



MEETRAPPORT IMAGESHELL SNELHEID

Jasper Oosterbroek, Jens Bouman

Doel

Voor de opgave Imageshell en Intensity is een opdracht het maken van een ImageShell. Deze klasse is een huls voor de afbeelding en dient gemaakt te worden voor zowel RGB en Intensity afbeeldingen. Bij het implementatieplan is gebleken dat de eendimensionale array de meest efficiënte keuze is.

Dit experiment

Hypothese

Aangezien ik geen informatie heb over de standaard implementatie van de ImageShell is het lastig om een verwachting te maken. Aangezien er weinig rare dingen gebeuren in de code verwacht ik dat de snelheid gelijk zal zijn aan die van de standaard implementatie.

Werkwijze

Deze test zal geheel door middel van code gedaan worden. De computer waarop de test is uitgevoerd heeft de volgende eigenschappen:

- CPU: Intel i7-4519U @2.00ghz
- RAM: 8gb
- Opslag: 7400rpm hdd

De test zelf omvat de volgende stappen:

- Zet de implementatie van de ImageFactory op default
- Zet debug mode op false bij de ImageIO klasse
- Start de timer
- Loop N keer over het volgende proces:
 - o Maak een nieuwe afbeelding (deze test is uitgevoerd met de testafbeelding "child-1.png")
 - o Controleer of de afbeelding bestaat
 - o Maak een DLL-executor aan met als input de afbeelding
 - o Voer alle stappen uit zoals aangegeven in de executeSteps functie
 - o Maak het geheugen van de input en de executor vrij.
- Stop de timer
- Sla de tijd op onder het kopje "Test Default n = N" waar N gelijk is aan het aantal iteraties dat getest is
- Zet de implementatie van de ImageFactory op Student
- Start de timer
- Herhaal het iteratieproces
- Stop de timer
- Sla de tijd op onder het kopje "Test Student n = N" waar N gelijk is aan hetzelfde aantal iteraties als bij de standaard implementatie

Na het volgend van deze test is van zowel de standaard als de student implementatie een tijd bekend. Met deze totaal tijd kunnen we berekenen hoe lang elke implementatie gemiddeld over de stappen doet.

Resultaten

Bovenstaande test is 10x gedaan met waarden van N gelijk aan 50-500 met een interval van 50.

Hieronder staan de gegevens (het getal achter de ';' is de verstreken tijd):

Test Default n = 100;17260961

Test Default n = 150;19816480

Test Default n = 200;27602767

Test Default n = 250;35810697

Test Default n = 300;43505317

Test Default n = 350;51465060

Test Default n = 400;62493527

Test Default n = 450;88019767

Test Default n = 50;6098901

Test Default n = 500;104460279

Test Student n = 100;7995322

Test Student n = 150;11856745

Test Student n = 200;15682441

Test Student n = 250;19656751

Test Student n = 300;23599192

Test Student n = 350;27605862

Test Student n = 400;37495178

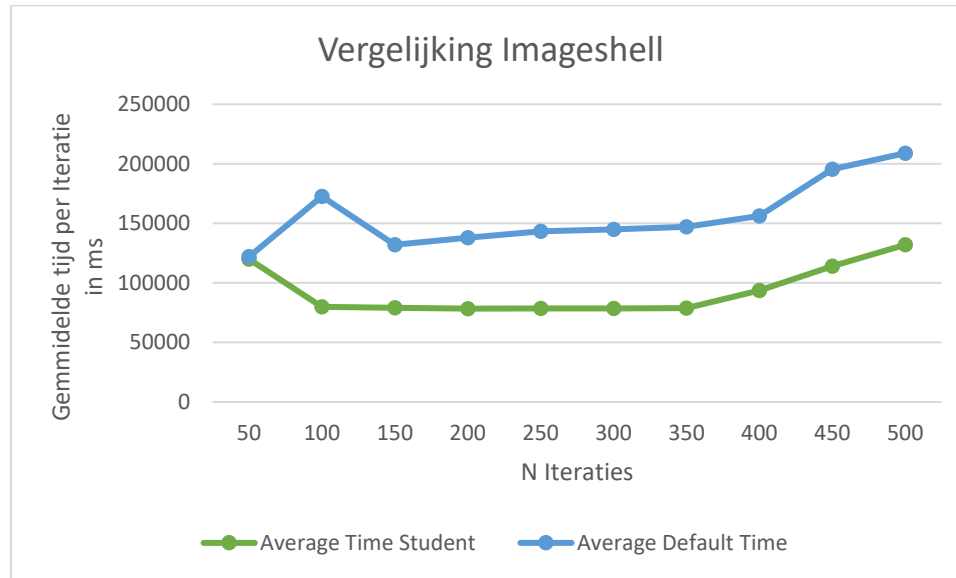
Test Student n = 450;51263956

Test Student n = 50;5993242

Test Student n = 500;66002024

Verwerking

De data is gegeven in milliseconden voor het aantal iteraties, dus voor dat we kunnen kijken welke implementatie sneller is moeten we het gemiddelde pakken (tijd/N). Hierna kan de volgende grafiek opgesteld worden:



Conclusie

Op basis van de verwerking kan gesteld worden dat de imageshell implementatie van ons sneller is dan de standaard implementatie.

Evaluatie

Ik had niet verwacht dat onze implementatie zo veel sneller zou zijn dan de standaard.