

1. Implementatieplan PreProcessing 3 & 4

1.1. Namen en datum

1-6-2014

Daan Leijen

1.2. Doel

Ons doel bij deze implementatie is het leveren van een implementatie van zowel de edge detection als de thresholding die betrouwbaardere resultaten levert dan de standaardimplementatie. Hierbij achten we het niet nodig dat de implementatie sneller is dan de bestaande implementatie.

1.3. Methoden

Voor de edge detection is de meest voorkomende aanpak om te filteren op ruis, en daarna een edge detection mask en een inverse mask eroverheen te gooien.

Voor de thresholding is de keuze tussen statische en dynamische thresholding, oftewel of de grenzen van de threshold-filter zich aan moeten passen aan de grijswaarden-afbeelding.

1.4. Keuze

Voor de edge detection is gekozen om een instelbaar gauss-mask voor ruisfiltering te combineren met een laplacian-mask (gecombineerd edge- en inverse-mask). Hierbij is ook nog geëxperimenteerd met een high-pass filter erbij om de afbeelding scherper te krijgen, maar dit pakte negatief uit voor de resultaten.

Voor de thresholding is gekozen voor dynamische thresholding, omdat we de resultaatkwaliteit belangrijker vinden dan de calculatiesnelheid.

1.5. Implementatie

Gauss-filter:

- Dynamisch gegenereerd, kan voor iedere positieve gehele radius-waarde en positieve sigma-waarde berekend worden.

High-Pass filter:

- Standaard kernel

Laplacian-filter:

- Standaard kernel
- Pixelwaardes dienen eerst met 128 verhoogd te worden voordat mask toegepast word, en na toepassing van het mask weer met 128 verlaagd te worden.

Threshold-filter:

- Berekend eerst gemiddelde intensity van hele afbeelding
- Pixels die 5% of meer boven het gemiddelde zitten worden zwart, de rest wit

1.6. Evaluatie

We zullen hierbij kijken naar hoeveel afbeeldingen met de standaardimplementatie door alle stappen komen, en hoeveel met onze implementatie