

1. Meetrapport gray scaling en Image shell

1.1. Namen en datum

- Gerris van Os en Bryan Campagne
- 19-3-2019

1.2. Doel

Het doel van het experiment is aantonen dat onze implementatie sneller is dan de standaard implementatie .

1.3. Hypothese

We verwachten dat onze implementatie van de image shell en gray scaling sneller zijn dan die van de standaard implementatie.

1.4. Werkwijze

Door de tijd te meten van het begin tot het eind van een bepaald onderdeel. Omdat deze tijd nog is kan verschillen maken we een gemiddelde van 10 proeven. We meten deze tijd door middel van een `std::chrono::high_resolution_clock::now()` deze doen we voor het desbetreffende onderdeel en na het desbetreffende onderdeel.

```
//Execute the four Pre-processing steps
auto startGrayScale= std::chrono::high_resolution_clock::now();
if (!executor->executePreProcessingStep1(false)) {
    std::cout << "Pre-processing step 1 failed!" << std::endl;
    return false;
}
auto finishGrayScale = std::chrono::high_resolution_clock::now();
elapsedGrayScale = finishGrayScale - startGrayScale;
```

Code voorbeeld van het uitrekenen van de grayscale tijd.

1.5. Resultaten

	Grayscale default	imageshel default	grayscale student	imageshell student
	0.000964559	0.0023959	0.000621402	0.00225951
	0.000783316	0.00238131	0.000663339	0.00260923
	0.00150026	0.00251004	0.000703818	0.00220226
	0.000803373	0.00690179	0.00067355	0.00184087
	0.00377509	0.00504196	0.00068449	0.00184852
	0.000860262	0.00196011	0.000600615	0.00235907
	0.00088506	0.00492745	0.000567795	0.00252572
	0.000888341	0.00228577	0.000565607	0.00225951
	0.002401	0.00252134	0.000596239	0.00181206
	0.000733721	0.00275218	0.000543362	0.00214828
gemiddelde	0.001359498	0.003367785	0.000622022	0.002186503

In bovenstaande tabel zijn alle tijden die we gevonden hebben genoteerd. Daarbij is in de onderste rij een gemiddelde over 10 tests genomen.

1.6. Verwerking

Van alle gemeten tijden hebben we een gemiddelde berekent en naar het verschil gekeken van de tijden. Hieruit bleek dat de grayscale en image shell die wij zelf geïmplementeerd hebben sneller is dan de default implementatie. Tevens is het zo dat de student implementatie een stuk meer consistent is dan de default. Echter kan hier weinig van gezegd worden met maar 10 tests.

1.7. Conclusie

De conclusie is dat zoals verwacht de zelf geïmplementeerde image shell en gray scaling zijn sneller dan de default implementatie. In het geval van de Image shell, de image shell die wij hebben geïmplementeerd heeft minder operaties omdat we een 1D array gebruiken ipv een 2D array.

1.8. Evaluatie

Het doel van dit experiment is behaald echter is het een onderzoek maar 10x uitgevoerd voor elke mogelijkheid dit resulteert in een niet zo heel accurate en representatieve uitkomst. Echter is er in dit kleine onderzoek al een duidelijk verschil te zien tussen de 2 methodes.