

1. Verbetering Grayscale Algorithm

1.1. Namen en datum

Baartman, Michel
Bout, Nick
03 April 2018

1.2. Doel

Het oude algoritme voor grayscaling tijdens de pre-processing te vervangen door een nieuw algoritme en hierbij de snelheid en volledigheid te verbeteren. Het nieuwe algoritme dat gebruikt gaat worden van een van de onderstaande methoden beschreven in hoofdstuk 1.3.

Het is niet bekend welk algoritme gebruikt wordt binnen de geleverde applicatie.

Dit document zal een onderzoek bevatten naar de verschillende bestaande grayscaling algoritmen en informatie over de onderbouwde keuzes die gemaakt worden.

1.3. Methoden

Er wordt huidig gebruik gemaakt van een directe grijs aanwijzing d.m.v. een simpele averaging methode. Het is aan ons om een alternatief te vinden, of het huidige algoritme te verbeteren.

De volgende methoden en algoritmen hebben wij in overweging genomen:

Averaging:

Bij averaging wordt er ge-itereert over elke pixel van een color image en wordt er gekeken naar het gemiddelde van de rood+green+blue waarde.

$$\text{Gray} = (\text{Red} + \text{Green} + \text{Blue}) / 3$$

Luma/Luminance:

Bij deze algoritme wordt de waarde per kleur ook gemiddeld, maar dan wordt er ook een gewicht achter elke kleur gehangen gebaseerd op de menselijke perceptie.

$$\text{Gray} = (\text{Red} * 0.299 + \text{Green} * 0.587 + \text{Blue} * 0.114)$$

Desaturation:

Bij desaturation zal er gekeken worden naar de hoogste en laagste en hoogste RGB waarde van een selectie van pixels en daartussen het gemiddelde wordt aangewezen.

$$\text{Gray} = (\text{Max}(\text{Red}, \text{Green}, \text{Blue}) + \text{Min}(\text{Red}, \text{Green}, \text{Blue})) / 2$$

Decomposition:

Bij decomposition wordt er gekeken naar het geheel en wordt er een max en min grayscale waarde aangemaakt. Vervolgens worden de waarden binnen dat kader aangewezen.

$$\text{Gray} = \text{Max}(\text{Red}, \text{Green}, \text{Blue}), \text{Gray} = \text{Min}(\text{Red}, \text{Green}, \text{Blue})$$

1.4. Keuze

Met onze flexibele werkomgeving kunnen wij alle methoden testen en tegenover elkaar zetten. Zo kunnen wij met een meetrapport het beste algoritme boven water halen voor facial recognition en gelijk ook testen wat de eindresultaten zijn in het gehele systeem waarin dit process zich bevind.

1.5. Implementatie

Er zal een stuk code worden geïnjecteerd ter onderschepping van het huidige applicatie. Hierin zal hetzelfde (pre)process herhaald worden per algoritme voor grayscaling, inclusief het oude algoritme. De volgende waardes zullen opgeslagen worden:

Duratie grayscaling process.

Exportatie van de image als .png.

1.6. Evaluatie

Om te kijken of het nieuwe algoritme het gewenste doel heeft bereikt zullen er verschillenden testen worden uitgevoerd:

- Voor de snelheid zal zowel het nieuwe als het huidige meerdere keren uitgevoerd worden. Voor ieder algoritme wordt apart een de gemiddelde duratie gemeten en aan de hand hiervan kan er gekeken worden welk algoritme sneller is; Dit kan gevisualiseerd worden binnen een grafiek en direct vergeleken worden met de resultaten van het vorige algoritme.
- Voor de volledigheid worden ook voor zowel de oude als nieuwe algoritmes uitgevoerd, hierbij worden de resultaten opgeslagen als afbeeldingen en wordt bij de opgeslagen afbeeldingen gekeken welke afbeelding een lager aantal fouten heeft en een meer gewenst resultaat levert.

Deze tests worden vervolgens gevisualiseerd in grafieken en tabellen en toegelicht binnen het meetrapport.

1.7. Literatuurlijst

Wikipedia: Grayscale. (z.d.). Geraadpleegd op 6 april 2018, van <https://en.wikipedia.org/wiki/Grayscale>

D Cook, J. O. H. N. (z.d.). Three algorithms for converting color to grayscale. Geraadpleegd op 4 april 2018, van <https://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/>

Helland, T. A. N. N. E. R. (z.d.). Seven grayscale conversion algorithms (with pseudocode and VB6 source code). Geraadpleegd op 4 april 2018, van <http://www.tannerhelland.com/3643/grayscale-image-algorithm-vb6/hms-convert-color-grayscale/>

Klette, R. E. I. N. H. A. R. D. (z.d.). *Concise Computer Vision*. London, United Kingdoms: Springer.

Szeliski, R. I. C. H. A. R. D. (z.d.). *Computer Vision: Algorithms and Applications*. London, United Kingdoms: Springer.