

1. Meetrapport titel

1.1. Namen en datum

Robbin van den Berg & Danny Horvath

20 mei. 2015

1.2. Doel

Het doel van dit experiment is om de meest geheugenefficiënte functie voor het omzetten van een RGBImage naar een Intensity image te vinden. Er worden 2 functie met elkaar vergeleken.

1.3. Hypothese

Wij verwachten dat het verschil in geheugengebruik minimaal is maar dat de eigen gemaakte functie minder geheugen gebruikt dan de originele functie.

1.4. Werkwijze

Er zal in totaal 2x een programma uitgevoerd worden om een RGB image om te zetten naar een Intensity Image. Eerst 1x met de eigen gemaakte functie, via taakbeheer wordt er gemeten wat het geheugengebruik is van het programma. Vervolgens hetzelfde maar dan met de originele functie. Zo kan er worden vergeleken en worden bekeken welke functie efficiënter is.

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

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

1.5. Resultaten

Geef de meetresultaten overzichtelijk weer in de vorm van een tabel en/of diagram.

#	Geheugen eigen functie (KB)	Geheugen originele functie (KB)
1	2060	2680

1.6. Verwerking

Laat zien hoe je de meetresultaten verwerkt om een conclusie te kunnen trekken. Het is niet nodig om alle berekeningen op te schrijven, als je bijvoorbeeld maar laat zien welke formule(s) je gebruikt voor het verwerken van de meetresultaten en daar zo nodig één voorbeeldberekening aan toevoegt.

#	Geheugen eigen functie (KB)	Geheugen originele functie (KB)
1	2060	2680
Vershil	2680-2060 = 620 KB	

1.7. Conclusie

Uit de verwerking van de meetresultaten kan worden geconcludeerd dat de eigen gemaakte functie (2060KB) minder geheugen gebruikt dan de originele functie (2680 KB), precies gezegd 620 KB minder.

1.8. Evaluatie

Het doel was om te onderzoeken wat de meest efficiënte functie was voor het omzetten van een RGB image naar een Intensity Image. Uit de meetresultaten is gebleken dat onze hypothese klopte, onze eigen gemaakte functie bleek inderdaad sneller dan de originele functie.