Pose Estimation using Camera

The second goal of the project is to estimate the rotation angles in the world frame using the camera. However, we need to calibrate the camera first.

## Camera Calibration Definition

Camera Calibration refers to the process of getting the internal and external parameters of a camera, where the internal parameters are:

* fx and fy, the focal length multiplied by the pixel density in the x and y axes, mx and my, respectively
* Ox and Oy, the image center points

And the external parameters are:

* Rotation matrix R, defining the orientation of the camera in the world frame
* Translation matrix T, defining the position of the camera in the world frame

If we have the camera parameters, then we can establish a transformation from 3D world coordinates to 2D image coordinates, and vice versa.

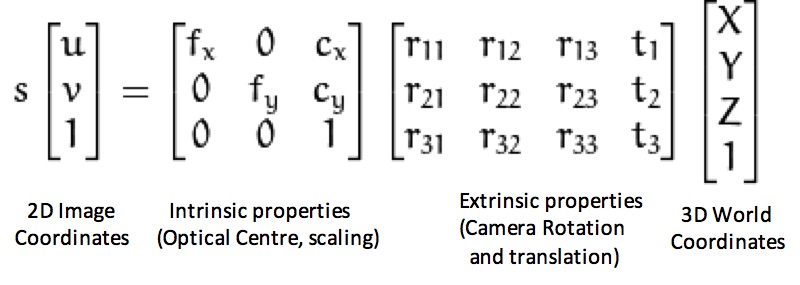


Figure 1. Transformation between 3D world coordinates to 2D camera coordinates

## Camera Calibration Process

To calibrate the camera, we need to do so using an object of known dimensions; for instance, a known calibration pattern such as a chessboard

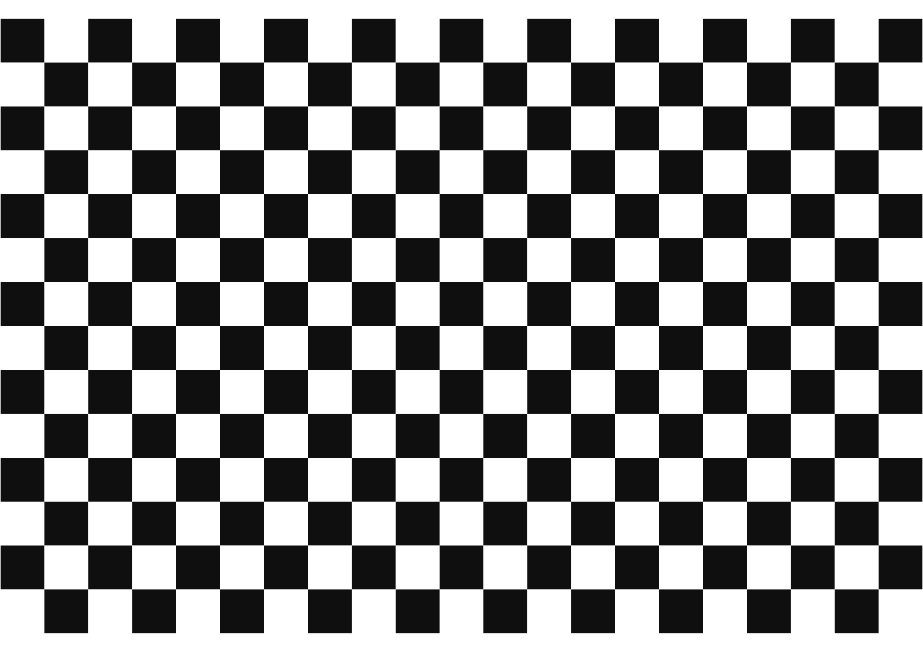


Figure 2. Chessboard pattern with known dimensions

Then the steps are as follows:

1. Take multiple images of the chessboard from different angles and positions. The pattern should be flat.
2. Detect the corners or dots in the calibration pattern in each image. For a chessboard, this involves finding the intersection points of the black and white squares.
3. Using the known geometry of the calibration pattern and the detected key points, an initial estimate of the intrinsic parameters can be made using the Direct Linear Transformation method (DLT).
4. For each calibration image, compute the extrinsic parameters (R and T) that relate the 3D coordinates of the calibration pattern to the 2D image coordinates.

Steps 3 and 4 could be done using the Direct Linear Transformation method (DLT), where the known 2D camera and 3D world coordinates are combined in the A matrix, and the unknown parameters are combined into the P matrix, then:

Since scaling P by a factor of K will produce the same 2D image coordinates as P, we can define , then the problem can be rewritten as:

The goal is to minimize Ap with , this could be achieved with the following loss function:

Taking the derivative w.r.t p,

The problem then becomes an eigenvalue/eigenvector problem

The matrix P will then be the eigenvector of the smallest eigenvalue of

1. Estimate the nonlinear distortion effects, such as the radial and tangential distortion coefficients, for better calibration accuracy.

## Estimating Orientation using Camera

After calibrating the web camera, i.e. getting the internal and external parameters, we can now design an application that estimates the orientation of the chessboard in a live video stream and plot the readings in real-time.

The program is divided into two parallel threads using the threading library in python, the first thread is used to capture the live video stream using opencv library

A person holding a chess board

Description automatically generated

Figure 3. live stream of the subject with chessboard

After that, the camera parameters are passed to the cv2.solvePnP method to get the coordinates of the board, basically the PnP method determines the coordinates of the board using the 3D coordinates of n points on the chessboard, then from the coordinates of the board we get the orientation

On the other thread, a function that plots the 3 angles of the orientation (roll, pitch, yaw) in real time is implemented.

A screenshot of a graph

Description automatically generated

Figure 4. Real time plot of angles

## Using Orientation readings for a rehabilitation game

The goal of obtaining the orientation readings is to develop an interactive rehabilitation game that encourages hip mobility.

The game was developed in python using the pygame library, where the patient wearing the chessboard around their chest controls the position of a shark in an underwater environment.

To score more points, the patient should bend in a combination of forward and sideway bending to be able to make the shark reach the fish to eat it. After the shark eats the fish another fish appears at a new random location.

A screenshot of a computer game

Description automatically generated

Figure 5. Interface of Sharky! game implemented in python

## Reflection on effectiveness of the camera

The web camera is effective at getting accurate orientation readings when the chessboard is close to the camera; however, the lighting conditions should be optimal; low lighting will not provide clear image details and will be noisy due to an increase in ISO, while at excessive lighting details are washed out.

Another limitation of the webcam is its low resolution; indeed, this can be evident when the subject moves away from the webcam; the readings become jerky even when the chessboard is kept still, also this behavior is reflected in the game where the shark’s movement becomes jerky. A solution to this problem is to use a high-resolution camera with a large depth-of-field (short focal length) so the subject stays in focus when they are further away and, at the same time, the image of the subject is represented by a large amount of pixels.

## References

1. First Principles of Computer Vision (2021) *Linear Camera Model | Camera calibration*. https://www.youtube.com/watch?v=qByYk6JggQU.
2. First Principles of Computer Vision (2021a) *Camera calibration | Camera calibration*. https://www.youtube.com/watch?v=GUbWsXU1mac.
3. First Principles of Computer Vision (2021b) *Intrinsic and extrinsic matrices | Camera calibration*. https://www.youtube.com/watch?v=2XM2Rb2pfyQ.
4. *OpenCV: Camera calibration* (no date). https://docs.opencv.org/4.x/dc/dbb/tutorial\_py\_calibration.html.