Meetrapport

Imageshell en grayscaling over tijd .

Bram van Bergeijk, Leo Jenneskens

2-4-2017

# Doel

Het doel van de metingen is om te onderzoeken wat het gebruik van een ander algoritme voor RGB naar grayscale voor een effect heeft op de snelheid van het programma. Voor de algoritme zullen we single color (enkele color), average(gemiddelde van alle kleuren) en luminosity(verschillende verdelingen van rood, groen en blauw).gebruiken.

# Hypothese

In het onderzoek is de verwachting dat single color het snelst zou zijn aangezien single color de minste berekeningen vereist vergeleken met de andere algoritme.

# Werkwijze.

## Imageshell.

De imageshell is waar de image in pixels wordt opgeslagen. De imageshell die tijdens het onderzoek is gemaakt, maakt gebruik van en single dimensionaal array waarin alle pixels staan als in een alle pixels in horizontale orde achter elkaar één array.

## RGB to grayscale.

Voor de conversie van RGB naar grayscale hebben we drie verschillende algoritme gebruikt single color, avarage en luminosity. voor single color hebben we rood, groen en blauw apart gemeten.

## Tijd metingen.

Voor de metingen hebben we elke combinatie. 10 maal getest om een correct resultaat te krijgen .

# Resultaat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| in seconde | default | color.r | color.b | color.g | (color.r+color.g+color.b)/3 | 0.21 R + 0.72 G + 0.07 B. |
|  | n.v.t. | single color red | single color blue | single color green | gemiddeld | luminosity |
| value | 0.737133 | 0.855676 | 0.733936 | 0.777998 | 0.730799 | 0.742846 |
| value | 0.76982 | 0.742455 | 0.73495 | 0.759512 | 0.73727 | 0.730903 |
| value | 0.725824 | 0.73663 | 0.714354 | 0.734647 | 0.728498 | 0.736198 |
| value | 0.73886 | 0.72819 | 0.71343 | 0.756406 | 0.726957 | 0.728552 |
| value | 0.75759 | 0.729563 | 0.725698 | 0.748325 | 0.741794 | 0.725038 |
| value | 0.736136 | 0.736148 | 0.656247 | 0.67267 | 0.747423 | 0.731799 |
| value | 0.750985 | 0.745464 | 0.715656 | 0.737287 | 0.679755 | 0.726906 |
| value | 0.769295 | 0.727707 | 0.723159 | 0.746112 | 0.730594 | 0.740876 |
| value | 0.762521 | 0.732298 | 0.725786 | 0.746626 | 0.751619 | 0.750911 |
| value | 0.76867 | 0.717619 | 0.727259 | 0.740251 | 0.739324 | 0.745208 |
| gemiddelde | 0.7516834 | 0.745175 | 0.7170475 | 0.7419834 | 0.7314033 | 0.7359237 |
| tegenover default( sneller) | n.v.t. | 0.0065084 | 0.0346359 | 0.0097 | 0.0202801 | 0.0157597 |

# Verwerking.

Voor de verwerking van de gemeten gegevens hebben we het gemiddelde genomen van alle 10de metingen. Dit geeft een relatief correct antwoord op hoe snel het programma runt met bepaalt algoritme. Na de gemiddelde hebben we elke gemiddelde van de default implementatie gemiddelde getrokken om te zien hoeveel sneller een bepaald algoritme is vergeleken met de default implementatie.

# Conclusie.

Uit de verwerking van de gegevens is te halen dat het single color algoritme zoals verwacht het snelst is in het geval van het gebruik van de blauw waarde van RGB.

# Evaluatie.

Zoals uit het onderzoek blijkt is single color het snelst en dat is zoals de hypothese zegt te verwachten door de minimale berekeningen die er voor nodig zijn. Er moet wel rekening gehouden worden dat bij de metingen er achtergrond processen hebben kunnen innerveren met correcte metingen. dit kan effect hebben gehad op de resultaten.