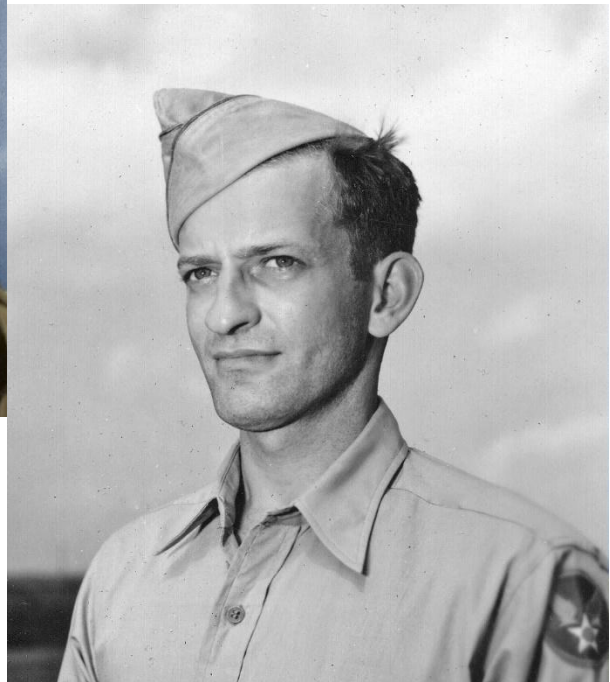


Image Shell en Intensity

Implementatieplan



Namen:	Koen de Groot, René de Kluis
Klas:	TICT-V3A2
Docent:	Arno Kamphuis
Versie:	V.2.0.0
Datum	23/03/2018

Inhoudsopgave

Inhoudsopgave	2
Inleiding	3
2. Doel	3
3. Methoden	4
3.1. Omzetting Methoden	4
3.1.1. Colorimetrische conversie	4
3.1.2. Luminance	4
3.1.3. Averaging	4
3.1.4. Desaturation	5
3.1.5. Decompositie	5
3.1.6. Single color channel	5
4. Keuze	5
5. Implementatie	6
5.1. Luminance	6
5.2. Single color channel	6
6. Evaluatie	6
7. Bibliografie	7

1. Inleiding

In dit document zal het implementatieplan van de opdracht Imageshell en Intensiteit beschreven worden. Deze opdracht maakt deel uit van de Cursus Vision, dat in het tweede jaar van HBO-ICT gegeven wordt. In dit document zal eerst het doel van de opdracht beschreven worden. Hier zal gekeken worden naar wat moet gebeuren en waarop gelet zal worden bij de implementatie. Daarna zullen de verschillende methoden voor het omrekenen van RGB waarde naar grijswaarden uitgelegd worden en op basis hiervan zal een keuze gemaakt worden welke geïmplementeerd zullen worden. Zodra deze keuze gemaakt is kan zal de implementatie worden besproken van de gekozen methoden en als laatste op welke manier wij de geworven resultaten zullen evalueren.

2. Doel

Het doel van deze opdracht is dat er een correct werkende conversie van RGB naar Intensity (ook wel Grayscale genoemd) wordt gemaakt. Arno Kamphuis heeft een grayscale manier uitgewerkt, wat in een programma voor gezichtsherkenning verwerkt zit. Het doel van deze opdracht is dat de uitwerking en vergelijkbaar en/of beter resultaat geeft dan de uitwerking in het programma.

Om de RGB waarden van een afbeelding om te kunnen zetten naar Grayscale zal een ImageShell klasse gemaakt moeten worden. De ImageShell dient als de huls voor een afbeelding. Deze klasse bevat functies die het mogelijk maken om statische data uit de afbeelding te halen of te wijzigen. Zo kan bijvoorbeeld de RGB waarde van een pixel uit een afbeelding opgevraagd worden om vervolgens hierop een grayscale algoritme uit te kunnen voeren waardoor de pixel een grijswaarde krijgt.

Bij de realisatie van dit project zal gekeken worden naar de snelheid van de omzetting in milliseconden. dit zal gedaan worden op vier verschillende afbeeldingen. Hierdoor zal een beeld gevormd kunnen worden over welke manier van Grayscale het snelste is en of de afbeelding hier invloed op heeft.

3. Methoden

Het omzetten van RGB afbeeldingen naar grayscale afbeeldingen kent verschillende methoden. Deze methodes lopen erg uiteen van de meest eenvoudige pakt een kleurwaarde van de drie kleurkanalen (RGB) en gebruikt dit als grijswaarden. Meer ingewikkelde conversie manieren gebruiken een formule om een nieuwe waarde te berekenen. Deze formules zijn gebaseerd op de waarden in de afbeelding zelf, bijvoorbeeld een berekening op de maximale en minimale RGB waarde van een pixel, terwijl anderen juist meer de werking van het menselijk oog gebruiken. Het menselijk oog ziet namelijk niet alle kleuren gelijk. Dit is te gebruiken om de RGB waarden van een pixel om te rekenen naar grijswaarden. Hieronder zal meer uitleg gegeven worden over de verschillende manieren en hoe deze werken.

3.1. Omzetting Methoden

Zoals gezegd zijn er verscheidene methoden beschikbaar om een Grayscale afbeelding te maken uit een RGB afbeelding.

De methoden die in dit document besproken worden zijn de volgende:

- Colorimetrische conversie
- Luminance
- Averaging
- Desaturation
- Decompositie
- Single color channel

3.1.1. Colorimetrische conversie

Een colorimetrische conversie werkt met de intensiteit per pixel. De conversie zorgt dat de intensiteit van de omgezette foto hetzelfde is als de intensiteit van de originele.

3.1.2. Luminance

Deze methode gaat uit van het feit dat mensen niet alle kleuren even goed zien. Mensen zien bijvoorbeeld groen beter dan rood en rood beter dan blauw. Hierdoor worden de RGB waarden bij deze methode niet gelijk behandeld. Door deze methode wordt de grijswaarde van de afbeelding meer afgestemd op wat het menselijk oog ziet.

3.1.3. Averaging

Deze manier van Grayscale is eenvoudig. Bij deze methode worden de rood, groen en blauw waarden bij elkaar opgeteld en daarna gedeeld door drie. De uitkomst hiervan wordt de grijswaarde. Omdat deze methode makkelijk is, brengt dit een probleem met zich mee. Aangezien de methode geen rekening houdt met de verschillende verzadigingen van de pixel kleuren, kan dit de helderheid van de afbeelding beïnvloeden.

3.1.4. Desaturation

Bij Desaturation wordt de RGB (Rood Groen Blauw) waarde omgezet naar een HSL (Hue, Saturation, Lightness) waarde. Hierbij is Hue de kleur van de pixel uitgedrukt in een hoek, waarbij puur rood 0° is, puur groen 120° en puur blauw 240° . De Saturation is de helderheid van de pixelkleur. Wanneer deze waarde hoog ingesteld is, zal de kleur van de pixel helder zijn. Bij de waarde nul zal de pixel een grijswaarde krijgen, dit komt aangezien grijswaarden alleen te maken hebben met intensiteit, maar niet met helderheid. Als laatste is de Lightness. Dit is de verzadigdheid van de pixelkleur. Hierbij geldt dat de kleur van de pixel volledig wit is bij de hoogste waarde en volledig zwart bij de waarde nul.

3.1.5. Decompositie

Decompositie kan op twee manieren gebruikt worden. Je kan uitgaan van de max RGB waarden van een pixel of de min RGB waarden. Het maakt bij decompositie niet uit welke kleur de max of min is het kijkt namelijk alleen naar de waarde en zet deze als grijswaarde.

3.1.6. Single Color Channel

Met deze conversie kunnen er drie foto's verkregen worden. Deze conversie zet namelijk de RGB waarden allemaal op de waarde van rood, groen of blauw. Omdat deze methode slechts 1 van de 3 kleurwaardes gebruikt zou het heel snel resultaat moeten kunnen leveren. Het nadeel hieraan is dat de resulterende afbeelding heel erg afhankelijk is van de gekozen kleur. Hierdoor kan de kwaliteit van de afbeelding heel erg verschillen.

4. Keuze

Aan de hand van de hierboven beschreven methodes is er gekeken naar wat interessant is om als onderzoek op het gebied van grayscale te doen rond het vak vision. Zoals hierboven te zien is zijn er 6 verschillende methoden om een afbeelding te grayscalen. Elke van deze methoden heeft zijn voor- en nadelen en de manieren hoe dit geïmplementeerd wordt lopen erg uiteen. De vraag voor ons was dan dus ook welke van deze methode of methodes zijn interessant om te bekijken?

De keuze is uiteindelijk gevallen om single color en luminance te gebruiken en de resultaten te vergelijken met de standaard implementatie van de software. Deze keuze hebben we gemaakt, omdat het verschil tussen single color en luminance erg groot is.

Single color gebruikt namelijk één van de drie al bestaande waardes en kijkt dus verder niet naar de rest en zal ook nooit berekenen wat nu het beste is. Gezien deze methode alleen maar een kleurwaarde op hoeft te vragen en hier verders niets mee doet zal deze waarschijnlijk de hoogste snelheid behalen. Luminance daar aan tegen heeft alle drie de waardes nodig om vervolgens aan de hand van een berekening te bepalen wat de nieuwe grayscale waarde is. Hierdoor zal naar alle waarschijnlijkheid de omzetting trager zijn dan de Single color methode, maar uit de testen zal blijken of dit ook trager is dan de standaard implementatie.

Dit is dan ook wat wij willen gaan testen, is single color sneller? En natuurlijk is het verschil groot genoeg dat het uitmaakt? De standaard implementatie zal gebruikt worden als basis om de methode met elkaar te vergelijken.

5. Implementatie

Voor het implementeren van de gekozen methoden, single color channel en luminance zullen de pixels op dezelfde manier benaderd worden. De pixels van de originele afbeelding worden namelijk opgevraagd. Van deze pixels kunnen de kleurwaarden teruggekregen worden. Met deze kleurwaarden zijn nodig om een grijswaarde te bepalen die opgeslagen zal worden in de nieuwe intensity image.

5.1. Luminance

Zoals eerder beschreven, gaat luminance ervan uit dat het oog bepaalde kleurintensiteiten meer ziet dan anderen. Wij zullen hierbij de BT.709 standard, ook wel bekend als Rec. 709, gebruiken. Dit is namelijk de standaard die ook gebruikt wordt voor HDTV's. De formule die gebruikt wordt door Rec.709 voor wat het menselijk oog ziet lijkt erg op die van de iets oudere rec.601 maar de Rec.709 houdt meer rekening met moderne schermen.

Voor de implementatie zullen wij de volgende formule gebruiken voor het berekenen van de grijswaarden: $Grayscale = (R * 0.2126 + G * 0.7152 + B * 0.0722)$

5.2. Single color channel

Voor de uitwerking van single color hebben wij gekozen om de groenwaarde van een pixel te gebruiken als grijswaarde. Zoals hierboven bij luminance beschreven is, kan het menselijk oog de kleur groen het beste zien. Hierdoor hebben wij de keuze gemaakt om ook deze waarde te gebruiken voor de implementatie van de methode. Om deze methode als formule uit te drukken zal dit als volgt eruit komen te zien: $Grayscale = G$.

6. Evaluatie

Om te bewijzen welk van de methode het snelste is zullen we de standaard implementatie van de software gebruiken als basis en Luminance en Single Color Channel hiermee vergelijken.

De snelheid zal bepaald worden door elk van de methoden 50 keer uit te voeren over vier verschillende afbeeldingen en de tijd die het programma nodig heeft om de grayscale conversie uit te voeren op te slaan. Aan de hand van deze tijden zal een grafiek gemaakt worden waarin af te lezen is welke conversie het snelst was. Ook kan aan de hand van deze tijden een conclusie worden getrokken welke afbeelding het snelst geconverteerd kan worden bij de verschillende methoden.

7. Bibliografie

Helland, T. (2011, Oktober 1). *Seven grayscale conversion algorithms*. Opgehaald van tannerhelland: <http://www.tannerhelland.com/3643/grayscale-image-algorithm-vb6/>

N.B. (2017, Februari 2). *Grayscale*. Opgehaald van Wikipedia: <https://en.wikipedia.org/wiki/Grayscale>

N.B. (2017, Januari 26). *Kleur*. Opgehaald van Wikipedia: <https://nl.wikipedia.org/wiki/Kleur>

N.B. (2017, Januari 12). *Rec. 709*. Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/Rec._709

Sean. (sd). *Digital camera sensors*. Opgehaald van Cambridge in colour: <http://www.cambridgeincolour.com/tutorials/camera-sensors.htm>