation

Given the object points and image points (detected chessboard corners), we conduct the following steps (Zhang's method).

- Compute the initial **intrinsic parameters**. The **distortion coefficients** are all **set to zeros** initially.
- Estimate the initial **extrinsic parameters** as if the intrinsic parameters have been already known.

- Run the **gradient descent algorithm** to minimize the reprojection error to jointly optimize/estimate intrinsic, extrinsic, and distortion parameters.

Image Undistortion

- Joint Estimation

Reprojection error is the Euclidean distance (in pixels) between an **observed image point** and the corresponding **3D point reprojected** onto the camera frame.

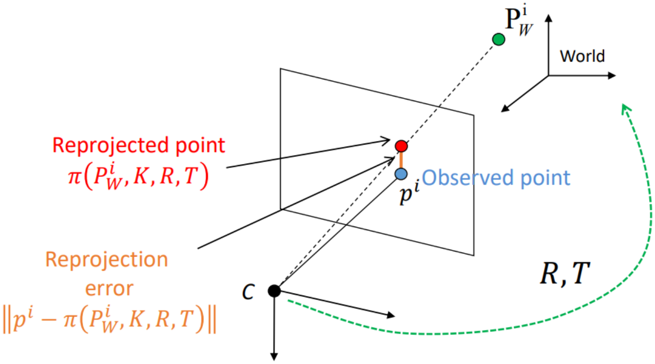
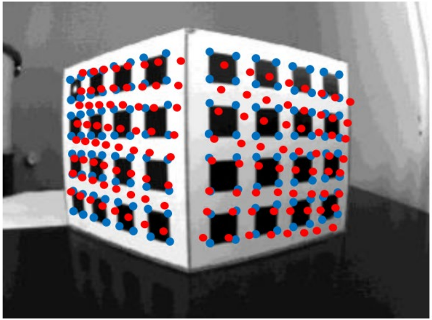


Image Undistortion

➢ Joint Estimation

The calibration parameters K, R, T determined by the DLT can be refined by minimizing the following cost/objective function

$$K, R, T, lens\ distortion = argmin_{K,k_1,R,T} \sum_{i=1}^{n} \left\| p^i - \pi(P_W^i, K, k_1, R, T) \right\|^2$$



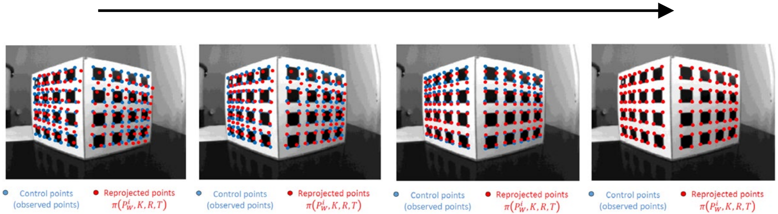
● Control points (observed points) ● Reprojected points $\pi(P_W^i, K, R, T)$

Image Undistortion

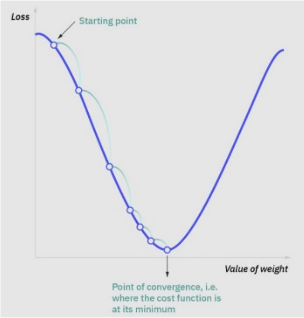
➢ Joint Estimation

$$K, R, T, lens\ distortion = \arg\min_{K,k_1,R,T} \sum_{i=1}^{n} \left\| p^i - \pi(P_W^i, K, k_1, R, T) \right\|^2$$

The cost function can be minimized using gradient descent algorithm (details will be introduced in Chapter 11: Bundle Adjustment and Optimization).



Reprojection error is minimized iteratively



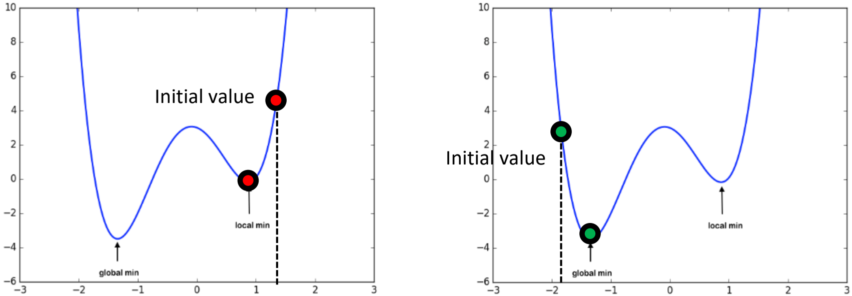
Derivative computation

Image Undistortion

➢ Joint Estimation

$$K, R, T, lens\ distortion = argmin_{K,k_1,R,T} \sum_{i=1}^{n} \left\| p^i - \pi(P_W^i, K, k_1, R, T) \right\|^2$$

The cost function can be minimized using gradient descent algorithm (details will be introduced in Chapter 11: Bundle Adjustment and Optimization).



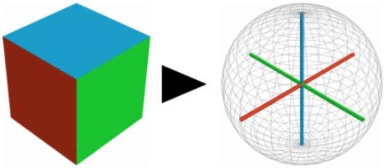
Global optimum finding

Image Undistortion

- Line-based Undistortion

If we only have a single image obtained in a Man-made environment (Manhattan world), can we still manage to undistort an image?

	Constraint	Parameters to estimate
Multiple images with points	Multi-view constraint	Intrinsic parameters Distortion parameters Extrinsic parameters
Single image with lines in Manhattan world	Structural regularity constraint	Intrinsic parameters Distortion parameters Vanishing points



Manhattan world

This knowledge will not be asked in the exam.

Image Undistortion

➤ Line-based Undistortion

✓ Recap on explicit distortion model

We use the explicit model with respect to a single radial distortion parameter r (instead of the polynomial model) to convert the distorted point (x', y') to the original point (x, y)

unknown

$$\begin{cases} x' = c_x + (x - c_x) \cdot \dfrac{\sqrt{1 + 4 \cdot r \cdot d} - 1}{2 \cdot r \cdot d} \\ y' = c_y + (y - c_y) \cdot \dfrac{\sqrt{1 + 4 \cdot r \cdot d} - 1}{2 \cdot r \cdot d} \end{cases}$$

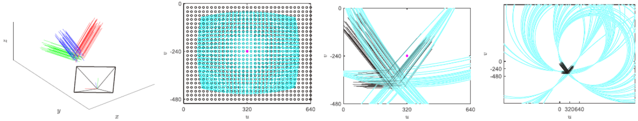
$$d = (x - c_x)^2 + (y - c_y)^2$$

Image Undistortion

➢ Line-based Undistortion

We leverage the fact that reliable calibration (intrinsic+distortion) parameters lead to the vanishing points maximizing the number of inlier lines.

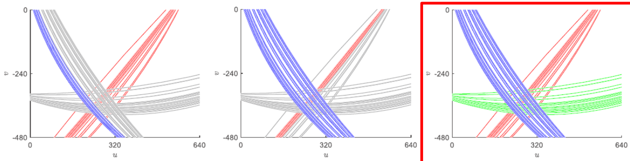
Input data



90 Lines Aligned to MF Axes — Sampled Pixels — 90 Lines and 90 Circles — Zoom Out

(a) 3D Scene — (b) Original and Distorted Images (Input of Hyb-Dis is 90 Circles)

Vanishing point estimation



(100 %, 40.96 %)

$$\begin{bmatrix} -0.6390 & 0.6607 & -0.3940 \\ 0.7520 & 0.6443 & -0.1391 \\ -0.1619 & 0.3851 & 0.9085 \end{bmatrix}$$

(c) 1×10^{-6}

(100 %, 46.99 %)

$$\begin{bmatrix} 0.5215 & -0.4743 & 0.7093 \\ 0.7737 & 0.6134 & -0.1587 \\ -0.3598 & 0.6315 & 0.6868 \end{bmatrix}$$

(d) 2×10^{-6}

(100 %, 100 %)

$$\begin{bmatrix} 0.6730 & 0.5260 & 0.5200 \\ -0.1307 & 0.7765 & -0.6164 \\ -0.7280 & 0.3469 & 0.5913 \end{bmatrix}$$

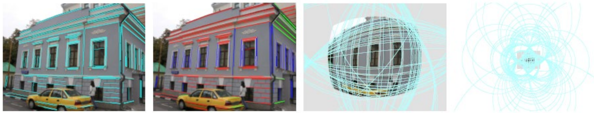
(e) 3×10^{-6}

All the circles (lines) are labelled as inliers when we use the correctly estimated calibrations parameters

Image Undistortion

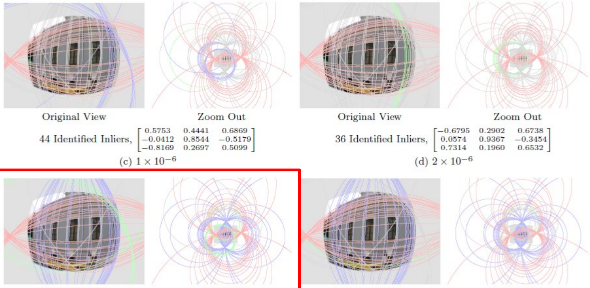
- ➢ Line-based Undistortion

We aim to find the optimal calibration parameters to estimate vanishing points that maximize the number of inlier lines.



238 Lines | Ground Truth Clusters | 75 Circles (Input of Hyb-Dis) | Zoom Out

(a) Original Image | (b) Distorted Image



Original View | Zoom Out

44 Identified Inliers, $\begin{bmatrix} 0.5753 & 0.4441 & 0.6869 \\ -0.0412 & 0.8544 & -0.5179 \\ -0.8169 & 0.2697 & 0.5099 \end{bmatrix}$

(c) 1×10^{-6}

Original View | Zoom Out

36 Identified Inliers, $\begin{bmatrix} -0.6795 & 0.2902 & 0.6738 \\ 0.0574 & 0.9367 & -0.3454 \\ 0.7314 & 0.1960 & 0.6532 \end{bmatrix}$

(d) 2×10^{-6}

Original View | Zoom Out

72 Identified Inliers, $\begin{bmatrix} -0.7672 & 0.6397 & 0.0472 \\ 0.0648 & 0.1505 & -0.9865 \\ 0.6382 & 0.7537 & 0.1569 \end{bmatrix}$

(e) 3×10^{-6}

Original View | Zoom Out

60 Identified Inliers, $\begin{bmatrix} 0.6504 & 0.7594 & 0.0187 \\ -0.0343 & 0.0539 & -0.9980 \\ -0.7588 & 0.6484 & 0.0612 \end{bmatrix}$

(f) 4×10^{-6}

Summary

- Overview of Calibration
- Tsai's Method: From 3D Objects
- Zhang's Method: From Planar Grids
- Image Undistortion

Thank you for your listening!

If you have any questions, please come to me :-)