

NAO als dansdocent

DAVID KNIGGE, MARCEL VELEZ, JOCHEM DOBBER, DAVID VOS

Zoeken, Sturen en Bewegen

30 juni 2017

I. INLEIDING

Menselijke poses vinden hun meest extreme vorm in de danswereld. Dansen vereist dat op hoog tempo van posities gewisseld kan worden, posities die uniek zijn voor een specifieke dans en niet regelmatig door mensen in hun dagelijks leven worden aangenomen.

Gepassioneerde mensen zijn vaak jaren ge-disciplineerd bezig met het tot in de perfectie ontwikkelen van de juiste danstechniek. Vele uren worden samen met een dansleraar doorgebracht, met als doel het perfectioneren van poses en de overgangen daartussen. Er komt echter een moment dat deze dansleraar niet meer van toegevoegde waarde kan zijn voor een leerling, omdat deze simpelweg zijn plafond heeft bereikt. Een nieuwe dansleraar met meer expertise kan de weg naar een verdere doorontwikkeling zijn, de kosten hiervoor zijn voor veel mensen echter te hoog. Een constatering die gelijkstaat aan veel verloren dansalent.

Humanoids zijn robots die gebaseerd zijn op het menselijk skelet. Door qua uiterlijk en mogelijke bewegingen een mens te kopiëren kan zo'n robot praktisch ingezet worden voor taken waar normaal gesproken menselijke poses en interactie voor nodig zijn. Een humanoid zou zo ook de rol van dansleraar op zich kunnen nemen, door dansbewegingen te interpreteren en voor te doen aan een leerling. Als een robot bewegingen van de meest gerenommeerde dansers ter wereld over zou nemen, dan zou deze zijn leerlingen dus zelfs verder kunnen helpen dan veel menselijke dansleraren die op dit moment actief zijn. Volgens Ioannou et al [1] zijn humanoid robots geschikt voor het on-

derwijzen van in het bijzonder kinderen. Jonge kinderen zien de robot als een van hen, en vooral bij activiteiten waar de humanoid fysiek beweegt kunnen de kinderen hun aandacht goed vasthouden.

Ioannou et al [1] spreken dus van een positieve relatie tussen humanoid robots in de educatie en de aandacht kinderen. Over hoe deze relatie zicht vertaalt naar (jong-) volwassenen is nog weinig bekend. Door het inzetten van een robot als dansleraar voor volwassenen kan dit onderzocht worden.

De hoofdvraag van dit onderzoek luidt dan ook: hoe kan een humanoid robot het beste geprogrammeerd kan worden als dansdocent? Deze vraag kan worden opgedeeld in twee subvragen. Hoe kunnen de bewegingen van een dansdocent worden opgeslagen, en hoe kan een humanoid deze bewegingen imiteren? Nu volgt een korte schets van de resultaten die bij deze deelvragen verwacht worden.

De robot NAO heeft 26 gewrichten tot zijn beschikking, om zijn bewegingen zo menselijk mogelijk te laten lijken. Het menselijk lichaam telt een aanzienlijk aantal meer gewrichten. De Nao is in staat de belangrijkste gewrichten te simuleren. Zesentwintig vrijheidsgraden die er voor zouden kunnen dat deze humanoid praktisch ingezet kan worden als dansleraar. Elk gewricht neemt een hoek in radialen als input en verstelt zich naar deze hoek. De data die gebruikt zal worden voor het onderzoek kan uit bestaande databases met hoeken voor de gewrichten worden gehaald, maar zal uiteindelijk zelf opgenomen moeten worden door professionele dansers. De NAO beschikt zelf over een camera om de omgeving te scannen, echter is deze niet geavanceerd genoeg om bewegingen van mensen te analyseren. Dit opnemen zal

door een externe sensor genaamd Kinect gebeuren. In combinatie met de Kinect kan een SDK (Software Development Kit) gebruikt worden, die het skelet van een mens kan volgen. Door het implementeren van deze techniek, genaamd human skeleton tracking, kan data van de positie van een persoon worden gelezen. Deze data is in de vorm van quaternions, een uitbreiding van de complexe getallen. Uit deze quaternions kunnen de euler hoeken worden berekend. De euler hoeken bestaan uit de roll, de pitch en de yaw van een bepaald gewricht. De zijn de drie draaisassen waar vanaf een bepaald punt in een 3D wereld om gedraaid kan worden.

Feedback geven kan door met de Kinect de bewegingen van de leerling te scannen, waarbij constant de posities van zijn ledematen worden vergeleken met de posities van de professionele danser. In de NAO zit een functie waardoor tekst, en dus feedback, kan worden uitgesproken naar de leerling. Ook andersom kan de leerling om feedback van de NAO vragen, door de ingebouwde spraakherkenning van de robot.

II. METHODE

Om te onderzoeken hoe de NAO soepel menselijke bewegingen van externe bronnen kan overnemen kan gebruik gemaakt worden van een database gevuld met hoeken in radialen. De Motion Capture Database is hier een voorbeeld van. De hoeken die gegeven worden in de database kunnen door een Python script in Choregraphe worden uitgevoerd door de NAO. Choregraphe is het programma dat een connectie heeft met de robot en interactie mogelijk maakt. Ook is het binnen dit programma mogelijk om op een virtuele NAO het Python script te testen zodat er niet direct met een echte robot gewerkt hoeft te worden. Het aantal gewrichten van de robot die is gebruikt voor de Motion Capture Database, namelijk 30, bleek niet overeen te komen met het aantal gewrichten van de NAO, wat er 26 zijn. Ook was er bij deze database weinig documentatie en waren er amper relevante publicaties aangaande dit

probleem. In de documentatie en publicaties stond ook niet benoemd op welke volgorde de gewrichten in de database stonden, waardoor niet duidelijk was welke hoek bij welk gewricht van de NAO hoort. Het zou dus vrijwel onmogelijk zijn om achter de volgorde van de gewrichten te komen, ook zou de kans op een accurate beweging dan minimaal zijn vanwege het feit dat de NAO vier gewrichten minder heeft dan de robot van de database.

Bij een database met uitgebreide documentatie over het aantal gewrichten en informatie over welk gewricht bij welke hoek in de database hoort, zou er verder onderzoek gedaan kunnen worden. Relevanter is echter het analyseren van een danser, en de NAO zijn bewegingen over te laten nemen.

Het overnemen van bewegingen wordt gedaan met een Kinect, door gebruik te maken van Human Skeleton Tracking. Uit onderzoek van Schwarz et al [2] blijkt dat Kinect in staat is tot het meten en tekenen van een skelet. Met deze techniek kunnen de hoeken van verschillende gewrichten worden bepaald, waarna deze toegewezen kunnen worden aan een gewricht van de NAO. De Kinect is een product van Microsoft met een first-party SDK voor Windows, inclusief drivers. Hierbij is het belangrijk dat deze drivers ondersteund worden door de PC waar de Kinect op wordt aangesloten. Bij dit onderzoek werd gebruik gemaakt van een Xbox 360 Kinect, die niet werkt met versie 2.0 van de SDK. Versie 1.8 werkt wel voor dit model Kinect. Om met de Kinect human skeleton tracking te faciliteren wordt gebruik gemaakt van de Microsoft Kinect SDK. Deze SDK is een kit die het mogelijk maakt om applicaties te ontwikkelen die gebruik maken van 3D sensors. Binnen deze SDK is er voor dit onderzoek een aangepaste versie van het standaardprogramma "Basic Skeleton D2D" geschreven. Deze SDK is met name gekozen vanwege zijn precisie. Aan het begin van het onderzoek was het plan een combinatie van de programma's OpenNI en NITE te gebruiken, die ruwweg dezelfde functionaliteit zou hebben als de Microsoft Kinect SDK. Tijdens dit onderzoek bleek echter vrij snel dat

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \operatorname{atan2}(2(q_0 q_1 + q_2 q_3), 1 - 2(q_1^2 + q_2^2)) \\ \operatorname{asin}(2(q_0 q_2 - q_3 q_1)) \\ \operatorname{atan2}(2(q_0 q_3 + q_1 q_2), 1 - 2(q_2^2 + q_3^2)) \end{bmatrix}$$

Figuur 1: *Conversie Quaternionen naar Euler hoeken, resp. yaw, pitch en roll*

de combinatie OpenNI en NITE op gebied van "human skeleton tracking" niet accuraat genoeg is om dit onderzoek tot een goed eind te brengen.

Het programma "Basic Skeleton D2D", geschreven in C++, geeft de gebruiker de mogelijkheid aan de hand van de posities van ledematen, hoeken tussen deze ledematen uit te lezen. Binnen dit programma kunnen deze hoeken opgehaald worden in Quaternionen. Quaternionen zijn een vierdimensionale uitbreiding van de reële getallen. Quaternionen worden in veel computerprogramma's gebruikt voor relatieve plaatsbepaling, en zo ook binnen "Basic Skeleton D2D". Door middel van een standaardberekening zijn uit deze quaternionen de Hoeken van Euler te destilleren. De Hoeken van Euler zijn in dit geval de yaw(ϕ), pitch(θ) en roll(ψ) van een bepaald gewricht. Een ideaal gewricht kan in een 3D-wereld om drie assen draaien. Roll, pitch en yaw staan voor de hoeken waarover wordt gedraaid. Het algoritme is geschreven in C++, in het programma Visual Studio. De Euler hoeken die uit dit algoritme komen worden naar een tekstbestand geschreven, zodat de NAO ze later kan uitlezen. Zie Figuur 1 voor de conversie van Quaternionen naar Euler Hoeken.

De gewrichten van de NAO hebben een bepaalde volgorde, die door middel van een lijst in Python allemaal een hoek krijgen waarover ze moeten draaien. Dit zijn een roll, een pitch en een yaw, als dit natuurlijk voor het gewricht in kwestie van toepassing is. Het is belangrijk dat het C++ programma de hoeken dus in de juiste volgorde naar het tekstbestand schrijft, zodat de NAO de juiste gewrichten draait. Zie Figuur 2 voor een overzicht van de hiërarchie

```
volgordeNAO = ['HeadYaw', 'HeadPitch', 'LShoulderPitch',
               'LShoulderRoll', 'LElbowYaw', 'LElbowRoll',
               'LWristYaw', 'LHand', 'LHipYawPitch',
               'LHipRoll', 'LHipPitch', 'LKneePitch',
               'LAnklePitch', 'LAnkleRoll', 'RHipYawPitch',
               'RHipRoll', 'RHipPitch', 'RKneePitch',
               'RAnklePitch', 'RAnkleRoll', 'RShoulderPitch',
               'RShoulderRoll', 'RElbowYaw', 'RElbowRoll',
               'RWristYaw', 'RHand']
```

Figuur 2: *Hierarchie NAO gewrichten*

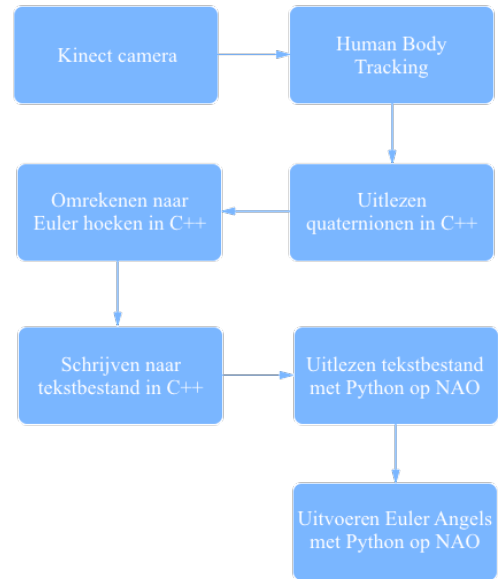
van gewrichten zoals deze gedefinieerd zijn voor de NAO robot. Voordat de hoeken in de NAO kunnen worden ingevoerd moet er een vertaalslag worden gemaakt. Na enig onderzoek bleek namelijk dat de Euler hoeken zoals ze in de NAO geprogrammeerd staan niet overeen komen met de definities van de hoeken van Euler zoals ze uit "Skeleton Basics D2D" verkregen worden. Het principe van de hoeken van Euler is de mogelijkheid tot universele as-bepaling. Hoewel de yaw, pitch en roll overeen horen te komen, geldt er dus een verschil in definitie in een aantal van de joints. Voor een overzicht van alle translaties zie Table V in de bijlage. Om te kijken of de hoeken die via de Kinect verkregen werden wel overeen zouden komen met de Nao, zijn de hoeken van de NAO in staande positie vergeleken met de hoeken die uit de Kinect-berekeningen kwamen op het moment dat deze staande positie voor de Kinect geïmmiteerd werd. Vervolgens is er voor de geobserveerde afwijkingen gecorrigeerd, dit betrof het negatief maken van de Shoulder joint hoeken die uit de Kinect-berekeningen kwamen. De data die via de Kinect verkregen was werd vervolgens via een python script, zie listing 1, in de Nao ingelezen die dit dan direct uitvoerde.

Listing 1: Invoeren data in NAO

```

open de file waar de hoeken in staan
zet de stiffness op 1 zodat de joints
kunnen worden aangestuurd
while(True):
    try:
        ga verder met lezen waar hij was
        gebleven
        begin een counter
        if counter > max lines
            stop script
        if not geen lines:
            continue
        if geen lines:
            sluit file en wacht tot weer
            lines zijn
            if te lang wacht:
                stop script
            try:
                kijk of er nieuwe lines zijn
                ga verder bij de laatst
                onthouden line
        else:
            splits de lines in coördinaat
            stukjes
            maak floats van de strings
            vraag de desbetreffende joint op
            veilige joint speed instellen
            actie uitvoeren

```

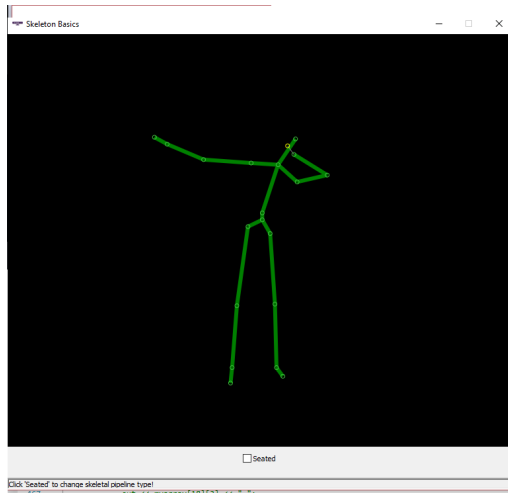


Figuur 3: *Overview werking motion-capture aansturing*

III. RESULTATEN EN DISCUSSIE

De eerste versie van het programma was een combinatie van OpenNI en NITE software. Met deze setup zijn er een aantal tests gedaan, waarbij elke keer de Hoeken van Euler (roll, pitch, yaw) op verschillende manieren doorgegeven werden aan de NAO om te kijken of het eerder beschreven probleem daarmee opgelost kon worden. Uiteindelijk is er van uit gegaan dat het aan de softwarecombinatie lag en is op zoek gegaan naar een beter alternatief.

Na het aanpassen van ons programma om het met de Microsoft Kinect SDK compatibel te maken is er eerst een aantal keer getest hoe accuraat deze SDK de hoeken kon berekenen, en of de uitkomsten na berekening van de Hoeken van Euler logisch waren. Dit is gedaan door in de ChoreGrappe software de beginwaarden van de NAO in rustpositie te observeren, en deze te vergelijken met data die uit de Kinect-berekeningen kregen wanneer deze positie voor de Kinect geïmmiteerd werd. Vervolgens zijn voor de Shoulder en Elbow joints stuk voor stuk de Hoeken van Euler afgegaan,



Figuur 4: *Skeleton Basics D2D*

om zo de goede translatie van Kinect naar Nao te verkrijgen. Zie Figuur 4 voor een indicatie van dit proces, een screenshot van "Skeleton Basics D2D".

Na de nodige berekeningen is er uiteindelijk een demonstratie gehouden met de software. Tijdens deze demonstratie zijn er door de pseudo-dansdocent simpele bewegingen uitgevoerd. De Nao is op dit moment geconfigureerd om iedere joint die door de sensor wordt gemeten uit te voeren. Dit gaf bij de eerste moment van testen problemen. Namelijk door het imiteren van de joints uit het onderlichaam van de docent, viel de Nao om. Door slechts de Nao de joints uit het bovenlichaam te laten overnemen, kon de demonstratie nu wel goed worden uitgevoerd. Nu bleek dat bewegen op het twee dimensionale vlak goed door de sensor werden geregistreerd, de Nao voerde dezelfde bewegingen uit die voor werden gedaan door de docent. Bewegingen in de diepte, bijvoorbeeld het naar voren steken van armen, werden niet goed geregistreerd. Bij deze bewegingen waren er geen overeenkomsten te zien tussen de docent en de Nao.

Hoewel de Microsoft Kinect SDK een stuk accurater is dan de combinatie van OpenNI en NITE, is er toch aardig wat ruimte voor verbetering. Allereerst hadden beide programma's veel moeite met bewegingen in "diepte", als er

bijvoorbeeld een arm naar de Kinect toe beweegt worden de joint rotation waarden die "Skeleton Basics D2D" teruggeeft steeds minder precies, dit is onder andere te zien aan het beeld wat dit programma schetst, maar ook uit te lezen aan de zogenaamde "confidence factor". Deze waarde berekent "Skeleton Basics D2D" berekent voor elke joint in elk frame, en geeft een indicatie van hoe hoog de zekerheid en precisie van een joint in die frame is. Een goede verbetering van het proces zou zijn om een Kinect 2.0 te gebruiken voor het onderzoek. Deze staat er om bekend beter met diepte te kunnen werken, en zou dus een betere schets van de werkelijkheid kunnen geven. Een andere verbetering zou het toevoegen van een tweede Kinect zijn. Als deze in een rechte hoek op de het beeldveld van de eerste Kinect wordt gezet, kan deze combinatie met veel grotere precisie bewegingen in de diepte berekenen. De Microsoft Kinect SDK ondersteunt van nature meerdere Kinect aansluitingen tegelijk, dus met enkele minimale aanpassingen zou het programma een stuk preciezere hoeken kunnen teruggeven, en dus preciezer bewegingen kunnen imiteren.

Een laatste verbetering die dit onderzoek zou kunnen ondergaan is het gebruik maken van een meer humanoïde robot. De NAO v4 is een redelijke representatie van het menselijk lichaam, maar een écht mens heeft meer gewrichten dan de NAO v4. Dit houdt in dat de NAO minder vrijheidsgraden heeft dan een mens, en er dus bewegingen zijn die een NAO gewoonweg niet kan maken. Een voorbeeld van een meer humanoïde robot is de nieuwere versie van de NAO. Deze NAO heeft meer gewrichten, en is dus een meer accurate spiegeling van het menselijk lichaam.

IV. CONCLUSIE

Zoals uit dit onderzoek blijkt is het zeker mogelijk om een humanoid zo te programmeren dat hij bewegingen van een dansdocent kan imiteren. Het opnemen van dansbewegingen wordt mogelijk gemaakt door middel van Human Skeleton Tracking. Bewegingen worden

door dan met berekeningen doorvertaald naar draaiingen die een humanoïde robot kunnen aansturen. Wel is het zo dat voor nu de precisie nog veel te wensen over laat. In toekomstig onderzoek zal het mogelijk zijn om aan de hand van de hierboven beschreven aanpassingen meer accurate en uitgebreide resultaten te behalen. Deze hogere precisie gaat hand in hand met de mogelijkheid tot opnemen en afspelen van meer complexe dansroutines, en op ten duur het overnemen van de taak van dansdocent.

REFERENTIES

- [1] Andri Ioannou, Emily Andreou, and Maria Christofi. Pre-schoolers' interest and caring behaviour around a humanoid robot. *TechTrends*, 59(2):23, 2015.
- [2] Loren Arthur Schwarz, Artashes Mkhitarian, Diana Mateus, and Nassir Navab. Human skeleton tracking from depth data using geodesic distances and optical flow. *Image and Vision Computing*, 30(3):217–226, 2012.

V. BIJLAGEN

Joint number	Kinect SDK	Nao	SDK joint to Nao joint	RPY
0	Hip center	Head	2 to 3	Yaw, Roll
1	Spine	Left shoulder	4 to 5	Pitch, Roll
2	Shoulder center	Left elbow	5 to 6	Yaw, Roll
3	Head	Left wrist	6 to 7	Yaw
4	Left shoulder	Left hand	/	/
5	Left elbow	Left hipyawpitch	0 to 12	Yaw
6	Left wrist	Left hip	12 to 13	Roll, Pitch
7	Left hand	Left knee	13 to 14	Pitch
8	Right shoulder	Left ankle	14 to 15	Pitch, Roll
9	Right elbow	Right hipyawpitch	0 to 16	Yaw
10	Right wrist	Right hip	16 to 17	Roll, Pitch
11	Right hand	Right knee	17 to 18	Pitch
12	Left hip	Right anke	18 to 19	Pitch, Roll
13	Left knee	Right shoulder	8 to 9	Pitch, Roll
14	Left ankle	Right elbow	9 to 10	Yaw, Roll
15	Left foot	Right wrist	10 to 11	Yaw
16	Right hip	Right hand	/	/
17	Right knee			
18	Right ankle			
19	Right foot			

Bijlage 1: Translaties van Kinect naar NAO