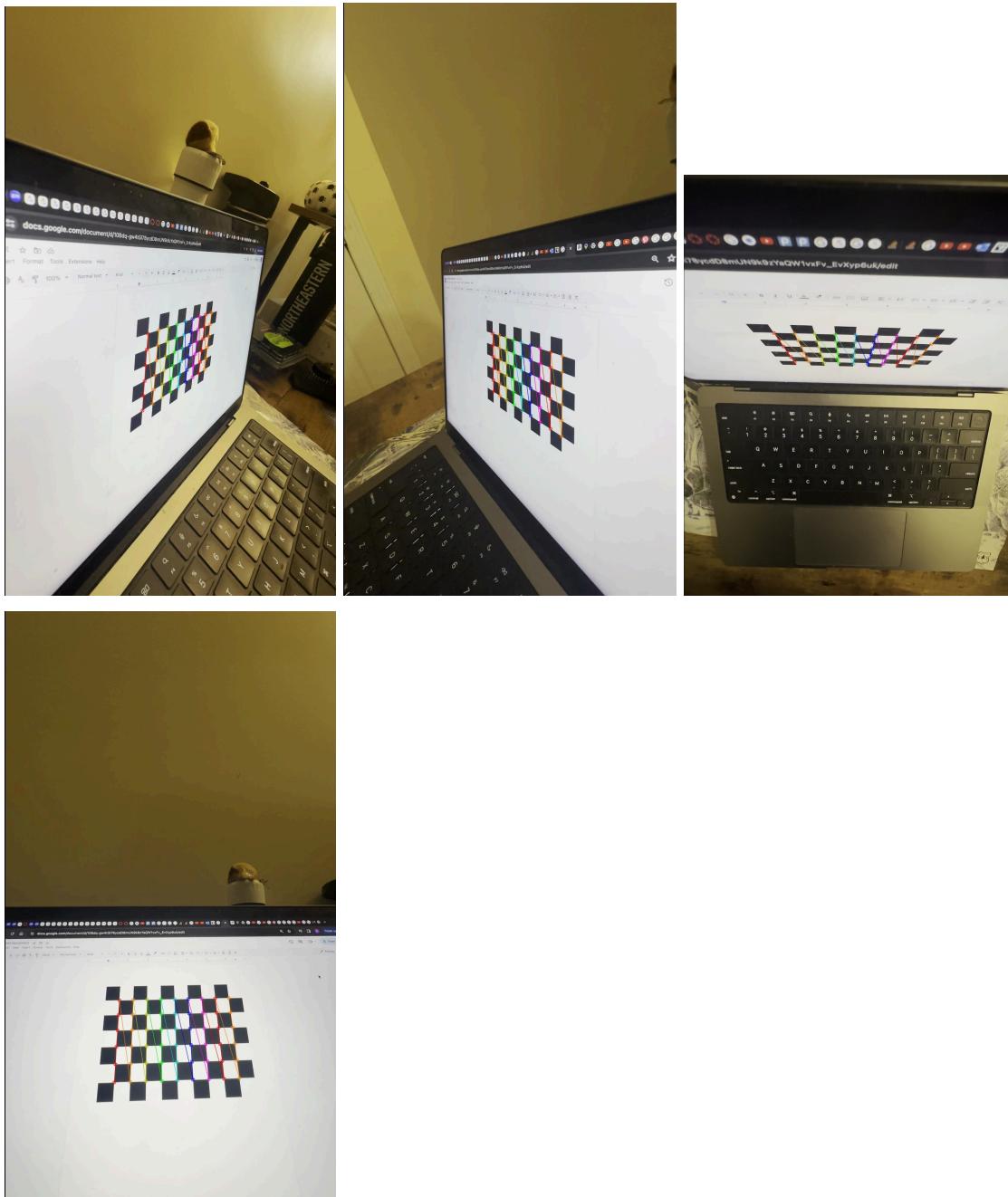


# Calibration and Augmented Reality

## 1. Detect and extract Corners:

For this project, I have used a checkerboard as the target and I have used camera calibration with open cv to process calibration.

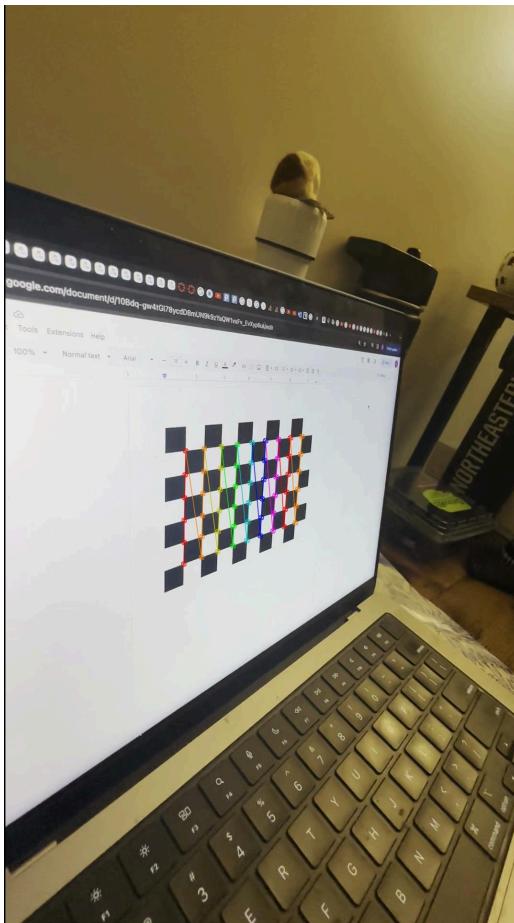
Below are the frames with corners detected.



One of the major issues using opencv checkerboard functions is that, the checkerboard is not outlined with white background, it will have difficulty in identifying the corners and will not yield good results,

whereas aruco markers are more effective in detecting the markers efficiently.

## 2. Select Calibration Images:



The above is one of the image used for calibration

## 3. Calibrate the Camera:

Below is the screenshot of the camera calibration values and the reprojection error

```
Finale camera_matrix values:  
[883.991101721898, 0, 541.2277601047368;  
 0, 883.9189577803977, 955.1766506247974;  
 0, 0, 1]  
Final distortionCoefficients values:  
[-0.004270181298894915;  
 -0.02757870989992522;  
 0.0001150047687392748;  
 0.001179681959826299;  
 0.03595080923106949]  
Final reprojection_error = 0.152481
```

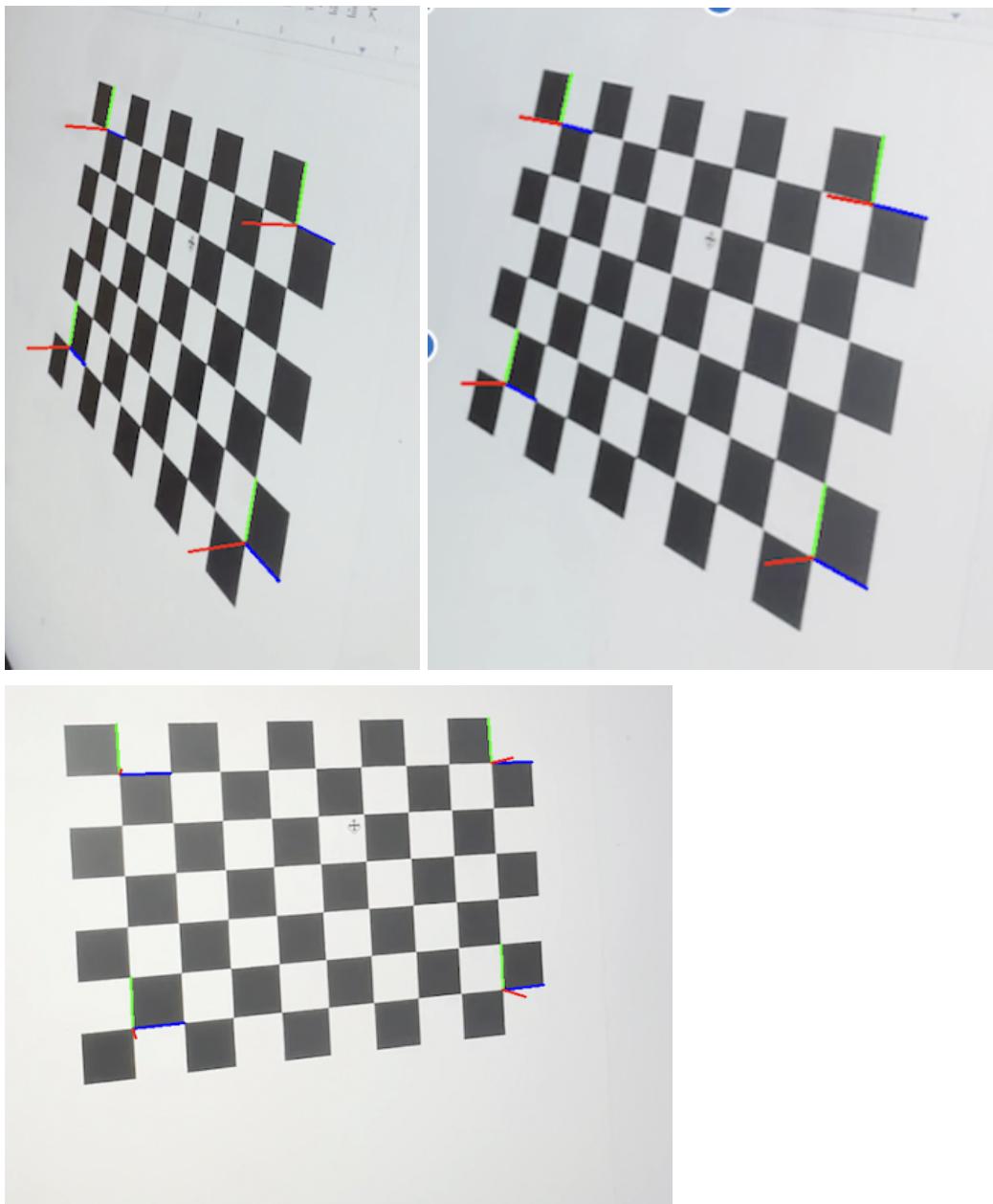
The u0 and v0 values are close to the x center and y center and also the two focal lengths are almost equal.

#### 4. Calculate Current Position of the Camera:

With change in the position of the camera with the target, both the rotation and the translation matrix values change because rotation matrix represents the amount of rotation across the 3 axes and translation represents the distance of the target object relative to the camera. Hence, if when the camera is moved along the x axis, the rotation and translation value corresponding to c changes. Going right of the target, increases x value and vice versa. This is the same for the other two axes as well.

```
Rotation vector:  
[1.693341049921935;  
 -1.51649378753235;  
 -0.8226703881045634]  
Translation vector:  
[-4.197087103366148;  
 7.706575330803512;  
 22.2125130476256]  
patternfound = 1  
Number of Corners for this Frame = 54  
Rotation vector:  
[1.701986165614613;  
 -1.525354353828743;  
 -0.8229589390548613]  
Translation vector:  
[-4.270980141967018;  
 7.677809948492701;  
 22.14974786253076]
```

5. Project outside corners of 3-d Axes:

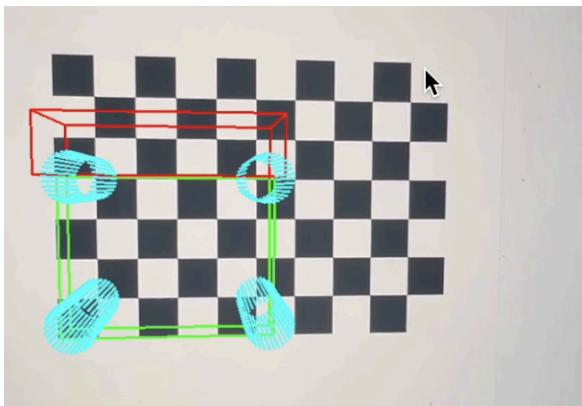


We can see the axis points projected on the target frame on all 4 corners with different colors representing different axes.

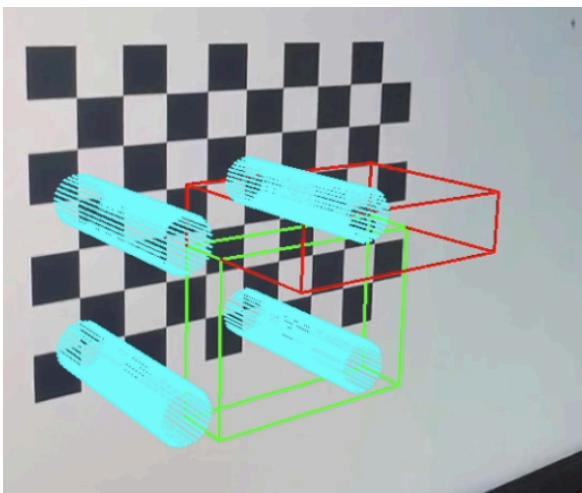
## 6. Virtual Object:

I have created a virtual object chair, which consists of a horizontal seat and a vertical back support. The chair is supported by 4 cylindrical legs.

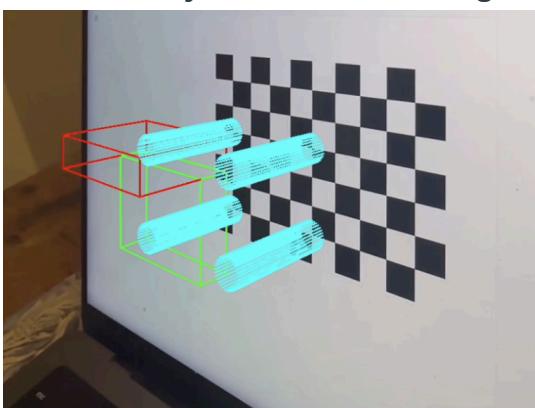
**1. Virtual Object viewed from front:**



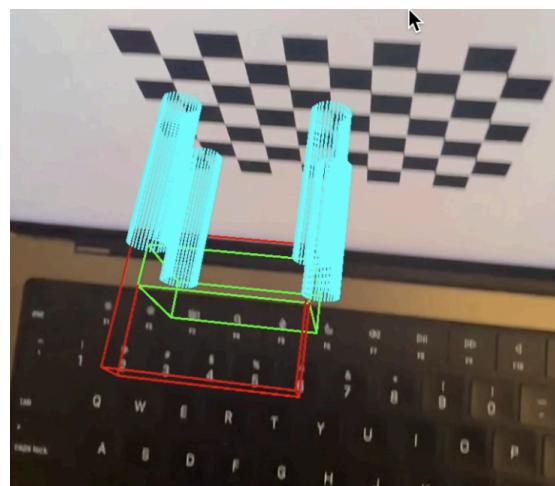
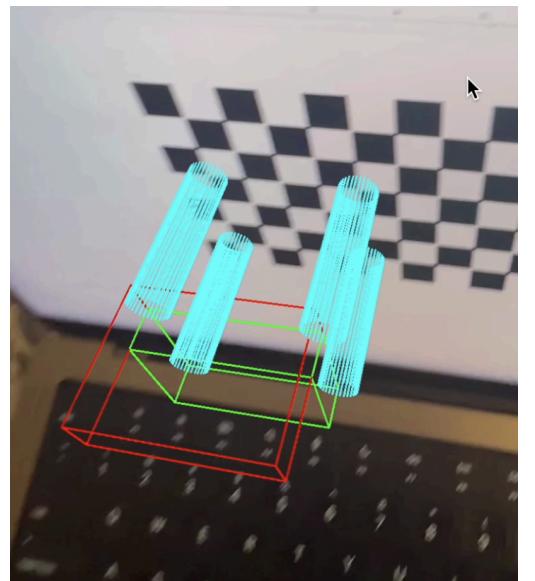
**2. Virtual object viewed from left**



**3. Virtual Object looked from Right**

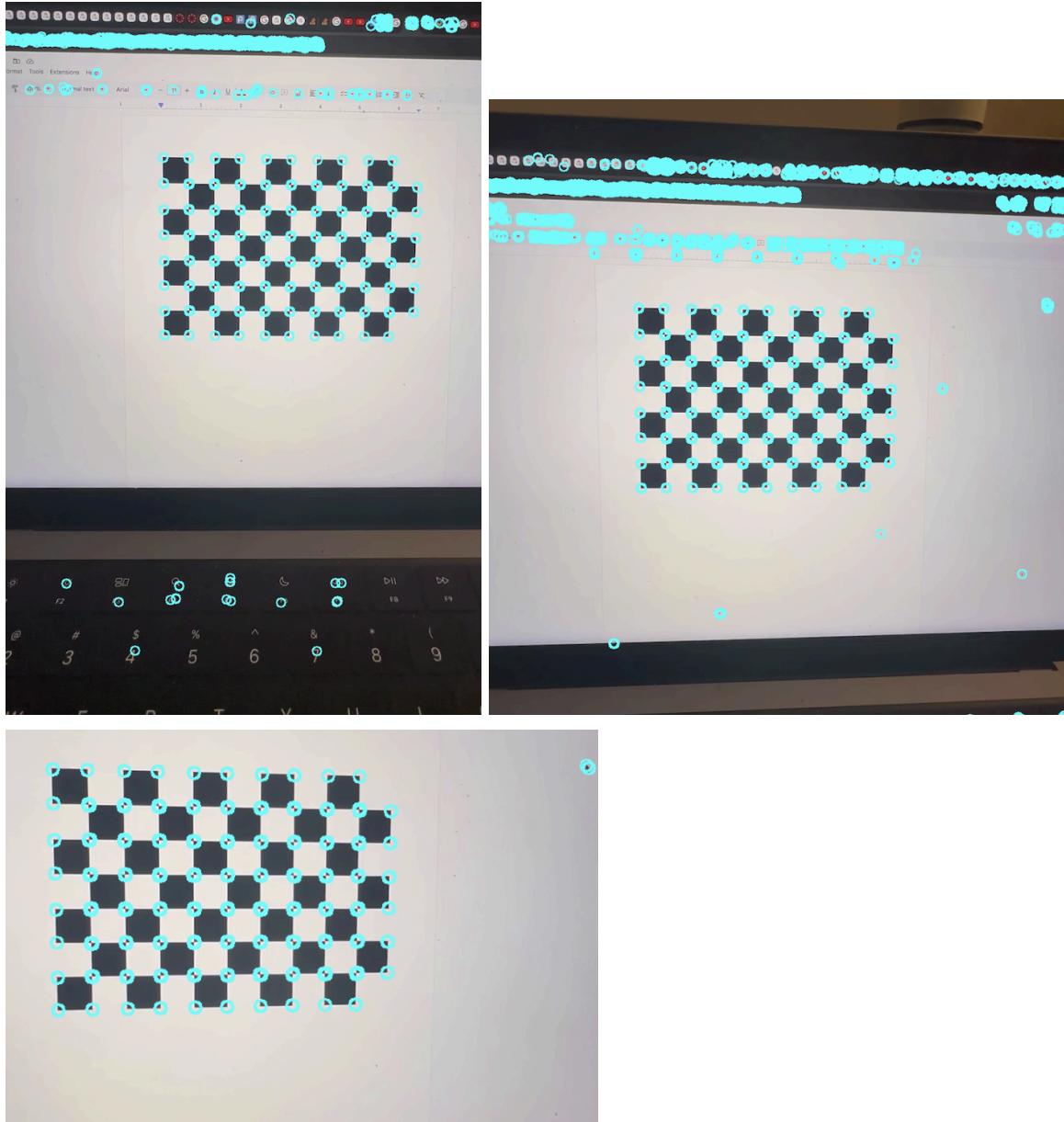


#### 4. Virtual Object viewed from Top:



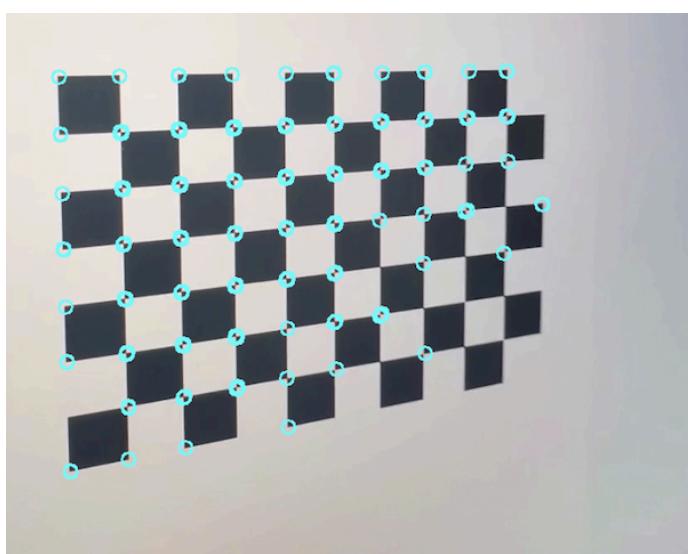
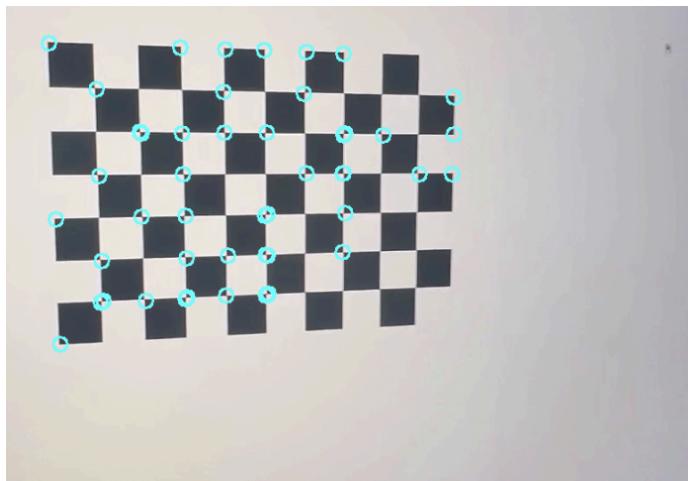
#### 7. Detect Robust Features:

I have used harrisCorner detector to calculate Harris features and use it to apply threshold to detect corners.



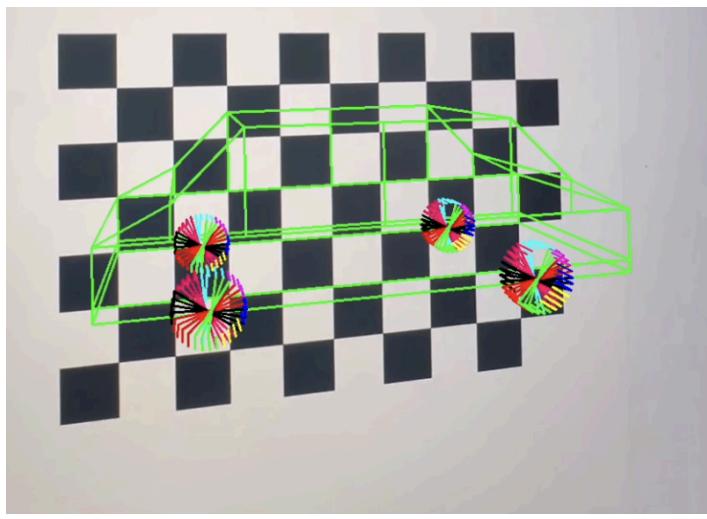
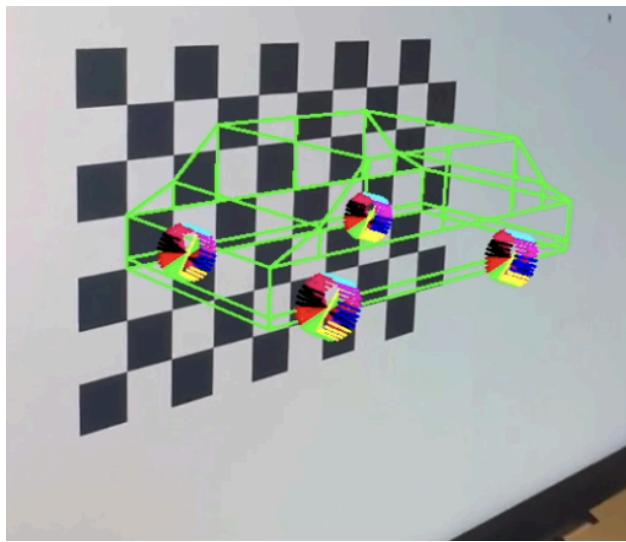
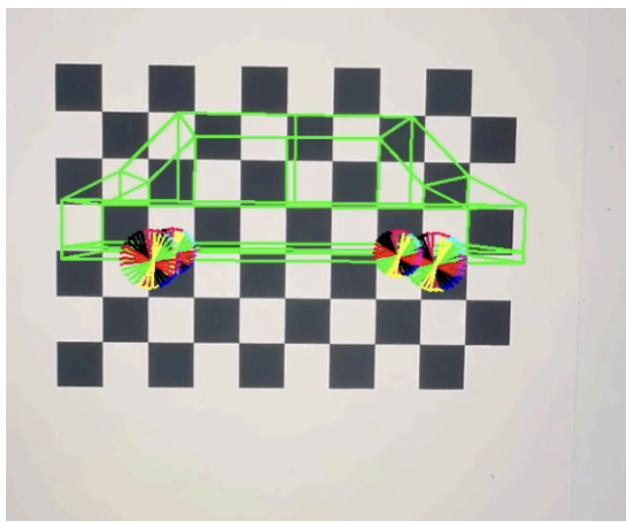
We can see that it clearly detects corners. Here, I have used a threshold 100 to pick the corner points and have created a circle to point the corners.

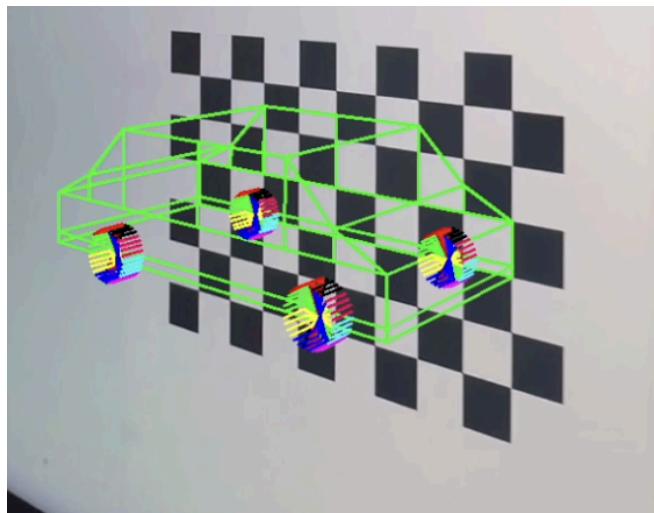
When I increase the threshold to 150, I can see that some points are detected and some points are not.

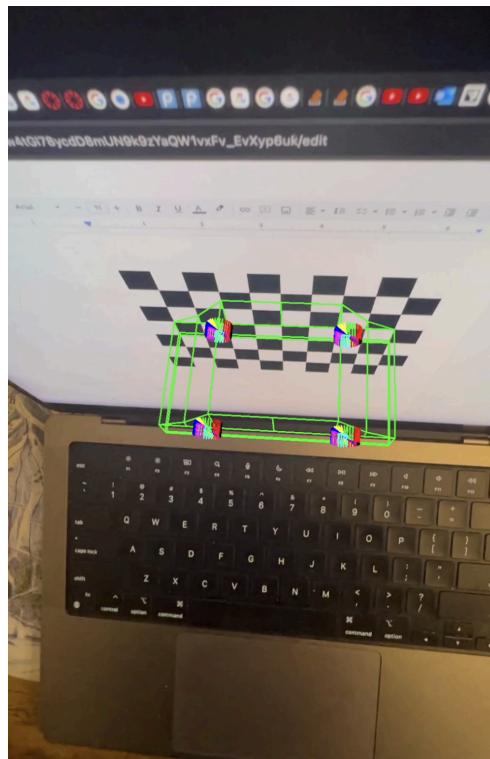
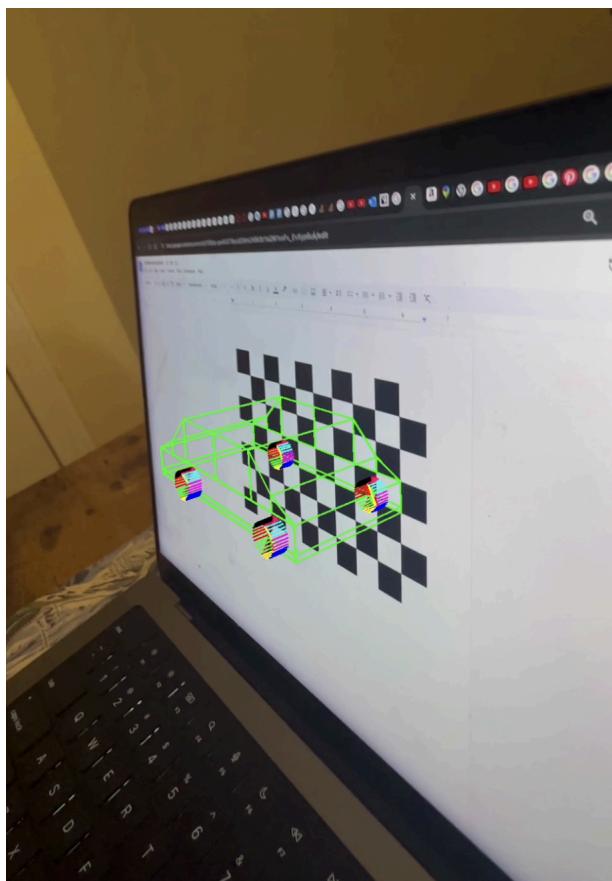
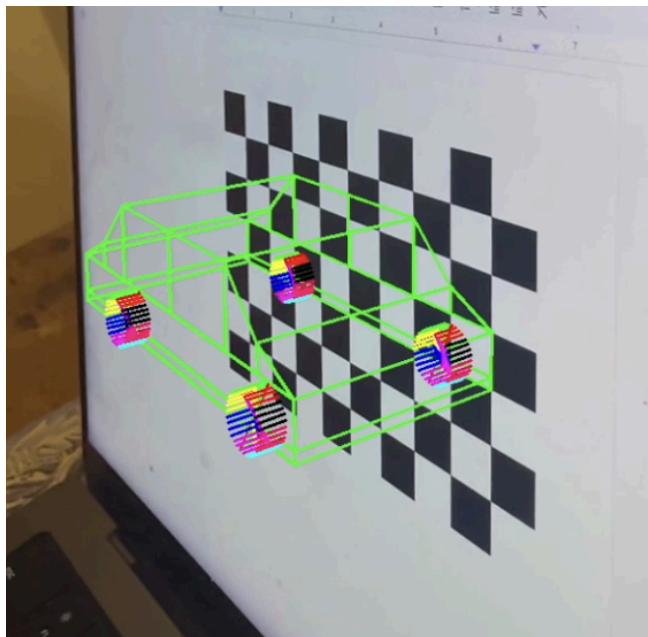


### Extensions:

1. I have also created one more virtual object, which is a car. Here, I have created an object which simulates a moving car.





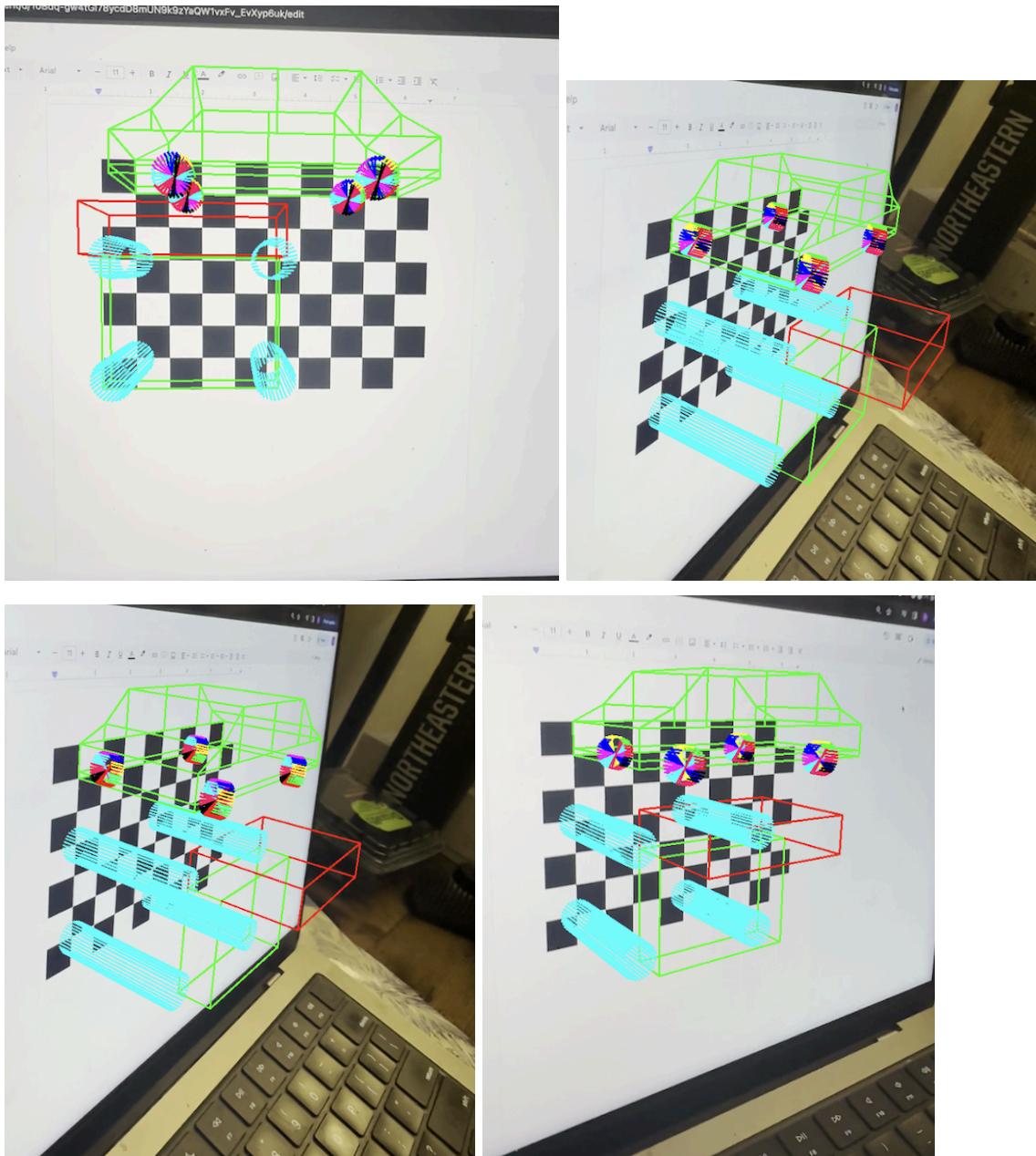


Demo Video Link:

<https://drive.google.com/drive/folders/1gEdjl6LNktKfVPiJa3bbL9dGjKz0oPGi?usp=sharing>

## 2. Integrating multiple objects in the same frame:

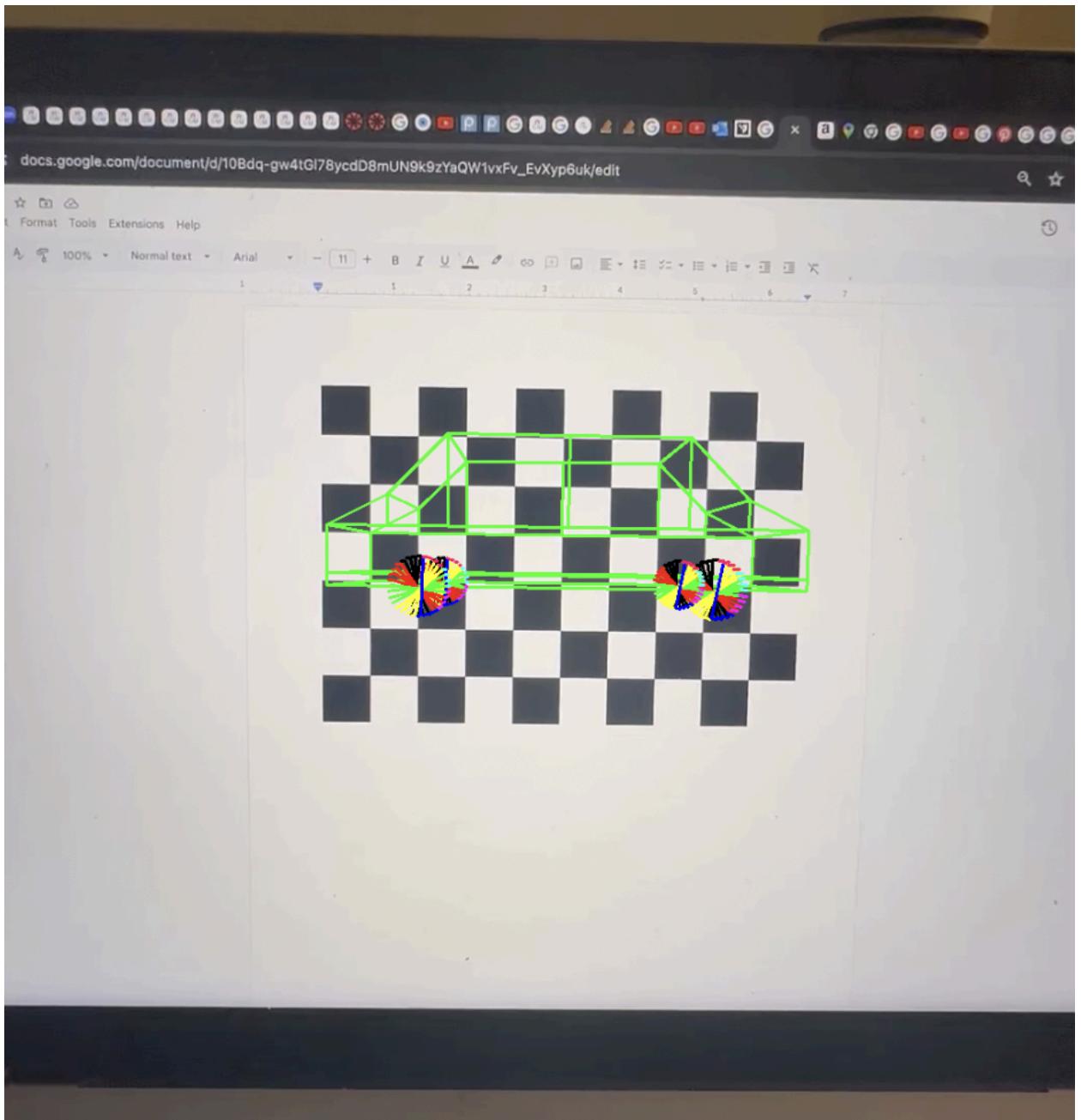
I have added multiple objects in the same frame. The two objects are chair and the car virtual object



## 3. Enabled using static images or pre captured videos:

I have integrated a command line interface to provide options to load custom videos or static images and create virtual objects on it.

Below image is obtained by passing a static image from command line



Similar results can be obtained for video as well by passing custom video.