A Report for Assignment -3 of COL780
Computer Vision

On

Camera Calibration: Finding intrinsic and extrinsic parameters and generating augmented reality pictures

Submitted to

Anurag Mittal
(Instructor COL 780)

by

Nitesh Dohre (2022MCS2070)

COL780 Assignment3

Department of Computer Science and Engineering

Indian Institute of Technology, Delhi

Session: 2022-23

CONTENTS:

0. PROBLEM STATEMENT

1. Calibrate the camera on your smartphone (intrinsic parameters) using the system using a checkerboard pattern as described in class.

2. Placing the checkerboard on a table, insert some artificial objects on the table (such as a simple pyramid etc.) and generate some mixed/augmented-reality pictures. This need not be real-time but can be done on pictures taken from the smartphone.

1.INTRODUCTION

Camera calibration is the process of determining the intrinsic and extrinsic parameters of a camera that are required to transform 2D image points into 3D world coordinates. The intrinsic parameters of a camera refer to its internal characteristics such as focal length, principal point, and lens distortion coefficients, whereas extrinsic parameters define the camera's position and orientation in the world coordinate system.

$$K = \begin{bmatrix} Fx & 0 & ux \\ 0 & Fy & uy \\ 0 & 0 & 1 \end{bmatrix}$$

Where K = intrinsic parameter matrix of camera

Fx= focal length of lens in x direction

Fy = focal length of lens in y direction

ux = distance of camera center from image plane center in x direction

uy = distance of camera center from image plane center in y direction

$$M_{Ext} = [R,t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & tz \end{bmatrix}$$

Here R= Rotation matrix of 3X3 orthonormal matrix

|R|=1

t= translation matrix of 3*1

Projection matrix: A projection matrix is a 3x4 matrix that defines the mapping between 3D world coordinates and their corresponding 2D image coordinates. The

projection matrix is used to project 3D points onto a 2D image plane by multiplying the homogeneous coordinates of the 3D point with the projection matrix.

$$x = PX$$

x= image coordinates (2D points)

X = World Coordinates(3D points)

The projection matrix can be decomposed into two components: the intrinsic matrix and the extrinsic matrix.

$$P = K [R\ t]$$

The assignment aimed to calibrate the camera of a smartphone and create mixed/augmented-reality pictures using the camera's intrinsic parameters. Camera calibration is essential in computer vision and is a crucial step to accurately measure distances, perform 3D reconstructions, and create mixed/augmented-reality content. In this report, we will present the process of calibrating the smartphone camera using a checkerboard pattern and inserting artificial objects on a table to generate mixed/augmented-reality pictures. I have described the methodology used for camera calibration and the process of creating mixed/augmented-reality images. Finally, we will present the results of the experiment and discuss the potential applications of mixed/augmented-reality in various domains such as entertainment, education, and gaming.

2. TOOLS AND TECHNOLOGY USED

1.Jupyter notebook IDE

2.Python3

3.Libraries: numpy, matplotlib, cv2, math, os, glob

3.CALIBRATE THE CAMERA OF THE SMARTPHONE USING A CHECKERBOARD PATTERN

1. Calibrating camera and intrinsic parameter of the camera.

The image absolute conic (IAC) is a mathematical concept in projective geometry that is related to the intrinsic parameters of a camera. The IAC is a 3x3 matrix that can be computed from a set of 2D image points and their corresponding 3D world coordinates.

The image absolute conic (IAC) can also be computed using vanishing points in an image. Vanishing points are the points at which parallel lines in 3D space appear to converge when projected onto a 2D image plane.

To compute the IAC using vanishing points, we first need to identify at least three vanishing points in the image. This can be done by identifying pairs of parallel lines in the image and finding the point at which they intersect.

The relation between the image of the absolute conic,ω and K is given as

$$W^{-1} = KK^{T}$$

K= intrinsic parameter matrix

Perpendicularity Relations between Vanishing Points

The angle between two rays with directions d1, d2 and corresponding images as x1, x2 is given by the following Equation.

$$cos\ \theta = \frac{\mathbf{x_1^\top} \omega \mathbf{x_2}}{\sqrt{\mathbf{x_1^\top} \omega \mathbf{x_1}} \sqrt{\mathbf{x_2^\top} \omega \mathbf{x_2}}}$$

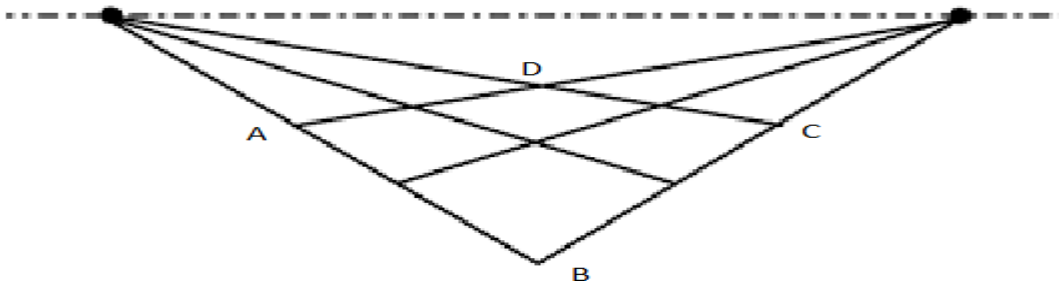
where ω is the image of absolute conic. If we assume v1, v2 are the vanishing points corresponding to orthogonal directions, then we have

$$V1^{T} wV1 = 0.$$

2. Vanishing points and lines

A **vanishing point** is the point at which parallel lines in 3D space appear to converge when projected onto a 2D image plane. In other words, if we draw a set of parallel lines in a 3D scene and project them onto a 2D image, they will appear to converge at a single point. This point is called the vanishing point.

A **vanishing line** is a line in the image that corresponds to a set of parallel lines in 3D space. Like vanishing points, vanishing lines are useful in computer vision because they provide information about the 3D structure of a scene. By identifying the vanishing lines in an image, we can estimate the directions of the corresponding parallel lines in 3D space and use this information to infer the 3D structure of the scene.



Here in above picture AB and CD are parallel lines but they seem to be converge,
Lets suppose at points v1 they intersect each other,
Similarly, AD and BC two parallel lines, intersect each other at point v2,

v1 and v2 are the points at infinity also known as vanishing points, and if we join them, we'll get a line at infinity also known as vanishing line.

3. Algorithm :

1. Start
2. Find 5 or more square box in of different images checkerboard pattern
3. Find lines by doing cross product of corner points of square box
4. Find the intersection points of the lines by doing cross product of this parallel lines which seems to converge
5. Find pairs of vanishing points for all the images
6. For the set of homogeneous equation using this vanishing points
7. Apply SVD(singular value decomposition) on the matrix formed using the set of homogeneous equation and find w(image of absolute conic)
8. From image of IAC we can find intrinsic parameter matrix K

   Using this relation $W^{-1} = KK^{T}$
9. Print K
10. Stop

4. Implementation:

For Finding vanishing points we have taken corner points in the checkerboard square grid.

```
COORDS_14 = [[(247,543),(512,552),(171,710),(579,725)],\
             [(318,575),(589,604),(177,705),(590,766)],\
             [(164,215),(449,335),(176,676),(485,659)],\
             [(294,391),(565,310),(223,680),(549,672)],\
             [(211,323),(541,327),(157,639),(603,639)],\
             [(385,516),(653,565),(210,623),(580,724)]]
```

COORD_14 contains the coordinate points of rectangle grid(seem to be quadrilateral)

To get

l1 =cross_product(COORD_14[i][0],COORD_14[i][1])

l2 =cross_product(COORD_14[i][3],COORD_14[i][3])

v1 = cross_product(cross_product(l1,l2)

Similarly find v2 and so on.

5. Input Images :

All the input frames can be find from here [COL780 Assignment3]

6. Results:

```
For Dataset2
Intrinsic Parameter of Camera k:
[[786.806454 -0.000000 371.690298]
 [0.000000 786.806454 481.756913]
 [0.000000 0.000000 1.000000]]

 Image of Absolute Conic w:

[[0.000001 0.000000 -0.000376]
 [0.000000 0.000001 -0.000487]
 [-0.000376 -0.000487 1.000000]]
```

4.FINDING EXTRINSIC PARAMETERS OF THE CAMERA :

1. Extrinsic Parameter of the camera (Translation and Rotation) :

The **rotation matrix** specifies the orientation of the camera in the reference frame. It is typically represented as a 3x3 matrix, denoted by R, which maps a point from the world coordinate system to the camera coordinate system. The rotation matrix is orthogonal, meaning that its columns form an orthonormal basis for the camera coordinate system.

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$

The **translation vector** specifies the position of the camera center in the reference frame. It is typically represented as a 3D vector, denoted by t, which gives the displacement of the camera center from the origin of the reference frame.

2. Process of finding extrinsic parameter matrix

Till now we have computer, Intrinsic parameter matrix of camera,

x=PX

Where
x is 2D image points coordinate,
X is 3D world point coordinates
P is Projection matrix of dimension 3X4

P = $M_{intrinsic}$ * $M_{Extrinsic}$ = K[R t]

x = K [R t]

$K^{-1}x = K^{-1}K[R\ t]X$

b = A X

Where A=[R t]

b = $K^{-1}x$

Now,

We can find, 6 or more pair of World to image coordinate pair to solve the AX=b
By least square method,

3. Algorithm :

Inputs:

Intrinsic parameter matrix K (3x3)
List of 10 world-to-image coordinate correspondences, [(X1, x1), (X2, x2), ..., (X10, x10)],
where Xi is a 3D point in the world coordinate system and xi is its corresponding 2D image point in the camera coordinate system.

Outputs:

Extrinsic parameter matrix [R | t] (3x4)
Algorithm:

Define a function reprojection_error that takes the intrinsic parameter matrix K, extrinsic parameter matrix [R | t], and a world-to-image correspondence (X, x) as input and returns the reprojection error e as follows:
a. Convert the 3D world point X to homogeneous coordinates, Xh = [X; 1].
b. Compute the predicted image point x' = K [R | t] Xh.
c. Normalize x' by dividing by its third coordinate, x' = x' / x'[2].

d. Compute the reprojection error e as the Euclidean distance between the observed image point x and the predicted image point x', e = ||x - x'||.

Define a function objective_function that takes the intrinsic parameter matrix K, world-to-image correspondences, and the flattened vector of extrinsic parameters p=[r11, r12, r13, r21, ..., tz] as input and returns the sum of squared reprojection errors as the objective function value. Inside the function, do the following:

1. Reshape the flattened vector p into the 3x4 extrinsic parameter matrix [R | t].
2. Compute the sum of squared reprojection errors over all world-to-image correspondences by calling the reprojection_error function for each correspondence and summing up the squared errors.

3.Initialize the flattened vector p0 to zeros, with a length of 12 (r11, r12, r13, r21, ..., tz).

Use a nonlinear optimization method such as the Levenberg-Marquardt algorithm to minimize the objective_function with respect to the flattened parameter vector p, subject to the constraint that the first 9 elements of p are reshaped to a 3x3 rotation matrix R using the Rodrigues formula. The optimization algorithm should return the optimal value of p, denoted as p*.

4.Reshape the optimal flattened parameter vector p* into the 3x4 extrinsic parameter matrix [R | t].

5.Return the extrinsic parameter matrix [R | t].

Output:

```
Extrinsic Parameter Matrix for DataSet2
[[0.033908 0.000244 -0.013660 -0.000519]
 [0.000038 -0.033648 0.015239 0.000579]
 [0.000000 -0.000000 0.037968 0.001444]]


 Projection Matrix For Dataset 2 :

 [[26.678954 0.192308 3.364866 0.127942]
  [0.030161 -26.474359 30.281409 1.151384]
  [0.000000 -0.000000 0.037968 0.001444]]
```

5. PLACING THE CHECKERBOARD ON THE TABLE AND INSERT AN ARTIFICIAL OBJECT

In this step, a checkerboard pattern is placed on a table or surface, which serves as a known reference point for the camera or sensor. The artificial object refers to a virtual 3D object that is inserted into the scene and aligned with the checkerboard.

The purpose of this step is to calibrate the camera or sensor and establish the relationship between the 3D world and the 2D image or video captured by the device. This calibration allows for accurate tracking and positioning of the artificial object in the real world.

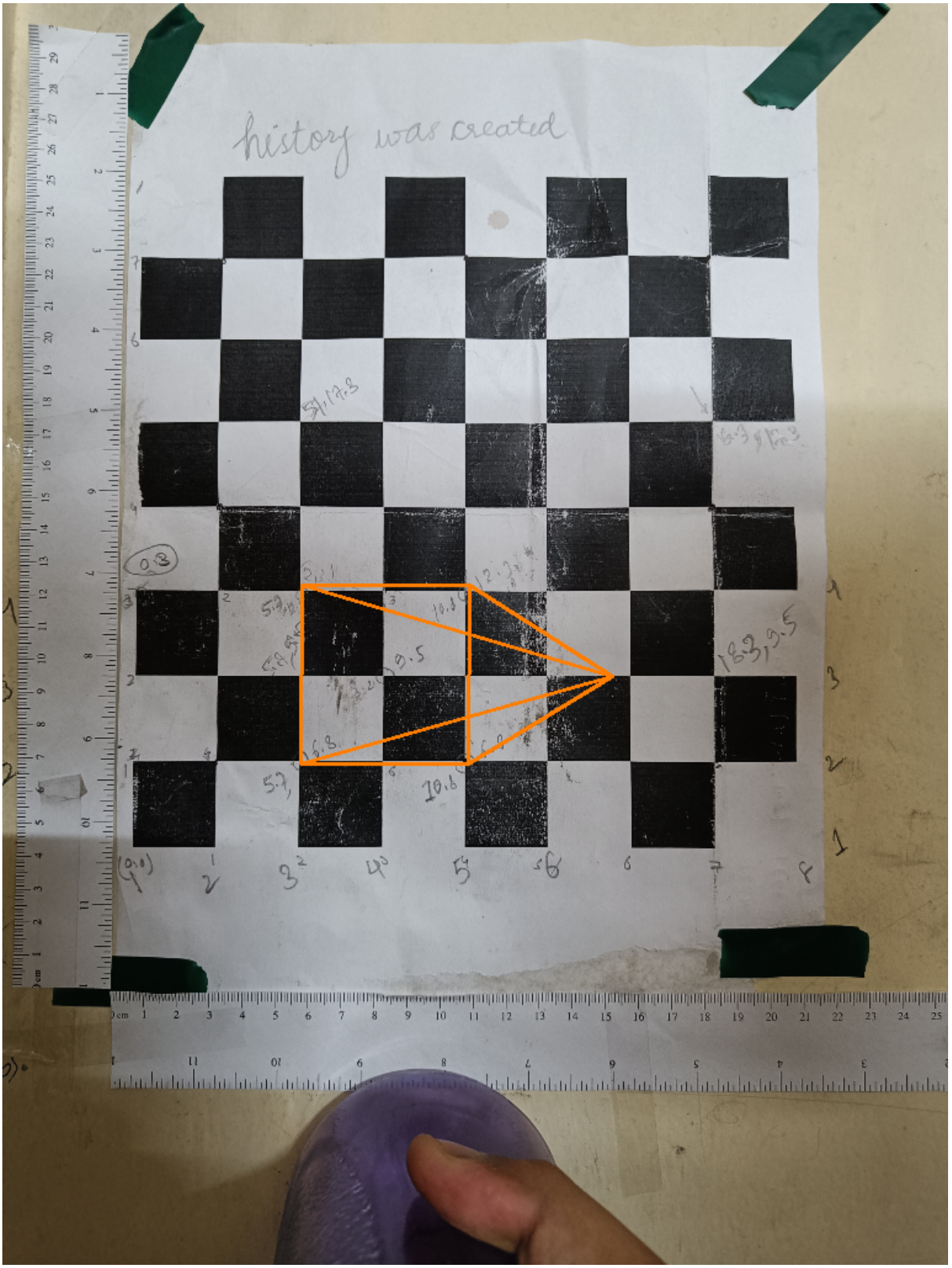
Algorithm:

1.Load the 3D object you want to project in the scene. You can use a 3D modeling software to create or import the object.
2. Define the camera parameters, such as the intrinsic and extrinsic matrices. These parameters determine how the camera sees the scene and the object.
3. For each vertex of the object, apply the camera transformation matrix to obtain the
4. corresponding image point. This can be done using matrix multiplication.
5. Normalize the image point by dividing its coordinates by its z-coordinate. This is necessary to obtain the correct projection in 2D.
6. Convert the normalized image point to pixel coordinates by rounding its x and y coordinates and casting them to integers.
7. Draw the projected object on the scene image using OpenCV's line drawing function. You can connect the projected vertices using lines or draw polygons.
8. Show and save the output image

World coordinates of Object(Prism):

```
# world coordinate point of a Prism
a = np.array([5.7,12.3,26.3,1])
b = np.array([10.8,12.3,26.3,1])
c = np.array([10.8,6.8,26.3,1])
d = np.array([5.7,6.8,26.3,1])
e = np.array([15.2,9.5,26.3,1])
```

RESULTS:



6. APPLICATIONS AND CONCLUSION

APPLICATIONS:

The calibration of a smartphone camera is essential for several applications such as augmented reality, image processing, object detection, and tracking. Once the intrinsic parameters of the camera are determined, we can obtain accurate measurements of real-world objects from images captured by the camera. This calibration can be used in many applications, such as:

1.Augmented Reality: Augmented reality (AR) overlays virtual images on the real world. Using the calibrated camera parameters, we can create more realistic AR experiences by accurately positioning virtual objects in the real world.

2.Object Detection and Tracking: Object detection and tracking algorithms rely heavily on the accuracy of the camera parameters. Calibrating the camera ensures that the position and orientation of detected objects are accurate and consistent.

3. Image Processing: Calibrating the camera can improve the accuracy of image processing algorithms such as edge detection, image segmentation, and feature extraction.

4. 3D Scanning: Calibrating the camera is essential for 3D scanning applications that use photogrammetry techniques to reconstruct 3D models from images.

CONCLUSION:

In conclusion, this assignment demonstrated the process of calibrating a smartphone camera using a checkerboard pattern. Once the intrinsic parameters of the camera are determined, we can use them to generate mixed/augmented reality pictures, object detection and tracking, image processing, and 3D scanning. These applications have numerous real-world applications in areas such as entertainment, education, healthcare, and industrial automation. Therefore, calibration of the camera is an essential step to ensure the accuracy and reliability of these applications.

7. BIBLIOGRAPHY AND REFERENCES

[1].Multiple View Geometry in Computer Vision Second Edition
Richard Hartley and Andrew Zisserman

[2]. Computer Vision: Algorithms and Applications Book by Richard Szeliski

[3]. https://opencv.org/

[4]. https://docs.python.org/3/

Codes and Dataset can be found here : COL780 Assignment3