Implementatieplan scaling/interpolatie

Bart Muelders & Feiko Wielsma

## Doel

Het doel van deze implementatie is het schalen van de afbeelding zodat deze geschikt is voor verdere detectie. Er zal onderzoek gedaan worden naar verschillende methoden/algoritme om een afbeelding te kunnen schalen.

## Methoden

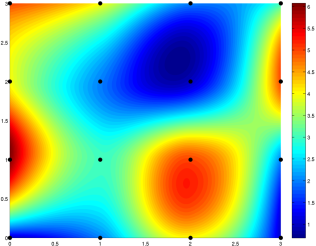
#### Nearest-neighbour interpolation

Nearest-neighbour interpolatie is een eenvoudige methode. Get algoritme zoekt het dichtsbijzijnde punt dat bij een gegeven positie ligt zonder te kijken naar de waardes van omliggende punten. Deze manier wordt vaak geimplementeerd in real-time 3D rendering. (Nearest-neighbor interpolation, 2015)

#### http://upload.wikimedia.org/wikipedia/commons/thumb/9/91/Bilinear_interpolation_visualisation.svg/512px-Bilinear_interpolation_visualisation.svg.pngBilinear interpolation

Met Bilinear interpolatie wordteerst gebruik gemaakt van een linear interpolatie. Vervolgens worden de omliggende pixels (2x2) gescant en de gemiddelde kleur hiervan berekend. Op deze manier worden randen mooier. Deze methode kost wel meer rekenkracht dan de Nearest-neighbour interpolatie. (Bilinear interpolation, 2015)

#### Bicubic interpolatie

Bicubic interpolatie levered een scherper eresultaat op dan Nearest-neighbour of Bilinear interpolatie. Bicubic interpolatie kost veel rekentijd en wordt dus vaak gebruikt in applicaties waarbij snelheid geen rol speelt. Bicubic interpolatie scant, net zoals Bilinear, de omliggende pixels, maar gebruikt hiervoor een 4x4 grid, dus 16 pixels. Hierdoor worden (randen) van afbeeldingen scherper, maar de rekentijd wordt aanzienelijk verhoogt. (Bicubic interpolation, 2015)

## Keuze

## Implementatie

## Evaluatie