

# **1. Implementatieplan RGB naar Grayscale transformatie**

## **1.1. Namen en datum**

Bas van Eck en Lennart Jensen. 26-3-2019

## **1.2. Doel**

Het doel van deze implementatie is het ontwikkelen van een efficiënt en foutloos werkend algoritme voor het converteren van RGB afbeeldingen naar grayscale afbeeldingen. Dit algoritme zal vervolgens geïmplementeerd worden in computer vision.

## **1.3. Methoden**

Als eerste zullen we een basis implementatie maken, dit doen we zodat we vervolgens de algoritmes kunnen testen in het uiteindelijke programma. Als we deze standaard implementatie hebben, kunnen we de verschillende algoritmes gaan vergelijken op nauwkeurigheid en snelheid. Op basis van deze testresultaten kunnen we vervolgens een beslissing maken welk algoritme we gaan gebruiken in het uiteindelijke programma.

## **1.4. Keuze**

Tijdens de tests zijn er drie methoden vergeleken: average, lightness en luminosity. De lightness methode gebruikt de hoogste RGB-waarde en de laagste en neemt daarvan het gemiddelde. De average methode gebruikt het gemiddelde van R, G en B. De luminosity methode lijkt op de average methode maar houdt rekening met de menselijke perceptie, kleuren waarvoor het menselijk oog het meest gevoelig is wegens het zwaarst in het gemiddelde. Uit de tests bleek dat lightness minder nauwkeurig is en dat luminosity significant langzamer werkt. Om die redenen hebben wij gekozen om de average methode gebruiken voor deze implementatie.

## **1.5. Implementatie**

In de implementatie lopen we door middel van een for-loop over zowel de breedte als de lengte van de bron-afbeelding. Voor elk pixel halen we de kleurwaarde op, berekenen we op basis daarvan de intensiteit van de grayscale afbeelding en schrijven we dit weg naar de IntensityImage.

## **1.6. Evaluatie**

De conversie wordt getest door de implementatie te schrijven, en de resultaten te vergelijken met de resultaten van de standaard implementatie. Zolang de resultaten niet te ver afwijken werkt het programma correct. Verder wordt ook de snelheid getest en vergeleken met de snelheid van de andere implementaties.