



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

# **Gestų atpažinimo sistemos InMoov robotui sukūrimas ir tyrimas**

Baigiamasis magistro projektas

---

**Aivaras Pilelis**

Projekto autorius

**Doc. Dr. Gintaras Dervinis**

Vadovas

---

**Kaunas, 2022**



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

# **Gestų atpažinimo sistemos InMoov robotui sukūrimas ir tyrimas**

Baigiamasis magistro projektas

Valdymo technologijos (6211EX014)

---

**Aivaras Pilelis**

Projekto autorius

**Doc. Dr. Gintaras Dervinis**

Vadovas

**Prof. Dr. Vidas Raudonis**

Recenzentas

---

**Kaunas, 2022**



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

Aivaras Pilelis

## **Gestų atpažinimo sistemos InMoov robotui sukūrimas ir tyrimas**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs;
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Aivaras Pilelis

*Patvirtinta elektroniniu būdu*

Aivaras Pilelis. Gestų atpažinimo sistemos *InMoov* robotui sukūrimas ir tyrimas. Magistro baigiamasis projektas. vadovas Doc. Dr. Gintaras Dervinis; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas.

Studijų kryptis ir sritis (studijų kryptių grupė): .Elektronikos inžinerija (inžinerijos mokslai).

Reikšminiai žodžiai: Dirbtinis intelektas, CNN, Neuroniniai tinklai, Kinect, BlazePose, InMoov.

Kaunas, 2022. 50 p.

### **Santrauka**

Šiame darbe kuriamas algoritmas *InMoov* robotui žmogaus gestams atpažinti ir atkartoti naudojantis naujausiais dirbtinio intelekto modeliais žmogaus pozos sekimui 3D koordinatinių sistemoje. Darbe supažindinama su *InMoov* roboto įrangos baze ir galimybėmis. Pateikiama naujausių darbų apžvalga žmogaus roboto sąveikos srityje. ir išanalizuojami trys metodai žmogaus sąnarių 3D koordinatėms gauti: *BlazePose*, *MobileHumanPose*, *ResidualPose*. Taip pat darbe siūlomas kombinuotas metodas susidedantis iš neuroninio tinklo modelio 2D koordinatėms nustatyti bei analitinio algoritmo gautų 2d taškų perkėlimui į 3D erdvę. Darbe taikomas tyrimo metodas – literatūros analizė, apibendrinimas ir algoritmų bandymai bei rezultatų pateikimas.

Aivaras Pilelis. Gestų atpažinimo sistemos *InMoov* robotui sukūrimas ir tyrimas. Master's Final Degree Project. supervisor doc. dr. Gintaras Dervinis; Faculty Of Electrical And Electronics Engineering, Kaunas University of Technology.

Study field and area (study field group): Electronics engineering (Engineering science).

Keywords: Artificial intelligence, Neural network. CNN, Kinect, BlazePose, InMoov

Kaunas, 2022. 50 p.

### **Summary**

This project development of an algorithm for the *InMoov* robot to recognize and reproduce gestures using the latest artificial intelligence model to track human posture in a 3D coordinate system is presented. The work introduces the InMoov robot equipment base and capabilities. An overview of recent work in the field of human robot interaction is provided. Three methods for obtaining 3d coordinates of human joints are analyzed: *BlazePose*, *MobileHumanPose*, *ResidualPose*. In paper Author proposed combined method consisting of a neural network model for determining 2D coordinates and an analytical algorithm for transferring the obtained 2D points to 3D coordinate system. The research method used in the work is literature analysis, generalization and testing of algorithms, presentation of results.

## Turinys

<b>Lentelių sąrašas.....</b>	<b>7</b>
<b>Paveikslų sąrašas .....</b>	<b>8</b>
<b>Santrumpų ir terminų sąrašas.....</b>	<b>10</b>
<b>Įvadas.....</b>	<b>11</b>
<b>1. Probleminė dalis .....</b>	<b>13</b>
<b>2. Naudojama aparatūra ir programinė įranga.....</b>	<b>16</b>
2.1. <i>InMoov</i> robotas.....	16
2.2. Naudojama programinė įranga .....	19
2.2.1. <i>MyRobotLab</i> – robotų valdymo aplinka .....	19
2.3. <i>Jupyter Notebook</i> – programavimo ir skaičiavimų platforma .....	20
2.4. <i>Kita programinė įranga</i> .....	20
<b>3. Tyrimo dalis .....</b>	<b>22</b>
3.1. 3D žmogaus pozos koordinacių gavimas. ....	22
3.1.1. Neuroninis modelis <i>MobileHumanPose</i> .....	22
3.1.2. <i>MediaPipe</i> neuroninis modelis <i>BlazePose</i> . ....	23
3.1.3. <i>Residual Pose</i> neuroninio tinklo modelis. ....	24
3.1.4. Siūlomas metodas .....	25
3.2. Kampų skaičiavimas.....	29
<b>4. Eksperimentinė dalis .....</b>	<b>32</b>
4.1. <i>IR</i> kadro perskaičiavimas. ....	32
4.2. Kinect kameros kalibravimo rezultatai.....	32
4.3. 2D Žmogaus kūno taškų aptikimo bandymai.....	33
4.4. Kampų nustatymo – pozos atitikties bandymas .....	37
4.5. Kampų judesio bandymai .....	42
<b>Išvados .....</b>	<b>47</b>
<b>Literatūros sąrašas .....</b>	<b>48</b>

## Lentelių sąrašas

2.1 lentelė. <i>InMoov</i> roboto naudojamos standartinės servo pavaros [25] .....	17
4.1 lentelė. Modelių 2D koordinatinių nustatymo bandymo rezultatai.....	36
4.2 lentelė. Kampų atitikties bandymų rezultatai .....	37

## Paveikslų sąrašas

1.1 pav. <i>TIANgo</i> roboto ir žmogaus alkūnės padėties grafikai [19] .....	13
1.2 pav. mini roboto valdymas naudojant <i>OpenNI</i> biblioteką ir <i>kinect v1</i> [2] .....	14
1.3 pav. žmogaus užduotos pozicijos roboto rankos atkartota pozicija [3] .....	14
1.4 pav. Baxter robotas kartuoja žmogaus judesius .....	14
1.5 pav. NAO roboto valdymas naudojant judesio atpažinimą <i>HMM</i> modeliu [5] .....	15
2.1 pav. <i>InMoov</i> robotas .....	16
2.2 pav. <i>InMoov</i> roboto valdymo sistema .....	17
2.3 pav. <i>Kinect v2 RGB-D</i> kamera .....	19
2.4 pav. <i>InMoov</i> roboto simulatorius .....	19
2.5 pav. <i>Jupyter Notebook</i> - programavimo pavyzdys .....	20
2.6 pav. <i>Matlab - Camera Calibrator</i> .....	21
3.1 pav. a) <i>MobileHumanPose</i> naudojamas modifikuotas invertuotas likutinis blokas [36], b) originalus invertuotas likutinis blokas [35] .....	23
3.2 pav. <i>MobileHumanPose</i> modelio bendra struktūra [36] .....	23
3.3 pav. <i>BlazePose</i> tinklo architektūra [37] .....	23
pav. 3.4 <i>BlazePose</i> veikimo algoritmas [36] .....	24
3.5 pav. <i>Residual pose</i> algoritmo seka. ....	24
3.6 pav. a) gylis kadrus, b) infraraudonos spalvos kadrus .....	26
3.7 pav. Pikselio užpildymo algoritmas .....	26
3.8 pav. 3D koordinačių koregavimo algoritmo diagrama .....	27
3.9 pav. <i>InMoov</i> roboto rankos judėjimo galimybės .....	29
3.10 pav. <i>Kinectv2</i> koordinačių sistema .....	29
4.1 pav. IR kadro histograma .....	32
4.2 pav. Neskaliuotas ir perskaliuotas <i>IR</i> kadrus .....	32
4.3 pav. Kalibravimo: kairėje <i>IR</i> kadrus, dešinėje <i>RGB</i> kadrus .....	33
4.4 pav. Teisingu spėjimu laikomas paveikslėlis .....	33
4.5 pav. <i>MobileHumanPose</i> modelio bandymų rezultatai: <i>RGB</i> kairėje, <i>IR</i> dešinėje .....	34
4.6 pav. <i>ResidualPose</i> 2D pozos aptikimo rezultatai .....	34
4.7 pav. <i>BlazePose</i> pozos nustatymas naudojant <i>RGB</i> kadrus. Pirma eilė – lengvas modelis, antra – sunkus, trečia – pilnas .....	35
4.8 pav. <i>BlazePose</i> pozos nustatymas naudojant <i>IR</i> kadrus. Pirma eilė – lengvas modelis, antra – sunkus, trečia – pilnas .....	36
4.9 pav. <i>BlazePose</i> išduotos 3D koordinatės .....	38
4.10 pav. <i>ResidualPose</i> 3D koordinatės .....	39
4.11 pav. <i>BlazePose</i> su koordinačių korekcijos algoritmu 3D koordinatės .....	40
4.12 pav. <i>InMoov</i> roboto ir žmogaus pozicija .....	41
4.13 pav. Taškų perkėlimo į 3D koordinačių sistema problema .....	41
4.14 pav. Rankos lenkimo per alkūnę grafikas .....	42
4.15 pav. alkūnės z ašies vertė = 0 .....	43
4.16 pav. rankos sulenkimo per alkūnę grafikas iš sąnarių pozicijos po užpildymo algoritmo imties padidėjimo .....	43
4.17 pav. Rankos sukimo nuo kūno į išorę grafikas .....	44
4.18 pav. Rankos kėlimo į šoną grafikas .....	44
4.19 pav. <i>InMoov</i> robotas $K_4$ kampas 90 ir 40 (min vertė) .....	45



4.20 pav. Kampo $K_I$ keitimo kitimo grafikas .....	46
4.21 pav. Judesio kadrai nuo 83 iki 89 kadro .....	46

## Santrumpų ir terminų sąrašas

*CNN* – Konvoliucinis neuroninis tinklas (angl. Convolutional neural network).

*HMM* – Paslėptas Markovo modelis (angl. Hidden Markov model).

*HTML* – kompiuterinė žymėjimo kalba turinio internete pateikimui (angl. *HyperText Markup Language*).

*IR* – Infraraudona šviesa

*RGB* – nuotraukų spalvų formatas susidedantis iš raudonos, žalios ir mėlinos spalvos.

*LaTeX* – Tai programinė dokumentų rengimo sistema.

*RGB-D* – Spalvotos nuotraukos ir gylio nuotraukos rinkinys, arba kameros tipas galintis pateikti tokį rinkinį.

*PNG* – Bitų masyvo formatas glaudinamas be kokybės praradimo (angl. Portable network graphics).

*SVG* – Vektorinis paveikslėlio saugojimo formatas. (angl. Scalable vector Graphics )

*WBC* – *Viso kūno valdymas algoritmas* skirtas apsaugoti robotą ir kaip galima tiksliau pakartoti sugeneruotą užduotį (angl. *whole body control*).

## Įvadas

Terminas robotas pirmą kartą buvo panaudotas Čekų novelių rašytojo Karel Čapek kūrinyje *Rossum's Universals robots* išleistame 1921 metais. Ten robotai yra vaizduojami kaip žmonių vergai, kurie vėliau sukyla prieš savo kūrėjus [1]. Pats žodis yra kilęs iš čekų žodžio „robota“, kuris reiškia baudžiauninkas arba darbininkas [2]. 1942 metais rusų kilmės amerikiečių mokslinės fantastikos rašytojas Aizekas Asimovas (angl. Isaac Asimov) sukūrė terminą *robotika* ir suformulavo pagrindines robotų taisykles:

1. robotas negali sužeisti žmogaus ir negali būti neaktyvus, kai žmogus gali susižeisti pats
2. robotas privalo paklusti žmogaus įsakymams, išskyrus tuos atvejus, kai tai prieštarauja pirmajai taisyklei
3. robotas privalo išsaugoti savo egzistenciją, nebent tai prieštarautų pirmajai arba antrajai taisyklei [3].

Iki 1950-tųjų robotai vis dar buvo tik mokslinė fantastika. Roboto kaip realaus inžinerinio kūrinio idėją 1954-tais metais užpatentavo Džordžas Čarlzas Devolas (angl. George Charles Devol) pavadindamas aparatą programuojamu manipulatoriumi. Pirmąjį prototipą jis sukūrė kartu su inžinieriumi Džozefu Engelbergu (angl. Joseph Engelberger) 1959-ais metais pavadinimu *Unimate* [4]. Dar po dviejų metų 1961 metais *Unimate* robotai buvo pradėti naudoti *General Motors* gamykloje paimti dalims iš liejimo įrenginių. Po didelio pasisekimo su pirmaisiais industriniais robotais, daug kitų įmonių irgi pradėjo kurti naujus industrinius robotus [5].

Nors industriniai robotai atsirado jau šeštajame dešimtmetyje, pirmieji modernūs humanoidiniai robotai atsirado tik aštunto dešimtmečio pradžioje. Tokį robotą, 1973-ais metais, sukūrė Japonijos Wasedos universiteto mokslo ir inžinerijos departamento bioinžinerijos tyrimų grupė. Robotas buvo pavadintas *WOBAT-1*. Šis robotas galėjo vaikščioti, išmatuoti atstumus ir kryptis iki kitų objektų ir komunikuoti japonų kalba. Šis robotas turėjo ausis, akis ir dirbtinę burną. 1984 metais jie sukūrė robotą *WABOT-2*, kuris gebėjo groti vargonais ir 1995 metais mokslo parodoje grojo Japonijos vyriausybės paviljone [6].

1996 metais gruodį *Honda* kompanija paskelbė kurianti humanoidinį robotą pavadinimu *P2*. Šio roboto kūrimas buvo pradėtas dar 1986 metais. *Honda* tikslas buvo sukurti robotą kuris galėtų egzistuoti ir dirbti kartu su žmonėmis, bei atlikti užduotis, kurių žmonės atlikti negali. Šis robotas galėjo pereiti pro siaurus tarpus, lipti laiptais, perlipti kliūtis, eiti nuolydžiu, kurio dydis nedidesnis nei 10%, paimti ir laikyti objektus iki 5 kg masės ir atlikti lengvus darbus naudojantis nuotoliniu valdymu [7]. 2011 metais *Honda* ištobulino kitą robotą pavadinimu *ASIMO*, kuris laikomas vienu pažangiausių humanoidiniu robotu, dėl savo gebėjimo greitai bėgti, lipti laiptais, bėgti atbulomis, šokinėti ant vienos ar dviejų kojų, vaikščioti nelygiais paviršiais. Taip pat *ASIMO* robotas turi dirbtinį intelektą leidžianti jam priimti sprendimus judesių perplanavimui, suprasti natūralią kalbą, gestais remtą kalbą bei kitus gebėjimus [8]. 2018 metais *Honda* nutraukė *ASIMO* roboto tobulinimą [9].

Šie robotų prototipai buvo orientuoti į užduočių atlikimą, tačiau įvairių šalių mokslininkai kuria ir socialinius robotus. Vienas tokių robotų yra 2014 metais pristatytas *SoftBank* kompanijos robotas *pepper*. Šis robotas naudoja kūno kalbą, suvokia ir geba sąveikauti su aplinka bei judėti aplinkoje. Jis gali analizuoti žmogaus mimikas ir balso toną bei atpažinti emocijas naudodamas naujausius

algoritmus skirtus balso ir emocijų atpažinimui. Tai jam leidžia sąveikauti su žmonėmis. Priešingai nei kiti aptarti robotai *pepper* yra kuriamas masiškai [10]. Robotas *pepper* buvo naudojamas senelių namuose treniruotėms [11], vaikų mokymui [12], šokolado kioske [13], metro stotyse [14] ir daugelyje kitų vietų. Taip pat 2014 metais atsirado robotas, pavadinimu *Eugene Goostman*, kuris įveikė Alano Turingo testą privertęs ekspertus manyti, jog jis yra žmogus [15]. Po dviejų metų pasirodė robotas pavadinimu *Sofija* (angl. *Sophia*). Tai pirmas robotas pasaulyje, kuriam buvo suteikta valstybės pilietybė. Šis robotas, naudodamas naujausius dirbtinio intelekto pasiekimus, geba bendrauti su žmonėmis, bei gali mokytis iš pokalbių ir kaupti žinias, priimti protingus sprendimus. Šis robotas dalyvavo daugelyje interviu, kuriuose net juokavo [16]. Nors *Sofijos* intelektinės savybės yra pribloškiančios, tačiau jos išvaizda vis dar nėra žmogiška. Šiuo metu žmogiškiausiai išvaizdą turintis robotas yra *Ameca*. Šį robotą 2021 metų gruodį pristatė Jungtinės Karalystės bendrovė *Engineering Arts* [17]. *Ameca* laikoma pasaulio pačiu pažangiausiu humanoidiniu robotu. Šis robotas sukurtas specifiskai kaip platforma ateities humanoidinių robotų technologijų kūrimui. Šio roboto realistiška išvaizda buvo sukurta naudojant tos pačios kompanijos *Mesmer* technologiją. Kiekvienas roboto dizainas kuriamas pagal realių žmonių 3D skenavimus, taip sukuriant tikslią žmogaus kaulų ir odos struktūrą bei mimikas [18]. Nuo pirmojo roboto sukūrimo praėjo 63 metai ir buvo padaryta labai didelė pažanga.

Šiais laikais, senėjančioje visuomenėje, trūksta vis daugiau darbo jėgos, todėl robotų technologijos tampa vis aktualesnės šiuolaikiniame pasaulyje. Šių technologijų vystymas turi būti prioritetinga mokslininkų ir inžinierių veiklos sritis.

Šio baigiamojo projekto pagrindinis tikslas yra sukurti ir iširti algoritmą, leisiantį *InMoov* robotui atpažinti priešais jį stovinčio žmogaus rankos gestus ir juos atkartoti.

Darbo uždaviniai:

1. atlikti literatūros šaltinių analizę tiriamą tematiką;
2. išbandyti egzistuojančius algoritmus žmogaus pozos skeleto 3D koordinatėms nustatyti ir pritaikyti *InMoov* roboto gestų atpažinimui;
3. sukurti algoritmą žmogaus skeleto 3D koordinatėms perskaičiuoti į kampus *InMoov* robotui;
4. atlikti perskaičiuojamų kampų judesio kitimo bandymus ir įvertinti jų kitimo kokybę.

## 1. Probleminė dalis

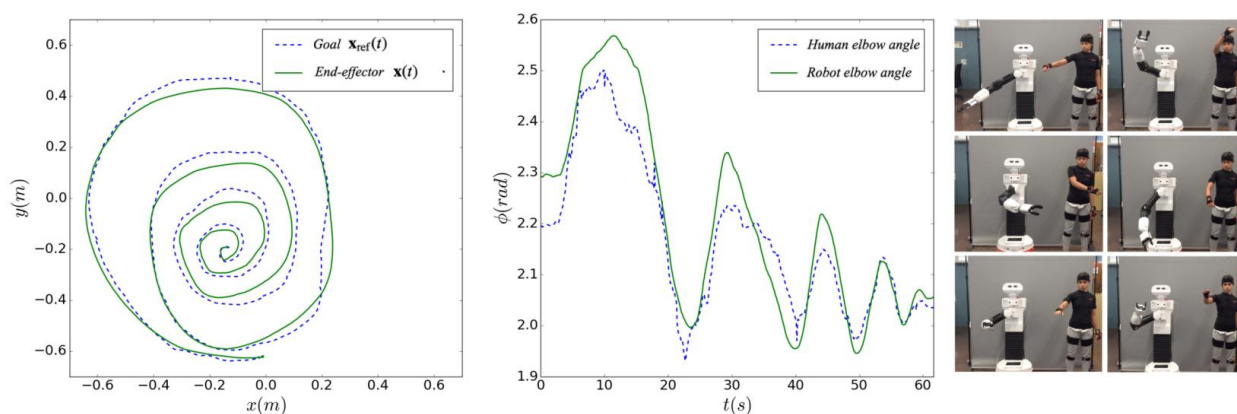
Siekiant supaprastinti robotų valdymą pastaruoju metu mokslininkai ieško vis efektyvesnių būdų kaip robotui perteikti informacija, žmogui būdingais būdais.

Šaltinyje [19] pateikiamas naujas būdas, skirtas mobiliam robotui perduoti visą žmogaus kūno judėjimo informaciją naudojant *Xsens MVN* judesio fiksavimo kostiumą (šiuo kostiumu matuojami 17 inercinių matavimo vienetų, kurie leidžia realiu laiku nuspėti žmogaus pozą, duomenys perduodami UDP/IP protokolu) tam, jog robotas imituotų žmogaus judesius realiu laiku. Tyrimo tikslas buvo perduoti duomenis į robotą išsprendžiant korespondencijos problemą: t. y. įveikiant tokias problemas kaip skirtumų atsiradimas dėl objektų judėjimo, laiko skirtumo arba kameros judėjimo.

Korespondencijos problema sprendžiama ją padalinus į tris dalis:

1. stebėjimą: išmatuojama žmogaus būseną;
2. lygiavertiškumą: sukuriamas ryšys tarp stebimos žmogaus būsenos ir norimos roboto pozos;
3. imitaciją: nusakoma roboto konfigūracija tikslui pasiekti.

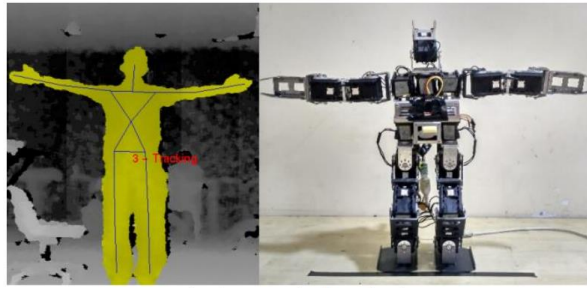
Korespondenciją sujungus kartu su *WBC* (angl. *whole body control*, skirtas apsaugoti robotą ir kaip galima tiksliau pakartoti sugeneruotą užduotį) valdymu buvo pasiekti neblogi realaus laiko žmogaus judesių atkartojimo rezultatai.



1.1 pav. *TIANgo* roboto ir žmogaus alkūnės padėties grafikai [19]

Robotas atkartojė judesius realiu laiku. Robotų greitis atkartojant žmogų buvo apribotas tik jo paties pavarų greičio.

[20] šaltinio autoriai robotų valdymui naudodami *Kinect v1* kamerą ir mažą humanoidinį robotą pristatė humanoidinio robotų valdymą, kuris naudodamas *OpenNI* bibliotekos skeleto sekimo algoritimą ir remdamasis kampų tarp kūno segmentų dydžiais sukūrė paprastą robotų valdymo metodą, kuris suteikė galimybę robotui atkartoti žmogaus judesius, taip pat ir suktis kartu su žmogumi. Taip pat robotas gali stabiliai vaikščioti. Atsiradus kūno judesiams, ėjimo funkcija buvo išjungiamą siekiant išlaikyti robotų stabilumą. Atliekant bandymus buvo patiriamas nedidelis atsilikimas.



1.2 pav. mini roboto valdymas naudojant *OpenNI* biblioteką ir *kinect v1* [2]

Panaši metodologija buvo naudojama ir [21] šaltinyje, tačiau vietoje *OpenNI* bibliotekos buvo naudojama *Kinect SDK* skeleto sekimo funkcija. Šiuo atveju buvo valdoma roboto ranka. Nebuvo pasiekti labai geri rezultatai, greičiausiai dėl to, jog *Kinect SDK* funkcija nėra labai stabili ir esant greitesniems judesiams nepajėgi tinkamai nustatyti žmogaus kūno taškus. Pasak pačių kūrėjų, tik keturios iš dešimties bandytų pozų buvo atkartojamos teisingai.



1.3 pav. žmogaus užduotos pozicijos roboto rankos atkartota pozicija [3]

Roboto pavaroms valdyti buvo naudojamas „*arduino*“ kontroleris ir *Windows presentation foundation* programinė įranga.

Šaltinyje [22] pateikiamas roboto judesio atpažinimo ir pakartojimo būdas, kai naudojama tik *RGB* kamera ir *CNN* neuroninis tinklas, kuris nuspėja šiluminių žemėlapių pavidalu žmogaus pozą, kurią vėliau koreguoja naudojant papildomą apdorojimą „Lokalaus vaizdo interpoliacijos algoritmą“. Šis algoritmas, *CNN* sukurtus šiluminius žemėlapius (angl. *Heat maps*), kiekvienam sąnariui atskirai interpoliuoja aplink maksimalaus pasitikėjimo pikselius pradiname šiluminiame žemėlapyje ir tada vėl nustatomas sąnario buvimo taškas. Taip atliekama taškų paieška yra 2,7 karto greitesnė negu tai darant naudojant visą nuotrauką, o tikslumas išliko toks pat. Algoritmas buvo išbandytas su *Baxter* ir *DRORUS* robotais. Gauti rezultatai rodo, jog tai labai patikimas būdas valdyti skirtingus robotus. Vienintelis apribojimas yra pačių robotų konstrukcijos.



1.4 pav. *Baxter* robotas kartoja žmogaus judesius

Straipsnyje [23] žmogaus gestams atpažinti buvo panaudotas *Kinect SDK*, o roboto valdymas buvo atliekamas atpažįstant judesius naudojant *HMM* (angl. *hidden markow model*) modelius. Nors šaltinyje pateiktas metodas, lyginant su kitais panašiu principu veikiančiais metodais, veikė geriau, tačiau, robotas yra apribotas atlikti tik tuos judesius, kuriuos buvo išmokyta atlikti modelis.



**1.5 pav.** NAO roboto valdymas naudojant judesio atpažinimą *HMM* modeliu [5]

Šaltinyje [24] yra pateikiamas vienas naujausių roboto valdymo metodų naudojant smegenų-kompiuterio sąsają (angl. *brain-computer interface*). Robotas šiuo atveju buvo valdomas žmogui įsivaizduojant atliekamą užduotį. Žmogaus smegenų signalai naudojant *MI* žmogaus-kompiuterio sąsajos technologiją taip paverčiant signalus į skaitmeninėmis komandomis robotui. Tyrimo metu buvo nustatyta, jog tokiu būdu patikimai galima valdyti *UGV* (angl. *Unmanned Ground Vehicles*) roboto judėjimą. Valdant robotą tokiu būdu nebereikia naudoti žmogaus kūno judesių, ir robotą galima lengviau valdyti pavojingose vietose.

## 2. Naudojama aparatūra ir programinė įranga

### 2.1. *InMoov* robotas

*InMoov* robotas yra atviras roboto modelis, kurį gali susikonstruoti visi norintys. Roboto kūrėjas yra Gael Langevin prancūzų skulptorius ir dizaineris. Roboto projektas pradėtas dar 2012 metais. Tai yra pirmas žmogaus ūgio robotas, kurį galima atsispausdinti. Šis robotas buvo pradėtas kaip prostetinė ranka, bet vėliau išaugo iki pilno humanoidinio roboto projekto. Šis robotas naudojamas universitetuose, laboratorijose ir individualiuose projektuose [25].

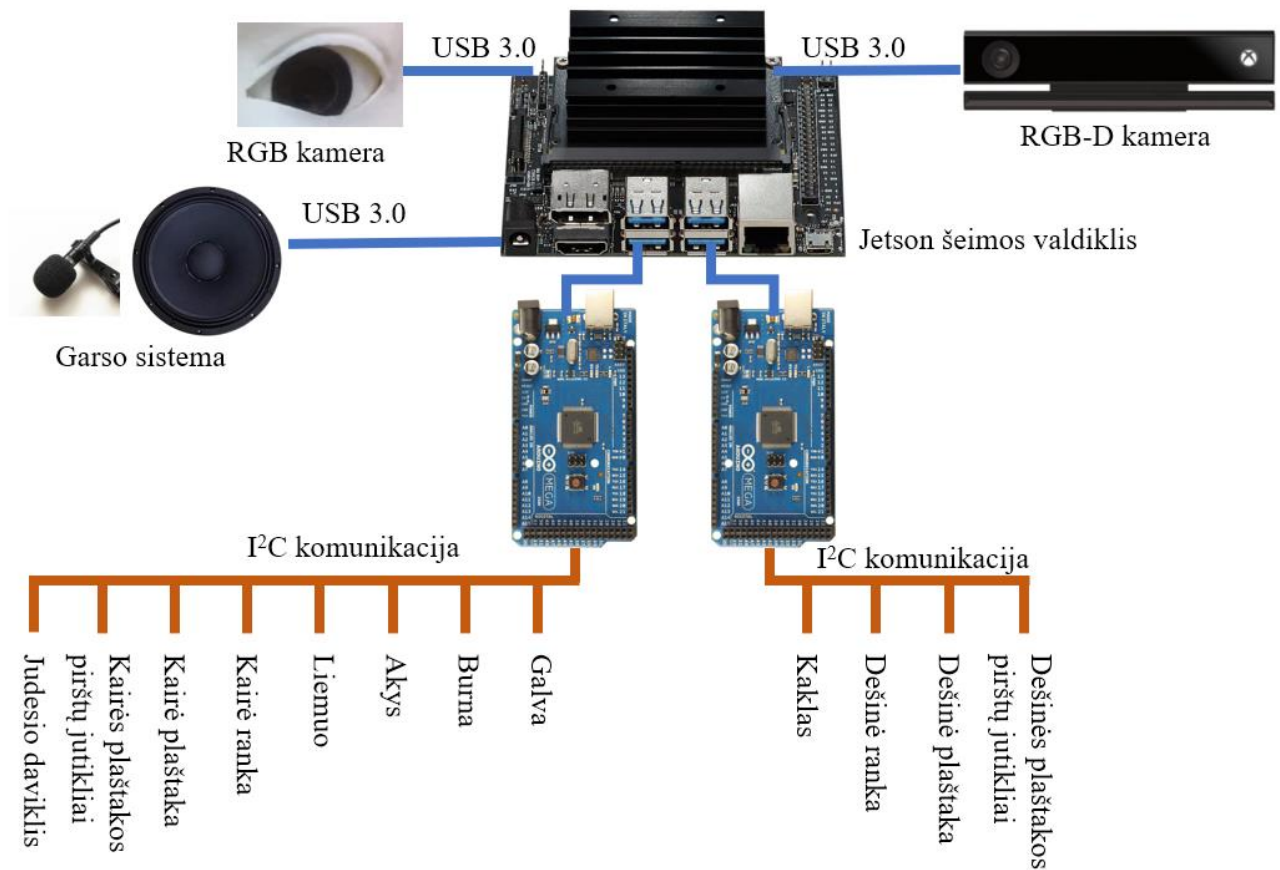


2.1 pav. *InMoov* robotas

*InMoov* roboto konstrukcija susideda iš: rankų, plaštakų galvos ir liemens. Roboto konstrukcija leidžia valdyti roboto galvos, akių, burnos, alkūnės, pečių, plaštakos bei pirštų judesius. Visiems judesiams atlikti yra naudojamos servo pavaros [25].

*InMoov* roboto servo pavarų valdymo kontrolierių bendra schema pateikiama 2.2 pav. žemiau:





2.2 pav. InMoov roboto valdymo sistema

Servo pavaros yra valdomos naudojant du *Arduino ATMEGA 2560* kontrolerius ir  $I^2C$  komunikaciją. Šio projekto atveju pagrindinis InMoov procesorius yra *NVIDIA* kompanijos *Jetson* šeimos kompiuteris. Kiekvienas *Arduino* kontroleris valdo po vieną ranką. Vienam kontroleriui yra priskirtos papildomos pavaros skirtos liemeniui, akims, burnai ir galvos judinimui, o kaklo pavaras valdo antras kontroleris [25].

Servo pavarų judėjimo ribos pateikiamos lentelėje žemiau:

2.1 lentelė. InMoov roboto naudojamos standartinės servo pavaros [25]

Kūno dalis	Elementai	Servo pavaras	Minimali pozicija	Maksimali pozicija
Plauštaka	Nykštys	HK15298B	0	180
	Smilius	HK15298B	0	180
	Didysis	HK15298B	0	180
	Bevardis	HK15298B	0	180
	Mažylis	HK15298B	0	180
	Riešas	MG996r	0	180
Ranka	Alkūnė	HS805BB	0	90
	Petys (į priekį)	HS805BB	0	180
	Petys (į šoną)	HS805BB	10	80
	Petys (sukimas)	HS805BB	40	180
Galva	Kaklas sukimas	HS805BB	20	160

	Galvos sukimas	HS805BB	30	150
	Kaklo lenkimas	HK15298B	60	130
	Akių vokai	DS929hvCorona	60	120
Akis X		DS929hvCorona	60	120
Akis Y		DS929hvCorona	60	120
Liemuo	Viršutinė dalis	HS805BB	60	120
	Vidurinė dalis	HS805BB	60	120
	Apatinė dalis	-	Gali kisti	Gali kisti
Burna	Burna	HK15298B	10	25

*InMoov* robotas turi dvi kameras, vieną *RGB* ir vieną *RGB-D* kamerą.

*RGB* kamera yra *Microsoft LifeCam HD-3000* galinti filmuoti 30 kadrų per sekundę 720p HD kokybe, bei galinti pateikti 16:9 formato vaizdą [26].

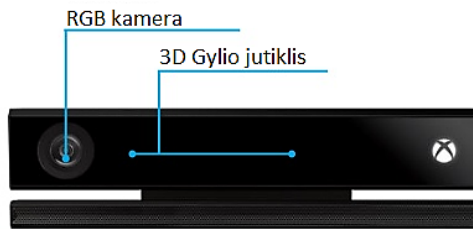
*RGB-D* buvo parinkta *Kinect v2* kamera. Ši kamera paremta *TOF* (angl. time-of-flight) technologija. Tokia kamera veikia apšviesti aplinką su moduliuotos šviesos šaltiniu ir sekdamas atsispindėjusią šviesą. Sukeltas fazės pokytis tarp šaltinio ir atspindžio šviesos perskaiciuojamas į atstumą [27].

*Kinect v2* techniniai duomenys:

- *RGB* kameros rezoliucija: 1920 x 1080;
- *RGB* kameros greitis: 30 kadrų per sekundę;
- gylis kameros rezoliucija: 512 x 424;
- gylis kameros greitis: 30 kadrų per sekundę;
- atstumo (arba gylis) matavimas: 0.5 m - 4.5 m;
- duomenų perdavimo sąsaja: USB 3.0 [28].

*Kinect v2* sistemos reikalavimai:

- palaikoma operacinė sistema *Windows 8*, arba vėlesnės *Windows* operacinės sistemos iki *Windows 10*;
- 64-bitų dviejų branduolių 3.2 GHz arba galingesnis procesorius;
- būtina USB 3.0 magistralė;
- 2 GB RAM operatyvioji atmintis [29].



2.3 pav. Kinect v2 RGB-D kamera

Nors *Windows* kūrė šią kamerą veikti *Windows* platformoje, tačiau egzistuoja atviro kodo bibliotekos kaip *pylibfreenect*, kuri leidžia naudoti *Kinect v2 Linux* operacinėje sistemoje.

Visiems bandymams atlikti naudojamas stacionarus kompiuteris su *Intel i5-3470* procesoriumi ir *NVIDIA GEFORCE GTX1650*.

## 2.2. Naudojama programinė įranga

### 2.2.1. *MyRobotLab* – robotų valdymo aplinka

*MyRobotLab* yra atviro kodo *Java* servisas remta aplinka (angl. framework) skirta robotikai ir mašinų valdymui. *MyRobotLab* veikia *Java 1.8* arba aukštesnėje *JVM* aplinkoje. Šioje aplinkoje yra *Javacv/OpenCv*, *Sphinx4 FreeTTS* bibliotekų elementų, motorų, servo pavarų valdymo modulių ir GUI valdymas, mikrokontrolerių komunikacijos, bei daugelis kitų trečiųjų šalių servisų, kurie yra įskiepyti *MyRobotLab* aplinkoje. Priklausomai nuo roboto įrangos, *MyRobotLab* gali būti įdiegtas pačiame roboto kompiuteryje, o jeigu jo nėra, tada programinę įrangą galima įdiegti į išorinį kompiuterį, kuris valdo robotą per nuotolį naudojant kokią nors komunikaciją (pvz. *Bluetooth*).

Jeigu yra naudojami keli kompiuteriai, kurie naudojami *MyRobotLab*, tokiu atveju galima juos susieti ir kompiuteriai gali matyti, kokius *MyRobotLab* servisuos kuris kompiuteris vykdo.

Neturint realaus roboto, galima naudotis *MyRobotLab InMoov* roboto simulatoriumi (2.4 pav.), kuris leidžia susikonfigūruoti robotą pagal norimą konfigūraciją (arba konfigūraciją, kurią turės realus robotas) ir išbandyti esančias roboto valdymo programas bei išbandyti savo sukurtas programas [30].



2.4 pav. *InMoov* roboto simulatorius.

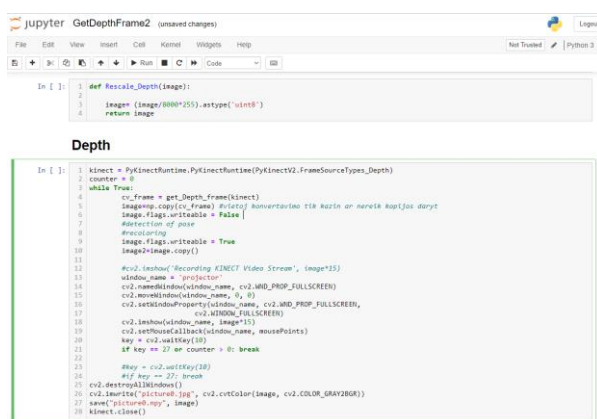
### 2.3. Jupyter Notebook – programavimo ir skaičiavimų platforma

*Jupyter notebook* suteikia naršyklės pagrindu sukurtą darbo aplinką, tinkančią visam projekto procesui fiksuoti: kodo kūrimui, dokumentavimui ir vykdymui, taip pat rezultatų perdavimui. *Jupyter notebook* sujungia du komponentus:

- naršyklės programa: naršyklės įrankis, skirtas interaktyviai kurti dokumentus, kuriuose derinamas aiškinamasis tekstas, matematika, skaičiavimai ir jų raiškiosios medijos išvestis;
- užrašinės dokumentai: viso žiniatinklio programoje matomo turinio, įskaitant skaičiavimų įvestis ir išvestis, aiškinamąjį tekstą, matematiką, bei vaizdavimo priemones.

Pagrindinės *Jupyter notebook* savybės:

- kodo redagavimas naršyklėje su automatišku sintaksės paryškinimu, įtrauka ir skirtukų užbaigimu / savistaba;
- galimybė vykdyti kodą iš naršyklės su skaičiavimų rezultatais, pridedamais prie kodo, kuris juos sugeneravo;
- skaičiavimo rezultato rodymas naudojant paplitusias struktūras, pvz., *HTML*, *LaTeX*, *PNG*, *SVG* ir kt. Pavyzdžiui, grafikai, sugeneruoti *matplotlib* bibliotekos, gali būti įtraukti į eilutę;
- raiškiojo teksto redagavimas naršyklėje naudojant *Markdown* žymėjimo kalbą, kuri leidžia komentuoti kodą ne vien paprastu tekstu;
- galimybė lengvai įtraukti matematinis žymėjimus į *Markdown* langelius naudojant *LaTeX* ir *MathJax* [31].



```
1 def Rescale_Depth(image):
2
3     image = (image/255).astype('uint8')
4     return image

Depth

In [ ]: 1 Kinect = PyKinectRuntime(PyKinectv2.FrameSourceTypes_Depth)
2 counter = 0
3 while True:
4     cv_frame = get_Depth_frame(Kinect)
5     image = copy(cv_frame) #Kinect suvertinama tik kartin an neriskiuo kaptjos daryt
6     image.flags.writeable = False
7     #Detection of pose
8     #recoloring
9     image.flags.writeable = True
10    image = image.copy()
11
12    #cv2.imshow("Recording KINECT Video Stream", image*15)
13    window_name = "projector"
14    cv2.namedWindow(window_name, cv2.WND_PROP_FULLSCREEN)
15    cv2.moveWindow(window_name, 0, 0)
16    cv2.setWindowProperty(window_name, cv2.WND_PROP_FULLSCREEN,
17    cv2.WND_PROP_FULLSCREEN)
18    cv2.imshow(window_name, image*15)
19    cv2.setMouseCallback(window_name, mousePoints)
20    key = cv2.waitKey(30)
21    if key == 27 or counter > 0: break
22
23    #key = cv2.waitKey(30)
24    #if key == 27: break
25    cv2.destroyAllWindows()
26    cv2.imwrite("picture%03d.jpg" % counter, cv2.cvtColor(image, cv2.COLOR_GRAY2BGR))
27    save("picture%03d.jpg" % counter, image)
28    Kinect.close()
```

2.5 pav. *Jupyter Notebook* - programavimo pavyzdys

### 2.4. Kita programinė įranga

*Matlab* yra vienas iš populiariausių programinių paketų tarp mokslininkų. *matlab* turi atskirą aplikaciją skirtą kamerų kalibravimui, vadinama *Camera Calibrator*.



**2.6 pav.** *Matlab - Camera Calibrator*

Ši programa naudojama gauti kameros židinio nuotolius, optinius centrus, radialinius bei tangentinius nuokrypių koeficientus. Programa leidžia gauti kameros vidinius ir išorinius parametrus naudojant skirtingus kalibravimo metodus [32].

### 3. Tyrimo dalis

Ištyrus įvairius šaltinius ir atsižvelgiant į turimą įrangą, pasirinkti 2 metodai, kuriais bus bandoma perteikti duomenis robotui.

Pirmasis metodas gauti 3D žmogaus skeleto koordinatas naudojantis tik *RGB* kamera. Jei šis metodas būtų pakankamai tikslus, būtų galima išvengti poreikio naudoti *RGB-D* kamerą taip išsaugant daug procesoriaus resursų.

Antras parinktas metodas naudoti *RGB-D* kamerą. Šis metodas suteikia daug daugiau informacijos apie kadre matomus objektus t. y. suteikia z ašies koordinatę. Todėl nebereikia jos nuspėti, taip gaunama tikslesnė informacija. Dirbant šiuo metodu tenka atmesti *Kinect SDK* skeleto sekimo metodą, nes *Kinect v2* skeleto sekimą galima naudoti tik Windows aplinkoje, Jį išbandyti būtų beprasmiška, nes *Jetson* valdiklyje veikia *Linux* operacinė sistema.

3.1 poskyrio skyreliuose pateikti metodai, kurie bus išbandyti siekiant gauti 3D žmogaus skeleto koordinatas gestams atpažinti. Skyrelyje 3.2 pateikti metodai, kuriais remiantis bus perskaiciuotos 3D koordinatos į kampus roboto pavaroms, siekiant imituoti žmogaus gestus.

#### 3.1. 3D žmogaus pozos koordinačių gavimas.

Šiuo metu labai populiarėja metodas, kai žmogaus skeleto 3D koordinatės yra išgaunamos naudojant *RGB* kamera. Tai buvo pasiekta naudojant *CNN* neuroninius tinklus ir įvairias šio tinklo struktūras. Šiuo metu dažniausiai naudojama šio tinklo architektūra naudoja likutinius neuroninius tinklus (angl. *Residual neural networks*) kuriuose pirminių sluoksnių išėjimai yra perkeliama į gilesnius sluoksnius, siekiant išvengti nykstančio gradiento problemos [33]. Tinklo mokymui yra naudojamos jau pagarsėjusios duomenų bazės, kaip pavyzdžiui *Human3.6m* turinti tris milijonus šešis šimtus tūkstančių žymėtų pavyzdžių [34], arba kūrėjų pačių sukurtos ar modifikuotos bazės.

##### 3.1.1. Neuroninis modelis *MobileHumanPose*.

*MobileHumanPose* modelis yra sukurtas naudoti mobiliose (mažai resursų turinčiuose) įrenginiuose, tai labai lengvas gilusis neuroninis tinklo modelis, galintis aptikti keleto žmonių skeletus pagal *COCO* architektūrą (t. y. aptinkami 17 taškų). Modelis yra paremtas *MobileNetV2* [35] modelio statymo bloku (statymo blokas, tai tam tikros neuroninių tinklų sluoksnių konfigūracijos). Tinklo statymo bloke autoriai pakeitė bloko pirmųjų sluoksnių dimensijas (3.1 pav. parodyti skirtumai tarp originaliojo modelio ir modifikuoto) tam, kad būtų galima naudoti didesnius paveikslėlius ir aktyvacijos funkciją iš *ReLU6* į *PReLU* (*ReLU6* didžiausias funkcijos rezultatas yra lygus 6, o *PReLU* rezultatas yra parametrizuojamas tinkle mokymo etapo metu). Šiame modelyje kaip ir daugelyje kitų modelių žmogaus pozos atpažinimui yra naudojama Enkoderio-dekoderio struktūra (dar vadinama smėlio laikrodžio struktūra) [36]. Šios struktūros pirmieji tinklo sluoksniai aptinka objektus, tada kaklelyje (tinklo susiaurėjime) yra atpažįstami objektai, o tolimesni nuo kaklelio (dekoderio) sluoksniai pažymi atrastus elementus. Bazinei neuroninių tinklų linijai autoriai taip pat naudoja praleidimo jungtis (3.2 pav.).

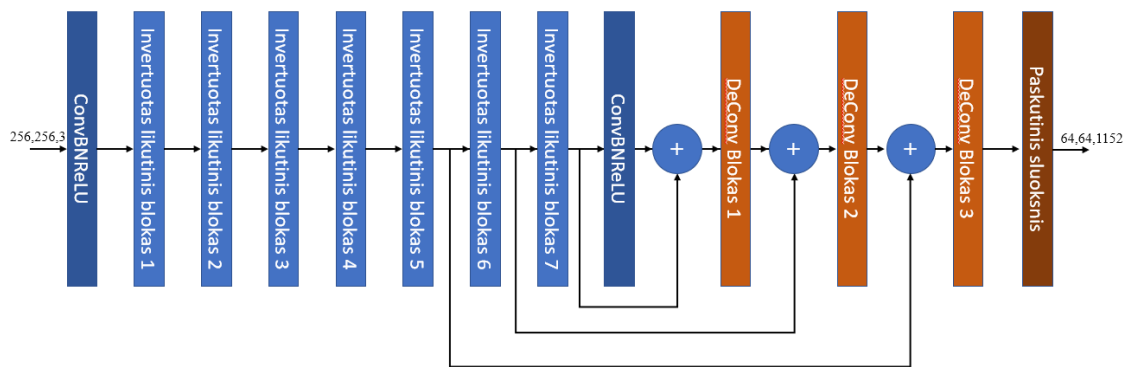
Duomenų įvadas	Operatorius	t	c	n	s
256x256x3	conv2d	-	48	1	2
128x128x48	bottleneck	1	64	1	2
64x64x64	bottleneck	6	48	2	2
32x32x48	bottleneck	6	48	3	2
16x16x48	bottleneck	6	64	4	2
16x16x64	bottleneck	6	96	3	2
8x8x96	bottleneck	6	160	3	1
8x8x160	bottleneck	6	320	1	1
8x8x320	conv2d 1x1	-	1280	1	1
8x8x1280	conv2d 1x1	-	-	1	-
1x1x1280	avgpool 8x8	-	k	-	-

a)

Duomenų įvadas	Operatorius	t	c	n	s
224x224x3	conv2d	-	32	1	2
112x112x32	bottleneck	1	16	1	1
112x112x16	bottleneck	6	24	2	2
56x56x24	bottleneck	6	32	3	2
28x28x32	bottleneck	6	64	4	2
14x14x64	bottleneck	6	96	3	1
14x14x96	bottleneck	6	160	3	2
7x7x160	bottleneck	6	320	1	1
7x7x320	conv2d 1x1	-	1280	1	1
7x7x1280	conv2d 1x1	-	-	1	-
1x1x1280	avgpool 8x8	-	k	-	-

b)

3.1 pav. a) *MobileHumanPose* naudojamas modifikuotas invertuotas likutinis blokas [37], b) originalus invertuotas likutinis blokas [35]

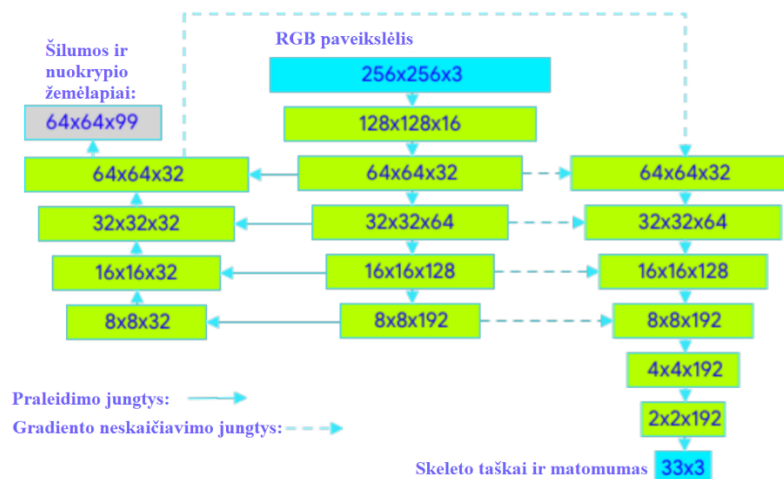


3.2 pav. *MobileHumanPose* modelio bendra struktūra [36]

Modeliui treniruoti autoriai naudojo *Human3.6M*, *COCO*, *MuCo-3DHP* duomenų bazes.

### 3.1.2. *MediaPipe* neuroninis modelis *BlazePose* .

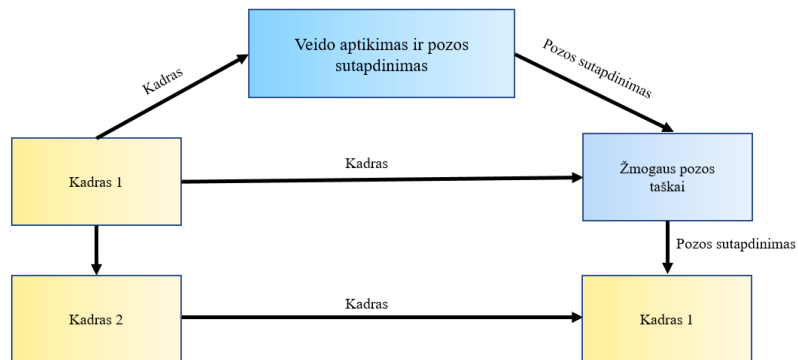
Darbe [38] pateiktas neuroninis tinklas pavadinimu *BlazePose*. Šis tinklas buvo apmokytas naudojant pačių autorių sukurtą duomenų bazę, iš viso su 66 tūkstančiais paveikslėlių, kurie visi buvo pažymėti žmogaus. *BlazePose* tinklas turi struktūrą kaip parodyta 3.3 pav.:



3.3 pav. *BlazePose* tinklo architektūra [38]

Tai yra smėlio laikrodžio architektūra. Modelis šilumos žemėlapius (angl. *Heat maps*) naudoja tik mokymo metu. Vėliau jie yra nenaudojami [38]. Praleidimo jungtys tai jungtys kurios informaciją iš pirminių tinklo sluoksnių perkelia į tolimesnius sluoksnius siekiant išvengti nykstančio gradiento problemos.

Tinkle naudojamas pirminis apdorojimas, kuris aptinka žmogaus veidą ir pagal tai sudeda visus žmogaus skeleto taškus. Tai atliekama tik pirminio žmogaus kontūro aptikimo metu, vėliau šis apdorojimas nebenaudojamas, iki tos akimirkos, kai dingsta žmogaus kontūras, tokiu atveju vėl yra įjungiamas pirminis apdorojimas [38].



**pav. 3.4** *BlazePose* veikimo algoritmas [36]

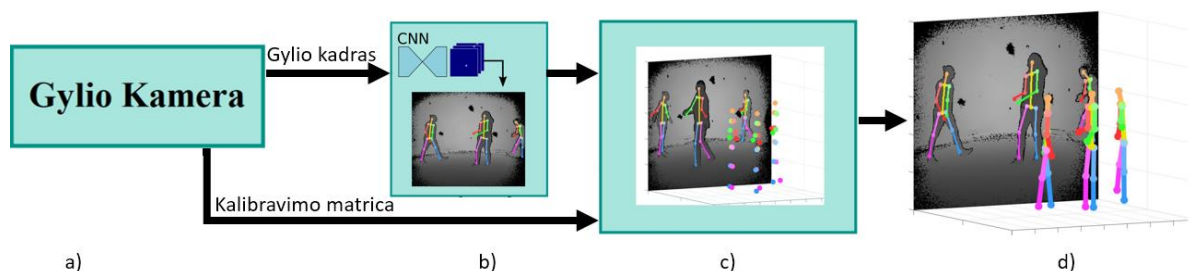
*BlazePose* modelis aptinka 33 žmogau kūno taškus [38].

Taip pat modeliui buvo pateikti augmentuoti (dirbtinai pakeisti) duomenys, tam jog tinklas gebėtų reaguoti į nežymius taškų pokyčius esant žmogaus judesiui, bei atvejus kai tam tikri žmogaus kūno taškai yra nematomi [38].

Iš viso *BlazePose* turi tris modelius *Light*, *Heavy* ir *Full*, iš kurių *Full* yra tiksliausias, bet reikalaujantis daugiausia resursų, o *Light* greičiausias, bet ne toks tikslus.

### 3.1.3. *Residual Pose* neuroninio tinklo modelis.

[39] šaltinyje pateiktas 3D koordinatų gavimo būdas, kuriame yra naudojama gylio kamera. Gylio kamera gautas kadras yra apdorojamas autorių parinktu *CNN* enkoderio-dekoderio struktūros modeliu, taip gaunant 2D žmogaus koordinatas paveikslėlyje. Gautos koordinatės iš paveikslėlio koordinatų sistemos yra perkeliamos į kameros koordinatų sistemą pasinaudojant kameros kalibravimo matrica. Perkeltos koordinatės yra koreguojamos autorių pasiūlytu 3D koordinatų koregavimo algoritmu. Koreguoti sąnarių koordinatų taškai yra apdorojami autorių sukurtu 3D koordinatų regresijos neuroninių tinklų modeliu, kuris sukuria 3D žmogaus pozos skeleto koordinatas.



**3.5 pav.** *Residual pose* algoritmo seka.



Algoritmo seka pavaizduota 3.5 pav. yra tokia:

- a. gaunamas gylio kadras.
- b. kadras apdorojamas 2D žmogaus pozos *CNN* modeliu.
- c. 2D koordinačių taškai perkeliama į kameros koordinačių sistemą (gaunamos 3D koordinatės) ir panaudojamas koordinačių korekcijos algoritmas.
- d. panaudojamas 3D pozos regresijos modelis galutinėms koordinatėms gauti.

#### 3.1.4. Siūlomas metodas

Pasinaudojus tokiu pat principu kaip ir *ResidualPose* galima sukurti naują algoritmą panaudojant naujausius šiuo metu esančius 2D žmogaus pozicijos modelius. Algoritmas, kuris atliekamas naudojant gylio ir infraraudonos spalvos kadrus:

1. gaunami gylio ir infraraudonos šviesos kadrai;
2. atliekamas IR kadro perskaliavimas;
3. infraraudonos spalvos kadras apdorojamas naujausiais 2D žmogaus pozos aptikimo modeliais;
4. 2D koordinačių taškai perkeliama į pasaulio koordinačių sistemą, kur z koordinatės yra gaunamas infraraudonos spalvos kadro žmogaus pozos taškus perkeliama į gylio kadra;
5. panaudojamas koordinačių korekcijos algoritmas:
  - a. papildomai gautų taškų neapdoroti;
  - b. panaudoti algoritmą duomenų apdorojimui iš *ResidualPose* algoritmo;
6. gaunama 3D žmogaus poza naudojant skirtingus būdus:
  - a. loginis taškų atidėjimas naudojantis gylio arba papildomais 2D modelio teikiamais duomenimis;
  - b. naudojant *ResidualPose* regresoriaus modelį.

Siūlomo metodo atveju naudojami infraraudonos spalvos kadrai tam, kad modeliai, kurie yra treniruoti naudojant *RGB* kadrus, galėtų pasinaudoti įvairiais žmogaus kūno bruožais ir tiksliau nustatyti žmogaus sąnarių taškus (*CNN* ypatybių filtrai buvo apmokyti naudojant *RGB* kameras, todėl yra didelė tikimybė, jog tokie taškai kaip akys, nosis t. t. yra atrandami pagal tam tikrą ypatybę, kurios negalima pastebėti gylio kadre). *IR* kadru taškų vertės yra labai plačios: 0-65535. Tam, jog nebūtų prarandama didelė dalis informacijos konvertuojant iš *IR* į *RGB* formatą, nuotrauka perskaliuojama pagal formulę:

$$Pikselis_{RGB} = \begin{cases} \frac{Pikselis_{IR}}{Param} * 254, & \text{Je}i \text{ Pikselis}_{IR} \leq Param \\ 254, & \text{Je}i \text{ Pikselis}_{IR} > Param \end{cases} \quad (1)$$

Čia *Param* yra parametras, kuriuo pašalinamos *IR* kadro vertės, kurios nesuteikia informacijos apie kadre esantį asmenį.



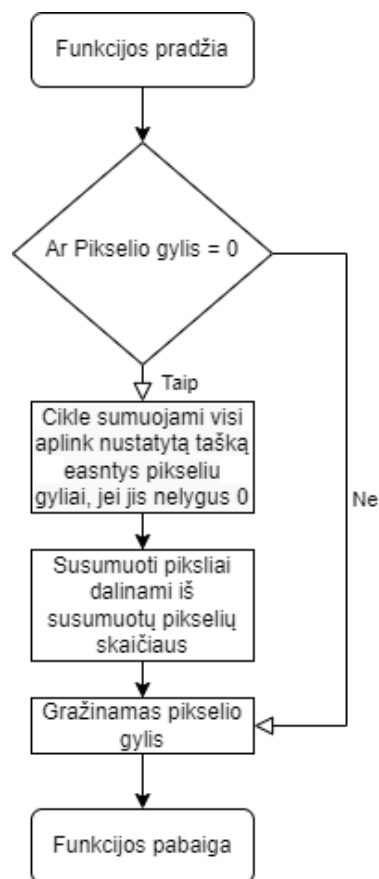
3.6 pav. a) gylio kadras, b) infraraudonos spalvos kadras

Tada žmogaus 2D koordinatės galima gauti naudojant tokius modelius *Posenet*, *BlazePoze* (nenaudojant 3D koordinatėms), *Movenet* ir kitus šiam tikslui sukurtus modelius.

Gavus žmogaus skeleto taškus *IR* kadre, juos galima tiesiogiai perkelti į gylio kadra t. y:

$$IR \text{ kadro pikselio koordinatė} = Gylio kadro pikselio koordinatė$$

Gylio kadras turi taškų, kurie yra neapibrėžti ir jų vertė lygi nuliui. Jei aptikto taško gylio vertė lygi nuliui (nėra gylio informacijos), tokiu atveju taškas užpildomas algoritmu:



3.7 pav. Pikselio užpildymo algoritmas

Čia pikseliai aplink nustatytą tašką yra pikseliai nutolę per nurodytą skaičių pikselių spindulį paveikslėlio koordinatinių sistemoje.

Gauti 2D žmogaus pozos taškai perkeliami į 3D koordinatinių sistemą naudojant formules [40]:

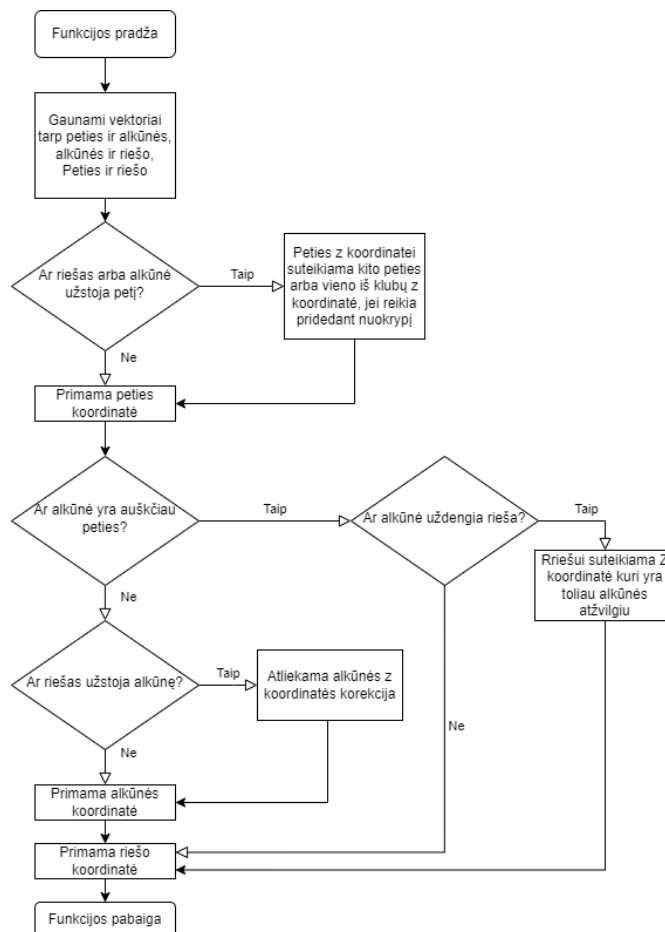
$$P' = \begin{cases} P_x' = \frac{P_x - C_x}{F_x} P_z \\ P_y' = \frac{P_y - C_y}{F_y} P_z \\ P_z' = P_z \end{cases} \quad (2)$$

Čia  $P'$  yra realaus pasaulio 3D koordinatė. Čia  $F_x$  ir  $F_y$  yra kalibravimo matricos židinio nuotoliai, o  $C_x$  ir  $C_y$  kameros centro koordinatės.

Gautos 3D koordinatės gali uždengti viena kitą (pvz. ranka ištiesta link kameros savo plaštaka užstoja alkūnę ir petį), todėl vien perkelti taškus neužtenka. Tam, kad 3D koordinatės būtų teisingai atvaizduojamos, galima naudotis *ResidualPose* 3D koordinatinių regresoriumi, arba loginiu būdu išskaičiuoti tariamą žmogaus sąnario vietą.

Remiantis tuo, jog žmogus turi stovėti priešais kamerą bei kūno galūnės turi hierarchiją galima loginiu būdu išskaičiuoti atskiras kūno dalių koordinates.

Siūlomas algoritmas pateiktas 3.8 pav.:



3.8 pav. 3D koordinatinių koregavimo algoritmo diagrama

Ar vienas iš sąnarių yra užstotas patikrinama naudojant vektorių koordinačių dydžius lyginant su nustatytomis koordinačių ribomis. X ir y koordinatėms tikrinti yra pasirenkama paveikslėlio 2D koordinačių sistema, o z koordinatė yra tikrinama pagal pasaulio koordinačių sistemą t. y. nuotolį metrais (gyli):

$$\begin{cases} |\vec{V}_{AR}(x)| < H_x \\ |\vec{V}_{AR}(y)| < H_y \\ |\vec{V}_{AR}(z)| < H_z \end{cases} \quad \text{Patenkinus sąlygas, atliekama korekcija} \quad (3)$$

Čia pavaizduotas alkūnės, riešo vektoriaus patikrinimas. Esant koordinatėms mažesnėms, nei užduotos ribos ( $H_{xyz}$ ), atliekama koordinačių korekcija.

Peties korekcija labai paprasta – jai yra suteikiama žmogaus korpuso koordinatė kuri yra labiausiai nutolusi nuo kameros, pirmenybę teikiant klubams, nes petys turi didesnę galimybę judėti erdvėje.

Alkūnės korekcija yra sudėtingesnė, nes jos padėtis erdvėje yra daug mažiau suvaržyta nei peties, todėl korekcija gali būti atliekama dviem atvejais:

1. riešas, petys ir alkūnė yra vienoje tiesėje;
2. riešas ir alkūnė yra vienoje tiesėje, o petys ne.

Pirmu atveju alkūnės z koordinatę gauname pagal formulę:

$$P_A(z) = \frac{P_P(z) + P_R(z)}{2} \quad (4)$$

Čia atitinkamai  $P_A(z)$  alkūnės,  $P_P(z)$  peties ir  $P_R(z)$  riešo koordinačių z koordinatė.

Antras atvejis sudėtingesnis, nes reikia įvertinti, kampą tarp peties ir alkūnės. Alkūnė gali būti priekyje arba už korpuso, arba ji gali būti lygiagreči korpusui. Tam įvertinti naudojama tiesės lygtis [40]. Darant prielaidą, jog rankos atkarpa yra apytiksliai tiesė, panaudojama tiesės lygties formulė trimatėje erdvėje z koordinatės vertei gauti:

$$P_A(z) = \frac{P_A(y) - P_{pagalb}(y)}{\vec{V}_{krypties}(y)} \vec{V}_{krypties}(z) + P_{pagalb}(z) \quad (5)$$

Čia  $P_{pagalb}$  yra gaunamas pagal formulę:

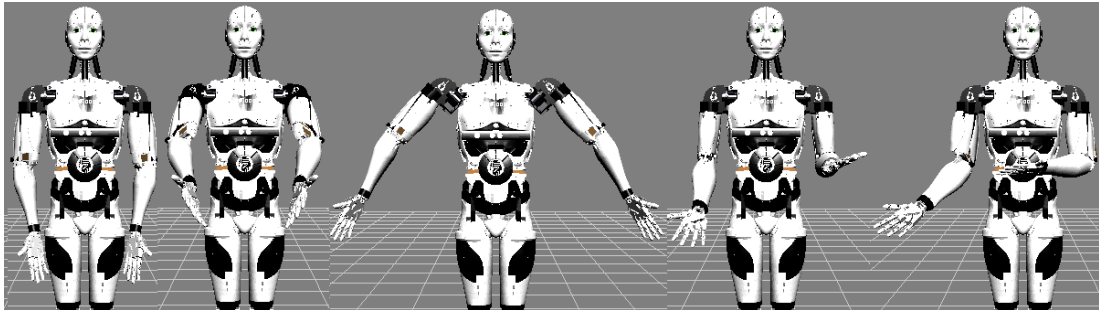
$$P_{pagalb} = \begin{cases} \frac{P_A(x) + P_P(x)}{2} \\ \frac{P_A(y) + P_P(y)}{2} \\ P_{pagalb}(z) = \text{GylioKadras}(P_{pagalb}(x, y)) \end{cases} \quad (6)$$

$\vec{V}_{Krypties}$  vektorius gaunamas taip [41]:

$$\vec{V}_{krypties} = \begin{cases} P_{Pagalb}(x) - P_P(x) \\ P_{Pagalb}(y) - P_P(y) \\ P_{Pagalb}(z) - P_P(z) \end{cases} \quad (7)$$

### 3.2. Kampų skaičiavimas

*InMoov* roboto ranka turi keturis laisvės laipsnius. Jis gali ranka tiesyti pirmyn, į šoną, sulenkti ranką per alkūnę, ir sukti petį.



3.9 pav. *InMoov* roboto rankos judėjimo galimybės

Kampams apskaičiuoti naudojama geometrinė kampų apskaičiavimo metodika kaip [42] šaltinyje, atsižvelgiant į *Kinectv2* koordinačių sistemą (žr. 3.10 pav.) ir *InMoov* roboto kampų sistemą.



3.10 pav. *Kinectv2* koordinačių sistema

Kampų pavadinimai ir paaiškinimai:

1.  $K_1$  – tai kampas, kuris gaunamas ranką tiesiant link kameros (žr. 3.9 pav. antrą pozą);
2.  $K_2$  – tai kampas, kuris gaunamas ranką tiesiant į šoną lygiagrečiai liemeniui (žr. 3.9 pav. trečią pozą);
3.  $K_3$  – tai kampas, kuris gaunamas ranką lenkiant per alkūnę (žr. 3.9 pav. ketvirtą pozą);
4.  $K_4$  – tai kampas, kuris gaunamas sukant petį (žr. 3.9 pav. penktą pozą).

Visi kampų skaičiavimo metodai pateikti žemiau esančiose lygtyse yra sutapatinti su *InMoov* koordinačių sistema.

$K_1$  kampas yra apskaičiuojamas naudojant vektorių, kuris sudaromas iš peties ir alkūnės sąnarių taškų:

$$\vec{V}_{PA} = P_A' - P_P' \quad (8)$$

Tada kampas gaunamas pagal formules:

$$K_1 = \begin{cases} \deg\left(\frac{\pi}{2} + \arctg\left(\frac{\vec{V}_{PA}(y)}{\vec{V}_{PA}(z)}\right)\right) + 30^\circ, & \text{Jei } \vec{V}_{PA}(z) < 0 \\ \deg\left(\arctg\left(\frac{\vec{V}_{PA}(z)}{\vec{V}_{PA}(y)}\right)\right) + 30^\circ, & \text{Jei } \vec{V}_{PA}(z) > 0 \text{ ir } \vec{V}_{PA}(y) < 0 \\ \deg\left(-\arctg\left(\frac{\vec{V}_{PA}(z)}{\vec{V}_{PA}(y)}\right)\right) + 30^\circ, & \text{Jei } \vec{V}_{PA}(z) > 0 \text{ ir } \vec{V}_{PA}(y) > 0 \end{cases} \quad (9)$$

Čia  $\deg()$  yra radianų į laipsnius vertimo funkcija. *InMoov* roboto rankos atkarpa tarp peties ir alkūnės yra statmena liemeniui tik pavarai uždavus  $30^\circ$  kampą, todėl yra pridedama konstanta.

Kai  $y$  ir  $z$  koordinatės sunyksta aukščiau pateikta formulė nebegali gerai nusakyti  $K_1$  kampo. Todėl naudojamos formulės remiantis [43] šaltiniu:

$$\cos(K_1) = \frac{1 + |\vec{V}_{PA}|^2 - \left(\sqrt{(P_A'(x) - 1)^2 + P_A'(y)^2 + P_A'(z)^2}\right)^2}{2|\vec{V}_{PA}|} \quad (10)$$

$$\sin(K_1) = \sqrt{1 - \cos(K_1)^2} \quad (11)$$

$$K_1 = \deg\left(\arctg\left(\frac{\sin(K_1)}{\cos(K_1)}\right)\right) + 30^\circ \quad (12)$$

$K_2$  kampas skaičiuojamas naudojant du vektorius: vektorių tarp klubų  $\vec{V}_K$  ir vektorių tarp peties ir alkūnės  $\vec{V}_{PA}$ . Vektorius tarp klubų yra pasirinktas todėl, nes yra stabilesnis lyginant su vektoriumi tarp pečių, kuris yra mobilesnis. Prieš naudojant formulę, abu vektoriai yra normalizuojami:

$$\vec{V}_{PA_n} = \frac{\vec{V}_{PA}}{|\vec{V}_{PA}|} \quad (13)$$

Tada formulė kampui gauti:

$$K_2 = \deg\left(\frac{\pi}{2} - \arccos\left(\vec{V}_{PA_n} \cdot \vec{V}_{Kn}^T\right)\right) \quad (14)$$

Čia tarp vektorių yra atliekama matricinė daugyba.

Kampas  $K_3$  gaunamas naudojant formulę:

$$K_3 = 90^\circ - \deg\left(\arccos\left(\vec{V}_{PA_n} \cdot \vec{V}_{AR_n}^T\right)\right) \quad (15)$$

Kampas  $K_4$  apskaičiuojamas naudojantis trimis vektoriais  $\vec{V}_P$  – vektoriaus tarp pečių,  $\vec{V}_{PA}$  ir  $\vec{V}_{AR}$ . Naudojant vektorinę sandaugą apskaičiuojami plokštumų susidarančių tarp vektorių  $\vec{V}_{PA}$  ir  $\vec{V}_{AR}$ , bei  $\vec{V}_{PA}$  ir  $\vec{V}_S$  normalės:

$$\vec{N}_1 = \vec{V}_{PA} \times \vec{V}_{AR} \quad (16)$$

$$\vec{N}_2 = \vec{V}_{PA} \times \vec{V}_P \quad (17)$$

Abu normalės vektoriai normalizuojami ir kampas yra apskaičiuojamas pagal formulę:

$$K_4 = \deg \left( \pi - \arccos \left( \vec{N}_{1n} \cdot \vec{N}_{2n}^T \right) \right) \quad (18)$$

Kampas  $K_4$ , kaip ir  $K_I$  tampa neapibrėžtas, kai  $\vec{V}_{PA}$  z ir y koordinatė sunyksta, todėl pasiekus labai mažą y ir z vertes, formulė, pagal kurią skaičiuojamas kampas  $K_4$  skaičiuojamas pakeitus normalę  $N_2$  taip:

$$\vec{N}_2 = \vec{V}_{PA} \times \vec{V}_K \quad (19)$$

Tada kampas  $K_4$ :

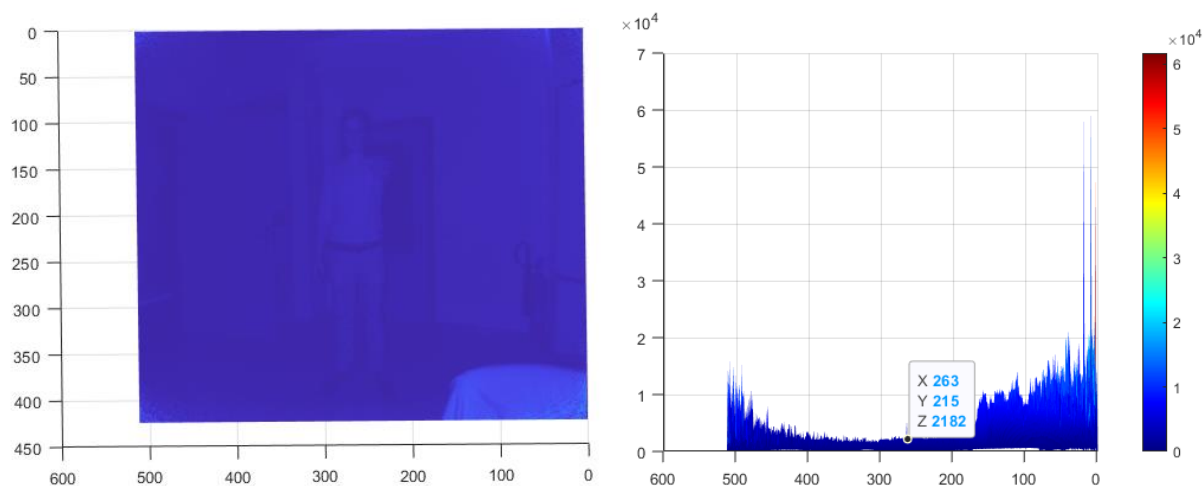
$$K_4 = \deg \left( \arccos \left( \vec{N}_{1n} \cdot \vec{N}_{2n}^T \right) \right) \quad (20)$$

Abiejų rankų kampam skaičiuoti yra naudojamos tos pačios formulės.

## 4. Eksperimentinė dalis

### 4.1. IR kadro perskaliavimas.

Naudojantis *Matlab* paketu buvo sukurta IR kadro histograma (4.1 pav.).



4.1 pav. IR kadro histograma

Atlikus bandymus buvo nustatyta, jog vertės, kurios viršija 5000, gali būti atmestos, todėl perskaliuojant IR kadrus bus naudojama 5000 riba.

Rezultatas pavertus kadra iš IR į RGB formatą su skaliavimu ir be jo pateiktas žemiau:

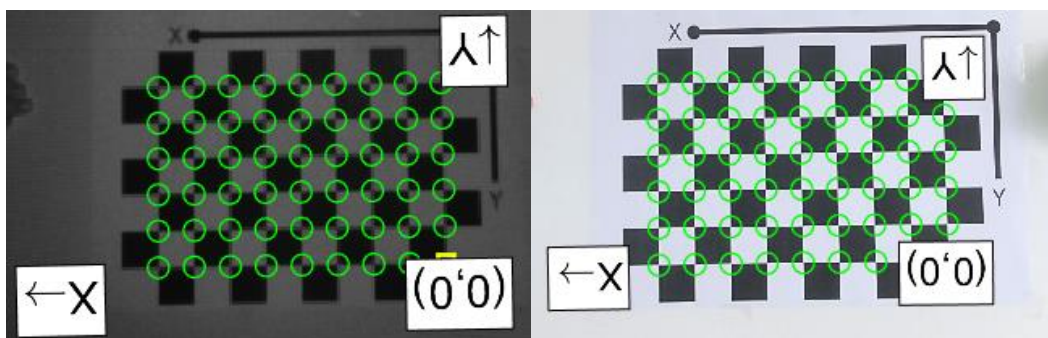


4.2 pav. Neskaliuotas ir perskaliuotas IR kadras

### 4.2. Kinect kameros kalibravimo rezultatai.

*HumanMobilePose* bei *ResidualPose* modeliams yra reikalingos kameros kalibravimo matricos. Kalibravimas buvo atliktas naudojantis *Matlab* įskiepiu *Camera calibrator* ir naudojant šaškių lentos šabloną. Kiekvienai kamerei buvo panaudota po 29 skirtingus kadrus iš RGB ir IR kamerų.





4.3 pav. Kalibravimo: kairėje IR kadras, dešinėje RGB kadras

Gauta gylio kameros kalibravimo matrica:

$$M_{GylioK} = \begin{bmatrix} 364.683 & 0 & 257.003 \\ 0 & 364.112 & 205.491 \\ 0 & 0 & 1 \end{bmatrix}$$

Gauta RGB kameros kalibravimo matrica:

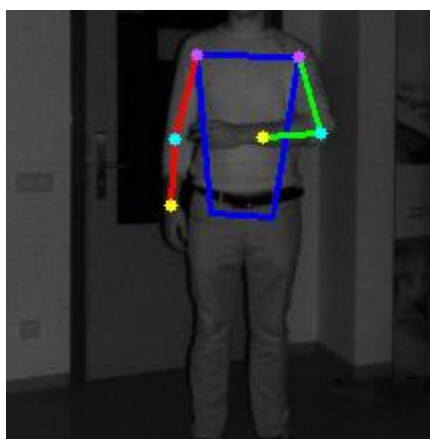
$$M_{RGBK} = \begin{bmatrix} 1075.324 & 0 & 957.147 \\ 0 & 1073.917 & 531.871 \\ 0 & 0 & 1 \end{bmatrix}$$

RGB kameros kalibravimo matrica reikalinga *MobileHumanPose* modelio veikimui. Gylio kamera reikalinga rastų 2D taškų perkėlimui į kameros koordinatų sistemą (3D koordinatų sistema) *ResidualPose* modeliui bei siūlomam algoritmui.

#### 4.3. 2D Žmogaus kūno taškų aptikimo bandymai.

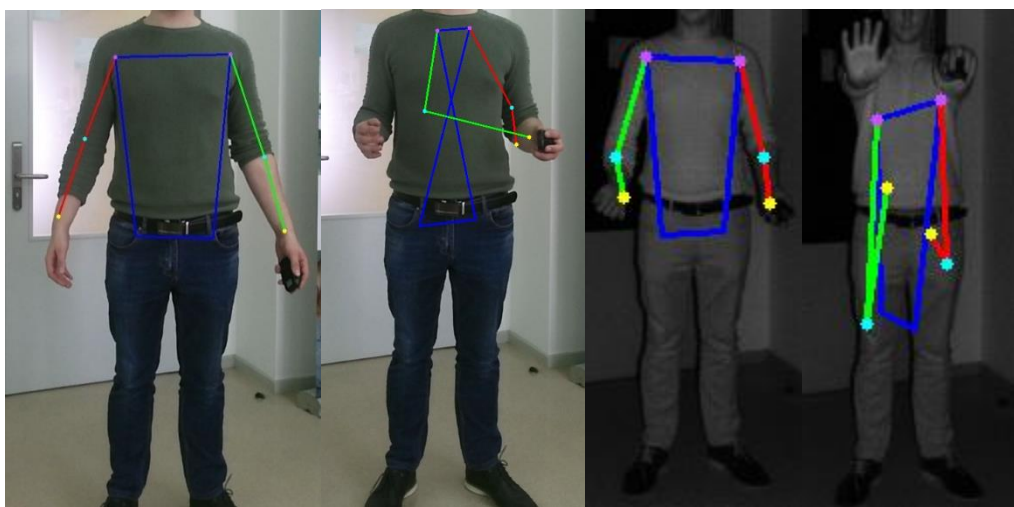
Šio bandymo metu buvo vertinamas modelio spėjimo greitis ir modelio tikslumas. Vertinimui buvo naudojami 33 paveikslėliai, kuriuose objektas stovi tiesiai priešais kamerą įvairiomis rankų pozicijomis.

Pozicija laikoma teisinga jei gaunamas žmogaus kūno kontūras yra panašus į žmogaus atliekama pozą neatsižvelgiant į atkarpų ilgius, nes kampų apskaičiavimui (2D koordinatų sistemoje), įtakos neturi. Pateiktame pavyzdyje 4.4 pav. matyti, jog kairės nuspėta riešo koordinatė nėra ant riešo, bet matyti, jog ir tiksliau padėjus riešo tašką,  $K_3$  kampas išliks panašus.



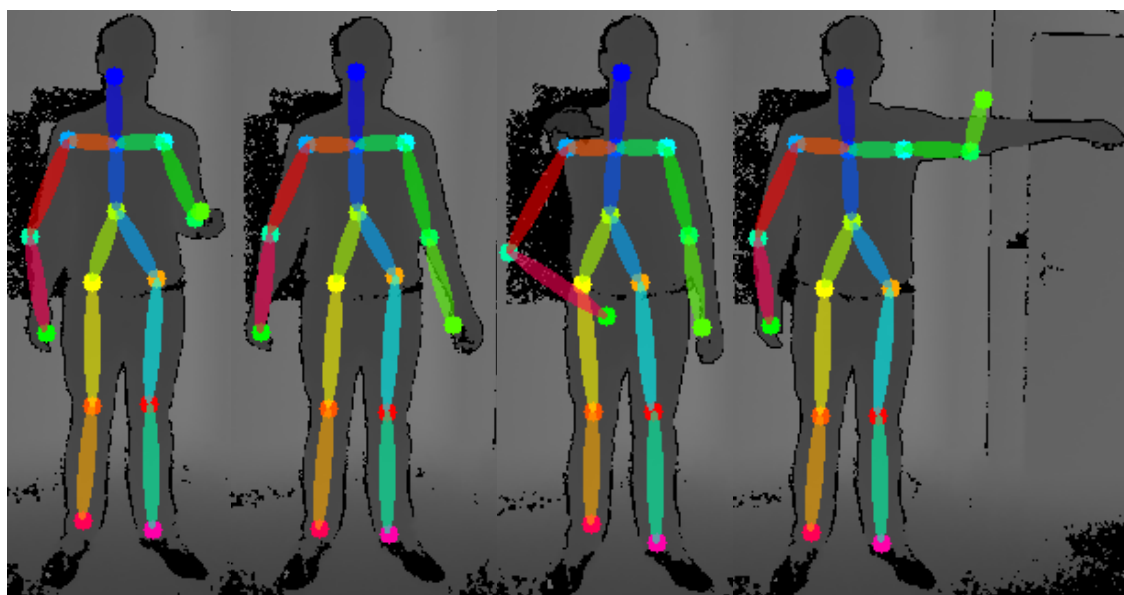
4.4 pav. Teisingu spėjimu laikomas paveikslėlis

Prasčiausi rezultatai buvo gauti naudojant *MobileHumanPose*. Bandymai buvo atlikti *RGB* ir *IR* kadrais. Modelio greitis geras, vidutiniškai kadras buvo apdorotas per 49 *ms* *RGB* kadro atveju, ir 53 *ms* *IR* atveju. Abejais atvejais rezultatai buvo labai prasti. *RGB* kadru atveju buvo gautos 2 teisingos pozos, *IR* atveju tik 1 teisinga poza. Modelis tampa labai nestabilus jei uždengiamas bent vienas žmogaus kūno taškas (žr. 4.5 pav.). Stovint natūralioje pozicijoje tiesiai prieš kamerą modelis dar susitvarko su užduotimi, bet jokia kita poza nėra spėjama teisingai.



4.5 pav. *MobileHumanPose* modelio bandymų rezultatai: *RGB* kairėje, *IR* dešinėje

*ResidualPose* modelio gauti geresni rezultatai. Iš viso teisingai buvo nustatytos 7 pozos, bet modelio greitis prastas. Vidutiniškai kadras buvo apdorotas per 2.3 sekundes. Norint gauti pilnus rezultatus reikalingas laikas siekia 2.6 sekundes.



4.6 pav. *ResidualPose* 2D pozos aptikimo rezultatai

Naudojant *BlazePose* modelį su *RGB* kadrais, rezultatai daug geresni. Naudojant lengvą (angl. *Light*) modelį nustatyta 30 teisingų pozų, o greitaveika siekia 30 *ms* per kadram. Sunkus (angl. *Heavy*) modelis buvo ne daug tikslesnis – nustatyta 31 teisinga poza iš 33 su 45 *ms* greitaveika. Pilnas (angl. *Full*) modelis nustatė teisingai net 32 pozas, bet greitaveika suprastėja iki 162 *ms* (5.4 karto lėtesnis už lengvą modelį ir 3.6 karto lėtesnis už sunkų modelį). 4.7 pav. pateiktos penkios pozos iš kurių *lengvas*

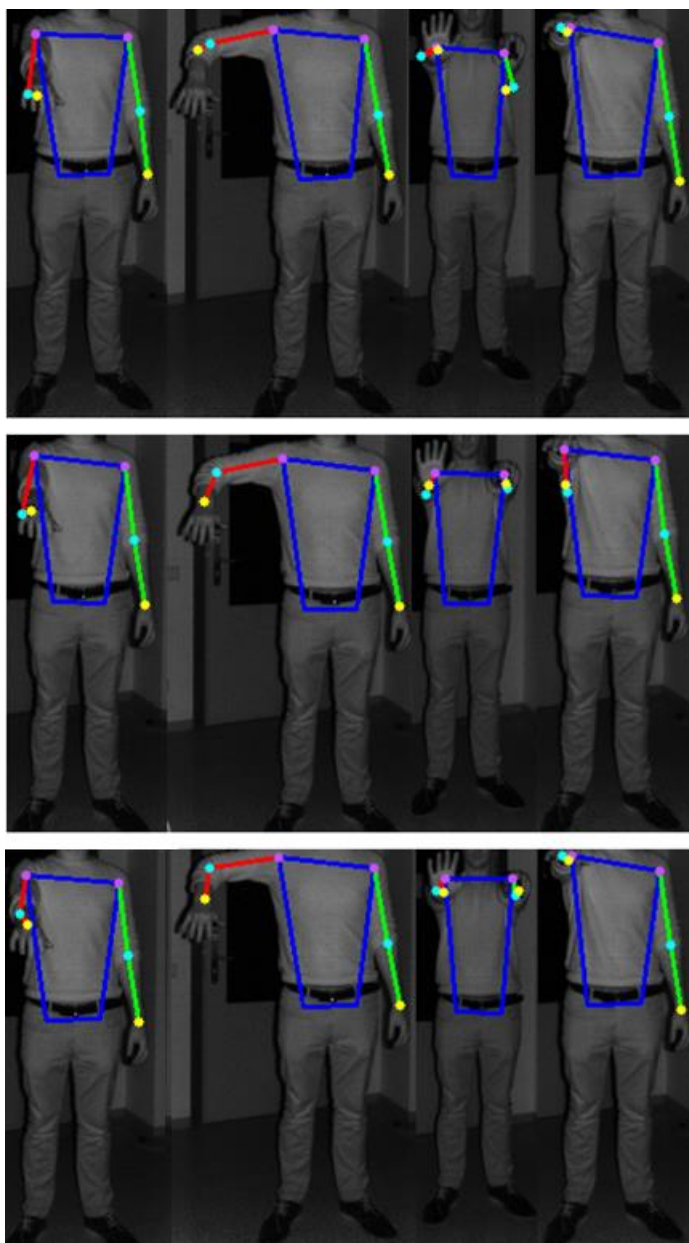
modelis neteisingai nustatė žmogaus pozą pirmuose 3 paveikslėliuose, sunkus antroje ir ketvirtoje, o pilnas tik paskutiniame paveikslėlyje nustatė neteisingą pozą. Tikslumas išlaikomas labai geras, kol rankos nėra tiesiamos pirmyn.



**4.7 pav.** *BlazePose* pozos nustatymas naudojant *RGB* kadrus. Pirmą eilę – lengvas modelis, antra – sunkus, trečia – pilnas.

*BlazePose* naudojant su *IR* kadru gauti labai panašūs rezultatai kaip ir su *RGB* kadrų, nors duomenų kiekis sumažėja dėl spalvų nebuvimo. Lengvas ir sunkus modeliai nustatė teisingai 30 pozų, o jų

greitaveika atitinkamai 38 ir 54 milisekundės. Sunkus modelis buvo vienintelis modelis, kuris teisingai nustatė visas pozas pateiktas paveikslėliuose. Paveikslėlyje žemiau galima matyti 4 kadrus kuriuose pozos buvo nustatytos su klaidomis. Lengvas modelis klaidingai nustatė pozas pirmuose trejuose kadruose, sunkus neteisingai pozą nustatė pirmame, antrame ir ketvirtame kadre, o pilnas modelis klaidų kadruose nepadarė.



**4.8 pav.** *BlazePose* pozos nustatymas naudojant IR kadrus. Pirma eilė – lengvas modelis, antra – sunkus, trečia – pilnas.

Lentelėje žemiau pateikti apibendrinti rezultatai.

**4.1 lentelė.** Modelių 2D koordinačių nustatymo bandymo rezultatai

Modelis	Konfiguracija	Paveikslėlio tipas	Greitis	Atitikmuo
BlazePose	Light (0)	<i>RGB</i>	30ms	30/33
BlazePose	Heavy (1)	<i>RGB</i>	45ms	31/33
BlazePose	Full (2)	<i>RGB</i>	162ms	32/33
BlazePose	Light (0)	<i>IR</i>	38ms	30/33

BlazePose	Heavy (1)	IR	54ms	31/33
BlazePose	Full (2)	IR	169ms	33/33
MobileHumanPose	-	RGB	49ms	2/33
MobileHumanPose	-	IR	53ms	1/33
ResidualPose	GPU	Gylio	2301ms	7/33

Rezultatai rodo, jog toliau bandyti *ResidualPose* ir *MobileHumanPose* neverta. Todėl 3D koordinatėms nustatyti bus bandomi tik *BlazePose* modeliai. Taip pat bus išbandyti ir *ResidualPose* 3D korekcijos ir regresijos algoritmai 3D koordinatėms gauti.

#### 4.4. Kampų nustatymo – pozos atitikties bandymas

Prieš pradėdant šį bandymą parinkti parametrai kampams  $K_1$  ir  $K_4$ . Kampui  $K_1$  parinktas  $\vec{V}_{PA}(y)$  koordinatei sumažėjus iki 3 cm. Kampui  $K_4$  parinktas  $\vec{V}_{PA}(y)$  sumažėjus iki 6 cm o  $\vec{V}_{PA}(z)$  koordinatei padidėjus iki -5 cm (minuso ženklas dėl *Kinectv2* koordinatės sistemos).

Alkūnės pozicijos korekcijai (ribos  $H_{xyz}$ ) parinktas 20 pikselių skirtumas x, y koordinatėms ir 5 cm z koordinatės skirtumas.

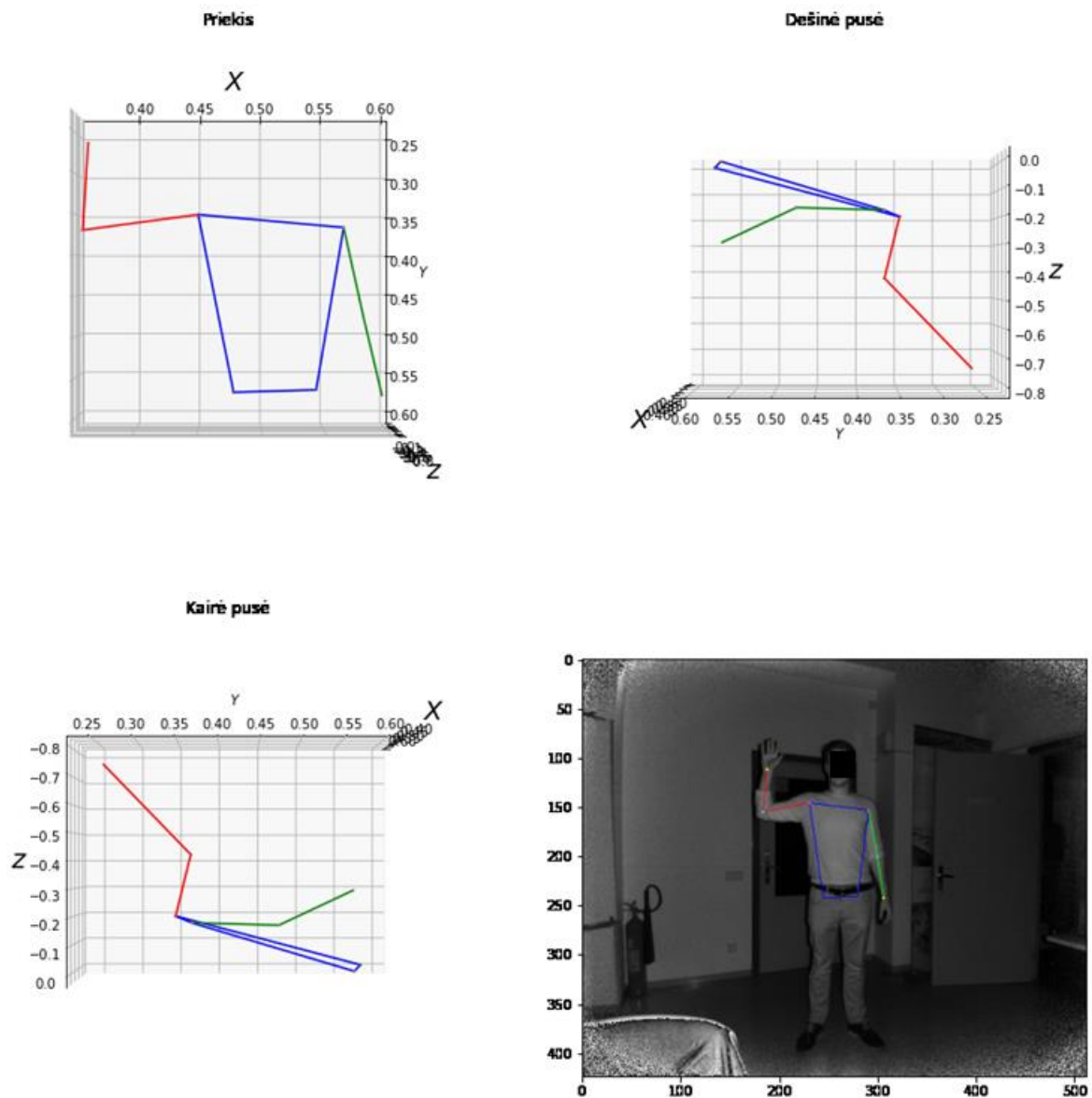
Šio bandymo tikslas yra įsitikinti, jog kampai gauti pasinaudojus modelio ar algoritmo suteiktomis 3D sąnarių koordinatėmis, yra tinkami siekiant, kad robotas būtų panašioje pozoje kaip ir žmogus. Šio bandymo metu svarbu jog vizualiai robotas atkartotų žmogaus pozą. Poza laikoma teisinga jei roboto poza panaši į žmogaus arba yra pasiektos roboto pavarų ribos ir tik dėl to robotas negali atkartoti gesto pilnai.

Lentelėje žemiau pateikiami bandymų rezultatai:

**4.2 lentelė.** Kampų atitikties bandymų rezultatai

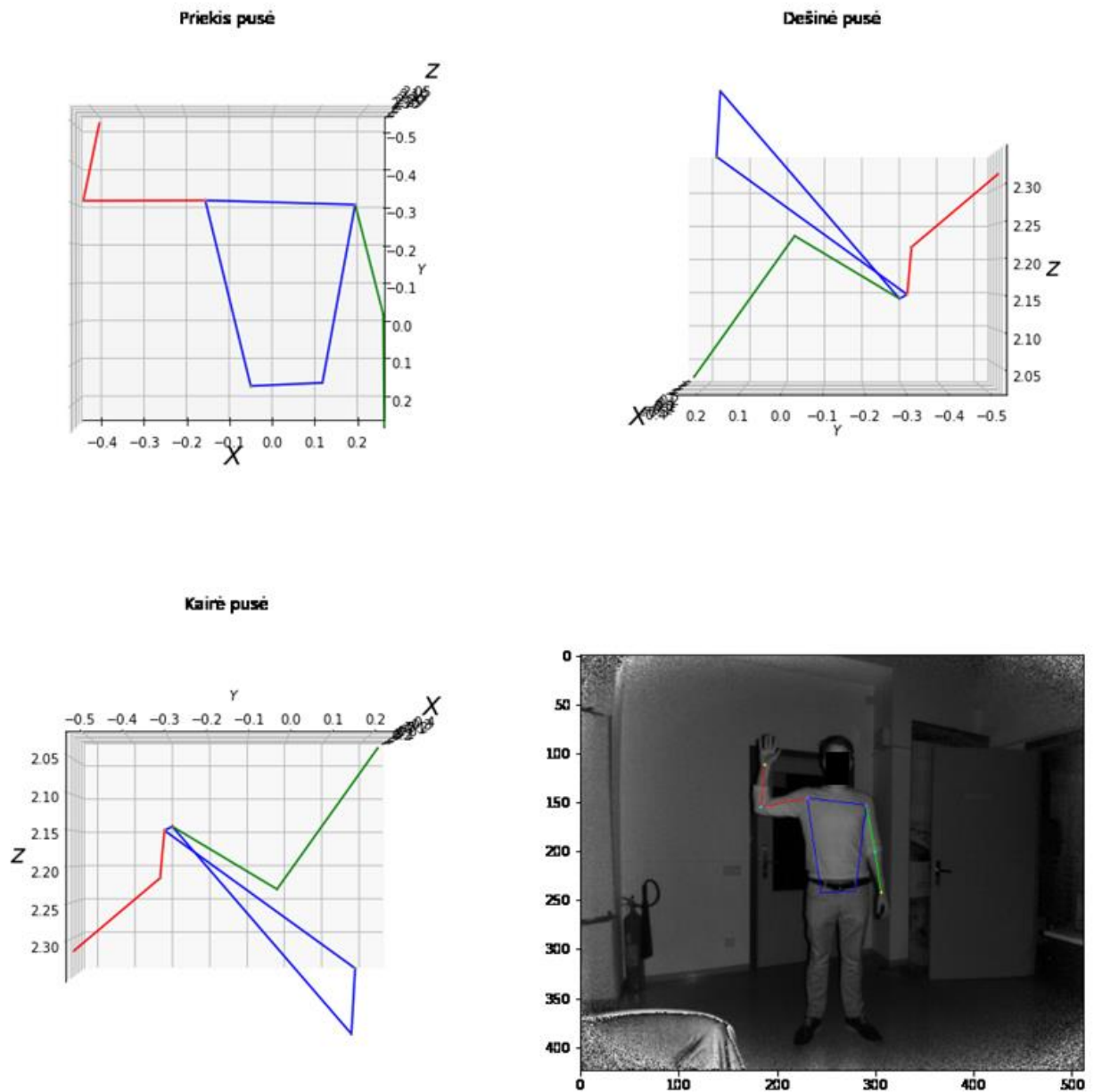
Metodas	Atitiktis	Atitiktis %
<i>BlazePose Full</i>	1/33	3%
<i>ResidualPose</i>	2/33	6%
<i>BlazePose</i> su gylio informacija ir korekcija	28/33	85%

Bandymu rezultatai parodė, jog *BlazePose* algoritmas nors ir išduoda 3D koordinates, bet jis nėra tinkamas pozai nustatyti. *BlazePose* išduoti 2D taškai yra tikslūs, bet jo apskaičiuojama z koordinatė yra labai netiksli, žr. 4.9 pav. Iš paveikslėlio matyti, jog žmogaus ranka yra apytiksliai lygiagreči žmogaus liemeniui, tačiau gautos koordinatės rodo, jog žmogus ranką yra ištiesęs į priekį, todėl šiuo atveju gauti kampai buvo neteisingi. Testai buvo atlikti su *RGB* ir *IR* nuotraukomis bet skirtumo tarp jų efektyvumo nebuvo.



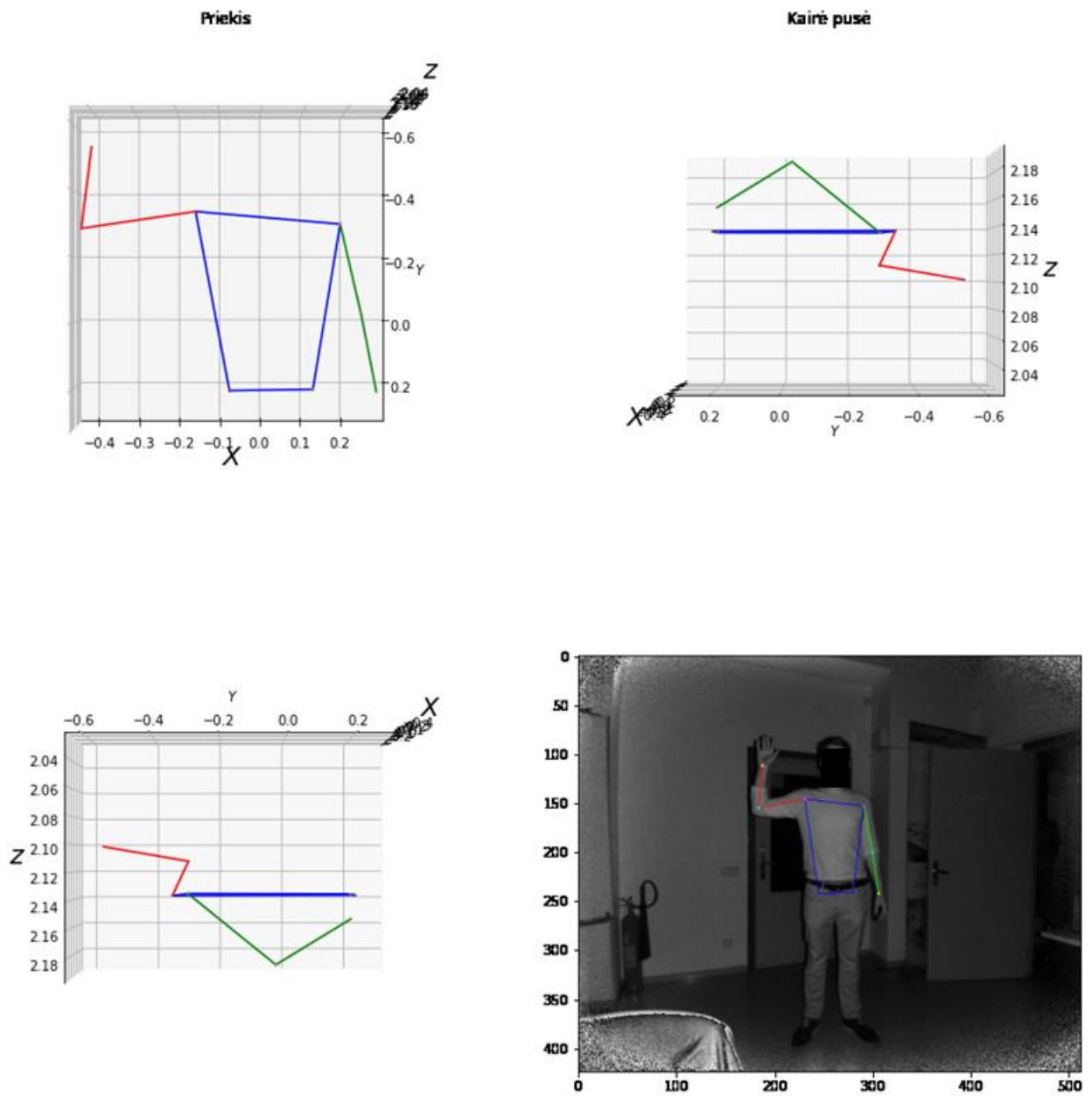
4.9 pav. *BlazePose* išduotos 3D koordinatės

Naudojant gylio kadrus ir *ResidualPose* korekcijos algoritmą taip pat nebuvo gauti geri rezultatai. Regresijos neuroninis tinklas kartu su korekcijos algoritmu taip pat veikė labai netiksliai, paveikslėlyje žemiau matyti gautas rezultatas po regresijos.



4.10 pav. *ResidualPose* 3D koordinatės

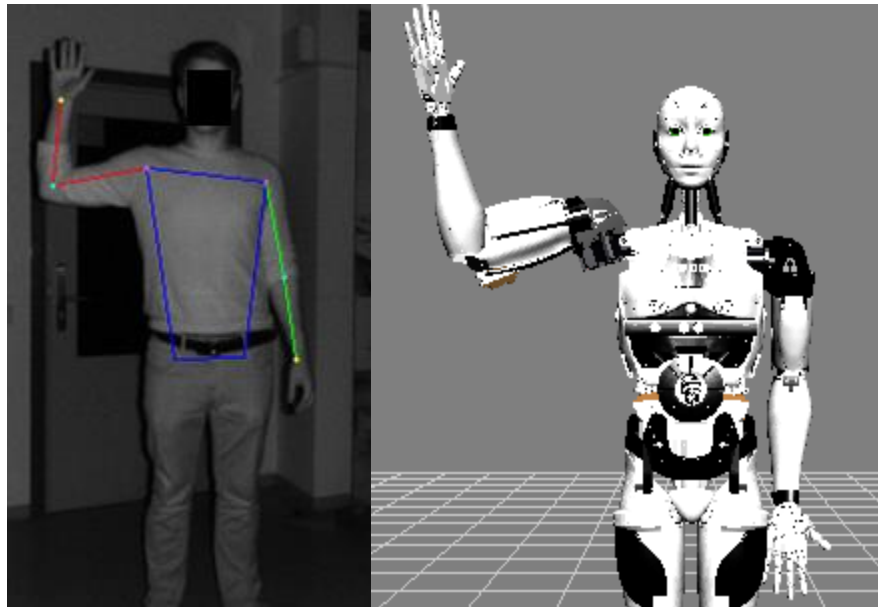
Naudojantis siūlomu algoritmu, rezultatai buvo daug geresni. Paveikslėlyje žemiau yra pateiktas pavyzdys, kai korekcijos algoritmas naudojamas tik dalinai (žmogaus liemens z koordinatės suvienodintos) t. y. paimamos 2D koordinatės naudojantis *BlazePose* modeliu, o 3D koordinatės gaunamos iš gilio kadro ir pasinaudojus kalibravimo matrica.



4.11 pav. *BlazePose* su koordinatų korekcijos algoritmu 3D koordinatės

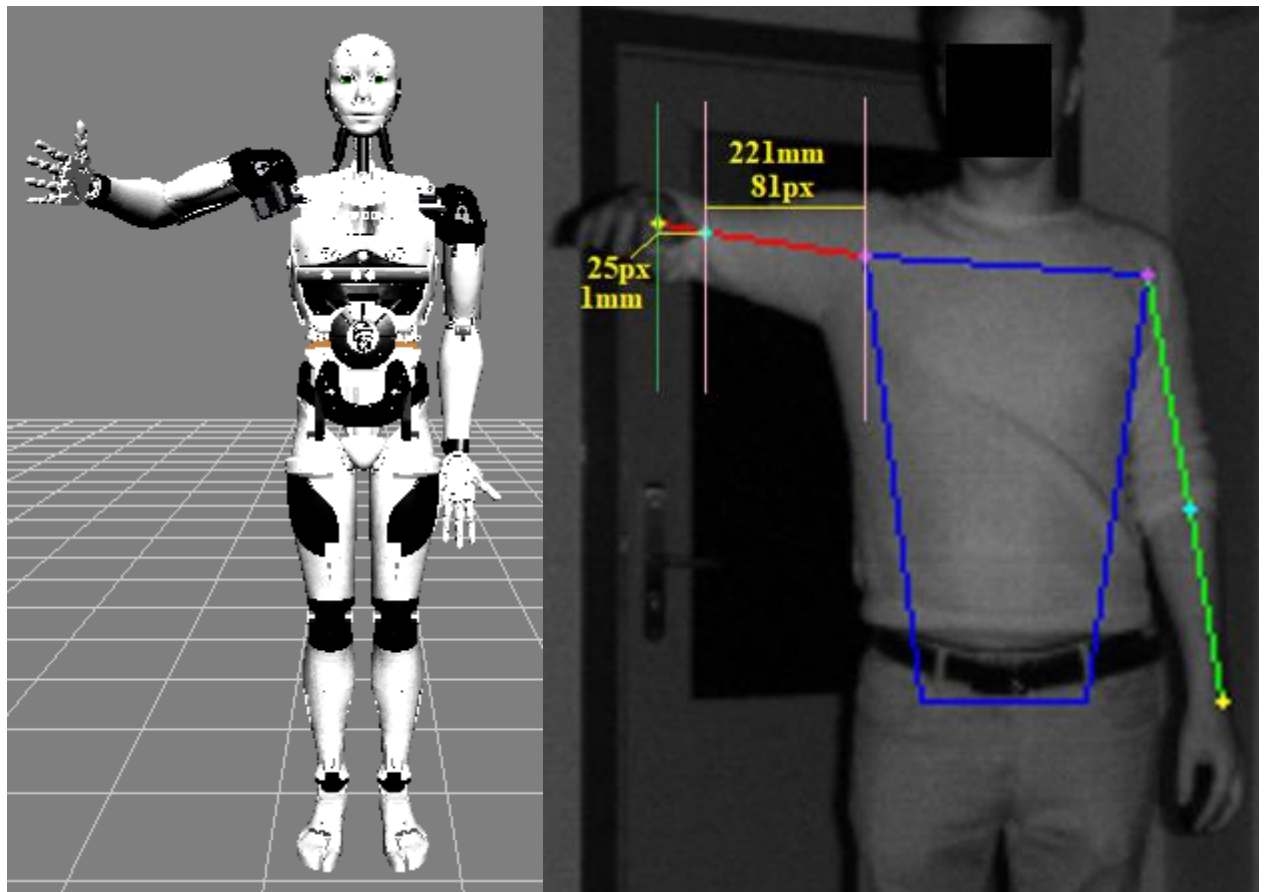
Simuliatoriui buvo pateikti apskaičiuoti kampai. 4.12 pav. pateikta roboto poza buvo priimta kaip teisinga, nes robotas negali atkartoti šios pozos tik dėl to, jog roboto ranka per alkūnę gali susilenkti tik  $90^\circ$  (apskaičiuota  $108^\circ$ ).





4.12 pav. *InMoov* roboto ir žmogaus pozicija

Skaičiuojant kampus šiuo metodu iškyla problema, kad kai kurių nors dviejų taškų x (pvz. alkūnė ir riešas) koordinatės yra labai artimos viena kitai, o z koordinatės labai skiriasi. Tokiu atveju atkarpos projekcija x ašyje tampa labai maža nors taip neturėtų būti. Paveikslėlyje žemiau parodytas vienas iš tokių atvejų:



4.13 pav. Taškų perkėlimo į 3D koordinatžių sistema problema

Kaip matyti iš 4.13 pav., apskaičiuotas atstumas, tarp peties ir alkūnės, kurie x ašyje yra apytiksliai lygiagrečioje padėtyje, o skirtumas tarp z koordinatės yra didelis, milimetrais yra 221 kartus didesnis nei tarp riešo ir alkūnės, nors pikseliais šis atstumas skiriasi tik 3.24 karto. Taip nutinka nes kalibravimo matrica yra skirta tam tikram atstumui, kuriam ji buvo gauta, todėl ji veikia geriau, kai elementai z ašyje yra lygiagretūs. Šiai problemai spręsti būtų galima sukurti adaptyvią kalibravimo matricą, kuri priklausomai nuo nuotolio galėtų keisti savo parametrus.

#### 4.5. Kampų judesio tyrimas

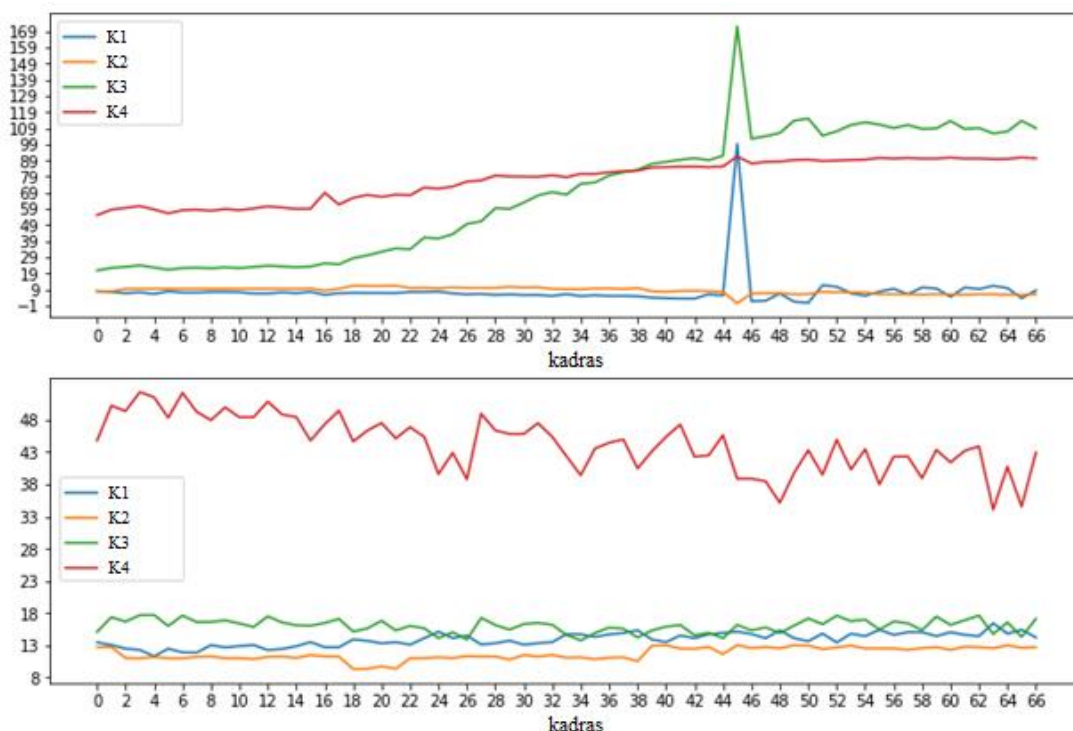
Šio tyrimo tikslas ištirti perskaičiuojamų kampų kitimo kokybę esant judesiams.

Judesio tikrinimo bandymui buvo įrašyti keturių skirtingų pagrindinių judesių vaizdo įrašai. Bandymuose atlikti veiksmai:

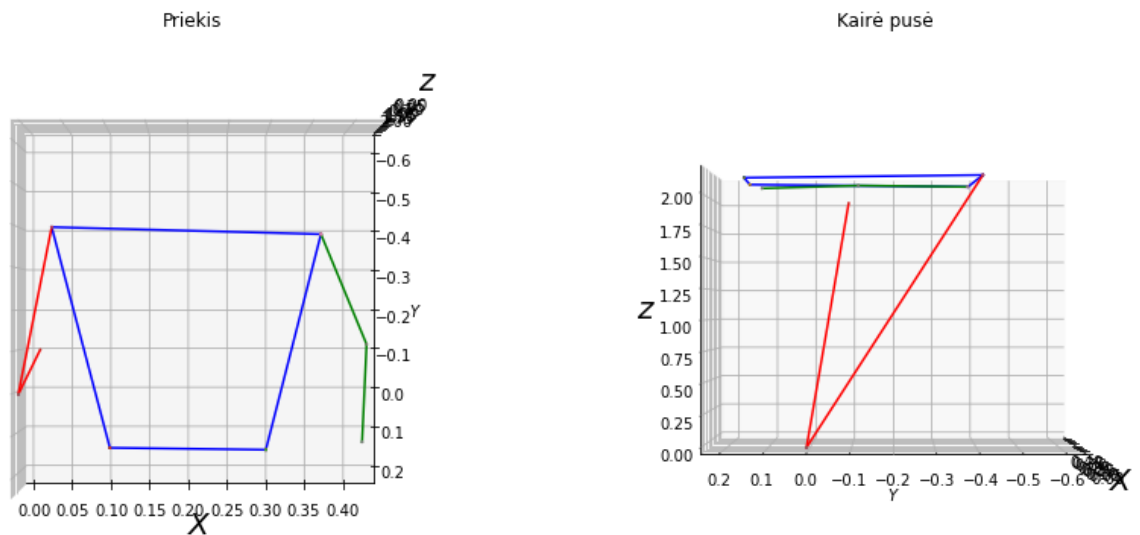
1. rankos sulenkimas per alkūnę iki  $\sim 90^\circ$  ( $K_3$ );
2. ištiestos rankos pakėlimas į šoną iki  $90^\circ$  ( $K_2$ );
3. ištiestos rankos kėlimas į priekį iki  $180^\circ$  ( $K_1$ );
4. per alkūnę sulenktos rankos ( $\sim 90^\circ$ ) sukimas nuo kūno į išorę ( $K_4$ ).

Visi įrašai buvo daryti tik dešinei rankai.

4.14 pav. pateiktas grafikas kuriame matyti per alkūnę kameros link lenkiamos dešinės rankos kitimo grafikas, o žemiau matomas kairės rankos grafikas. Iš gauto grafiko matyti, jog tik vieno kadro metu buvo gautas netinkamas kampas. To priežastis yra pikselio vertės korekcijos algoritmo nesuveikimas dėl per mažos užpildymo korekcijos pikselių imties spindulio (žr. 4.15 pav.).

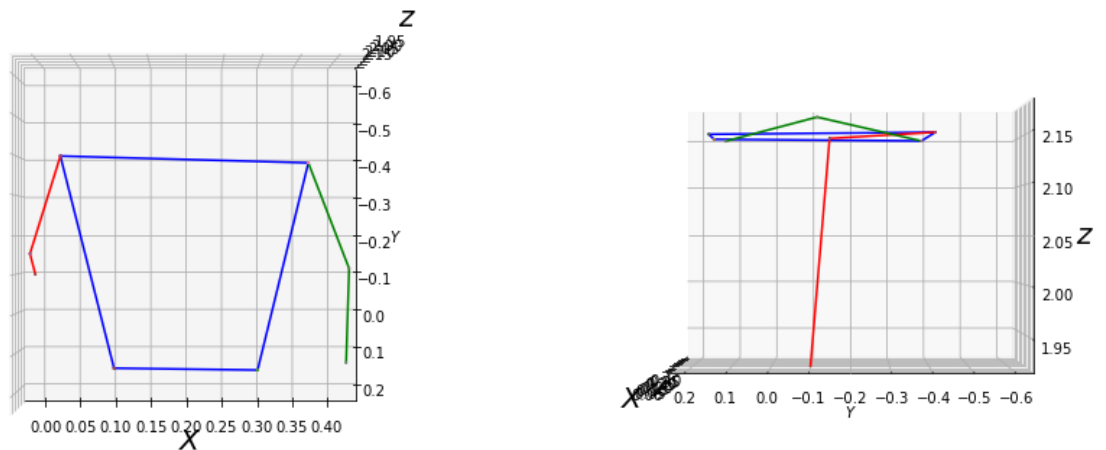
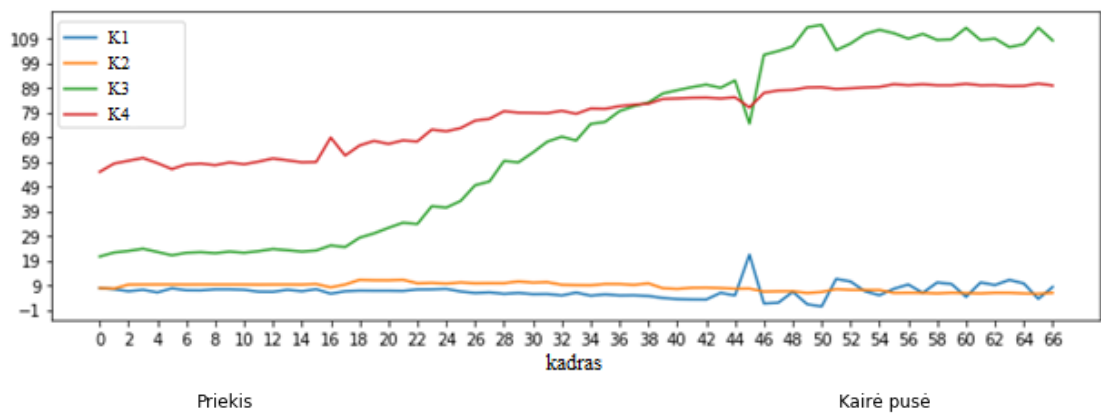


4.14 pav. Rankos lenkimo per alkūnę grafikas



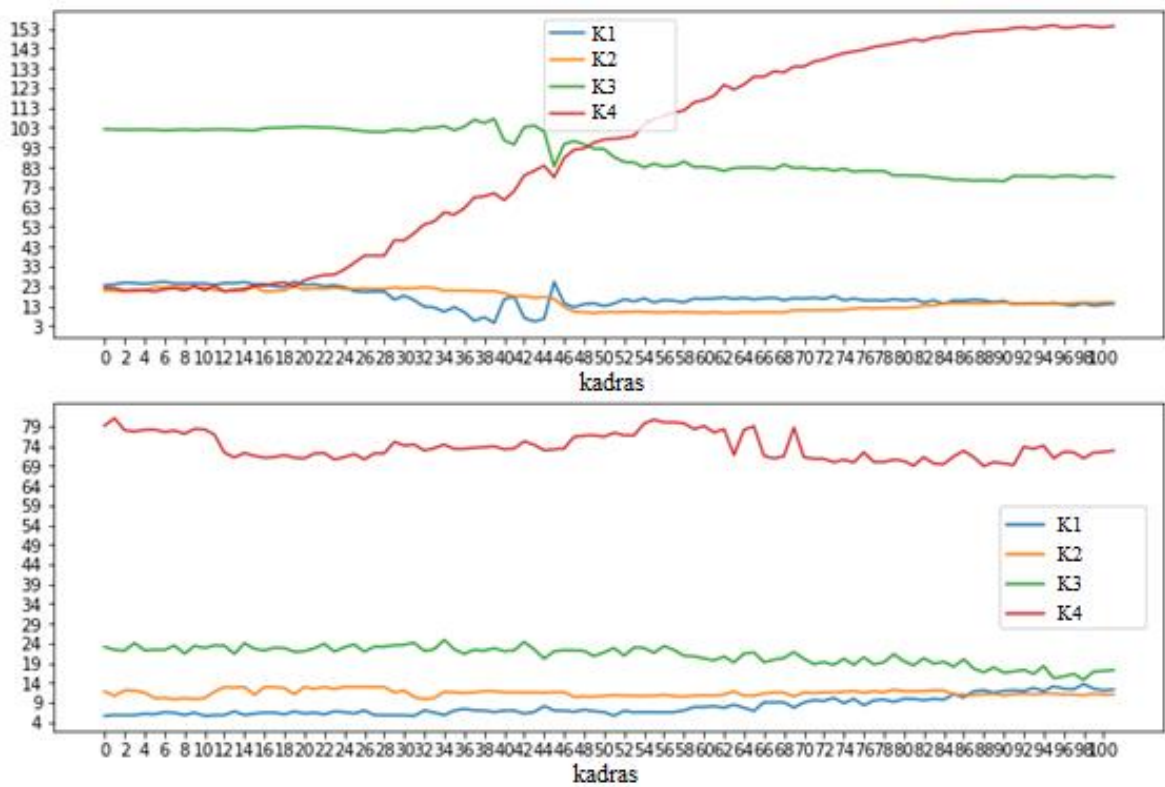
4.15 pav. alkūnės z ašies vertė = 0

Padidinus pikselio užpildymo algoritmo spindulio dydį į 5 pikselius nuo sąnario padėties taško, grafikas tampa tolygesnis (žr. 4.16 pav.). Kitiems judesio testams taip pat bus naudojamas penkių pikselių spindulys.



4.16 pav. rankos sulenkimo per alkūnę grafikas iš sąnarių pozicijos po užpildymo algoritmo imties padidėjimo

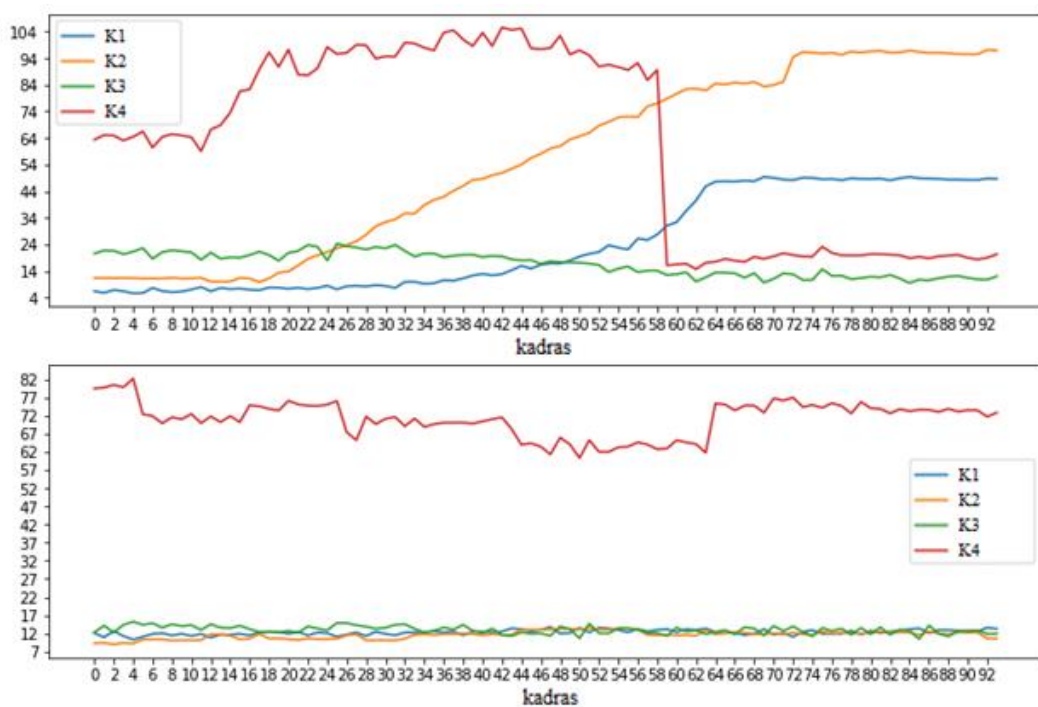
Rankos sukimo nuo kūno į išorę testo rezultatas pateiktas grafike 4.17 pav.:



4.17 pav. Rankos sukimo nuo kūno į išorę grafikas

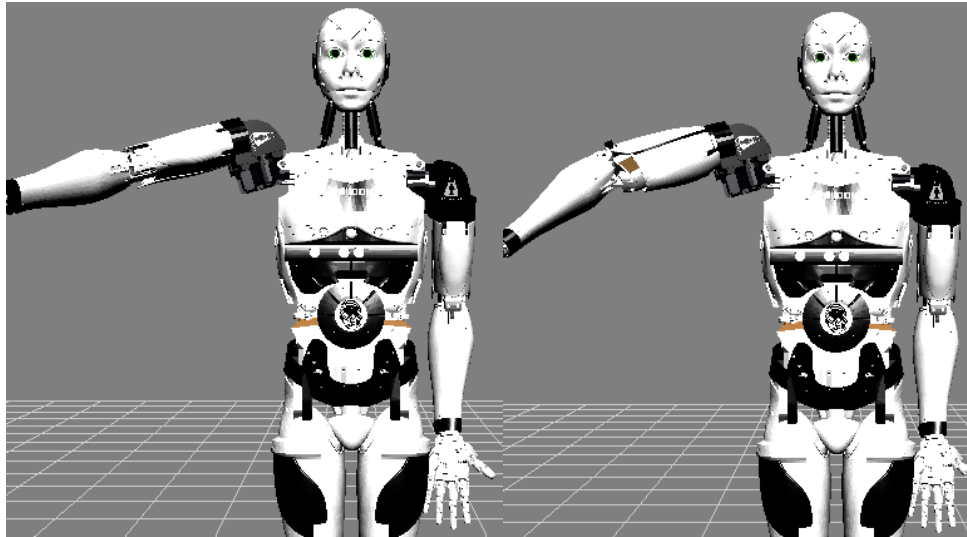
Grafike matyti, jog atliekamas judesys yra atpažįstamas gerai. Grafike matyti keli kampų pikai atskirais kadrais, bet roboto judėjimui tai neturėtų daryti didelės įtakos, nes servo pavaros negali taip greitai keisti kampo.

4.18 pav. pateiktas rankos kėlimo į šoną (kampas  $K_2$ ) testo rezultatas.



4.18 pav. Rankos kėlimo į šoną grafikas

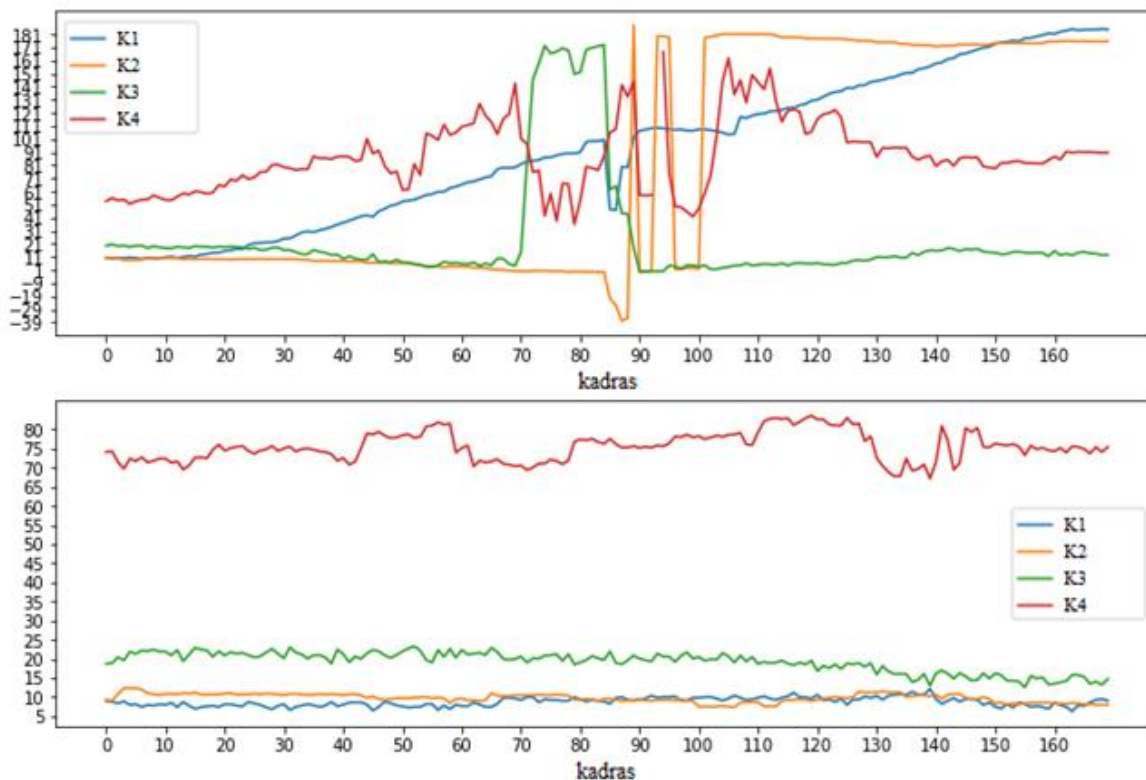
Iš grafiko matyti, jog keliant ranką į šoną, kampo  $K_2$  grafikas kinta tolygiai link  $80^\circ$  kampo. Keliant ranka į šoną ilginiui visiškai sunyksta atkarpos tarp peties ir alkūnės z ir y projekcijos, todėl skaičiuojant pirmąją formulę  $K_1$  kampas pradeda eksponentiškai kilti link  $90^\circ$  kampo. Tada yra naudojama (12) formulė, kuri nors ir nėra tiksli, bet apriboja šio kampo tolimesnį augimą. Geresnis rezultatas būtų gautas apribojant  $K_1$  kampą ties  $30^\circ$  konstanta, jei šios koordinatės sunyksta. Taip pat matyti didelis  $K_4$  kampo kitimas. Taip nutinka dėl to, nes ištiesus ranką į šoną plokštumos tarp kurių skaičiuojamas kampas sunyksta ir kampo skaičiavimas pagal pateiktą formulę tampa labai nestabilus. Didelis kampo sumažėjimas, įvyksta dėl formulės pasikeitimo iš (18) į (20). Tačiau tai nėra didelė problema, nes ištiesus rankos  $K_4$  kampas vizualiai nedaro didelės įtakos roboto rankos padėčiai. Tai matyti 4.19 pav.



4.19 pav. InMoov robotas  $K_4$  kampas 90 ir 40 (min vertė)

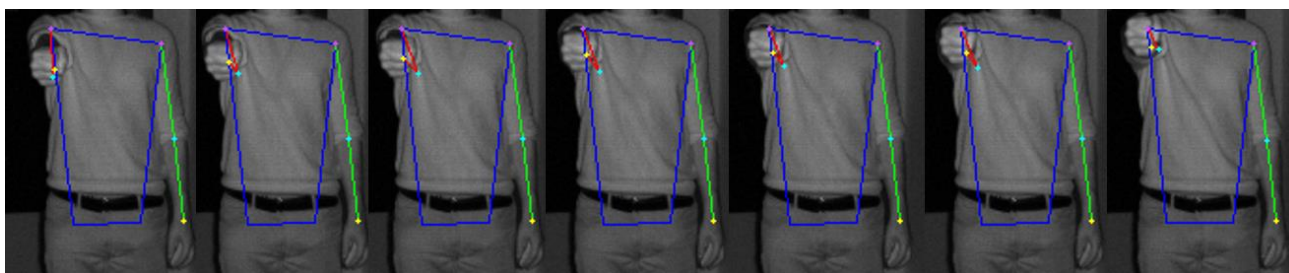
Siekiant, jog nebūtų tokio didelio šuolio, reikėtų pakeisti sąlygas kampo skaičiavimo formulės pakeitimui.

Ranką keliant kameros link rezultatai pateikti 4.20 pav. grafike.



4.20 pav. Kampų  $K_i$  keitimo kitimo grafikas

Tiriamasis kampas kinta tolygiai tik ties 85-90 kadrų yra šuolis žemyn. Bet taip pat matyti, jog kiti kampai yra labai nestabilūs. Tai nutinka dėl 2D taškų aptikimo modelio neteisingai nustatomų rankos koordinatų, tai galima pamatyti 4.21 pav.



4.21 pav. Judesio kadrai nuo 83 iki 89 kadro

Šią problemą išspręsti būtų galima pasinaudojus tikslesniu 2D koordinatų gavimo modeliu, arba sukūrus algoritmą, kuris koreguotų 2D sanarių koordinatų taškus pagal žmogaus kūno apribojimus.

## Išvados

1. Išbandyti metodai *MobileHumannPose* ir *ResidualPose* nėra tinkami nustatyti 2D žmogaus pozos koordinates, todėl jų teikiamos z koordinatės taip pat nėra tinkamos naudoti. Visi *BlazePose* modeliai davė gerus rezultatus nustatant 2D žmogaus pozos koordinačių taškus iš kurių visas tikrintas pozas geriausiai nustatė *BlazePose Full* modelis naudojant IR kadrus. Jo tikslumas siekė 100%.
2. *BlazePose Full* 3D koordinates nustatinėjo blogai. z koordinatė beveik visuose kadruose nustatė neteisingą z koordinatę, dėl ko rezultate buvo blogai apskaičiuojami kampai  $K_1$ - $K_4$ .
3. *ResidualPose* 3D taškų regresorius nėra tinkamas gautoms 3D koordinatėms paversti į žmogaus pozos 3D koordinates. Regresorius visiškai atsitiktinai perkelia taškus.
4. *BlazePose Full* su pateiktais korekcijos algoritmais nustatė 3D pozą teisingai 85% atvejų. Problemos kildavo tada, kai perkeliant taškus iš kameros koordinačių sistemos į realaus pasaulio koordinates, kai jungiamų sąnarių z koordinatė labai skyrėsi, o x koordinatės buvo labai artimos. Tai lėmė kalibravimo matrica, kuri skirtingiems atstumams turėtų būti kalibruojama atskirai.
5. Žmogaus dešinės rankos judesių kampų kreivės gautos naudojantis *BlazePose Full* su pateiktais korekcijos algoritmais puikiai susitvarkė su  $K_2$ ,  $K_3$  ir  $K_4$  kampų kitimu. Problemos kilo su kampe  $K_1$  tiesiant ranką kameros link, nes *BlazePose* modelis negalėjo tinkamai nustatyti 2D rankos koordinačių, dėl to kampų intervale nuo  $80^\circ$  iki  $101^\circ$  kampai buvo apskaičiuoti labai triukšmingai. Tuo tarpu kairės rankos kampai išliko gana stabilūs, išskyrus kampą  $K_4$ , nes jis priklauso ir nuo dešinio peties padėties.
6. Norint pasiekti tolygesnį kampų kitimą būtų galima naudoti filtrą, kuris panaikintų staigius kampų pokyčius, kai žmogaus sąnarių koordinatės yra nustatomos neteisingai. Taip pat, galima surasti arba sukurti geresnį 2D taškų aptikimo modelį.

## Literatūros sąrašas

1. WALLÉN, Johanna. The history of the industrial robot. Linköping University Electronic Press, 2008.
2. HOCKSTEIN, N. G., GOURIN, C. G., FAUST, R. A. ir TERRIS, D. J. A history of robots: from science fiction to surgical robotics. iš Journal of robotic surgery [interaktyvus]. 2007, 1(2), pp. 113-118 [žiūrėta 2021-10-03]. Prieiga per: <https://doi.org/10.1007/s11701-007-0021-2>
3. GIPS, James. Toward the ethical robot. 1994.
4. NOF, Shimon Y. (ed.). Handbook of industrial robotics. John Wiley & Sons, 1999. ISBN 0471177830
5. SICILIANO, Bruno; KHATIB, Oussama; KRÖGER, Torsten (ed.). Springer handbook of robotics. Berlynas: springer, 2008. ISBN: 9783540303015
6. LASCHI, Cecilia, et al. Grasping and manipulation in humanoid robotics. Scuola Superiore Sant Anna, Italija, 2000.
7. HIRAI, Kazuo, et al. The development of Honda humanoid robot. In: Proceedings. 1998 IEEE international conference on robotics and automation. IEEE [interaktyvus]. 1998. pp. 1321-1326 [žiūrėta 2021-10-05]. ISSN 1050-4729. Prieiga per: doi: 10.1109/ROBOT.1998.677288
8. NENCHEV, Dragomir N., KONNO, Atsushi; TSUJITA, Teppei. Humanoid robots: Modeling and control. Butterworth-Heinemann, 2018.
9. DAEHL, Dani. Honda retires its famed Asimo robot [interaktyvus]. 2018 [žiūrėta 2021-10-05]. Prieiga per: <https://www.theverge.com/2018/6/28/17514134/honda-asimo-humanoid-robot-retire>
10. PANDEY, Amit Kumar; GELIN, Rodolphe. A mass-produced sociable humanoid robot: Pepper: The first machine of its kind. IEEE Robotics & Automation Magazine [interaktyvus]. 2018, 25(3): psl. 40-48 [žiūrėta 2022-10-05]. Prieiga per: doi: 10.1109/MRA.2018.2833157
11. LEHTO, Paula. Robots with and for the Elderly People: Case Study Based on Action Research. ICERI2017 Proceedings [interaktyvus]. 2017 pp. 381-387 [žiūrėta 2021-10-05]. Prieiga per: [doi:10.21125/iceri.2017.0153](https://doi.org/10.21125/iceri.2017.0153)
12. TANAKA, Fumihide, et al. Pepper learns together with children: Development of an educational application. 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids) [interaktyvus]. 2015, pp. 270-275 [žiūrėta 2022-10-05]. ISBN 9781479968855. Prieiga per: doi: 10.1109/HUMANOIDS.2015.7363546.
13. DE GAUQUIER, Laurens, et al. Humanoid robot pepper at a Belgian chocolate shop. In: Companion of the 2018 ACM/IEEE international conference on human-robot interaction. 2018. pp. 373-373. Prieiga per: doi : 10.1145/3173386.3177535
14. "Pepper" to be placed at Ueno Station on a trial basis! [Interaktyvus]. 2016 [žiūrėta 2021-10-05]. Prieiga per : [https://www.tokyometro.jp/lang\\_en/news/157651.html](https://www.tokyometro.jp/lang_en/news/157651.html)
15. WARWICK, Kevin; SHAH, Huma. Can machines think? A report on Turing test experiments at the Royal Society. Journal of experimental & Theoretical artificial Intelligence. 2016 [žiūrėta 2021-10-05]. 28(6): 989-1007. Prieiga per: <https://doi.org/10.1080/0952813X.2015.1055826>
16. RETTO, Jesús. Sophia, first citizen robot of the world. ResearchGate [interaktyvus]. 2017 [žiūrėta 2021-10-05]. Prieiga per: <https://www.Researchgate.net>
17. HART, Matthew. INCREDIBLY HUMANLIKE ROBOT GETS ANGRY WHEN SOMEONE BOOPS ITS NOSE [Interaktyvus]. 2022 [žiūrėta 2022-04-27]. Prieiga per <https://nerdist.com/article/ameca-humanlike-robot-engineered-arts-angry-video/>



18. BOGUE, Robert. The role of robots in entertainment. *Industrial Robot: the international journal of robotics research and application* [interaktyvus]. 2022 [žiūrėta 2022-04-27]. ISSN: 0143-991X. Prieiga per: <https://doi.org/10.1108/IR-02-2022-0054>
19. Arduengo Miguel, Arduengo Ana, Colome Adria, Lobo-Prat Joan, Torras Carme. Human to Robot Whole-Body Motion Transfer. 2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids) [interaktyvus]. 2021, 299-305 [žiūrėta 2021-09-15]. ISSN: 1558-223X. Prieiga per: doi: 10.1109/HUMANOIDS47582.2021.9555769.
20. Sripada, Aditya, Asokan, Harish, Warriar, Abhishek, Kapoor, Arpit, Gaur, Harshit, Patel, Rahil, R, Sridhar. Teleoperation of a Humanoid Robot with Motion Imitation and Legged Locomotion. 2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM). [interaktyvus]. 2018, pp. 375-379 [žiūrėta 2021-09-15]. Prieiga per doi: 10.1109/ICARM.2018.8610719.
21. Syakir Muhammad, Ningrum Endah Suryawati, Adji Sulistijono Indra. Teleoperation Robot Arm using Depth Sensor. 2019 International Electronics Symposium (IES). [interaktyvus] 2019. 394-399 [žiūrėta 2021-09-16], ISSN: 0143-991x. Prieiga per: doi: 10.1109/ELECSYM.2019.8901679.
22. Wu Xin, Yang Canjun, Zhu Yuanchao, Wu Weitao, Wei Qianxiao. An integrated vision-based system for efficient robot arm teleoperation. *Industrial Robot: the international journal of robotics research and application*. ahead-of-print [interaktyvus]. 2020, [žiūrėta 2021-09-16] Prieiga per: doi: 10.1108/IR-06-2020-0129.
23. Balmik, A., Jha, M., Nandy, A. NAO Robot Teleoperation with Human Motion Recognition. *Arab J Sci Eng* 47 [interaktyvus]. 1137–1146 (2021) [žiūrėta 2021-09-16]. Prieiga per: <https://doi.org/10.1007/s13369-021-06051-2>
24. Liu Yizhi, Habibnezhad Mahmoud, Jebelli Houtan. Brain-computer interface for hands-free teleoperation of construction robots. *Automation in Construction* [interaktyvus]. 2021. [žiūrėta 2021-09-18] ISSN 0926-5805. Prieiga per: <https://doi.org/10.1016/j.autcon.2020.103523>.
25. *InMoov* Open source 3D printed life-size Robot [interaktyvus]. Prieiga per: <https://inmoov.fr>
26. LifeCam HD-3000 [interaktyvus]. [žiūrėta 2020-03-10]. Prieiga per: <https://www.microsoft.com/en-ww/accessories/products/webcams/lifecam-hd-3000?activetab=overview%3aprimar2>
27. LI, Larry. Time-of-flight camera—an introduction. Techninė dokumentacija, 2014, SLOA190B.
28. JIAO, Jichao, et al. A post-rectification approach of depth images of Kinect v2 for 3D reconstruction of indoor scenes. *ISPRS International Journal of Geo-Information* [interaktyvus]. 2017, 6(11): 349. Prieiga per: <https://doi.org/10.3390/ijgi6110349>
29. Set up Kinect for Windows v2 with a Kinect Adapter for Windows 10 PC [interaktyvus]. [žiūrėta 2021-09-20]. Prieiga per: <https://support.xbox.com/en-US/help/hardware-network/kinect/kinect-for-windows-v2-setup-with-adapter>
30. *MyRobotLab* dokumentacija [interaktyvus]. [žiūrėta 2021-11-23]. Prieiga per: <http://myrobotlab.org/documentation>
31. The Jupyter Notebook [interaktyvus]. [žiūrėta 2021-11-23] Prieiga per: <https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>
32. Using the Single Camera Calibrator App [interaktyvus]. [žiūrėta 2021-11-23]. Prieiga per <https://www.mathworks.com/help/vision/ug/using-the-single-camera-calibrator-app.html>.
33. HE, Kaiming, et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*[interaktyvus]. 2016. pp. 770-778 [žiūrėta 2021-12-18]. Prieiga per:

- [https://openaccess.thecvf.com/content\\_cvpr\\_2016/html/He\\_Deep\\_Residual\\_Learning\\_CVPR\\_2016\\_paper.html](https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html)
34. IONESCU, Catalin, et al. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence* [interaktyvus]. 2014, 36(7): pp. 1325-1339 [žiūrėta 2021-12-18]. Prieiga per : doi: 10.1109/TPAMI.2013.248
  35. SANDLER, Mark, et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* [interaktyvus]. 2018. pp. 4510-4520 [žiūrėta 2021-12-28]. Prieiga per [https://openaccess.thecvf.com/content/CVPR2018W/MAI/html/Choi\\_MobileHumanPose\\_Toward\\_Real-Time\\_3D\\_Human\\_Pose\\_Estimation\\_in\\_Mobile\\_Devices\\_CVPRW\\_2018\\_paper.html](https://openaccess.thecvf.com/content/CVPR2018W/MAI/html/Choi_MobileHumanPose_Toward_Real-Time_3D_Human_Pose_Estimation_in_Mobile_Devices_CVPRW_2018_paper.html)
  36. CHOI, Sangbum; CHOI, Seokeon; KIM, Changick. MobileHumanPose: Toward real-time 3D human pose estimation in mobile devices. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* [interaktyvus]. 2021. pp. 2328-2338 [žiūrėta 2021-12-18]. Prieiga per: [https://openaccess.thecvf.com/content/CVPR2021W/MAI/html/Choi\\_MobileHumanPose\\_Toward\\_Real-Time\\_3D\\_Human\\_Pose\\_Estimation\\_in\\_Mobile\\_Devices\\_CVPRW\\_2021\\_paper.html](https://openaccess.thecvf.com/content/CVPR2021W/MAI/html/Choi_MobileHumanPose_Toward_Real-Time_3D_Human_Pose_Estimation_in_Mobile_Devices_CVPRW_2021_paper.html)
  37. Github Code of "MobileHumanPose: Toward real-time 3D human pose estimation in mobile devices" [interaktyvus]. [žiūrėta 2022-01-30]. Prieiga per: <https://github.com/SangbumChoi/MobileHumanPose>
  38. BAZAREVSKY, Valentin, et al. Blazepose: On-device real-time body pose tracking [interaktyvus]. *arXiv preprint arXiv:2006.10204*, 2020. 2338 [žiūrėta 2022-02-03] Prieiga per: <https://doi.org/10.48550/arXiv.2006.10204>
  39. MARTÍNEZ-GONZÁLEZ, Angel, et al. Residual pose: A decoupled approach for depth-based 3D human pose estimation. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* [interaktyvus]. IEEE, 2020. pp. 10313-10318 [žiūrėta 2022-02-22].
  40. JIAO, Jichao, et al. A post-rectification approach of depth images of Kinect v2 for 3D reconstruction of indoor scenes. *ISPRS International Journal of Geo-Information* [interaktyvus]. 2017, (6)11: 349.[žiūrėta 2022-02-22]. Prieiga per: <https://doi.org/10.3390/ijgi6110349>
  41. 3D Coordinate Geometry - Equation of a Line [interaktyvus]. [žiūrėta 2022-02-26] Prieiga per: <https://brilliant.org/wiki/3d-coordinate-geometry-equation-of-a-line/>
  42. ZHANG, Zhijun, NIU, Yaru, KONG, Lingdong, LIN, Shuyang and WANG, Hao. A real-time upper-body robot imitation system. *International Journal of Robotics and Control* [interaktyvus]. 2019[žiūrėta 2022-02-27], 2(1): pp. 49-56. Prieiga per: DOI 10.5430/ijrc.v2n1p49
  43. ASSAD UZ ZAMAN, Md, et al. Robot sensor system for supervised rehabilitation with real-time feedback. *Multimedia Tools and Applications* [interaktyvus]. 2020, 79(35), p.p. 26643-26660.[žiūrėta 2022-04-10] Prieiga per: doi: 10.1007/s11042-020-09266-x