Assignment 3

Augmented Reality Application

Computer Vision
COL780

Report By:

Rushil Gupta - 2017CS10487
Gohil Dwijesh - 2017CS50407

NOTE: All the parts have been done using the webcam in our laptop

Part 1

Ques) Calculate the camera calibration matrix for your webcam using multiple views of a chessboard pattern.

Algorithm:

- We took multiple views of a chessboard pattern, all captured by the webcam.
- In all these multiple views, we detect the location of the corners in the chessboard whose location in world coordinate system is known to us.
- After we get point correspondences, we perform matrix calculations to extract the intrinsic camera matrix and the (R,t) of the camera

  Answer) Intrinsic Camera Matrix

```
[[637.58117676   0.         329.60614713]
 [  0.         630.13122559 251.07025762]
 [  0.           0.           1.        ]]
```

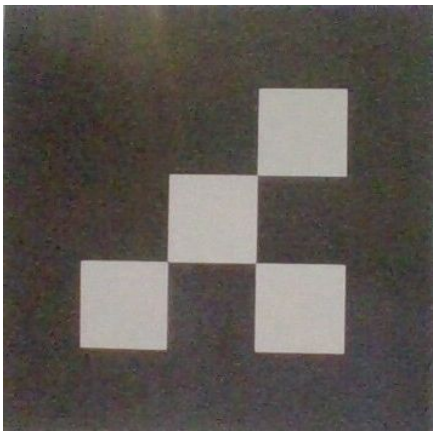
Part 2-3

Ques) Create 3 visual markers that are easy to detect. For a single visual marker, take video input from the webcam. Estimate and explicitly print out the pose of the camera (R,t) for every frame. Render a 3D model (.obj file) on top of the visual marker as if the model is lying vertically on the marker.

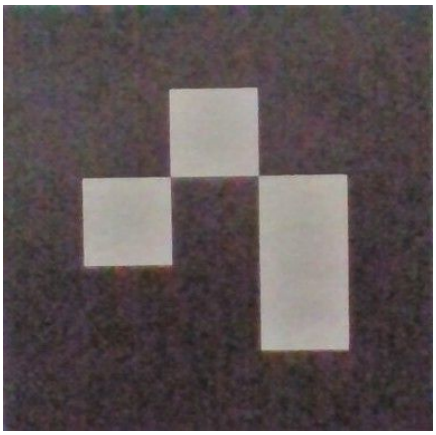
Ans)

Markers Used:

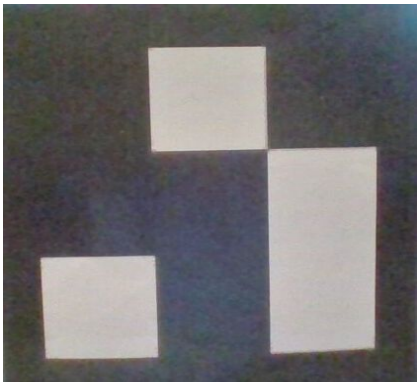
Glyph 1



Glyph 2



Glyph 3

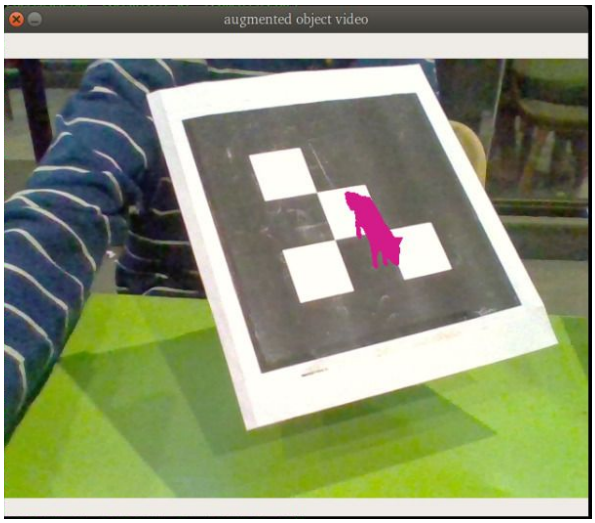


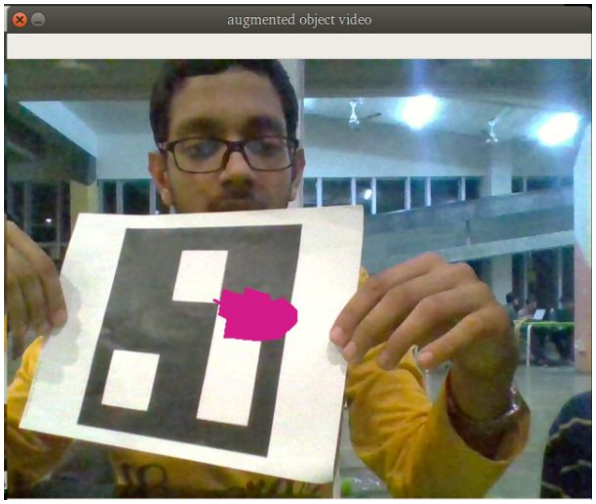
Algorithm:

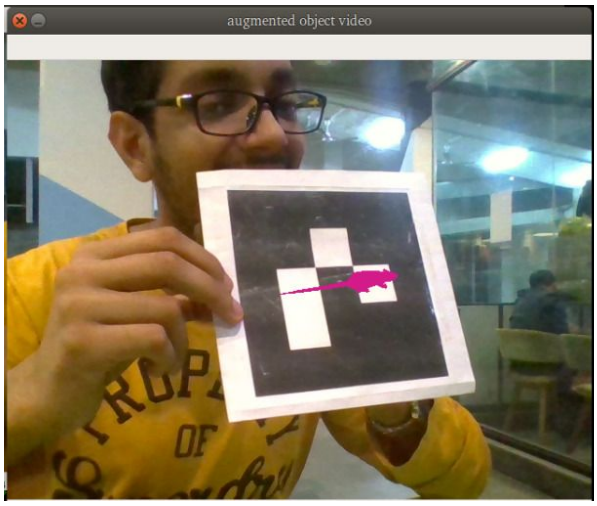
1) We capture the frames to analyze in real-time with the help of a webcam.
2) We convert the image first to a grayscale image and apply a Gaussian Blur and Canny edge detector on the grayscale image.
3) On the output of the Canny edge detector, we detect the closed contours.
4) For each of the contours, we try to approximate the contour via the help of a polygon using the function approxPolyDP in OpenCV.
5) This function returns the coordinates of the vertices of the polygon approximating the contour.
6) We check that in our case for a contour containing marker it will be best estimated by a polygon with 4 vertices.
7) From the contour which may actually contain the marker pattern, we first transform the image area enclosed by contour to bring the pattern upright.
8) This transformed area is then resized and then a median blur is applied to mitigate the effect of noise in neighboring pixels.
9) This area is then checked for the marker pattern. Every marker is represented as a 9 element vector of 0 and 1 denoting area of black and white. This 9 element pattern is generated for each of the areas that passes the above tests to confirm whether it is the marker or not. All the four rotated orientations of the area are considered for matching to the template glyph.

10) After we get the 4 points in the image where the marker is present, we compute the 3D transformation for the respective glyph.

11) This 3D transformation matrix is then used to project the points in the obj file onto the image and hence render the 3D object on top of our marker in the image.







Part 4

Ques) In this part, there are two visual markers. The first visual marker lies on a flat surface. The second marker is in a plane roughly perpendicular to the plane of the first marker. The augmented object starts moving from the first marker and stops when it hits the second marker.

Algorithm:

1) With the help of the algorithm employed in the previous step, we detect the two markers in the image.
2) Then the object is always rendered on the glyph 2 mentioned above.
3) Now, we need to focus on the direction of motion of the object and the distance it needs to travel before it would actually hit the other marker that lies in the perpendicular plane.
4) For the first, we compute the midpoint of the marker 2 and the lower line of the other marker boundary.
5) The line joining these two points gives us the direction of motion of our object.
6) For the distance, we calculate the distance between these two points and take a fraction of this distance.
7) Now, as per the movement we have encoded Newton's Laws of Motion in our code.
8) We have kept an acceleration for our object and an initial velocity.
9) The object begins moving in the direction calculated above.

10) Between two frames an object is assumed to be moving with constant speed i.e. for a time of 1/fps (fps stands for frames per second)
11) As soon as the object gets through half of the distance it has to cover, it receives the same acceleration but with the sign reversed i.e. a deceleration of same magnitude. Because of this deceleration, it stops exactly at the point where it strikes the other marker.
12) The terminating condition in our motion is that the object automatically comes to rest when it is about to strike the other marker.