A Comprehensive Real-Time Road-Lanes Tracking Technique for Autonomous Driving

http://dx.doi.org/10.12785/ijcds/090302

summary:

laneDB Algorithm using one ccd camera Located in the windshield and gives a front view on the road.

The algorithm gets an RGB image as input, first remove the "noises" While transferring the image in the process of color changes (HSL, HSV, LAB, LUV, YUV) by pipeline process.

Next is the process of **sobel operators** (Magnitude Gradients, Absolute Gradients and Direction Gradient)

And at the end by moving from 3 dimensions to 2 dimensions creates a road image from a "top" view and finds approximately polynomials describing the shape of the path.

Methods:

Sobel operator

Multiply 3x3 matrices from the image with a Gx matrix and a Gy matrix to highlight the edges.

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \tan^{-1}\frac{G_y}{G_x}$$

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

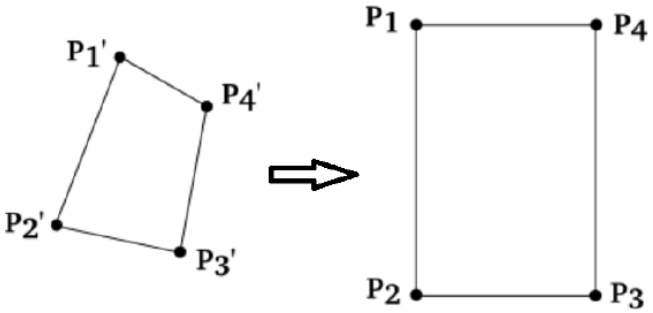
Gy

Figure 1. Sobel operator uses a 3x3 kernel mask.

Perspective transform

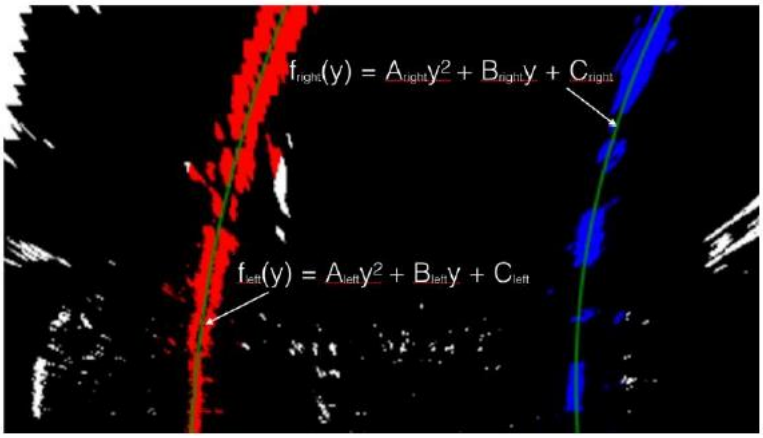
A mapping of a square represented by 4 points in 3 dimensions to a square in 2 dimensions to create for a front image of the road "view from above".

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} * \begin{bmatrix} P_1'.x \\ P_1'.y \\ 1 \end{bmatrix} = \begin{bmatrix} w * P_1.x \\ w * P_1.y \\ w * 1 \end{bmatrix} \qquad (3)$$



Measuring lane curvature

Finding a polynomial approximately approximates curved edges obtained from the 2-dimensional mapping.



Camera calibration

Adjust the data from the transition from a 3-dimensional image to a 2-dimensional image.

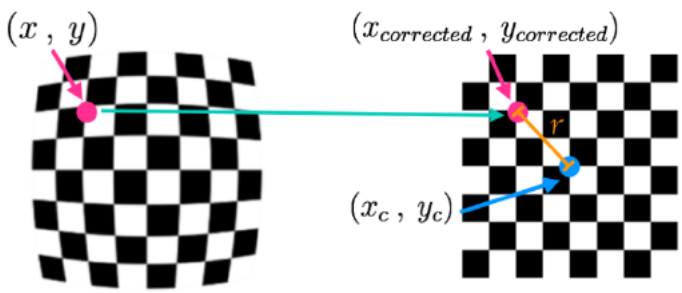


Figure 5. Points in a distorted and undistorted (corrected) images.

Figure 6. Mapping from a distorted chessboard image to an undistorted one.

Image processing pipeline

Using computer vision by the Opencv library to perform image processing in stages and analyze the relevant information at each stage.

conclusions

The article presents a reliable and sophisticated solution for locating and marking the vehicle's travel route based on computer vision and combines playing methods with the colors of the frame and algorithms such as sobel operators and perspective transform.

The calculations are relatively standard and do not require very high CPU capabilities.

In practice a large part of the processes done during the solution can be implemented simply by the opencv library.

Some processes require a mathematical understanding of some of the algorithms used to find the path.

Open source:

https://github.com/Nikhil22/python-lane-detection

summarized by: Oren Lacker

3.1.2021