

# Kijkrichtingen detecteren door middel van elektromyografie

**Kevin Boets**

Katholieke Universiteit Leuven  
Departement Computerwetenschappen  
kevin.boets@student.kuleuven.be

**Gertjan Franken**

Katholieke Universiteit Leuven  
Departement Computerwetenschappen  
gertjan.franken@student.kuleuven.be

## Abstract

In de zoektocht naar natuurlijke user interfaces gebeurt er onderzoek naar besturing door middel van de ogen. In deze paper maken we gebruik van EMG (Elektromyografie) signalen om eenvoudige oogbewegingen te detecteren. Hiervoor onderzoeken we twee verschillende methodes: thresholds en patronen. In deze tekst bespreken we beide methoden uitvoerig. We hebben in dit onderzoek ondervonden dat de thresholds-methode zeer geschikt kan zijn, indien bepaalde problemen opgelost kunnen worden. Daarnaast zou de patronen-methode ook geschikt kunnen zijn, indien er verder gezocht wordt naar de optimale parameters.

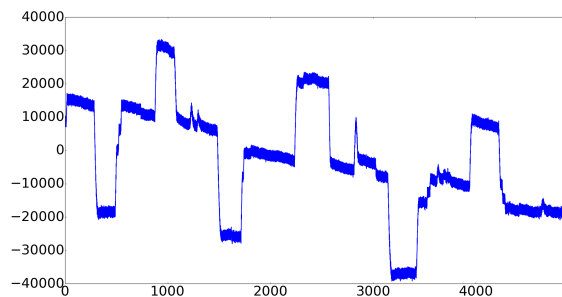
## 1 Introductie

Reeds verschillende manieren om oogbewegingen te detecteren zijn bedacht. Zo kan men al door middel van camera's de ogen volgen. Hierbij gaat men op zoek naar de positie van de iris om de stand van het oog te achterhalen. Deze technieken zijn al ver ontwikkeld, maar bieden nog geen oplossing voor alle problemen. Daarom onderzoeken wij een andere manier. We maken gebruik van sensoren die rond de ogen zijn aangebracht om elektromyografie (EMG) signalen te meten. Aan de hand van deze signalen proberen we te achterhalen in welke richting de gebruiker kijkt. Als invoer krijgen we twee elektrooculogrammen die respectievelijk de kijkrichtingen links-rechts en boven-onder voorstellen. Figuur 1 geeft een mooi voorbeeld weer van het elektrooculogram betreffende de kijkrichtingen links-rechts. Hierbij stellen de dalen de kijkrichting links voor en de bergen de kijkrichting rechts.

## 2 Preprocessing

Het signaal dat we krijgen doorgestuurd, afkomstig van de sensoren, is zeer ruw. Om hier kijkrichtingen uit af te kunnen leiden, gaan we de data eerst beter leesbaar maken. Dit is de preprocessing-stap. Deze stap voeren we uit op elke dataset, alvorens we deze gaan analyseren. Beschouw de data in figuur 1 als de originele data.

Als eerste gaan we de ruis en knipperingen zo goed mogelijk proberen weg te werken, waarna we dit ook proberen voor de biopotentiaalschommelingen. Dit gebeurt telkens



Figuur 1: Originele data rechtstreeks afkomstig van de hardware. De grootte op de x-as is tijd en op de y-as is dit het voltage.

door middel van een low-pass filter. Deze filter verzacht het signaal door de hoogfrequentie te verminderen. Hiervoor gebruiken we de butterworth filter van SciPy voor Python. Door met de parameter  $Wn$  van deze filter te experimenteren, proberen we een optimale filtering te vinden voor het originele signaal.  $Wn$  staat voor de genormaliseerde cutoff frequentie. Aan de hand van deze gekozen variabele, berekenen we dan de cutoff frequentie  $cf$  zoals in vergelijking 1 staat beschreven. Alle frequenties hoger dan deze  $cf$  zullen worden afgezwakt. In deze vergelijking staat  $sf$  voor de sampling frequentie, die geeft aan hoe snel datapunten doorgestuurd worden door de sensoren. Bij ons is dit gelijk aan 25 Hz.

$$cf = \frac{sf \cdot Wn}{2} \quad (1)$$

Voor het filteren van ruis en knipperingen, hebben we een cutoff frequentie  $cf_1$  gevonden. Die wordt gebruikt in de filtering in vergelijking 2, waarbij  $s$  het originele signaal is en  $s'$  het gefilterde signaal. Voor deze filtering gaven we  $Wn_1$  de waarde 0,05, zodat we uitkomen op een waarde van 0,625 Hz voor  $cf_1$ .

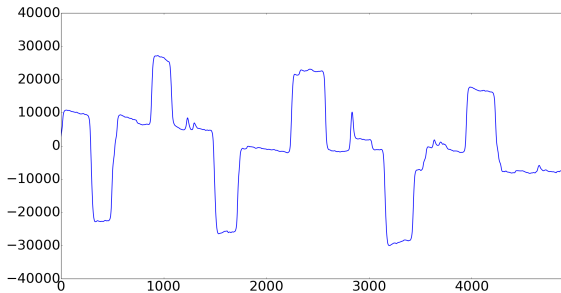
$$s' = \text{lowpass}(s, cf_1) \quad (2)$$

Daarna focussen we ons op de biopotentiaalschommelingen. Deze schommelingen kunnen we niet op voorhand voorspellen en verschillen van persoon tot persoon. Als we deze schommelingen kunnen verminderen,

kunnen we meer steunen op absolute waarden van het signaal. Deze keer gebruiken we een lagere cutoff frequentie  $cf_2$  voor de low-pass filter. Hier hebben we  $W_{n_2}$  de waarde 0,0003 toegekend en verkrijgen we uiteindelijk volgens vergelijking 1 een waarde van 0,004 Hz voor  $cf_2$ . Na deze filtering van de data, blijven enkel de lage frequenties over. Dit is de benadering van de biopotential van de gebruiker die doorheen de tijd fluctueert. We kunnen nu de biopotentialschommelingen uit onze originele data verminderen door de gevonden benadering hiervan af te trekken, zoals in vergelijking 3. In het resulterend signaal,  $s''$ , zijn de schommelingen nog niet volledig verdwenen, maar wel sterk verminderd. Figuur 2 stelt het resultaat van de besproken filteringen voor.

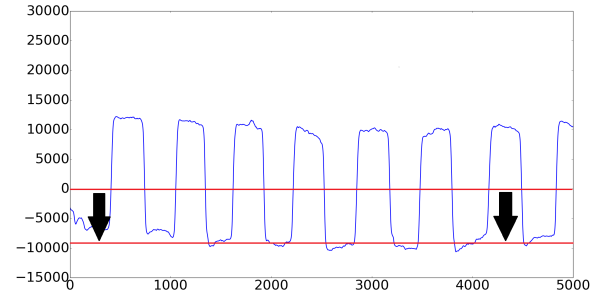
$$s'' = s' - \text{lowpass}(s', cf_2) \quad (3)$$

Bij veelvuldige herhaling van dezelfde kijkrichting, wordt een probleem in verband met deze methode zichtbaar. Als er veel en opeenvolgend naar éénzelfde richting gekeken wordt, zullen de waarden in de data algemeen lager of hoger zijn. De waarden van de gefilterde data zullen hierdoor ook algemeen lager of hoger liggen. Dit is een probleem aangezien het de benadering van de biopotential moet voorstellen. Doordat de gevonden benadering foutief is, zullen ook alle volgende bewerkingen een incorrecte uitkomst hebben. Wanneer we deze benadering aftrekken van de oorspronkelijke data, zal de data ongewenst verschuiven volgens de y-as. Deze verschuiving illustreren we in figuur 3. Hierdoor zal het werken met absolute waarden slechte resultaten opleveren. Omdat de gebruiker zelf kiest welke reeks van kijkrichtingen hij uitvoert, kunnen we deze situatie niet vermijden. We willen dan ook een oplossing vinden voor de gevallen wanneer we steunen op absolute waarden.



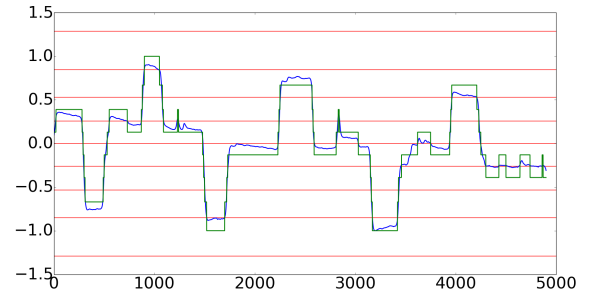
Figuur 2: Het resultaat van de twee filteringen op de originele data. De ruis is sterk verminderd en de sterkte van de knippering is afgenomen. De biopotentialschommelingen zijn niet helemaal weg, maar wel voldoende verminderd.

Na toepassing van de vorige filteringen, discretiseren we de data. Deze stap is eerder optioneel en wordt enkel in de patronen-methode gebruikt. Het grote voordeel van deze stap is dat we minder berekeningen zullen moeten doen, wat de uitvoeringstijd van het programma ten goede komt. Hiervoor gebruiken we de SAX-discretisatie (Symbolic Aggregate apRoXimation) [Keogh *et al.*, 2005] op genormaliseerde data.



Figuur 3: Na aftrek van de foutieve benadering van de biopotential, zien we dat de waarden ongewenst afgenomen zijn. Hierdoor is recht door kijken niet meer rond de  $y = 0$  lijn, maar wel veel lager rond de  $y = -10000$  lijn.

We definiëren op voorhand hoeveel waarden een letter voor moet stellen. Dit is de constante  $w$ . De data die we willen discretiseren, verdelen we op in partities van lengte  $w$ . Elke partitie zal een bepaalde letter krijgen, afhankelijk van zijn gemiddelde. Het aantal verschillende letters dat we gebruiken, hangt af van de alfabetgrootte  $a$ . Volgens de normale verdeling wordt nu elke partitie aan een letter gekoppeld. Zie figuur 4 voor de visuele werking. Eigenlijk stelt elke letter nu een reeks van waarden voor. Om SAX-woorden voor te kunnen stellen in grafieken, geven we elke letter ook een waarde. Dit is het gemiddelde van de intervalgrenzen die de letter op zich neemt.



Figuur 4: De groene lijn stelt de discretisatie voor van de gefilterde data die in het blauw staat. De rode horizontale lijnen geven de distributie weer van de SAX-letters. Deze distributie is volgens de normale verdeling met gemiddelde 0 en standaarddeviatie 1.

### 3 Gebruikte methoden

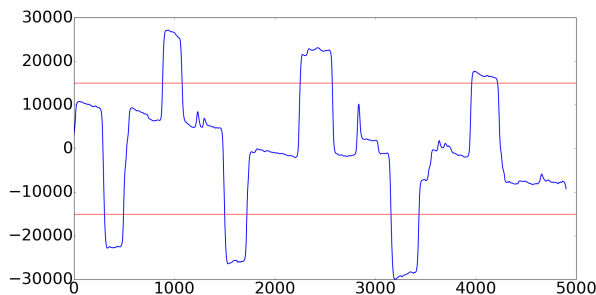
In ons onderzoek hebben we gebruik gemaakt van twee verschillende methoden om kijkrichtingen te herkennen. Deze methoden steunen respectievelijk op thresholds en patronen. Beide methoden beginnen met een calibratie-fase, gevolgd door een herkenning-fase. In de calibratie-fase laten we de gebruiker naar een aantal richtingen kijken, waarna we deze data analyseren. De gevonden informatie koppelen we aan de

juiste kijkrichting. In de herkenning-fase wordt deze informatie gebruikt om in de nieuwe data kijkrichtingen te detecteren.

### 3.1 Thresholds

De eenvoudigste methode is de methode gebruikmakend van thresholds. Het steunt vooral op de absolute waarden van de data. Het basis idee van deze methode is het volgende. Er wordt een bovengrens gedefinieerd voor elke kijkrichting in de calibratie-fase. Als deze overschreden wordt in de herkenning-fase, gaan we er van uit dat er in de overeenkomstige kijkrichting gekeken wordt.

In de calibratie-fase wordt aan de gebruiker gevraagd om een bepaalde volgorde van kijkrichtingen uit te voeren. Vervolgens voeren we de preprocessing-stap uit in verband met de ruis, knippering en biopotentiaal. Hier maken we geen gebruik van de SAX-discretisatie omdat deze methode al efficiënt genoeg is. De preprocessed dataset gebruiken we dan om goede bovengrenzen te vinden. Figuur 5 illustreert een voorbeeld van welke bovengrenzen gebruikt kunnen worden op het signaal dat we eerder al hadden besproken.



Figuur 5: In deze afbeelding stellen de rode horizontale lijnen mogelijke bovengrenzen voor.

Eens we de bovengrenzen hebben gevonden, kunnen we deze in de herkenning-fase gebruiken om kijkrichtingen te detecteren. Om deze gevonden bovengrenzen te kunnen gebruiken, zullen we eerst dezelfde preprocessing stappen uitvoeren op de nieuwe data. Belangrijk is dat we het signaal ten allen tijde stabiel houden, we werken immers met absolute waarden.

Om kijkrichtingen te detecteren, gaan we de bovengrenzen vergelijken met de waarden uit de data. Wanneer een bovengrens overschreden wordt, toont dit aan dat de gebruiker mogelijk in een van de kijkrichtingen aan het kijken is. Dit kunnen we echter niet zomaar aannemen. Een enkele piek, zoals een knippering, zou dan ook ongewild herkend kunnen worden als kijkrichting. Om voor meer zekerheid te zorgen, wachten we tot dezelfde grens opeenvolgend meerdere keren overschreden is. Hoeveel keren definiëren we op voorhand. Het aantal dat we kiezen is een afweging tussen detectiekans en zekerheid. Bij een klein aantal is de kans op detectie hoog, maar zijn we minder zeker over de geldigheid van de detectie. Een groot aantal zal anderzijds veel zekerheid verschaffen, waarbij er wel een grotere kans is dat korte kijkrichtingen

niet worden gedetecteerd. Hierbij is een goede balans belangrijk.

### 3.2 Patronen

In tegenstelling tot de thresholds-methode steunt de patronen-methode weinig op absolute waarden, maar eerder op relatieve waarden. We gaan gebruik maken van sequenties, dit zijn opeenvolgingen van punten. We nemen telkens sequenties van lengte 100 uit de verkregen data. Met deze sequenties wordt er naar veel voorkomende patronen gezocht in de calibratie-fase, die dan in de herkenning-fase met de nieuwe data vergeleken worden [Yankov *et al.*, 2007]. Deze patronen worden motieven genoemd. In deze methode maken we gebruik van de SAX-discretisatie om de efficiëntie te verhogen.

Ook hier wordt in de calibratie-fase aan de gebruiker gevraagd om in een aantal kijkrichtingen te kijken. Voor elke kijkrichting hebben we nu een aantal datasets van signalen die we met elkaar gaan vergelijken. Voor elke kijkrichting zoeken we de meest succesvolle motieven. Dit zijn de motieven die het meest aantal matches hebben. We zeggen dat twee sequenties met elkaar matchen indien deze voldoende veel op elkaar lijken. Hiervoor moeten specifieke voorwaarden gedefinieerd worden. Dit doen we voor ons programma in de implementatie-sectie.

In de paper [Yankov *et al.*, 2007] gebruikt men collision matrices om zo efficiënt mogelijk de motieven te vinden. Hierop hebben we ons gebaseerd en dit wordt dan ook uitgebreid uitgelegd in de implementatie-sectie.

De gevonden motieven, gekoppeld aan de bijhorende kijkrichtingen, worden doorgegeven aan de herkenning-fase. De binnekomende data wordt verdeeld in sequenties die omgezet worden in SAX-woorden. Pas als er voldoende overeenkomst is tussen een binnengekomen SAX-woord en een SAX-woord van een motief, vergelijken we deze sequenties op preciezer niveau. Als deze sequenties weer goed genoeg op elkaar lijken, vermoeden we dat de kijkrichting gekoppeld aan het herkende motief is uitgevoerd.

## 4 Implementatie

Bij het implementeren van ons programma, hebben we ons gebaseerd op de twee eerder uitgelegde methoden. In deze sectie leggen we uit hoe we de implementatie hebben verwezenlijkt. Hierbij geven we ook weer hoe we bepaalde problemen hebben opgelost.

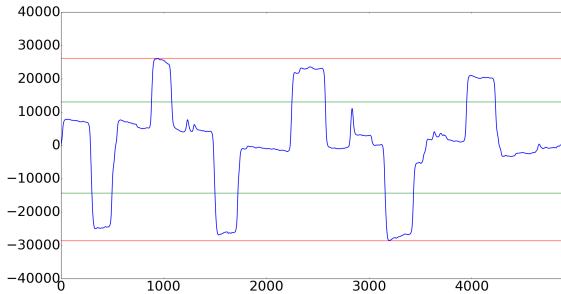
In beide methodes gebruiken we een buffer die de 500 recentste datapunten bijhoudt. Om de 10 datapunten die we ontvangen van de sensoren, wordt deze buffer geüpdatet. De 10 oudste punten worden verwijderd en de 10 recentste punten worden vooraan toegevoegd. Elke keer als dit gebeurt, voeren we ook de preprocessing-stap uit op de data in de buffer. Deze buffer wordt ook getoond aan de gebruiker tijdens de uitvoering van het programma. In het begin van het programma wordt ook altijd een buffer fill uitgevoerd. Hierbij laten we de gebruiker voor zich uit kijken tot dat de buffer helemaal gevuld is. Dit duurt 21 seconden.

### 4.1 Thresholds

In de calibratie-fase van de thresholds-methode gaan we dus op zoek naar goede bovengrenzen. Nadat de preprocessing-

stap is uitgevoerd op de dataset, halen we er de minimale en maximale waarde uit. Deze twee waarden horen elk bij een andere kijkrichting. Per kijkrichting hebben we ook een constante  $\alpha$  waarvoor geldt:  $0 < \alpha \leq 1$ . We vermenigvuldigen  $\alpha$  met het minimum of maximum (afhankelijk van de kijkrichting) en beschouwen de uitkomst als de uiteindelijke bovengrenzen voor de kijkrichting. Dit wordt aangetoond in figuur 6.

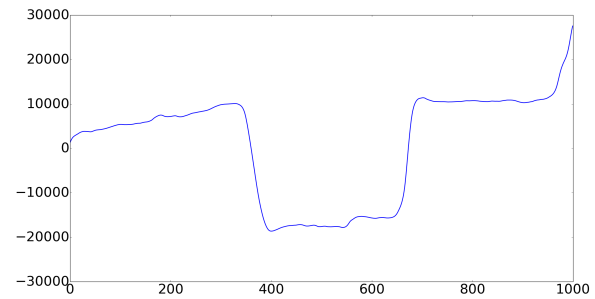
In ons programma hebben we  $\alpha$  voor beide kijkrichtingen de waarde 0,5 gegeven, maar deze kan veranderd worden naar behoefte. Zo kunnen we de methode bijvoorbeeld strenger maken door  $\alpha$  te verhogen.



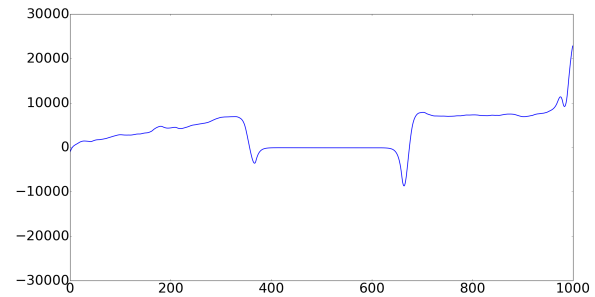
Figuur 6: In deze afbeelding stellen de rode lijnen het minimum en maximum voor. De groene lijnen stellen de bovengrenzen voor. Dit is telkens het minimum en maximum vermenigvuldigd met  $\alpha$ .

Nu we de bovengrenzen hebben gevonden, moeten we kijkrichtingen gaan zoeken in de nieuwe data. We vragen telkens de 10 recentste datapunten op van de buffer. Om de invloed van een enkele uitschieter in de data te beperken, nemen we telkens het gemiddelde van deze 10 punten. Dit gemiddelde gaan we dan vergelijken met alle bovengrenzen. Slechts wanneer de grenzen 3 opeenvolgende keren overschreden worden, zullen we beslissen dat er een kijkrichting is gevonden.

Omdat we telkens opnieuw de preprocessing-fase uitvoeren op de bufferdata, moeten we oppassen voor het probleem bij veelvuldige herhaling van dezelfde kijkrichting. Zoals eerder vermeld in de thresholds-sectie van gebruikte methoden, zorgt dit voor een verschuiving van het signaal ten opzichte van de y-as. Aangezien we in de herkenning-fase weinig controle hebben over de volgorde van kijkrichtingen, zullen we dit probleem hier moeten oplossen. Hiervoor gebruiken we afvlakking van het signaal. Telkens er een threshold overschreden wordt, gaan we het gemiddelde van de vorige 490 datapunten berekenen. Vervolgens vervangen we de originele grensoverschrijdende waarden door dit gemiddelde. Door deze afvlakking, zal de benadering van de biopotentiaal eerder stabiel zijn. Het nadeel van deze techniek is dat de data destructief wordt aangepast en zo mogelijk data verloren gaat. Dit heeft echter geen invloed op dit systeem, aangezien we enkel geïnteresseerd zijn in de nieuwste data. In de figuren 7 en 8 is te zien hoe het signaal verandert onder afvlakking.



Figuur 7: Dit is een voorbeeld van een gefilterd signaal waarbij geen afvlakking op uitgevoerd is.

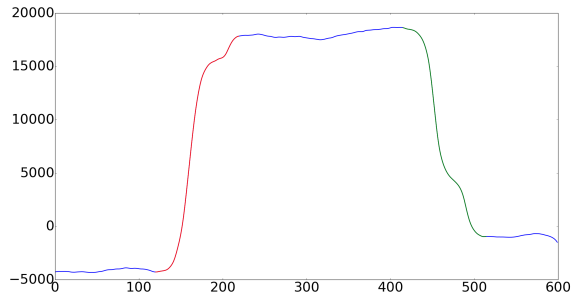


Figuur 8: Dit is hetzelfde gefilterd signaal uit figuur 7 waarbij wel een afvlakking op is uitgevoerd. De waarden in deze grafiek zijn nu minder extreem en liggen dichterbij 0.

## 4.2 Patronen

In de calibratie-fase beschikken we over verschillende datasets die gelinkt zijn aan een bepaalde kijkrichting. Voor elke kijkrichting zoeken we één paar van motieven. Deze motieven noemen we het beginmotief en het eindmotief. Een voorbeeld hiervan is te zien in figuur 9. Deze motieven nemen elk een deel van de hele kijkrichting voor hun rekening. Om telkens zo een paar te verkrijgen, leggen we een aantal voorwaarden op. Zo mogen de motieven elkaar nooit overlappen en moet elk motief een verschil in hoogte bestrijken. Dit laatste zorgt ervoor dat we nooit quasi horizontale lijnen, die weinig informatie bevatten, als motief zullen vinden.

Voordat we de echte sequenties met elkaar gaan vergelijken, doen we dit eerst met hun genormaliseerde SAX-woorden. Zo kunnen we, zonder al te veel berekeningen, achterhalen welke sequenties in aanmerking komen voor een match. Hiervoor gebruiken we 10 maskers die we op voorhand hebben gedefinieerd. Zo een masker dekt telkens enkele letters van twee te vergelijken SAX-woorden af. Bij het handmatig opstellen van de maskers, hebben we geprobeerd elke letter even belangrijk te maken. Dit doen we door elke letter even vaak zichtbaar te houden. Een voorbeeld voor een masker is [1,1,0,0], het houdt enkel de eerste twee letters zichtbaar. Zie (4) voor werking en voorbeelden.



Figuur 9: In deze dataset is het beginmotief aangeduid in het rood en het eindmotief in het groen.

Tabel 1: Dit is een voorbeeld van een complete collision matrix. De vier SAX-woorden van verschillende sequenties staan bij een kolom en rij. De matrix is maar half ingevuld omdat hij symmetrisch is. De diagonaal is leeg, omdat sequenties niet met zichzelf worden vergeleken. De gebruikte maskers staan onder de tabel.

	accd	dcda	acda	acda
accd	/	0	1	1
dcda	/	/	2	2
acda	/	/	/	4
acda	/	/	/	/

$[1, 1, 0, 0]$ ,  $[0, 0, 1, 1]$ ,  $[1, 0, 0, 1]$ ,  $[0, 1, 1, 0]$

$$\begin{aligned}
 abcd \times [1, 1, 0, 0] &= ab \\
 abcd \times [0, 1, 1, 0] &= bc \\
 abcd \times [1, 0, 0, 1] &= ad
 \end{aligned} \tag{4}$$

Deze maskers gebruiken we bij het opstellen van de collision matrix. Een voorbeeld hiervan is te zien in tabel 1. Elke cel in deze matrix hoort bij een paar van sequenties. De waarde in een cel geeft weer hoe goed de bijhorende sequenties op elkaar lijken. We beginnen met een lege collision matrix. Eerst nemen we het eerste masker en dekken er de SAX-woorden van alle sequenties mee af. Dan kijken we naar elk paar van sequenties en indien de gemaskeerde woorden hetzelfde zijn, incrementeren we de waarde in de bijhorende cel. Dit herhalen we voor elk masker. Daarna kijken we naar de waarden uit de gevulde collision matrix. Voor elke cel met een waarde groter dan of gelijk aan de *collision threshold* komt het bijhorende paar in aanmerking als match. Elk paar van sequenties dat in aanmerking komt om een match te zijn, wordt dan uiteindelijk echt vergeleken. Dit gebeurt door de euclidische afstand tussen de genormaliseerde sequenties te berekenen. Pas als deze afstand kleiner is dan een vooraf gedefinieerde variabele *range*, spreken we van een match.

Nu we alle matches hebben gevonden, kunnen we elke sequentie aan een lijst met al zijn matches koppelen. Met behulp van deze lijsten, kunnen we deze sequenties rangschikken. We sorteren alle sequenties op het aantal matches dat ze hebben, met de sequentie met de meeste matches vooraan. In het geval dat twee sequenties evenveel matches zouden heb-

ben, krijgt de sequentie met de kleinste totale euclidische afstand tot zijn matches voorrang.

In de herkennings-fase worden, na elke nieuwe toevoeging van tien datapunten, de laatste honderd datapunten opgevraagd. Deze honderd datapunten vormen een sequentie. Net zoals in de calibratie, wordt ook voor deze sequenties gecontroleerd of ze wel een groot genoeg hoogteverschil hebben. De sequentie wordt genormaliseerd en gediscrètiseerd naar een SAX-woord. Met behulp van de eerder gebruikte maskers wordt nu ook nagegaan of de sequentie goed genoeg lijkt op een motief. Telkens beide gemaskeerde SAX-woorden gelijk zijn, spreken we van een collision. Indien het aantal collisions groter is dan de *collision threshold*, worden de sequenties zelf weer met elkaar vergeleken. Dit gebeurt ook hier met behulp van de euclidische afstand. Indien de afstand voldoet aan de eerder vermelde *range*, wordt het label van het motief gegeven aan deze sequentie. Dit betekent dat we het motief hebben herkend.

Pas wanneer achtereenvolgens het beginmotief en het eindmotief herkend worden, gaan we uit van een kijkrichting. Indien het beginmotief herkend wordt en de herkenning van het eindmotief op zich laat wachten, zal een timer ervoor zorgen dat er niet oneindig lang gewacht wordt op de herkenning van het eindmotief.

## 5 Experimenten

Bij het uitvoeren van experimenten hebben we ons vooral geconcentreerd op de thresholds-methode. Aangezien de patronen-methode zoveel afhangt van verschillende parameters, hebben we dit niet kunnen optimaliseren. Hierom hebben we op deze methode geen experimenten uitgevoerd, zodat we geen vertekend beeld zouden scheppen over de mogelijkheden van patronen.

De opstelling van de experimenten zag er als volgt uit. De sensoren zijn bevestigd op het gezicht van de gebruiker door middel van stevige plakkers. Deze moeten blijven plakken zonder tussenkomst van de handen. De sensoren die verantwoordelijk zijn voor het meten van de kijkrichtingen links en rechts, zijn bevestigd op de slapen, tegen de ooghoeken. De sensoren verantwoordelijk voor de kijkrichtingen boven en onder hebben we niet gebruikt in deze experimenten. De gebruiker zit op een stoel en kijkt voor zich uit. Hierbij is het ook belangrijk dat de gebruiker niet afgeleid kan worden door bewegende voorwerpen. Het gebruiken van andere gezichtsspieren dan de oogspieren moet zo veel mogelijk vermeden worden. De sensoren vangen de signalen van deze spieren namelijk ook op. Lachen kan bijvoorbeeld niet, rustig praten wel zolang het niet zichtbaar is in het gemeten signaal.

De sensoren zijn aangesloten op een EOG Bord, gemaakt door InnovationLab van KU Leuven. Dit apparaat voert al een naïve biopotentialfiltering uit op het signaal. Het vlak het signaal altijd af naar de neutrale waarde (de waarde verkregen bij het rechtdoor kijken). Hierdoor wordt de intensiteit van het kijken naar links en rechts geleidelijk aan afgezwakt.

We hoeven niet te weten wat er gedurende een tijdsinterval gebeurt, maar we concentreren ons eerder op bepaalde events. We hebben drie events, namelijk de volgende kijkrichtingen: *links*, *rechts* en *rechtdoor*. We duiden reeksen



van kijkrichtingen aan volgens omschrijving 5. We bepalen drie verschillende kijkrichtingen elk met een tijd. Deze tijd  $t_i$  staat voor het aantal seconden dat er ongeveer in de bijhorende kijkrichting gekeken wordt.  $h$  geeft aan hoeveel keer de reeks binnen de accolades herhaald wordt.

$$\begin{aligned} & \text{rechtdoor}(t_1), \text{links}(t_2), \text{rechts}(t_3) \\ & \{A, B\} \times h = A_1, B_1, A_2, B_2, \dots, A_h, B_h \end{aligned} \quad (5)$$

In de calibratie-fase wordt er telkens volgens reeks 6 gekeken.

$$\begin{aligned} & \{\text{rechtdoor}(1), \text{links}(1), \\ & \text{rechtdoor}(1), \text{rechts}(1)\} \times 2, \\ & \text{rechtdoor}(1) \end{aligned} \quad (6)$$

Resultaten verwerken we door middel van een confusion matrix, zie tabel 2. Elke rij en kolom stelt een kijkrichting voor. Aan de hand van deze matrix berekenen we de *accuracy* (7), *recall<sub>k</sub>* (8) en *precision<sub>k</sub>* (9). Hierbij staat  $k$  voor de relevante kijkrichting. *accuracy* staat voor de verhouding tussen de correct herkende kijkrichtingen en alle te herkennen kijkrichtingen. *recall<sub>k</sub>* staat voor de verhouding tussen de correct herkende kijkrichtingen  $k$  en alle werkelijk gekeken kijkrichtingen  $k$ . *precision<sub>k</sub>* staat voor de verhouding tussen correct herkende kijkrichtingen  $k$  en alle herkende kijkrichtingen  $k$ . De kijkrichting *rechtdoor* geven we een klein gewicht van 1/10 mee, omdat we hier weinig in geïnteresseerd zijn. Anders zou dit ook een vertekend beeld kunnen scheppen, omdat deze kijkrichting zo vaak voorkomt.

Tabel 2: Deze tabel stelt de confusion matrix voor. Als de gebruiker bijvoorbeeld naar *links* kijkt, maar we detecteren de kijkrichting *rechts*, zal de cel d met 1 verhoogd worden. Op de diagonaal (de cellen a, e en i) staan alle succesvolle voorspellingen.

Herkend \ Echt	<i>links</i>	<i>rechts</i>	<i>rechtdoor</i>
<i>links</i>	a	b	c
<i>rechts</i>	d	e	f
<i>rechtdoor</i>	g	h	i

$$\text{accuracy} = \frac{a + e + \frac{i}{10}}{a + b + c + d + e + f + g + h + \frac{i}{10}} \quad (7)$$

$$\text{recall}_{\text{links}} = \frac{a}{a + d + g} \quad (8)$$

$$\text{recall}_{\text{rechts}} = \frac{e}{b + e + h}$$

$$\text{precision}_{\text{links}} = \frac{a}{a + b + c} \quad (9)$$

$$\text{precision}_{\text{rechts}} = \frac{e}{d + e + f}$$

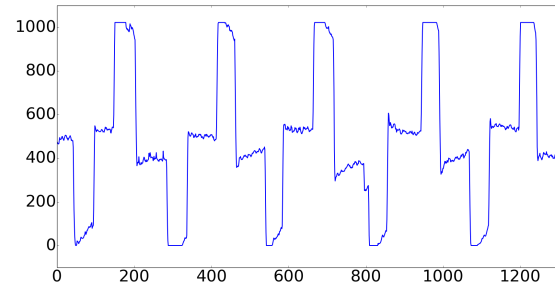
Deze testen zijn uitgevoerd op twee verschillende personen. We noemen ze *gebruiker 1* en *gebruiker 2*. We laten de gebruikers elke reeks vijf keer uitvoeren, om zo een betrouwbaar beeld te krijgen van de prestaties van het programma.

## 5.1 Experiment 1

In dit experiment laten we een gebruiker reeks 10 uitvoeren. Figuur 10 toont een voorbeeld van hoe het signaal er uitziet voor deze reeks.

$$\begin{aligned} & \{\text{rechtdoor}(5), \text{links}(2)\} \times 5, \\ & \{\text{rechtdoor}(5), \text{rechts}(2)\} \times 5, \\ & \text{rechtdoor}(5) \end{aligned} \quad (10)$$

De gebruiker moet volgens reeks 10 afwisselend links en rechts kijken. We gebruiken eerst deze volgorde, omdat deze zoals eerder vermeld geen problemen oplevert in de preprocessing-stap. We willen eerst weten hoe goed het systeem werkt in ideale situaties, voordat we de minder ideale situaties bekijken. Deze resultaten hebben we in een confusion matrix, tabellen 3 en 4, weergegeven.



Figuur 10: Een voorbeeld signaal bij uitvoering van experiment 1. Er is duidelijk te zien dat er telkens van kijkrichting gewisseld wordt.

Tabel 3: Confusion matrix na uitvoering van experiment 1 door *gebruiker 1* en de berekende maatstaven.

Herkend \ Echt	<i>links</i>	<i>rechts</i>	<i>rechtdoor</i>
<i>links</i>	18	0	0
<i>rechts</i>	0	25	0
<i>rechtdoor</i>	7	0	55

$$\text{accuracy} = 0,87$$

$$\text{precision}_{\text{links}} = 1 \quad \text{recall}_{\text{links}} = 0,72$$

$$\text{precision}_{\text{rechts}} = 1 \quad \text{recall}_{\text{rechts}} = 1$$

De precision van beide kijkrichtingen, links en rechts, zijn 1. Dit geldt bij beide gebruikers. Hierdoor weten we dat het systeem nooit incorrect links of rechts herkend heeft. Bij de *recall* merken we wel dat er voor *gebruiker 1* problemen waren bij het herkennen van van links. Voor *gebruiker 2* waren er echter vooral problemen bij het herkennen van links. Hieruit kunnen we afleiden dat een kleine verandering in omstandigheden een verschil maakt voor de *recall*, maar niet voor de precision. Anderzijds merken we op dat er geen fouten zijn gebeurd telkens er rechtdoor werd gekeken. De *recall* voor

rechtdoor is bijgevolg 1. Ook zien we dat beide gebruikers toevallig exact dezelfde *accuracy* hebben. Deze *accuracy* van 0.87 is al vrij goed, maar betekent wel dat er nog verbetering mogelijk is.

Tabel 4: Confusion matrix na uitvoering van experiment 1 door *gebruiker 2* en de berekende maatstaven.

Herkend \ Echt	links	rechts	rechtdoor
links	24	0	0
rechts	0	19	0
rechtdoor	1	6	55

$$accuracy = 0,87$$

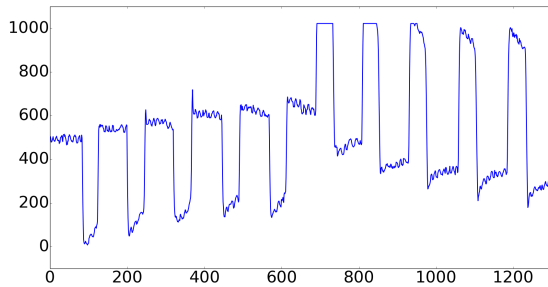
$$precision_{links} = 1 \quad recall_{links} = 0,96$$

$$precision_{rechts} = 1 \quad recall_{rechts} = 0,76$$

## 5.2 Experiment 2

Nu we weten hoe goed thresholds werken in de ideale omstandigheden, gaan we in het tweede experiment een minder ideale situatie bekijken. We laten de gebruiker meerdere opeenvolgende keren naar dezelfde richting kijken, volgens reeks 11. Figuur 11 geeft een voorbeeld van hoe het signaal er uit ziet. Zoals we in de preprocessing-sectie hebben vermeld, zal deze volgorde voor problemen kunnen zorgen in de preprocessing-stap.

$$\begin{aligned} &\{rechtdoor(5), links(2) \\ &rechtdoor(5), rechts(2)\} \times 5, \\ &rechtdoor(5) \end{aligned} \quad (11)$$



Figuur 11: Een voorbeeld signaal bij uitvoering van experiment 2. Er is duidelijk te zien dat er eerst naar links werd gekeken en daarna naar rechts.

Dit experiment heeft veel minder goede resultaten dan het eerste experiment, zie tabellen 5 en 6. De *accuracy* en *recall* zijn allebei duidelijk slechter. Dit kunnen we verklaren aan de hand van figuur 11. Er is duidelijk een verschuiving volgens de y-as te zien, maar dit is een gevolg van de biopotentialfiltering van de sensoren zelf. Door het geleidelijk afzwakken van het signaal terwijl er naar links of rechts gekeken wordt, zal bij het terug rechtdoor kijken het signaal niet rond

Tabel 5: Confusion matrix na uitvoering van experiment 2 door *gebruiker 1* en de berekende maatstaven.

Herkend \ Echt	links	rechts	rechtdoor
links	16	0	5
rechts	0	14	0
rechtdoor	9	11	50

$$accuracy = 0,58$$

$$precision_{links} = 0,76 \quad recall_{links} = 0,64$$

$$precision_{rechts} = 1 \quad recall_{rechts} = 0,56$$

Tabel 6: Confusion matrix na uitvoering van experiment 2 door *gebruiker 2* en de berekende maatstaven.

Herkend \ Echt	links	rechts	rechtdoor
links	17	0	1
rechts	0	8	0
rechtdoor	8	17	54

$$accuracy = 0,54$$

$$precision_{links} = 0,94 \quad recall_{links} = 0,68$$

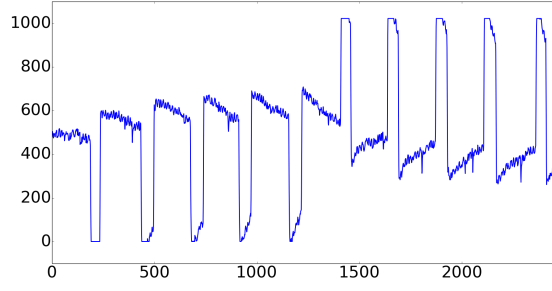
$$precision_{rechts} = 1 \quad recall_{rechts} = 0,47$$

de neutrale lijn liggen. Het zal eerder uitwijken naar de tegenovergestelde kijkrichting, waardoor het moeilijker wordt om herhaalde kijkrichtingen te herkennen. Dit verklaart de lage *recall* voor beide kijkrichtingen. Hieruit leiden we af dat deze naïve biopotentialfiltering niet de beste oplossing is voor alle situaties.

## 5.3 Experiment 3

Zoals we al vermeld hebben, probeert het apparaat dat we gebruiken de biopotentialschommelingen te filteren door telkens het signaal langzaam af te vlakken naar de neutrale staat (de staat wanneer er rechtdoor gekeken wordt). Wanneer er bijvoorbeeld lang naar links gekeken wordt, verschuift het signaal volgens de y-as. Als er nu terug rechtdoor gekeken wordt, lijkt het alsof er naar rechts gekeken wordt. We vermoeden dat als tussen het links en rechts kijken voor een langere periode rechtdoor gekeken wordt, dit effect teniet wordt gedaan. Dit hebben we getest in dit experiment, waarbij reeks 12 werd uitgevoerd door de gebruiker. Zie figuur 12 als voorbeeld.

$$\begin{aligned} &\{rechtdoor(10), links(3) \\ &rechtdoor(10), rechts(3)\} \times 5, \\ &rechtdoor(10) \end{aligned} \quad (12)$$



Figuur 12: Een voorbeeld signaal bij uitvoering van experiment 3. We merken op dat door langer rechtdoor te kijken tussen andere kijkrichtingen in, het besproken effect niet meer voorkomt.

Tabel 7: Confusion matrix na uitvoering van experiment 3 door gebruiker 1 en de berekende maatstaven.

Herkend \ Echt	links	rechts	rechtdoor
links	24	0	1
rechts	0	20	0
rechtdoor	1	5	54

$$accuracy = 0,88$$

$$precision_{links} = 0,96 \quad recall_{links} = 0,96$$

$$precision_{rechts} = 1 \quad recall_{rechts} = 0,8$$

Tabel 8: Confusion matrix na uitvoering van experiment 3 door gebruiker 2 en de berekende maatstaven.

Herkend \ Echt	links	rechts	rechtdoor
links	25	0	4
rechts	0	15	1
rechtdoor	0	10	50

$$accuracy = 0,75$$

$$precision_{links} = 0,86 \quad recall_{links} = 1$$

$$precision_{rechts} = 0,94 \quad recall_{rechts} = 0,6$$

De resultaten zijn fel verbeterd ten opzichte van experiment 2, zie tabellen 7 en 8. Hier vinden we de verbetering van de *recall* voor beide kijkrichtingen het interessantst. Dit bevestigt ons vermoeden. Er vindt namelijk een kleinere verticale verschuiving plaats, waardoor er minder kijkrichtingen gemist worden door het programma. De *recall<sub>rechts</sub>* waarde van gebruiker 2 is bijna niet verbeterd, maar we vermoeden dat dit aan andere factoren ligt, zoals de plaats van de sensoren op het gezicht. De potentiaalverschillen gemeten bij deze gebruiker waren namelijk veel kleiner dan bij gebruiker 1. Uit dit experiment hebben we afgeleid dat langere pauzes nemen bij het herhalen van kijkrichtingen, het besproken effect vermindert.

## 6 Conclusies

We hebben de mogelijkheid onderzocht om kijkrichtingen te detecteren door middel van EMG signalen. Dit kan gebruikt worden in toepassingen omtrent natural user interfaces. We onderzochten twee verschillende methodes.

De thresholds-methode hebben we uitvoerig besproken en getest. Hierbij bleek het grote probleem de filtering van de biopotentiaalfluctuaties te zijn. Deze filtering kon de absolute waarden ongewild wijzigen. Hiervoor hebben we een oplossing bedacht, namelijk afvlakking. De testen voor deze oplossing gaven een licht vertekend beeld omwille van de hardware die al deels op een naïve manier een filtering uitvoerde. We zijn van mening dat indien deze problemen omtrent de biopotentiaalfluctuaties onder controle gehouden kunnen worden, deze methode zeer accuraat kan zijn.

De patronen-methode hebben we uitvoerig besproken, maar niet getest. Door de vele parameters waarvan deze methode afhankelijk is, is het moeilijk een optimale configuratie samen te stellen. We hebben ook gemerkt dat deze methode meer intensief is dan de thresholds-methode.

Volgens ons is de thresholds-methode beter geschikt voor het herkennen van kijkrichtingen, als de besproken problemen verder onderzocht worden. De patronen-methode lijkt ons voor deze applicatie te intensief en kan misschien beter gebruikt worden voor problemen waarbij accuraatheid van groter belang is.

## Acknowledgements

We zouden graag Dr. Wannes Meert en Prof. Dr. Luc de Raedt willen bedanken voor hun begeleiding. Daarnaast danken we ook InnovationLab van KU Leuven voor het ter beschikking stellen van de gebruikte sensoren en toebehoren.

## Referenties

- [Keogh *et al.*, 2005] Eamonn Keogh, Jessica Lin, and Ada Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 226–233, November 2005.
- [Yankov *et al.*, 2007] Dragomir Yankov, Eamonn Keogh, Jose Medina, Bill Chiu, and Victor Zordan. Detecting time series motifs under uniform scaling. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 844–853, August 2007.