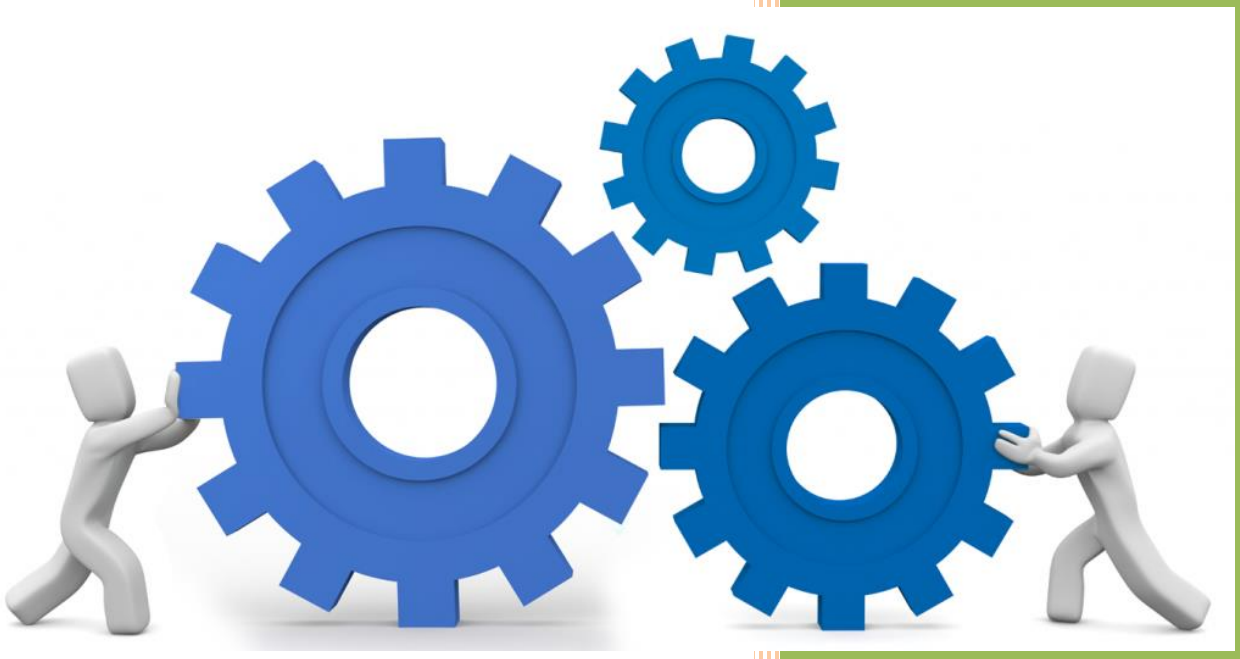
2020

Grayscale algorithm



Jeffrey de Waal & Diego Nijboer

Vision, Diederik Yamamoto-Roijers

4/4/2020

Inhoudsopgave

1.1. Doel

Het doel van onze implementatie is om het huidige systeem te verbeteren door middel van nieuwe grayscaling algoritmes te gebruiken. Het doel is de snelheid en efficiëntie van ons systeem te verbeteren. De conversie moet niet al te complex worden met lange algoritmes en moet daarnaast ook makkelijk te implementeren zijn.

1.2. Methoden

1.2.1. Imageshell

Er zijn meerdere methoden om een container te maken, hieronder staan een aantal methoden met hun voordelen

Vector:

Voordelen:

- Dynamisch aanpassen van de container grote is mogelijk.
- Bug wat betreft unkown memory location worden opgevangen en zorgen niet voor een programma crash.
- Geen handmatige geheugen management nodig, hierdoor worden er minder fouten gemaakt.

Nadelen:

- Vector alloceert geheugen wanneer hij dit nodig hebt, dit kan ervoor zorgen dat bij het opvragen van veel data, er veel opnieuw gealloceerd wordt, dit zorgt ervoor dat er veel kopieën ontstaan.

Std::array:

Voordelen:

- Efficiënter met geheugen dan Vector.
- Zelfde beveiliging als Vector.

Nadelen:

- Grote van Array moet vooraf gedefinieerd worden.

Map:

Voordelen:

- Dynamisch aanpassen van de container grote is mogelijk.
- Een reeks RGB-objecten kunnen worden opgeslagen per coördinaat als key.

Nadelen:

- Onnodig complex en minder efficiënt dan andere Array mogelijkheden.

C-pointer array:

Voordelen:

- Meest efficiënte manier om een container te implementeren, omdat dit geen extra objecten en abstractie benodigd.
- Heel snel on run-time waarde uitlezen.

Nadelen:

- Complexer te implementeren.
- Geen handmatige geheugen management nodig, hierdoor worden er minder fouten gemaakt.

1.2.2. Conversie

Schatten:
Waarde van rode, groene en blauwe kanaal worden met elkaar vergeleken en hiervan wordt een gemiddelde bepaald.

Voordelen:

- Omzetten is zeer snel en vereist geen complexe berekeningen.

Nadelen:

- Helderheid zou minder moeten zijn.

ITU-R:
Menselijke manier om grijstinten te bepalen. Waarde van rode, groene en blauwe kanaal worden vermenigvuldigd met vaste waardes.

Voordelen:

- Accurater dan schatten.

Nadelen:

- Minder efficiënt dan schatten.

Desaturation:
Omzetten van RGB-waarde naar een HSL waarde en vervolgens de saturatie op 0 zetten.

Voordelen:

- Accurater dan schatten, minder dan ITU-R.
- Mogelijkheid om HSL-waarde om te zetten naar grayscale.

Nadelen:

- Foto heeft minder contrast en wordt ook donkerder.

Decomposition:
Je neemt afhankelijk van hoe donker je de foto wil hebben het maximum of minimum van de RGB-waardes.

Voordelen:

- Makkelijk om foto’s licht te laten ogen.

Nadelen:

- Geen middenweg om een normaal belichte foto te krijgen.

Random:
Het is mogelijk om de waarde van het rode, groene of blauwe kanaal te nemen en dit te gebruiken al grijstint.

Voordelen:

- Makkelijkst en het snelst te implementeren.
- Geen calculaties nodig.

Nadelen:

- Geheel niet accuraat.
- Kleine veranderingen in kleur kanalen hebben heel veel invloed op het resultaat.

1.3. Keuze

Wij baseren onze keuze op de vergelijking van de verschillende methoden die we zojuist gedaan hebben.

1.3.1. Imageshell

Wat wij voornamelijk belangrijk vinden voor de image shell, is de snelheid, dat je van tevoren niet hoeft te definiëren wat de grootte is van de data & daarnaast dat het efficiënt is. De ideale datacontainer hiervoor is dan de pointer array, aangezien die het meeste aansluit bij onze wensen als je de voor- en nadelen bekijkt.

1.3.2. Conversie

Wat wij voornamelijk belangrijk vinden voor de datacontainer bij conversie, is dat het accuraat het is, opgevolgd door hoe makkelijk het te implementeren is. Dit is dan ook de reden dat wij de voorkeur geven aan ITU-R.

1.4. Implementatie

Onze implementatie wordt als uitbreiding in IntensityImageStudent en RGBImageStudent verwerkt. De volgende Functies worden in de code code toegevoegd:

- _fill_image_container: 2d-pointer-array wordt gevuld met pixelwaardes van meegegeven image.
- _init_image_container: 2d-pointer-array wordt gevuld met lege pixel waardes.
- Get_intensity_value: Nieuwe intensity waarde wordt berekend door gebruik te maken van de beste conversie methode.
- To_intensity: Deze functie zet RGBImage om naar een IntensityImage.

In de code zullen wij waarschijnlijk gebruik gaan maken van de ITU-R-methode aangezien deze snel en een goed resultaat zou moeten opleveren, maar dit wordt in het meetrapport nog getest.

1.5. Evaluatie

- **Snelheid en Kwaliteit:** We gaan kijken welke conversie methode het meest efficiënt en geschikt is om RGB-waardes om te zetten in grijswaarden, daarnaast nemen we ook de kwaliteit van het opgeleverde plaatje mee in de beoordeling.
- **Snelheid:** Daarnaast willen we gaan testen welke manier van het opslaan van data het meest efficiënt is. We testen de verschillende mogelijkheden die bij de methoden genoemd zijn.