# Implementatieplan titel

## Namen en datum

Groepsnaam : Kaasbroodje

Leden : Jalu Glissenaar, Thomas Smeele

Datum : 7 Februari 2019

## Doel

Het doel van de Implementatie is om het resultaat van de gezicht herkenning te verbeteren. Verbeteren in deze zin betekent hoe accuraat het gezicht herkenning werkt, niet dat het sneller werkt of minder memory gebruikt.

## Methoden

Voordeel en Nadelen

Keuze criteria formule

De vereiste input van methode

* Canny Edge Detection

Sources :

* <https://en.wikipedia.org/wiki/Canny_edge_detector>
* <https://www.youtube.com/watch?v=sRFM5IEqR2w>

De Canny edge detector is an edge detectie operator die

De Canny edge-detector is een randdetectie algoritme die meerdere stappen gebruikt om een breed scala aan randen in afbeeldingen te detecteren. Het werd ontwikkeld door John F. Canny in 1986. De stappen die gebruikt worden door Canny’s methode zijn:

1. Pas een Gaussian filter aan het plaatje om de ruis te verwijderen.

2. Zoek de intensiteitsgradiënten van de afbeelding

3. Pas niet-maximale onderdrukking toe om valse reactie op randdetectie te verwijderen

4. Pas dubbele drempel toe om potentiële randen te bepalen

5. Volg rand met hysteresis: Voltooi de detectie van randen door alle andere randen die zwak zijn te onderdrukken en niet verbonden zijn met sterke randen.

1. Apply [Gaussian filter](https://en.wikipedia.org/wiki/Gaussian_filter) to smooth the image in order to remove the noise
2. Find the intensity gradients of the image
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by [hysteresis](https://en.wikipedia.org/wiki/Hysteresis): Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

* Deriche

**Deriche edge detector** is an [edge detection](https://en.wikipedia.org/wiki/Edge_detection) operator developed by [Rachid Deriche](https://en.wikipedia.org/wiki/Rachid_Deriche) in 1987. It's a multistep [algorithm](https://en.wikipedia.org/wiki/Algorithm) used to obtain an optimal result of edge detection in a discrete two-dimensional image. This algorithm is based on [John F. Canny](https://en.wikipedia.org/wiki/John_F._Canny)'s work related to the edge detection ([Canny's edge detector](https://en.wikipedia.org/wiki/Canny_edge_detector" \o "Canny edge detector)) and his criteria for optimal edge detection:

1. Detection quality – all existing edges should be marked and no false detection should occur.
2. Accuracy - the marked edges should be as close to the edges in the real image as possible.
3. Unambiguity - a given edge in the image should only be marked once. No multiple responses to one edge in the real image should occur.

Deriche edge detector, like [Canny edge detector](https://en.wikipedia.org/wiki/Canny_edge_detector), consists of the following 4 steps:

1. *Smoothing*
2. *Calculation of magnitude and gradient direction*
3. *Non-maximum suppression*
4. *Hysteresis thresholding (using two thresholds)*

* Sobel

Sobel and Feldman presented the idea of an "[Isotropic](https://en.wikipedia.org/wiki/Isotropy) 3x3 Image Gradient Operator" at a talk at SAIL in 1968.[[1]](https://en.wikipedia.org/wiki/Sobel_operator#cite_note-1) Technically, it is a [discrete differentiation operator](https://en.wikipedia.org/wiki/Difference_operator), computing an approximation of the [gradient](https://en.wikipedia.org/wiki/Image_gradient) of the image intensity function. At each point in the image, the result of the Sobel–Feldman operator is either the corresponding gradient vector or the [norm](https://en.wikipedia.org/wiki/Norm_(mathematics)) of this vector. The Sobel–Feldman operator is based on convolving the image with a small, separable, and integer-valued filter in the horizontal and vertical directions and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation that it produces is relatively crude, in particular for high-frequency variations in the image.

The operator uses two 3×3 kernels which are [convolved](https://en.wikipedia.org/wiki/Kernel_(image_processing)#Convolution) with the original image to calculate approximations of the [derivatives](https://en.wikipedia.org/wiki/Image_Derivatives) – one for horizontal changes, and one for vertical.

* Prewitt

The **Prewitt operator** is used in [image processing](https://en.wikipedia.org/wiki/Image_processing), particularly within [edge detection](https://en.wikipedia.org/wiki/Edge_detection) algorithms. Technically, it is a [discrete differentiation operator](https://en.wikipedia.org/wiki/Difference_operator), computing an approximation of the [gradient](https://en.wikipedia.org/wiki/Image_gradient) of the image intensity function. At each point in the image, the result of the Prewitt operator is either the corresponding gradient vector or the norm of this vector. The Prewitt operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical directions and is therefore relatively inexpensive in terms of computations like [Sobel](https://en.wikipedia.org/wiki/Sobel_operator) and Kayyali[[1]](https://en.wikipedia.org/wiki/Prewitt_operator" \l "cite_note-1) operators. On the other hand, the gradient approximation which it produces is relatively crude, in particular for high frequency variations in the image. The Prewitt operator was developed by Judith M. S. Prewitt[[2]](https://en.wikipedia.org/wiki/Prewitt_operator#cite_note-2).

In simple terms, the operator calculates the [*gradient*](https://en.wikipedia.org/wiki/Image_gradient) of the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. The result therefore shows how "abruptly" or "smoothly" the image changes at that point, and therefore how likely it is that part of the image represents an *edge*, as well as how that edge is likely to be oriented. In practice, the magnitude (likelihood of an edge) calculation is more reliable and easier to interpret than the direction calculation.

Mathematically, the [gradient](https://en.wikipedia.org/wiki/Gradient) of a two-variable function (here the image intensity function) is at each image point a 2D [vector](https://en.wikipedia.org/wiki/Vector_(geometric)) with the components given by the [derivatives](https://en.wikipedia.org/wiki/Derivative) in the horizontal and vertical directions. At each image point, the gradient vector points in the direction of largest possible intensity increase, and the length of the gradient vector corresponds to the rate of change in that direction. This implies that the result of the Prewitt operator at an image point which is in a region of constant image intensity is a zero vector and at a point on an edge is a vector which points across the edge, from darker to brighter values.

* Roberto Carlos

The **Roberts cross** operator is used in [image processing](https://en.wikipedia.org/wiki/Image_processing) and [computer vision](https://en.wikipedia.org/wiki/Computer_vision) for [edge detection](https://en.wikipedia.org/wiki/Edge_detection). It was one of the first edge detectors and was initially proposed by [Lawrence Roberts](https://en.wikipedia.org/wiki/Lawrence_Roberts_(scientist)) in 1963.[[1]](https://en.wikipedia.org/wiki/Roberts_cross#cite_note-1) As a [differential operator](https://en.wikipedia.org/wiki/Difference_operator), the idea behind the Roberts cross operator is to approximate the [gradient](https://en.wikipedia.org/wiki/Gradient) of an image through discrete differentiation which is achieved by computing the sum of the squares of the differences between diagonally adjacent pixels.

According to Roberts, an edge detector should have the following properties: the produced edges should be well-defined, the background should contribute as little noise as possible, and the intensity of edges should correspond as close as possible to what a human would perceive. With these criteria in mind and based on then prevailing psychophysical theory Roberts proposed the following equations:

{\displaystyle y\_{i,j}={\sqrt {x\_{i,j}}}}

{\displaystyle z\_{i,j}={\sqrt {(y\_{i,j}-y\_{i+1,j+1})^{2}+(y\_{i+1,j}-y\_{i,j+1})^{2}}}}

where x is the initial intensity value in the image, z is the computed derivative and i,j represent the location in the image.

The results of this operation will highlight changes in intensity in a diagonal direction. One of the most appealing aspects of this operation is its simplicity; the kernel is small and contains only integers. However with the speed of computers today this advantage is negligible and the Roberts cross suffers greatly from sensitivity to noise.

## Keuze

Onze keuze wordt de *Canny Edge Detector* Algoritme. Uit ons onderzoek blijkt dat canny het beste resultaat geeft met andere vergeleken algoritmes die dezelfde computer kracht gebruiken. Deze opdracht heeft namelijk niet super goede *edge detection* nodig omdat de foto’s op het algemeen vrij simpel zijn.

De gekozen methode/techniek is relevant voor de taak die is gegeven; De keuze voor de methode/techniek is onderbouwd; De opzet voor de implementatie van de gekozen methode/techniek is functioneel.

De stappen om het resultaat te bereiken zijn aangegeven. De methode/techniek zijn in deze stappen aangegeven. Meerdere oplossingen zijn besproken. Voor- en nadelen zijn aangegeven. Een keuze-criteria is geformuleerd. De vereiste input van de methode/techniek is beschikbaar en wordt gebruikt. De methode/techniek geeft de vereiste output in alle gevallen.

## Implementatie

Je geeft aan hoe deze keuze is geïmplementeerd 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.