

Meetrapport Histogramimplementaties

Martijn van Dijk
Joost Wagenveld
04-09-2016

Doel

Bij gezichtsherkenning is lokalisatie een belangrijk onderdeel. Namelijk voordat er begonnen kan worden met het herkennen van gezichten moet er eerst een lokalisatie gedaan worden van het gezicht. Door middel van lokalisatie kunnen bijvoorbeeld de bovenkant en zijkanten van het hoofd worden gevonden, dit is nodig om het kader af te perken waartussen de rest van de onderdelen van het hoofd zich bevinden. Dit is dus de eerste stap in het vinden van ogen, mond, neus, ogen enzovoorts. Wij richten ons voornamelijk op het lokaliseren van de bovenkant en de zijkanten van het hoofd.

Methoden

Voor de het vinden van de bovenkant en de zijkant van het hoofd kunnen de volgende methoden gebruikt worden:

- edge detect
- histogram per y-regel

Bij edge detection wordt er gekeken naar de omringende pixels. Als er een grote overgang is in intensiteit tussen een pixel en zijn omgeving dan wordt dat hier mee zichtbaar.

Een histogram is een diagram waarin de intensiteitsverdeling staat uitgebeeld.

Keuze

Wij hebben gekozen voor het histogram. Dit hebben wij gedaan omdat die het voordeel heeft van performance. Bij edge detection moeten er voor iedere pixel meerdere berekeningen uitgevoerd worden, het aantal is afhankelijk van de grootte van de kernel.

Bij een histogram wordt er alleen een array met intensiteiten bijgehouden. Dit levert per pixel een enkele optelling op.

Voor het genereren van het histogram zijn er twee implementaties geschreven en getest. Een met een integer array, en een implementatie met `std::map`. Een onderzoek heeft uitgewezen dat er geen significant performanceverschil is. Hierom is gekozen voor de integer-array implementatie omdat deze code minder complex is.

Implementatie

Er zal een methode geschreven worden die het histogram bepaald voor de breedte van het plaatje. Ook zal er een methode geschreven worden die de bovenkant van het hoofd bepaald en eentje die op grond van een histogram eerst de ene zijkant zoekt en vervolgens de andere.

Bij `findSidesInHistogram` wordt er van links naar rechts door het histogram heen gelopen en als de waarde van een pixel boven een drempelwaarde ligt dan zal dit betekenen dat daar een verticale rand in de afbeelding zit en dat betekend dus dat er een zijkant van het gezicht zit.

De methode die de bovenkant van het hoofd vindt werkt door de preprocessed afbeelding lijn voor lijn door te lopen totdat er een donkere pixel gevonden wordt. Dan wordt de huidige y-waarde terug gegeven.

De histogram implementatie met het integer-array loopt door een horizontale lijn in de afbeelding heen, en zet de respectievelijke array waarde op 1. De `std::map` implementatie voegt iedere zwarte pixel met de respectievelijke x-waarde als key in een map. Vervolgens wordt bij beide implementaties over dan wel het array als de map geïtereerd. Bij de `std::map` gaat dit met een `std::iterator`.

Evaluatie

Er zullen twee experimenten uitgevoerd worden: een experiment om de kwaliteit van de lokalisaties te testen, en een om het verschil in performance te meten voor de twee verschillende histogram implementaties.

Het experiment voor het testen van de snelheid van de lokalisaties werkt door met `std::chrono` de tijd te meten die het hele programma draait, en een gemiddelde te nemen over 10 monsters voor beide implementaties.

Bij het experiment voor het testen van de kwaliteit van de lokalisaties worden voor een aantal input foto's beoordelingen toegekend aan een aantal makkelijk te beoordelen lokalisaties zoals ogen, oren en mond.