



Instituto Superior de Engenharia de Lisboa

ISEL
INSTITUTO SUPERIOR DE
ENGENHARIA DE LISBOA

Computer Vision and Mixed Reality

Marker Based Augmented Reality

Mestrado em Engenharia Informática de Multimédia

Pedro Gonçalves, 45890

June 2022

Introduction

During this project, a computer vision application that implements marker based augmented reality will be developed. It will allow the inclusion of virtual elements aligned with real fiducial markers. The application should work for any camera, meaning the project will have to include a calibration phase. **OpenCV** (Open-Source Computer Vision) library will be used, as well as **ArUco** marker-based library.

The first part of this project will be to compute the intrinsic parameters of the camera that's being used. The computed parameters should be saved to a **.yaml** file to be used later.

Then, with the camera parameters, make an application capable of detecting the **ArUco** markers and draw **3D** axes that have the exact same size as the markers, to make the final part of the project easier.

To finish, an application will be developed to render virtual objects (**2D**) in the previously detected **ArUco** markers.

Camera Calibration

This first part of the project will handle the necessary algorithms and logic to make the rest of the project compatible with any camera. It will compute the intrinsic parameters of the currently in-use camera. This process is called the **Perspective Transformation**. The script developed to implement this process is in the file **camera_calibration.py**.

When executed, this program will access the camera and the user should press 'c' to take a photo and 'q' to quit. When pressing 'c', the camera should be pointing at a chessboard of **WIDTH** squares by **HEIGHT** squares, and a square side of **SQUARE_SIZE** in centimetres. The application will save the computed camera intrinsic parameters to **calibration_chessboard.yaml** and two images: **distorted.jpg** – the image before calibration and **undistorted.jpg** – the image after calibration.

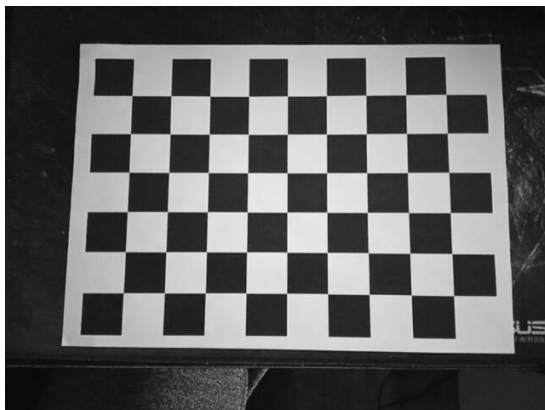


Figure 1 - Distorted

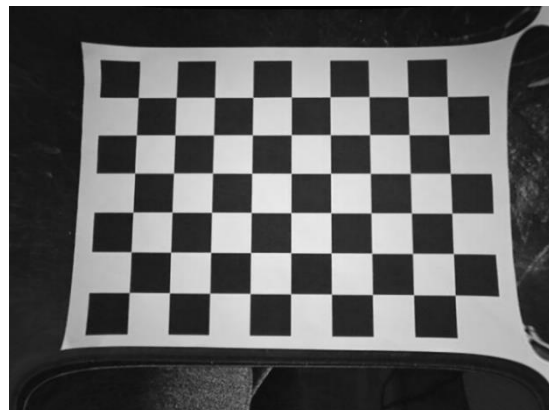


Figure 2 - Undistorted

Detection and Camera Pose Estimation with ArUco Library

First, it's important to understand what **ArUco** markers are. **ArUco** markers are binary square fiducial markers that can be used for camera pose estimation. Their main benefit is that their detection is robust, fast, and simple.

The **ArUco OpenCV** module includes the detection of these types of markers and the tools to employ them for pose estimation and camera calibration. For this project, **DICT_6X6_250** dictionary will be used.

Now, an application capable of detecting these markers will be developed. It should also be able to detect multiple markers, differentiate them (the different markers in the dictionary) and draw the **3D** axes. These axes should be the exact same length as the marker itself. An **A4** paper sheet with printed markers will be used, in which each marker is **5.5** centimetres long (side). That means the axes length should half of that value, since the axes origin will be placed right in the centre of the marker, as shown in the following figure.

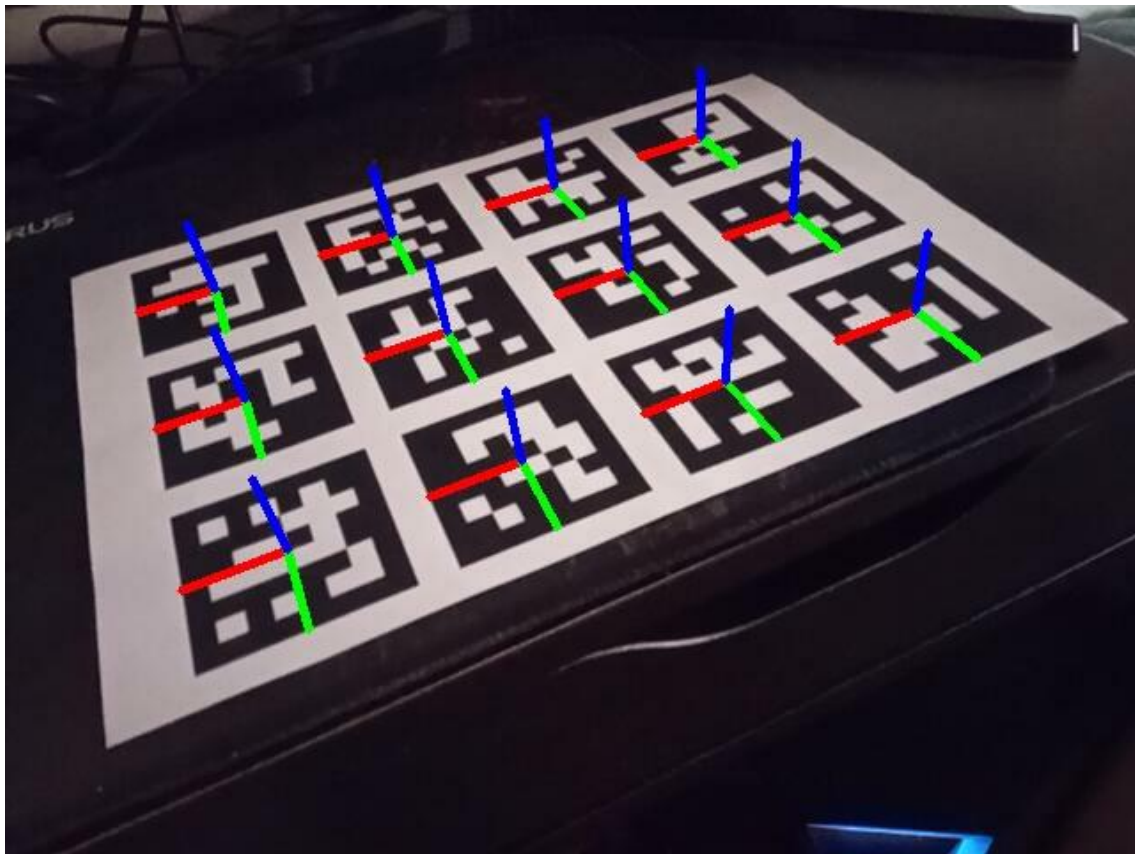


Figure 3 - ArUco markers with 3D axes

Registration of virtual objects

For this final part of the project, an application that is capable of rendering virtual objects in the ArUco markers position will be developed, taking advantage of the concepts explored in the previous two chapters. The application should be able to also render different objects for different markers.

For this project, album cover objects will be used. Here's a figure showing the result.



Figure 4 - Virtual objects

Conclusion

Once more, this kind of project allows us to explore computer vision concepts in a very practical way. Reviewing the steps that were taken, the developed applications are capable of calibrating any camera (according to a reference chessboard), identify ArUco markers and render virtual objects in their positions. The next step would be to render 3D virtual objects in the ArUco markers.