



LANE AND MARKINGS DETECTION IN COLOR IMAGES

Razvan Pop

- Identificarea benzilor de circulație pe drum este o sarcină obișnuită efectuată de toți șoferii umani pentru a se asigura că vehiculele lor se încadrează în limitele benzii de circulație atunci când conduc, astfel încât să se asigure că traficul este lin și să se reducă la minimum șansele de coliziune cu alte mașini din cauza nealinierii benzii.



Pașii de realizare a proiectului

- Izolarea galbenului și albului din imaginea HSL;
- Convertirea imaginii color într-o imagine grayscale;
- Reducerea zgomotului folosind un filtru Gaussian;
- Identificarea muchiilor din imagine folosind algoritmul Canny;
- Găsirea liniilor folosind Transformata Hough;
- Medierea și extrapolarea liniilor (Interpolarea gradientului și extrapolarea liniei);
- Identificarea regiunii de interes;

Selectia albului si a galbenului

- Mai întâi, se izolează galbenul și albul de imaginea originală. După aceasta, se poate observa cum galbenul și albul benzilor sunt foarte bine izolate.

```
bool isAllWhite(int a, int b, int c)
{
    if (a == 255 && b == 255 && c == 255)
    {
        return true;
    }
    return false;
}
```

```
bool isYellow(int a, int b, int c)
{
    if (a == 255 && b == 255 && c == 0)
    {
        return true;
    }
    return false;
}
```

Convertirea imaginii color intr-o imagine grayscale

- Suntem interesați să detectăm linii albe sau galbene pe imagini, care prezintă un contrast deosebit de ridicat atunci când imaginea este în tonuri de gri. Cum drumul este negru, astfel încât orice lucru care este mult mai luminos pe drum va ieși cu un contrast ridicat într-o imagine în tonuri de gri.



Reducerea zgomotului folosind un filtru Gaussian

- Facem acest pas deoarece vrem să ne concentrăm doar pe cele mai semnificative linii (cele pe bandă), nu pe cele de pe fiecare obiect. Trebuie să fim atenți pentru a nu estompa prea mult imaginile, altfel va deveni greu să alcătuim o linie.



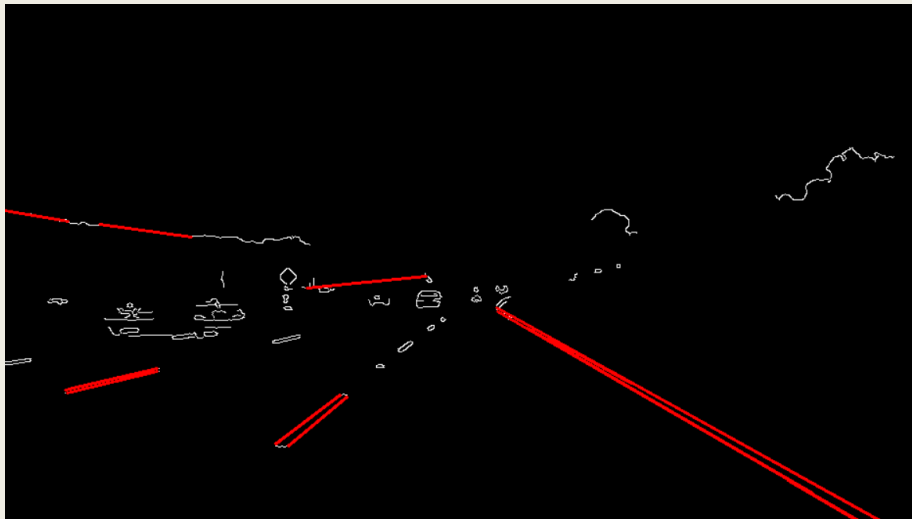
Identificarea muchiilor din imagine folosind algoritmul Canny

- După prelucrarea imaginii, putem aplica un Canny Edge Detector care are rolul de a identifica marginile dintr-o imagine și de a neglija toate celelalte date. Imaginea rezultată ajunge să fie neobișnuită, ceea ce ne permite să ne concentrăm și mai mult pe detectarea benzii, deoarece ne preocupă liniile.



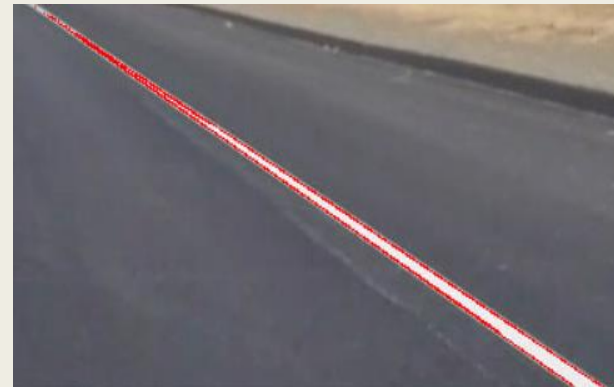
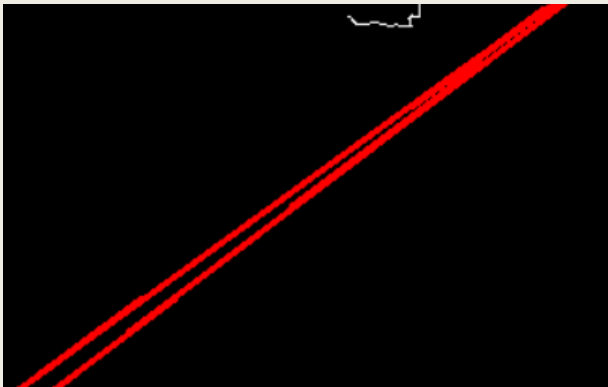
Găsirea liniilor folosind Transformata Hough

- Se poate folosi Hough Line Transform pentru a detecta liniile din imaginile de pe margine.
- Scopul Hough Transform este de a găsi linii prin identificarea tuturor punctelor care stau pe ele.



Medierea și extrapolarea liniilor (Interpolarea gradientului și extrapolarea liniei)

- Pentru a urmări o linie completă din partea de jos a ecranului până în cel mai înalt punct al regiunii noastre de interes, trebuie să putem interpola diferitele puncte returnate de funcția noastră de transformare Hough și să găsim o linie care să minimizeze distanța dintre acele puncte. Vom încerca să găsim linia pe o bandă dată prin minimizarea erorii celor mai mici pătrate.



Identificarea regiunii de interes

- Când găsim linii de bandă, nu este nevoie să verificăm cerul și dealurile.
- O presupunere crucială în această sarcină este că camera rămâne în același loc în toate aceste imagini, iar benzile sunt plate, prin urmare putem identifica regiunea critică care ne interesează.



Concluzii

- Am realizat un algoritm pentru detecția și marcarea liniilor. Nu am folosit niciun model sau caracteristici complexe ale imaginii. În schimb, soluția noastră s-a bazat pe anumite operațiuni de pre-procesare a imaginilor. Cu toate acestea, vor exista multe scenarii în care această soluție nu va funcționa. De exemplu, atunci când nu vor exista marcaje de bandă sau când traficul este prea mare pe drum, acest sistem va eșua. Există metode mai sofisticate pentru a depăși astfel de probleme în detectarea benzii.

Bibliografie

- http://www.iaeng.org/publication/WCECS2011/WCECS2011_pp499-502.pdf
- <https://medium.com/@naokishibuya/finding-lane-lines-on-the-road-30cf016a1165>
- <https://edforson.me/posts/using-computer-vision-to-find-lane-lines-on-road/>
- <https://janhalozan.com/2019/06/01/lane-detector/>
- <https://www.analyticsvidhya.com/blog/2020/05/tutorial-real-time-lane-detection-opencv/>