

# 1. Meetrapport titel

## 1.1. Namen en datum

Robbin van den Berg & Danny Horvath

1 juni 2015

## 1.2. Doel

Het doel van dit experiment is om de meest efficiënte functie voor het detecteren van edges in een Intensity image te vinden. Er worden 2 functies met elkaar vergeleken.

## 1.3. Hypothese

Wij verwachten dat het verschil in snelheid minimaal is maar dat de gemiddelde snelheid van de originele functie sneller is dan de eigen gemaakte functie.

## 1.4. Werkwijze

Er zal in totaal 20x een programma uitgevoerd worden om edges te detecteren in een Intensity Image. Eerst 10x met de eigen gemaakte functie, een timer houdt bij wat de snelheid van het gedeelte is van de functie. Vervolgens hetzelfde maar dan met de originele functie. Tot slot wordt van beide functies de gemiddelde uitvoersnelheid berekend en zal blijken welke functie gemiddeld het snelst is.

Eigen functie: `IntensityImage * StudentPreProcessing::stepEdgeDetection(const IntensityImage &image);`

Originele functie: `IntensityImage * DefaultPreProcessing::stepEdgeDetection(const IntensityImage &src);`

## 1.5. Resultaten

#	Snelheid Eigen functie (micro sec)	Snelheid Originele functie (micro sec)
1	63381	15738
2	60784	12525
3	66691	13431
4	63884	15275
5	65838	11542
6	65466	11691
7	65591	17349
8	67561	11850
9	68305	12301
10	60836	12898

## 1.6. Verwerking

#	Snelheid Eigen functie (micro sec)	Snelheid Originele functie (micro sec)
1	63381	15738
2	60784	12525
3	66691	13431

4	63884	15275
5	65838	11542
6	65466	11691
7	65591	17349
8	67561	11850
9	68305	12301
10	60836	12898
<b>Totaal:</b>	<b>648337</b>	<b>134600</b>
<b>Gemiddelde:</b>	<b>64833.7</b>	<b>13460</b>

## 1.7. Conclusie

Uit de verwerking van de meetresultaten kan worden geconcludeerd dat de originele functie (13460 microseconden) gemiddeld sneller is dan de eigen gemaakte functie (648337 microseconden), precies gezegd 51373.7 microseconden sneller.

## 1.8. Evaluatie

Het doel was om de onderzoeken wat de meest efficiënte functie was voor het detecteren van edges in een Intensity Image. Uit de meetresultaten is gebleken dat onze hypothese klopte, de originele functie bleek inderdaad sneller dan onze eigen functie. Doordat er tijdens het meten ook nog andere processen draaide dan alleen het geteste programma, kan het zo zijn dat de omstandigheden waarin getest is niet gelijk zijn geweest, en er als gevolg dus meet onzekerheden kunnen zijn.