Meetrapport

Snelheid

Jeroen

Stefan

22-03-2017

Inhoudsopgave

[**Doel**](#_o7yknm9tm4fc) **3**

[**Hypothese**](#_5mtpjhv8z7bn) **4**

[**Werkwijze**](#_491kd8lg1mn0) **5**

[**Resultaten**](#_i9a42w4bcmbo) **6**

[**Verwerking**](#_mbrbszl7bigv) **7**

[**Conclusie**](#_n53bd6aqtqpd) **8**

[**Evaluatie**](#_jfkusewlt9n1) **9**

[**Bijlage**](#_meuavd41i91q) **10**

# Doel

Het doel van dit meetrapport is om te kijken welke van de gekozen methodes beschreven in het implementatieplan de snelste is. Des te minder tijd onze code kost om te runnen, des te beter.

# Hypothese

Wij vermoeden dat BGD het snelst is van alle methoden, maar dat het iets meer false positives oplevert dan de andere methodes Sobel en Laplacian, omdat BGD de kleinste kernel is. Onder andere omdat BGD geen ruisfiltering heeft, kan het mooi in een 1\*3 kernel die 2x wordt gerund. In tegenstelling tot Sobel’s 3x3 kernel die twee keer moet worden gebruikt.

# Werkwijze

Wij zullen op snelheid testen door de ExternalDLL.exe. via een Python script - te vinden in de bijlage - 10.000 keer op dezelfde foto aan te roepen. Van elke run van de ExternalDLL.exe krijgen wij de tijd in microseconden terug van het edge detection gedeelte(zonder de thresholding). Deze waardes voegen wij toe aan een lijst waar wij het gemiddelde, het minimum en het maximum opvragen. De minimum en maximum gebruiken wij om mogelijke uitschieters te herkennen, maar we gebruiken het gemiddelde om de uiteindelijke snelheid te bepalen.

# Resultaten

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Average  Millisecond/run | | Minimum  Millisecond/run | | Maximum Millisecond/run | |
|  | Stefan | Jeroen | Stefan | Jeroen | Stefan | Jeroen |
| Orginele code |  | 0.7856205 |  | 0.685 |  | 19.529 |
| Originele kernel | 16.3950648 | 12.5989881 | 14.87 | 11.544 | 29.659 | 98.883 |
| BGD |  | 3.7060111 |  | 3.423 |  | 35.433 |
| Sobel |  | 2.4922526 |  | 2.184 |  | 40.24 |
| Laplacian |  | 2.309039 |  | 2.169 |  | 37.544 |

# Verwerking

Wat opvalt is dat de maximum bij Jeroen erg extreem uitslaan en bij Stefan niet.

BGD is een kwart trager dan Sobel en Laplacian.

De orignele code is ~3x zo snel als BGD, Sobel en Laplacian.

De orginele kernel met onze code is >12x zo traag als de orginele code.

# 

# Conclusie

# Evaluatie

# 

# Bijlage

Het python script