Autonomous car using DIP

Shuvam Mondal

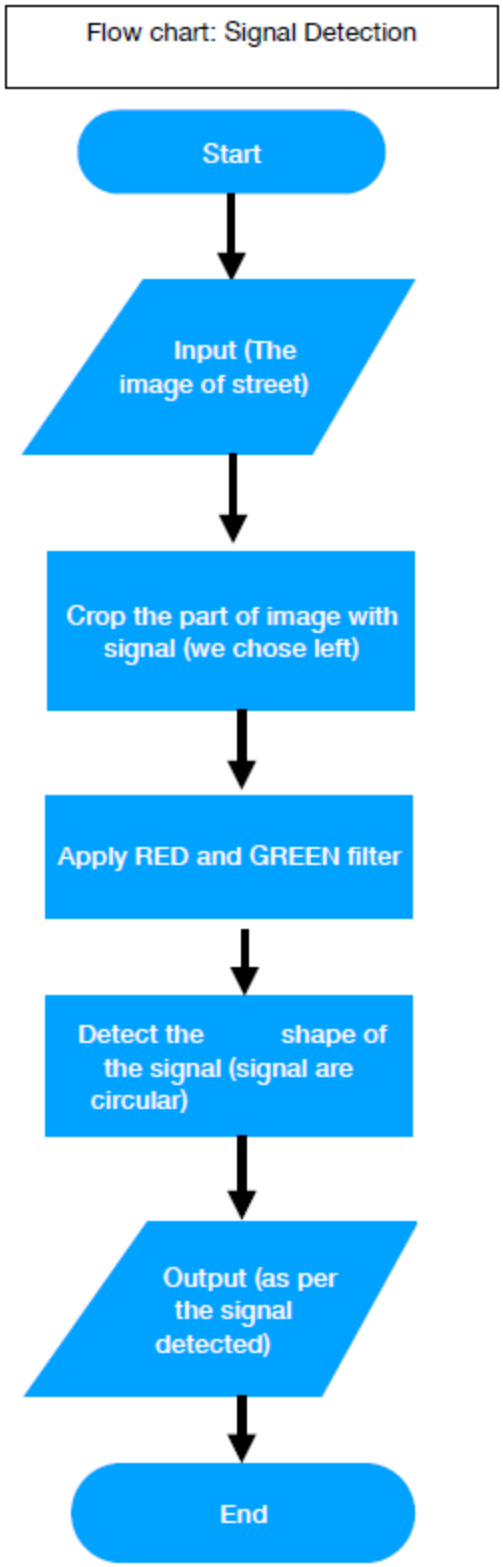
UID - 162406

Abstract:

This Digital Image Processing project has two parts. In both the parts we tried simple ways to how how a car can recognise if the signal is Red or Green and taking the appropriate action. In he second part we tried to find the coordinates of an object in 3-D space of local co-ordinate space using the corresponding address of the intensity value in the image. In the next part we give a way of keeping the vehicle on road and turn as the path goes.

Introduction:

The world is currently pursuing, making lives of human beings more comfortable than ever before, and the main role is played by the large tech. Companies like Google, Uber etc. in creating what we are all expecting soon enough the ‘Autonomous Car’ even though its trials are currently being carried out by most of them and we also hear about the difficulties that they face day to day functioning. Here we try to recreate the basic techniques of how using a simple algorithm we can check if the signal is red or green! This is important as we can all guess and here, the image of the street that we chose only contains red and green signals and not the yellow colour as we are deeming it not important. In the second part we use the mathematical concept of Transformation called Direct Linear Transformation or called DLT in finding the 3-D co-ordinates of an object, using the 2-D co-ordinate of that object in the image plane (technically the z co-ordinate is 0 when we consider the image co-ordinate). Here the thing that I would like to point out is that we are using the row and column address of the corresponding object co-ordinate in the image plane as the x and y co-ordinates here. This is really a poor way to replicate a real life scenario and its accuracy is not that good compared to what tech companies use in real life. Here I have applied DLT to find what would be the distance of lets say, the tyre from the edge of the road and I believe that if we can apply this to commute what would be the distance of the car from an obstacle in the front by just using the DLT method.


Flow chart: Signal Detection
Start
Input (The image of street)
Crop the part of image with signal (we chose left)
Apply RED and GREEN filter
Detect the shape of the signal (signal are circular)
Output (as per the signal detected)
End

Signal detection:

1. For signal detection we used two very simple techniques
• Color filtering and
• Circle detection

2. The image of street (for simplicity, the image used is not that of a real world) is cropped in such a way that most of the unnecessary background is removed and we have part of the image with signal. We chose left side.

3. The cropped image then is filtered with Red and Green filter. If the signal is Red, on applying the Red filter we get a ‘bright’ circle and rest of the background appears dull or darker in shade similarly with Green signal.

4. The circle detector checks for the shape of signal (which usually is circular). It accordingly gives command to move or stop. The detector has a certain range so as to avoid the errors in signal detection. In case of failure or exception the previous state is retained.

Figure- 1.0 : Sample input image for signal detection.

```
clc; clear; pkg load image;
im = imread('streetr.jpg');
b = imcrop(im,[1 1 500 500]); subplot(3,2,1); imshow(b);title("signal");
imr1 = b(:,:,1); subplot(3,2,2); imshow(imr1); title("red filter applied");

[centerl,radiil] = imfindcircles(imr1,[10 200],'ObjectPolarity','bright','sensitivity',0.75);
if 15<radiil & radiil< 30
  printf("STOP (as red filter is active--> RED signal is detected.)");
  imr2 = b(:,:,2); subplot(3,2,4); imshow(imr2); title("green filter applied")
  subplot(3,2,3)
  imshow(b): title("red signal detected");
  h = viscircles(centerl, radiil);
else
  imr2=b(:,:,2); subplot(3,2,4); imshow(imr2); title("green filter applied")

  [center2,radii2] = imfindcircles(imr2,[10 200],'ObjectPolarity','bright','sensitivity',0.75);
  if 15<radii2 & radii2<30
    printf("MOVE (as green filter is active--> GREEN signal is detected.)");
    subplot(3,2,5)
    imshow(b);title("green signal detected");
    h = viscircles(center2, radii2);
  endif
endif
```

Figure - 1.1 : Code for signal detection detects signal color and controls further movement of the car.

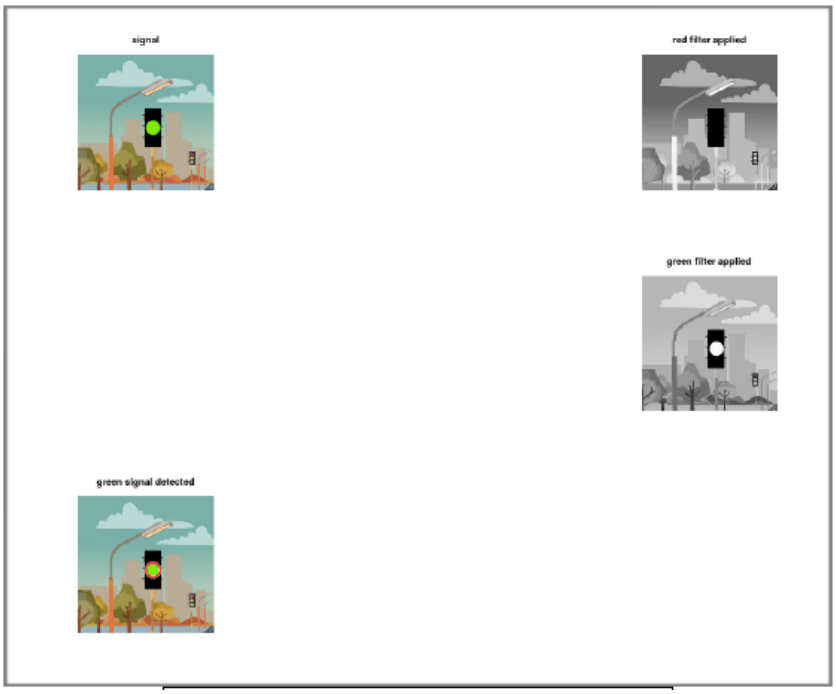


Figure-1.2: Output images for signal detection (Green signal detected).

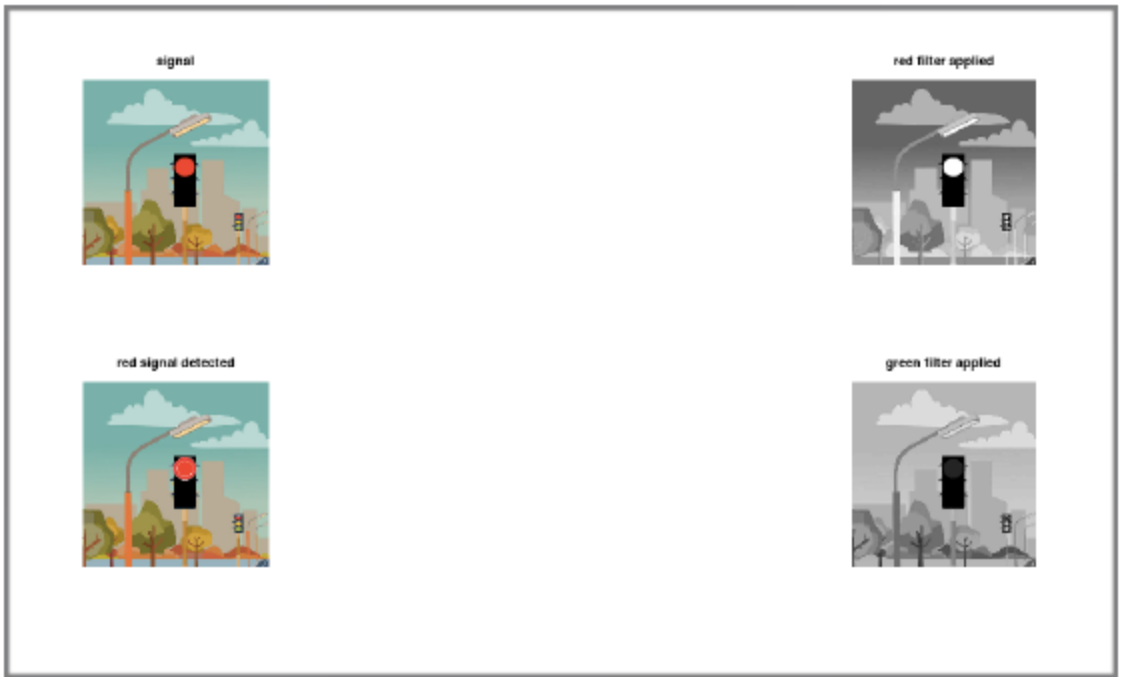


Figure-1.3: Output images for signal detection (Red signal detected).

Direct Linear Transformation (DLT):

Recording images using a camera is equivalent to mapping object point O in the object space to image point I' in the film plane (Fig.1a). For digitisation, this recorded image will be projected again to image I in the projection plane (Fig. 1b).

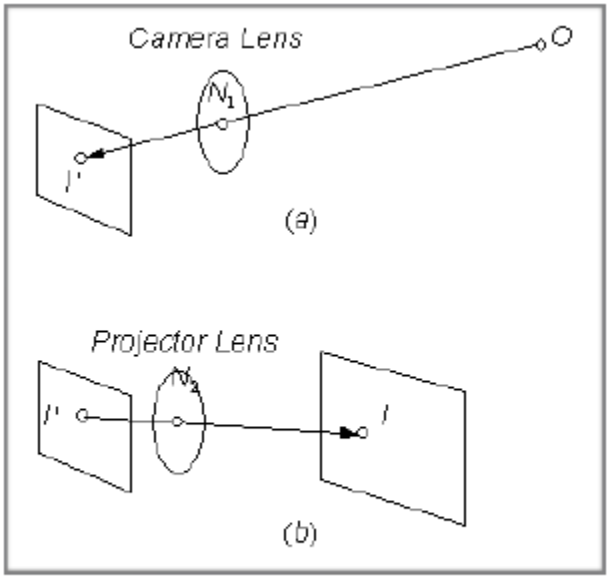


Fig. 1

But, for simplicity, it is possible to directly relate the projected image and the object (Fig.2). Object O is mapped directly to the projected image I. The projection plane is called image plane. Point N is the new node or projection center.

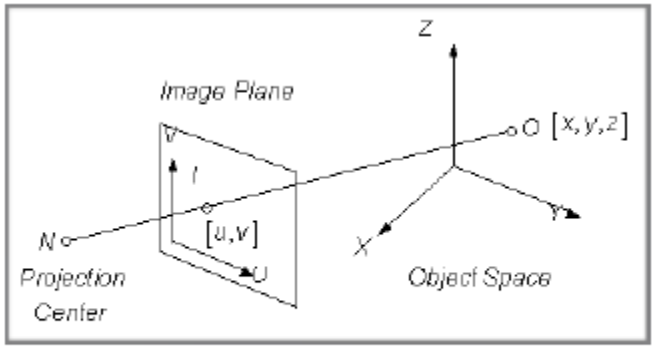


Fig. 2

Two reference frames are defined in Fig. 2: object-space reference frame (the XYZsystem) and image-plane reference frame (the UV-system). The optical system of the camera/projector maps point O in the object space to image I in the image plane. [x, y, z] is the object-space coordinates of point O while [u, v] is the image-plane coordinates of the image point I. Points I, N & O thus are collinear. This is the so-called collinearity condition, the basis of the DLT method. Now, assume that the position of the projection center (N) in the object-space reference frame to be [xo, yo, zo] (Fig. 3). Vector A drawn from N to O then becomes [x - xo, y - yo, z - zo].

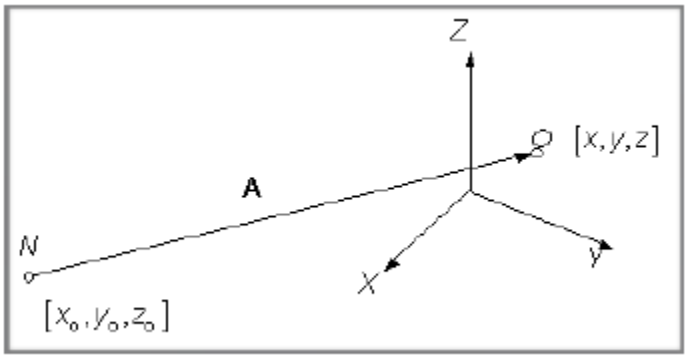


Fig. 3

Add axis W to the image plane reference frame as the third axis to make the image-plane reference frame 3-dimensional (Fig. 4). The Wcoordinates of the points on the image plane are always 0, and the 3-dimensional position of point I becomes [u, v, 0].

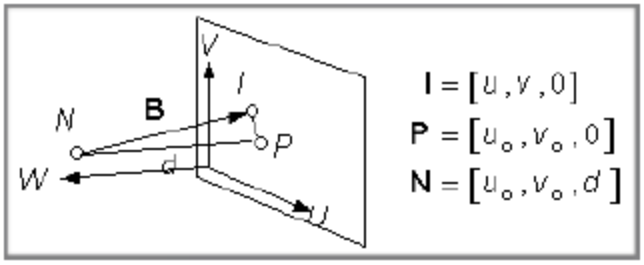


Fig. 4

A new point P, the principal point, was introduced in Fig. 4. The line drawn from the projection center N to the image plane, parallel to axis W and perpendicular to the image plane, is called the principal axis and the principal point is the intersection of the principal axis with the image plane. The principal distance d is the distance between points P and N. Assuming the image plane coordinates of the principal point to be [uo, vo, 0], the position of point N in the imageplane reference frame becomes [uo,vo,d]. Vector B drawn from point N to I is becomes [u-uo, v-vo,-d].

3-D DLT Method:

$$u = \frac{L_1X + L_2Y + L_3Z + L_4}{L_9X + L_{10}Y + L_{11}Z + 1}$$
$$v = \frac{L_5X + L_6Y + L_7Z + L_8}{L_9X + L_{10}Y + L_{11}Z + 1} \quad \text{Eq.1}$$

Eq.1 is the standard 3-D DLT equation, but one may include in Eq.1 the optical errors from the lens:

$$u - \Delta u = \frac{L_1X + L_2Y + L_3Z + L_4}{L_9X + L_{10}Y + L_{11}Z + 1}$$
$$v - \Delta v = \frac{L_5X + L_6Y + L_7Z + L_8}{L_9X + L_{10}Y + L_{11}Z + 1} \quad \text{Eq.2}$$

where [Δu,Δv] = the optical errors.Optical errors can be expressed as

$$\Delta u = \xi(L_{12}r^2 + L_{13}r^4 + L_{14}r^6) + L_{15}(r^2 + 2\xi^2) + L_{16}\xi\eta$$
$$\Delta v = \eta(L_{12}r^2 + L_{13}r^4 + L_{14}r^6) + L_{15}\eta\xi + L_{16}(r^2 + 2\eta^2)$$

Eq.3

Where,

$$[\xi, \eta] = [u - u_o, v - v_o]$$
$$r^2 = \xi^2 + \eta^2$$

Now the next step is to find the DLT parameters that is L1 to L16. To find that have to use Eq.4

$$\begin{bmatrix} \frac{x_1}{R_1} & \frac{y_1}{R_1} & \frac{z_1}{R_1} & \frac{1}{R_1} & 0 & 0 & 0 & 0 & \frac{-u_1x_1}{R_1} & \frac{-u_1y_1}{R_1} & \frac{-u_1z_1}{R_1} \\ 0 & 0 & 0 & 0 & \frac{x_1}{R_1} & \frac{y_1}{R_1} & \frac{z_1}{R_1} & \frac{1}{R_1} & \frac{-v_1x_1}{R_1} & \frac{-v_1y_1}{R_1} & \frac{-v_1z_1}{R_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{x_n}{R_n} & \frac{x_n}{R_n} & \frac{x_n}{R_n} & \frac{1}{R_n} & 0 & 0 & 0 & 0 & \frac{-u_nx_n}{R_n} & \frac{-u_ny_n}{R_n} & \frac{-u_nz_n}{R_n} \\ 0 & 0 & 0 & 0 & \frac{x_n}{R_n} & \frac{y_n}{R_n} & \frac{z_n}{R_n} & \frac{1}{R_n} & \frac{-v_nx_n}{R_n} & \frac{-v_ny_n}{R_n} & \frac{-v_nz_n}{R_n} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\xi_1 r_1^2 R_1}{R_1} & \frac{\xi_1 r_1^4 R_1}{R_1} & \frac{\xi_1 r_1^6 R_1}{R_1} & \frac{(r_1^2 + 2\xi_1^2)R_1}{R_1} & \frac{\xi_1\eta_1 R_1}{R_1} \\ \frac{\eta_1 r_1^2 R_1}{R_1} & \frac{\eta_1 r_1^4 R_1}{R_1} & \frac{\eta_1 r_1^6 R_1}{R_1} & \frac{\eta_1\xi_1 R_1}{R_1} & \frac{(r_1^2 + 2\eta_1^2)R_1}{R_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\xi_n r_n^2 R_n}{R_n} & \frac{\xi_n r_n^4 R_n}{R_n} & \frac{\xi_n r_n^6 R_n}{R_n} & \frac{(r_n^2 + 2\xi_n^2)R_n}{R_n} & \frac{\xi_n\eta_n R_n}{R_n} \\ \frac{\eta_n r_n^2 R_n}{R_n} & \frac{\eta_n r_n^4 R_n}{R_n} & \frac{\eta_n r_n^6 R_n}{R_n} & \frac{\eta_n\xi_n R_n}{R_n} & \frac{(r_n^2 + 2\eta_n^2)R_n}{R_n} \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_{15} \\ L_{16} \end{bmatrix} = \begin{bmatrix} \frac{u_1}{R_1} \\ \frac{v_1}{R_1} \\ \vdots \\ \frac{u_n}{R_n} \\ \frac{v_n}{R_n} \end{bmatrix}$$

Eq.4

The equation is applied to control points (control points are the points whose image Coordinate and 3-D object co-ordinate is known). but since we don't the value of R in the equation we have to use the following steps.

1) Find the values from L1 to L11 using the above equation.
2) Substitute the value of L9, L10,L11 in the formula of R and find the corresponding values of R for each control points.
3) After finding the values of R again use the above equation to find the parameters from L12 to L16.

2-D DLT Method:

In the case of 2-D analysis, the z-coordinate is always 0 and the mapping from the object-plane reference frame into the image-plane reference frame reduces to

$$u = \frac{L_1 x + L_2 y + L_3}{L_7 x + L_8 y + 1}$$

$$v = \frac{L_4 x + L_5 y + L_6}{L_7 x + L_8 y + 1} \quad \text{Eq.5}$$

Apply Eq.5 to n (n >= 4) control points and m (m >= 1) cameras:

$$\begin{bmatrix} \frac{x_1}{R_1} & \frac{y_1}{R_1} & \frac{1}{R_1} & 0 & 0 & 0 & \frac{-u_1x_1}{R_1} & \frac{-u_1y_1}{R_1} \\ 0 & 0 & 0 & \frac{x_1}{R_1} & \frac{y_1}{R_1} & \frac{1}{R_1} & \frac{-v_1x_1}{R_1} & \frac{-v_1y_1}{R_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{x_n}{R_n} & \frac{y_n}{R_n} & \frac{1}{R_n} & 0 & 0 & 0 & \frac{-u_nx_n}{R_n} & \frac{-u_ny_n}{R_n} \\ 0 & 0 & 0 & \frac{x_n}{R_n} & \frac{y_n}{R_n} & \frac{1}{R_n} & \frac{-v_nx_n}{R_n} & \frac{-v_ny_n}{R_n} \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_7 \\ L_8 \end{bmatrix} = \begin{bmatrix} \frac{u_1}{R_1} \\ \frac{v_1}{R_1} \\ \vdots \\ \frac{u_n}{R_n} \\ \frac{v_n}{R_n} \end{bmatrix}$$

Eq.6

And,

$$\begin{bmatrix} \frac{u^{(1)}L_7^{(1)} - L_1^{(1)}}{R_1} & \frac{u^{(1)}L_8^{(1)} - L_2^{(1)}}{R_1} \\ \frac{v^{(1)}L_7^{(1)} - L_4^{(1)}}{R_1} & \frac{v^{(1)}L_8^{(1)} - L_5^{(1)}}{R_1} \\ \vdots & \vdots \\ \frac{u^{(m)}L_7^{(m)} - L_1^{(m)}}{R_n} & \frac{u^{(m)}L_8^{(m)} - L_2^{(m)}}{R_n} \\ \frac{v^{(m)}L_7^{(m)} - L_4^{(m)}}{R_n} & \frac{v^{(m)}L_8^{(m)} - L_5^{(m)}}{R_n} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{L_3^{(1)} - u^{(1)}}{R_1} \\ \frac{L_6^{(1)} - v^{(1)}}{R_1} \\ \vdots \\ \frac{L_3^{(m)} - u^{(m)}}{R_n} \\ \frac{L_6^{(m)} - v^{(m)}}{R_n} \end{bmatrix}$$

Eq.6

Where,

$$R_i = L_7 x_i + L_8 y_i + 1$$

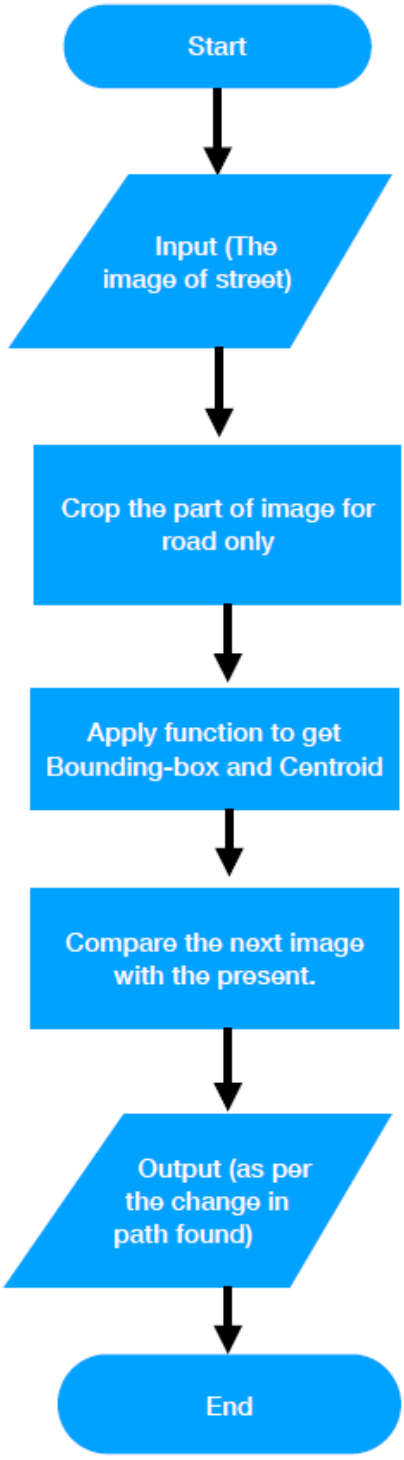
The object plane and the image plane do not have to be parallel. 2-D DLT guarantees accurate plane- to-plane mapping regardless of the orientation of the planes. The control points must not be collinear and must form a plane.

Keeping the vehicle on road:

Change in direction of motion along with the road is given in the following flow chart.

1. To keep the vehicle on road we compare the next image with the present one.
2. The centroids of the road marking are found for both the images. If the the road does not turn the centroids will essentially have same Y- coordinate (in DIP it is horizontal axis). We are bothered about
Ycoordinate only because X-coordinates don't have any role in turning the vehicle.
3. If the final comparison value turns out to be positive the past of left wheel is increased accordingly by the system. We are not going in to the details of how the tase in increased because focus here is on
just motion and a way of implementing turn. Similarly, if comparison value turns out to be negative the right wheel's pace increased by the system (depending on the magnitude of value).

Fig-1.4: Input to keep the vehicle on the road.



Flow chart: To keep the vehicle on road along the turns.

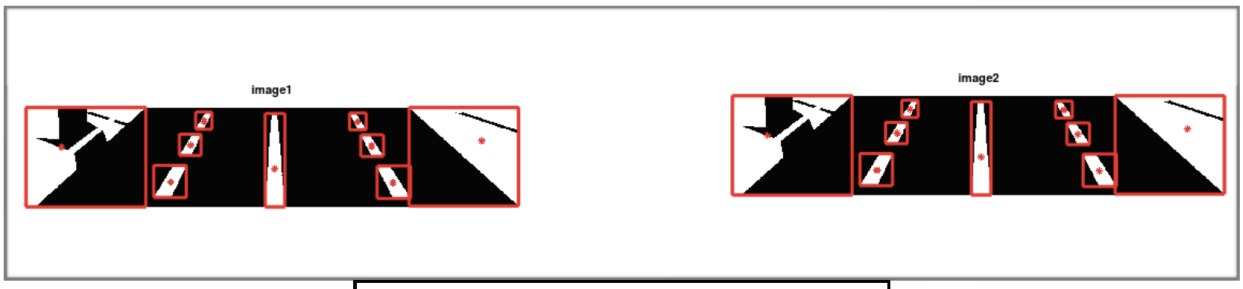


Figure-2.2: output images with bounding-box and centroids.

DLT parameter for camera-1	DLT parameter for camera-2
-0.8809	9.6580
-1.0541	-6.9954
0.1178	-16.0306
46.9417	283.0318
-1.0000	-2.1648
0.3965	1.6640
0.3332	-9.0026
31.7865	105.8744
-0.0330	0.0794
0.0005	0.0543
0.0030	0.2283
0.0000	0.0000
0.0000	0.0172
0.0010	0.0000
-0.0039	0.0000

Conclusion:

In this project we tried to find a way of building an autonomous vehicle/car using DIP. And we were successful in giving a way for checking the signal colour, a method to keep the vehicle on

road and make it turn as the road turns and finally we gave a way to find distance between objects and the camera.

Citation:

- http://www.kwon3d.com/theory/dlt/dlt.html
- Digital Image Processing-Third Edition; Rafael C. Gonzalez, University of Tennessee; Richard E. Woods, MedData Interactive