CS764: Assignment 6

Report

Siddharth Saha 170100025 **Parth Shettiwar** 170070021 **Parikshit Bansal** 170050040

1 Introducing AR

1.1 Drawing book

In this subpart, we had to draw the book on the image after proper transoformation of its coordinates. The following methodlogy was used:

- We collected 6 images of our notebook, which we used in last lab. Using a pen, we had created a grid with 3 cm apart points.
- Now we decided the world coordinates which have to be mapped to the image coordinates. The world coordinates used are: $[0,0,0],[3,0,0],[6,0,0],[0,3,0],[3,3,0],[6,3,0]$. The corresponding image coodinate for origin is the bottommost leftmost point in the images with x-axis as 1st coordinate and y-axis as second coordinate.
- Using the lab4 script, we got our manual coorespondences by clicking on the points in all 6 images.
- We used the `cv2.calibrateCamera` function to get our camera calibration matrix, used in later parts.
- Now we use `cv2.solvePnP` to get the rotational and translational vectors required for projection of any 3d points.
- Now we use the `cv2.projectPoints` function to project our book real coordinates to the image. The book coordinates used are : $[[0,0,0],[0,b,0],[l,b,0],[l,0,0],[0,0,w],[0,b,w],[l,b,w],[l,0,w]]$ where l,b,w are length breadth width respectively, with l along x-axis, b along y-axis and w along z -axis.
- Now finally after receiving our 8 corners, we use `cv2.drawContours` and `cv2.line` functions to draw the boundaries of our cube.

1.2 Drawing book given any position and rotation angle

For this part, we do a little twick to find the new projected book points given a rotated angle theta and x,y coordinates on image for book to be placed.

- The new centres x,y are given in percent form of the image width and height. First we convert these percent to the pixels we want our book to be at. Call it newCentre. Then in the original setup, we find the centre of the bottom face of book by averaging the project coordinates of diagnol points. Call it oldCentre. We compute the difference: newCentre-oldCentre and add this translation to all the projected corners of the book directly.

- Now to account for the rotation part, we first cinvert our input theta into radians and then compute the standard rotation matrix. We do our rotation around the bottom centre of the book (l/2,b/2). Unlike translation, we do our rotation business on the real world coordinates. We pre-multiple the following matrix with the book coordinates (defined mentioned in previos part).

$$\begin{bmatrix} cos(\theta) & -sin(\theta) & l/2] \\ sin(\theta) & cos(\theta) & b/2 \\ 0 & 0 & 1 \end{bmatrix}$$

  Important to now: While pre-multiplying with this matrix, we specifically made the z-coordinate of the given points = 1 , since we know rotation is alway in the place parallel to x-y plane for all corners.

1.3 Adding Realism

In this part, we had to add the texture covers over the drawn book images. The texture images have been used which were given in the data folder. The main challenge in this lied in finding which faces of the book to cover, since at max we can observe 3 faces of a cuboid and rest are occluded. The algorithm is mentioned in the question asked at last. See Section 3 for further details

2 Results

2.1 Drawing Book

As mentioned previosuly, we placed our book such that leftmost corner is on origin of the grid and length , breadth on x and y axis respectively. Following are results obtained for few of the input images. In our conversion, we found that 1cm = 1unit. Hence when l,b,w = 3, we get a cube with its bottom face exactly aligned with 4 adjacent gird points on the image.

Figure 1: Results for w,l,b = 3

2.2 Drawing book given any position and rotation angle

Following are results obtained for few of the input images with varying parameters for the x,y and theta.

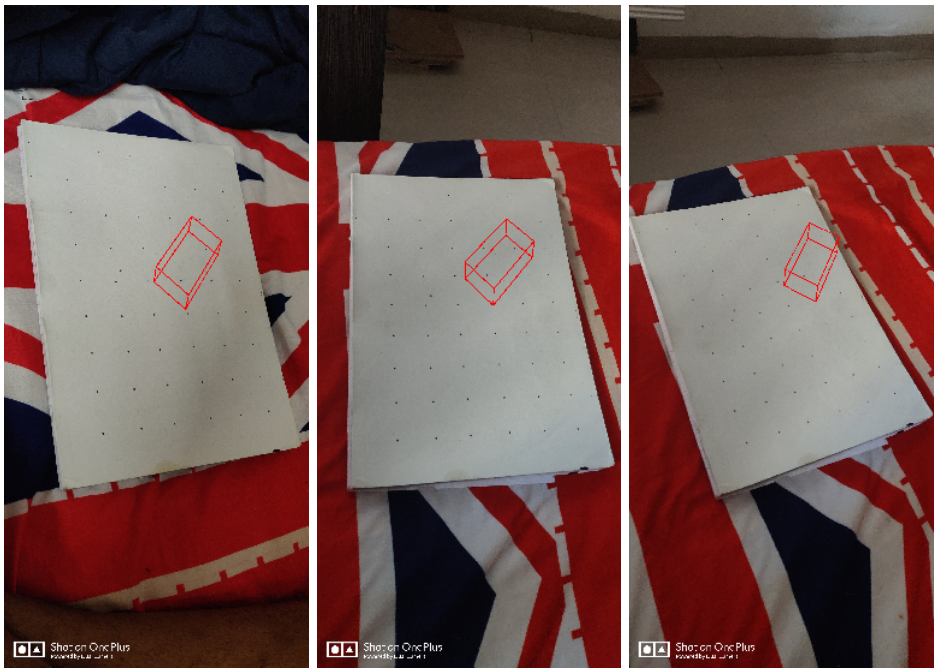

Figure 2: Results for w = 1.5, l = 5, b = 3,x = 60, y = 40 ,theta = 45 degrees

Figure 3: Results for w = 1.5, l = 5, b = 3, x = 30, y = 50, theta =150 degrees

2.3 Addin Realism

In this part, we show our final output images of the book. It is ensured that all images cover all the sides of the cube atleast once.

Figure 4: Results for w = 1.5, l = 5, b = 3,x = 60, y = 40 ,theta = 45 degrees

We perform another analysis, where we rotate our book, in the same image. Following is the result:

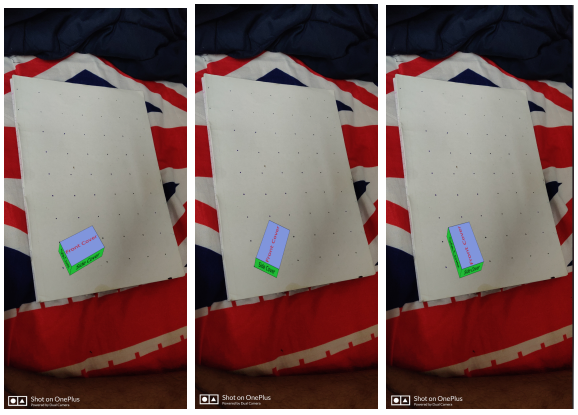

Figure 5: Results for w = 1.5, l = 5, b = 3,x = 60, y = 40 ,theta = 20, 60, 100 degrees

3 Questions

Q1.1 Have the choices of views been made so that all the surfaces (except the back-cover) visible in at least one view?

Ans Yes, we have used 6 images, and they cover all the 5 sides of the cube atleast.

Q1.2 How did you texture the book in the various views? If we were to provide you with a new view, or rotate the book by 90 degrees will your code be able to render the final result correctly? We are interested in making sure that the surfaces that would be visible for a real book are consistent with the ones visible in AR.

Ans

Some observations to be noted here are that there are only 3 surfaces visible from any view, with one of them being the top surface and the other being 2 of the 4 side surfaces. To find these visible 2 sides we do the following

1. Find the Camera Position in the real world [6], which is done using the rotation and translation matrix returned by solvePnP.
2. Find the closest and farthest of the bottom 4 points from the camera
3. Note : the closest and farthest points are diagonally opposite and hence don't share a side.
4. Draw the 2 sides passing through the closest point [5]

Note we have ensure that the Side Cover is consistent with all the views.

References

[1] https://docs.opencv.org/master/d7/d53/tutorial_py_pose.html

[2] https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contours_begin/py_contours_begin.html

[3] https://stackoverflow.com/questions/35902139/opencv-3-1-drawcontours-215-npoints-0

[4] https://answers.opencv.org/question/117734/opencv-error-bad-argument-unknown-array-type-in-cvcvar

[5] https://stackoverflow.com/questions/14878467/how-to-generate-texture-mapping-images

[6] https://stackoverflow.com/questions/14444433/calculate-camera-world-position-with-opencv-python