# Implementatieplan Edge detection

## Namen en datum

Kiet van Osnabrugge en Wiebe van Breukelen, datum

## Doel

*Geef aan wat het doel van de implementatie is.*

Het doel van onze implementatie is om meer bruikbare informatie uit de edges te halen met als doen om een nauwkeurige en robuustere edge afbeelding te verkrijgen. Denk hierbij aan het berekenen van hoeken en lijndiktes met behulp van de edge gradienten. Met deze informatie is het bijvoorbeeld mogelijk om bepaalde edges te kunnen negeren.*. Dit valt buiten de scope van deze opdracht.*

## Methoden

*Je geeft hier aan welke methoden er zijn, wat de verschillende tussen de methodes zijn.* <http://aishack.in/tutorials/sobel-laplacian-edge-detectors/>

Methoden:

* **Laplacian edge detection** (standaard implementatie):

Levert alle edges op maar beeld dit slecht af en is erg gevoelig voor ruis (dit is iets te verbeteren met behulp van Gaussian ruis). Met Laplacian edge detection is het niet mogelijk om een richting van een edge te berekenen, met Prewiit en Sobel is dit wel mogelijk.

* **Prewitt edge detection**

Edge detection voor de x en y gradient met lager threshold. Deze gradienten worden gebruikt om de hoek van de edges te bereken.

Levert alle (mogelijk te veel onnodige edges) edges op met richting van de edge.

* **Sobel edge detection**

Edge detection voor de x en y gradient met hoger threshold (in tegenstelling tot Prewiit). Deze gradienten worden gebruikt om de hoek van de edge te bereken.

Levert alle (minder onnodige edges dan prewitt) edges op met de richting van de edge.

* **Canny** (als verwerking op sobel of prewitt):

Filtert op de relevante edges. Verfijnt de relevante lijnen en maakt ze dunner. Linkt lijnen aan elkaar waarbij er maar delen van de lijn uitgefilterd zijn.

## Keuze

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

Wij hebben gekozen voor (methode), omdat (onderbouwing).

## Implementatie

*Je geeft aan hoe deze keuze is geïmplementeerd in de code. Moet dit van de voren al bepaald worden?*

## 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.

@Kiet Ik heb hieronder wat voorbeelden gezet, we moeten deze nog bespreken.

* **Nauwkeurigheid**: Onafhankelijke beoordeling door andere personen. Hierbij zal een persoon een simpele afbeelding moeten overtrekken met de voor hem/haar relevante edges. Het resultaat hiervan zal vergeleken worden met de twee edge detection implementatie.
* **Robuustheid**: Extra ruis toevoegen, detecteren in welke mate edges nog correct bepaald worden. (Hoe wordt dit op correctheid gecontroleerd).
* **Performance**: RAM-gebruik, CPU-gebruik, etc.