Camera Calibration

Chun-Jui Lai

Outline

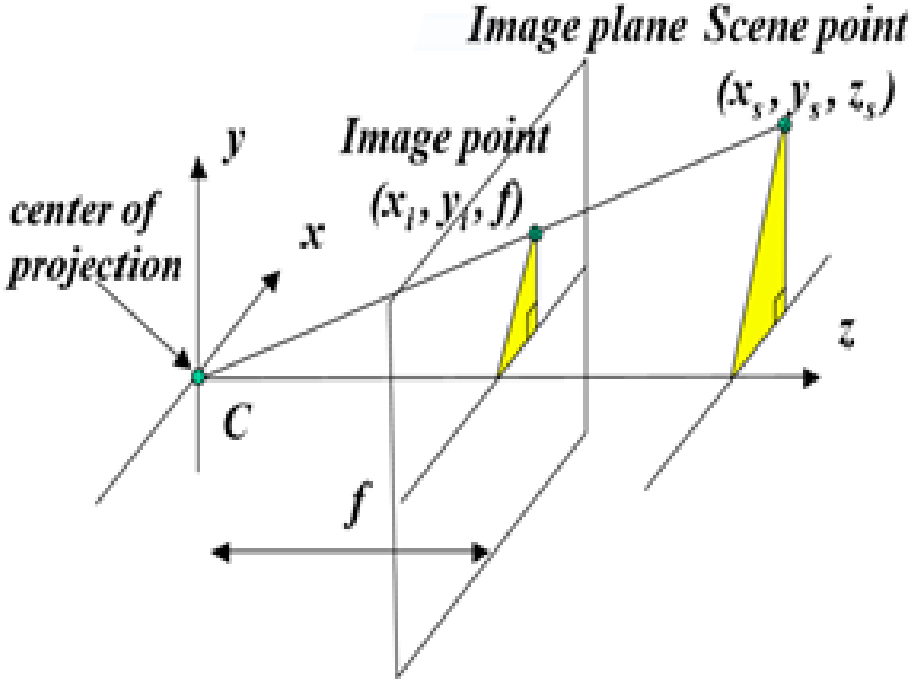
- Camera Parameter
- Distortion Coefficient
- Camera calibration
- Camera calibration sample

Camera Parameter

- Intrinsic Parameter(A)

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \frac{1}{z_s} \times A \times \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

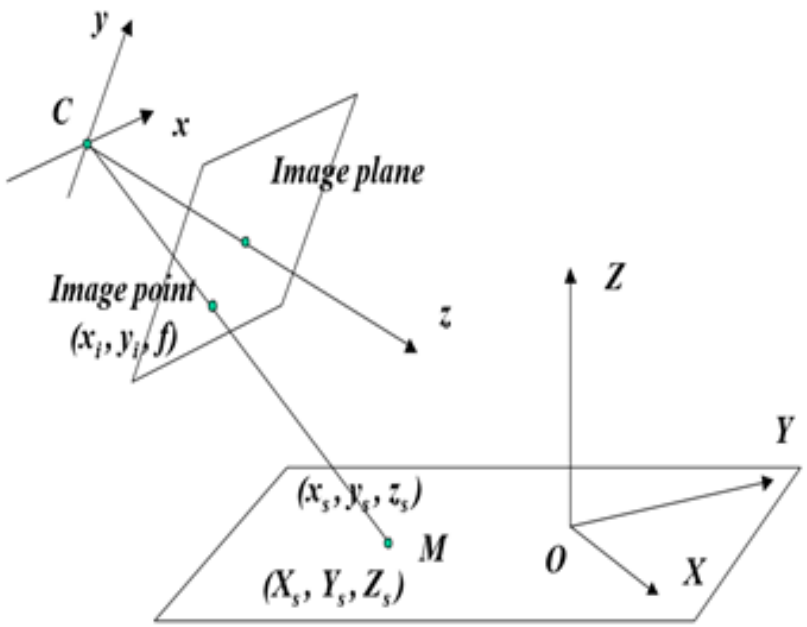


Camera Parameter

- Extrinsic parameter [Rotate|Translate]

$$\begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = [R|T] \times \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix}$$

$$[\mathrm{R}|\mathrm{T}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$



Distortion Coefficient

- For normal camera (narrow FOV)
- (x, y) image coordinate, z focal length

$$a = x/z \text{ and } b = y/z$$

$$r^2 = a^2 + b^2$$

Radial distortion

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

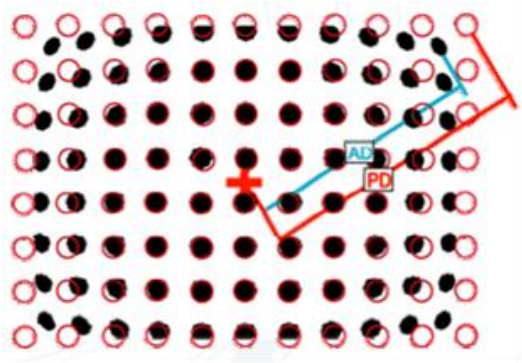
$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

Tangential distortion

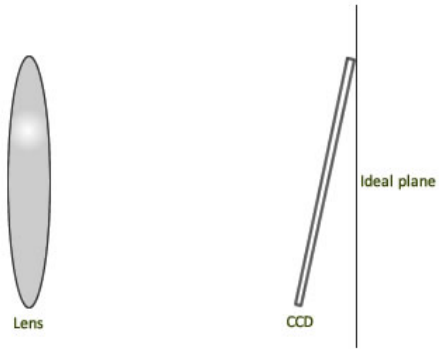
$$x_{corrected} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$

$$Distortion_{coefficients} = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3)$$



Radial distortion



Tangential distortion

Distortion Coefficient

- For fish-eye camera (wide FOV)
- (x, y) image coordinate, z focal length

$$a = x/z \text{ and } b = y/z$$

$$r^2 = a^2 + b^2$$

$$\theta = \text{atan}(r)$$

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

Camera Calibration

- OpenCV : Zhang's method[1] to find the intrinsic parameter
- Capture several images with chessboard/regular pattern/ obvious features

Camera Calibration

- Let intrinsic A = $\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$, extrinsic = $[r_1\ r_2\ r_3\ t]$
- Image coordinate = $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$
- chessboard coordinate = $\begin{bmatrix} X \\ Y \\ 0 \end{bmatrix}$,

  WLOG, we assume Z of chessboard is zero

Camera Calibration

- By camera model, we have

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[r_1 \;\; r_2 \;\; r_3 \;\; t]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A[r_1 \quad r_2 \quad t]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

$$=> s\tilde{m} = H\tilde{M},$$

$$\tilde{m} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \mathrm{H} = A[r_1 \;\; r_2 \; t] \;, \tilde{M} = \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Camera Calibration

- Let homography H = $[h_1 \;\; h_2 \;\; h_3] = \lambda A[r_1 \;\; r_2 \;\; t]$
- Because $r_1 \;\; r_2 \;\; r_3$ are orthogonal to each other,
- We have following equation

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 = 0$$
$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2$$

We can solve H by enough chessboard corners.

See [1] for detail.

Camera Calibration

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

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

Camera Calibration

- B is symmetric matrix, let

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

$$h_i = [h_{i1}\ h_{i2}\ h_{i3}]^T$$

- We can get following equation

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

$$\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2},$$
$$h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T$$

Camera Calibration

- For each corner(feature), we have following equation

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 = 0$$
$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2$$

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

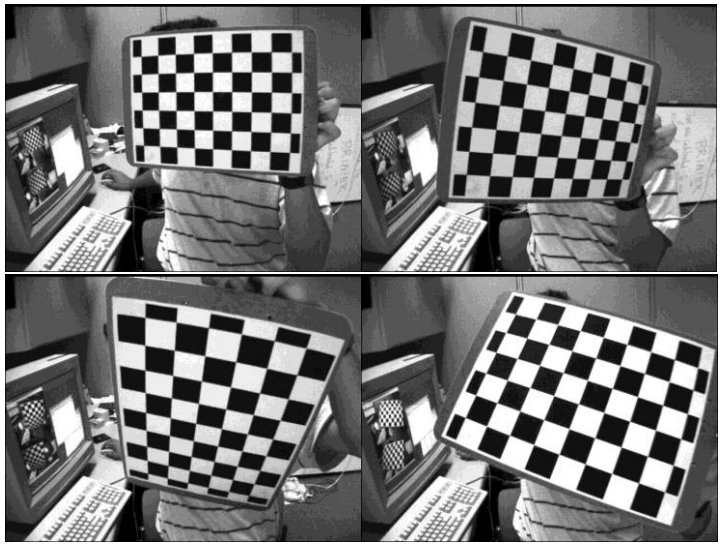
- For n images, we have $\mathbf{V}\mathbf{b} = \mathbf{0}$
- If n > 3 the equation can be solved
- By solving SVD, we can get b, then we can solve intrinsic A

Camera Calibration sample

- OpenCV calibration sample code
- Source files can be found in “opencv\sources\samples\cpp\tutorial_code\calib3d\camera_calibration”

Camera Calibration Example

- Step1. Capture chessboard images with different views(> 5 is better)



Chessboard images

Camera Calibration Example

- Step2. setting configuration file(in_VID5.xml)

```
<?xml version="1.0"?>
<opencv_storage>
<Settings>
  <!-- Number of inner corners per a item row and column. (square, circle) -->
  <BoardSize_Width> 9</BoardSize_Width>
  <BoardSize_Height>6</BoardSize_Height>

  <!-- The size of a square in some user defined metric system (pixel, millimeter)-->
  <Square_Size>24</Square_Size>

  <!-- The type of input used for camera calibration. One of: CHESSBOARD CIRCLES_GRID ASYMMETRIC_CIRCLES_GRID -
  <Calibrate_Pattern>"CHESSBOARD"</Calibrate_Pattern>

  <!-- The input to use for calibration.
       To use an input camera -> give the ID of the camera, like "1"
       To use an input video  -> give the path of the input video, like "/tmp/x.avi"
       To use an image list   -> give the path to the XML or YAML file containing the list of the images, like
       -->
  <Input>"cameracalibration/VID5.xml"</Input>
  <!--  If true (non-zero) we flip the input images around the horizontal axis.-->
  <Input_FlipAroundHorizontalAxis>0</Input_FlipAroundHorizontalAxis>
```

Chessboard inner corner size

Chessboard square size(mm)

Input image setting file path

Camera Calibration Example

```
<!-- Time delay between frames in case of camera. -->
<Input_Delay>100</Input_Delay>

<!-- How many frames to use, for calibration. -->
<Calibrate_NrOfFrameToUse>29</Calibrate_NrOfFrameToUse>
<!-- Consider only fy as a free parameter, the ratio fx/fy stays the same as in the input cameraMatrix.
     Use or not setting. 0 - False Non-Zero - True-->
<Calibrate_FixAspectRatio> 1 </Calibrate_FixAspectRatio>
<!-- If true (non-zero) tangential distortion coefficients  are set to zeros and stay zero.-->
<Calibrate_AssumeZeroTangentialDistortion>1</Calibrate_AssumeZeroTangentialDistortion>
<!-- If true (non-zero) the principal point is not changed during the global optimization.-->
<Calibrate_FixPrincipalPointAtTheCenter> 1 </Calibrate_FixPrincipalPointAtTheCenter>

<!-- The name of the output log file. -->
<Write_outputFileName>"cameracalibration/out_data.xml"</Write_outputFileName>
<!-- If true (non-zero) we write to the output file the feature points.-->
<Write_DetectedFeaturePoints>1</Write_DetectedFeaturePoints>
<!-- If true (non-zero) we write to the output file the extrinsic camera parameters.-->
<Write_extrinsicParameters>1</Write_extrinsicParameters>
<!-- If true (non-zero) we show after calibration the undistorted images.-->
<Show_UndistortedImage>1</Show_UndistortedImage>

</Settings>
</opencv_storage>
```

Camera parameter output file path

Camera Calibration Example

- Step3. setting input file(VID5.xml)

```
<?xml version="1.0"?>
<opencv_storage>
<images>
cameracalibration/1.png
cameracalibration/2.png
cameracalibration/3.png
cameracalibration/4.png
cameracalibration/5.png
cameracalibration/6.png
cameracalibration/7.png
cameracalibration/8.png
cameracalibration/9.png
</images>
</opencv_storage>
```

Input images path

Camera Calibration Example

- Step4. command argument

  configuration file path

  (ex. “cameracalibration/in_VID5.xml”)
- Step5. just build and run

Reference

- [1]Zhang, Zhengyou. "A flexible new technique for camera calibration."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.11 (2000): 1330-13
- [2]Camera calibration With OpenCV

  http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html