Implementatie plan RGB ->Grayscale.

Geschreven door Mike Hoogendoorn & Richard Janssen.

Datum: 21-02-2020

Datum laatste wijziging: 06-03-2020

Inhoudsopgave:

[1. Doel: 2](#_Toc34391922)

[2. Methoden: 2](#_Toc34391923)

[3. Keuze: 3](#_Toc34391924)

[4. Implementatie: 3](#_Toc34391925)

[5. Evaluatie: 3](#_Toc34391926)

# Doel:

We hebben bestaande code die een foto converteert van het RGB-kleurenruimte naar pure grijswaarden om er vervolgens allerlei andere operaties op uit te voeren en om uiteindelijk op gezichtsherkenning uit te komen, deze code is slecht geoptimaliseerd en er gaat informatie verloren. Het stukje code verantwoordelijk voor de eerste stap (RGB naar grijs) moet worden herschreven om de hoeveelheid detailverlies te verminderen en de accuraatheid te verbeteren zodat alle vervolg stappen soepeler verlopen.

# Methoden:

Na een uitgebreide studie naar de meerdere mogelijkheden om RGB om te zetten naar grijswaardes zijn we tot 3 methodes gekomen, alle 3 van deze methodes geven iedere kleur een prioriteit, deze prioriteit bepaald de uiteindelijke grijswaardes. 2 van deze methodes hebben wij in onderzoeksverslagen gevonden op research gate, beide deze research papers zijn in de bronvermelding te vinden. Optie 1 uit [de paper van tarun Kumar & Karun Verma](https://www.researchgate.net/publication/46286639_A_Theory_Based_on_Conversion_of_RGB_image_to_Gray_image) geeft rood een prioriteit van 33,3%, groen van 50% en blauw van 16,66% om tot de volgende formule uit te komen:

**Iy=0.333\*190+0.5\*183+0.1666\*175 =183.925**

Bovenstaande formule is gebaseerd op de formule van [research paper 2](https://www.researchgate.net/publication/252363505_Color-to-grayscale_conversion_to_maintain_discriminability) geschreven door Raja Bala & Karen M. Braun die de prioriteiten rood als 30%, groen als 59% en blauw als maar 11% om tot deze formule te komen:

**GRAY = 0.30 R + 0.59 G + 0.11 B**

Vervolgens komen we uit op optie 3, een formule zonder ook maar enige complexe wiskunde uit, in deze formule pakken we simpelweg het gemiddelde van de 3 kleuren. (Alle kleuren een prioriteit van 33,3%) Een duidelijk nadeel aan deze methode is dat er geen rekening wordt gehouden met dat er in de meeste kleurenruimtes weinig blauw en heel veel mogelijke groentinten bestaan. Beide bovenstaande methodes wijzen de prioriteiten toe op basis van de hoeveelheid mogelijke kleuren in elke groep (R, G & B)

**R + G + B / 3 = G**

Alle 3 deze methodes (+ de originele implementatie van de docent) zullen worden vergeleken op de volgende vlakken:

Voor het menselijk oog op welke van de 4 grijze plaatjes het fijnst “oogt” en op zichtbare details.

Voor computer vision worden er op de plaatjes meerdere gezichtsherkenning operaties uitgevoerd, deze zullen worden vergeleken op accuraatheid voor alle 4 implementaties en voor snelheid en de hoeveelheid bewaarde details tussen onze implementatie met de 3 bovenstaande formules in vergelijking met de originele implementatie van de docent.

[Research paper 1: Tarun Kuma & Karun Verma](https://www.researchgate.net/publication/46286639_A_Theory_Based_on_Conversion_of_RGB_image_to_Gray_image)

[Research paper 2: Raja Bala & Karen M. Braun](https://www.researchgate.net/publication/252363505_Color-to-grayscale_conversion_to_maintain_discriminability)

# Keuze:

We hebben gekozen om alle 3 de bovenstaande formules te implementeren. We hebben deze keuze gemaakt omdat de functie / code voor alle 3 de formules vrijwel exact hetzelfde is en het vrijwel geen extra moeite kost of we er nou 1 of 3 implementeren. Het enige verschil zijn de waardes waarmee de berekeningen gemaakt worden. Hierdoor kunnen we de eindresultaten goed vergelijken tussen de verschillende formules en achteraf kiezen welke van de 3 het beste resultaat oplevert. De resultaten van deze tests en onderzoeken zullen in het meetrapport document weergegeven worden.

# Implementatie:

De functie voor het converteren van RGB naar G zal worden vervangen door onze eigen functie binnen de bestaande code van “Externaldll2017.sln” in het bestand “studentPreProcessing.cpp” in de functie “stepToIntensityImage”. Er zullen geen nieuwe hulpklassen worden gebruikt. Er kunnen geen realistische aannames gemaakt worden omdat de originele code die we vervangen onbekend is. Wij hopen / verwachten dat onze implementatie betere resultaten zal opleveren maar er zijn geen zekerheden.

# Evaluatie:

We willen aantonen dat de bestaande codebase in zijn geheel verbeterd kan worden door de “fundering” ofwel de eerste stap te optimaliseren. Dit is relevant omdat de resultaten van deze eerste stap in alle volgende stappen gebruikt wordt. We gaan de snelheid van de volledige ketting aan stappen meten en vergelijken met het origineel, er bestaat al code die deze snelheid voor ons opmeet.

Ook gaan we de eindresultaten van de oude tegenover de nieuwe code vergelijken op accuraatheid en details. We zijn van plan om een set van 30 fotos van gezichten van verschillende etniciteit, geslacht en accessoires (bril etc.) te nemen voor onze tests.

We zullen ook een korte test doen om het geheugengebruik te achterhalen, maar naar verwachting zal dit verschil binnen de foutmarge zitten en geen nuttig resultaat opleveren.

We verwachten dat de accuraatheid en detail van de resultaten een beetje tot gemiddeld verbeteren en dat de snelheid niet veel verschil zal aantonen.