Project 2

3D model generation using single view metrology

Team Members :-

Harish Pullagurla :- hpullag@ncsu.edu

Sharva V Hiremath :- shirema@ncsu.edu

Problem Description: -

We try to generate a 3D model from a single image, through the concepts of vanishing points and vanishing lines to compute the camera matrix.

In this project, we try to compute the vanishing points automatically.

The steps followed during implementation of this process are as follows

1. **Image Acquisition**

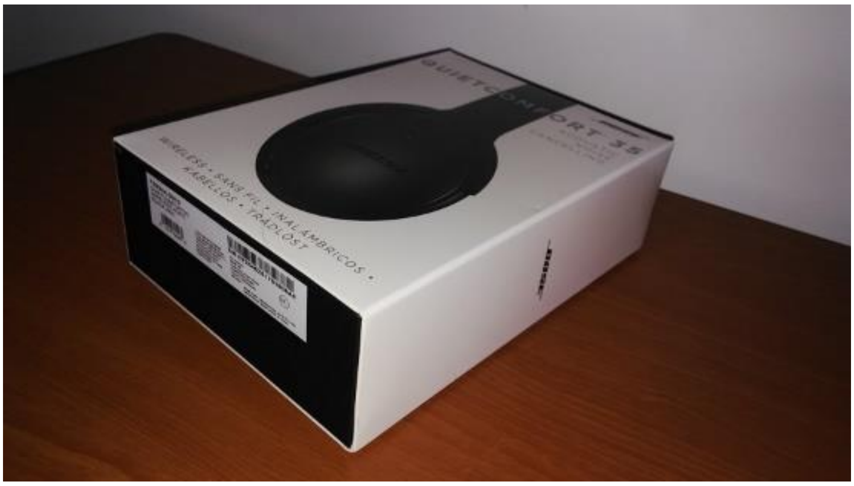

Fig 1 : showing the image acquired

An image of a box on top the table is captured using a mobile camera as shown above. It is taking keeping in view of the perspective geometry.

2. **Calculation of Vanishing Points**

Automatic detection of vanishing points is tried here.

1. **Lines Segment Detection**

First the image is converted to a gray scale image and is given as an input to Line Segment Detector module.

This module is present in the library pylsd and is called in the format lsd (gray_image). Output from this module is a set of lines, each of which is specified by 2 end points (x,y) and the width corresponding to it.

Post processing of the obtained lines is done to ignore the erroneously detected line segments. Here we remove the lines with width less than 2 pl and length less than 20 pl

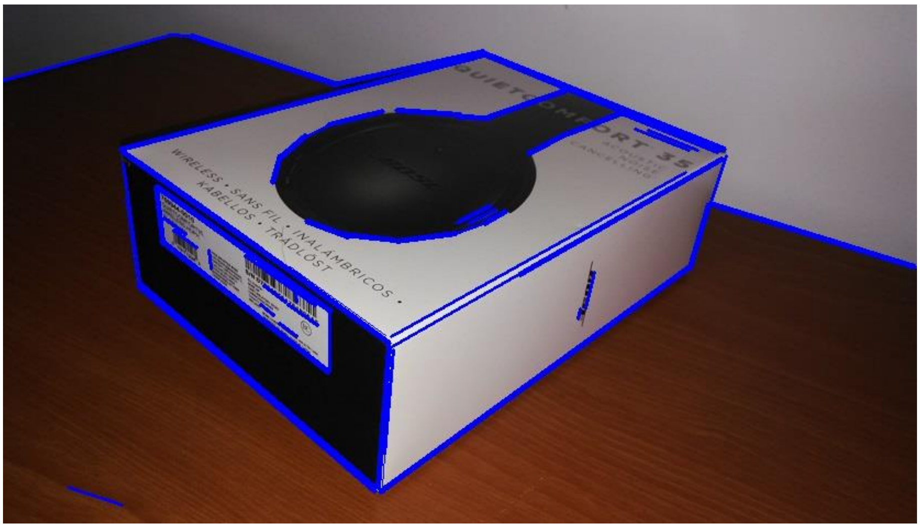

Fig 2: All the lines obtained from the line segment detector function

2. Line Segment Clustering

Before attempting to calculate the best fit point from the lines detected. It is important to cluster the lines based on the orientation. A vanishing point in each of the orientations is needed. Hence the given lines are clustered into 3 parts corresponding to 3 axis. Equation for the lines is found out by a cross product function of points on the line segment represented in homogeneous coordinate format. Slope is found out for each of the lines and the angle corresponding to each of them. Angles come in the range of 0 to 180 degrees which are to be clustered into 3 categories based on the orientation.

Here the lines with 0 & 180 are actually the same, so the data – in angles format is circularly periodic. K means clustering is used to classify the given data into 3 clusters.

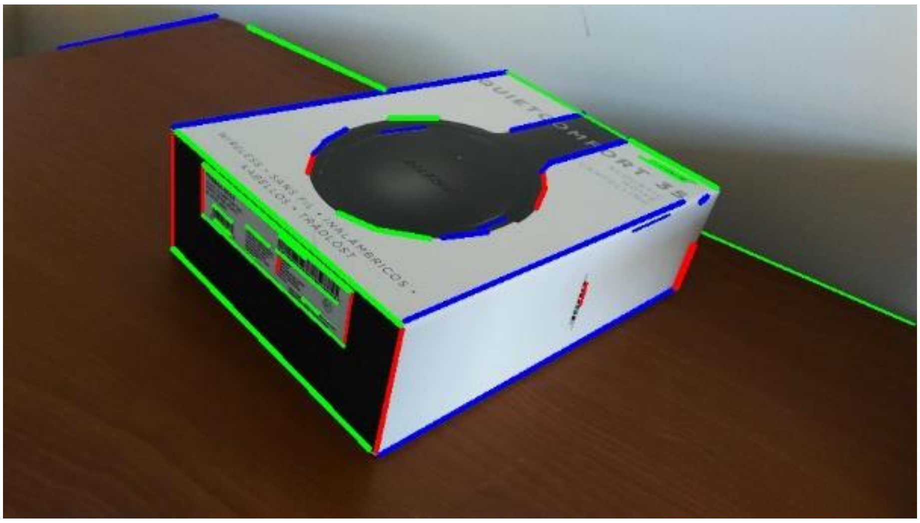

Fig 3: Figure showing the Lines after line detection in 3 different clusters (using different colours i.e. red, blue and green)

3. RANSAC

Once the lines are obtained from using line segment detection, we use RANSAC to find the vanishing points from these lines in 3 different clusters.

For the RANSAC we iterate over 'N' times. Following which we randomly select 2 lines from cluster 1 (cluster 1 lines correspond to lines along x axis) and find the intersection point and normalize it. The number of inliers are calculated by measuring the perpendicular distance between the point and each of the normalized lines in the cluster. If the distance is less than the threshold then it's included in the inliers count. Then we consider the line set with the maximum inlier count and minimum distance. This gives us the vanishing point along x axis i.e. V_x. The same process is used to calculate V_y and V_z.

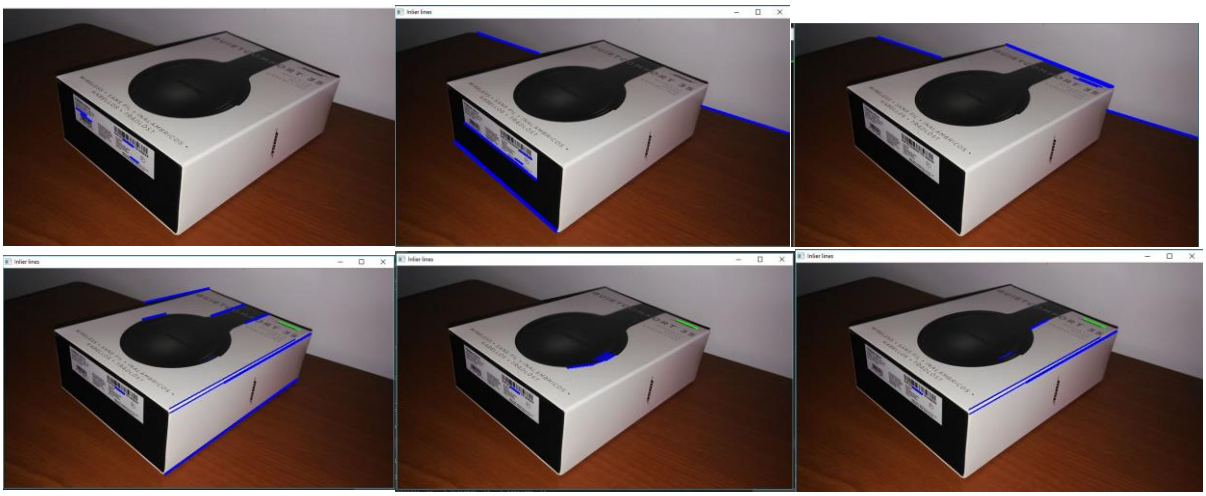

Fig 4 : Figure showing the different iterations of the RANSAC algorithm showing the inliers

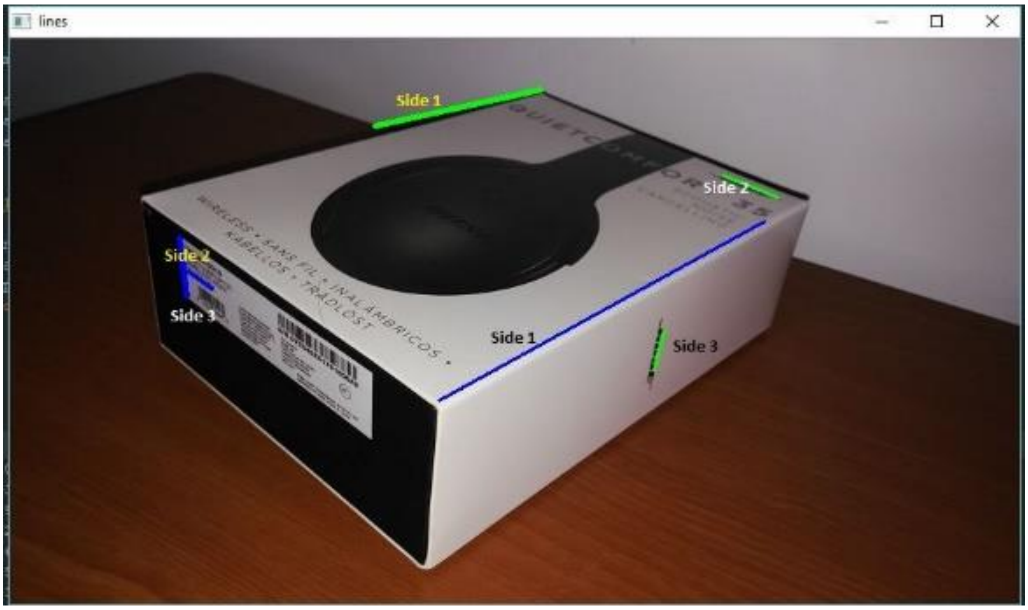

Fig 5 : Figure showing the lines chosen for calculation of vanishing points

3. Computing the Projection and Homograph matrix

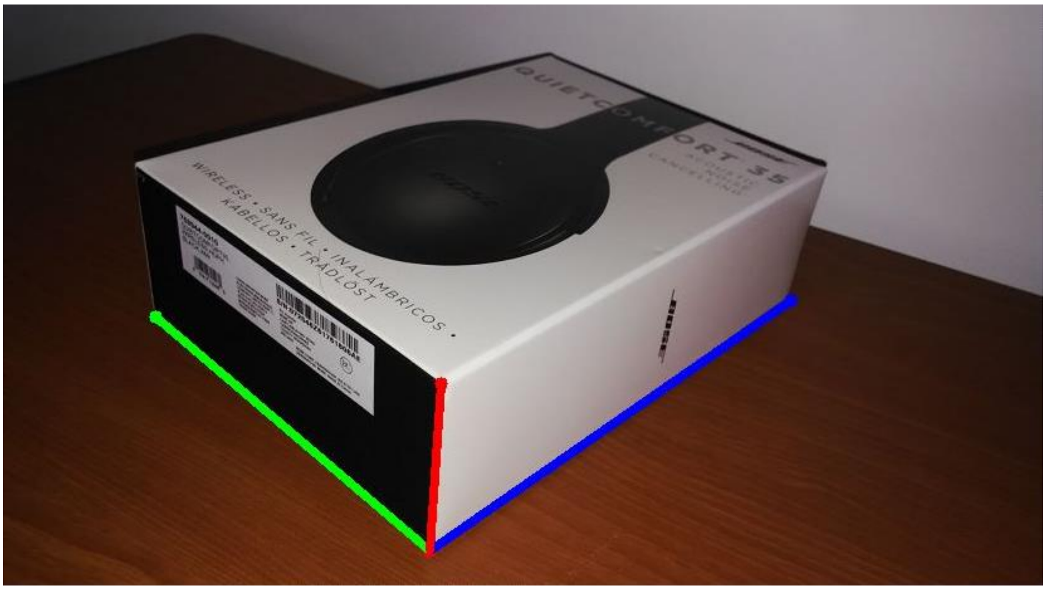

Figure showing the axes (x, y and z) on the object

Once the vanishing points in each of the directions is obtained. It is possible to compute the Camera Projection matrix from it.
Vanishing point in the x direction corresponds to the point at infinity in the 3d world. Representing it in as homogeneous point Vx gives us (1,0,0,0) . which corresponds to P1 in the projection matrix. Similarly with Vy,Vz being P2,P3.
The origin point selected on the image acts as P4 which is subjects to scaling
Thus , The projection matrix P = [$C_1.V_x$ C2.Vy C3.Vz 0], where C_1, C_2 and C_3 are constants.

These constants are evaluated by comparing the measurements to the real world values
Scaling constant = lease sqr sol (Vx,R1-Origin) / physical distance between R1 & O
Where R1 is the point on x axis

4. Computing the texture maps

The homograph matrices H_{xy}, H_{yz} and H_{zx} are obtained from the projection matrix itself by considering the 1st, 2nd and 4th columns of P for H_{xy}, 2nd, 3rd and 4th columns for H_{yz} and 1st 3rd and 4th matrix for H_{zx}.
Perspective warping is done on the image with each of the Homography matrices with the inverse warping flag set. The output image appears as if take from a different view.
Cropping out the surfaces we get

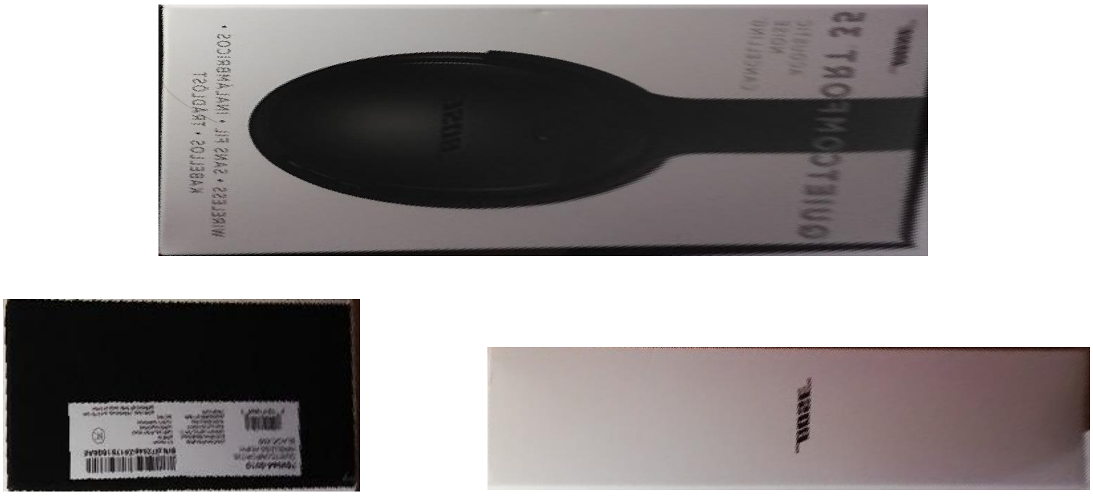

Figure showing the three texture maps obtained

5. Visualizing the 3D reconstructed model

For the 3D model, the coordinates for each face of the box is defined and the corresponding images to be mapped on to it is specified. For the model generation we have assumed the opposite sides to be similar.
Image of the 3D model is shown below:

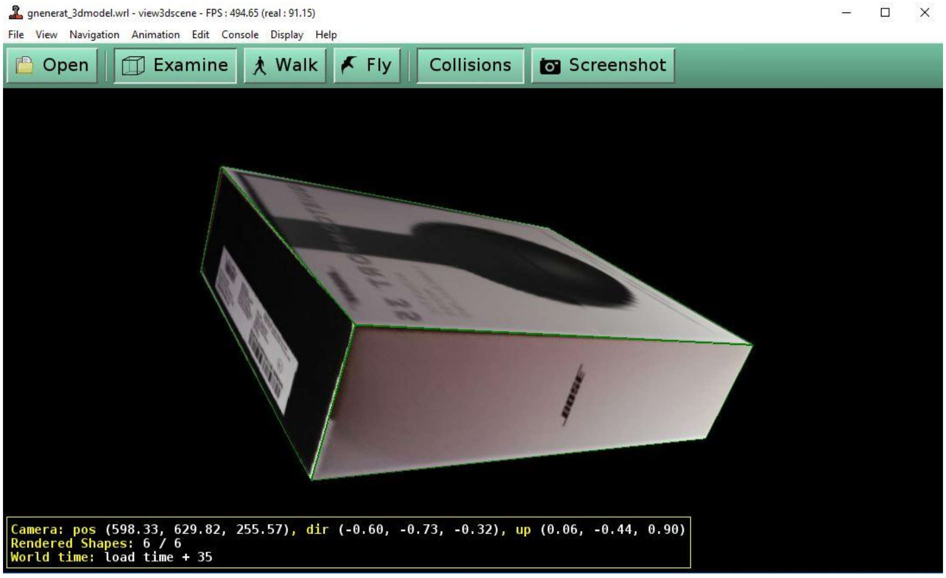

Figure showing the 3D reconstructed model of the box