Nu, net voor prototype 3, zijn we tot de conclusie gekomen dat het normaliseren van het beeld niet te doen is. De voornaamste complicatie is dat er te veel rekenkracht nodig is voor gecompliceerde manieren van normaliseren. Het lijkt erop dat een makkelijke manier niet afdoende is.

Dit komt voornamelijk omdat het beeld niet genormaliseerd moet worden op het gehele beeld, maar alleen op de achtergrond. De achtergrond wordt pas bepaald in de stap na de normalisatie, maar als je het omdraait is er weer geen goede normalisatie voor het segmenteren. Het is lastig een goede manier te vinden die ook daadwerkelijk elk beeld real-time kan normaliseren op basis van de achtergrond.

Het systeem werkt wel zo dat er eerste op de achtergrond gekalibreerd moet worden. Dit doormiddel van het maken van een foto van de achtergrond, zonder mensen in het beeld dus. Op basis van de gemaakte “foto” en het huidige beeld theoretisch heel simpel de achtergrond worden bepaald. Dit blijkt toch nog redelijk lastig door de ruis, maar met namen door de verandering van het lichtbalans van de webcam.

Het is dus een noodzaak om de webcam standaard zo in te stellen dat het lichtbalans niet veranderd. Dit levert direct al een significant beter resultaat op bij onder andere de segmentatie. En dat zonder dat er een verdere normalisatie heeft plaatsgevonden.

In eerste instantie zijn de volgende normalisatie methodes gebruikt:

* De gemiddelde intensiteit over het gehele beeld en alle lagen samen normaliseren rond ongeveer de helft van 255, genomen is 125. Dit door simpelweg alle waarden in het beeld te vermenigvuldigen met de verhouding van het gemiddelde tot de 125 normalisatie grens. Daarna worden de pieken of dalen afgekapt en terug gebracht naar respectievelijk 1 of 0. Deze overduidelijk simpele methode werkt laat maar zeggen te dof. Wanneer door bijvoorbeeld een object met veel intensiteit in beeld komt, zal de gemiddelde waarde flink stijgen. Waardoor ook de waardes van alle pixels sterk zullen veranderen. Dit resulteert in een compleet ander intensiteit van de achtergrond waardoor de drempelwaardes in de segmentatie niet meer werken.
* Met deze methode wordt de gemiddelde beeld intensiteit van het gekalibreerde (originele) beeld genomen. Deze wordt dan gebruikt om naar toe te kalibreren net als in de vorige methode, maar dan als vervanging voor de standaard waarde van 125.
* Voor deze methode worden er 2 versies van de achtgrond/kalibratie afbeelding opgeslagen. Een met het originele beeld en de ander met een normalisatie naar 125. Bij de normalisatie voor de beelden na de kalibratie wordt het beeld van de achtergrond afgetrokken. Daarvan wordt de inverse genomen. Theoretisch bevat dat beeld dan alleen de achtergrond. Dit natuurlijk zolang de waardes van de achtergrond niet veranderen. Wat dus is gebleken, en hierboven al uitgelegd, is dat het vaak niet correct zal werken door de automatische lichtbalans van de webcam. Dit beeld van de achtergrond wordt als binair masker over het genormaliseerde gekalibreerde en live beeld heen gelegd. Nu zijn van beide beelden dezelfde pixels geselecteerd die in het live beeld tot de achtergrond zouden moeten behoren. Door net zoals de methodes hierboven de gemiddelde intensiteit van de gehele afbeeldingen uit te reken en deze aan elkaar gelijk te stellen, zal zich een redelijk goed genormaliseerd beeld moeten vormen.

Een eerste oplossing als hierboven al vaker genoemd is het uitzetten van de automatische lichtbalans van de webcam. Dit voorkomt dat het lichtbalans tussentijds veranderd. Naar het eisen dat deze instellingen (al) van toepassing zijn op de webcam wanneer het programma wordt gedraaid, is het ook handig de volgende code bij het opstarten van de webcam uit te voeren:

src = getselectedsource(vid);

set(src, 'Brightness', 0);

set(src, 'Contrast', 0);

set(src, 'BacklightCompensation', 'off');

Hierdoor werkt voor alle methodes de segmantie al overduidelijk beter. De vraag is nu of de dit goed genoeg werkt om helemaal geen normalisatie toe te gaan passen. Het normaliseren lijkt in dit stadium niet veel voordelen met zich me te brengen, louter verbruik van systeem rekensnelheid dat beter gebruikt zou kunnen woorden voor andere delen binnen de pipeline.

De volgende link (http://research.microsoft.com/en-us/um/people/ablake/papers/ablake/criminisi\_cvpr06.pdf) geeft een paper waarin de herkenning van mensen wordt gedaan gebaseerd op het analyseren en uitwerken van bewegingen tussen de beelden. Hierbij wordt gebruik gemaakt van het kleur gebied van de pixels die bewegen. Daarnaast wordt met statistiek en kansberekening gekeken hoeveel kans omliggende pixels hebben om bij het bewegende beeld te horen. Dit levert na een paar frames al een goed resultaat op.

Bla Bla

Het lijkt uiteindelijk dus beter te zijn meer tijd te stoppen in de segmentatie en opvolgende processen dan in het normaliseren. Wanneer de webcam een vast lichtbalans heeft gaat de segmentatie al relatief goed. Normalisatie kost teveel tijd en rekenkracht in vergelijking met wat het uiteindelijk oplevert. Met de juiste methodes is er veel meer te winnen in de andere onderelen van de pipeline.

Normalise color layers based on all the other layers.

http://www.mathworks.se/matlabcentral/newsreader/view\_thread/171190

Normalise in different N x M subarea:

http://mmc36.informatik.uni-augsburg.de/mediawiki/images/f/f3/VMDL97.pdf

Normalization and segmentation based on motion and change likelihood of the color

http://research.microsoft.com/en-us/um/people/ablake/papers/ablake/criminisi\_cvpr06.pdf

General color normalization:

http://en.wikipedia.org/wiki/Color\_normalization