University of Burgundy

Masters in Computer Vision and Robotics

VISUAL PERCEPTION

Calibration and Triangulation Lab

by

Vamshi Kodipaka

Supervisor: Dr. David Fofi

UNIVERSITÉ DE BOURGOGNE

INTRODUCTION

Calibration: Calibration is the way toward designing an instrument to give an outcome for a sample within an adequate range.

Geometric camera calibration, likewise alluded to as *camera* re-sectioning, gauges the parameters of a lens and image sensor of an image or camcorder. You can utilize these parameters to address for lens distortion, measure the size of an object in world coordinates, or estimate the location of the camera in the scene. These tasks are utilized in applications for example, machine vision to detect and measure objects. They are likewise utilized in robotics, for navigation systems, and 3-D scene reconstruction.

Camera parameters incorporate intrinsics, extrinsics, and distortion coefficients. To appraise the camera parameters, you need 3-D world points and their corresponding 2-D image points. You can obtain these correspondences using multiple images of a calibration pattern, for example a checkerboard. Using the correspondences, you can solve for the camera parameters. After you calibrate a camera, to find the accuracy of the estimated parameters, you can:

- Plot the relative locations of the camera and the calibration pattern
- Calculate the re-projection errors.
- Calculate the parameter estimation errors.

In Computer Vision Toolbox calibration algorithm works on the camera model proposed by Jean-Yves Bouguet. The model incorporates:

- The pinhole camera model.
- Lens distortion.

The pinhole camera model does not account for lens distortion because an ideal pinhole camera does not have a lens. To precisely represent a real camera, the full camera model used by the algorithm incorporates the radial and tangential lens distortion.

Triangulation: In computer vision, triangulation refers to the process of estimating a point in 3D space given its projections onto two, or more, images. In order to solve this problem it is essential to know the parameters of the camera projection function from 3D to 2D for the cameras involved, in the simplest case those can be represented by the camera matrices. It sometimes also referred to as reconstruction.

The triangulation problem is in theory trivial. Since each point in an image corresponds to a line in 3D space, all points on the line in 3D are projected to the point in the image. If a pair of corresponding points in two, or more images, can be found it must be the case that they are the projection of a common 3D point x. The set of lines generated by the image points must intersect at x (3D point) and the algebraic formulation of the coordinates of x (3D point) can be computed in a variety of ways, as is presented below.

In practice, however, the coordinates of image points cannot be estimated with arbitrary accuracy. Rather, various types of noise, such as geometric noise from lens distortion or interest point detection error, lead to errors in the measured image coordinates. As a result, the lines generated by the corresponding image points do not always intersect in 3D space. The issue, then, is to find a 3D point which ideally fits the measured image points.

1. Pin Hole Camera Model

A pinhole camera is a basic camera without a lens and with a single small aperture. Light rays pass through the aperture and project an inverted image on the contrary side of the camera. Think of the virtual image plane as being placed in front of the camera and containing the upright image of the scene. It is the most frequently used model. The model is inspired by the simplest cameras as shown in Fig. 1.

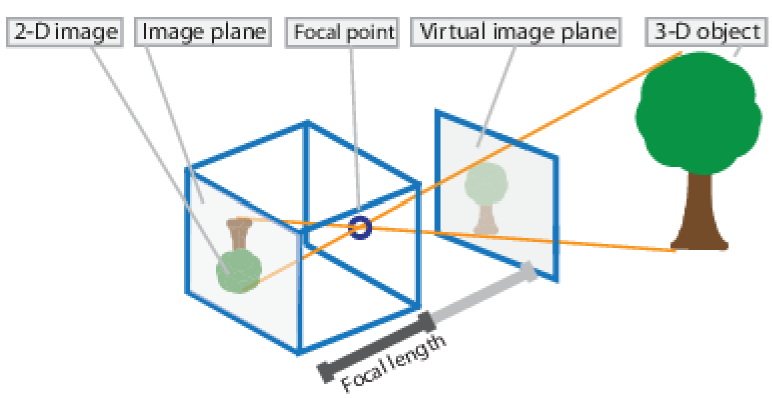


Fig.1: Pin Hole Camera Model

The pinhole camera parameters are represented in a 4-by-3 matrix called the *camera matrix*. This matrix maps the 3-D world scene into the image plane. The calibration algorithm computes the camera matrix making use of the extrinsic and intrinsic parameters. The extrinsic parameters features the location of the camera in the 3-D scene. The intrinsic parameters represent the optical center and focal length of the camera.

Note that in contrast to a real pinhole camera we have put the image plane in front of the camera center. This has the impact that the image won’t appear upside down as in the real model.

$$\underbrace{w}_{\text{Scale factor}}\ [\underbrace{x\ y}_{\text{Image points}}\ 1] = [\underbrace{X\ Y\ Z}_{\text{World points}}\ 1]\ P$$

$$\underbrace{P}_{\text{Camera matrix}} = \underbrace{\begin{bmatrix} R \\ t \end{bmatrix}}_{\substack{\text{Extrinsics} \\ \text{Rotation and translation}}} \underbrace{K}_{\text{Intrinsic matrix}}$$

The world points are transformed to camera coordinates using the extrinsics parameters. The camera coordinates are mapped into the image plane making use of the intrinsics parameters.

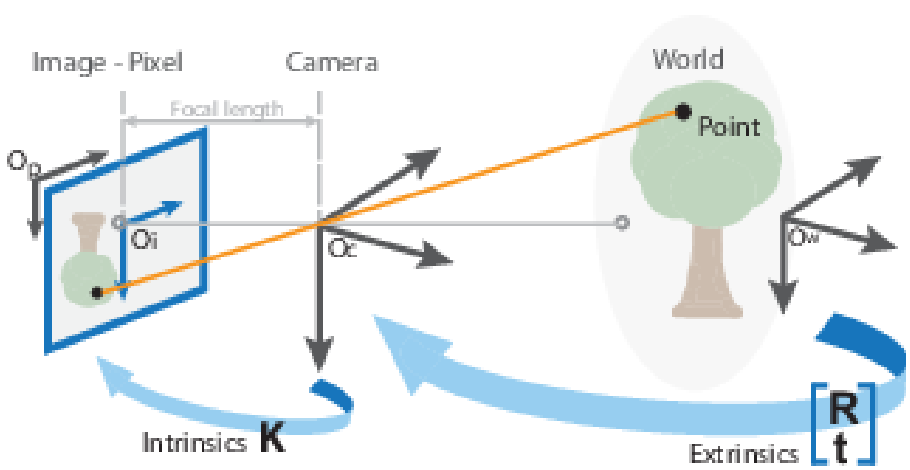
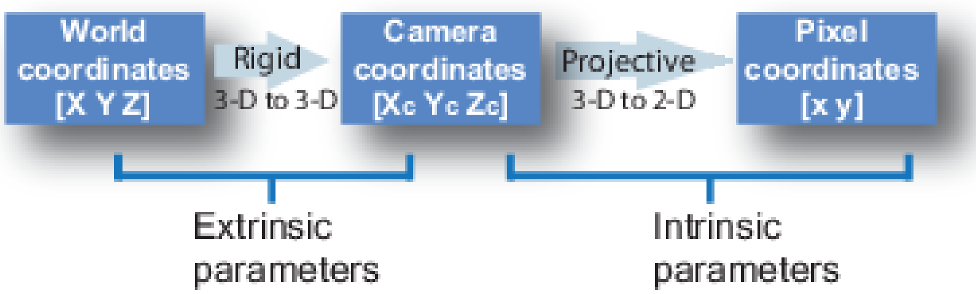


Fig.2: Mapping 3D to 2D

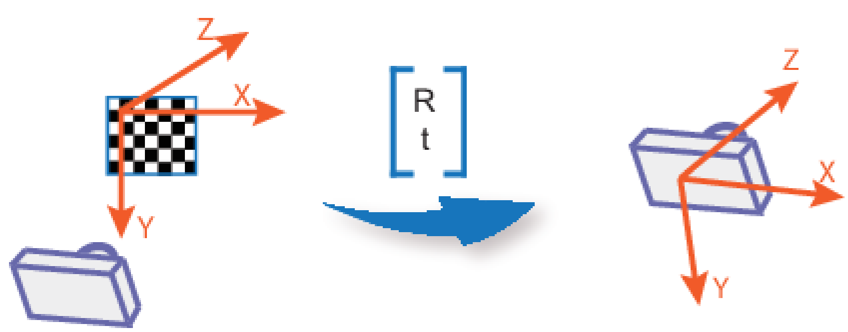
Camera Calibration Parameters

The calibration algorithm determines the camera matrix using the extrinsics and intrinsics. The extrinsic parameters represent a rigid transformation from 3-D world coordinate frame to the 3-D camera's coordinate frame. The intrinsics represent a projective transformation from the 3-D camera's points into the 2-D image points.



Extrinsic Parameters

The extrinsic parameters possesses of a rotation, *R*, and a translation, *t*. The origin of the camera's coordinate system is at its optical center and its *x*- and *y*-axis characterize the image plane.

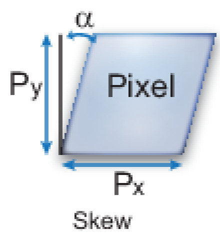


Intrinsic Parameters

The intrinsic parameters possesses of the focal length, the optical center, also known as the *principal point*, and the skew coefficient. The camera intrinsic matrix, *K*, is defined as:

$$K = \begin{bmatrix} fx & 0 & 0 & 0 \\ s & fy & 0 & 0 \\ Cx & Cy & 1 & 0 \end{bmatrix}$$

The pixel skew is defined as:



$[c_x \quad c_y]$ — Optical center (the principal point), in pixels.
$(f_x, \quad f_y)$ — Focal length in pixels. $f_x = F / p_x$ $f_y = F / p_y$ F — Focal length in world units, typically expressed in millimeters. $(p_x, \quad p_y)$ — Size of the pixel in world units.
s — Skew coefficient, which is non-zero if the image axes are not perpendicular. $s = f_x \tan \alpha$

1.1 Our Experimentation-Mappings

Here, we have 3 mappings from World coordinate system to Image plane. The four coordinate systems are as shown in Fig.2.

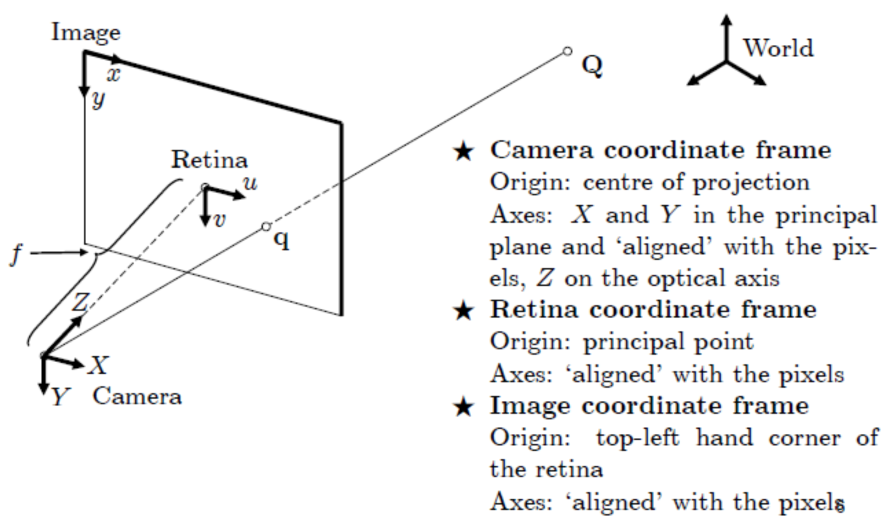


Figure 3: Coordinate frames in the model [1]

1.1.1 World to Camera coordinate system

This is a 3D displacement (Euclidean transformation).

$$Q_c = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} Q$$

where, **R** is a Rotation matrix (3 x 3) with $R^T R = 1$ and det(R) = 1

t is a translational vector (3 x 1) "**translation of origin of world coordinate frame w.r.to camera coordinate frame**"

Q is world homogenous coordinate of 3D point

Q_c is camera coordinate of 3D point

These are extrinsic parameters.

1.1.2 Camera to Retina coordinate system

This is a projection.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

where, f is the focal length, i.e., the distance between center of projection to image plane.

1.1.3 Retinal to Image coordinate system

This is a 2D mapping

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & 0 & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

where, k_x , k_y are density of pixels (pixels/mm)

u_0, v_0 is the translation from retinal to image coordinate system.

The complete mappings looks like:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} fk_x & 0 & x_0 \\ 0 & fk_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2. Experimentation

<u>NOTE : A well commented code is attached with this report, run main.m file to see the output od camera1 and camera2 and all other supporting functions areincluded in this lab submission.</u>

In the lab tutorial, we had taken a 3D world scene as a point cloud. The focal length, we choose is 50 and we assume that my pixel density is same along X and y directions which is 1 pixel/mm and also my retinal and image plane are at same reference.

The camera 1 is at origin of world coordinate system and camera 1 is translated in X-direction by some value. So, here I define two camera matrices P_1 and P_2 for two cameras as :

We take 3D Point:

P3D

8x3 double

1	2	3
20	10	800
100	10	800
100	50	800
20	50	800
20	10	900
100	10	900
100	50	900
20	50	900

There are the coordinates of the 3D scene of the cuboid, Assume it is P3D.

$$P_1 = P3D.\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

$$\text{where, } \boldsymbol{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \boldsymbol{t} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}'$$

and

$$P_2 = P3D.\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

where, $\boldsymbol{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $\boldsymbol{t} = \begin{bmatrix} -t_{xx} & 0 & 0 \end{bmatrix}'$

"here negative sign (-) because its translation of origin of world coordinate frame with respect to camera2"

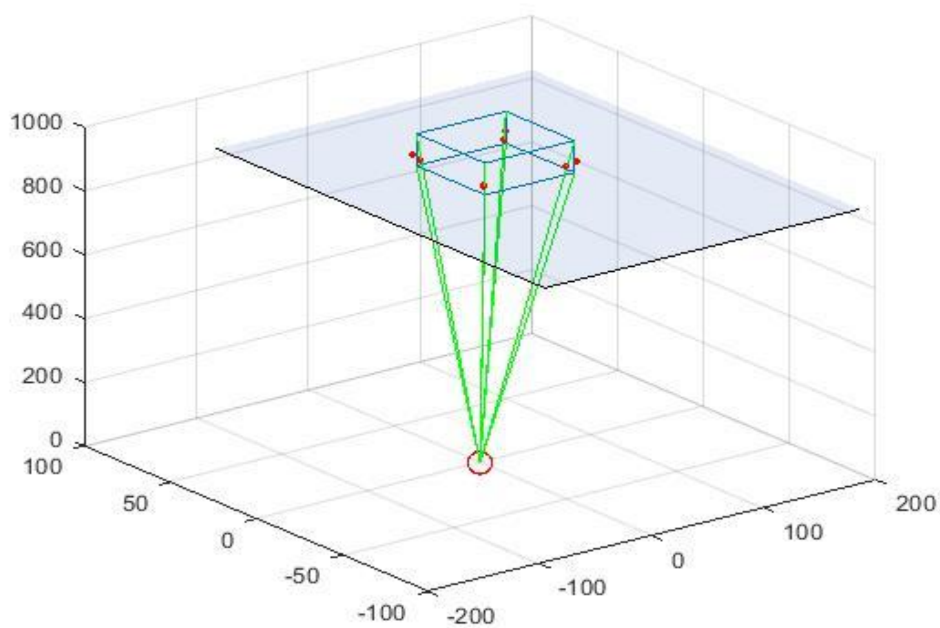
Task: Projection of 3D points onto camera planes

We project the 3D points onto the camera planes by using the equations:

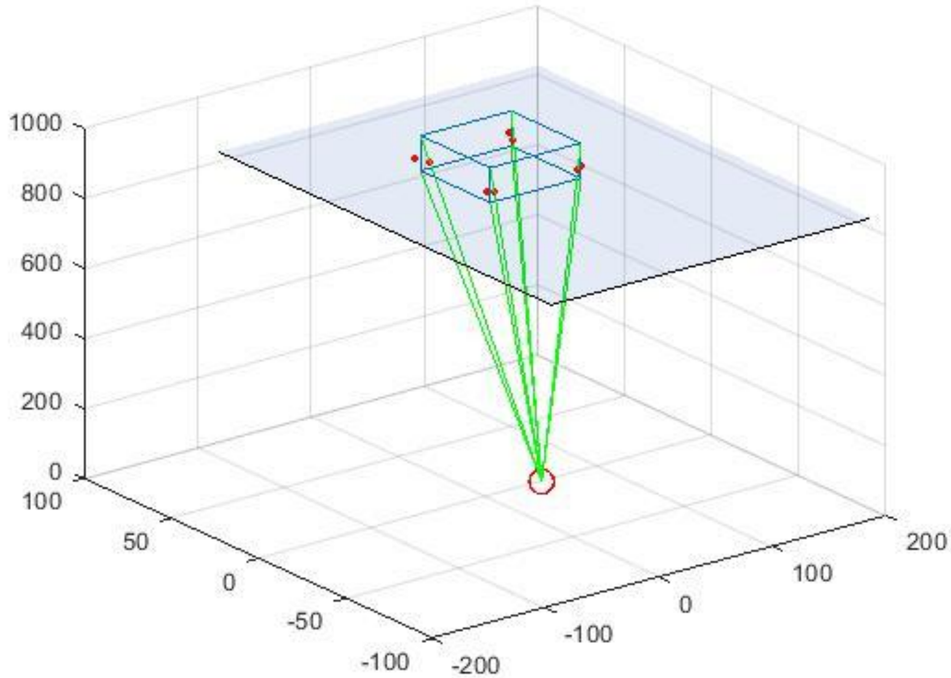
$$x = PX$$

where, X is 3D point, P is camera projection matrix and x is 2D point.

The results obtained are as shown in Fig.3.



The results of projection of a 3D object onto image plane w.r.t Camera1 (Fig.4 above).



The results of projection of same 3D object onto image plane w.r.t Camera2 (Fig.5 above).

P_tri

8x3 double

	1	2	3
1	20.0000	10.0000	800.0000
2	100.0000	10.0000	800.0000
3	100.0000	50.0000	800.0000
4	20.0000	50.0000	800.0000
5	20.0000	10.0000	900.0000
6	100.0000	10.0000	900.0000
7	100.0000	50.0000	900.0000
8	20.0000	50.0000	900.0000

Fig.6: Triangulation result

error

3x8 double

	1	2	3	4	5	6	7	8
1	3.2338e+11	4.3838e+11	4.3838e+11	3.2338e+11	4.1009e+11	5.3855e+11	5.3855e+11	4.1009e+11
2	2.0400e+11	2.0400e+11	2.4851e+11	2.4851e+11	2.5677e+11	2.5677e+11	3.0645e+11	3.0645e+11
3	1.2768e+06	1.2768e+06	1.2768e+06	1.2768e+06	1.6164e+06	1.6164e+06	1.6164e+06	1.6164e+06

Fig.7: Triangulation Error

3. Conclusion

This lab provides deep insights into Calibration and Triangulation. We observe the real 3D points and the simulated 3D points after triangulation are the same.

References

[1] Lecture slides of Dr. David Fofi

[2] Triangulation and Calibration text/script source, Wikipedia

[3] Multi View Geometry in Computer Vision by Richard Hartley and Andrew Zisserman

[4] Dr. Salvi Toolbox, Univeristy of Girona, Spain(Future Reference)

[5] Mathworks: https://www.mathworks.com/help/vision/ug/camera-calibration.html