6.x Mens-detectie

Voordat het programma werkelijke ingaand en/of uitgaand verkeer detecteert, het is noodzakelijk om het aantal personen in beeld te tellen. Dit wordt gedaan in een aantal stappen:

1. Filtering op basis van object-size.
2. Afkeuring van false-positives.
3. Verdere benadering.

Elke van de drie stappen wordt in dit hoofdstuk uitgelegd respectievelijk in paragraaf 6.x.1, 6.x.2 en 6.x.3.

## 6.x.1 Filtering op basis van object-size

Gegeven is een segmenteerde beeld, met vermoedelijk alle segmenten van de personen die zich in het oorspronkelijke frame bevinden (figuur 6.x.1 en 6.x.2).

Het is mogelijk dat sommige segmenten niet personen zijn. Die moeten we weg-filteren op basis van meerdere condities en tests. De eerste filtratie is ook het simplest – op basis van object-grootte. Elke 8-connected object wordt gelabeled alleen als het op een bepaalde grootte-threshold voldoet (figuur 6.x.3). Het valt te zien dat in bepaalde gevallen, worden er meer objecten gelabled dan het aantal personen die daar werkelijk zijn. Dit probleem wordt via de volgende stappen opgelost.



*Figuur 6.x.3 - Labeling*

*Figuur 6.x.1& 6.x.2 – Een voorbeeld frame en*

*de persoon-segmenten daarvan.*



## 6.x.2 Afkeuring van false-positives

Het blauwe object in figuur 6.x.3 is een voorbeeld van een false-positive. We hebben al een object die de persoon het best beschrijft (groen). Maar blauw is ook gelabeled omdat het toevallig op de size-constraint voldoet.

Er zijn meerdere methoen om false-positives te detecteren. Hier zijn een aantal condities die het programma in rekening neemt:

1. Object breedte/hoogte ration is raar (bijvoorbeeld te dun om een persoon te zijn).
2. Het boundingbox is te klein.
3. De positie van het boundingbox is te hoog of laag.

In figuur 6.x.3 zal blauw aan conditie 3 niet voldoen, en dus zal ook geen deel nemen in opvolgende verwerkingen.

## 6.x.3 Verdere Benadering

*Figuur 6.x.4 – Twee personen in hetzelfde object.*



Soms wordt een object als een persoon beschowd , terwijl het meer dan één persoon bevat (figuur 6.x.4). Om dit op te lossen werden er twee methoden bedacht, waarvan één niet geschikt bleek te zijn. De twee methoden zijn :

1. Custom opening en body-telling.
2. Hoofden tellen.

**Custom Opening en Body-Telling**

Deze methode gebruikt een custom structuring-element die heel dun en heel hoog is. Hiermee wordt een opening operatie gevoerd die vershillende personen uit elkaar trekt. Als laatste wordt het beeld opnieuw gelabeled en het aantal objecten geteld.

Deze methode werkt redelijk goed maar bleek heel traag te zijn, omdat het zo veel operaties (opening, labeling, measurements) inhoudt. Het was niet praktische om die te gebruiken.

**Hoofden Tellen**

Deze method neemt het bovenste deel van het object, en probeert daar hoofden te tellen. Dit wordt gedaan door horizontale scanlines te compressen tot een matrix die een soort samenvatting van elk scanline representeert. Bijvoorbeeld gegeven het volgende scanline:

[ 0 0 0 0 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 1 1 1 1 0 0 ]

Zo’n samenvatting zou zijn:

[ 0 1 0 1 0 1 0 ]

In het kort, het algoritme verkort lange patterns tot één concreet getal. U ziet dat de som van deze “samenvatting” is ook een goede afschatting van het aantal hoofden aanwezig op die scanline.

Als we dit op elke scanline toepassen en een gemiddelde van nemen een goede benadering van het aantal mensen is bereikt. Deze methode bleek heel goed te werk is ook tenminste 6x zo snel dan het alternatief (custom opening). Daarom wordt deze methode ook gebruikt in de definitieve versie van het programma.