

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

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

## סיכום מאמר:

### רעיון כללי-

הצגת laneDB Algorithm המשתמש במצלמת ccd אחת הממוקמת בשמשה הקדמית ונותנת תצוגה קדמית על הדרך.

האלגוריתם מקבל כקלט תמונה בפורמט RGB, מסיר ממנה תחילה את ה"רעשים" המיותרים תוך העברת התמונה בתהליך של שינויי צבעים (HSL, HSV, LAB, LUV, YUV) באופן של pipeline process.

בנוסף התמונות עוברות תהליך של **sobel operators** (Magnitude Gradients, Absolute Gradients and Direction Gradient) כדי ליצור תמונות שמבליטות את הצלעות.

ובסוף על ידי מעבר מ3 ממדים ל2 ממדים יוצר תמונת דרך ממבט "מלמעלה" ומוצא בקירוב פולינומים המתארים את צורת הנתיב.

## שיטות-

### Sobel operator •

הכפלת מטריצות  $3 \times 3$  מתוך התמונה עם מטריצה  $G_x$  ועם מטריצה  $G_y$  כדי להבליט את הצלעות.

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

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

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

$G_x$

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

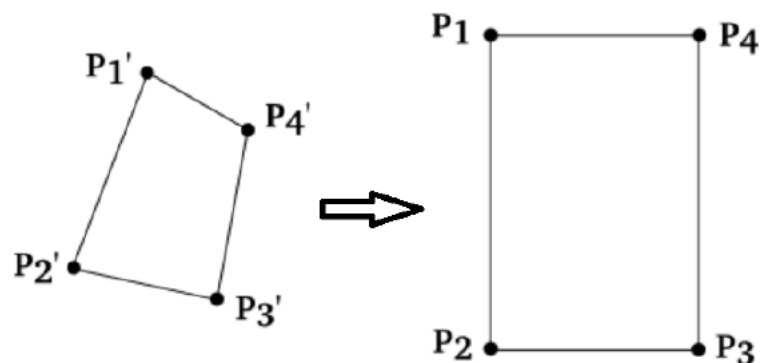
$G_y$

Figure 1. Sobel operator uses a  $3 \times 3$  kernel mask.

### Perspective transform •

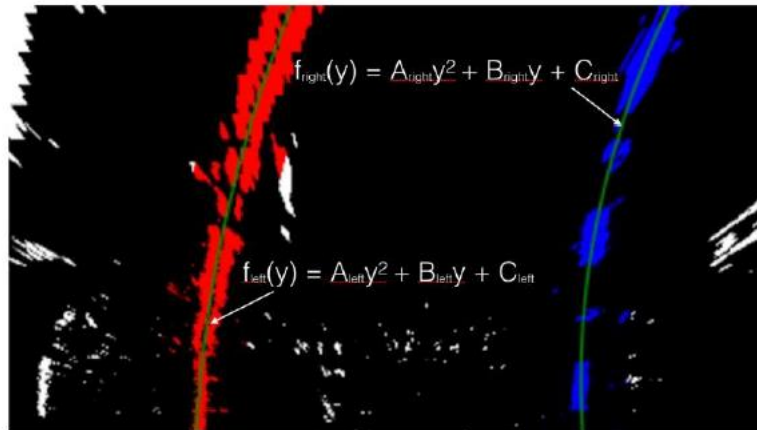
מיפוי של מרובע המיוצג על ידי 4 נקודות ב 3 ממדים ל מרובע ב 2 ממדים כדי ליצור עבור תמונה קדמית של הדרך "מבט מלמעלה".

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} * \begin{bmatrix} P'_1.x \\ P'_1.y \\ 1 \end{bmatrix} = \begin{bmatrix} w * P_1.x \\ w * P_1.y \\ w * 1 \end{bmatrix} \quad (3)$$



## • Measuring lane curvature

מציאת פולינום מתאים בקירוב לצלעות מעוקלות המתקבלות מהמיפוי ה-2 ממדי.



## • Camera calibration

התאמה של הנתונים מהמעבר מתמונה 3 ממדית לתמונה 2 ממדית.

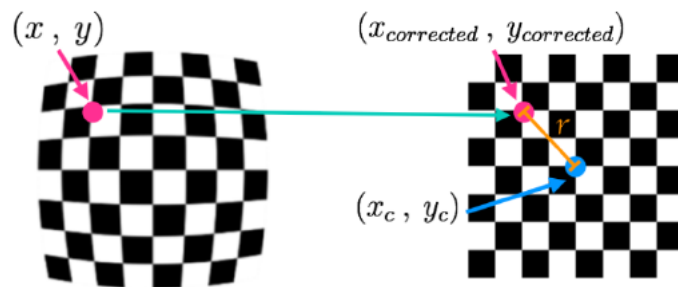


Figure 5. Points in a distorted and undistorted (corrected) images.



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

## • Image processing pipeline

שימוש ב computer vision על ידי הספרייה OpenCV כדי לבצע עיבוד תמונה בשלבים ולנתח בכל שלב את המידע הרלוונטי.

### מסקנות-

המאמר מציג פתרון אמין ומתוחכם לאיתור וסימון נתיב הנסיעה של הרכב המבוסס על computer vision ומשלב שיטות משחק עם הצבעים של הפריים ואלגוריתמים כמו sobel operators ו perspective transform. החישובים יחסית סטנדרטיים ולא דורשים יכולות מעבד מאוד גבוהות. מבחינה מעשית חלק גדול מהתהליכים שנעשים במהלך הפתרון ניתנים למימוש פשוט על ידי הספרייה opencv. חלק מהתהליכים דורשים הבנה מתמטית של כמה אלגוריתמים המשמשים למציאת הנתיב.

Open source של פרויקט דומה:

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