

Meetrapport pre processing snelheid

1.1. Namen en datum

Bouke Stam (1664653)
Scott Mackay (1662769)
1 mei 2016

1.2. Doel

Het doel van dit meetrapport is het testen van de snelheid van de pre processing functies `stepEdgeDetection` en `stepThresholding`. Daarbij wordt vervolgens aangetoond dat onze functies snel genoeg zijn voor later gebruik. Als dat niet zo is moet het geoptimaliseerd worden.

1.3. Hypothese

We verwachten dat onze implementatie sneller is dan de geleverde `DefaultPreProcessing` omdat een kleine mask is gebruikt bij onze implementatie.

1.4. Werkwijze

Om te testen hoe snel de methodes zijn hebben we beide methodes 100 keer uitgevoerd en gemeten hoe lang de `stepEdgeDetection` en `stepThresholding` methode duurde. Vervolgens hebben we de totale tijd gedeeld door 100 om uit te vinden hoe lang de methode gemiddeld duurt.

1.5. Resultaten

Hieronder zijn de vergelijkingen in snelheid van onze code en de al geleverde code voor elk van de afbeeldingen uit de test set.

Afbeelding	StudentPreProcessing	DefaultPreProcessing
stepEdgeDetection	17 ms	8 ms
stepThresholding	2 ms	7 ms

1.6. Conclusie

Uit de resultaten blijkt dat bij onze implementatie de stepEdgeDetection langzamer is en de stepThresholding veel sneller is.

1.7. Evaluatie

De hypothese blijkt niet juist te zijn. Onze implementatie bleek over het geheel iets langzamer te zijn dan de DefaultPreProcessing implementatie. Wij denken dat de stepEdgeDetection langzamer is omdat wij de low-pass filter en high-pass filters apart na elkaar uitvoeren. De stepThresholding is waarschijnlijk sneller omdat wij een standaard threshold waarde hebben genomen.