

# 1. Implementatieplan Practicum week 5 - Scaling

## 1.1. Namen en datum

Danny Horvath & Robbin van den Berg 31-05-2015

## 1.2. Doel

Het doel van de implementatie is om een afbeelding met het behouden van de aspect ratio zo goed mogelijk te verkleinen als deze groter dan 40.000 pixels is, naar een formaat wat het dichtst in de buurt van 40.000 pixels ligt.

## 1.3. Methoden

Als eerste zijn er twee methoden waarmee alle transformaties uitgevoerd kunnen worden namelijk:

- Forward mapping
- Backward mapping

Bij forward mapping bereken je voor elke oude pixel de nieuwe coördinaten en kopieer je de huidige kleurwaarde.

Dit heeft 3 nadelen:

- Sommige coördinaten van de outputpixels kunnen buiten het beeld liggen en dus niet weergegeven worden
- Berekende getallen zijn geen gehele getallen dus deze worden afgerond
- Niet alle output pixels krijgen een waarde

Bij backward mapping bereken je voor elke nieuwe pixel de oude coördinaten en kopieer je de kleurwaarde, dit heeft geen van de nadelen van forward mapping. De verder besproken methodes worden dan ook gebaseerd op backward mapping.

Voor het scalen zelf zijn er ook een aantal methodes:

- Nearest neighbor(Zero-order interpolation)
- Bilinear interpolation
- Bicubic interpolation

Bij nearest neighbor wordt er gekeken bij welke pixel het tussenliggende punt zich het dichtst bevindt, hierna wordt het tussenliggende punt de kleur van de dichtstbijzijnde pixel toegewezen.

Een voordeel hiervan is dat het vrij snel is, een nadeel hiervan is dat het de afbeelding blokkerig maakt.

Bij bilinear interpolation wordt gekeken naar de vier omliggende burens van het tussenliggende punt, hierna wordt aan de hand van het gewogen gemiddelde van de vier burens het tussenliggende punt een kleur toegewezen.

Deze transformatie geeft een mooie afbeelding terug.

Bij bicubic interpolation wordt hetzelfde gedaan als bij bilinear interpolation alleen wordt dan gekeken naar de 16 omliggende burens.

Dit geeft een veel mooier resultaat als bilinear interpolation het nadeel is wel dat dit computationeel heel duur is.

## **1.4.       Keuze**

Wij hebben gekozen voor backward mapping met als scaling methode bilinear interpolation.

Deze keuze hebben wij gemaakt om zo de nadelen van forward mapping te omzeilen, wij hebben dan ook voor bilinear interpolation gekozen omdat wij een nette afbeelding wilden opleveren en geen blokkerige zoals bij nearest neighbor. Wij hebben niet gekozen voor bicubic interpolation omdat wij een goede performance van ons systeem ook cruciaal vonden en deze methode nogal veel rekenkracht kost.

## **1.5.       Implementatie**

We kijken als eerste wat de aspect ratio van de huidige afbeelding is zodat deze niet verloren gaat. Hierna berekenen we de nieuwe breedte en grootte voor de image.

Hierna voeren we een berekening uit om de nieuwe x en y ratio van de afbeelding te berekenen hierna lopen we door de nieuwe afbeelding heen waarna we elke pixel terugrekenen via backward mapping en kopiëren en berekenen we de nieuwe kleurwaarden.

## **1.6.       Evaluatie**

We testen of de afbeelding daadwerkelijk kleiner wordt gemaakt, ook testen we of het programma een kopie van het origineel verstuurd als de afbeelding kleiner dan 40.000 pixels is. We testen ook weer de algehele performance van onze methode dit houdt in het geheugengebruik en de snelheid van onze methode.