Project Stereo

Author: Yongshun Zhang

March 25, 2018

1 Camera Basics

1.1 the Intrinsics and the Extrinsics of a Camera and the Camera Matrix

A **Camera Matrix** is used to denote a projective mapping from World Coordinates to Pixel Coordinates, which can be represented as follows. [1]

$$Camera\ Matirx\ \ \mathbf{P} = K \times \begin{bmatrix} R & T \end{bmatrix} \tag{1}$$

In equation (1), the matrix **K** is called intrisic parameters, and the matrix [R T] is the extrinsic parameters.

The intrinsic matrix **K** contains 5 intrinsic, and has the following form.

$$\mathbf{K} = \begin{bmatrix} \alpha_x & \gamma & \mu_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

These parameters encompass focal length, image sensor format and principal point(image centre). Before introducing the intrinsic matrix ,we need to know how the pixel skew is define, which is shown in Figure 1 [2].
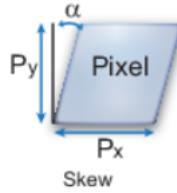

Figure 1: skew

The meanings and metrics of the parameters α_x, α_y, γ, μ_0, v_0 are described in tabulation 1 [1] [2].

$(\mu_0,\ v_0)$ // Principal point, in pixels
$(\alpha_x,\ \alpha_y)$ //focal length, in pixels $\alpha_x \leftarrow F/p_x, \alpha_y \leftarrow F/p_y$ F // Focal length in world units, typically expressed in millimeters $(p_x,\ p_y)$ // Size of the pixel in world units
s $\leftarrow$ $alpha_y * tan(\alpha)$ // Skew coefficient

Table 1: parameters of intrinsic matrix

Extrinsic parameters consisting of matrix **R**,**T**, are the extrinsic parameters which denote the coordinate system transformations from 3D world coordinates to 3D camera coordinates. Extrinsic parameters define the position of camera center and the camera's heading in real world. **T** is the position of the origin of the world coordinate system expressed in coordinates of camera-centered coordinate system, **R** is a rotation matrix used to perform a rotation in Euclidean space, which properties are $\mathbf{R}^T = \mathbf{R}^{-1}$ and det **R**=1 [3].

Often, we use $[\mu \ \upsilon \ 1]^T$ to represent a 2D point position in pixel coordinates, and $[x_w \ y_w \ z_w \ 1]^T$ is used to present a 3D point position in world coordinates, which are expressed in augmented notation of Homogeneous coordinates. Referring to the pinhole camera model, the camera matrix denotes a projective mapping from world coordinates to pixel coordinates.

$$z^c \times \begin{bmatrix} \mu \\ \upsilon \\ 1 \end{bmatrix} = K \times \begin{bmatrix} R & T \end{bmatrix} \times \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{3}$$

z_c represents the coordinate of z axis in used to make the constant of proportionality of two matrices [4].

Problems There are some little problems during learning Camera Matrix. The first is the position, **C**, of the camera expressed in world coordinate, I didn't understand the equation $\mathbf{C} = -\mathbf{R}_{-1}\mathbf{T} = -\mathbf{R}_T\mathbf{T}$ originally, and I find the answer in the meaning of the Homogeneous coordinates. When we transfer the coordinates from 3D world to 2D pixel, the coordinate expressed in Homogeneous coordinates can realize rotation, zoom and translation, so translation T, which represents the origin of world coordinates expressed in pixel coordinates, used to make the 2D pixel coordinate taking the origin of pixel coordinates(image centre) as its origin of coordinates. And the second problem is that the five intrinsic parameters in matrix **K** are expressed indistinct in wikipedia, and I found the distinct expression in MathWorks website [2], which is shown in tabulation 1.

Written by Yongshun, 22:00, 21th Mar 2018

1.2 Camera Imaging

Problems When I start to deal with the transform question, I find that I'm still in a preliminary understanding of camera resectioning and camera matrix, so I read the detailed introduction of **Camera Matrix** [5] and the **Pinhole Camera model** [6] from wikipedia firstly, and have a progressive understanding of Camera Matrix including its relation with pinhole camera model and the derivation of its equation. But the next problem comes, I find that I'm confused of the axes' directions and the relationship between principal point's coordinate and optical centre's coordinate.

So I try to search the relationship between principal point and optical centre by Google, such as "the relationship between principal point and optical centre" and also try to search in Chinese, finally I found the relationship between in wikipedia "Pinhole Camera Model" [6], it tells "A point R at the intersection of the optical axis and the image plane. This point is referred to as the principal point or image center.". As for the axes' directions, I review the web pages about the camera matrix, camera resectioning I have read, and also read some documents and blogs [7] [8] [9], and finally solve the problem temporarily.

The answer of transform is as follows.

First, transform the 3D point **X** from World coordinates to Camera coordinates, using the extrinsic parameters.$[X_c, \ Y_c, \ Z_c]_T$ represents the point in Camera coordinates.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} R & T \end{bmatrix} \times \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix} \tag{4}$$

Second, transform the 3D point in Camera coordinates to 2D point in Image plane.

$$Z_c \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \tag{5}$$

Finally, transform the 2D point from image plane to pixel coordinates.

$$u = x + c_x, \quad v = y + c_y \tag{6}$$

All equations above can be replaced with the following one.

$$Z_c \times \begin{bmatrix} \mu \\ \upsilon \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} R & t \end{bmatrix} \times \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix} \tag{7}$$

1.3 Relationship Between 2D Image Point and 3D Camera Coordinate

A 2D image point (μ, υ) correspond to a line in 3D camera coordinate. The derivation is as follows.

First, transform the image point (μ, υ) from pixel coordinate to image plane.

$$x = (\mu - \mu_0), \quad y = (\upsilon - \upsilon_0) \tag{8}$$

Second, according to the pinhole camera model, transform the point from image plane to camera coordinate.

$$X_c = x \times \frac{Z_c}{f}, \quad Y_c = y \times \frac{Z_c}{f} \tag{9}$$

According to equation (9), we set $\frac{x}{f} = k$, $\frac{y}{f} = t$, k and t are constants, so the 3D point in camera coordinate can be describe as $Z_c \times (k, t, 1)$, which represents a line in 3D coordinate.

Written by Yongshun, 00:43, 22th Mar 2018

1.4 Distortion

When I read the Camera Calibration and 3D Construction in opencv website [10], I find that I once read this document which has been translated in Chinese [9], and I read the document again carefully and review the introduction of distortion in Mathworks [2].

As for distortions in camera calibration, it can be classified into two types: radial distortion and tangential distortion, real lenses usually have some distortion, mostly radial distortion and slight tangential distortion.

Figure 2 show two common types of Radial distortion: barrel distortion(typically $k_1 > 0$) and pincushion distortion(typically $k_1 < 0$).
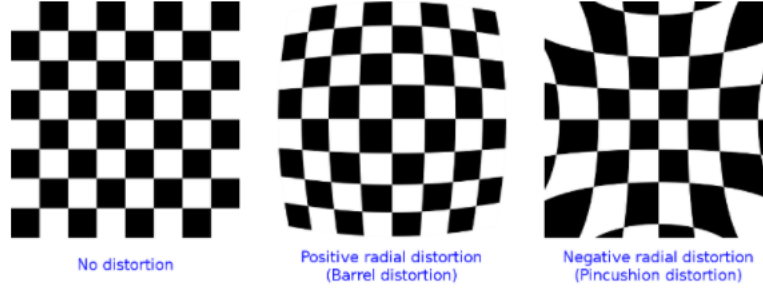

Figure 2: Radial Distortion

Tangential distortion occurs when the lens and the image plane are not parallel, shown in Figure 3.
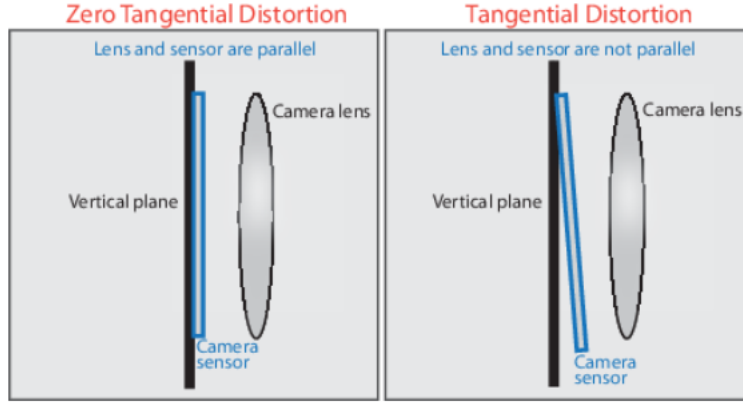

Figure 3: Tangential Distortion

When the distortions are considered, the camera resectioning model will be extend as:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} R & T \end{bmatrix} \times \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \tag{10}$$

$$x = \frac{X_c}{Z_c}, \quad y = \frac{Y_c}{Z_c} \tag{11}$$

$$\begin{aligned} x_d &= x \times \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 xy + p_2(r^2 + 2x^2) \\ y_d &= y \times \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1(r^2 + 2y^2) + 2p_2 xy \\ r^2 &= x^2 + y^2 \end{aligned} \tag{12}$$

$$u = f_x \times x_d + c_x, \quad v = f_y \times y_d + c_y \tag{13}$$

Find the Origin Point Considering the distortion model with 2 distortion coefficients, the transform equation with distortions is as follows. Point (x,y) represents the ideal point without distortion.

$$\check{x} = x + x(k_1(x^2 + y^2) + k_2(x^2 + y^2)^2)$$
$$\check{y} = y + y(k_1(x^2 + y^2) + k_2(x^2 + y^2)^2)$$
(14)

And we have

$$\check{x} = \frac{u - u_0}{f}$$
$$\check{y} = \frac{u - u_0}{f}$$
(15)

Then we make a square of each equation in (14), and add them together, we got

$$\check{x}^2 + \check{y}^2 = (x^2 + y^2) + 2k_1(x^2 + y^2)^2 + (2k_2 + k_1^2)(x^2 + y^2)^3 + 2k_1 k_2(x^2 + y^2)^4 + k_2^2(x^2 + y^2)^5 \quad (16)$$

This is a liner equation of $(x^2 + y^2)$, we can solve it by gradient descent. After we get the value of $(x^2 + y^2)$, we get the origin point (x, y) by bring $(x^2 + y^2)$ into equation (14).

1.5   Calibration

Base on what I know about Camera Calibration, I reckon that Camera Calibration aims to get an exact approximation of the intrinsic parameters including the distortion coefficients which only depends on the camera itself, and extrinsic parameters which represent the camera's direction and location in 3D space.

References

[1] **Camera resectioning** Wikipedia https://en.wikipedia.org/wiki/Camera_resectioning

[2] **What Is Camera Calibration?** MathWorks https://cn.mathworks.com/help/vision/ug/camera-calibration.html?s_tid=gn_loc_drop

[3] **Rotation matrix** Wikipedia https://en.wikipedia.org/wiki/Rotation_matrix

[4] **Scale factor** Wikipedia https://en.wikipedia.org/wiki/Scale_factor

[5] **Camera Matrix** Wikipedia https://en.wikipedia.org/wiki/Camera_matrix

[6] **Pinhole Camera Model** Wikipedia https://en.wikipedia.org/wiki/Pinhole_camera_model

[7] **Fundamentals of Computer Vision** Magill University http://www.cim.mcgill.ca/%7Elanger/558/4-cameramodel.pdf

[8] 关于 **OpenCV** 的那些事——相机标定 CSDN https://blog.csdn.net/aptx704610875/article/details/48914043

[9] **Cv** 照相机标定和三维重建 Opencv http://wiki.opencv.org.cn/index.php/Cv%E7%85%A7%E7%9B%B8%E6%9C%BA%E5

[10] **Camera Calibration and 3D Reconstruction** Opencv https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calib