

Assignment2

Computer Vision 2023 Spring

1953921 陈元哲

1

(Math) In the augmented Euclidean plane, there is a line $x - 3y + 4 = 0$, what is the homogeneous coordinate of the infinity point of this line?

The homogeneous coordinate of the line is $x - 3y + 4z = 0$. In the projective plane, a point at infinity is represented by setting $z = 0$. We plug $z = 0$ into the equation to find the homogeneous coordinates of the infinity point

$$x - 3y = 0$$

let $x = 3$, we have $y = 1$

Therefore, the homogeneous coordinate of the infinity point is $(3, 1, 0)$.

2

(Math) On the normalized retinal plane, suppose that \mathbf{p}_n is an ideal point of projection without considering distortion. If distortion is considered, $\mathbf{p}_n = (x, y)^T$ is mapped to $\mathbf{p}_d = (x_d, y_d)^T$ which is also on the normalized retinal plane. Their relationship is,

$$\begin{cases} x_d = x(1 + k_1 r^2 + k_2 r^4) + 2\rho_1 xy + \rho_2(r^2 + 2x^2) + xk_3 r^6 \\ y_d = y(1 + k_1 r^2 + k_2 r^4) + 2\rho_2 xy + \rho_1(r^2 + 2y^2) + yk_3 r^6 \end{cases}$$

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

For performing nonlinear optimization in the pipeline of camera calibration, we need to compute the Jacobian matrix of \mathbf{p}_d w.r.t \mathbf{p}_n , i.e.,

$$\frac{d\mathbf{p}_d}{d\mathbf{p}_n^T}$$

It should be noted that in this question \mathbf{p}_d is the function of \mathbf{p}_n and all the other parameters can be regarded as constants.

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

$$\begin{cases} x_d = x[1 + k_1(x^2 + y^2) + k_2(x^4 + 2x^2y^2 + y^4)] + 2\rho_1 xy + \rho_2(3x^2 + y^2) + xk_3(x^6 + 3x^4y^2 + 3x^2y^4 + y^6) \\ y_d = y[1 + k_1(x^2 + y^2) + k_2(x^4 + 2x^2y^2 + y^4)] + 2\rho_2 xy + \rho_1(x^2 + 3y^2) + yk_3(x^6 + 3x^4y^2 + 3x^2y^4 + y^6) \end{cases}$$

$\mathbf{p}_n = (x, y)^T$ is mapped to $\mathbf{p}_d = (x_d, y_d)^T$, we have $F: \mathbb{R}_2 \rightarrow \mathbb{R}_2$

$$\begin{aligned} \frac{d\mathbf{p}_d}{d\mathbf{p}_n^T} &= J_F(x, y) = \begin{bmatrix} \frac{\partial x_d}{\partial x} & \frac{\partial x_d}{\partial y} \\ \frac{\partial y_d}{\partial x} & \frac{\partial y_d}{\partial y} \end{bmatrix} \\ &= \begin{bmatrix} 1 + k_1(3x^2 + y^2) + k_2(5x^4 + 6x^2y^2 + y^4) + & 2k_1xy + k_2(4x^3y + 4xy^3) + \\ 2\rho_1y + 6\rho_2x + k_3(7x^6 + 15x^4y^2 + 9x^2y^4 + y^6) & 2\rho_1x + 2\rho_2y + k_3(6x^5y + 12x^3y^3 + 6xy^5) \\ 2k_1xy + k_2(4x^3y + 4xy^3) + & 1 + k_1(x^2 + 3y^2) + k_2(x^4 + 6x^2y^2 + 5y^4) + \\ 2\rho_1x + 2\rho_2y + k_3(6x^5y + 12x^3y^3 + 6xy^5) & 2\rho_2x + 6\rho_1y + k_3(x^6 + 9x^4y^2 + 15x^2y^4 + 7y^6) \end{bmatrix} \end{aligned}$$

3

(Math) In our lecture, we mentioned that for performing nonlinear optimization in the pipeline of camera calibration, we need to compute the Jacobian of the rotation matrix (represented in a vector) w.r.t its axis-angle representation. In this question, your task is to derive the concrete formula of this Jacobian matrix. Suppose that

$$\mathbf{d} = \theta \mathbf{n} \in \mathbb{R}^{3 \times 1}, \text{ where } \mathbf{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \text{ is a 3D unit vector and } \theta \text{ is a real number denoting the rotation angle.}$$

With Rodrigues formula, \mathbf{d} can be converted to its rotation matrix form,

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

$$\text{and obviously } \mathbf{R} \triangleq \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \text{ is a } 3 \times 3 \text{ matrix.}$$

Denote \mathbf{r} by the vectorized form of \mathbf{R} , i.e.,

$$\mathbf{r} \triangleq (r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33})^T$$

Please give the concrete form of Jacobian matrix of \mathbf{r} w.r.t \mathbf{d} , i.e., $\frac{d\mathbf{r}}{d\mathbf{d}^T} \in \mathbb{R}^{9 \times 3}$.

In order to make it easy to check your result, please follow the following notation requirements,

$$\alpha \triangleq \sin \theta, \beta \triangleq \cos \theta, \gamma \triangleq 1 - \cos \theta$$

In other words, the ingredients appearing in your formula are restricted to $\alpha, \beta, \gamma, \theta, n_1, n_2, n_3$.

We have

$$\mathbf{n} \mathbf{n}^T = \begin{bmatrix} n_1^2 & n_1 n_2 & n_1 n_3 \\ n_1 n_2 & n_2^2 & n_2 n_3 \\ n_1 n_3 & n_2 n_3 & n_3^2 \end{bmatrix} \quad \mathbf{n}^\wedge = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}$$

We plug them into \mathbf{R}

$$\mathbf{R} = \begin{bmatrix} \cos \theta + n_1^2(1 - \cos \theta) & n_1 n_2(1 - \cos \theta) - n_3 \sin \theta & n_1 n_3(1 - \cos \theta) + n_2 \sin \theta \\ n_1 n_2(1 - \cos \theta) + n_3 \sin \theta & \cos \theta + n_2^2(1 - \cos \theta) & n_2 n_3(1 - \cos \theta) - n_1 \sin \theta \\ n_1 n_3(1 - \cos \theta) - n_2 \sin \theta & n_2 n_3(1 - \cos \theta) + n_1 \sin \theta & \cos \theta + n_3^2(1 - \cos \theta) \end{bmatrix}$$

let $\mathbf{d} = (d_1, d_2, d_3)^T$, we have

$$\mathbf{d} = \theta \mathbf{n} = \begin{bmatrix} \theta n_1 \\ \theta n_2 \\ \theta n_3 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

$$d_i = \theta n_i \quad (i = 1, 2, 3)$$

\mathbf{n} is a 3D unit vector, therefore

$$d_1^2 + d_2^2 + d_3^2 = \theta^2(n_1^2 + n_2^2 + n_3^2) = \theta^2$$

We plug the relationship above, have

$$\begin{aligned} \frac{\partial \alpha}{\partial d_i} &= \frac{\partial \sin \theta}{\partial d_i} = \cos \theta \frac{\partial \theta}{\partial d_i} = \cos \theta \frac{\partial \sqrt{d_1^2 + d_2^2 + d_3^2}}{\partial d_i} = \cos \theta \frac{d_i}{\theta} = n_i \cos \theta \\ \frac{\partial \beta}{\partial d_i} &= \frac{\partial \cos \theta}{\partial d_i} = -\sin \theta \frac{\partial \theta}{\partial d_i} = -n_i \sin \theta \\ \frac{\partial \gamma}{\partial d_i} &= \frac{\partial 1 - \cos \theta}{\partial d_i} = -\frac{\partial \cos \theta}{\partial d_i} = n_i \sin \theta \\ \frac{\partial n_i}{\partial d_i} &= \frac{\partial \frac{d_i}{\theta}}{\partial d_i} = \frac{\theta - d_i \frac{\partial \theta}{\partial d_i}}{\theta^2} = \frac{\theta - \frac{d_i^2}{\theta}}{\theta^2} = \frac{1 - n_i^2}{\theta} \\ \frac{\partial n_j}{\partial d_i} &= \frac{\partial \frac{d_j}{\theta}}{\partial d_i} = \frac{-d_j \frac{\partial \theta}{\partial d_i}}{\theta^2} = \frac{-\frac{d_i d_j}{\theta}}{\theta^2} = -\frac{n_i n_j}{\theta} \quad (i \neq j) \end{aligned}$$

$\mathbf{d} = (d_1, d_2, d_3)^T$ is mapped to $\mathbf{r} = (r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33})^T$, we have $F: \mathbb{R}_3 \rightarrow \mathbb{R}_9$

$$\frac{dr}{dd^T} = J_F(r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33})$$

$$= \begin{bmatrix} \frac{\partial r_{11}}{\partial d_1} & \frac{\partial r_{11}}{\partial d_2} & \frac{\partial r_{11}}{\partial d_3} \\ \frac{\partial r_{12}}{\partial d_1} & \frac{\partial r_{12}}{\partial d_2} & \frac{\partial r_{12}}{\partial d_3} \\ \vdots & \vdots & \vdots \\ \frac{\partial r_{33}}{\partial d_1} & \frac{\partial r_{33}}{\partial d_2} & \frac{\partial r_{33}}{\partial d_3} \end{bmatrix}$$

$$= \begin{bmatrix} n_1(1 - n_1^2)(\frac{2\gamma}{\theta} - \alpha) & n_1^2 n_2(\alpha - \frac{2\gamma}{\theta}) - n_2 \alpha & n_1^2 n_3(\alpha - \frac{2\gamma}{\theta}) - n_3 \alpha \\ n_1^2 n_2(\alpha - \frac{2\gamma}{\theta}) + n_1 n_3(\frac{\alpha}{\theta} - \beta) + n_2 \frac{\gamma}{\theta} & n_1 n_2^2(\alpha - \frac{2\gamma}{\theta}) + n_2 n_3(\frac{\alpha}{\theta} - \beta) + n_1 \frac{\gamma}{\theta} & n_1 n_2 n_3(\alpha - \frac{2\gamma}{\theta}) - (1 - n_3^2) \frac{\alpha}{\theta} - n_3^2 \beta \\ n_1^2 n_3(\alpha - \frac{2\gamma}{\theta}) - n_1 n_2(\frac{\alpha}{\theta} - \beta) + n_3 \frac{\gamma}{\theta} & n_1 n_2 n_3(\alpha - \frac{2\gamma}{\theta}) + (1 - n_2^2) \frac{\alpha}{\theta} + n_2^2 \beta & n_1 n_3^2(\alpha - \frac{2\gamma}{\theta}) - n_2 n_3(\frac{\alpha}{\theta} - \beta) + n_1 \frac{\gamma}{\theta} \\ n_1^2 n_2(\alpha - \frac{2\gamma}{\theta}) - n_1 n_3(\frac{\alpha}{\theta} - \beta) + n_2 \frac{\gamma}{\theta} & n_1 n_2^2(\alpha - \frac{2\gamma}{\theta}) - n_2 n_3(\frac{\alpha}{\theta} - \beta) + n_1 \frac{\gamma}{\theta} & n_1 n_2 n_3(\alpha - \frac{2\gamma}{\theta}) + (1 - n_3^2) \frac{\alpha}{\theta} + n_3^2 \beta \\ n_1 n_2^2(\alpha - \frac{2\gamma}{\theta}) - n_1 \alpha & n_2(1 - n_2^2)(\frac{2\gamma}{\theta} - \alpha) & n_2^2 n_3(\alpha - \frac{2\gamma}{\theta}) - n_3 \alpha \\ n_1 n_2 n_3(\alpha - \frac{2\gamma}{\theta}) - (1 - n_1^2) \frac{\alpha}{\theta} - n_1^2 \beta & n_2^2 n_3(\alpha - \frac{2\gamma}{\theta}) + n_1 n_2(\frac{\alpha}{\theta} - \beta) + n_3 \frac{\gamma}{\theta} & n_2 n_3^2(\alpha - \frac{2\gamma}{\theta}) + n_1 n_3(\frac{\alpha}{\theta} - \beta) + n_2 \frac{\gamma}{\theta} \\ n_1^2 n_3(\alpha - \frac{2\gamma}{\theta}) + n_1 n_2(\frac{\alpha}{\theta} - \beta) + n_3 \frac{\gamma}{\theta} & n_1 n_2 n_3(\alpha - \frac{2\gamma}{\theta}) - (1 - n_2^2) \frac{\alpha}{\theta} - n_2^2 \beta & n_1 n_3^2(\alpha - \frac{2\gamma}{\theta}) + n_2 n_3(\frac{\alpha}{\theta} - \beta) + n_1 \frac{\gamma}{\theta} \\ n_1 n_2 n_3(\alpha - \frac{2\gamma}{\theta}) + (1 - n_1^2) \frac{\alpha}{\theta} + n_1^2 \beta & n_2^2 n_3(\alpha - \frac{2\gamma}{\theta}) - n_1 n_2(\frac{\alpha}{\theta} - \beta) + n_3 \frac{\gamma}{\theta} & n_2 n_3^2(\alpha - \frac{2\gamma}{\theta}) - n_1 n_3(\frac{\alpha}{\theta} - \beta) + n_2 \frac{\gamma}{\theta} \\ n_1 n_3^2(\alpha - \frac{2\gamma}{\theta}) - n_1 \alpha & n_2 n_3^2(\alpha - \frac{2\gamma}{\theta}) - n_2 \alpha & n_3(1 - n_3^2)(\alpha - \frac{2\gamma}{\theta}) \end{bmatrix}$$

4

(Programming) Bird's-eye-view generation. The geometric transform between the physical plane and its bird's-eye-view image can be simply described by a **similarity transformation** matrix. Bird's-eye-view is very useful in autonomous industrial inspection, ADAS, etc. In this question, your task is to create the bird's-eye-view image of a physical plane, e.g., the wall of your room. For this purpose, you may need to,

- 1) make a calibration board with chessboard patterns;
- 2) calibrate your camera (the camera mounted on your laptop or the camera of your mobile phone with fixed focal length) to get its intrinsics;
- 3) attach regular patterns (e.g., chessboard patterns) to the wall, determine the 2D coordinate system C_W of the wall, and determine the coordinates $\{\mathbf{x}_{Wi}\}_{i=1}^N$ of the feature points of the regular patterns with respect to C_W ;
- 4) take the image I_d of the wall with regular patterns;
- 5) undistort image I_d with the camera's intrinsics to get the undistorted image I ;
- 6) For each \mathbf{x}_{Wi} , determine its image \mathbf{x}_{Ii} on I ;
- 7) solve the homography matrix $P_{W \rightarrow I}$ between the wall and the image I wall using $\{\mathbf{x}_{Wi} \leftrightarrow \mathbf{x}_{Ii}\}_{i=1}^N$;
- 8) generate the final bird's-eye-view image of the wall using the technique introduced in our lecture.

For submission, you **only** need to submit the following items to TA:

- 1) the intrinsic parameters of your camera;
- 2) the original image of the wall (or other physical planes) taken by your camera; make sure that your name is painted or attached on the wall (or the plane); (maybe similar to following image I provide to you)
- 3) the generated bird's-eye-view image of the wall (or other physical planes).

Where is my submission

1. the intrinsic parameters of my camera

In the file `camera_intrinsic.txt`, here is the content

```
Camera Matrix:
[1658.360919958824, 0, 1210.66914163179;
 0, 1626.412676277487, 556.1001190421206;
 0, 0, 1]
Distortion Coefficients:
[-0.2582572766295772, 1.287844698344425, 0, 0, -0.8617458966204036]
```

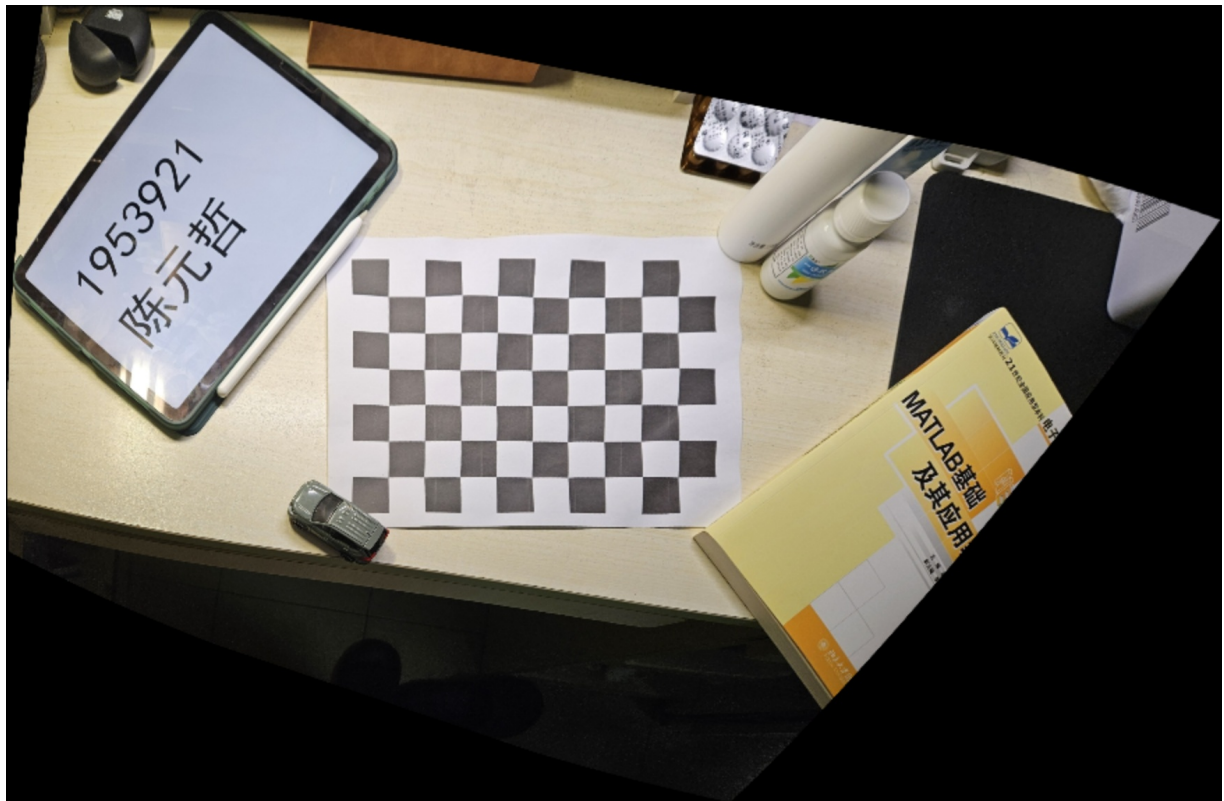
2. the original image of the physical planes taken by my camera

In the file `input.jpg`



3. the generated BEV image

In the file `result.jpg`



And the source code is in file `BEV.cpp`