# Implementatieplan titel

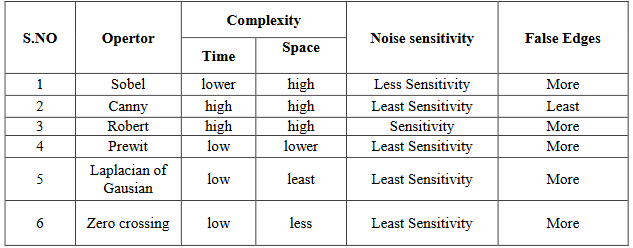
## Namen en datum

## Doel

Een nauwkeurigere implementatie van de edge detection.

## Methoden

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



* **Canny**.
* Deriche.
* Differential.
* Sobel.
* Prewitt.
* Roberts cross.

**Wij hebben gekozen voor het Canny algoritme in combinatie met de gaussian filter.  
Voordeel:**  
Dit algoritme is een van de nauwkeurigste.  
**Nadeel:**Sloom en kost veel geheugen.

Een methode om kernel filters te gebruiken  
<https://en.wikipedia.org/wiki/Kernel_(image_processing)>

Een methode om gaussian filter te gebruiken  
Je kunt hier kiezen uit verschillende sizes.  
<https://en.wikipedia.org/wiki/Gaussian_filter>   
Een methode om edge detection toe te passen  
<https://en.wikipedia.org/wiki/Canny_edge_detector>   
Een methode om een threshold aan te geven en de edges buiten deze threshold te verwijderen.  
<https://en.wikipedia.org/wiki/Canny_edge_detector>

**Bronnen:**

<https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e>

<http://www.cs.cmu.edu/~aayushb/pubs/edges.pdf>

<https://arxiv.org/ftp/arxiv/papers/1405/1405.6132.pdf>

Methode:

## Keuze

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

Op het moment wordt gebruik gemaakt van Laplacian operator. Deze operator heeft een hogere snelheid en kost minder geheugen dan het Canny algoritme. Wij willen graag meer precisie, dus leveren wij graag tijd en ruimte in voor nauwkeurigheid.

**Settings: TODO**

## 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.