

# 1. Implementatieplan vinden top hoofd

## 1.1. Namen en datum

Bouke Stam (1664653)  
Scott Mackay (1662769)  
24 April 2016

## 1.2. Doel

Het doel van de deze implementatie is het uitvinden wat de beste manier is voor een edge detection en een thresholding functie.

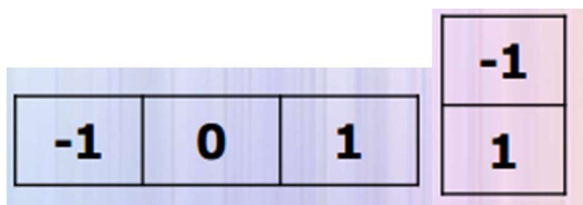
## 1.3. Methoden

Om de edge detection aan te pakken hadden we de keuze tussen de Laplacian en de Gradient masks. Daarbij voor beter noise reduced resultaat wordt eerst een low-pass filter overheen gehaald.

De Laplacian wordt uitgevoerd door een negatieve waarde in het midden van de mask te zetten. Vervolgens bij alle andere posities van de mask positieve waarde zetten die bij elkaar opgeteld dezelfde waarde als de negatieve waarde hebben. Deze waarden worden steeds kleiner naarmate het hokje verder van het midden af is. De Laplacian heeft als voordeel dat de randen een duidelijke overgang geven van donkere naar lichtere plekken.

```
{ 0.5, 1, 0.5 },  
{ 1, -6, 1 },  
{ 0.5, 1, 0.5 }
```

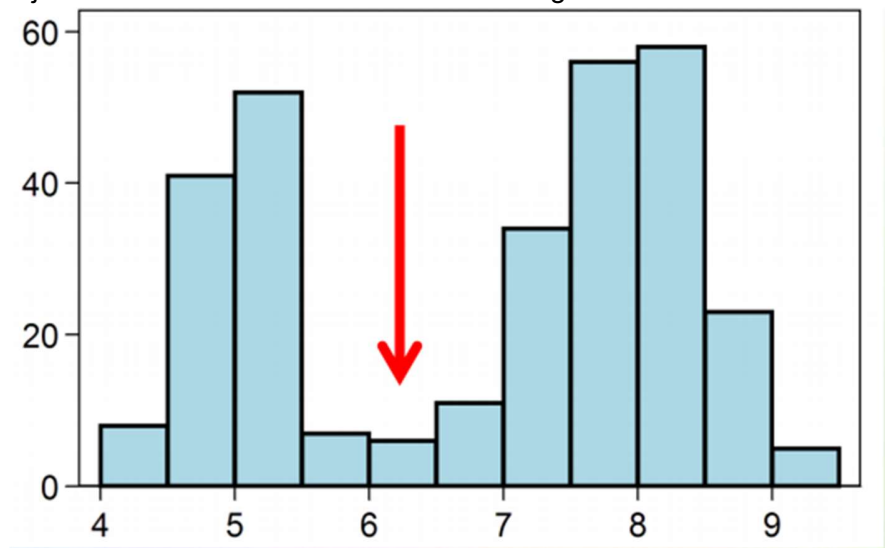
De gradient pakt de pixels aan de ene kant van de pixel waar het mee vergeleken wordt en trekt die bij de pixel waar het mee vergeleken wordt af (of de andere kant van de pixel trekt het van af). Het verschil van de twee pixels is hierdoor gecreëerd. Als een grote verandering in grijswaarde plaatsvindt is de gradient hoog. Hierdoor worden overgangen sterker naar zwart toe en ontstaan edges. De Gradient heeft als voordeel dat de lijnen scherper op de plek zelf zitten en niet om de randen heen.



Voor de thresholding zijn er de volgende keuzes voor het vinden van een drempelwaarde:

Een standaard waarde nemen die goed werkt op een gezicht. Dit is veruit de snelste manier. De threshold waarde moet namelijk niet berekend worden. Het nadeel van deze methode is dat de methode niet goed meer werkt op andere soort afbeeldingen met minder spreiding tussen donker en licht. Ook bij een erg donkere of lichte afbeelding gaat deze methode niet werken.

Een andere methode is Balanced histogram thresholding. Om te threshold waarde te vinden wordt een histogram gemaakt van de hoeveelheid pixels per grijswaarde. Dan worden de twee pieken gevonden in kleur van de histogram. En vervolgens wordt het midden genomen van de twee pieken als threshold waarde. Deze methode heeft als nadeel dat bij een histogram met maar 1 piek er geen goed resultaat zal zijn. Of bij een histogram waar de twee pieken erg dicht bij elkaar zitten zal deze methode ook niet goed werken.



## 1.4. Keuze

Voor edge detection hebben wij gekozen voor de Laplacian. Dit filter is geeft ons het voordeel om te kunnen zien welke kant de donkere kant van de edge is. Daarbij wordt eerst een mean filter gebruikt om de ruis te verminderen.

Verder hebben wij voor de standaard waarde methode gekozen voor de thresholding. Dit hebben wij gekozen om het simpel te houden. Omdat alle images van dezelfde soort zijn zal deze methode geen groot nadeel leveren voor de vision opdrachten die we in de toekomst zullen krijgen.

## 1.5. Implementatie

Om de edge detection met Laplacian te laten werken hebben we eerst een class geschreven die de mask opslaat. In de StudentPreProcessing file is ook een methode geschreven die de mask toepast op een IntensityImage.

De IntensityImage die meegegeven wordt, wordt eerst gemask met een mean filter van 3x3 met allemaal 1 gevult. Vervolgens wordt een Laplacian mask toegepast om de laplacian effect te geven. Daarna worden de waarden genormalized.

```
{ 0.5, 1, 0.5 },  
{ 1, -6, 1 },  
{ 0.5, 1, 0.5 }
```

(laplacian mask)

Vervolgens bij de thresholding wordt de static waarde vergeleken met iedere pixel van de IntensityImage om op zwart of wit te zetten. Bij een hogere waarde dan de threshold wordt de pixel op zwart gezet. Bij een lagere waarde wordt die op wit gezet.

## 1.6. Evaluatie

Om te bewijzen dat deze implementatie werkt zullen er twee tests geschreven worden. Allereerst moet er gekeken worden of beide filters goed werken. De tweede test zal een snelheidstest zijn. De snelheid van deze functies zal vergeleken worden met die van de al geleverde functies die hetzelfde doel hebben.