Implementatieplan scaling/interpolatie

Bart Muelders ~~& Feiko Wielsma~~

Doel

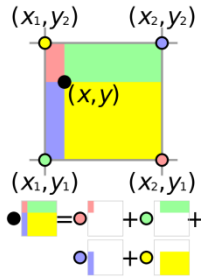
Het doel van deze implementatie is het schalen van de afbeelding zodat deze geschikt is voor verdere detectie. Er zal onderzoek gedaan worden naar verschillende methoden/algoritme om een afbeelding te kunnen schalen.

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)

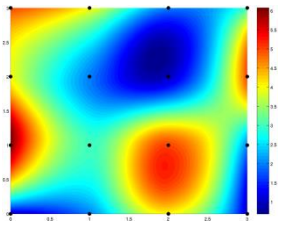
Bilinear interpolation



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

Voor het schalen van de gezichten wordt in eerste instantie gekozen voor de Nearest-neighboor interpolation methode. Dit is een methode die eenvoudig te implenteren is, en weinig rekenkracht gebruikt.

Implementatie

Voor de implementatie wordt gebruik gemaakt van backward mapping. Dit houdt in dat er vanuit de nieuwe afbeelding teruggekeken wordt in oude afbeelding om zo af te leiden welke pixel in de nieuwe

afbeelding gezet moet worden. Op deze manier worden ongedefineerde pixels voorkomen. Hiervoor wordt een inversematrix van 3x3 gebruikt.

$$\mathbf{A}^{-1} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^{-1} = \frac{1}{\det(\mathbf{A})} \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}^{T} = \frac{1}{\det(\mathbf{A})} \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix}$$

$$\begin{array}{lll} A = (ei - fh) & D = -(bi - ch) & G = (bf - ce) \\ B = -(di - fg) & E = (ai - cg) & H = -(af - cd) \\ C = (dh - eg) & F = -(ah - bg) & I = (ae - bd) \end{array}$$

Evaluatie

De nearest-neighbour methode blijkt uitstekend te werken. Er is weinig tot geen verschil te zien tussen de standaard schaling methode. Wel valt op dat de student schaling een iets kleinere afbeelding opleverd dan de standard methode.

Bronnen

Bicubic interpolation. (2015, 05 19). Retrieved 05 24, 2015, from Wikipedia: http://en.wikipedia.org/wiki/Bicubic_interpolation

Bilinear interpolation. (2015, 05 16). Retrieved 05 24, 2015, from Wikipedia: http://en.wikipedia.org/wiki/Bilinear_interpolation

Nearest-neighbor interpolation. (2015, 05 06). Retrieved 05 24, 2015, from Wikipedia: http://en.wikipedia.org/wiki/Nearest-neighbor_interpolation