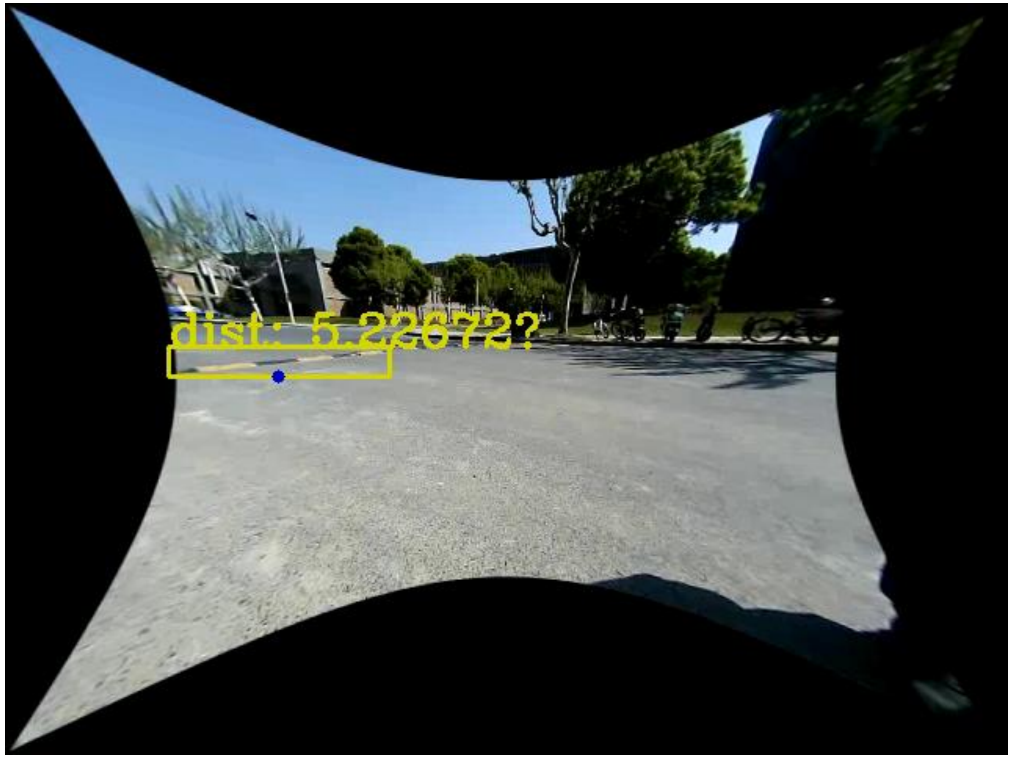
Lecture 6
Measurement Using a Single Camera

(All materials in this lecture are limited to a single camera)

Lin ZHANG, PhD
School of Software Engineering
Tongji University
Spring 2022

If I have an image containing a coin, can you tell me the diameter of that coin?


dist: 5.22672?


基于视觉的
泊车位检测
同济大学软件学院
计算视觉课题组
张林 李曦媛 黄君豪 李林申

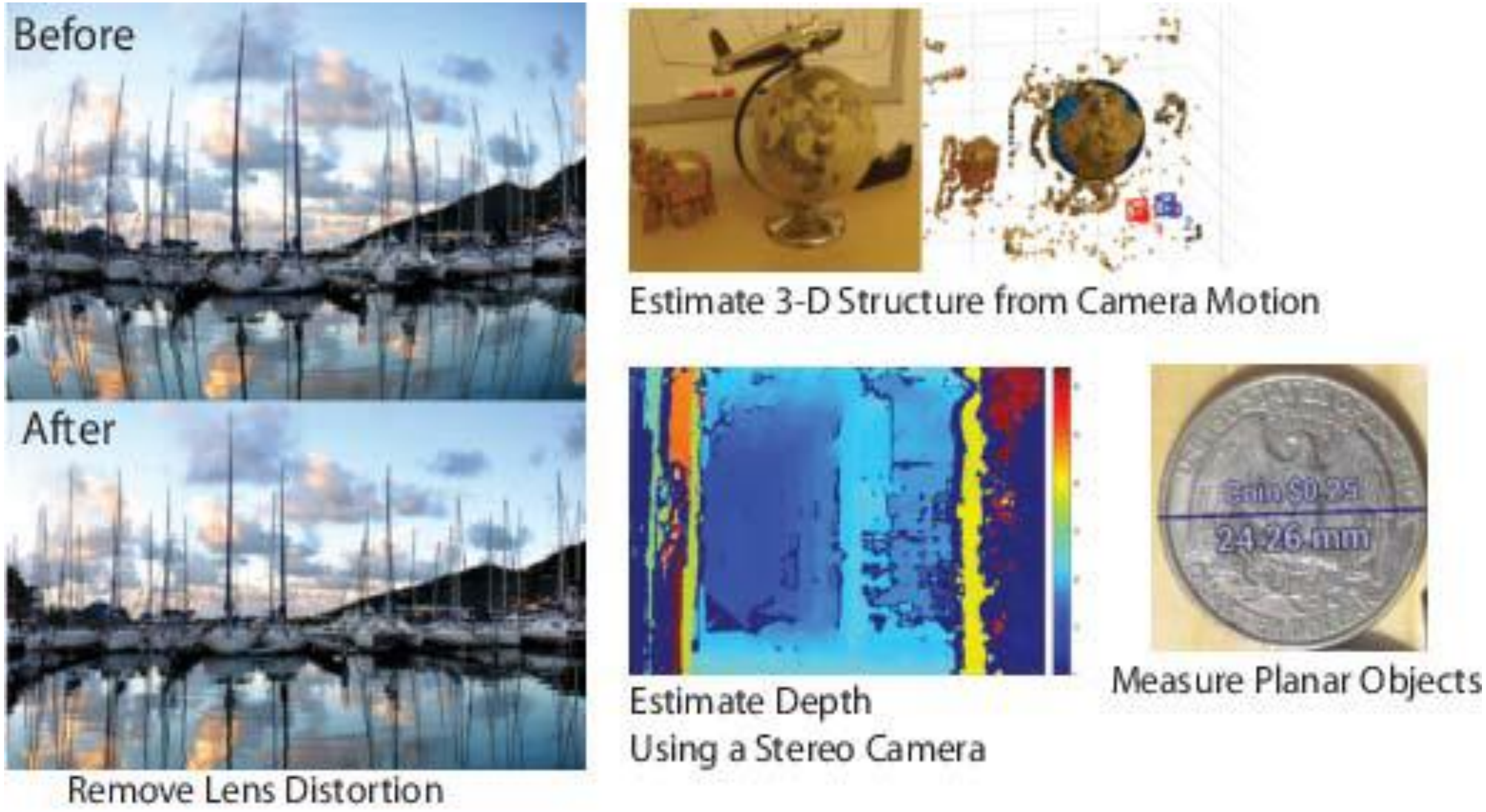
Outline

- What is Camera Calibration?
- Modeling for Imaging Pipeline
- General Framework for the Camera Calibration Algorithm
- Initial Rough Estimation of Calibration Parameters
- Nonlinear Least-squares
- Bird's-eye-view Generation

What is camera calibration?

- Camera calibration is a necessary step in 3D computer vision in order to extract metric information from 2D images
- It estimates the parameters of a lens and image sensor of the camera; you can use these parameters to correct for lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene
- These tasks are used in applications such as machine vision to detect and measure objects. They are also used in robotics, for navigation systems, and 3-D scene reconstruction

What is camera calibration?

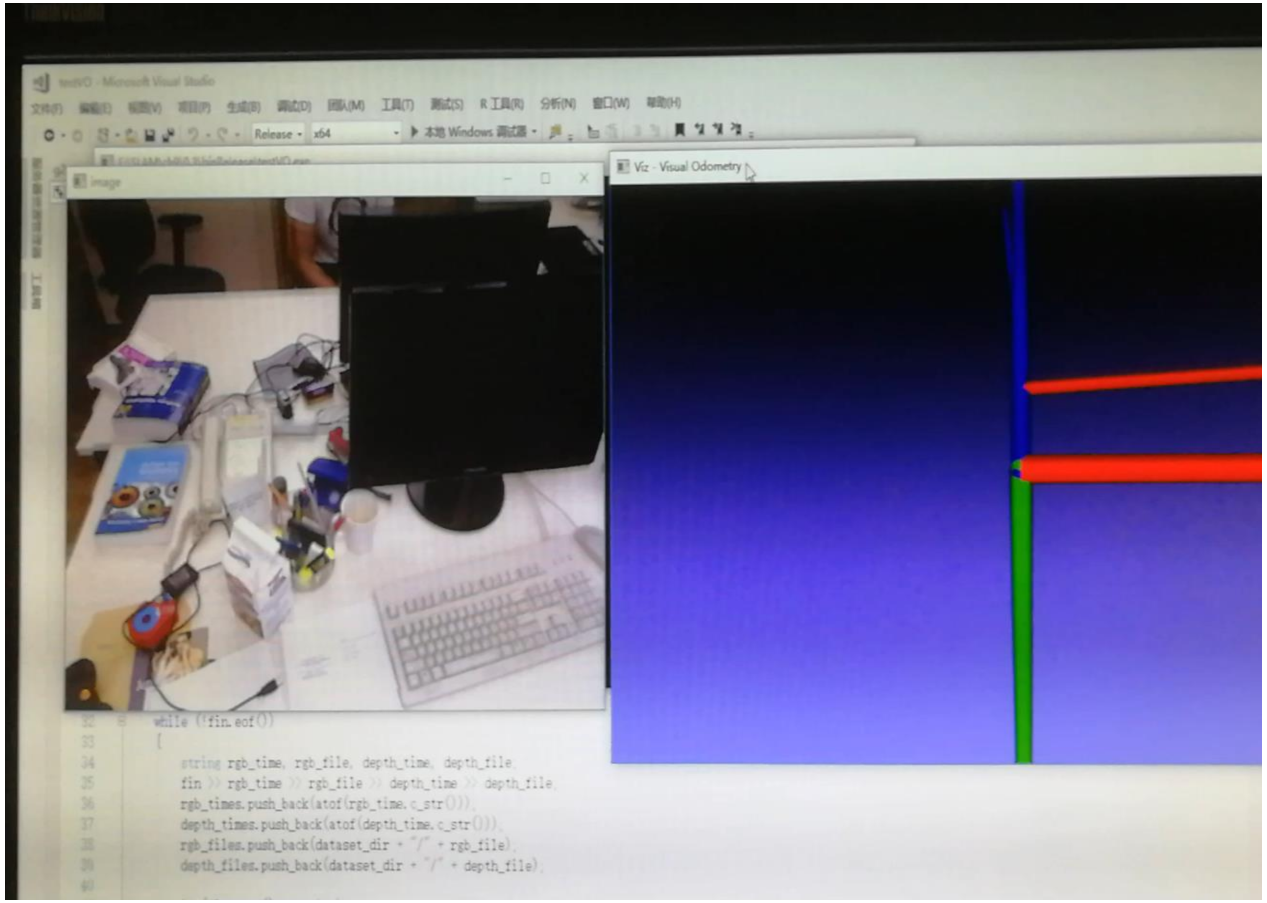


What is camera calibration?

Example: PnP (Perspective N Points) problem

Suppose a camera is calibrated (its intrinsics are known)

From a set of spatial points with known coordinates in the WCS and their pixel positions on the image, the pose of the camera with respect to the WCS can be recovered. This is a simple **visual odometry**.



What is camera calibration?

- Camera parameters include
  - Intrinsics
  - Distortion coefficients
  - Extrinsics

To perform single camera calibration, you need to know:

How to model the imaging process?

What is the general workflow for camera calibration?

How to get the initial estimation of parameters?

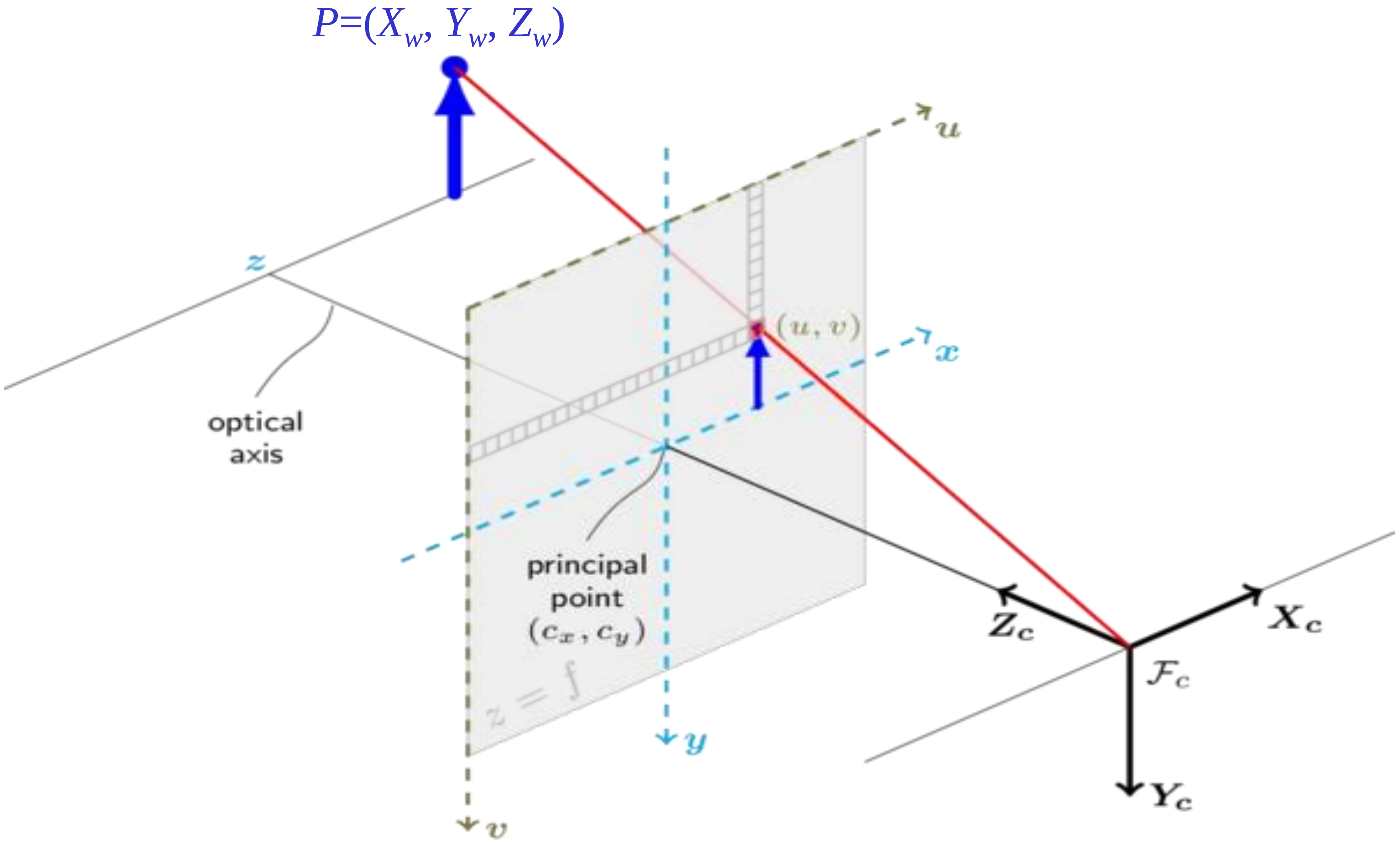
How to solve a nonlinear optimization problem?

Outline

- What is Camera Calibration?
- Modeling for Imaging Pipeline
- General Framework for the Camera Calibration Algorithm
- Initial Rough Estimation of Calibration Parameters
- Nonlinear Least-squares
- Bird's-eye-view Generation

Modeling for Imaging Pipeline

- For simplicity, usually we use a pinhole camera model



Modeling for Imaging Pipeline

Modeling for Imaging Pipeline

- To model the image formation process, 4 coordinate systems are required
  - World coordinate system (3D space)
  - Camera coordinate system (3D space)
  - Retinal coordinate system (2D space)
  - **Normalized retinal coordinate system (2D space)**
  - Pixel coordinate system (2D space)

Modeling for Imaging Pipeline

- From the world CS to the camera CS

$\left[X_w, Y_w, Z_w\right]^T$ is a 3D point represented in the WCS

In the camera CS, it is represented as,

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \mathbf{t} \Rightarrow \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \left[\mathbf{R} \quad \mathbf{t}\right]_{3\times 4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1)$$

a 3×1 translation vector

a 3×3 rotation matrix (orthogonal, $\det(\mathbf{R})=1$)

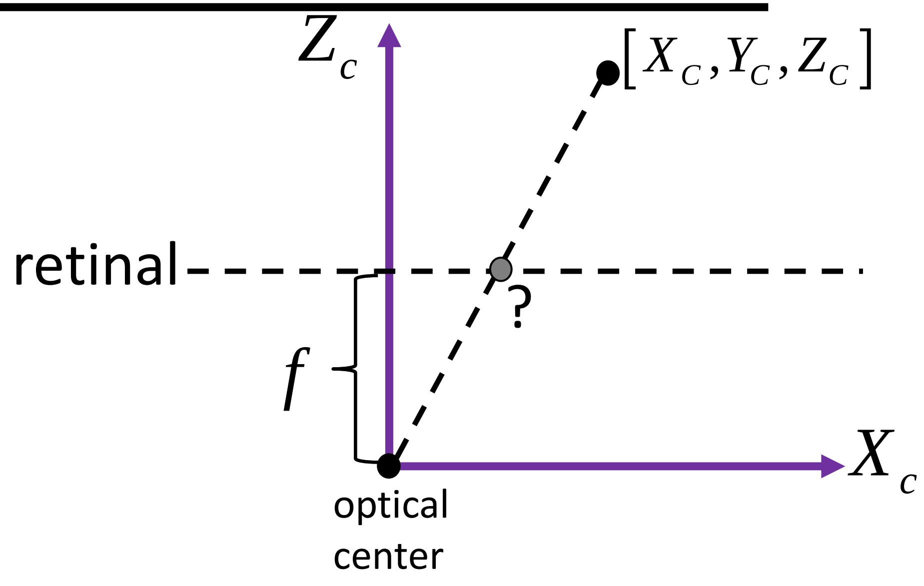
(inhomogeneous)

(normalized homogeneous)

Modeling for Imaging Pipeline

- From the camera CS to the retinal CS

We can use a pin-hole model to represent the mapping from the camera CS to the retinal CS



$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f\dfrac{X_C}{Z_C} \\ f\dfrac{Y_C}{Z_C} \end{bmatrix}$$

where f is the distance between the retinal plane and the camera origin
The retinal plane is perpendicular to the optical axis.

Note: From the view of the camera CS, the coordinates of the point (x, y) on the retinal plane are (x,y,f)

Modeling for Imaging Pipeline

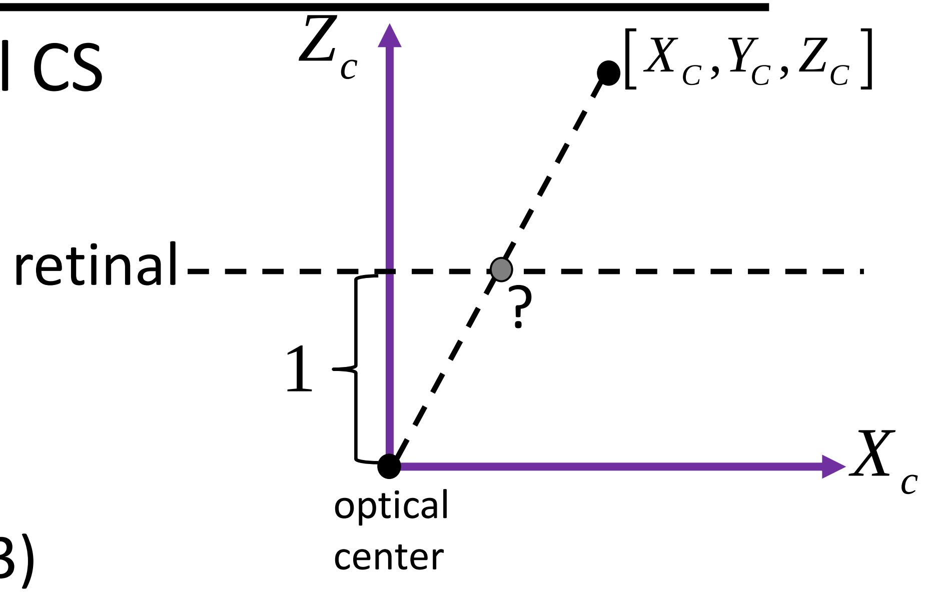
- From the camera CS to the retinal CS

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f\dfrac{X_C}{Z_C} \\ f\dfrac{Y_C}{Z_C} \end{bmatrix} \xrightarrow{\text{homogeneous form}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\text{normalized homogeneous}} = \frac{1}{Z_C}\begin{bmatrix} fX_C \\ fY_C \\ Z_C \end{bmatrix} = \frac{1}{Z_C}\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}\underbrace{\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix}}_{\text{inhomogeneous}} \quad (2)$$

Modeling for Imaging Pipeline

- *From the camera CS to the normalized retinal CS

Normalized retinal plane is a virtual plane with a distance 1 to the optical center



$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} \rightarrow \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \frac{X_C}{Z_C} \\ \frac{Y_C}{Z_C} \end{bmatrix} \xrightarrow{\text{homogeneous form}} \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} = \frac{1}{Z_C} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} \quad (3)$$

normalized homogeneous

Modeling for Imaging Pipeline

- From the retinal CS to the pixel CS

The unit for retinal CS (x-y) is physical unit (e.g., mm, cm) while the unit for pixel CS (u-v) is pixel

Suppose that one pixel represents dx physical units along the x-axis and represents dy physical units along the y-axis, and the image of the optical center is (c_x, c_y) (pixels)

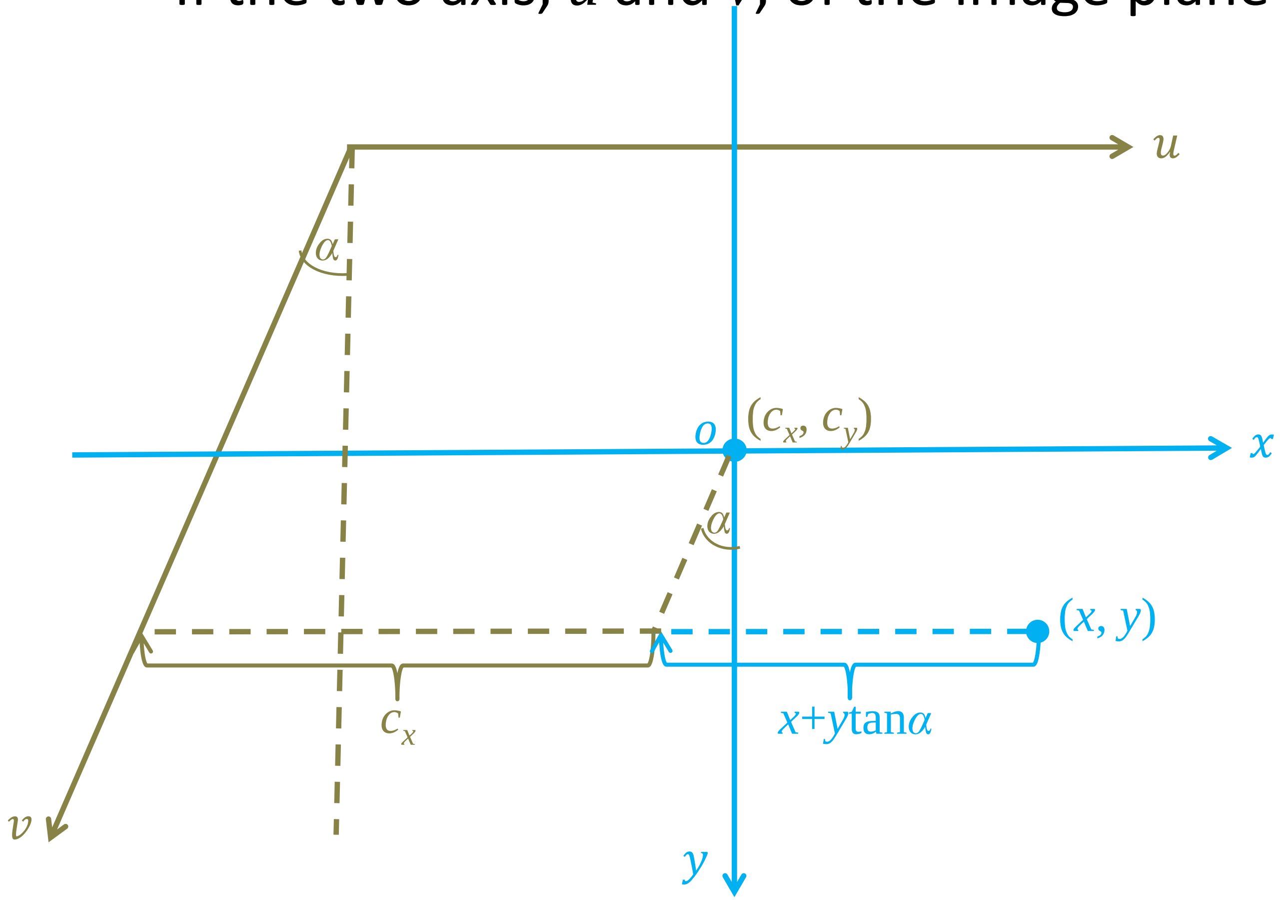
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & c_x \\ 0 & \frac{1}{dy} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

normalized homogeneous

Modeling for Imaging Pipeline

- From the retinal CS to the pixel CS

If the two axis, u and v, of the image plane are not perpendicular,



$$\begin{cases} u = c_x + \dfrac{x + y\tan\alpha}{dx} \\ v = c_y + \dfrac{y}{dy} \end{cases}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \dfrac{1}{dx} & \dfrac{\tan\alpha}{dx} & c_x \\ 0 & \dfrac{1}{dy} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (4)$$

Modeling for Imaging Pipeline

From Eqs.1, 2, and 4, we can have

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & \frac{\tan\alpha}{dx} & c_x \\ 0 & \frac{1}{dy} & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \frac{1}{Z_C} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \begin{bmatrix} \frac{f}{dx} & \frac{f\tan\alpha}{dx} & c_x \\ 0 & \frac{f}{dy} & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \frac{1}{Z_C} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} \triangleq \begin{bmatrix} f_x & f_x\tan\alpha & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \frac{1}{Z_C} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix}$$

$$= \frac{1}{Z_C} \underbrace{\begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinscis matrix}} \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{3\times 4}}_{\text{Extrinsics}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

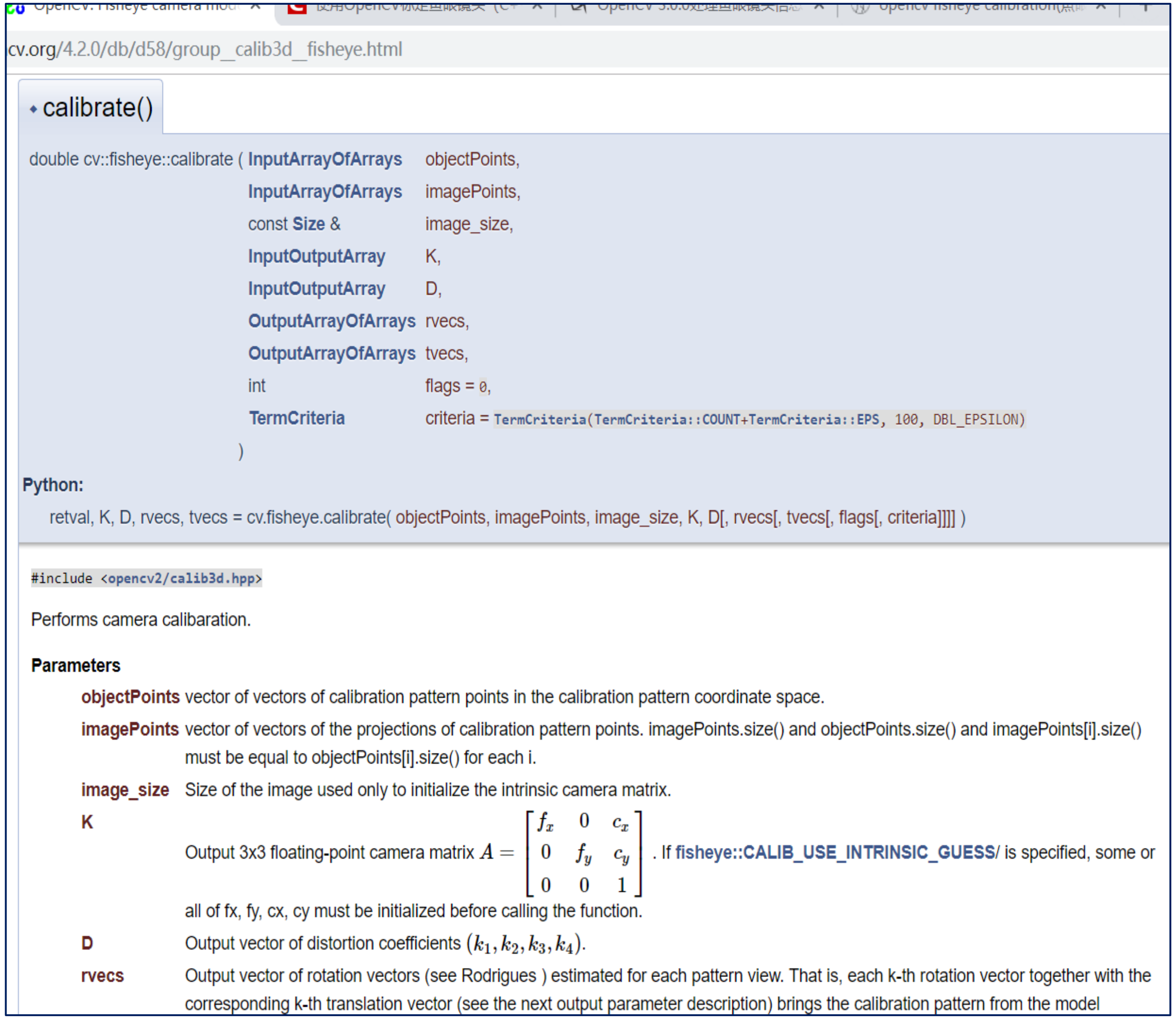
Image formation model without considering lens distortions,

$$\mathbf{u} = \frac{1}{Z_C} \cdot \mathbf{K}_{3\times 3} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{3\times 4} \mathbf{P}_{4\times 1} \qquad (5)$$

Note: $\mathbf{u}$ is the normalized homogeneous coordinates

Modeling for Imaging Pipeline

- Some notes about the intrinsic matrix in practical use
  - In matlab, the skew parameter s is modeled
  - In openCV, for ordinary cameras, s is not modeled, meaning that it only considers four intrinsic parameters
  - In openCV, for fisheye cameras, s is modeled (after calibrating the fisheye cameras, you really can get five parameters); **However, the related document has a mistake by saying that only four intrinsic parameters are considered**



cv.org/4.2.0/db/d58/group__calib3d__fisheye.html

◆ calibrate()

```
double cv::fisheye::calibrate ( InputArrayOfArrays    objectPoints,
                                InputArrayOfArrays    imagePoints,
                                const Size &          image_size,
                                InputOutputArray      K,
                                InputOutputArray      D,
                                OutputArrayOfArrays   rvecs,
                                OutputArrayOfArrays   tvecs,
                                int                   flags = 0,
                                TermCriteria          criteria = TermCriteria(TermCriteria::COUNT+TermCriteria::EPS, 100, DBL_EPSILON)
                              )
```

Python:

```
retval, K, D, rvecs, tvecs = cv.fisheye.calibrate( objectPoints, imagePoints, image_size, K, D[, rvecs[, tvecs[, flags[, criteria]]]] )
```

`#include <opencv2/calib3d.hpp>`

Performs camera calibration.

Parameters

- **objectPoints** vector of vectors of calibration pattern points in the calibration pattern coordinate space.
- **imagePoints** vector of vectors of the projections of calibration pattern points. imagePoints.size() and objectPoints.size() and imagePoints[i].size() must be equal to objectPoints[i].size() for each i.
- **image_size** Size of the image used only to initialize the intrinsic camera matrix.
- **K** Output 3x3 floating-point camera matrix $A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$. If fisheye::CALIB_USE_INTRINSIC_GUESS/ is specified, some or all of fx, fy, cx, cy must be initialized before calling the function.
- **D** Output vector of distortion coefficients (k_1, k_2, k_3, k_4).
- **rvecs** Output vector of rotation vectors (see Rodrigues) estimated for each pattern view. That is, each k-th rotation vector together with the corresponding k-th translation vector (see the next output parameter description) brings the calibration pattern from the model

Note: In this course, we do not consider s anymore

Modeling for Imaging Pipeline

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \frac{1}{Z_C} [\mathbf{R} \quad \mathbf{t}]_{3\times 4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Point on the normalized retinal CS

$$\frac{1}{Z_C} [\mathbf{R} \quad \mathbf{t}]_{3\times 4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}$$ (according to Eq.3)

Thus, we have a byproduct which states the relationship between the coordinates on the pixel CS and the coordinates on the normalized retinal CS,

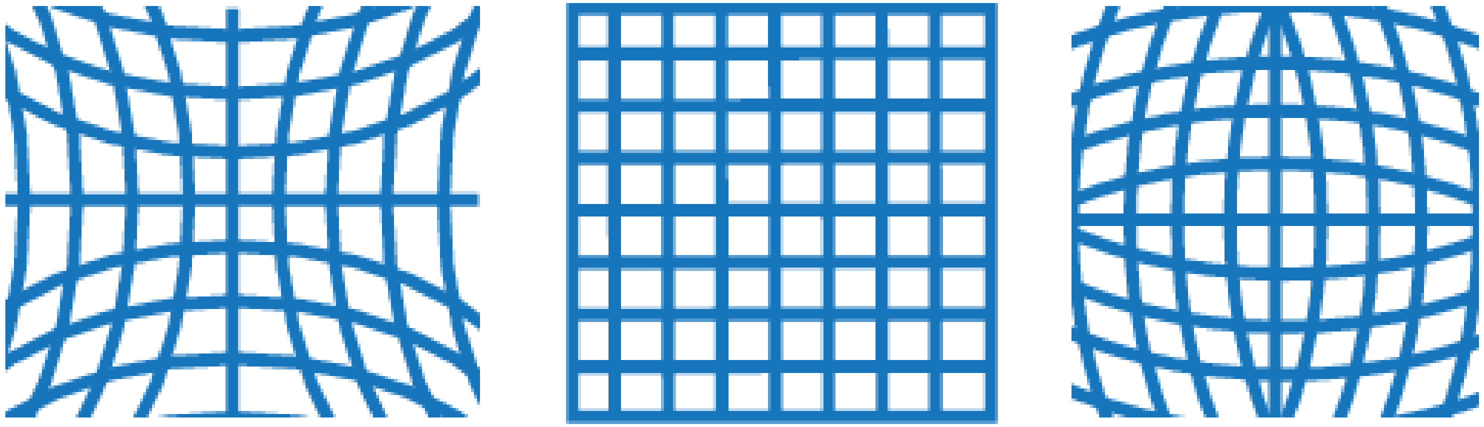
Normalized homogeneous

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} \quad (6)$$

Normalized homogeneous

Modeling for Imaging Pipeline

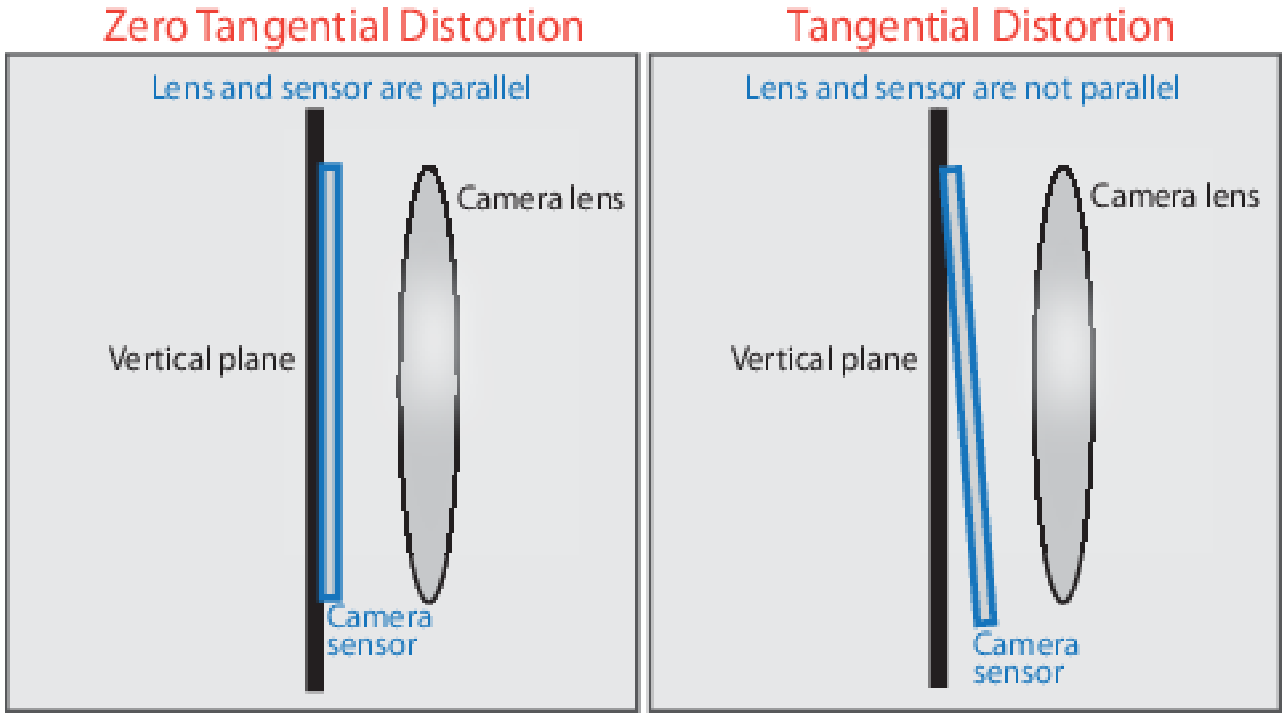
- To model the behavior of lens, we need to consider the distortion
  - Radial distortion occurs when light rays bend more near the edges of a lens than they do at its optical center; the smaller the lens, the greater the distortion



Negative radial distortion "pincushion" | No distortion | Positive radial distortion "barrel"

Modeling for Imaging Pipeline

- To model the behavior of lens, we need to consider the distortion
  - Tangential distortion occurs when the lens and the image plane are not parallel



Modeling for Imaging Pipeline

- To model the behavior of lens, we need to consider the distortion
  - Both the two types of distortions are modeled on the **normalized retinal plane**

To model radial distortion

$$x_{dr} = x_n\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right)$$

$$y_{dr} = y_n\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right)$$

where $r^2 = x_n^2 + y_n^2$

k_1, k_2, k_3 are the radial distortion coefficients

To model tangential distortion

$$x_{dt} = x_n + \left(2\rho_1 x_n y_n + \rho_2\left(r^2 + 2x_n^2\right)\right)$$

$$y_{dt} = y_n + \left(2\rho_2 x_n y_n + \rho_1\left(r^2 + 2y_n^2\right)\right)$$

where $r^2 = x_n^2 + y_n^2$

ρ_1, ρ_2 are the tangential distortion coefficients

If they both need to be considered,

$$\begin{cases} x_d = x_n\left(1 + k_1 r^2 + k_2 r^4\right) + 2\rho_1 x_n y_n + \rho_2\left(r^2 + 2x_n^2\right) + x_n k_3 r^6 \\ y_d = y_n\left(1 + k_1 r^2 + k_2 r^4\right) + 2\rho_2 x_n y_n + \rho_1\left(r^2 + 2y_n^2\right) + y_n k_3 r^6 \end{cases} \quad (7)$$

Note: This step cannot be represented by matrix multiplication

Modeling for Imaging Pipeline

- To model the behavior of lens, we need to consider the distortion
  - Both the two types of distortions are modeled on the **normalized retinal plane**
  - If the FOV is extremely large (larger than 100 degrees), i.g. the camera is a fisheye camera, we need to use another model to characterize lens distortions

A typical image collected by a fisheye camera

Modeling for Imaging Pipeline

- To model the behavior of lens, we need to consider the distortion
  - Both the two types of distortions are modeled on the **normalized retinal plane**
  - If the FOV is extremely large (larger than 100 degrees), i.e. the camera is a fisheye camera, we need to use another model to characterize lens distortions

To model the fisheye distortion

$$\theta = \arctan(r)$$

$$\theta_d = \theta\left(1 + k_1\theta^2 + k_2\theta^4 + k_3\theta^6 + k_4\theta^8\right)$$

$$x_d = \frac{\theta_d}{r} x_n$$

$$y_d = \frac{\theta_d}{r} y_n$$

where $r^2 = x_n^2 + y_n^2$

k_1, k_2, k_3, k_4 are the distortion coefficients

Modeling for Imaging Pipeline

The complete imaging pipeline is modeled as,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \underbrace{\mathcal{D}\left\{ \overbrace{\frac{1}{Z_C} [\mathbf{R} \;\; \mathbf{t}]_{3\times 4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}^{\text{point on the normalized retinal plane}} \right\}}_{\text{distorted point on the normalized retinal plane}} \quad (8)$$

point on the normalized retinal plane

distorted point on the normalized retinal plane

$f_x, f_y, c_x, c_y, k_1, k_2, \rho_1, \rho_2, k_3$ are the intrinsics of the camera (suppose it is an ordinary camera)

$\mathbf{R}$ (three DOFs) and $\mathbf{t}$ (three DOFs) are the extrinsics of the camera

Modeling for Imaging Pipeline

- The process to get the intrinsics and extrinsics of the camera is called single camera calibration
  - For most cases of single camera calibration, only the intrinsics are what we really need
- To model radial and tangential distortions, we use 5 parameters; Actually, more complicated models can be used, but the modeling pipeline is the same
  - E.g. the thin prism model, the tilted model used in OpenCV

Outline

- What is Camera Calibration?
- Modeling for Imaging Pipeline
- General Framework for the Camera Calibration Algorithm
- Initial Rough Estimation of Calibration Parameters
- Nonlinear Least-squares
- Bird’s-eye-view Generation

General Framework for the Camera Calibration Algorithm

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \mathcal{D} \left\{ \frac{1}{Z_C} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{3\times 4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \right\}$$ (Eq. 8, the imaging pipeline)

- General idea
  - If we have a set of known points $\{\mathbf{P}_i\}_{i=1}^{n}$ in the WCS and their images $\{\mathbf{u}_i\}_{i=1}^{n}$, using Eq. 8, we could have $2n$ equations
  - If the number of valid constraints (equations) are enough, Eq. 8 could be solved
- All the calibration algorithms follow the above general rules and among them, Zhengyou Zhang's idea[1] is the most widely used

[1] Z. Zhang, A flexible new technique for camera calibration, IEEE Trans. Pattern Analysis and Machine Intelligence, 2000

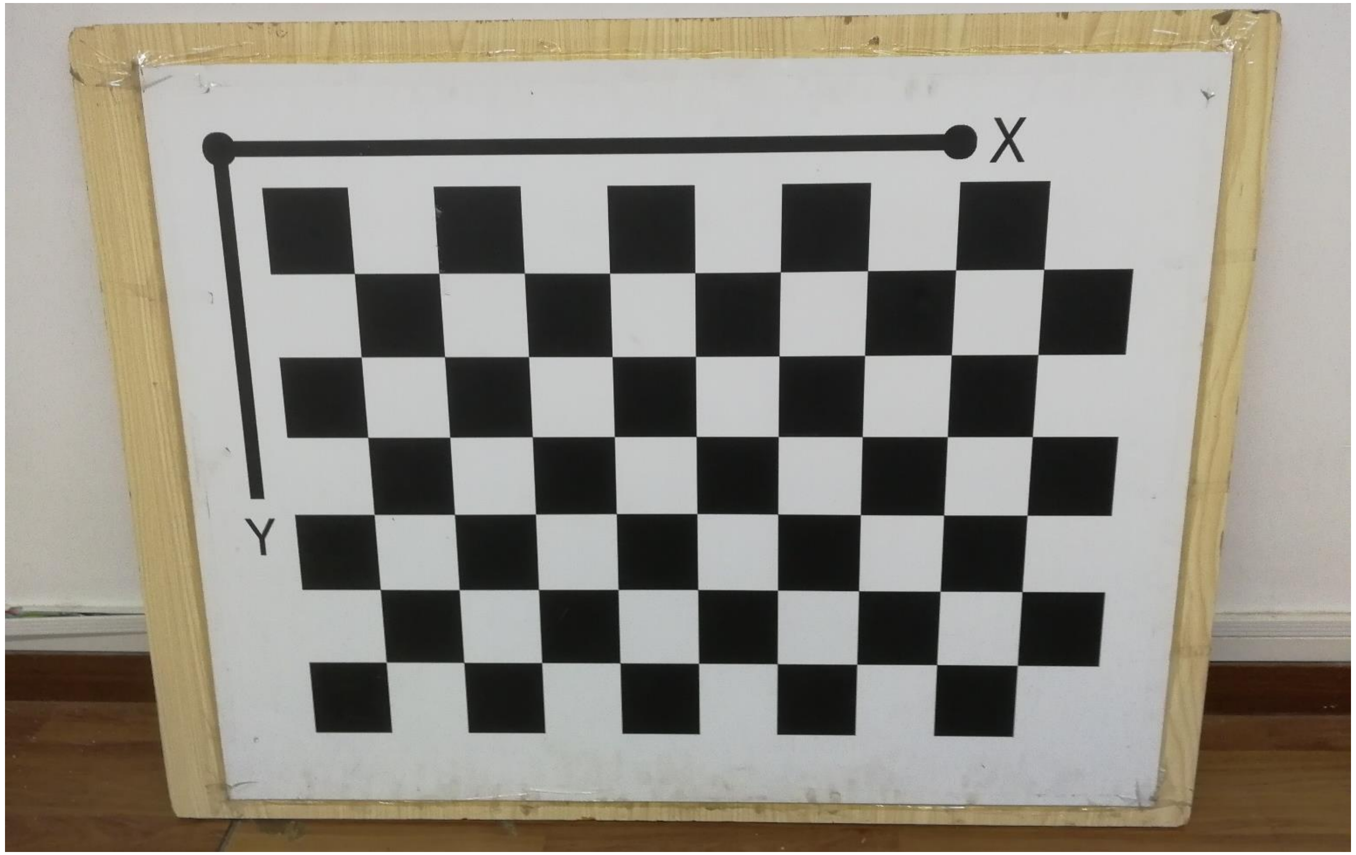
General Framework for the Camera Calibration Algorithm

- Zhengyou Zhang's calibration approach
  - A calibration board with a chessboard pattern is needed
  - Several images of the board need to be captured
  - Detect the feature points (cross points) in the images
  - Based on the correspondence pairs (pixel coordinate and world coordinate of a feature point), equation systems can be obtained
  - By solving the equation systems, parameters can be determined

Aug. 1, 1965~, now is the director of Tencent AI Lab

General Framework for the Camera Calibration Algorithm

- Zhengyou Zhang's calibration approach

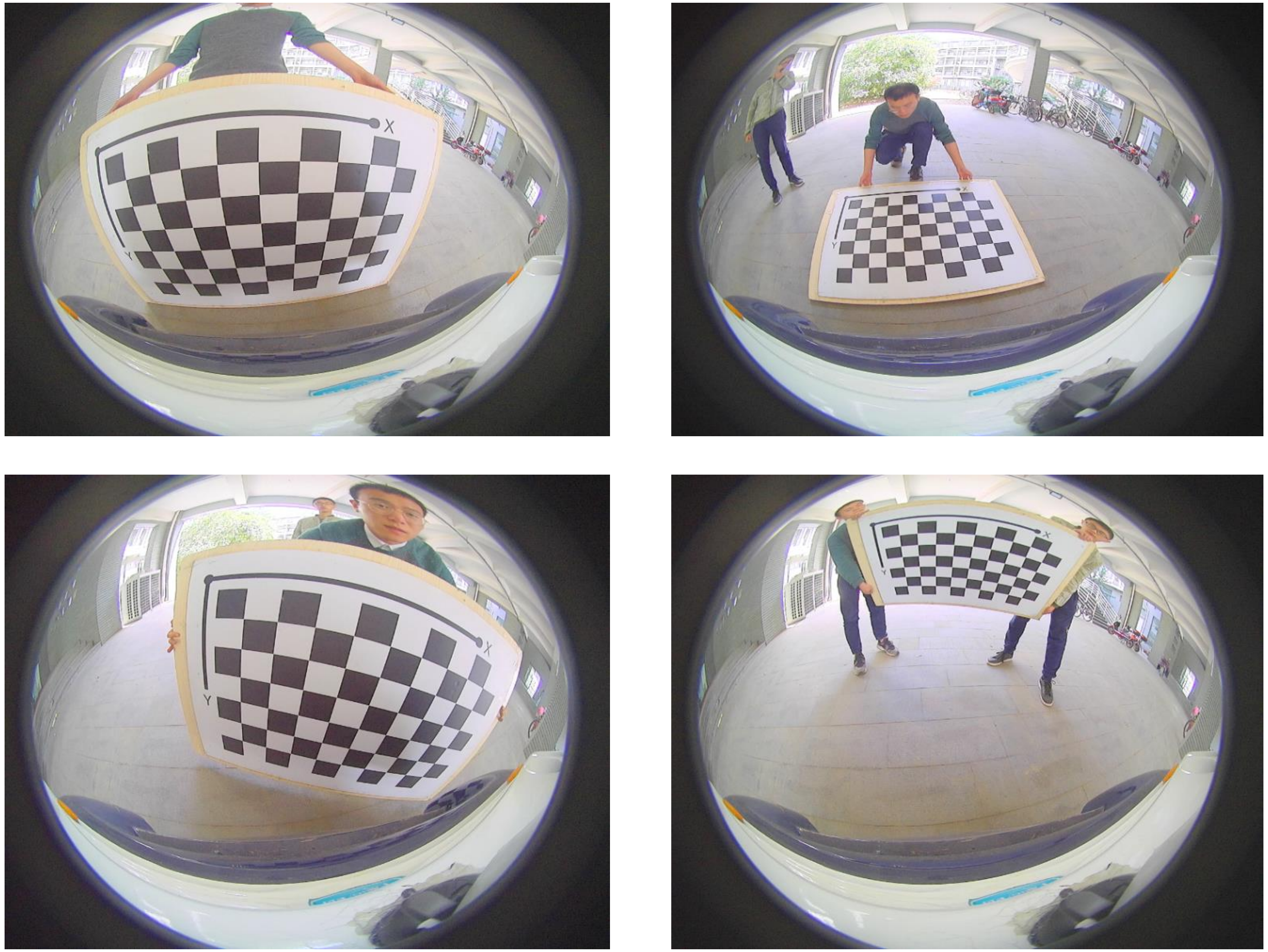


Calibration board

The number of blocks of one side should be even and the number of blocks of the other side should be odd

General Framework for the Camera Calibration Algorithm

- Zhengyou Zhang’s calibration approach



A set of Calibration board images (20~30)

General Framework for the Camera Calibration Algorithm

Suppose we have M board images and for each image we have N cross points, then the calibration amounts to the following optimization problem,

$$\Theta^* = \arg\min_{\Theta} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{1}{2} \left\| \mathbf{K} \cdot \mathcal{D}\left\{ \frac{1}{Z_{Cij}} \left[\mathbf{R}_i \ \ \mathbf{t}_i \right] \mathbf{P}_j \right\} - \mathbf{u}_{ij} \right\|_2^2 \tag{9}$$

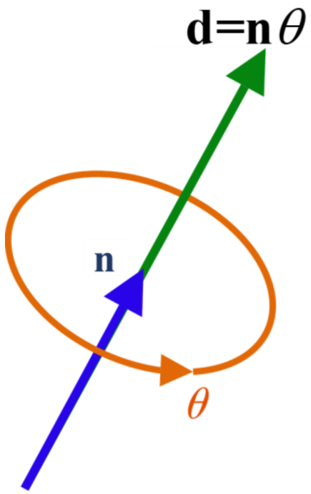
where $\Theta = \left\{ f_x, f_y, c_x, c_y, k_1, k_2, \rho_1, \rho_2, k_3, \left\{ \mathbf{R}_i \right\}_{i=1}^{M}, \left\{ \mathbf{t}_i \right\}_{i=1}^{M} \right\}$ denotes the parameters that needs to be optimized

$\mathbf{P}_j$ is the WCS coordinates (determined by the physical calibration board) of the jth cross-point, and $\mathbf{u}_{ij}$ is its projection (pixel coordinate) on ith image

$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$ denotes the intrinsics matrix

General Framework for the Camera Calibration Algorithm

- About the rotation
  - In 3D Euclidean space, a rotation has 3 DOFs (three Euler angles)
  - If we use a 3*3 matrix to denote the rotation, we must add extra constraints (the matrix should be orthonormal and its determinant should be 1) and that will make the optimization complicated
  - Thus, in all modern implementations, a rotation is finally represented by **axis-angle**



$\mathbf{n}$ is a unit 3D vector describing an axis of rotation according to the right hand rule; θ is the rotation angle

$\mathbf{d}=\mathbf{n}\theta$, a 3D vector denoting the rotation is called axis-angle

General Framework for the Camera Calibration Algorithm

- About the rotation
  - Axis-angle can be uniquely converted to a rotation matrix and vice versa via **Rodrigues** formula

From axis-angle $\mathbf{d}=\mathbf{n}\theta$ to rotation matrix $\mathbf{R}$

$$\mathbf{R} = \cos\theta \mathbf{I} + (1-\cos\theta)\mathbf{n}\mathbf{n}^T + \sin\theta \mathbf{n}^\wedge \quad (10)$$

where $\mathbf{I}$ is the identity matrix and

$$\mathbf{n}^\wedge = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}$$

From rotation matrix $\mathbf{R}$ to axis-angle $\mathbf{d}=\mathbf{n}\theta$

$$\theta = \arccos\left(\frac{tr(\mathbf{R})-1}{2}\right)$$

$$\mathbf{R}\mathbf{n} = \mathbf{n}$$

i.e., $\mathbf{n}$ is the eigenvector of $\mathbf{R}$ associated with the eigenvalue 1

General Framework for the Camera Calibration Algorithm

Suppose we have M board images and for each image we have N cross points, then the calibration amounts to the following optimization problem,

$$\Theta^* = \arg\min_{\Theta} \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{1}{2}\left\| \mathbf{K}\cdot\mathcal{D}\left\{\frac{1}{Z_{Cij}}\left[\mathbf{R}_i \ \ \mathbf{t}_i\right]\mathbf{P}_j\right\} - \mathbf{u}_{ij}\right\|_2^2 \qquad (9)$$

where the parameters that need to be optimized are,

$$\Theta = \left\{ f_x, f_y, c_x, c_y, k_1, k_2, \rho_1, \rho_2, k_3, \left\{\mathbf{d}_i\right\}_{i=1}^{M}, \left\{\mathbf{t}_i\right\}_{i=1}^{M} \right\}$$ ($\mathbf{d}_i$ is the axis-angle representation of $\mathbf{R}_i$)

Altogether, we have $2\times M\times N$ equations (error terms) and $9+6M$ unknown parameters

Eq. 9 is a nonlinear optimization problem and does not have a closed-form solution It can be solved by iterative methods. But before that, we need to have a good starting point, i.g. we need to have a rough estimate to Θ

Outline

- What is Camera Calibration?
- Modeling for Imaging Pipeline
- General Framework for the Camera Calibration Algorithm
- Initial Rough Estimation of Calibration Parameters
- Nonlinear Least-squares
- Bird's-eye-view Generation

Initial Rough Estimation of Calibration Parameters

- The task at this step
  - Given us a set of M images of planar calibration board, estimate the intrinsics (except the ones related to distortion) of the camera and the extrinsics of the camera poses when taking each image

Initial Rough Estimation of Calibration Parameters

- The task at this step
  - Given us a set of M images of planar calibration board, estimate the intrinsics (except the ones related to distortion) of the camera and the extrinsics of the camera poses when taking each image

$$\Theta = \left\{ f_x, f_y, c_x, c_y, \underbrace{k_1, k_2, \rho_1, \rho_2, k_3}_{\text{Distortion coefficients can be safely initialized as zeros}}, \left\{\mathbf{d}_i\right\}_{i=1}^{M}, \left\{\mathbf{t}_i\right\}_{i=1}^{M} \right\}$$

Thus, in initial estimation of other parameters, we use the imaging model without considering distortions,

$$\mathbf{u} = \frac{1}{Z_C} \cdot \mathbf{K}_{3\times 3} \left[\mathbf{R} \;\; \mathbf{t}\right]_{3\times 4} \mathbf{P}_{4\times 1} \qquad \text{(Eq. 5)}$$

Given a calibration board, $\mathbf{P}$ is a cross-point on it, thus $\mathbf{P}$ has the form $\mathbf{P} = \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$

Initial Rough Estimation of Calibration Parameters

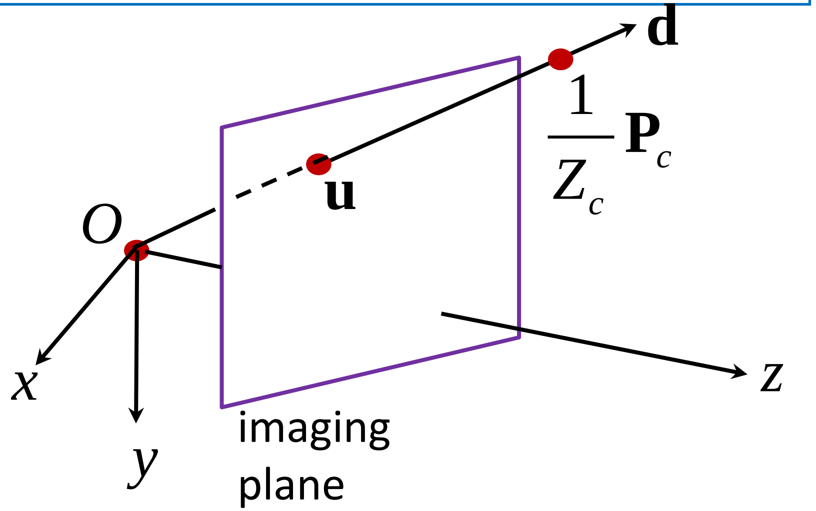
Result 1: In the camera coordinate system, the direction of the ray pointing from the optical center O to the pixel $\mathbf{u}$ (in homogeneous form) on the imaging plane is

$$\mathbf{d} = \mathbf{K}^{-1}\mathbf{u}$$

The imaging model is

$$\mathbf{u} = \frac{1}{Z_C}\mathbf{K}_{3\times3}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}_{3\times4}\mathbf{P}_{4\times1}$$ (Eq. 5, $\mathbf{u}$ is normalized homogeneous)

$$\mathbf{K}^{-1}\mathbf{u} = \frac{1}{Z_c}\begin{bmatrix}X_c\\Y_c\\Z_c\end{bmatrix} = \begin{bmatrix}x_n\\y_n\\1\end{bmatrix} \triangleq \mathbf{x}_n$$



Since $\mathbf{x}_n$ should on the ray $\overrightarrow{O\mathbf{u}}$ ⇒ $\overrightarrow{O\mathbf{u}}$'s direction is $\mathbf{d} = \overrightarrow{\mathbf{x}_n - \mathbf{0}} = \mathbf{K}^{-1}\mathbf{u}$

Actually, any $k\mathbf{K}^{-1}\mathbf{u} = \mathbf{K}^{-1}(k\mathbf{u})\ (k \neq 0)$ can represent the direction of $\mathbf{d}$

⇓

$\mathbf{u}$ actually does not need to be normalized homogeneous

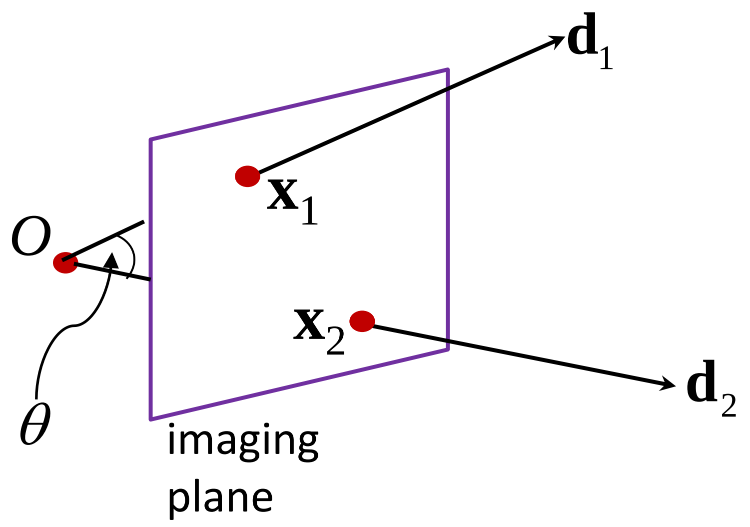
Initial Rough Estimation of Calibration Parameters

Result 2: In the camera coordinate system, the angle between two rays, pointing from O to $\mathbf{x}_1$ and $\mathbf{x}_2$ ($\mathbf{x}_1$ and $\mathbf{x}_2$ are the homogeneous coordinates of two pixels on the imaging plane), respectively, is determined as,

$$\cos\theta = \frac{\mathbf{d}_1 \cdot \mathbf{d}_2}{\|\mathbf{d}_1\|\|\mathbf{d}_2\|}$$

$$= \frac{\left(\mathbf{K}^{-1}\mathbf{x}_1\right)^T \mathbf{K}^{-1}\mathbf{x}_2}{\sqrt{\left(\mathbf{K}^{-1}\mathbf{x}_1\right)^T \left(\mathbf{K}^{-1}\mathbf{x}_1\right)}\sqrt{\left(\mathbf{K}^{-1}\mathbf{x}_2\right)^T \left(\mathbf{K}^{-1}\mathbf{x}_2\right)}}$$

$$= \frac{\mathbf{x}_1^T\left(\mathbf{K}^{-T}\mathbf{K}^{-1}\right)\mathbf{x}_2}{\sqrt{\mathbf{x}_1^T\left(\mathbf{K}^{-T}\mathbf{K}^{-1}\right)\mathbf{x}_1}\sqrt{\mathbf{x}_2^T\left(\mathbf{K}^{-T}\mathbf{K}^{-1}\right)\mathbf{x}_2}}$$



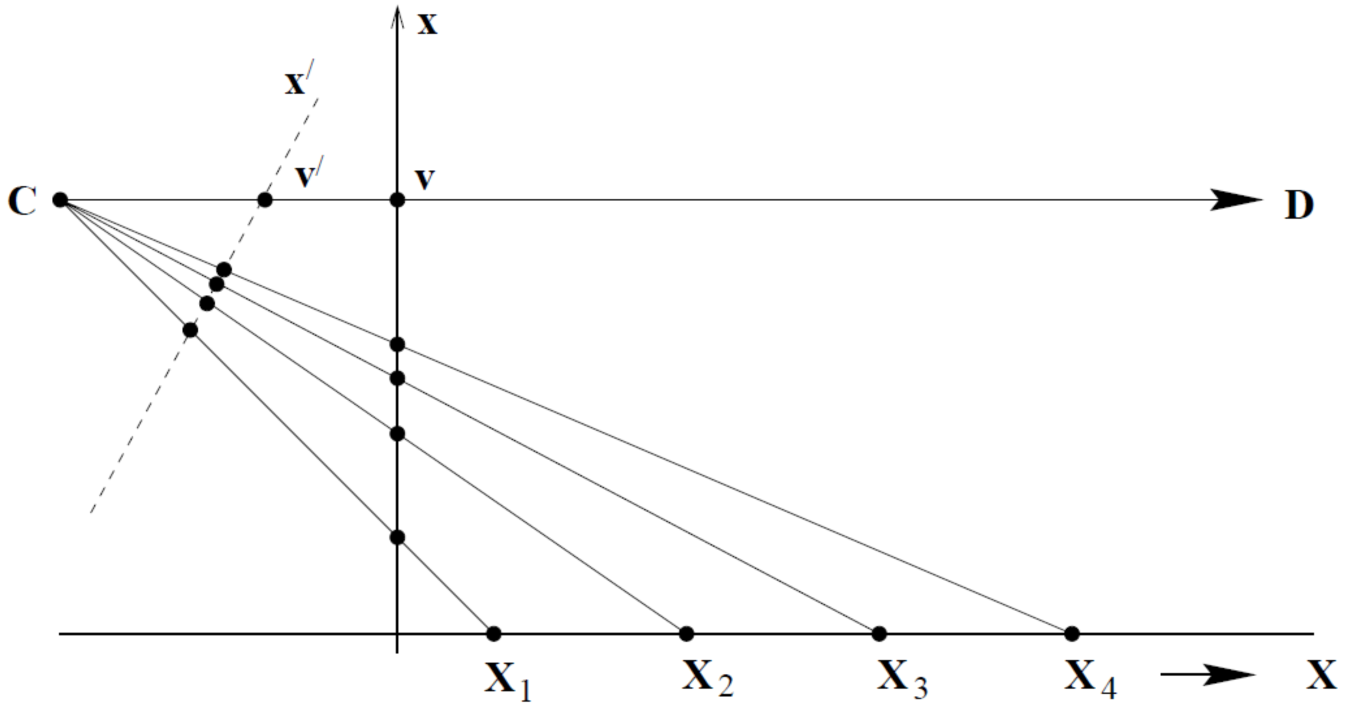
Initial Rough Estimation of Calibration Parameters

- Vanishing points
  - A feature of perspective projection is that the image of an object that stretches off to infinity can have finite extent. E.g., an infinite scene line is imaged as a line terminating in a vanishing point
  - Parallel world lines, such as railway lines, are imaged as converging lines and their image intersection is the **vanishing point for the direction of the railway**
  - **Vanishing point:** the vanishing point of a world line l is obtained by intersecting the image plane with a ray parallel to l and passing through the camera center

  Another definition: the vanishing point of a world line l is the image of l's infinity point on the imaging plane

Initial Rough Estimation of Calibration Parameters

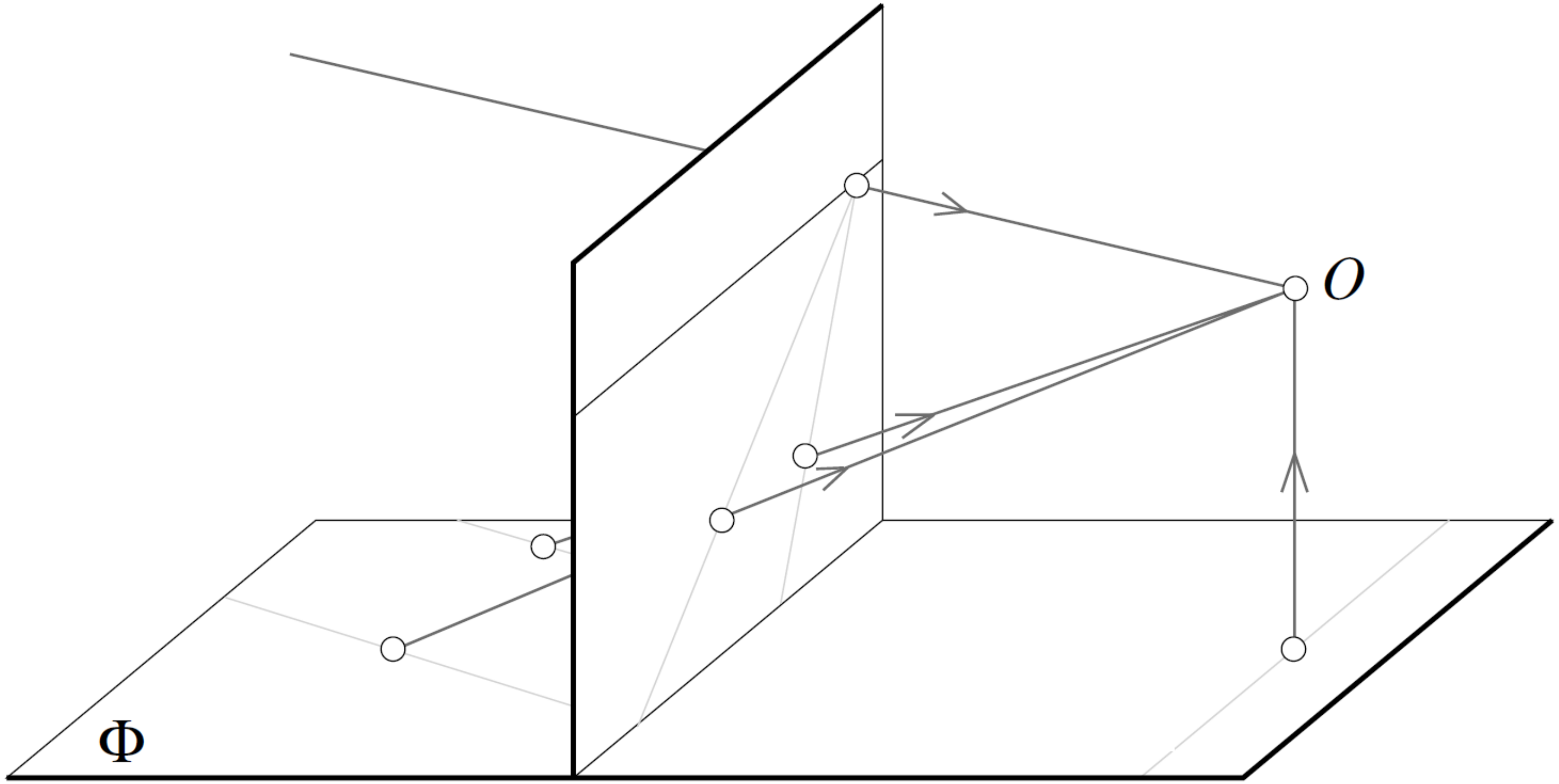
- Vanishing points: illustrations



The points $\mathbf{X}_i$, $i = 1, \ldots, 4$ are equally spaced on the world line, but their spacing on the image line monotonically decreases. In the limit $\mathbf{X} \rightarrow \infty$, the world point is imaged at $\mathbf{x} = \mathbf{v}$ on the vertical image line, and at $\mathbf{x}' = \mathbf{v}'$ on the inclined image line. Thus the vanishing point of the world line is obtained by intersecting the image plane with a ray parallel to the world line through the camera centre $\mathbf{C}$.

Initial Rough Estimation of Calibration Parameters

- Vanishing points: illustrations



Two parallel world lines should have the same infinity point; For each line, its vanishing point is the image of its infinity point; so the images of two parallel world lines would converge to the same vanishing point

Initial Rough Estimation of Calibration Parameters

Initial Rough Estimation of Calibration Parameters

JingHu High-speed railway: rails will “meet” at the vanishing point

Initial Rough Estimation of Calibration Parameters

- Properties of vanishing points
  - The vanishing point is on the imaging plane (indicating that it is expressed in pixels)
  - The vanishing point of the world line l depends only on its direction
  - A set of parallel world lines have a common vanishing point on the imaging plane
  - The ray $O\mathbf{v}$ (O is the optical center) is parallel to the world lines who share the same vanishing point $\mathbf{v}$

Result 3: l_1 and l_2 are two world lines and $\mathbf{v}_1$ and $\mathbf{v}_2$ are their vanishing points on the imaging plane, respectively. O is the optical center. Then, we have

$$\theta = l_1 l_2 = O\mathbf{v}_1 O\mathbf{v}_2$$

and

$$\cos\theta = \frac{\mathbf{v}_1^T\left(\mathbf{K}^{-T}\mathbf{K}^{-1}\right)\mathbf{v}_2}{\sqrt{\mathbf{v}_1^T\left(\mathbf{K}^{-T}\mathbf{K}^{-1}\right)\mathbf{v}_1}\sqrt{\mathbf{v}_2^T\left(\mathbf{K}^{-T}\mathbf{K}^{-1}\right)\mathbf{v}_2}}$$

(Using Result 2)

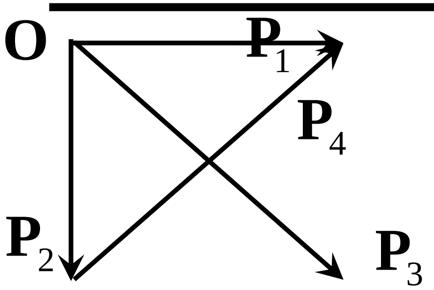
Initial Rough Estimation of Calibration Parameters

Result 4: l_1 and l_2 are two world lines perpendicular to each other, and $\mathbf{v}_1$ and $\mathbf{v}_2$ are their vanishing points on the imaging plane, respectively. We have

$$\mathbf{v}_1^T \left(\mathbf{K}^{-T} \mathbf{K}^{-1} \right) \mathbf{v}_2 = 0 \qquad \text{(Using Result 3)}$$

Note: This is a key result based on which camera calibration schemes roughly estimate camera's intrinsics

Initial Rough Estimation of Calibration Parameters



On the projective plane defined by the calibration boards, consider the four lines,

l_1: X-axis, the infinity point is $\mathbf{P}_1$= (1,0,0), its direction is $\overrightarrow{\mathbf{OP}_1}$

l_2: Y-axis, the infinity point is $\mathbf{P}_2$= (0,1,0), its direction is $\overrightarrow{\mathbf{OP}_2}$

l_3: line(s) with the infinity point $\mathbf{P}_3 = \mathbf{P}_1 + \mathbf{P}_2 = (1,1,0)$, its direction is $\overrightarrow{\mathbf{OP}_3}$

l_4: line(s) with the infinity point $\mathbf{P}_4 = \mathbf{P}_1 - \mathbf{P}_2 = (1,-1,0)$, its direction is $\overrightarrow{\mathbf{OP}_4}$

It can be verified: $l_1 \perp l_2$, $l_3 \perp l_4$

Initial Rough Estimation of Calibration Parameters

The plane of the calibration board and its image is linked via a homography

$$c\mathbf{u}_{3\times1} = \mathbf{H}_{3\times3}\mathbf{P}_{3\times1}$$

point on the image

homogeneous planar point on the calibration board

$\mathbf{H}$ can be estimated in advance for each calibration board image using the techniques introduced in Lect. 3

Denote $\mathbf{H}$ by,

$$\mathbf{H}_{3\times3} = \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix}$$

Images of $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$ are,

$$\mathbf{v}_1 = \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix} = \mathbf{h}_1, \mathbf{v}_2 = \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix} = \mathbf{h}_2$$

$$\mathbf{v}_3 = \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} = \mathbf{h}_1 + \mathbf{h}_2$$

$$\mathbf{v}_4 = \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix}\begin{bmatrix}1\\-1\\0\end{bmatrix} = \mathbf{h}_1 - \mathbf{h}_2$$

Initial Rough Estimation of Calibration Parameters

Based on Result 4, we have

$$\begin{cases} \mathbf{v}_1^T \left(\mathbf{K}^{-T}\mathbf{K}^{-1}\right)\mathbf{v}_2 = 0 \\ \mathbf{v}_3^T \left(\mathbf{K}^{-T}\mathbf{K}^{-1}\right)\mathbf{v}_4 = 0 \end{cases} \tag{11}$$

$\mathbf{K}^{-T}\mathbf{K}^{-1}$ is symmetric

If we have M calibration board images, we can finally have $2M$ such equations and then we can solve the elements in $\mathbf{K}$.

Initial Rough Estimation of Calibration Parameters

- OpenCV's implementation adopts a simplified strategy
  - It does not estimate c_x and c_y at this step; instead, they are simply taken as the width/2 and height/2 of the image

$$\mathbf{K}=\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}=\underbrace{\begin{bmatrix} 1 & 0 & c_x \\ 0 & 1 & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{P}}\underbrace{\begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{Q}} \Rightarrow \mathbf{K}=\mathbf{PQ} \Rightarrow \mathbf{K}^{-T}\mathbf{K}^{-1}=(\mathbf{PQ})^{-T}(\mathbf{PQ})^{-1}=\mathbf{P}^{-T}\left(\mathbf{Q}^{-T}\mathbf{Q}^{-1}\right)\mathbf{P}^{-1}$$

$$\begin{cases} \mathbf{v}_1^T\left(\mathbf{K}^{-T}\mathbf{K}^{-1}\right)\mathbf{v}_2=0 \\ \mathbf{v}_3^T\left(\mathbf{K}^{-T}\mathbf{K}^{-1}\right)\mathbf{v}_4=0 \end{cases} \quad (11) \quad \Rightarrow \begin{cases} \left(\mathbf{P}^{-1}\mathbf{v}_1\right)^T\left(\mathbf{Q}^{-T}\mathbf{Q}^{-1}\right)\mathbf{P}^{-1}\mathbf{v}_2=0 \\ \left(\mathbf{P}^{-1}\mathbf{v}_3\right)^T\left(\mathbf{Q}^{-T}\mathbf{Q}^{-1}\right)\mathbf{P}^{-1}\mathbf{v}_4=0 \end{cases} \quad (12)$$

$$\mathbf{Q}^{-T}\mathbf{Q}^{-1}=\begin{bmatrix} \frac{1}{f_x^2} & 0 & 0 \\ 0 & \frac{1}{f_y^2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{P}^{-1}\mathbf{v}_1 \triangleq \begin{pmatrix} a_1 \\ b_1 \\ c_1 \end{pmatrix}, \mathbf{P}^{-1}\mathbf{v}_2 \triangleq \begin{pmatrix} a_2 \\ b_2 \\ c_2 \end{pmatrix}, \mathbf{P}^{-1}\mathbf{v}_3 \triangleq \begin{pmatrix} a_3 \\ b_3 \\ c_3 \end{pmatrix}, \mathbf{P}^{-1}\mathbf{v}_4 \triangleq \begin{pmatrix} a_4 \\ b_4 \\ c_4 \end{pmatrix}$$

Initial Rough Estimation of Calibration Parameters

- OpenCV's implementation adopts a simplified strategy

(12) becomes
$$\begin{cases} \dfrac{a_1 a_2}{f_x^2} + \dfrac{b_1 b_2}{f_y^2} = -c_1 c_2 \\ \dfrac{a_3 a_4}{f_x^2} + \dfrac{b_3 b_4}{f_y^2} = -c_3 c_4 \end{cases} \Rightarrow \begin{bmatrix} a_1 a_2 & b_1 b_2 \\ a_3 a_4 & b_3 b_4 \end{bmatrix} \begin{bmatrix} \dfrac{1}{f_x^2} \\ \dfrac{1}{f_y^2} \end{bmatrix} = \begin{bmatrix} -c_1 c_2 \\ -c_3 c_4 \end{bmatrix}$$

If we have M calibration board images, we can finally have $2M$ such equations,

$$\mathbf{A}_{2M\times 2} \begin{bmatrix} \dfrac{1}{f_x^2} \\ \dfrac{1}{f_y^2} \end{bmatrix} = \mathbf{b}_{2M\times 1}$$

We can solve $\left[\dfrac{1}{f_x^2}, \dfrac{1}{f_y^2}\right]$ using the least squares technique, and at last f_x and f_y are obtained

Initial Rough Estimation of Calibration Parameters

- Initial estimation of extrinsics

We know that the plane of the calibration board and its image on the normalized retinal plane is linked via a homography

$$c_i \begin{bmatrix} x_{ni} \\ y_{ni} \\ 1 \end{bmatrix} = \mathbf{H}_{3\times 3} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}$$

On the other hand, based on the imaging model,

$$Z_{Ci} \begin{bmatrix} x_{ni} \\ y_{ni} \\ 1 \end{bmatrix} = \underbrace{[\mathbf{R}\ \mathbf{t}] \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}}_{} = [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3\ \mathbf{t}] \begin{bmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{bmatrix} = [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}] \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}$$

$\mathbf{H}$ and $[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}]$ map $(X_i, Y_i, 1)^T$ to the same point on the normalized retinal plane

↓

$\mathbf{H}$ and $[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}]$ actually represent the same homography

↓

$$[\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3] = \mathbf{H} = \lambda [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}]$$

↓

Initial Rough Estimation of Calibration Parameters

- Initial estimation of extrinsics

$$\lambda \mathbf{r}_1 = \mathbf{h}_1, \lambda \mathbf{r}_2 = \mathbf{h}_2, \lambda \mathbf{t} = \mathbf{h}_3$$

$$\Downarrow$$

$$\mathbf{r}_1 = \frac{1}{\lambda}\mathbf{h}_1, \mathbf{r}_2 = \frac{1}{\lambda}\mathbf{h}_2, \mathbf{t} = \frac{1}{\lambda}\mathbf{h}_3$$

Since $\|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1 \Rightarrow \lambda = \|\mathbf{h}_1\| = \|\mathbf{h}_2\|$

Note: In OpenCV, λ is estimated as $\lambda = \frac{1}{2}\left(\|\mathbf{h}_1\| + \|\mathbf{h}_2\|\right)$

Since $\mathbf{r}_3 \perp \mathbf{r}_1, \mathbf{r}_3 \perp \mathbf{r}_2, \|\mathbf{r}_3\| = 1, \det\left([\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3]\right) = 1 \Rightarrow \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$

Then, $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$, and $\mathbf{t}$ are all initialized

Finally, $\mathbf{R}=[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3]$ is converted to its axis-angle representation

Note: Initial estimation of extrinsics needs to be performed to every calibration board image

Outline

- What is Camera Calibration?
- Modeling for Imaging Pipeline
- General Framework for the Camera Calibration Algorithm
- Initial Rough Estimation of Calibration Parameters
- Nonlinear Least-squares
- Bird's-eye-view Generation

Nonlinear Least-squares

- For nonlinear least-square solutions, please refer to Lecture 5

The camera calibration problem is to solve,

$$\Theta^* = \arg\min_{\Theta} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{1}{2} \underbrace{\left\| \overbrace{\mathbf{K} \cdot \mathcal{D}\left\{ \frac{1}{Z_{Cij}} \left[\mathbf{R}_i \ \mathbf{t}_i \right] \mathbf{P}_j \right\}}^{\mathbf{p}_{ij}} - \mathbf{u}_{ij} \right\|_2^2}_{err\_term_{ij}} \tag{9}$$

In all modern implementations, Eq. 9 is solved by L-M method whose updating step is

$$\mathbf{h}_{lm} = -\left(\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}\right)^{-1} \mathbf{J}^T \mathbf{f}$$

our case

$$\mathbf{f} = \begin{bmatrix} err\_term_{11}(\Theta) \\ err\_term_{12}(\Theta) \\ \vdots \\ err\_term_{1N}(\Theta) \\ err\_term_{21}(\Theta) \\ \vdots \\ err\_term_{MN}(\Theta) \end{bmatrix}_{2MN\times 1}, \quad \mathbf{J} = \begin{bmatrix} derr\_term_{11} / d\Theta^T \\ derr\_term_{12} / d\Theta^T \\ \vdots \\ derr\_term_{1N} / d\Theta^T \\ derr\_term_{21} / d\Theta^T \\ \vdots \\ derr\_term_{MN} / d\Theta^T \end{bmatrix}_{2MN\times(9+6M)} = \begin{bmatrix} d\mathbf{p}_{11} / d\Theta^T \\ d\mathbf{p}_{12} / d\Theta^T \\ \vdots \\ d\mathbf{p}_{1N} / d\Theta^T \\ d\mathbf{p}_{21} / d\Theta^T \\ \vdots \\ d\mathbf{p}_{MN} / d\Theta^T \end{bmatrix}_{2MN\times(9+6M)}$$

Nonlinear Least-squares

The core problem is to determine $\dfrac{d\mathbf{p}_{ij}}{d\Theta^T}$,

i.e, to determine

$$\frac{d\mathbf{p}_{ij}}{df_x}, \frac{d\mathbf{p}_{ij}}{df_y}, \frac{d\mathbf{p}_{ij}}{dc_x}, \frac{d\mathbf{p}_{ij}}{dc_y}, \frac{d\mathbf{p}_{ij}}{dk_1}, \frac{d\mathbf{p}_{ij}}{dk_2}, \frac{d\mathbf{p}_{ij}}{d\rho_1}, \frac{d\mathbf{p}_{ij}}{d\rho_2}, \frac{d\mathbf{p}_{ij}}{dk_3}, \frac{d\mathbf{p}_{ij}}{d\mathbf{d}_i^T}, \frac{d\mathbf{p}_{ij}}{d\mathbf{t}_i^T}$$

Note that: $\dfrac{d\mathbf{p}_{ij}}{d\mathbf{d}_m^T} = \mathbf{0}, \dfrac{d\mathbf{p}_{ij}}{d\mathbf{t}_m^T} = \mathbf{0}, \forall m \neq i$

For derivation simplicity, in the following, we denote

$$\mathbf{p} = \begin{bmatrix} u \\ v \end{bmatrix} \triangleq \mathbf{p}_{ij}, \mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \triangleq \mathbf{d}_i$$

Denote $\mathbf{d}$'s rotation matrix representation by $\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$ and its vector form by $\mathbf{r} = \begin{bmatrix} R_{11} \\ R_{12} \\ R_{13} \\ R_{21} \\ R_{22} \\ R_{23} \\ R_{31} \\ R_{32} \\ R_{33} \end{bmatrix}$

Nonlinear Least-squares

Denote the 3D point corresponding to $\mathbf{p}_{ij}$ in the WCS (determined by the physical calibration board) by $\mathbf{P}=[X,Y,Z]^T$

Denote $\mathbf{P}$'s position w.r.t the camera coordinate system by $\mathbf{P}_C=[X_C,Y_C,Z_C]^T$

Denote $\mathbf{P}$'s ideal projection on the normalized retinal plane by $\mathbf{p}_n=[x_n, y_n]^T$

Denote $\mathbf{P}$'s distorted projection on the normalized retinal plane by $\mathbf{p}_d=[x_d, y_d]^T$

Let's derive the above-mentioned derivatives one by one……

Nonlinear Least-squares

According to Eq, 6 (from the projection on the normalized retinal plane to the final pixel position), we have

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \Rightarrow \begin{cases} u = f_x x_d + c_x \\ v = f_y y_d + c_y \end{cases}$$

$$\frac{d\mathbf{p}}{df_x} = \begin{bmatrix} \frac{\partial u}{\partial f_x} \\ \frac{\partial v}{\partial f_x} \end{bmatrix} = \begin{bmatrix} x_d \\ 0 \end{bmatrix}, \frac{d\mathbf{p}}{df_y} = \begin{bmatrix} \frac{\partial u}{\partial f_y} \\ \frac{\partial v}{\partial f_y} \end{bmatrix} = \begin{bmatrix} 0 \\ y_d \end{bmatrix}, \frac{d\mathbf{p}}{dc_x} = \begin{bmatrix} \frac{\partial u}{\partial c_x} \\ \frac{\partial v}{\partial c_x} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \frac{d\mathbf{p}}{dc_y} = \begin{bmatrix} \frac{\partial u}{\partial c_y} \\ \frac{\partial v}{\partial c_y} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

We also have a byproduct which will be used later,

$$\frac{d\mathbf{p}}{d\mathbf{p}_d^T} = \begin{bmatrix} \frac{\partial u}{\partial x_d} & \frac{\partial u}{\partial y_d} \\ \frac{\partial v}{\partial x_d} & \frac{\partial v}{\partial y_d} \end{bmatrix} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix}$$

Nonlinear Least-squares

According to Eq, 7 and the notation $\mathbf{k} \triangleq [k_1\ k_2\ \rho_1\ \rho_2\ k_3]^T$ we have

$$\frac{d\mathbf{p}_d}{d\mathbf{k}^T} = \begin{bmatrix} \frac{\partial x_d}{\partial k_1} & \frac{\partial x_d}{\partial k_2} & \frac{\partial x_d}{\partial \rho_1} & \frac{\partial x_d}{\partial \rho_2} & \frac{\partial x_d}{\partial k_3} \\ \frac{\partial y_d}{\partial k_1} & \frac{\partial y_d}{\partial k_2} & \frac{\partial y_d}{\partial \rho_1} & \frac{\partial y_d}{\partial \rho_2} & \frac{\partial y_d}{\partial k_3} \end{bmatrix} = \begin{bmatrix} x_n r^2 & x_n r^4 & 2x_n y_n & r^2 + 2x_n^2 & x_n r^6 \\ y_n r^2 & y_n r^4 & r^2 + 2y_n^2 & 2x_n y_n & y_n r^6 \end{bmatrix}$$

Then we have,

$$\boxed{\frac{d\mathbf{p}}{d\mathbf{k}^T} = \frac{d\mathbf{p}}{d\mathbf{p}_d^T} \cdot \frac{d\mathbf{p}_d}{d\mathbf{k}^T} = \begin{bmatrix} f_x x_n r^2 & f_x x_n r^4 & 2 f_x x_n y_n & f_x (r^2 + 2x_n^2) & f_x x_n r^6 \\ f_y y_n r^2 & f_y y_n r^4 & f_y (r^2 + 2y_n^2) & 2 f_y x_n y_n & f_y y_n r^6 \end{bmatrix}}$$

Also based on Eq. 7, we can have

$$\frac{d\mathbf{p}_d}{d\mathbf{p}_n^T} = \begin{bmatrix} \frac{\partial x_d}{\partial x_n} & \frac{\partial x_d}{\partial y_n} \\ \frac{\partial y_d}{\partial x_n} & \frac{\partial y_d}{\partial y_n} \end{bmatrix} = [...]$$

Its concrete form is a little complicated, but not difficult

Nonlinear Least-squares

According to Eq, 3, we have

$$\frac{d\mathbf{p}_n}{d\mathbf{P}_C^T} = \begin{bmatrix} \frac{\partial x_n}{\partial X_C} & \frac{\partial x_n}{\partial Y_C} & \frac{\partial x_n}{\partial Z_C} \\ \frac{\partial y_n}{\partial X_C} & \frac{\partial y_n}{\partial Y_C} & \frac{\partial y_n}{\partial Z_C} \end{bmatrix} = \begin{bmatrix} \frac{1}{Z_C} & 0 & \frac{-X_C}{Z_C^2} \\ 0 & \frac{1}{Z_C} & \frac{-Y_C}{Z_C^2} \end{bmatrix}$$

According to Eq, 1, we have

$$\mathbf{P}_C = \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \begin{bmatrix} R_{11}X + R_{12}Y + R_{13}Z + t_1 \\ R_{21}X + R_{22}Y + R_{23}Z + t_2 \\ R_{31}X + R_{32}Y + R_{33}Z + t_3 \end{bmatrix} \Rightarrow$$

$$\frac{d\mathbf{P}_C}{d\mathbf{r}^T} = \begin{bmatrix} \frac{\partial X_C}{\partial R_{11}} & \frac{\partial X_C}{\partial R_{12}} & \frac{\partial X_C}{\partial R_{13}} & \frac{\partial X_C}{\partial R_{21}} & \frac{\partial X_C}{\partial R_{22}} & \frac{\partial X_C}{\partial R_{23}} & \frac{\partial X_C}{\partial R_{31}} & \frac{\partial X_C}{\partial R_{32}} & \frac{\partial X_C}{\partial R_{33}} \\ \frac{\partial Y_C}{\partial R_{11}} & \frac{\partial Y_C}{\partial R_{12}} & \frac{\partial Y_C}{\partial R_{13}} & \frac{\partial Y_C}{\partial R_{21}} & \frac{\partial Y_C}{\partial R_{22}} & \frac{\partial Y_C}{\partial R_{23}} & \frac{\partial Y_C}{\partial R_{31}} & \frac{\partial Y_C}{\partial R_{32}} & \frac{\partial Y_C}{\partial R_{33}} \\ \frac{\partial Z_C}{\partial R_{11}} & \frac{\partial Z_C}{\partial R_{12}} & \frac{\partial Z_C}{\partial R_{13}} & \frac{\partial Z_C}{\partial R_{21}} & \frac{\partial Z_C}{\partial R_{22}} & \frac{\partial Z_C}{\partial R_{23}} & \frac{\partial Z_C}{\partial R_{31}} & \frac{\partial Z_C}{\partial R_{32}} & \frac{\partial Z_C}{\partial R_{33}} \end{bmatrix} = \begin{bmatrix} X & Y & Z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X & Y & Z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & X & Y & Z \end{bmatrix}$$

$$\frac{d\mathbf{P}_C}{d\mathbf{t}^T} = \begin{bmatrix} \frac{\partial X_C}{\partial t_1} & \frac{\partial X_C}{\partial t_2} & \frac{\partial X_C}{\partial t_3} \\ \frac{\partial Y_C}{\partial t_1} & \frac{\partial Y_C}{\partial t_2} & \frac{\partial Y_C}{\partial t_3} \\ \frac{\partial Z_C}{\partial t_1} & \frac{\partial Z_C}{\partial t_2} & \frac{\partial Z_C}{\partial t_3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Nonlinear Least-squares

According to Rodrigues formula (Eq, 10), we can derive the form of

$$\frac{d\mathbf{r}}{d\mathbf{d}^T} \in \mathbb{R}^{9\times 3}$$

Assignment!

Then, we can compute,

$$\boxed{\frac{d\mathbf{p}}{d\mathbf{d}^T} = \frac{d\mathbf{p}}{d\mathbf{p}_d^T} \cdot \frac{d\mathbf{p}_d}{d\mathbf{p}_n^T} \cdot \frac{d\mathbf{p}_n}{d\mathbf{P}_C^T} \cdot \frac{d\mathbf{P}_C}{d\mathbf{r}^T} \cdot \frac{d\mathbf{r}}{d\mathbf{d}^T}}$$

$$\boxed{\frac{d\mathbf{p}}{d\mathbf{t}^T} = \frac{d\mathbf{p}}{d\mathbf{p}_d^T} \cdot \frac{d\mathbf{p}_d}{d\mathbf{p}_n^T} \cdot \frac{d\mathbf{p}_n}{d\mathbf{P}_C^T} \cdot \frac{d\mathbf{P}_C}{d\mathbf{t}^T}}$$

Nonlinear Least-squares

With the calibrated camera, many amazing applications can be continuously performed….

One naive example, the distorted image can be undistorted

One point on the undistorted image

The corresponding point on the original image with distortion

$$\begin{pmatrix} u \\ v \end{pmatrix} \Longleftrightarrow \mathbf{K}\mathcal{D}\left(\mathbf{K}^{-1}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}\right)$$

Outline

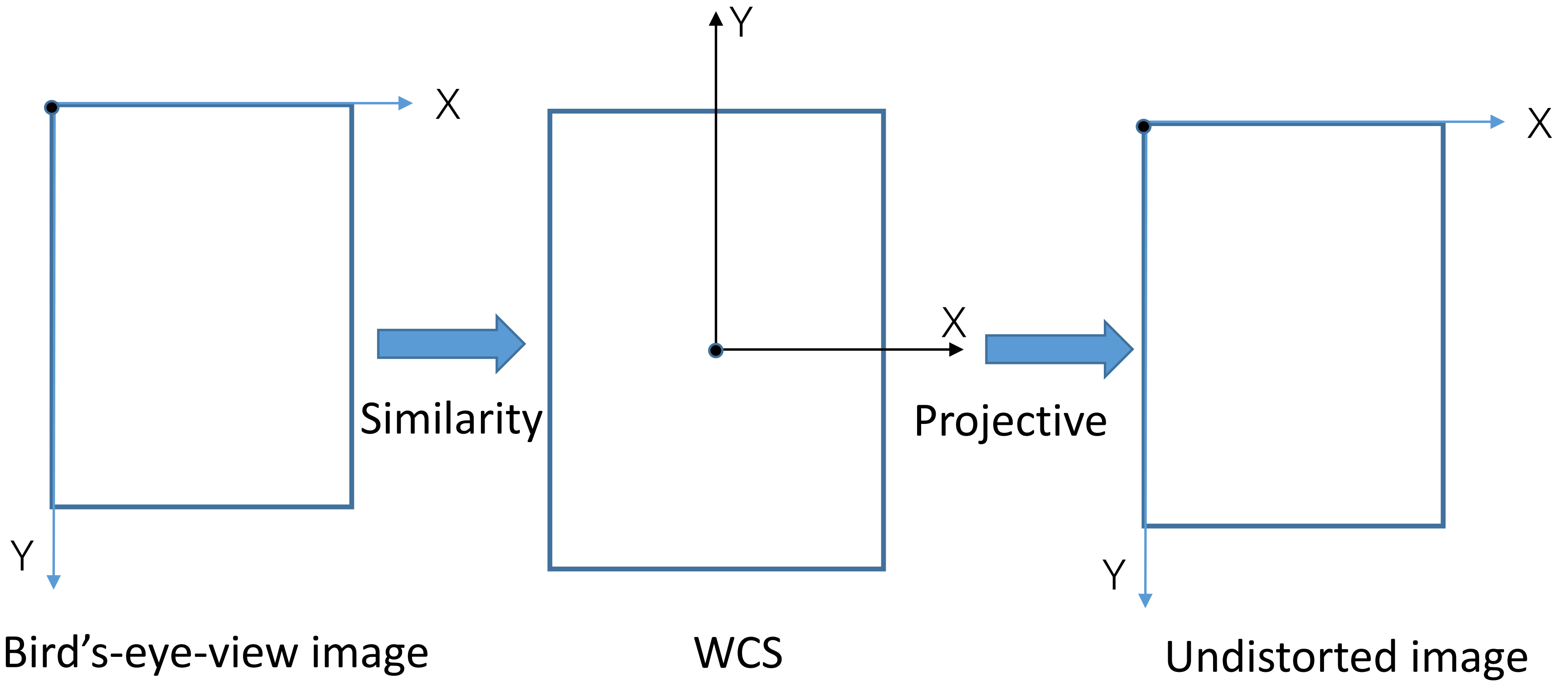
- What is Camera Calibration?
- Modeling for Imaging Pipeline
- General Framework for the Camera Calibration Algorithm
- Initial Rough Estimation of Calibration Parameters
- Nonlinear Least-squares
- Bird's-eye-view Generation

Bird’s-eye-view Generation

- Our task is to measure the geometric properties of objects on a plane (e.g., conveyor belt)
- Such a problem can be solved if we have its bird’s-eye-view image; bird’s-eye-view is easy for object detection and measurement

Bird’s-eye-view Generation

- Three coordinate systems are required
  - Bird’s-eye-view image coordinate system
  - World coordinate system
  - Undistorted image coordinate system



Bird's-eye-view Generation

- Basic idea for bird's-eye-view generation

Suppose that the transformation matrix from bird's-eye-view to WCS is $P_{B\to W}$ and the transformation matrix from WCS to the undistorted image is $P_{W\to I}$

Then, given a position $\left(x_B, y_B, 1\right)^T$ on bird's-eye-view, we can get its corresponding position in the undistorted image as

$$\mathbf{x}_I = P_{W\to I} P_{B\to W} \begin{pmatrix} x_B \\ y_B \\ 1 \end{pmatrix}$$

Then, the intensity of the pixel $\left(x_B, y_B, 1\right)^T$ can be determined using some interpolation technique based on the neighborhood around $\mathbf{x}_I$ on the undistorted image

Bird's-eye-view Generation

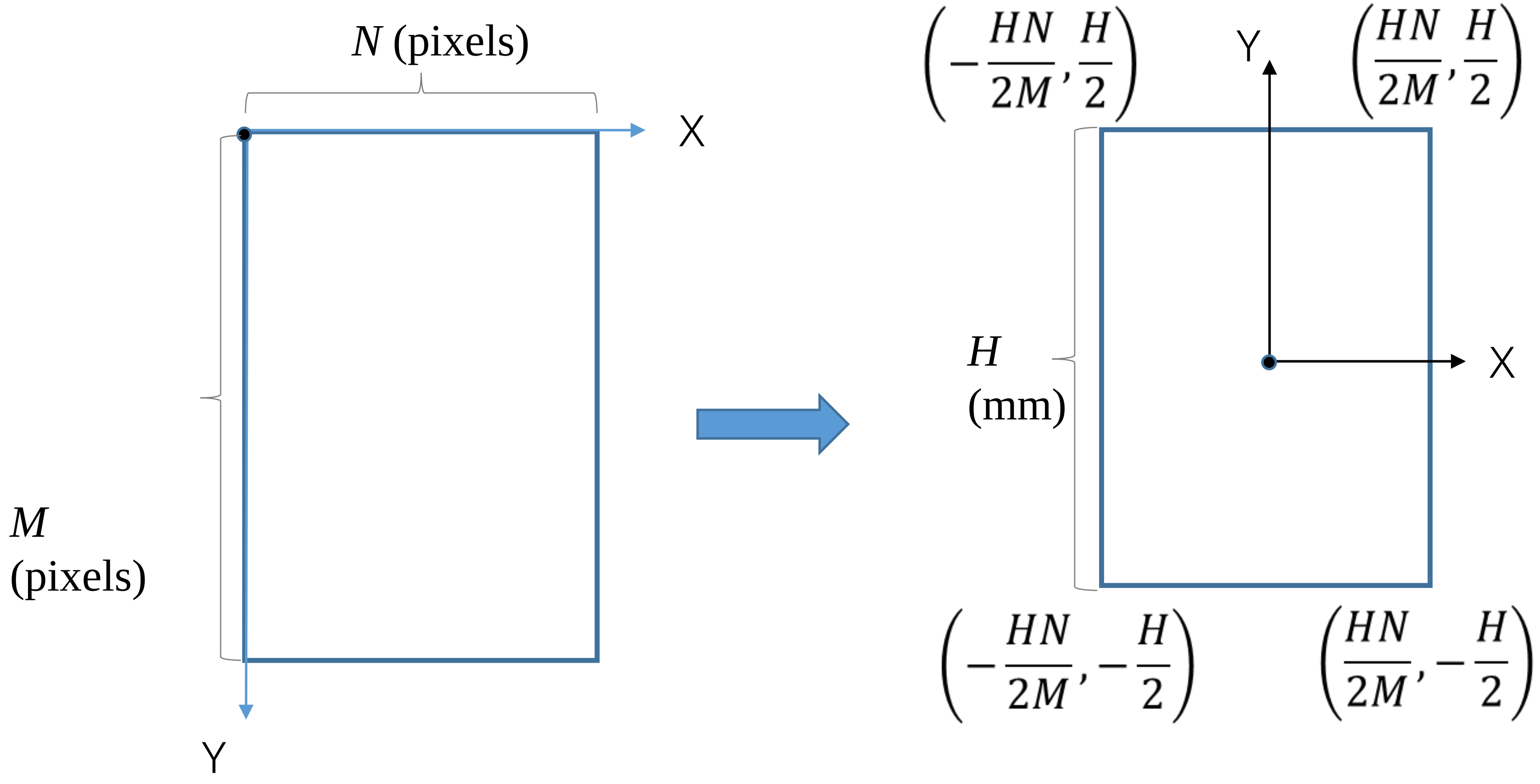
- Basic idea for bird's-eye-view generation

Suppose that the transformation matrix from bird's-eye-view to WCS is $P_{B\to W}$ and the transformation matrix from WCS to the undistorted image is $P_{W\to I}$

The key problem is how to obtain $P_{B\to W}$ and $P_{W\to I}$?

Bird's-eye-view Generation

- Determine $P_{B\to W}$



Note: It is valid only when you think the origin of the world CS is at the center of the bird's-eye-view image

Bird's-eye-view Generation

- Determine $P_{B\to W}$

For a point $(x_B, y_B, 1)^T$ on bird's-eye-view, the corresponding point on the world coordinate system is,

$$\begin{pmatrix} x_W \\ y_W \\ 1 \end{pmatrix} = \begin{bmatrix} \frac{H}{M} & 0 & -\frac{HN}{2M} \\ 0 & -\frac{H}{M} & \frac{H}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_B \\ y_B \\ 1 \end{pmatrix} \equiv P_{B\to W} \begin{pmatrix} x_B \\ y_B \\ 1 \end{pmatrix}$$

Please verify!!

Bird's-eye-view Generation

- Determine $P_{W\to I}$

The physical plane (in WCS) and the undistorted image plane can be linked via a homography matrix $P_{W\to I}$

$$\mathbf{x}_I = P_{W\to I}\mathbf{x}_W$$

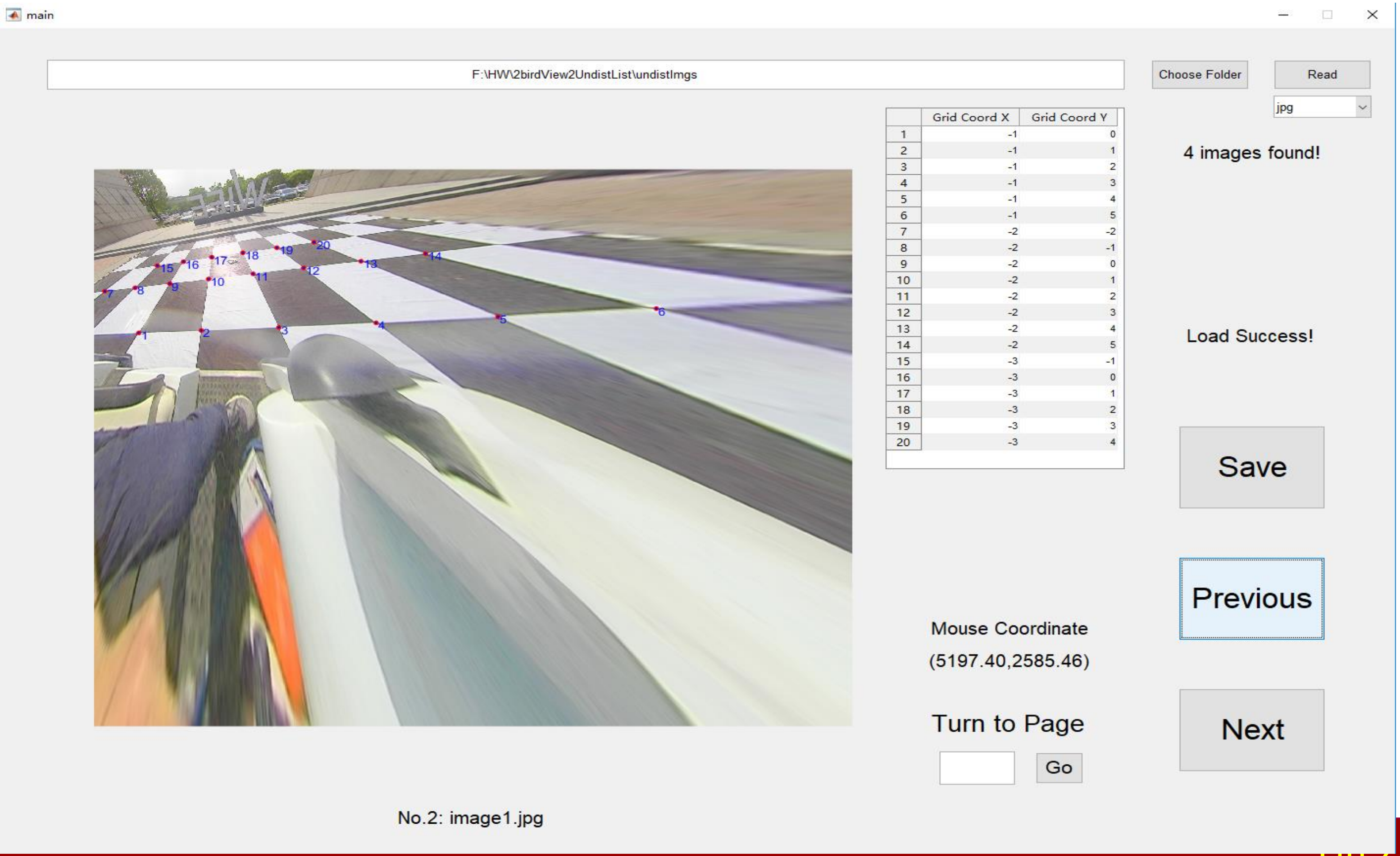
If we know a set of correspondence pairs $\left\{\mathbf{x}_{Ii}, \mathbf{x}_{Wi}\right\}_{i=1}^{N}$,

$P_{W\to I}$ can be estimated using the least-square method

Bird's-eye-view Generation

- Determine $P_{W \to I}$

A set of point correspondence pairs; for each pair, we know its coordinate on the undistorted image plane and its coordinate in the WCS



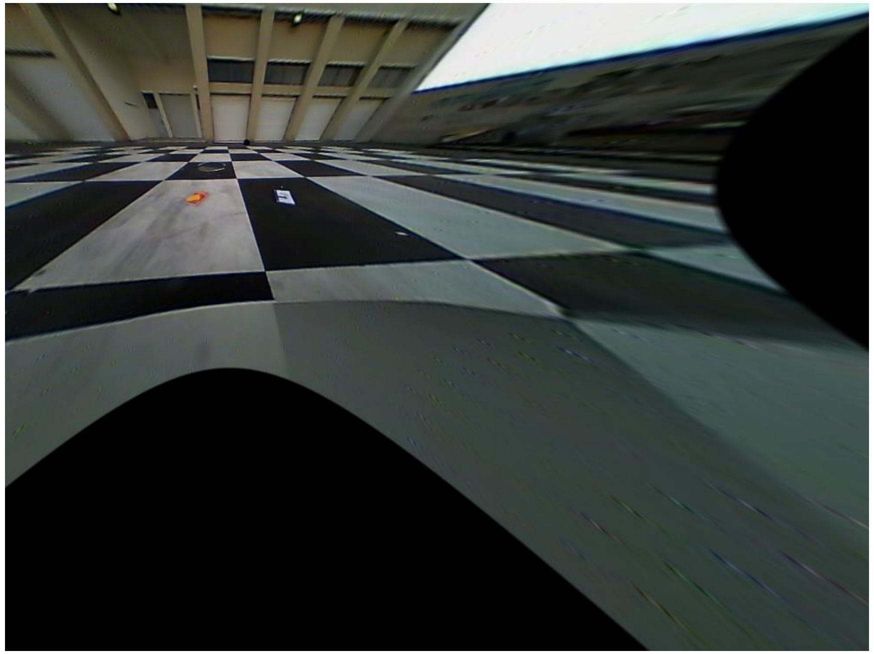
Bird's-eye-view Generation

When $P_{B\to W}$ and $P_{W\to I}$ are known, the bird's-eye-view can be generated via,

$$\mathbf{x}_I = P_{W\to I} P_{B\to W} \begin{pmatrix} x_B \\ y_B \\ 1 \end{pmatrix} \equiv P_{B\to I} \begin{pmatrix} x_B \\ y_B \\ 1 \end{pmatrix}$$

Bird-view Generation

Another example



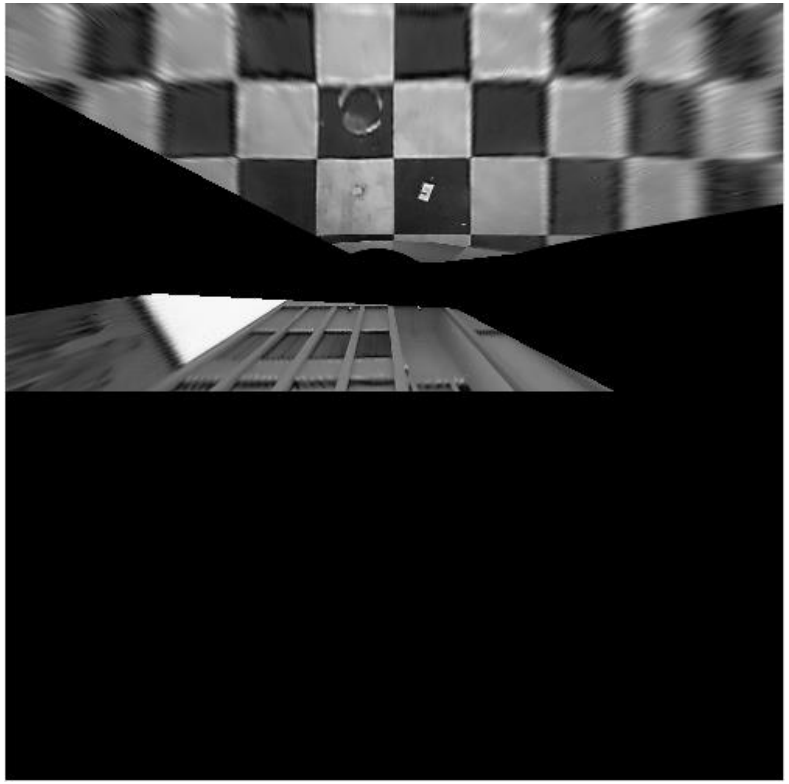
Original fish-eye image

Undistorted image

Bird-view Generation

Another example

Original fish-eye image



Bird's-eye-view


Q+A