

Simple Augmented Reality in OpenCV

Daniel Finger

CPSC 4820, Clemson University, Clemson SC, USA

E-mail: dzfinge@g.clemson.edu

Abstract:

Over the past few weeks in this course, we have worked on object detection (using aruco markers), camera calibration and facial detection; while working on these topics I became intrigued by the wide variety of usages they have, specifically augmented reality. This paper is aimed in attempting to demonstrate a simple creation of augmented reality aided by aruco markers.

Introduction

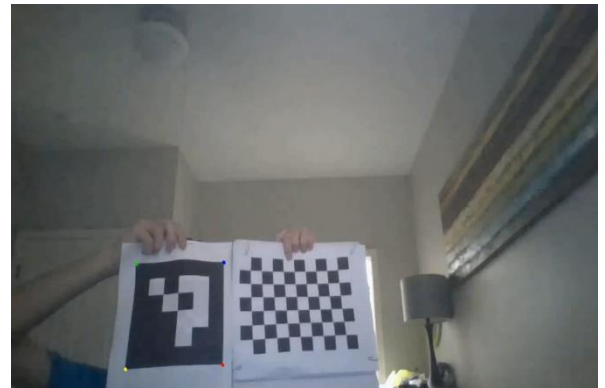
Coming into this class I had no prior knowledge on what an Aruco marker was, how to operate OpenCV, nor did I have any experience in facial detection, camera calibration or image manipulation. My goal for this project was to successfully track an aruco marker, determine its location and rotation on an image plane and use this data to attach an object to these boundaries to create an image that acted like it was in the image however its existence would be completely virtual. Initially I wanted my 3d object to be a high quality render (reminiscent of an actual real life item: bench, chair, dog, person etc); however upon starting the project I found I lacked the experience to implement a multi-level render with detail and mesh similar to what I described above. After making this realization I decided to keep my same goals in tact but reevaluate the detail in my rendered object, instead I would be rendering a bunch of 2d OpenCV draw functions to create a 3d object reminiscent of a cube, sphere or cone. By implementing it this was I would still be able to attach my object to the aruco marker but would be able to work with functions similar to what I'd explored in previous assignments.

Using OpenCV to detect Aruco Marker Corners

To create a 3d image from the aruco marker I would first need its boundary points so I could create a dynamic base for my image to lie on. OpenCV has functions that allow one to detect an Aruco Marker and return the X and Y coordinates for each of its corners in a matrix. After calling this function and extracting each x and y coordinate for all corners I was able to produce a video that followed the aruco marker and highlighted each corner in real time.

Single frame of the video shown to the right.

Each corner is color coated differently to make sure each corner contains its correct coordinate value.



Creating a Boundary

After detecting each corner, I now had the information needed to draw a boundary around the entire aruco marker. Each corner would be a start and endpoint of a line that together would encapsule the aruco marker entirely.

Detecting a Midpoint

At this point in the project, I decided my 3d object would be a triangular prism since there seemed to be a plethora of cubes on the internet, I wanted to try something new. In order to create a triangular prism though I would need the midpoint of the aruco marker. This was a fairly simple task to perform. I called all 4 x values and averaged them to get a “midpoint x” and then did the same for the y values. After creating a midpoint x and midpoint y value I could now use this coordinate pair to plot my midpoint on to the video.

Determining Rotation

The most demanding task of this assignment was trying to determine my rotation coefficient so I could show depth in my object by bringing it to the 3rd dimension. One way to solve this would be to use our rverts (object containing our rotational values relative to the camera screen), however, I was curious if I would be able to solve the rotation by hand since I had quite a few data points. I realized that when the aruco marker was directly centered and straight to the camera its local rotation would be 0 degrees. With that knowledge I could find its x dist and y dist to find its full dimensions. With the initial x and y dist values I could extract new x dist and y dist values every frame and compare them to the original and performing a few short math functions I could solve for the aruco markers local rotation. Since the aruco markers local rotation on the x and y axis would be identical to that of the object I was rendering I would be able to create an arrow protruding perpendicular to the center of the aruco marker.

Prism Creation

Now that I had an arrow perpendicular to the midpoint of the aruco marker I could set that arrow length to the height I wanted the triangular prism to be. I would then just need to gather the coordinates of the endpoint of the arrow every frame and create a line between it and all 4 corners to create a triangular prism.

One Frame of the prism shown to the right:

With this implemented I now had a fully augmented reality triangular prism in my video that moves and rotates with the aruco marker.

