

1. Implementatieplan Practicum week 3 – Edge detection

1.1. Namen en datum

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

1.2. Doel

Het doel van de implementatie is om in een gegeven afbeelding de edges te detecteren met behulp van een edge detection methode. Hierna dienen deze edges met een thresholding methode versterkt te worden, waarna er op de afbeelding alleen nog maar edges te zien zijn.

1.3. Methoden

Als eerste behandelen we de volgende edge detection methodes:

- Canny edge detection
- Sobel operator
- LoG

Canny edge detection maakt eigenlijk gebruik van een Gaussian filter om de afbeelding te smoothen, hierna wordt er met vier filter voor de detectie van horizontale, verticale en diagonale edges gedetecteerd. Ten slotte worden de edges met thresholding versterkt.

Een voordeel hiervan is dat het een vrij nauwkeurige edge detection methode is.

Nadelen zijn:

- Gevoelig voor ruis
- Veel stappen om te doorlopen

De sobel operator maakt gebruik van twee kernels, een voor horizontale en een voor verticale edge detection, smoothing wordt al gedaan bij de Sobel operator waardoor dit geen specifieke stap is. Er dient hierna echter wel nog steeds thresholding gedaan te worden. Deze methode heeft moeite met het detecteren van gradienten omdat deze gebruik maakt van een redelijk klein kernel namelijk 3x3.

Voordelen zijn dat deze methode vrij snel is en nauwkeurig edges detecteert.

Een nadeel is dat deze methode niet erg accuraat is in het detecteren van gradienten.

Met LoG wordt een gaussian filter gebruikt om de image te smoothen waarna er een laplacian operator op los wordt gelaten om de edges te detecteren. Deze methode is echter zeer gevoelig voor ruis, en is minder nauwkeurig in het detecteren van edges.

1.4. Keuze

Wij hebben gekozen voor de Sobel operator

Deze keuze hebben wij gemaakt omdat deze methode vrij snel te implementeren is, weinig last van ruis heeft en nauwkeurig is in het detecteren van edges.

De canny edge detection methode is nauwkeuriger maar is computationeel veel duurder en is qua implementatie een stuk complexer.

De LoG methode is zeer gevoelig voor ruis en bestaat uit twee stappen die uiteindelijk minder sterk edges detecteren als de Sobel operator.

1.5. Implementatie

Je geeft aan hoe deze keuze is geïmplementeerd in de code

We hebben twee 3x3 Sobel masks gemaakt 1 voor verticaal en 1 voor horizontaal, hierna lopen we door de gehele afbeelding waarbij we de kernel steeds op 1 pixel leggen. Wat dan overblijft zijn de randen, deze zetten wij op 0 wat erin resulteert dat ze zwart worden.

Hierna doen we de thresholding die bij ons op 159 staat alle pixels die boven deze waarden komen worden op zwart gezet waardoor de edges goed zichtbaar worden.

1.6. Evaluatie

We gaan testen of het programma nog alle facial features van de gewenste afbeelding kan herkennen, ook testen we de snelheid en de algehele performance van onze methode dit houdt in het geheugengebruik en de snelheid van onze methode.