

1. Implementatieplan titel

1.1. Namen en datum

Nick Goris & Roxanne van der Pol

09-04-2018

1.2. Doel

Het implementeren van andere greyscale algoritmen, in de hoop dat deze sneller zijn en tot kleinere object files leiden.

1.3. Methoden

Er is gekozen om een greyscale algoritme te implementeren dat meer gelijkend is op het menselijk oog. In het menselijk oog is de verdeling van kleuren namelijk niet precies egaal (zoals in het vorige implementatieplan gedaan werd door alle R+G+B bij elkaar op te tellen en dit door 3 te delen), maar volgt het een bepaalde formule – deze techniek heet “luminance”. (Helland, 2011)

1.4. Keuze

In het vorige experiment was er een methode opgenomen die een gelijkwaardige verdeling gebruikte van de R, G en B waarden, maar deze methode faalde regelmatig.

1.5. Implementatie

Voor het greyscalen is in de file StudentPreProcessing.cpp de stepToIntensityImage() functie geïmplementeerd.

1.6. Evaluatie

Wij zullen de code compileren en runnen. Als er dan een grayscale plaatje gegenereerd wordt en hier een gezicht in herkend wordt, weten we dat onze implementatie werkt. De tijd die het volledige programma duurt zal gemeten worden, evenals de grootte van de resulterende code.

Wij denken dat deze methode, in tegenstelling tot methode 2 uit het vorige experiment, wel tot de gewenste resultaten zal leiden.

2. Bibliography

Helland, T. (2011, October 1). *Seven grayscale conversion algorithms (with pseudocode and VB6 source code)*. Retrieved April 9, 2018, from tannerhelland: <http://www.tannerhelland.com/3643/grayscale-image-algorithm-vb6/>