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 2021
If I have an image containing a coin, can you tell me the diameter of that coin?
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?

Before

Remove Lens Distortion

Estimate 3-D Structure from Camera Motion

Estimate Depth Using a Stereo Camera

Measure Planar Objects

After
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
  • Extrinsic

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

\[
\begin{bmatrix}
X_w, Y_w, Z_w
\end{bmatrix}^T \text{ is a 3D point represented in the WCS}
\]

In the camera CS, it is represented as,

\[
\begin{bmatrix}
X_C \\
Y_C \\
Z_C
\end{bmatrix} = \mathbf{R} \begin{bmatrix}
X_w \\
Y_w \\
Z_w
\end{bmatrix} + \mathbf{t}
\]

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

\[
\begin{bmatrix}
V \\
W \\
U
\end{bmatrix} = \begin{bmatrix}
\mathbf{R} & \mathbf{t}
\end{bmatrix}_{3 \times 4}
\begin{bmatrix}
X_w \\
Y_w \\
Z_w \\
1
\end{bmatrix} \quad (2)
\]

where \( \mathbf{R} \) is a 3×3 rotation matrix (orthogonal, det(\( \mathbf{R} \))=1), \( \mathbf{t} \) is a 3×1 translation vector (inhomogeneous), \( \mathbf{X}_w \), \( \mathbf{Y}_w \), \( \mathbf{Z}_w \) are 3×1 vectors.
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

\[
\begin{bmatrix}
X_c \\
Y_c \\
Z_c
\end{bmatrix}
\rightarrow
\begin{bmatrix}
x \\
y
\end{bmatrix}
= \begin{bmatrix}
f & X_c & Z_c \\
0 & f & Y_c \\
f & 0 & Z_c
\end{bmatrix}
\]

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

\[
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix} = \begin{bmatrix}
  f \frac{X_c}{Z_c} \\
  f \frac{Y_c}{Z_c}
\end{bmatrix}
\]

Homogeneous form

\[
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix}
  fX_c \\
  fY_c \\
  Z_c
\end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix}
  f & 0 & 0 \\
  0 & f & 0 \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  X_c \\
  Y_c \\
  Z_c
\end{bmatrix}
\]

(normalized) Homogeneous

(2)

Normalized

Inhomogeneous
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

\[
\begin{bmatrix}
X_c \\
Y_c \\
Z_c
\end{bmatrix} \rightarrow \begin{bmatrix}
x_n \\
y_n \\
1
\end{bmatrix} = \begin{bmatrix}
\frac{X_c}{Z_c} \\
\frac{Y_c}{Z_c} \\
\frac{1}{Z_c}
\end{bmatrix}
\]

Homogeneous form

\[
\begin{bmatrix}
x_n \\
y_n \\
1
\end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix}
X_c \\
Y_c \\
Z_c
\end{bmatrix}
\]

(3)
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.

One pixel represents \(dx\) physical units along the x-axis and represents \(dy\) physical units along the y-axis; the image of the optical center is \((c_x, c_y)\).

\[
\begin{bmatrix}
1 & \frac{1}{dx} & c_x \\
0 & \frac{1}{dy} & c_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]
Modeling for Imaging Pipeline

• From the retinal CS to the pixel CS

If the two axis of the image plane are not perpendicular,

\[
\begin{bmatrix}
1 & \tan \alpha & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
= 
\begin{bmatrix}
x + y \tan \alpha \\
y \\
1
\end{bmatrix}
\]

\[
\begin{bmatrix}
\frac{1}{dx} & 0 & c_x \\
0 & \frac{1}{dy} & c_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & \tan \alpha & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
= 
\begin{bmatrix}
\frac{1}{dx} & \frac{\tan \alpha}{dx} & c_x \\
0 & \frac{1}{dy} & c_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

(4)
Modeling for Imaging Pipeline

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

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

\[
\begin{bmatrix}
f_x & s & c_x \\
0 & f_y & c_y \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
X_w \\
Y_w \\
Z_w
\end{bmatrix} = \frac{1}{Z_C} \begin{bmatrix}
R & t
\end{bmatrix}_{3\times4} \begin{bmatrix}
X_w \\
Y_w \\
Z_w
\end{bmatrix}
\]

Image formation model without considering lens distortions,

\[
u = \frac{1}{Z_C} \cdot K_{3\times3} \begin{bmatrix}
R & t
\end{bmatrix}_{3\times4} \cdot P_{4\times1}
\]

Note: \(u\) 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

Note: In this course, we do not consider $s$ anymore
Modeling for Imaging Pipeline

\[
\begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix} = \begin{bmatrix}
  f_x & 0 & c_x \\
  0 & f_y & c_y \\
  0 & 0 & 1
\end{bmatrix} \cdot \frac{1}{Z_C} \begin{bmatrix}
  X_w \\
  Y_w \\
  Z_w \\
  1
\end{bmatrix}
\]

Point on the normalized retinal CS

\[
\frac{1}{Z_C} \begin{bmatrix}
  R & t
\end{bmatrix}_{3x4} \begin{bmatrix}
  X_w \\
  Y_w \\
  Z_w \\
  1
\end{bmatrix} = \begin{bmatrix}
  x_n \\
  y_n \\
  1
\end{bmatrix} \quad \text{(according to Eq.3)}
\]

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

\[
\begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix} = \begin{bmatrix}
  f_x & 0 & c_x \\
  0 & f_y & c_y \\
  0 & 0 & 1
\end{bmatrix} \cdot \begin{bmatrix}
  x_n \\
  y_n \\
  1
\end{bmatrix} \quad \text{(6)}
\]
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
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
To model the behavior of lens, we need to consider the distortion

- Both the two types of distortions are modeled on the \textbf{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

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

where \( r^2 = x_n^2 + y_n^2 \)

\( \rho_1, \rho_2 \) are the tangential distortion coefficients

If they both need to be considered,

\[
\begin{align*}
    x_d &= x_n \left( 1 + k_1 r^2 + k_2 r^4 \right) + 2 \rho_1 x_n y_n + \rho_2 \left( r^2 + 2 x_n^2 \right) + x_n k_3 r^6 \\
    y_d &= y_n \left( 1 + k_1 r^2 + k_2 r^4 \right) + 2 \rho_2 x_n y_n + \rho_1 \left( r^2 + 2 y_n^2 \right) + y_n k_3 r^6
\end{align*}
\]  

(7)

\textbf{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.e. 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,

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

(8)

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

\( R \) (three DOFs) and \( t \) (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

\[
\begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix}
= \begin{bmatrix}
  f_x & 0 & c_x \\
  0 & f_y & c_y \\
  0 & 0 & 1
\end{bmatrix}
\cdot \mathcal{D}
\begin{bmatrix}
  1 \\
  Z_C
\end{bmatrix}
\begin{bmatrix}
  R \\
  t
\end{bmatrix}
\begin{bmatrix}
  X_w \\
  Y_w \\
  Z_w \\
  1
\end{bmatrix}
\]  
(Eq. 8, the imaging pipeline)

- **General idea**
  - If we have a set of known points \( \{P_i\}_{i=1}^n \) in the WCS and their images \( \{u_i\}_{i=1}^n \), using Eq. 8, we could have \( 2n \) equations
  - If the number of valid constraints (equations) are greater than unknowns, 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**

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
General Framework for the Camera Calibration Algorithm

- Zhengyou Zhang’s calibration approach

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

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

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

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

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

\[
\mathbf{d} = \mathbf{n}\theta
\]

\(\mathbf{n}\) is a unit 3D vector describing an axis of rotation according to the right hand rule; \(\theta\) 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 \( \mathbf{R} \)

\[
\mathbf{R} = \cos \theta \mathbf{I} + (1 - \cos \theta) \mathbf{n}\mathbf{n}^T + \sin \theta \mathbf{n}^\wedge
\]  

(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,

$$
\Theta^* = \arg \min_{\Theta} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{1}{2} \left\| K \cdot D \left\{ \frac{1}{Z_{Cij}} \left[ R_i \ t_i \right] P_j \right\} - u_{ij} \right\|_2^2
$$

where the parameters that need to be optimized are,

$$\Theta = \left\{ f_x, f_y, c_x, c_y, k_1, k_2, \rho_1, \rho_2, \rho_3, \{ d_i \}_{i=1}^{M}, \{ t_i \}_{i=1}^{M} \right\} \ (d_i \text{ is the axis-angle representation of } R_i)$$

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.e. we need to have a rough estimate to $\Theta$. 
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

$$\Theta = \left\{ f_x, f_y, c_x, c_y, k_1, k_2, \rho_1, \rho_2, k_3, \{d_i\}_{i=1}^M, \{t_i\}_{i=1}^M \right\}$$

Distortion coefficients can be safely initialized as zeros.

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

$$u = \frac{1}{Z_c} \cdot K_{3\times3} \cdot [R \ t]_{3\times4} \cdot P_{4\times1} \quad \text{(Eq. 5)}$$

Given a calibration board, $P$ is a cross-point on it, thus $P$ has the form $P = \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$
**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 $u$ (in homogeneous form) on the imaging plane is

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

The imaging model is

$$u = \frac{1}{Z_c}K_{3 \times 3} \begin{bmatrix} R & t \end{bmatrix}_{3 \times 4} P_{4 \times 1}$$

(Eq. 5, $u$ is normalized homogeneous)

Since $x_n$ should on the ray $\overline{Ou}$, $\overline{Ou}$'s direction is

$$d = \overline{x_n - 0} = K^{-1}u$$

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

$u$ actually does not need to be normalized homogeneous
Result 2: In the camera coordinate system, the angle between two rays, pointing from $O$ to $x_1$ and $x_2$ ($x_1$ and $x_2$ are the homogeneous coordinates of two pixels on the imaging plane), respectively, is determined as,

$$\cos \theta = \frac{d_1 \cdot d_2}{\|d_1\| \|d_2\|}$$

$$= \frac{\left(K^{-1}x_1\right)^T K^{-1}x_2}{\sqrt{\left(K^{-1}x_1\right)^T \left(K^{-1}x_1\right)} \sqrt{\left(K^{-1}x_2\right)^T \left(K^{-1}x_2\right)}}$$

$$= \frac{x_1^T \left(K^{-T}K^{-1}\right)x_2}{\sqrt{x_1^T \left(K^{-T}K^{-1}\right)x_1} \sqrt{x_2^T \left(K^{-T}K^{-1}\right)x_2}}$$
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 interesting 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
The points $X_i$, $i = 1, \ldots, 4$ are equally spaced on the world line, but their spacing on the image line monotonically decreases. In the limit $X \to \infty$, the world point is imaged at $x = v$ on the vertical image line, and at $x' = v$ 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 $C$. 
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 $Ov$ ($O$ is the optical center) is parallel to the world lines who share the same vanishing point $v$

Result 3: $l_1$ and $l_2$ are two world lines and $v_1$ and $v_2$ are their vanishing points on the imaging plane, respectively. $O$ is the optical center. Then, we have

$$\theta = \frac{l_1 l_2}{Ov_1 Ov_2}$$

and

$$\cos \theta = \frac{v_1^T (K^{-T} K^{-1}) v_2}{\sqrt{v_1^T (K^{-T} K^{-1}) v_1} \sqrt{v_2^T (K^{-T} K^{-1}) v_2}}$$

(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

\[
\mathbf{v}_1^T \left( \mathbf{K}^{-T} \mathbf{K}^{-1} \right) \mathbf{v}_2 = 0
\]

(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,

\[ l_1: \text{X-axis, the infinity point is } P_1 = (1,0,0) \]
\[ l_2: \text{Y-axis, the infinity point is } P_2 = (0,1,0) \]
\[ l_3: \text{lines with the direction (the infinity point) } P_3 = P_1 + P_2 = (1,1,0) \]
\[ l_4: \text{lines with the direction (the infinity point) } P_4 = P_1 - P_2 = (1,-1,0) \]

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

\[ c \begin{bmatrix} u \\ v \end{bmatrix} = H \begin{bmatrix} x \\ y \end{bmatrix} \]

point on the image  
homogeneous planar point on the calibration board

\( H \) can be estimated in advance for each calibration board image using the techniques introduced in Lect. 3

Denote \( H \) by,

\[ H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} \]

Images of \( P_1, P_2, P_3, P_4 \) are,

\[ \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = h_1, \quad \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = h_2 \]

\[ \begin{bmatrix} v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = h_1 + h_2 \]

\[ \begin{bmatrix} v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} = h_1 - h_2 \]
Initial Rough Estimation of Calibration Parameters

Based on Result 4, we have

\[
\begin{align*}
\mathbf{v}_1^T (\mathbf{K}^{-T} \mathbf{K}^{-1}) \mathbf{v}_2 &= 0 \\
\mathbf{v}_3^T (\mathbf{K}^{-T} \mathbf{K}^{-1}) \mathbf{v}_4 &= 0
\end{align*}
\]

(11)

\[\mathbf{K}^{-T} \mathbf{K}^{-1}\text{ is symmetric}\]

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

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

\begin{align*}
\begin{cases}
    v_1^T (K^{-T} K^{-1}) v_2 = 0 \\
    v_3^T (K^{-T} K^{-1}) v_4 = 0
\end{cases} & \Rightarrow
\begin{cases}
    \left( P^{-1} v_1 \right)^T (Q^{-T} Q^{-1}) P^{-1} v_2 = 0 \\
    \left( P^{-1} v_3 \right)^T (Q^{-T} Q^{-1}) P^{-1} v_4 = 0
\end{cases}
\end{align*}

$$
\begin{align*}
P^{-1} v_1 & \triangleq \begin{pmatrix} a_1 \\ b_1 \\ c_1 \end{pmatrix}, \quad P^{-1} v_2 \triangleq \begin{pmatrix} a_2 \\ b_2 \\ c_2 \end{pmatrix}, \quad P^{-1} v_3 \triangleq \begin{pmatrix} a_3 \\ b_3 \\ c_3 \end{pmatrix}, \quad P^{-1} v_4 \triangleq \begin{pmatrix} a_4 \\ b_4 \\ c_4 \end{pmatrix}
\end{align*}

Q^{-T} Q^{-1} = \begin{bmatrix}
  1 & 0 & 0 \\
  0 & \frac{1}{f_x^2} & 0 \\
  0 & 0 & \frac{1}{f_y^2}
\end{bmatrix}
\begin{bmatrix}
  0 & 0 & 1
\end{bmatrix}
Initial Rough Estimation of Calibration Parameters

• OpenCV’s implementation adopts a simplified strategy

\[
\begin{align*}
\begin{bmatrix}
\frac{a_1a_2}{f_x^2} + \frac{b_1b_2}{f_y^2} \\
\frac{a_3a_4}{f_x^2} + \frac{b_3b_4}{f_y^2}
\end{bmatrix} = -c_1c_2 \\
\begin{bmatrix}
\frac{1}{f_x^2} \\
\frac{1}{f_y^2}
\end{bmatrix} = \begin{bmatrix}
-c_1c_2 \\
-c_3c_4
\end{bmatrix}
\end{align*}
\]

(12) becomes

If we have \( M \) calibration board images, we can finally have \( 2M \) such equations,

\[
A_{2M \times 2} \begin{bmatrix}
\frac{1}{f_x^2} \\
\frac{1}{f_y^2}
\end{bmatrix} = b_{2M \times 1}
\]

We can solve \( \begin{bmatrix}
\frac{1}{f_x^2}, \frac{1}{f_y^2}
\end{bmatrix} \) 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

\[ c_i \begin{bmatrix} x_{ni} \\ y_{ni} \\ 1 \end{bmatrix} = H_{3 \times 3} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \]

On the other hand, based on the imaging model,

\[ Z_{Ci} \begin{bmatrix} x_{ni} \\ y_{ni} \\ 1 \end{bmatrix} = [R \ t] \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = [r_1 \ r_2 \ r_3 \ t] \begin{bmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{bmatrix} \]

\[ H \text{ and } [r_1 \ r_2 \ t] \text{ map } (X_i, Y_i, 1)^T \text{ to the same point on the normalized retinal plane} \]

\[ H \text{ and } [r_1 \ r_2 \ t] \text{ actually represent the same homography} \]

\[ [h_1, h_2, h_3] = H = \lambda [r_1 \ r_2 \ t] \]
Initial Rough Estimation of Calibration Parameters

• Initial estimation of extrinsics

\[
\hat{\lambda}r_1 = h_1, \hat{\lambda}r_2 = h_2, \hat{\lambda}t = h_3
\]

\[
\downarrow
\]

\[
r_1 = \frac{1}{\hat{\lambda}}h_1, r_2 = \frac{1}{\hat{\lambda}}h_2, t = \frac{1}{\hat{\lambda}}h_3
\]

Since \( \|r_1\| = \|r_2\| = 1 \Rightarrow \lambda = \|h_1\| = \|h_2\| \)

Note: In OpenCV, \( \hat{\lambda} \) is estimated as \( \lambda = \frac{1}{2}(\|h_1\| + \|h_2\|) \)

Since \( r_3 \perp r_1, r_3 \perp r_2, \|r_3\| = 1, \det([r_1 \ r_2 \ r_3]) = 1 \Rightarrow r_3 = r_1 \times r_2 \)

Then, \( r_1, r_2, r_3, \) and \( t \) are all initialized

Finally, \( R=[r_1 \ r_2 \ r_3] \) 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,

$$\Theta^* = \arg \min_{\Theta} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{1}{2} \left\| \text{err}_{\text{term}}_{ij} \right\|^2$$

where

$$\text{err}_{\text{term}}_{ij} = p_{ij} - \left( \frac{K \cdot D}{Z_{ij}} \begin{bmatrix} R_i & t_i \end{bmatrix} P_j \right)$$

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

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

for our case

$$f = \begin{bmatrix} err_{\text{term}}_{11} (\Theta) \\ err_{\text{term}}_{12} (\Theta) \\ \vdots \\ err_{\text{term}}_{1N} (\Theta) \\ err_{\text{term}}_{21} (\Theta) \\ \vdots \\ err_{\text{term}}_{MN} (\Theta) \end{bmatrix}_{2MN \times 1}, \quad J = \begin{bmatrix} derr_{\text{term}}_{11} / d\Theta^T \\ derr_{\text{term}}_{12} / d\Theta^T \\ \vdots \\ derr_{\text{term}}_{1N} / d\Theta^T \\ derr_{\text{term}}_{21} / d\Theta^T \\ \vdots \\ derr_{\text{term}}_{MN} / d\Theta^T \end{bmatrix}_{2MN \times (9+6M)}$$

$$= \begin{bmatrix} dp_{11} / d\Theta^T \\ dp_{12} / d\Theta^T \\ \vdots \\ dp_{1N} / d\Theta^T \\ dp_{21} / d\Theta^T \\ \vdots \\ dp_{MN} / d\Theta^T \end{bmatrix}_{2MN \times (9+6M)}$$
Nonlinear Least-squares

The core problem is to determine \( \frac{\partial p_{ij}}{\partial \Theta^T} \),

\[ \frac{dp_{ij}}{df_x}, \frac{dp_{ij}}{df_y}, \frac{dp_{ij}}{dc_x}, \frac{dp_{ij}}{dc_y}, \frac{dp_{ij}}{dk_1}, \frac{dp_{ij}}{dk_2}, \frac{dp_{ij}}{d\rho_1}, \frac{dp_{ij}}{d\rho_2}, \frac{dp_{ij}}{dk_3}, \frac{dp_{ij}}{d\mathbf{d}_i^T}, \frac{dp_{ij}}{dt_i^T} \]

\[ \text{Note that: } \frac{dp_{ij}}{dd_m^T} = 0, \frac{dp_{ij}}{dt_m^T} = 0, \forall m \neq i \]

For derivation simplicity, in the following, we denote

\[ p = \begin{bmatrix} u \\ v \end{bmatrix} \triangleq p_{ij}, \quad d = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \triangleq d_i \]

Denote \( d \)'s rotation matrix representation by

\[ R_1 R_2 R_3 \]

and its vector form by

\[ R_{11} R_{12} R_{13} R_{21} R_{22} R_{23} R_{31} R_{32} R_{33} \]

\[ R_{11} R_{12} R_{13} R_{21} R_{22} R_{23} R_{31} R_{32} R_{33} \]
Nonlinear Least-squares

Denote the 3D point corresponding to \( p_{ij} \) 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

\[
\begin{bmatrix}
  u \\
v \\
1
\end{bmatrix} =
\begin{bmatrix}
  f_x & 0 & c_x \\
  0 & f_y & c_y \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_d \\
y_d \\
1
\end{bmatrix}
\Rightarrow
\begin{align*}
u &= f_x x_d + c_x \\
v &= f_y y_d + c_y
\end{align*}
\]

We also have a byproduct which will be used later,

\[
\frac{dp}{df_x} = \begin{bmatrix}
\frac{\partial u}{\partial f_x} \\
\frac{\partial u}{\partial f_y} \\
\frac{\partial f_x}{\partial f_x} \\
\frac{\partial f_y}{\partial f_x}
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
1 \\
0
\end{bmatrix},
\frac{dp}{dc_x} = \begin{bmatrix}
\frac{\partial u}{\partial c_x} \\
\frac{\partial u}{\partial c_y} \\
\frac{\partial c_x}{\partial c_x} \\
\frac{\partial c_y}{\partial c_x}
\end{bmatrix} = \begin{bmatrix}
1 \\
0 \\
0 \\
1
\end{bmatrix}
\]

\[
\frac{dp}{dp_d} = \begin{bmatrix}
\frac{\partial u}{\partial x_d} & \frac{\partial u}{\partial y_d} \\
\frac{\partial v}{\partial x_d} & \frac{\partial v}{\partial y_d} \\
\frac{\partial x_d}{\partial x_d} & \frac{\partial y_d}{\partial x_d}
\end{bmatrix} = \begin{bmatrix}
f_x & 0 \\
0 & f_y
\end{bmatrix}
\]
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

\[
\frac{dp_d}{dk^T} = \left[\begin{array}{cccccc}
\frac{\partial x_d}{\partial k_1} & \frac{\partial x_d}{\partial k_2} & \frac{\partial x_d}{\partial \rho_1} & \frac{\partial x_d}{\partial \rho_2} & \frac{\partial x_d}{\partial k_3} \\
\frac{\partial y_d}{\partial k_1} & \frac{\partial y_d}{\partial k_2} & \frac{\partial y_d}{\partial \rho_1} & \frac{\partial y_d}{\partial \rho_2} & \frac{\partial y_d}{\partial k_3}
\end{array}\right] = \left[\begin{array}{cccccc}
x_n r^2 & x_n r^4 & 2x_n y_n & r^2 + 2x_n^2 & x_n r^6 \\
y_n r^2 & y_n r^4 & r^2 + 2y_n^2 & 2x_n y_n & y_n r^6
\end{array}\right]
\]

Then we have,

\[
\frac{dp}{dk^T} = \frac{dp}{dp_d^T} \cdot \frac{dp_d}{dk^T} = \left[\begin{array}{cccccc}
f_x x_n r^2 & f_x x_n r^4 & 2f_x x_n y_n & f_x (r^2 + 2x_n^2) & f_x x_n r^6 \\
f_y y_n r^2 & f_y y_n r^4 & f_y (r^2 + 2y_n^2) & 2f_y x_n y_n & f_y y_n r^6
\end{array}\right]
\]

Also based on Eq. 7, we can have

\[
\frac{dp_d}{dp_n^T} = \left[\begin{array}{cccc}
\frac{\partial x_d}{\partial x_n} & \frac{\partial x_d}{\partial y_n} \\
\frac{\partial y_d}{\partial x_n} & \frac{\partial y_d}{\partial y_n}
\end{array}\right] = [...]
\]

Its concrete form is a little complicated, but not difficult

Assignment!
Nonlinear Least-squares

According to Eq, 3, we have

$$\frac{dp_a}{dP_C} = \begin{bmatrix} \frac{\partial x_a}{\partial X_C} & \frac{\partial x_a}{\partial Y_C} & \frac{\partial x_a}{\partial Z_C} \\ \frac{\partial y_a}{\partial X_C} & \frac{\partial y_a}{\partial Y_C} & \frac{\partial y_a}{\partial Z_C} \\ \frac{\partial z_a}{\partial X_C} & \frac{\partial z_a}{\partial Y_C} & \frac{\partial z_a}{\partial Z_C} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -X_C \\ 0 & 1 & -Y_C \\ Z_C & Z_C^2 & 0 \end{bmatrix}$$

According to Eq, 1, we have

$$\mathbf{P}_C = \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \begin{bmatrix} R_{11}X + R_{12}Y + R_{13}Z + t_1 \\ R_{21}X + R_{22}Y + R_{23}Z + t_2 \\ R_{31}X + R_{32}Y + R_{33}Z + t_3 \end{bmatrix} \quad \Rightarrow \quad \frac{d\mathbf{P}_C}{d\mathbf{r}^T} = \begin{bmatrix} \frac{\partial X_C}{\partial R_{11}} & \frac{\partial X_C}{\partial R_{12}} & \frac{\partial X_C}{\partial R_{13}} & \frac{\partial X_C}{\partial R_{21}} & \frac{\partial X_C}{\partial R_{22}} & \frac{\partial X_C}{\partial R_{23}} & \frac{\partial X_C}{\partial R_{31}} & \frac{\partial X_C}{\partial R_{32}} & \frac{\partial X_C}{\partial R_{33}} \\ \frac{\partial Y_C}{\partial R_{11}} & \frac{\partial Y_C}{\partial R_{12}} & \frac{\partial Y_C}{\partial R_{13}} & \frac{\partial Y_C}{\partial R_{21}} & \frac{\partial Y_C}{\partial R_{22}} & \frac{\partial Y_C}{\partial R_{23}} & \frac{\partial Y_C}{\partial R_{31}} & \frac{\partial Y_C}{\partial R_{32}} & \frac{\partial Y_C}{\partial R_{33}} \\ \frac{\partial Z_C}{\partial R_{11}} & \frac{\partial Z_C}{\partial R_{12}} & \frac{\partial Z_C}{\partial R_{13}} & \frac{\partial Z_C}{\partial R_{21}} & \frac{\partial Z_C}{\partial R_{22}} & \frac{\partial Z_C}{\partial R_{23}} & \frac{\partial Z_C}{\partial R_{31}} & \frac{\partial Z_C}{\partial R_{32}} & \frac{\partial Z_C}{\partial R_{33}} \end{bmatrix} = \begin{bmatrix} X & Y & Z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X & Y & Z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & X & Y & Z \end{bmatrix}$$

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

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

$$\frac{dr}{dd^T} \in \mathbb{R}^{9 \times 3}$$

Assignment!

Then, we can compute,

$$\frac{dp}{dd^T} = \frac{dp}{dp_d^T} \cdot \frac{dp_d}{dp_n^T} \cdot \frac{dp_n}{dP_C^T} \cdot \frac{dr}{dr^T}$$

$$\frac{dp}{dt^T} = \frac{dp}{dp_d^T} \cdot \frac{dp_d}{dp_n^T} \cdot \frac{dp_n}{dP_C^T} \cdot \frac{dt}{dt^T}$$
Nonlinear Least-squares

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

One naive example, the distorted image can be undistorted

\[
\begin{pmatrix}
u \\
v
\end{pmatrix}
\]

One point on the undistorted image

The corresponding point on the original image with distortion

\[
K\mathcal{D}\begin{pmatrix}
u \\
v \\
1
\end{pmatrix}
\]
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

![Diagram showing the relationship between Bird’s-eye-view image, WCS, and Undistorted image through Similarity and Projective transformations.](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 $P_{B\rightarrow W}$ and the transformation matrix from WCS to the undistorted image is $P_{W\rightarrow I}$

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

$$x_I = P_{W\rightarrow I}P_{B\rightarrow W} \begin{pmatrix} x_B \\ y_B \\ 1 \end{pmatrix}$$

Then, the intensity of the pixel $(x_B, y_B, 1)^T$ can be determined using some interpolation technique based on the neighborhood around $x_I$ 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 $P_{B \rightarrow W}$ and the transformation matrix from WCS to the undistorted image is $P_{W \rightarrow I}$.

The key problem is how to obtain $P_{B \rightarrow W}$ and $P_{W \rightarrow I}$?
Bird’s-eye-view Generation

- Determine $P_{B \rightarrow W}$

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 $P_{B \rightarrow W}$

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

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

Please verify!!
Bird’s-eye-view Generation

• Determine $P_{W \rightarrow I}$

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

\[ x_I = P_{W \rightarrow I} x_W \]

If we know a set of correspondence pairs $\{x_{ii}, x_{wi}\}_{i=1}^N$, $P_{W \rightarrow I}$ can be estimated using the least-square method.
Bird’s-eye-view Generation

- Determine $P_{W ightarrow I}$

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

$$
\mathbf{x}_I = P_{W\rightarrow I} P_{B\rightarrow W} \begin{pmatrix} x_B \\ y_B \\ 1 \end{pmatrix} \equiv P_{B\rightarrow I} \begin{pmatrix} x_B \\ y_B \\ 1 \end{pmatrix}
$$
Bird-view Generation

Another example

Original fish-eye image

Undistorted image
Bird-view Generation

Another example

Original fish-eye image  Bird’s-eye-view