

Practicum Vision: Face Recognition

Practicum onderdeel #2: Lokalisatie

Auteurs:

Alexander Hustinx

Rolf Smit

Versie:

1.0

Cursus:

Vision (TCTI-V2VISN1-13)

Practicum onderdeel #2: Lokalisatie

In het practicum onderdeel #2 zal er gewerkt worden aan de lokalisatie van gezichtskenmerken voor in de *face recognition* applicatie. Er zijn deze week meerdere opdrachten beschikbaar waarvan de student er maar één hoeft uit te voeren. Na het toevoegen van de door de student gemaakte code zal het via de GUI getest kunnen worden en zal het moeten resulteren in een vergelijkbaar of beter resultaat dan verkregen met de aangeleverde code. In dit document zal er duidelijk gemaakt worden wat de leerdoelen van dit onderdeel van het practicum zijn en hoe deze behaald dienen te worden. Daarna zal er beschreven worden hoe het practicum beoordeeld gaat worden en hoe en wanneer de student het practicum dient in te leveren. Als laatst wordt de opdracht en hoe de student te werk moet gaan in detail beschreven.

Leerdoelen

De leerdoelen voor dit onderdeel van het practicum zijn het ontdekken waar en hoe histogrammen gebruikt kunnen worden, ontdekken van het nut van histogrammen bij *object recognition* en afhankelijk van de gekozen opdracht *morphology* ontdekken. Ook zullen de studenten vision software leren ontwerpen en implementeren. Deze leerdoelen zullen behaald worden door het toepassen of uitvoeren van de volgende algoritmen, visionstechnieken en werkzaamheden:

- Het uitvoeren van één (of meer) van de vijf lokalisatie opdrachten;
- Het gebruik van histogrammen op verschillende assen en voor verschillende doeleinden;
- Het onderbouwen van gemaakte keuzes in het lokaliseren van een gezichtskenmerk;

Hoe deze punten helpen met het behalen van de genoemde leerdoelen is doordat studenten tijdens de uitvoering van de opdracht(en) gebruik moeten maken van histogrammen met als doel om objecten te herkennen/lokalisieren. En omdat de studenten een ontwerp of plan moeten maken hoe ze een gezichtskenmerk willen lokaliseren en de gemaakte keuzes moeten onderbouwen.

Beoordelings- en inlevermethoden

Dit onderdeel van het practicum wordt beoordeeld aan de hand van de volgende punten:

- Snelheid;
- Memory efficiency;
- Robuustheid;
- Volledigheid;
- Gekozen plan
- Extra functionaliteiten.

Hoe meer van deze punten behaald/voldaan zijn, hoe hoger de resulterende beoordeling zal zijn. Dit is met uitzondering van plagiaat. Plagiaat resulteert in een NVD of een 1.

Er dienen een aantal (zelf te bepalen, maar meer dan één) meetrappen te worden gemaakt die aantonen dat de code en oplossingen daadwerkelijk werken. Daarbij kan gedacht worden aan snelheidsmetingen, geheugengebruik, vergelijk met de 'base'-implementaties, etc. Deze rapporten dienen in PDF formaat te worden opgeslagen in de 'meetrappen' folder van de repository. Bronbestanden voor deze rapporten kunnen in de 'working' folder worden opgeslagen, NIET in de 'meetrappen' hoofd-folder.

Vergeet niet om het implementatieplan te schrijven en deze in de folder 'implementatieplan' op te slaan!!

Practicumopdracht(en)

In deze paragraaf wordt behandeld wat de practicumopdracht precies is en wat er van de student verwacht wordt.

Dit onderdeel van het practicum bestaat uit een **keuze** uit vijf opdrachten:

1. Het lokaliseren van de bovenkant van het hoofd en de linker en rechter zijkant;
2. Het lokaliseren van de neus, mond en kin;
3. Het lokaliseren van de kincontouren;
4. Het lokaliseren van de linker en rechter neusvleugel, deel van de wang en deel van de ogen;
5. Het lokaliseren van de ogen.

Ook bestaat het practicum uit het beschrijven van de gemaakte keuzes en het plan voor de lokalisatie.

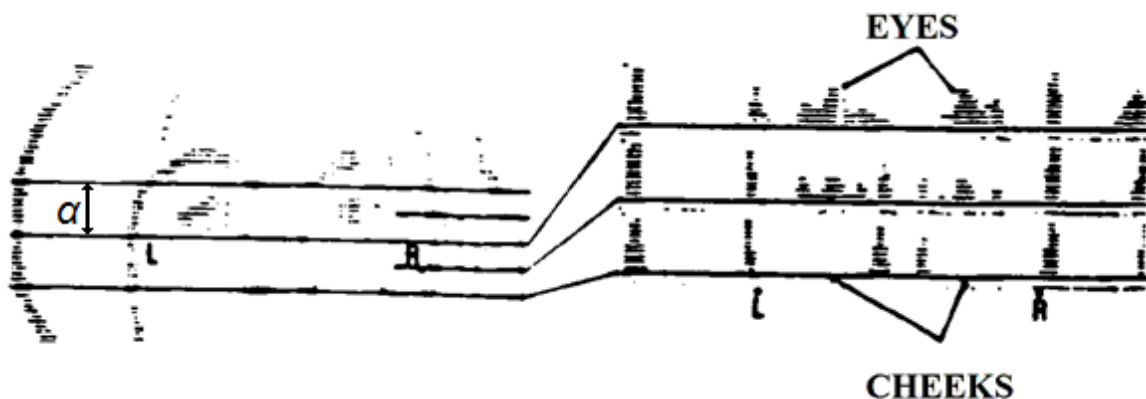
De opdrachten worden hieronder beschreven.

1. Het lokaliseren van de bovenkant van het hoofd en de linker en rechter zijkant

Het lokaliseren van de bovenkant van het hoofd (y-as) en de linker en rechter zijkant van het hoofd (x-as) is de allereerste stap van deze gezichtsherkenning. Deze stap moet als eerst uitgevoerd worden om te bepalen vanaf waar er gezocht gaat worden in het gezicht, binnen de bovenkant, linker en rechter zijkant dus.

Voor deze opdracht zal de student meerdere histogrammen moeten maken: één histogram van de gehele y-as en meerdere histogram van de x-as over de gehele breedte, maar op verschillende hoogten. Aan de hand van deze histogrammen kan je, door te zoeken naar zwarte pixels, de bovenkant en zijkanen bepalen.

Het is belangrijk dat er vrijwel altijd op dezelfde plekken de 'zijkanen' gevonden worden. Hierom moet er gezocht worden naar de wangen. De wangen zijn te herkennen aan het feit dat het (vrij grote) meestal lege/witte ruimten zijn onder de ogen, die tussen de zijkant van het hoofd en de neus liggen. Zie Figuur 1 voor meer informatie.



Figuur 1. (links) deel van het hoofd na pre-processing. (rechts) histogrammen van horizontale segmenten van de afbeelding. In de histogrammen is te herkennen, aan de hand van de hoogte and hoeveelheid pixels, waar de ogen en wangen (ongeveer) zitten. Hierin is L de linker zijkant, R de rechter zijkant en α de hoogte van een segment.

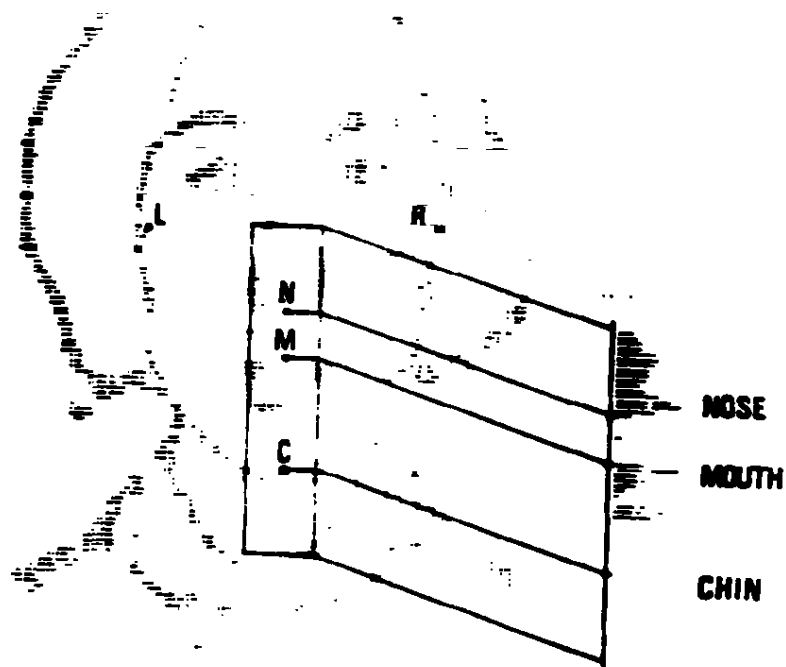
Let op de overlap van de histogrammen in Figuur 1!

2. Het lokaliseren van de neus, mond en kin

Het lokaliseren van de onderkant van de neus(N), de bovenkant, het middenpunt en de onderkant van de mond(M) en de onderkant van de kin(C), is de tweede stap die uitgevoerd wordt bij de lokalisatie. Deze stap is belangrijk voor de rest van de uit te voeren stappen in het gehele proces, want er wordt veel vanaf deze punten begonnen met zoeken.

Voor deze opdracht zal er gebruik gemaakt worden van één histogram waarin er gezocht wordt naar pieken, om aan de hand daarvan te bepalen waar welke gezichtskenmerk begint en ophoudt.

Dit wordt volgens de gebruikte methode gedaan door een segment-breedte te pakken van $\frac{1}{4}$ van de breedte van het hoofd. Zie Figuur 2 voor meer informatie.



Figuur 2. Figuur van het hoofd na pre -processing en na de bepaling van de linker en rechter zijkanten. Weergegeven is ee segment van (hoofd breedte /4) en een voorbeeld van de gevonden pieken in de histogram.

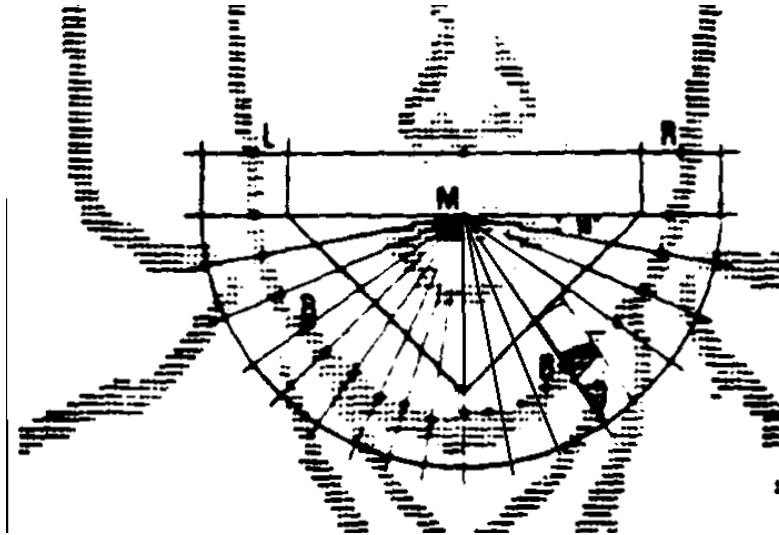
Let op dat een kleine dai in de histogram er niet voor zorgt dat hij opeens iets gevonden, of juist niet gevonden, heeft!

3. Het lokaliseren van de kincontouren

Het lokaliseren van de kincontouren is de derde stap die uitgevoerd wordt bij de lokalisatie. In deze stap worden er 19 punten op de kin gevonden om zo de contouren vast te stellen. De contouren worden later, bij de representatie, gebruikt.

Voor deze opdracht zal er per $^{\circ}10$ een 'lijn' getrokken worden vanaf de mond richting de kin. Aan het einde zouden er dus in principe 19 punten gevonden kunnen worden ($^{\circ}180$), maar meestal is dit niet het geval omdat er wel eens een deel van de kin weg valt tijdens de pre-processing. In dit geval worden de ontbrekende punten weggelaten (of geschat).

Zie Figuur 3 voor meer informatie.



Figuur 3. Proces waarin gezocht wordt naar de contouren van de kin. M is de bovenkant van de mond, B is het punt het dichtste bij M gevonden op de lijn.

Let op dat er ook punten gevonden kunnen worden in de mond, dit is niet de bedoeling. Het is een optie om pas vanaf een bepaalde afstand vanaf de mond te gaan zoeken.

4. Het lokaliseren van de linker en rechter neusvleugel, deel van de wang en deel van de ogen

Het lokaliseren van de linker (P) en rechter neusvleugel (Q), deel van de wang en deel van de ogen is de vierde stap die uitgevoerd moet worden bij het lokaliseren. Deze stap is van belang omdat de linker en rechter neusvleugels en het gedeelte van de wangen gebruikt worden bij de berekening van de representatie van het gezicht, en omdat een deel van de ogen vinden al de basis is voor de volgende stap.

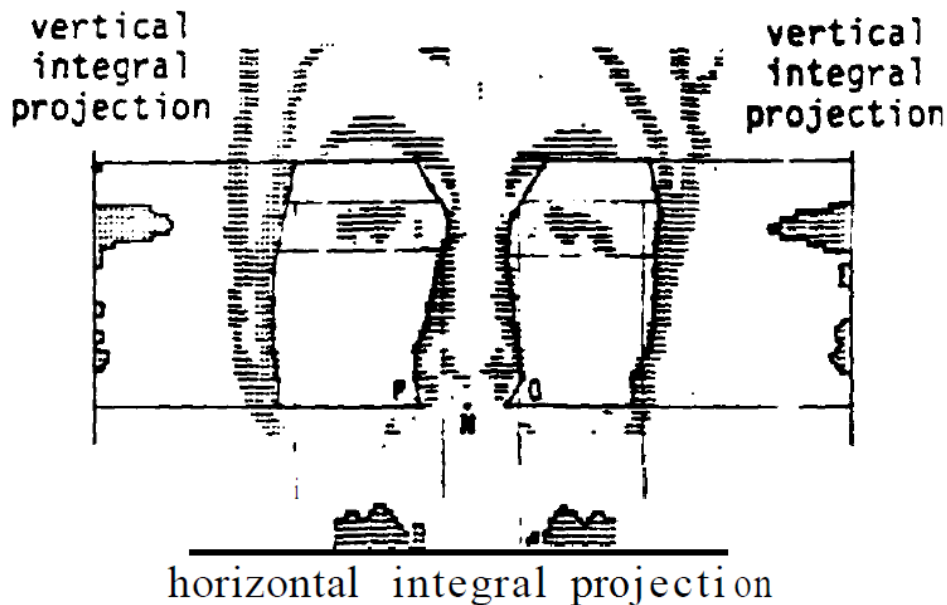
Voor deze opdracht zal er op de hoogte van de neus (N) gezocht worden naar de linker en rechter wang, dit kan bv. door middel van een histogram. Vervolgens zal er vanaf N naar boven gezocht worden door middel van histogrammen. Tijdens het naar boven gaan zullen de breedste delen links en rechts van de onderkant van de neus gelabeld worden als P en Q, hiermee worden de neusvleugels aangegeven.

Ook zal er bij deze opdracht gezocht worden naar de onderkant van het oog, met een breedte die (ongeveer) klopt. Een formule die gebruikt kan worden voor de hoogte van het verwachte *rectangle* is:

hoogte = breedte / 2.

Er mag doorgezocht worden naar de daadwerkelijke bovenkant van het oog, dit zorgt mogelijk voor betere resultaten in de rest van de stappen.

Zie Figuur 4 voor meer informatie.



Figuur 4. De neus (N) heeft links en rechts ervan een histogram, hierdoor kan makkelijker iedere neusvleugel (P & Q) bepaald worden. Vervolgens wordt er in deze histogrammen naar boven gegaan en wordt er een *rectangle* rond de ogen geschat.

Let op dat je niet ter hoogte van de ogen opeens een ‘breder’ punt vindt en dat markeerd als neusvleugel.

5. Het lokaliseren van de ogen

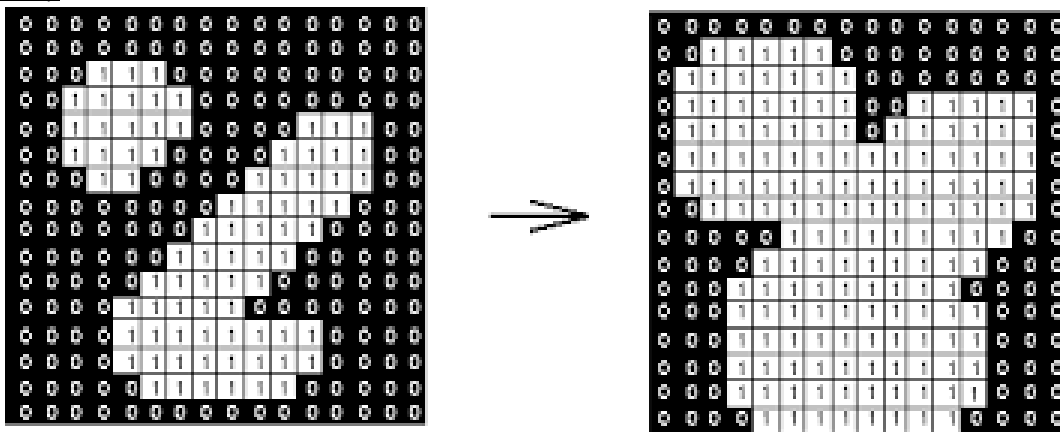
Het lokaliseren van de ogen is een belangrijk deel en de laatste uit te voeren stap van het lokaliseren. Meegegeven is een schatting van het *rectangle* waar het oog in zit, dit *rectangle* moet waar nodig vergroot of verkleind worden om zo precies mogelijk de informatie door te geven aan de volgende fase, de extractie.

In deze opdracht zal er door middel van histogrammen en *morphing* gezocht worden naar een exactere lokatie van de ogen. *Morphing* is een veel gebruikt principe in *computer vision* en leuk om mee te spelen. In dit geval gaat het om *dilation* (van de voorgrond) met de volgende kernel:

$$\text{kernel} = \begin{matrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{matrix}$$

Simpel gezegd betekent dit dat als je een zwarte pixel tegen komt (op de voorgrond) dat je in *4-connectivity* de pixels daar omheen ook zwart maakt.

Hiermee wordt er bereikt dat het oog in het meegegeven *rectangle*, qua zwarte pixels, vergroot wordt waardoor sommige delen van het oog terug aan elkaar kunnen komen te zitten. Zie Figuur 5 voor een voorbeeld van 8-connectivity dilation.



Figuur 5. (links) invoer afbeelding, (rechts) het resultaat van de 8-connectivity dilation.

Let op dat je alleen maar aanpassingen doet in een kopie van de afbeelding, anders zullen de resultaten van de *dilation* invloed uitoefenen op de rest van de resultaten.