

1. Imageshell en intensity

1.1. Bianca Krieger, Marianne Delmaar 1-6-2015

1.2. Doel

Controleren hoe snel de student implementatie is van de conversie van een RGBImage naar een IntensityImage.

1.3. Hypothese

De performance van de student implementatie is hoger dan de default implementatie.

1.4. Werkwijze

Er zal een timer worden gestart. De code zal vervolgens een x aantal keer worden uitgevoerd. Daarna zal de timer worden gestopt. Er zal een gemiddelde tijd worden berekend door de tijd die verstreken is door de code uit te voeren te delen door het x aantal keer dat de code is uitgevoerd. Dit zal over meerdere afbeeldingen worden getest.

1.5. Resultaten

Afbeelding	Tijd Student impl.	Default impl.
Child-1.png	3208,9us	11433,2us
	4311,6us	13098,7us
	3818,2us	12764,6us
Female-1.png	4484,3us	15404,9us
	4108,5us	10871,0us
	4357,1us	9692,6us
Female-2.png	2262,0us	4952,3us
	1043,0us	4283,6us
	1668,4us	4594,8us

1.6. Verwerking

De functie wordt 10x uitgevoerd. Daarna zal de tijd worden vastgesteld hoe lang de functie er over gedaan heeft om 10x uitgevoerd te moeten worden. Daarna wordt de tijd gedeeld door 10 om een gemiddelde te krijgen.

1.7. Conclusie

Zoals verwacht is de performance van de Student implementatie hoger dan de Default implementatie.

1.8. Evaluatie

De meetresultaten tonen aan dat de studentenimplementatie sneller is dan de default implementatie. Dit komt waarschijnlijk doordat de studentenimplementatie een simpele aanpak heeft om de IntensityImage te berekenen, door simpelweg 1 kleur van de RGBImage te pakken en die te gebruiken als grijsintensiteit. Voor veel afbeeldingen zal dit goed genoeg zijn, maar het is mogelijk dat bij sommige afbeeldingen de default implementatie beter functioneert. Aangezien de nadruk bij onze implementatie is gelegd op een zo hoog mogelijke performance en de geteste afbeeldingen er goed uit zien, lijkt de implementatie zijn doel te bereiken.