

Portfolio Beeldverwerking 4

Bosmans Stig

Probleemstelling

Is het mogelijk om via Matlab aan Nummerplaatherkenning te doen? Is het dan mogelijk om dit toe te passen op nummerplaten van verschillende landen?

In dit verslag wordt besproken in hoeverre dit realiseerbaar is doormiddel van Matlab. In het dagelijks verkeer is dit een probleem dat opgelost moet worden. Zeker omdat tussen verschillende landen de vorm, lettertype en kleur vaak anders zijn.

Literatuurstudie

Het eerste verslag dat we bespreken is: License platelocalization in camera pictures (paper 1)

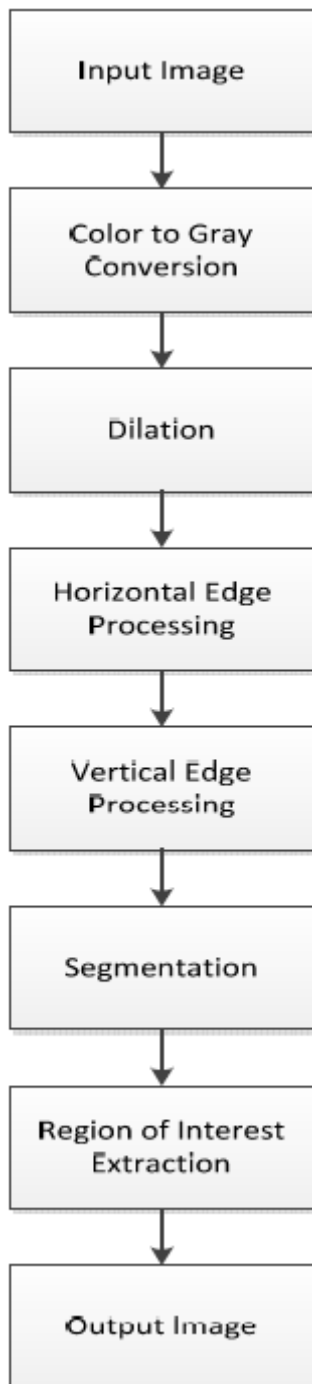
Dit is van Halina Halina Kwaśnicka en Bartosz Wawrzyniak (Wrocław University of Technology)

Eerst wordt er gehandeld over de condities van de nummerplaat: weer, licht, plaats van nummerplaat, beweging, schade en andere vormen van letters en cijfers in het beeld. Meeste van deze punten zou kunnen opgelost worden met het verbeteren van het licht inval, betere camera's en hoogstaande technieken van image processing.

Gebruik maken van algoritmes met vorm en kleur zou inefficiënt zijn, omdat er veel verschillende nummerplaten zijn. De juiste manier om Nummerplaat herkenning te doen is om het verschil van achtergrond kleur en letterkleur te gebruiken, dit is bij elke nummerplaat zeer duidelijk.

De tweede paper is van Naikur Bharat kumar Gohil genaamd Car License PlateDetection. Hier wordt de volledige implementatie binnen in Matlab code besproken.

Hier word als volgt gewerkt:



Color to gray conversion: als eerste word de afbeelding omgezet naar een grijswaarde afbeelding. Zo is het makkelijk om verschillende typen nummerplaten af te lezen

Dilation: hierbij wordt de afbeelding duidelijker, gaten worden opgevuld, hoeken zijn scherper en de ruis wordt verwijderd. Verschillen tussen donker en licht worden ook duidelijker.

Horizontal and Vertical Edge Processing: elke kolom en rij van de afbeelding wordt doorlopen door het algoritme, hiermee wordt het verschil in pixel waarde tussen naburige pixels berekend. Als het verschil in grijswaarde tussen de opeenvolgende pixels een bepaalde grootte grenswaarde overschrijdt, tellen we die op bij het totaal van die bepaalde kolom.

Segmentation: hier worden alle regio's in de afbeelding die een nummerplaat kunnen bevatten gezocht. De plaatste hiervan wordt opgeslagen in een array.

Region of interest extraction: hier wordt gebruikt gemaakt van de Edge processing, de regio met de hoogste histogram waarde van verticaal en horizontaal zal de nummerplaat bevatten.

Nummerplaat herkenning

2.1 Inleiding

Voor het herkennen van een nummerplaat gaan we in verschillende stappen te werk:

1. Afbeelding converteren naar binaire afbeelding (2.2)
2. Randen detecteren (2.2)
3. Edge processing en positie bepalen (2.3)
4. Segmentatie nummerplaat (2.3)
5. Optical Character recognition (2.5)

2.2 Edge detection

Om onnodige informatie te wissen uit de afbeelding, hoeven we enkel met de randen van de afbeelding te werken hierdoor volgen we de werkwijze beschreven in paper 3. Voor het detecteren van de randen maken we gebruik van een ingebouwde matlab functie. Maar allereerst zetten we de originele afbeelding om naar een grayscale afbeelding, hierdoor kunnen we matlab een threshold laten berekenen voor de verschillende intensiteiten in de grayscale afbeelding, alles boven die bepaalde threshold wordt wit en alles onder de threshold zwart. We hebben dus als het ware een binaire afbeelding, daarop wordt tot slot het edgedetection algoritme toegepast. In figuur 2.2.1 kan je het resultaat duidelijk waarnemen.

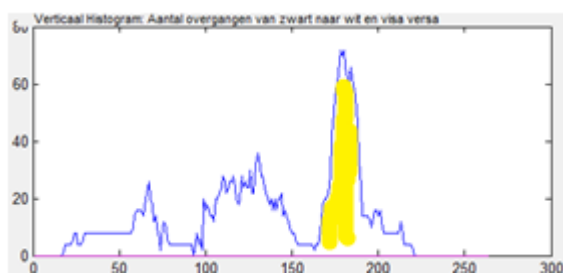


figuur 2.2.1

2.3 Edge Processing

Dankzij paper 2 (bijlage) zijn we gebruik gaan maken van horizontal en vertical edge processing, daarbij zetten we de afbeelding om naar grijswaarde, als het verschil in grijswaarde tussen de opeenvolgende pixels een bepaalde grenswaarde overschrijdt, tellen we die op bij het totaal van die bepaalde kolom.

Uiteindelijk worden alle kolomtotalen opgeslagen in een matrix. Als we die matrix zouden plotten krijgen we een resultaat zoals bij figuur 2.3.1. Hieruit is in veel gevallen de positie van de nummerplaat af te leiden, in het geval van figuur 2.3.1 is dat de geel gemarkeerde regio.



figuur 2.3.1 (x-as: positie van rij pixels op de afbeelding, y-as: aantal keer het verschil in grijswaarde tussen naburige pixels een grenswaarde overschrijdt)

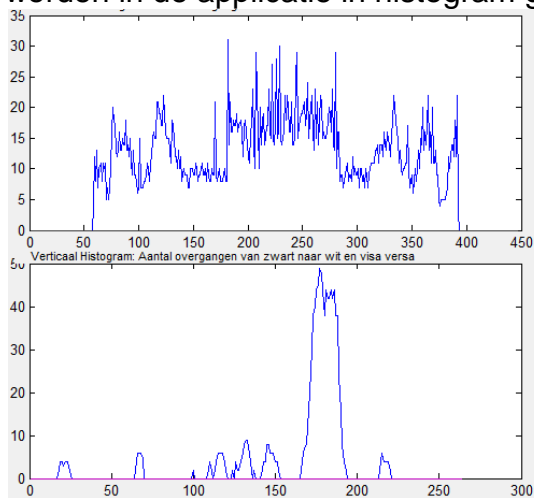
In paper 3 (bijlage) werd echter gebruik gemaakt van edgedetection (zoals in 2.2 vermeld is), en werd bij horizontaledge processing enkel het aantal witte pixels geteld.

Uiteindelijk komt het op hetzelfde neer als bij de eerste methode, maar het brengt toch vele voordelen met zich mee. Namelijk dat er een edge-detection functie is ingebouwd in matlab waarbij je verschillende methodes kan hanteren en vergelijken. Na uitgebreid testen hebben we gekozen voor de roberts-methode, deze leek voor

onze applicatie het beste te zijn.

Het detecteren van de juiste regio gaat bij horizontaal edge processing als volgt: als het aantal witte pixels (een edge is wit) per kolom gedurende een lange tijd boven een op voorhand berekende threshold ligt gaan we ervan uit die bepaalde regio de horizontale positie van de nummerplaat aanduidt. In het geval er meerdere mogelijke regio's gekozen worden dan gaan we verder werken met de breedste.

Voor verticaledge processing wordt dezelfde methode gehanteerd enkel zoeken we nu niet naar het aantal witte pixels per kolom maar het aantal keer een pixel tov van zijn buur in een rij van hoog naar laag gaat of omgekeerd. Bij de rijen van een nummerplaat zullen er namelijk veel verschillende naburige pixels zijn. De resultaten worden in de applicatie in histogram geplot (zie figuur 2.3.2).



Figuur 2.3.2 (x-as: positie rij op de afbeelding, y-as bovenste grafiek: aantal getelde witte pixels per rij, x-as onderste grafiek: aantal getelde overgangen (1->0,0->1) per rij)

Tot slot gebruiken we de berekende posities voor het segmenteren van de afbeelding. Met figuur 2.3.3 als resultaat.



Figuur 2.3.3

2.4 Beïnvloeding door de omgeving

Probleemstelling

De gehanteerde EdgeDetection methode brengt verschillende problemen met zich mee, waaronder dat randen van de omgeving buiten de auto ook weergegeven zullen worden. Daardoor werkt het gehele algoritme dikwijls niet meer. Een voorbeeld daarvan zie je in onderstaande figuren 2.4.1 en 2.4.2. In figuur 2.4.1 wordt er gebruik gemaakt van een Mercedes, en rond de Mercedes zijn duidelijk verschillende omgevingsfactoren zichtbaar. Dit heeft als resultaat dat na het Edgedetection algoritme naast de randen van de auto ook veel randen van de omgeving worden weergegeven. In het histogram is het vervolgens zeer moeilijk om de positie van de nummerplaat terug te vinden, het resultaat van de detectie van de nummerplaat is in dit geval dan ook niet goed. In figuur 2.4.2 gebruiken we echter dezelfde foto maar

zonder de omgevingsfactoren, het resultaat van de detectie is in dit geval zo goed als perfect.

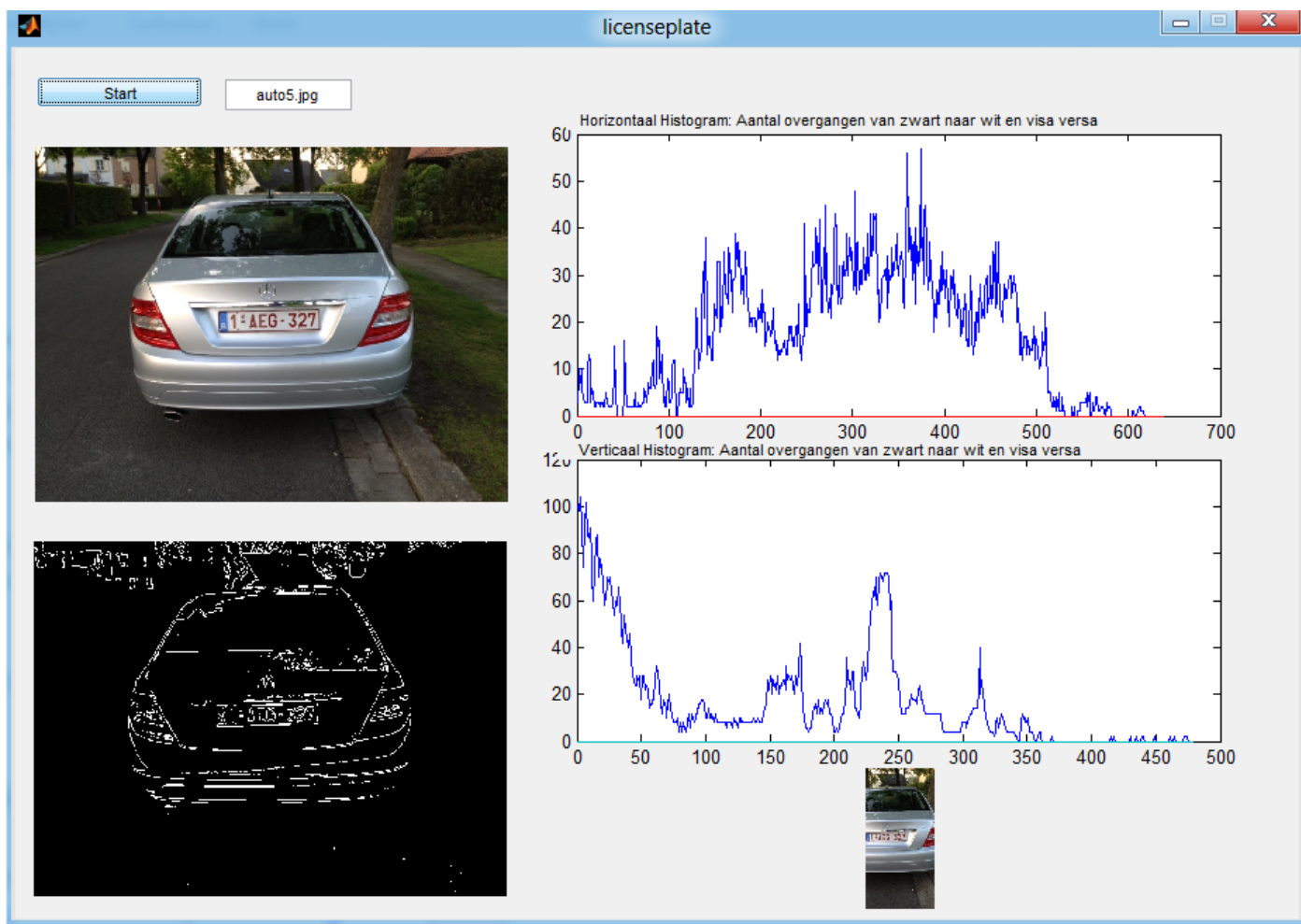
Oplossing

De oplossing moet in dit geval niet uitgevoerd worden in het programma maar eerder in de toepassing van het programma.

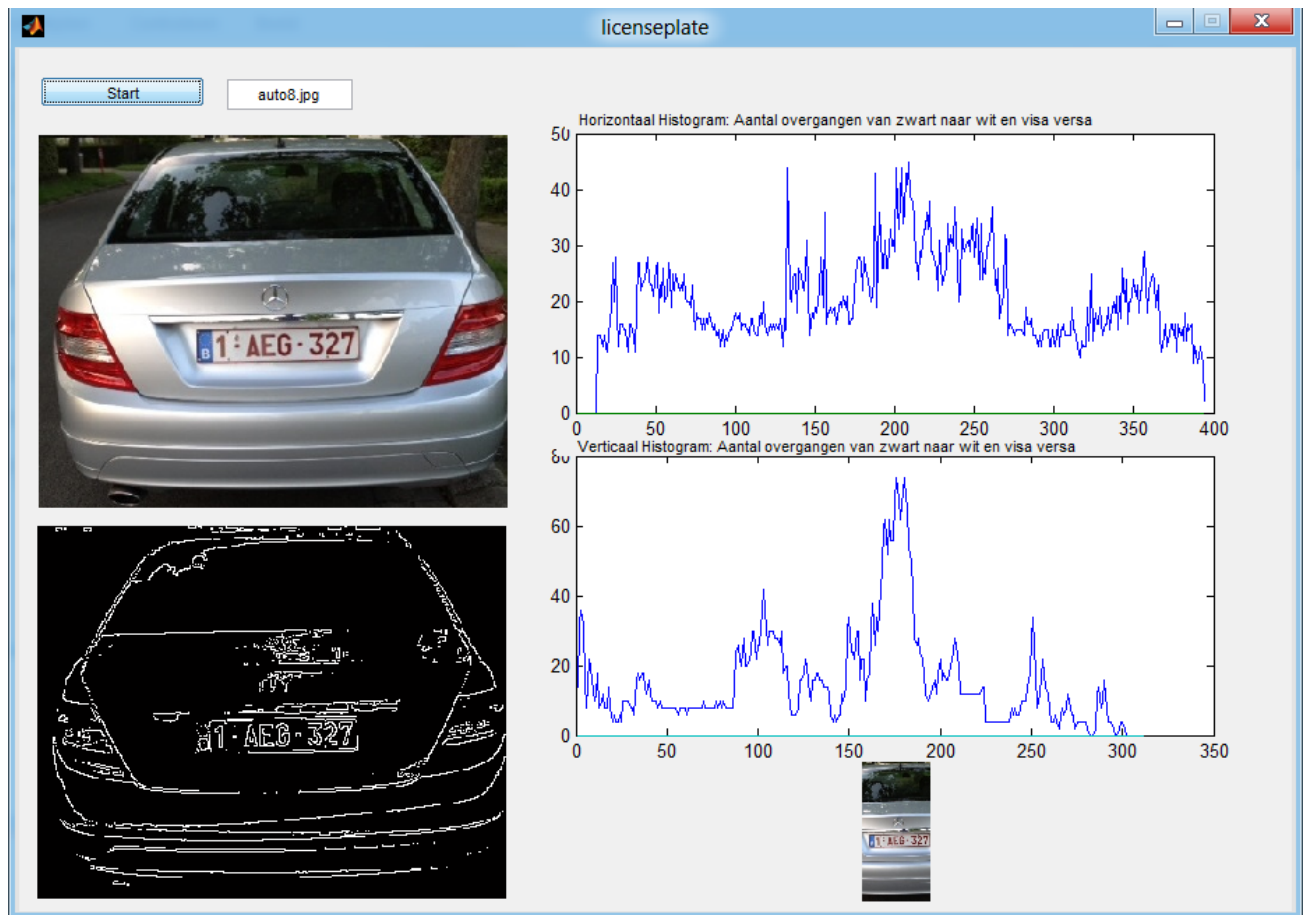
Bij de toepassing van een ALPR-systeem (Automatic License Plate Recognition) kan de camera zo geplaatst worden, dat enkel foto's van het bepaalde rijvak verwerkt moeten worden. De hoek van de camera kan er ook voor zorgen dat de horizon niet zichtbaar is.

Dit zal als resultaat hebben dat de enige omgevingsfactor naast de auto zelf, de rijstrook zal zijn.

En die rijstrook zal in de meeste gevallen volledig weggefilterd worden door het Edge-detection algoritme.



Figuur2.4.1



figuur2.4.2

2.5 Optical Character Recognition adhv template matching

Ondanks dit niet opgegeven was in onze specificatie zijn we nog een stap verder gegaan door het implementeren van een optical characterrecognition algoritme. De implementatie van het door ons gebruikte ocr algoritme wordt hieronder in stappen toegelicht:

1. We krijgen een reeds gesegmenteerde nummerplaat binnen in de ocr functie (figuur 2.5.1).
2. Deze afbeelding wordt omgezet naar grayscale voor het berekenen van een threshold (figuur 2.5.2).
3. Adhv de threshold wordt afbeelding omgezet naar een binaire afbeelding (figuur 2.5.3).
4. Alle geconnecteerde regio's worden gesegmenteerd.
5. Het gemiddelde aantal zwarte pixels per regio wordt berekend als threshold.
6. Als het aantal zwarte pixels per regio kleiner is dan de threshold (gemiddelde) wordt deze verwijderd.
7. Alle regio's waarbij de hoogte kleiner is dan de breedte (x10%) worden verwijderd. Het uiteindelijke resultaat zie je in figuur 2.5.4
8. Tot slot vergelijken we elke overgebleven regio met elke letter uit de database (figuur 2.5.5), die vergelijking gebeurt adhv een logische AND, de regio met het meest overeenkomende zwarte pixels wordt beschouwd als die letter.
9. Elke gevonden letter wordt bijgevoegd bij een 'totaal-string'.

Let op de geïmplementeerde OCR werkt enkel voor Duitse nummerplaten omdat we enkel een Duitse database van nummerplaatletters hebben gevonden.



figuur 2.5.1



figuur 2.5.2



figuur 2.5.3



figuur 2.5.4



figuur 2.5.5

Besluit

Nummerplaat herkenning d.m.v. Matlab is mogelijk. De grootste problemen dat er bij de implementatie zijn: de verschillende soorten nummerplaten in de wereld, het mogelijk uit filteren van ruis, de locatie van nummerplaten en de afstand/leesbaarheid van de nummerplaat.

Door verschillende algoritmes toe te passen, is het mogelijk om de nummerplaat uit de afbeelding te halen en de letters te vergelijken met de letters uit een database.

Bijlage

Licensplate.fig : GUI

Licensplate.m : Code voor detecteren nummerplaat

Ocr.m : Optical Characterrecognition algoritme

Demo.avi : filmpje van de werking

/images (map) : map met testafbeeldingen

/images/ocr(map) : afbeeldingen van letters voor ocr

/literatuur (map) : Verschillende papers (zie bronnen)

Bronnen

Paper 1:

HalinaKwaśnicka and BartoszWawrzyniak. (15/11/2002) License plate localization and recognition in camera pictures; 4 blz, beschikbaar op:

<http://www.ci.pwr.wroc.pl/~kwasnick/download/kwasnickawawrzyniak.pdf> [26/4/2012]

Paper 2:

NaikurBharatkumarGohil (2006) Car License Plate Detection; 72 blz,
beschikbaar op: [http://csus-
dspace.calstate.edu/bitstream/handle/10211.9/1009/Naikur_Gohil_Project_Report.pdf?sequence=3](http://csus-dspace.calstate.edu/bitstream/handle/10211.9/1009/Naikur_Gohil_Project_Report.pdf?sequence=3)

Paper 3:

Nicole Ketelaars (April 2002) Final Project: Automated License
PlateRecognition; 4blz, beschikbaar op:
http://www.win.tue.nl/aime/Files/apr2002_license.pdf