# Implementatieplan titel

## Namen en datum

## Doel

Geef aan wat het doel van de implementatie is.

Het doel is om een werkend edge detection te schrijven

## Methoden

Je geeft hier aan welke methoden er zijn, wat de verschillende tussen de methodes zijn.

Canny:

stappen:  
 Gaussian filter

vinden van de intensiteitsgradiënten van de afbeelding

Non-maximum onderdrukking

Hysteresis Thresholding

<https://docs.opencv.org/3.4.3/da/d22/tutorial_py_canny.html>

<https://en.wikipedia.org/wiki/Canny_edge_detector>

<http://robotics.technion.ac.il/courses/Advanced_Laboratory/Lab7/ARL_7_read.pdf>

Roberts:

Stappen:

Haal eerst de volgende twee kernels over de image heen:

|  |  |
| --- | --- |
| +1 | 0 |
| 0 | -1 |

|  |  |
| --- | --- |
| 0 | +1 |
| -1 | 0 |

Gx Gy

Bereken vervolgens het gradiënt aan de hand van de twee nieuwe afbeeldingen die zijn ontstaan uit bovenstaande kernels en het origineel. Dat kan met de volgende formule:

|G| = √(Gx2 + Gy2)

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/roberts.htm>

<http://robotics.technion.ac.il/courses/Advanced_Laboratory/Lab7/ARL_7_read.pdf>

<https://en.wikipedia.org/wiki/Roberts_cross>

Kleine kernel, alleen integers.

Grote gevoeligheid voor ruis.

prewitt

stappen:

haal eerst de volgende twee kernels over de image:

|  |  |  |
| --- | --- | --- |
| +1 | 0 | -1 |
| +1 | 0 | -1 |
| +1 | 0 | -1 |

|  |  |  |
| --- | --- | --- |
| +1 | +1 | +1 |
| 0 | 0 | 0 |
| -1 | -1 | -1 |

Gx = Gy =

Combineer de twee resulterende afbeeldingen met de volgende formule:

|G| = √(Gx2 + Gy2)

Maar om de berekening te versnellen word vaak de volgende formule gebruikt:

|G| = |Gx| + |Gy|

<https://en.wikipedia.org/wiki/Prewitt_operator>

twee aparte maskers eentje voor de x richting en eentje voor de y richting.

Sobel:

Stappen:

haal eerst de volgende twee kernels over de image:

|  |  |  |
| --- | --- | --- |
| +1 | 0 | -1 |
| +2 | 0 | -2 |
| +1 | 0 | -1 |

|  |  |  |
| --- | --- | --- |
| +1 | +2 | +1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

Gx = Gy =

Combineer de twee resulterende afbeeldingen met de volgende formule:

|G| = √(Gx2 + Gy2)

Maar om de berekening te versnellen word vaak de volgende formule gebruikt:

|G| = |Gx| + |Gy|

<https://en.wikipedia.org/wiki/Sobel_operator>

<http://robotics.technion.ac.il/courses/Advanced_Laboratory/Lab7/ARL_7_read.pdf>

gebruikt veel resources

is erg gevoelig voor ruis

Deriche:

Stappen:

stappen:  
 IIR filter

vinden van de intensiteitsgradiënten van de afbeelding

Non-maximum onderdrukking

Hysteresis Thresholding

<https://en.wikipedia.org/wiki/Deriche_edge_detector>

<https://apps.dtic.mil/dtic/tr/fulltext/u2/a130824.pdf>

Gebaseerd op Canny.

Eerste twee stappen verschilt.

Laplacian Edge Detector

Stappen:

Gaussian filter

Laplacian filter

Thresholding

Voorbeelden van Laplacian filter kernels:

|  |  |  |
| --- | --- | --- |
| 0 | -1 | 0 |
| -1 | 4 | -1 |
| 0 | -1 | 0 |

|  |  |  |
| --- | --- | --- |
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 1 |
| 1 | -8 | 1 |
| 1 | 1 | 1 |

<http://www.aishack.in/tutorials/sobel-laplacian-edge-detectors/>

## Keuze

Je geeft een onderbouwing over waarom een bepaalde methode is gekozen, en/of waarom bepaalde settings zijn gebruikt.

## Implementatie

Je geeft aan hoe deze keuze is geimplementeerd in de code

## Evaluatie

Je geeft aan welke experimenten er gedaan zullen worden om de implementatie te testen en te ‘bewijzen’ dat de implementatie daadwerkelijk correct werkt. Dit geeft direct informatie over de meetrapporten die er zullen worden gemaakt.