Opmerkingen:

Nu we een programma hadden om de hoek van de knie te meten aan de hand van een foto en die dan een voorstel gaf voor het verzetten van het zadel, moest het nog getest worden. Eerst en vooral hebben we de **invloed van de resolutie** getest. Daarvoor zijn we gestart met een foto van 1500 op 1500 pixels. Deze foto werd dan herschaald in MS paint om foto’s van lagere of hogere resolutie te krijgen. Bij de foto’s die een hogere resolutie hadden dan 500x500 zagen we dat de hoeken redelijk stabiel waren, met een maximale spreiding van 2°, bij minder grote foto’s was dit al snel 7°, wat onacceptabel is. Als we dan verder keken naar hoeveel centimeter het zadel omhoog of omlaag moest, zagen we iets anders. In dit geval zagen we pas vanaf echt hoge resolutie een redelijk stabiel resultaat met maar 2à3 mm speling. Onder de 1000x1000 was dit vaak meer dan 5 of 6mm.

Een oplossing zou kunnen zijn om het programma te laten lopen op meerdere resoluties van dezelfde gegeven foto en daarvan dan het gemiddelde te berekenen.

Probleem: We hebben een foto gevonden van een fietser waarop de hoeken van de knie en de enkel enz. zijn aangegeven. Deze heel accuraat want ze zijn berekend adhv vaste punten die aangeduid zijn op de persoon, het is dus geen schatting. We hebben deze hoeken bevestigd met een online geodriehoek. Maar openpose kiest niet altijd de keypoints juist op de gewrichten, dit komt omdat openpose ook **maar een schatting maakt van de pose**. Hierdoor kom je vaak een andere hoek uit dan in het echt en omdat we ons voor het verzetten van het zadel baseren op exacte hoeken zitten we hier dus in de problemen. Verder vragen we ook de **lengte van het dijbeen** als input, aan de hand daarvan kunnen we dan de verandering van het zadel in centimeter uit te drukken. Maar dit is een andere bron voor fouten, want we vergelijken namelijk de lengte van het dijbeen dat openpose berekent met de ingegeven waarde. Het is bijna onmogelijk dat je de exacte lengte van je dijbeen weet of meet zoals die door openpose is aangegeven. Dus als je vanaf het begin al de foute waarde ingeeft zal je nooit je ideale positie bereiken, zelfs al voer je het programma meerdere keren uit.

**Probleem bij schouderhoek:**

Deze hoek is idealiter 90°. We berekenen die hoek dus tussen de rug en de bovenarm. Maar openpose ziet de rug als één recht stuk, terwijl de rug in het echt enige buiging heeft, waardoor de hoek tussen de arm en de bovenrug wel 90° kan zijn, maar die hoek kan veel kleiner zijn als je de rug als een volledig recht stuk ziet.

Bikefitting is een heel **precieze wetenschap**, heel vaak gaat het maar om die halve centimeter verschil. Maar we hebben gemerkt dat openpose hiervoor dus niet de goede oplossing is. Openpose is niet exact genoeg bij het kiezen van de punten, er is te veel variatie door verschil in resolutie. Als er al zoveel variatie is door resolutie alleen, zullen de fouten nog veel groter zijn als de foto niet vanop de goeie positie is genomen.

Daarom denken we dat een bikefitting programma dat gebruik maakt van Openpose veel te omslachtig is. Waarschijnlijk zal je een beter en sneller resultaat hebben als je gewoon op gevoel je zadel verzet en weet dat je knie nog lichtjes gebogen moet zijn als je voet beneden staat.

Alle testen werden uitgevoerd met de assumptie dat de lengte van het dijbeen gemiddeld 26,74% van je lichaamslengte bedraagt( bron wikipedia: <https://en.wikipedia.org/wiki/Femur> ) en dat de gemiddelde mannelijke belg 178cm groot is(Wikipedia: <https://en.wikipedia.org/wiki/Average_human_height_by_country> ). Met deze waarden krijgen we een dijbeenlengte van 47,5972 cm.

Openpose toepassingen

<https://medium.com/beyondminds/an-overview-of-human-pose-estimation-with-deep-learning-d49eb656739b>