Problem: The video *hw4.avi* is a video of a CD player with two ArUco tags attached to it. Your mission is to use computer vision to recognize and identify the tags, and display the location of the "on/off" switch to the user. The markers are 4×4 bit patterns, size is 2 inches on a side, and the markers were created by specifying 100 markers in the dictionary. The marker on the left is id = 0; the marker on the right is id = 1.
Here is the location of the switch (in inches):
With respect to marker 0: $(X, Y, Z) = (2.5;\ -2.0;\ -1.0)$.
With respect to marker 1: $(X, Y, Z) = (-2.5;\ -2.0;\ -5.0)$.
The camera's intrinsic parameters are $f_x = 675,\ f_y = 675,\ c_x = 320,\ c_y = 240$, assume no distortion.

Your program should recognize a marker, determine its pose, and overlay a pyramid on the image showing the location of the on/off switch as an overlay.

Refer to attached script *HW4.py*. The pyramid corner coordinates *Pyramid* are defined; these were adjusted based on the appearance of the projected pyramid on the video. Also, the ArUco dictionary is defined for the 4×4 bit patterns with side length of 2 inches. The *aruco.detectMarkers()* function is called to detect any arUco tags. If there is any marker identified, the boundary is drawn using function *aruco.drawDetectedMarkers()* and the rotation and translation vectors for the pose are found using function *aruco.estimatePoseSingleMarkers()*. The XYZ coordinate axes of the marker are then drawn using the rotation and translation vectors in function *aruco.drawAxis*.
Next, homogeneous transformations are considered to map the pyramid onto the switch. For the $4 \times 4\ {}^{c}_{m}H$ matrix representing the transformation matrix for the marker with respect to the camera, the rotation and translation vectors are used. The rotation vector is transformed into a 3×3 matrix using function *cv2.Rodrigues()*; this R matrix is placed in the upper-left block matrix of ${}^{c}_{m}H$. The translation vector of the marker in homogeneous coordinates forms the last column of ${}^{c}_{m}H$. To find the transformation of the switch to the marker ${}^{m}_{s}H$, a rotation about the y axis of the marker is considered and the provided translation vectors are used for the respective markers. For marker 0, the switch is at a rotation of +90 degrees, $a_y = \pi/2$. For marker 1, the switch is at a rotation of -90 degrees, $a_y = -\pi/2$. These angles are used to form rotation matrix $R_y = ((c_y, 0, s_y), (0, 1, 0), (-s_y, 0, c_y))$, where $c_y = \cos(a_y),\ s_y = \sin(a_y)$. Rotation matrix R_y forms the the upper left block matrix in ${}^{m}_{s}H$ and the translation vector in homogeneous coordinates forms the fourth column of ${}^{m}_{s}H$. Finally, the transformation for the switch with respect to the camera is ${}^{c}_{s}H =^{c}_{m} H^{m}_{s}H$ and the projected pyramid points are found as $p_C = KM_{ext}Pyramid$, where M_{ext} include the first 3 rows of ${}^{c}_{s}H$ and K is the intrinsic camera matrix. Points p_C are converted to 2D homogeneous coordinates and are then drawn on the frame by using *cv2.line()* to form the wireframe of the pyramid.
The resulting video can be found here:
`https://drive.google.com/file/d/1gAUSfAKBav1ItA6ZgMnP8RDjUN6dzcrT/view?usp=sharing`.