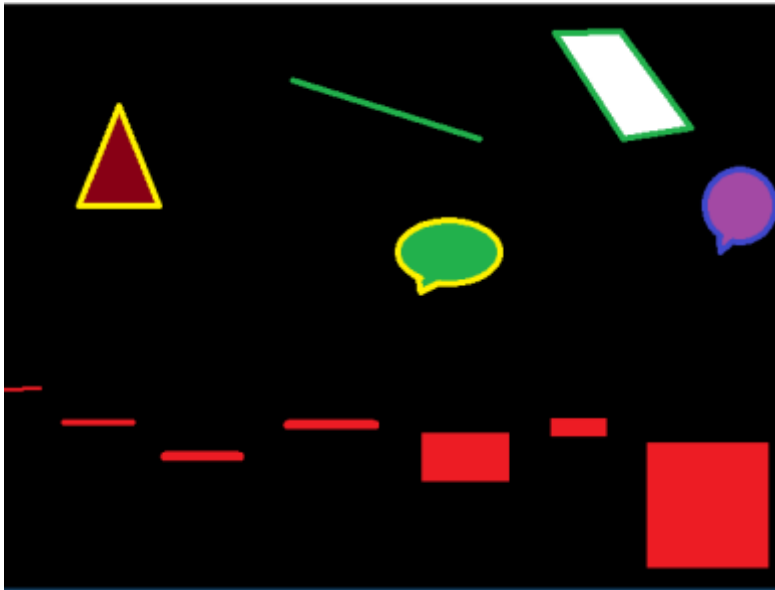


Unit Test Scanner

Het was de bedoeling om in eerste instantie de rode pixels bij elkaar op te tellen en zo een benadering maken van de werkelijke hoogte van een object. Hierbij is een Test-foto gemaakt die in figuur 1 is weergegeven.



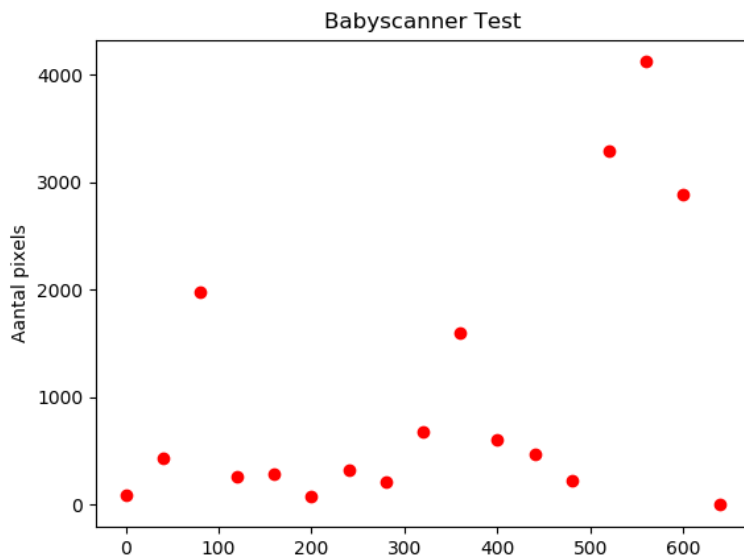
Figuur 1: Test Foto

In figuur 1 is duidelijk te zien, dat er meerdere kleuren en figuren op de foto staan. Om uiteindelijk in het praktijk een mooi beeld te krijgen van de XYZ punten van een object, zal er dus gefilterd moeten worden, waar alleen de rode pixels in de figuur blijven. Er is dus geprogrammeerd om met RGB waarden te werken, waarbij de minimum en maximum RGB waarde zodanig gekozen wordt, dat alleen het beeld die we willen overblijft. Figuur 2 is het resultaat. Hier is duidelijk te zien dat alleen de rode figuren aanwezig zijn in het beeld.



Figuur 2: Resultaat filter werking (tel rode pixels)

De volgende stap is dus dan ook om de pixels op te meten over de x-as van de frame. Hier wordt de frame verdeeld in stroken in de x-as. Dit levert een heel reeks van stroken, waarbij de pixels opgemeten kan worden. In figuur 3 is het pixel aantal weergegeven.



Figuur 3: Aantal pixels per x-as

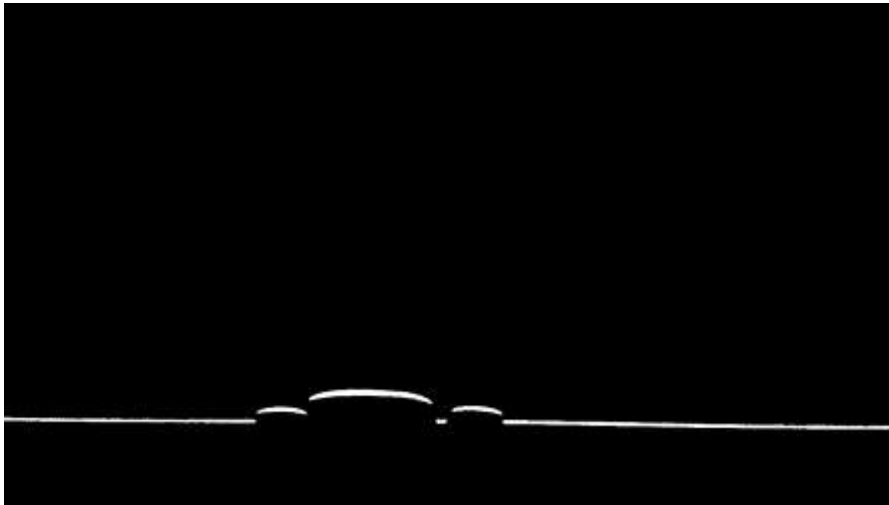
Wanneer er getest wordt in de praktijk, wanneer een foto gemaakt wordt van een object, lijkt dit heel wat ingewikkelder te zijn. Allereerst moet de minimum waarde voor rood aangepast worden, omdat de laserlijn geen fel rode licht geeft. Dit wordt het moeilijk om alles wat niet rood er uit te filteren, omdat er ook in het gebied van de laserlijn objecten zijn met de zelfde kleur intensiteit en dit wordt ook meegerekend als rood.

De eerste oplossing die bedacht werd, was om de omgeving zwart te maken, zodat alleen de laserlijn zichtbaar was voor de camera. Na het scannen in een donker omgeving, was de probleem, dat de laser teveel reflecteerde van het object af, waarbij er in het laserlijn middelpunt een witte lijn zat, zoals weergegeven in figuur 4. En door de programma, wat elke rode pixels telt, telde hij de middelste pixels niet op. Dit is makkelijk op te lossen, wanneer het aantal witte pixels in het midden van de lijn even groot waren, maar dat was helaas niet het geval.



Figuur 4: Foto met donker achtergrond

Als het ware brengt het handmatig bepalen van de threshold veel problemen met zich mee. Uiteindelijk was de OTSU filter in documentatie van OpenCV gevonden. De algoritme van dit filter gaat ervan uit dat de afbeelding 2 klassen pixels bevatten, één voor de voorgrond en één voor het achtergrond (Image Thresholding, sd). Dus nu de omgeving zwart is en het enig licht/object een ander kleur heeft het als het achtergrond was de OTSU filter de oplossing. In figuur 5 is een foto te zien met de OTSU filter.



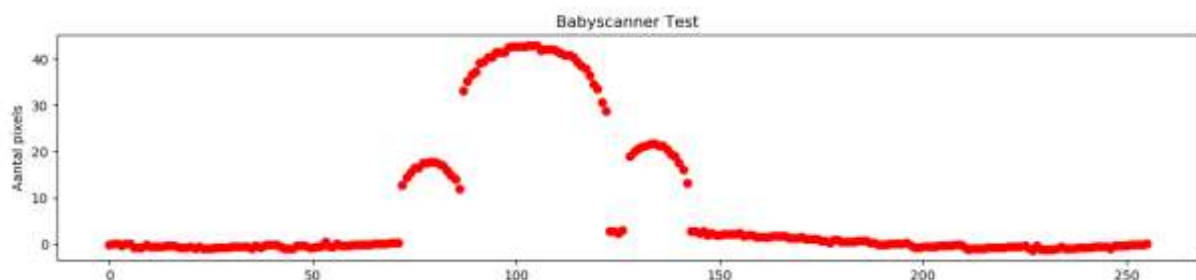
Figuur 5: Foto met OTSU filter

Een ander probleem dat ook optrad bij het tellen van het aantal pixels, was dat de intensiteit bij een object dat hoger lag niet significant hoger lag. Vanuit ging, hoe hoger het object (Hoe dichterbij het object van de camera is), hoe intensiever de laserlijn werd, maar helaas was het verschil tussen bepaalde hoogte zeer klein. Er moest een ander oplossing hiervoor bedacht worden en dat is om de lengte tussen een basislijn van de laser met de kromming van de laserlijn te bepalen. Wanneer een object geplaatst wordt, waar een laser op gericht is, zal door triangulatie een laser kromming ontstaan. Dit laser kromming kan voor vrij nauwkeurige metingen zorgen (Jao Guilherme D.M. Franca).

Om de lengte te kunnen weten van de kromming, zal een lijn beredeneerd worden aan de hand van de vergelijking 1:

$$y = ax + b$$

De vergelijking die een lijn zal voorstellen, zal precies op het pad zijn, waar de laserlijn zich bevindt. Alleen nu zonder de kromming. Wanneer een laserlijn, met een kromming, wordt vergeleken de bepaalde lijn, kan de lengte worden berekend. Wanneer de coördinaten op een bepaald punt van zowel de laserlijn, met kromming, als het bepaalde lijn bekend zijn, kan de y-coördinaat van elkaar afgetrokken worden voor de lengte. In figuur 6 is dit weergegeven.



Figuur 6: Pixel afstand t.o.v basis lijn

Nu de afstand bekend is tussen de basislijn en de kromming kan de Hoogte van het object bepaald worden. In “Onderzoek hoogtemeting” is de volgende vergelijking 1 bekend:

$$D = \frac{h}{\tan(pfc * rpc + ro)} \quad (1)$$

Duidelijk werd gemaakt dat het systeem gekalibreerd moet worden. Allereerst moet de afstand tussen de basis lijn en de kromming bepaald worden van objecten, waarvan de hoogte al bekend zijn. Er is hiervoor 5 verschillende objecten uitgekozen met alle 6 verschillende hoogte. Van daaruit is de afstand ook bepaald. In tabel 1 is dit weergegeven.

Tabel 1: Kalibratie deel 1

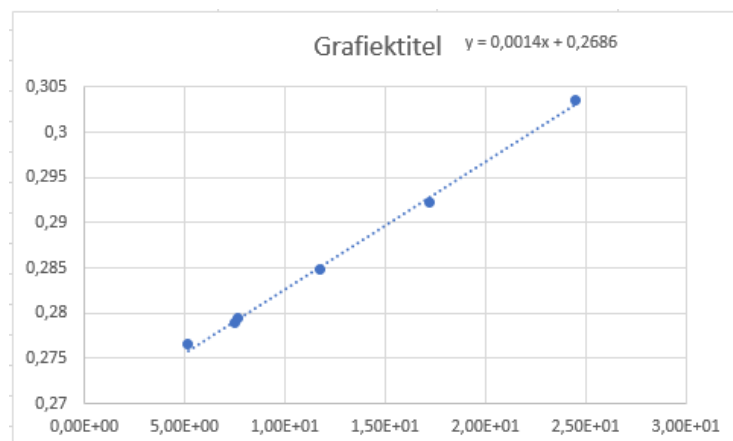
Object	Pix_afstand	Object-hoogte (in cm)
Hout_0	5,16690664325463	0,7
Hout_1	7,48854489164081	1,1
Hout_2	7,665248869	1,2
Hout_3	12	2,1
Hout_4	17,19547237	3,3
Hout_5	24,48657255	5

In “Onderzoek hoogtemeting” staat beschreven dat vergelijking 1 de afstand geeft tussen het object en de laser. Maar dat is niet de lengte die we willen hebben. In dit project is de hoogte van het object belangrijk. Een simpele oplossing hiervoor is om de vergelijking af te trekken van de totale hoogte waarover gemeten wordt en dit is 46,5 cm. Verder moet $\theta_{Werkelijk}$ bepaald worden voor elk kalibratiepunt, zoals beschreven in het onderzoek. In tabel 2 zijn de gegevens weergegeven.

Tabel 2: Kalibratie deel 2

Object	Afstand(46,5 – Objecthoogte)	$\theta_{Werkelijk}$
Hout_0	48,5	0,276568559
Hout_1	45,4	0,278881393
Hout_2	45,3	0,279465499
Hout_3	44,4	0,284831633
Hout_4	43,3	0,292306051
Hout_5	41,5	0,303570739

Uiteindelijk kan de formule bepaald uit figuur 8.



Figuur 7: Calibratie vergelijking

Zoals beschreven in de onderzoek kan θ ook geschreven worden als vergelijking 2:

$$\theta = pfc * rpc + ro \quad (2)$$

Wat opvalt is dat dit overeenkomt met de berekende kalibratie. We kunnen voor rpc en ro de volgende waardes invullen:

$$\begin{aligned} rpc &= 0,0014 \\ ro &= 0,2686 \end{aligned}$$

De formule voor de hoogte van een object kan geschreven worden als vergelijking 3:

$$D = 46,5 - \left(\frac{h}{\tan(pfc * 0,0014 + 0,2686)} \right) \quad (3)$$

Waarbij h gelijk is aan de afstand tussen de laser en de camera, ongeveer 13,6cm