SPARC
Servicii Performante de Asistență a Clienților prin Platforme Robotice

cod: PN-III-P2-2.1-BG-2016-0425

Grant in cadrul Programului Național PNIII - Programul 2 - Creşterea competitivităţii economiei româneşti prin cercetare, dezvoltare şi inovare, Transfer de cunoaștere la agentul economic „Bridge Grant”

Etapa 2

Dezvoltarea platformei software SPARC și integrarea cu robotul Tiago / Pepper

Raport științific și tehnic

Obiectiv proiect

Principalul obiectiv al proiectului este proiectarea și implementarea unei platforme pentru realizarea integrată a planificării sarcinilor roboților de interes în proiect, comanda execuției planurilor definite și monitorizarea acțiunilor acestora. Platforma trebuie să permite personalizarea facilă a programului robotului dar și crearea unui comportament adaptiv la anumite condiții neprevăzute.

Agentul economic care face obiectul actualei propuneri de proiect, Centrul IT pentru Știință și Tehnologie SRL (CITST) este un IMM orientat spre cercetare și dezvoltare de produse inovatoare. Tehnologia disponibilă la CITST pe care se bazează proiectul este robotul TIAGO. CITST vrea să exploateze mai bine tehnologia robotică disponibilă în companie oferind-o ca serviciu (închiriere, leasing) pentru potențialii clienți. Scenariile de utilizare avute în vedere se referă la promovare de produse și/sau asistență în locuri publice: centre comerciale, muzee, bănci, expoziții. CITST are în vedere și achiziționarea unui robot Pepper în același scop. Tehnologia disponibila la UPB este reprezentată de roboții Baxter de la Rethink Robotics și Pepper de la Aldebaran Robotics.

Colectivul proiectului este format din membrii grupului de cercetare AIMAS de la Facultatea de Automatică și Calculatoare din UPB și de studenți din cadrul programului de masterat Artificial Intelligence, tot de la Facultatea de Automatică și Calculatoare.

Cuprins

1. Introducere

Ideea de roboți de asistenta care interacționează și sprijină oamenii în viața de zi cu zi devine tot mai atractivă, atât în cercurile de cercetare, cât și în cele din industrie. Domeniile comune de asistență includ salutul, ghidarea și informarea clienților în locurile comerciale și locurile de expunere, descrierea unor produse sau manipularea în cabinele de prezentare, asistența vizitatorilor în muzee etc.

Există deja furnizori care oferă soluții robotice ale căror capabilități le pot face potrivite pentru domeniile de aplicare enumerate mai sus. Cu toate acestea, în timp ce există o mare varietate de platforme robotizate, este în prezent încă o provocare adaptarea comportamentului unui robot de asistență de la un scenariu la altul, în funcție de cerințele clientului.

În consecință, proiectul SPARC (Servicii Performante de Asistenta a Clientilor prin Platforme Robotice) are ca scop proiectarea și implementarea unei platforme care să permită o flexibilitate sporită și o ușurință în definirea comportamentelor specifice clienților pentru roboți de asistenta.

Abordarea se bazează pe conceptul de programare la nivel de obiectiv. Această abordare ajută dezvoltatorul să determine cu ușurință acțiunile unui robot folosind compoziția comportamentelor de bază, de exemplu, detectarea unui utilizator, răspunsul la interogarea utilizatorului, urmărirea utilizatorului, mutarea în locație, detectarea obiectului, indicarea unui obiect.

Se identifica, astfel, doua provocări ale acestei încercări, prin:

- Definirea si implementarea unui set de comportamente de bază autonome, din care pot fi ușor exprimate scenarii pentru domeniile de aplicare menționate. Aceste comportamente de bază trebuie să îndeplinească următoarele condiții:
    - Siguranța în interacțiunea cu utilizatorii umani;
    - Operație corectă și solidă (de ex. Toleranță la schimbări neașteptate în mediu, grad scăzut de eroare, manipulare a erorilor)

- Definirea și implementarea unui cadru dinamic de planificare și execuție bazat pe obiective capabile să combine comportamentele de bază într-un plan de acțiune consecvent pentru realizarea obiectivul dorit de utilizator. Cadrul trebuie să poată răspunde, în mod continuu, evenimentelor din interacțiunea cu utilizatorul și schimbărilor în mediu. Acestea vor acționa ca declanșatoare ale unei proceduri de re-planificare, care va actualiza obiectivele / sub-obiectivele robotului în consecință.

2. Etapa de Analiză

Prima parte a proiectului a constat în colectarea de informații despre roboții Tiago și Pepper și despre modul în care putem implementa caracteristicile dorite. În urma analizei am hotarât să orientăm eforturie către robotul Pepper și să dezvoltăm un set de module și comportamente de bază proprii care să poată fi portat și pe alte tipuri de roboți.

Am început sa analizam componenta hardware a robotului Pepper in ceea ce privește senzorii si camera. Am testat în situații diferite, am verificat dacă se comportă corespunzător, evitând obstacolele, navigând prin mediu. Am testat modulele de bază de la NAOqi API folosind Choregraphe. Din punct de vedere al navigației, am descoperit că senzorii nu sunt atât de preciși. Dacă luăm în considerare poziția de pornire a robotului ca (0, 0, 0) și o punem înainte și înapoi cu un metru pe axa x, noua poziție va fi diferită de (0, 0, 0). Această diferență este cauzată de odometrie, rotația fiind cea care rupe calculul.

Camerele robotului Pepper dar și lui Tiago au fost sub așteptări. S-au depus eforturi, la început, pentru a lua fluxul de imagini, deoarece informațiile furnizate de camere nu au fost aranjate așa cum s-a crezut inițial. Frecventa cadrelor a fost, de asemenea, un obstacol pentru proiectul nostru, deoarece este invers proporționala cu rezoluția imaginilor și aceasta a fost o problemă pentru tehnicile de tip computer vision.

S-a luat in calcul utilizarea ROS (Sistem de operare al roboților) ca o platformă universală pentru proiectul nostru. Cu toate acestea, a fost foarte dificil integrarea unor baze de date disponibile cu ajutorul robotului Pepper, așa că am decis să continuăm proiectul folosind NAOqi API, folosind atât Python, cât și Choregraphe pentru a programa comportamentele noastre.

3. Scenariu

Înainte de a trece în revistă setul de comportamente dezvoltate și modul în care gestionăm interacțiunea lor, introducem un scenariu de utilizare a robotul Pepper folosind comportamentele dezvoltate. Pe parcursul conturării scenariului vor deveni evidente tipurile de comportamente de bază necesare implementării acestuia. Modul de funcționare individual al fiecărui comportament este detaliat in secțiunea următoare.

Prima parte a scenariului este compusă dintr-o etapă pregătitoare. Această etapă este necesară datorită faptului ca robotul trebuie sa cunoască în prealabil locația în care va fi plasat, precum și personalul ce poate asista în desfășurarea evenimentului.

Pentru a putea realiza configurarea inițială, în dezvoltări ulterioare ce vor face obiectul etapei 3, ce va fi definită împreună cu operatorul economic în scenariul demonstrativ de exploatare, robotul va include o formă de realizare a hărții ce descrie locul evenimentului sau folosind metode externe o hartă va fi încărcată pe robot.

De asemenea, în interfața de configurare va exista o metodă pentru a introduce poziții ale exponatelor, unde robotul va naviga pentru prezentarea lor.

Fiecare dintre obiectele configurate va conține un identificator pe care Pepper îl va căuta. De asemenea în etapa pregătitoare, utilizatorii aplicației robotice pot introduce informații generale legate de eveniment și edita modul în care Pepper va întâmpina participanții.

Pentru a putea recunoaște personalul ce va participa la realizarea evenimentului o metodă de învățare a fețelor este pusă la dispoziția utilizatorilor. Aceștia inițiază memorarea printr-o comandă vocală și menționează datele de identificare.

Odată început evenimentul, robotul așteaptă în zona de intrare participanții la eveniment. În momentul în care în proximitatea lui apar persoane neidentificate, acesta îi

întâmpina și le transmite vocal informațiile generale legate de eveniment, apoi robotul va putea oferi informații suplimentare utilizatorilor la cerere în următoarele doua forme.

În momentul în care un participant îi cere robotului sa îi arate unul din exponatele predefinite în etapa pregătitoare, robotul îi cere utilizatorului sa îl urmeze și îl conduce în zona în care a fost configurată locația, începând căutarea identificatorului pentru prezentarea obiectului. După găsirea acestuia, Pepper prezintă informațiile referitoare la acesta, mulțumește pentru atenție și revine în poziția inițială.

În cea de-a doua formă, utilizatorul poate cere robotului să îl urmeze pentru a îi putea cere informații. Robotul identifică și memorează utilizatorul și începe să îl urmeze. În momentul în care robotul a ajuns în locația dorită, utilizatorul îl va informa. Acesta începe căutarea exponatul cel mai apropiat și oferă informațiile pre-configurate.

Acest comportament este de asemenea folosit în cazul în care utilizatorul îi cere robotului să facă o fotografie. Utilizatorul poate cere robotului să îl urmeze la locul în care dorește să realizeze fotografia, robotul realizează fotografia și o prezintă utilizatorului prin intermediul tabletei incorporate până când acesta este mulțumit de rezultat.

Dacă în oricare dintre comportamente descrise mai sus, utilizatorul cere asistență de la o persoană responsabilă de eveniment, robotul va notifica personalul responsabil. Daca nici o persoană nu răspunde la notificare, robotul va căuta și recunoaște persoana dorită pe baza memorării anterioare în etapa de pregătitoare. Robotul caută o perioadă predefinită și dacă o găsește, o va ruga să îl urmeze. În momentul în care vor ajunge la participantul la eveniment, Pepper va introduce persoana responsabilă de eveniment.

După terminarea scenariilor descrise, robotul va reveni la o poziția din locația inițială, revenind în etapa de întâmpinare a participanților. Pe tot parcursul etapei de întâmpinare, robotul va memora persoanele ce au intrat la eveniment și au fost deja întimpinate pentru a nu repeta introducerea.

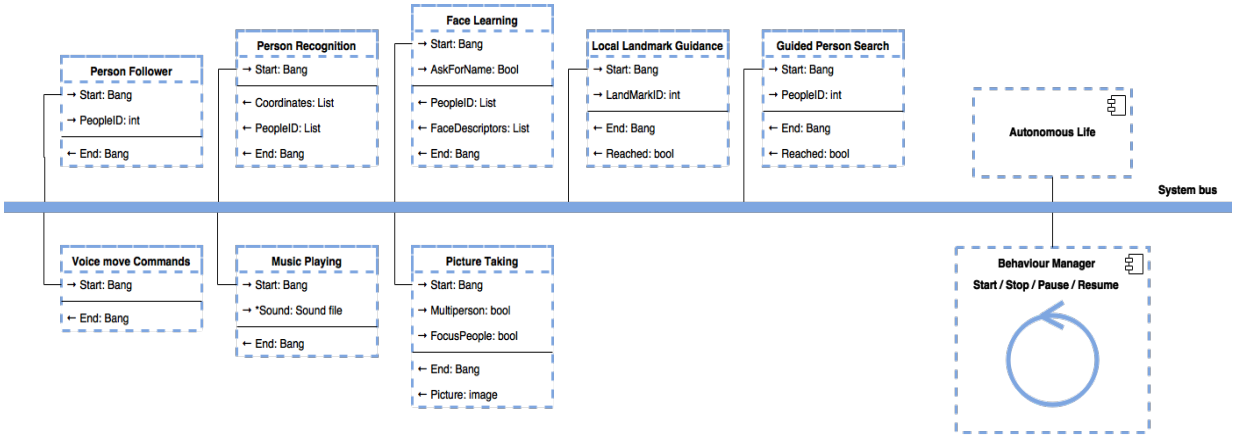
Diagrama din Figura 1 prezintă fluxul de interacțiune al comportamentelor ce compun scenariul prezentat anterior.

Figură 1. Diagrama de interacțiune si de flux al comportamentelor pentru scenariu de asistență robotică descris

4. Arhitectura platformei de comportamente

Unul din obiectivele proiectului SPARC este definirea unei platforme de gestiune pentru un set de comportamente “de bază”. Aceasta platformă trebuie să permită definirea de obiective si a secvenței de execuție (înlănțuire) pentru comportamentele de bază necesare realizării fiecărui obiectiv în parte.

În plus, platforma trebuie să fie capabilă să gestioneze ciclul de viață al fiecărui comportament de bază. Pentru fiecare comportament, componenta de control/planificare a platformei trebuie să mențină o stare a comportamentului (activ, inactiv, întrerupt, eșuat) și să poată acționa în sensul modificării stării (e.g. a declanșa un comportament, a întrerupe un comportament).



Figură 2. Diagrama bloc prezentând arhitectura platformei de gestiune a comportamentelor de bază în proiectul SPARC

Diagrama din Figura 2 exemplifică modulele de comportament mediu spre complex dezvoltate în sistem. Fiecare din comportamentele trecute in diagrama sunt detaliate în sectiunile următoare ale acestui raport. Comportamentele sunt implementate sub forma unor module independete de tipul “Box” ce pot fi gestionate, apelate direct sau pot fi integrate în cadrul altor module, permițând dezvoltarea aplicațiilor din ce în ce mai complexe bazate pe o structură ierarhică.

Sistemul este gestionat la nivel superior de către cele două module “Autonomous Life” și “Behaviour Manager”. Acestea au responsabilitatea de a porni, opri, a pune pauză sau a reporni diversele comportamente implementate pentru a oferi desfășurarea scenariilor oferite de proiect.

Comunicarea lor se realizează prin intermediul memoriei cu ajutorul unor cozi de evenimente, conform unor protocoale descrise. Fiecare modul dezvoltat prezintă o interfață pentru inputurile necesare pentru activare și outputurile pe care acestea le produc pe parcursul desfășurării și la finalizarea comportamentului. În mod obligatoriu fiecare modul va necesita un semnal binar de pornire (“Start”) și va declanșa un semnal binar de terminare la sfârșit (“End”). Pe lângă acestea alte semnale pot fi descrise ca obligatorii sau opționale.

Programarea unui scenariu are loc în cadrul modulului *Behavior Manager.* Acesta pune la dispoziție structuri de date ce reprezintă unități de execuție.

Fiecare unitate de execuție dispune de metode prin care pot fi verificate condițiile necesare declanșării, opririi sau pauzei unui comportament de bază, cât și metode utile în recuperarea rezultatelor execuției unui comportament de bază (preluat din semnalele asignate comportamentului, altele decât cele de *start* si *stop*).

Mai departe, modulul *Behavior Manager* dispune de metode prin care se poate preciza desfășurarea *în secvență* sau *în paralel* a mai multor unități de execuție.

La fiecare moment de timp, una sau mai multe unități de execuție pot fi active. Modulul *Behavior Manager* implementează un sistem de control de tip *round-robin* pentru a preda succesiv dreptul de rulare/decizie fiecarei unități de execuție active. Mecanismul de multi-tasking la nivelul platformei SPARC este așadar unul de natură colaborativă, bazându-se pe faptul ca o unitate de execuție va permite să fie întreruptă de evenimente de natură să influențeze alte unități de execuție aflate *în așteptare*.

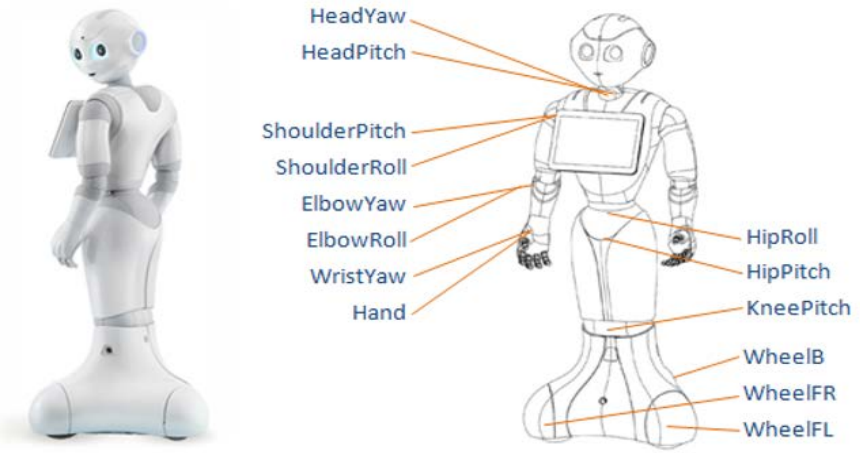
Considerând această arhitectură, bazată pe manageri, o bună interacțiune a modulelor poate produce un comportament inteligent de interacțiune între robot și oameni.

5. Platforma Robotică

Performanțele aplicațiilor noastre sunt direct influențate de platforma pe care sunt utilizate. Limitările platformei robotizate sunt atât la nivel hardware, cât și la nivel de software și, din acest motiv, le vom prezenta având în vedere platforma robotică Pepper.

5.1. Robotul Pepper

Robotul pe care am dezvoltat se numeste Pepper, un robot conceput de compania SoftBank Robotics. Este prima mașină robotizată special concepută pentru a interacționa cu oamenii cu un aspect uman foarte prietenos.



Figură 3. Aspect fizic al robotului Pepper robot, impreuna cu imbinarile mobile

Robotul Pepper are o înălțime de 1,2 metri și are două motoare instalate la nivelul gâtului, care permit mișcări sus-jos și stânga-dreapta, conform figurii 1. Prin combinarea celor două motoare, Pepper își poate muta capul într-o varietate de poziții, foarte asemănător cu mișcările capului uman.

Robotul are trei camere instalate la nivelul capului, două dintre ele oferind imagini 2D și una oferind informații 3D. Camerele 2D sunt montate pe frunte și oferă imagini la o rezoluție de 640x480 pixeli care primesc 30 de cadre pe secundă sau 2560x1920 cu un cadru pe secundă. Senzorul 3D este situat în spatele ochilor și oferă imagini 3D (spațiu cu adâncime de culoare) cu o rezoluție de 320x240 pixeli, primind 20 de cadre pe secundă.

Având în vedere interacțiunea cu mediul, Pepper are mai mulți senzori, care îi ajută să se miște și să evite obstacolele:

● Unitate inerțială - realizată dintr-un girometru cu 3 axe cu o viteză unghiulară de ~ 500^0/s și un accelerometru cu 3 axe cu o accelerație de ~ 2g;

● Laseri - care includ senzori laser, actuatori laser pentru evaluarea terenului din față, actuatori laser pentru împrejurimi;
Sonare - doi senzori ultrasonori (sau sonari) utilizați pentru estimarea distanței pana la obstacole în mediul robotului. Un sonar este compus din 2 actuatoare și 2 senzori.
● Infra-roșu - 2 senzori infraroșii.
● MRE - 30 x MRE (Magnetic Rotary Encoders) utilizând tehnologia senzorilor cu efect Hall. Precizia sa este de 14 biți, filtrate la 12 biți.

5.2. Sistemul de Operare Robotic

Robotul este compatibil cu două sisteme de operare diferite pentru implementarea noilor funcționalități: Sistemul de operare al roboților (ROS) și NAOqi. NAOqi este sistemul software propriu, care funcționează doar pe robotii creați de SoftBank Robotics și Aldebaran, în timp ce ROS este un sistem de operare general.

ROS este un sistem de operare compus din mai multe module open source, create pentru a ușura dezvoltarea software-ului pentru roboți. Acesta a fost creat prin contribuția mai multor laboratoare robotizate pentru a avea un mediu flexibil și solid pentru o varietate de platforme robotizate. ROS permite comunicarea între procese sau între platforme care rulează ROS folosind mesaje și abstractizează hardware-ul robotic pentru a avea control la nivel scăzut (de bază). Având în vedere problema identificării clienților, ROS are mai multe pachete care oferă o soluție pentru principalele problemele de tip computer vision ale acestei lucrări.

NAOqi este un sistem de operare special conceput pentru roboții creați de SoftBank Robotics și Aldebaran, precum Pepper, Nao și Romeo. Ca și în cazul precedent, acesta este compus din mai multe module care abstractizează hardware-ul roboților. Acesta oferă acces la senzorii și actuatorii roboților pentru o dezvoltare ușoară a software-ului. Permite comunicarea între module, locuri de muncă paralele și cresc numărul de evenimente. Chiar dacă nu este la fel de bine definit, așa cum este cazul ROS, NAOqi oferă câteva module care ajută la dezvoltarea acestui proiect.

6. Interacțiune Vocală

Clienții vor comunica cu Pepper prin interacțiune vocală dar există și posibilitatea interacțiunii tactile oferită de tableta atașata pe corpul său. În viața de zi cu zi, oamenii interacționează între ei prin comunicare orală, atunci când sunt fizic în același loc. În consecință, interacțiunea vocală va fi principalul canal de comunicare dintre client și Pepper.

Ne-am concentrat în primul rând pe limba engleză, urmând ca în etapa viitoare să adaptăm interacțiunea la limba română. Interfața vocală este compusă din cinci părți principale: recunoașterea automată a vorbirii (ASR), înțelegerea limbajului natural (NLU), modulul de gestionare a dialogului (DM), generarea limbajului natural (NLG) și sinteza text-vorbire (TTS).

6.1. Un Pipeline pentru Interacțiune Vocală

Modulul ASR permite unui sistem să identifice cuvintele rostite de o persoana și să le transforme într-un text scris. În anul 1994, ASR a fost definit de Stuckless [1] ca o transcriere independentă, bazată pe calculator, a limbii vorbite în text lizibil, în timp real.

Modulul NLU converteşte textul primit de la modulul ASR la o reprezentare de citire a maşinii. Extrage semnificaţia semantică din text. Construirea unui sistem de dialog cu analiză semantică independentă de un domeniu necesită ca analiza semantică sa parseze propozițiile in arbori. De asemenea, necesita si analiză sintactică in care etichetele semantice sunt ataşate la un arbore de parsare specific generat in cursul analizei semantice.

Modulul DM este responsabil pentru starea și fluxul conversației. Acesta primește exprimarea formatată a utilizatorului din NLU ca element de intrare și generează ca element de ieșire o reprezentare semantică pentru o listă de instrucțiuni din dialog. Lista determină care ar trebuie să fie răspunsul sistemului la intrarea procesată a utilizatorului.

Modulul NLG poate fi văzut ca opusul modulului NLU. Acesta este activitatea de prelucrare a limbajului natural de a genera limbajul natural dintr-un sistem de reprezentare a mașinii, cum ar fi o bază de cunoștințe sau o formă logică.

Modulul TTS convertește orice text generat din textul anterior într-un discurs care este produs în mod artificial și va fi auzit de difuzoarele sistemului. Similitudinea cu vocea umană și abilitatea de a fi înțeles clar sunt principalele criterii de evaluare a calitatii modulului TTS.

6.2. Implementare

În ceea ce privește partea de recunoaștere vocală a interfeței am folosit modulul *ALSpeechRecognition* furnizat de sistemul de operare NAOqi. Acest modul se bazează pe tehnologii sofisticate de recunoaștere vocală furnizate de NUANCE. Am populat modulul cu o listă de fraze care ar trebui să fie recunoscute, stocate într-o matrice. După aceasta începe lucrarea de recunoaștere, cheia *SpeechDetected* care conține un ***limbaj Boolean*** va specifica pentru ALSpeechRecognition dacă există un difuzor care este în prezent audiat sau nu. Dacă se aude un vorbitor, elementul listei care se potrivește cel mai bine cu ceea ce aude Pepper este plasat în cheia *WordRecognized* împreună cu gradul de încredere care reprezintă o estimare a probabilității ca elementul detectat să fie într-adevăr ceea ce a fost pronunțat de utilizator uman. În afară de a fi adăugate la cheia *WordRecognized*, acestea sunt adăugate și în cuvântul *WordRecognizedAndGrammar* cu adăugarea numelui de gramatică, care este folosit de motorul de recunoaștere. Cheile vor avea următoarea structură:

WordRecognized Key:
[element1, confidence1, element2, confidence2, …, elementN, confidenceN]
WordRecognizedAndGrammar Key:
[element1, confidence1, grammar1, element2, confidence2, grammar2, …, elementN, confidence_N, grammar_N]

După recunoaștere, fiecare element al listei conține unele acțiuni stocate care vor fi inițiate odată ce elementul se potrivește cu elementul de intrare pronunțat de un utilizator uman.

În ceea ce privește sinteza discursului, am folosit modulul ALTextToSpeech furnizat de sistemul de operare NAOqi. Acesta permite robotului să vorbească textul scris care a rezultat

din modulele anterioare. Acesta autorizează personalizarea vocii și trimite comenzi către motorul text-to-speech. Motorul "text-to-speech" se bazează pe tehnologiile furnizate de NUANCE. Parametrii fluxului audio de ieșire sunt flexibili, ceea ce ne-a permis să modificăm unii dintre ei, cum ar fi volumul, pasul inițial al vocii și viteza vorbirii, pentru a se potrivi cel mai bine nevoilor fiecărui caz. De asemenea, am folosit etichete pentru a adăuga o anumită expresivitate vorbind lui Pepper, făcând schimbări, în mijlocul unei propoziții a timbrului vorbirii, vitezei, volumului discursului, dar și adăugarea de pauze între cuvinte și schimbarea accentelor cuvântului. Rezultatul sintezei este trimis la difuzoarele robotului. De asemenea, am testat câteva API-uri online pentru diferitele module de interacțiune a vorbelor, dar exista un inconvenient să le integrăm in robotul Pepper, datorită numărului limitat de apeluri API și a timpilor lungi de răspuns. Următorul tabel ilustrează o serie de comenzi vocale, procentul de recunoaștere a comenzii (testat cu 7 utilizatori) și acțiunea robotului Pepper.

Comanda	Recunoaștere comanda (%)	Acțiunea robotului Pepper
Who are you?	88.12%	Se prezinta pe sine (I am Pepper)
Who painted Queuing for Bread?	86.08%	Afișează pictura si spune Nicolae Tonița.
What had Nicolae Grigorescu painted?	83.24%	Afișează o listă cu cele mai importante lucrări ale pictorului și o citește vocal
Where can I find the cleaning products?	88.56%	Afișează direcția care trebuie urmată pentru a ajunge în zona specifică din magazin. In același timp, răspunde.
In ce zi suntem?	91.15%	Spune data curenta.
Cat este ora?	89.74%	Spune ora curenta.
Realizează o fotografie	93.41%	Realizează o fotografie si o afișează
Play Music	90.25%	Întreabă (vocal) utilizatorul ce fel de muzica dorește sa asculte, după care reda o melodie, ținând cont de opțiune.
Who am I?	89.78%	Spune: “Lasă-mă sa mă gândesc”, după care, daca persoana este in baza de date (recunoscuta), Pepper va pronunța numele acesteia. Spune: “Lasă-mă sa ma gândesc”, după care, daca persoana nu se regăsește in baza de date, Pepper va spune “Nu te cunosc”
What is your battery level?	100.00%	Transmite (vocal) nivelul bateriei.
What is your IP address?	100.00%	Transmite (vocal) adresa IP
Are you connected to Internet?	96.12%	Transmite (vocal) daca este conectat sau nu la internet

Tabel 1. Unele comenzi vocale, procentul de recunoaștere a comenzii si acțiunile lui Pepper

7. Abordări de Computer Vision

În contextul proiectului, obiectivul specific al modulului de computer vision (vedere computerizată) este detectarea persoanelor care interacționează cu robotul, și anume:

● detectarea și recunoașterea persoanei

● dacă o persoană aflata în fața robotului se adresează acestuia, robotul trebuie să fie capabil să detecteze acea persoană și să înceapă să interacționeze cu ea;

● dacă un grup de persoane se află în fața robotului, acesta trebuie să aleagă o persoană din grup, cu care să interacționeze.

Pentru a implementa scenariile propuse, robotul trebuie să detecteze și să recunoască persoanele din jur, trebuie să identifice o persoană sau un grup de persoane din fața lui, iar în cazul în care o persoană este în picioare pentru a interacționa cu robotul, el trebuie să-si concentreze atenția pe acea persoană până când persoana va ieși din cadru. Dacă există mai mulți oameni care sunt orientați spre robot, acesta trebuie să aleagă una dintre acestea, asupra careia sa se concentreze, pe baza unor criterii specifice. Mai mult decât atât, robotul ar trebui să aibă un modul de recunoaștere a persoanei bazat pe o memorie pe termen scurt, astfel încât să poată recunoaște persoanelecu care a interacționat, deoarece dacă o persoană se retrage din focalizare, aceasta poate reveni pentru a pune alte întrebări ulterior.

Soluția pentru problema identificată este destul de complexă și trebuie să combine mai multe metode și abordări. Aceasta necesită mai multe module, iar cele principale sunt: detectarea persoanelor combinată cu urmărirea, detectarea feței și recunoașterea feței.

7.1. Detectarea Persoanei

Detectarea persoanelor este un modul extrem de important, iar, comportamentul de bază al acestui modul include captarea de imagini ca elemente de intrare și generarea de informații despre oamenii din imagini. Informația se referă la poziția relativa fata de aparatul de fotografiat al robotului, pentru a ști care dintre persoanele detectate sunt relevante în ceea ce privește identificarea utilizatorilor.

Au fost testate multiple abordări pentru detectarea persoanelor. Cele mai relevante sunt construite pe funcționalitatea care există în cadrul NAOqi și în rețeaua YOLO.

7.1.1. Detectarea Persoanelor cu NAOqi

Robotul este capabil să detecteze și să urmărească persoane în diferite poziții, dar detectările cele mai exacte sunt atunci când persoana din fața robotului se ridică în dreapta, se rotește sau nu. Cadrul oferă, de asemenea, posibilitatea de a detecta persoanele care stau jos, dar se află în afara sferei de aplicare a scenariului propus. Acest modul particular adună informații despre oamenii din jurul robotului și actualizează în permanență atributele după o anumită perioadă. Informațiile despre persoane sunt actualizate utilizând imaginile primite de la camerele RGB și de la senzorul 3D. Deoarece actualizează atributele în mod constant, aceasta integrează și o parte din urmărirea persoanelor. Cartografia dintre atribute și persoana se face

în funcție de ID-ul persoanei, care este invariabil în timp, dacă persoana nu este pierdută de catre robot.

Rezultatele modulului de detectare și urmărire a persoanelor sunt foarte puternic influențate de condițiile de luminozitate din imagine. Limitările acestui modul apar atunci când există ocluziuni, lumină de fundal extrem de puternice sau interacțiunea între persoane, astfel încât robotul să ia în considerare mai puține, mai multe sau nici o persoană în imagini. Dacă persoana aflată în fața robotului este în picioare și mai ales dacă se află în apropierea unui perete, modulul va detecta mai greu persoana, comparativ cu cazul în care persoana se mișcă. Cu toate acestea, aceasta nu este o problemă care ar putea afecta drastic funcționarea robotului.

7.1.2. Rețeaua YOLO2

Rețeaua YOLO2 (You Only Look Once) este în prezent cea mai bună opțiune a noastră pentru detectarea persoanelor. Este o metodă foarte solidă, care este aproape invariabilă de poziționare și luminozitate. Detectează persoane în poziții diferite, chiar și persoane care se află la birou în spatele laptopului. În plus, detectează persoane care se află la mai mult de 3 metri de cameră, folosind rezoluția robotului, de 640x480 la 15 cadre pe secundă. Pentru a urmări persoanele detectate, folosim un mecanism simplu de similitudine între detectări și le atribuim ID-uri diferite. Considerăm că îmbunătățim acest mecanism cu unul mai solid.

Figură 4. Detectarea persoanelor cu ajutorul rețelei YOLO2

Avantajul acestei metode este că oferă o detectare precisă foarte bună în timp real. Acesta oferă rareori negative false, ceea ce este foarte important pentru proiectul nostru. Dar, pentru a da rezultate în timp real, trebuie să fi rulat pe o mașină foarte puternică, mai specifică, neputând rula în timp real pe robot. De aceea luăm fluxul de imagine de la robot, il procesăm și returnăm doar detectările robotului. Un alt dezavantaj al metodei este că nu oferă informații despre adâncime, dar am combinat informațiile de la senzorul 3D cu detectările date de rețea, pentru a avea o distanță aproximativă a persoanei detectate. Problema cu care ne-am confruntat aici este că informațiile de adâncime date de senzorul 3D au intervalul cuprins între 0,4 și 3 metri. Fiecare detectare care este mai mare de 3 m este calculată la 0,4 m. Așadar,

ținem cont și de mărimea detecției atunci când se calculează distanța față de persoana din obiectiv.

7.2. Detectarea Facială

Detectarea feței este o problemă frecventă în aplicațiile de astăzi, așa că am putea încerca mai multe abordări pentru acest modul. Am testat funcționalitatea încorporată a robotului, dar am încercat și alte modele, cum ar fi cele definite în biblioteca OpenCV.

7.2.1. Detectarea Facială cu NAOqi

Modulul de detectare a feței în NAOqi este ALFaceDetection. Soluția sa pentru detectarea feței nu este publică, fiind o soluție dezvoltată de Compania Omron. Acest modul oferă rezultate bune. Având în vedere faptul că modulul de detectare a feței caută numai fețe, în timp ce modulul de detecție umană încearcă să detecteze persoane care se află în poziții mai diverse, cum ar fi așezata sau în picioare cu spatele, funcționalitatea pentru detectarea feței este mai precisă decât cea umană detectare. Peper este capabil să detecteze simultan mai multe fețe, furnizând informații suplimentare despre ele. În ceea ce privește modulul anterior, acest modul are o parte din urmărirea feței, astfel încât informațiile despre fețe să fie actualizate în mod constant. Cartografia se face folosind ID-ul feței, care este diferit de id-ul persoanei. Aceasta conduce la concluzia că numărul de persoane și chipurile imaginii poate să nu corespundă întotdeauna.

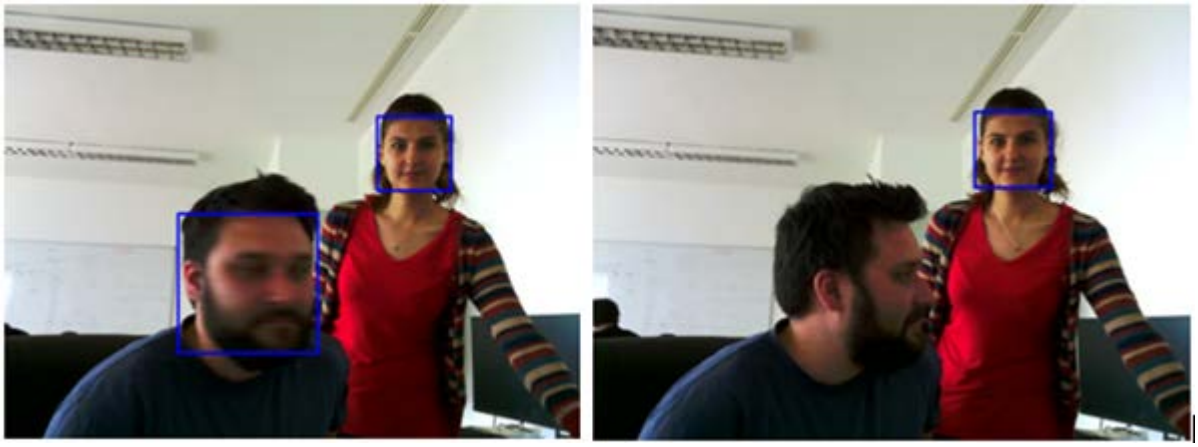
După ce o persoană este detectată, putem obține un set de caracteristici despre aceasta, cum ar fi vârsta, sexul, expresia feței etc. Ceea ce extragem este informații despre direcția privării, perioada în care persoana se află în față a robotului și informații despre poziția persoanei față de camera robotului. Acest tip de informații este util pentru interacțiunea umană, pentru a încerca să facă robotul să iasă din starea pasivă pentru a interacționa cu persoana din fața sa.

Avantajul modulului NAOqi este că după detectarea unei fețe, aceasta este urmărită, deci este mai greu de pierdut, chiar și atunci când fața se rotește cu 90 de grade. Dezavantajul este că are aceleași limitări importante. Nu funcționează foarte bine atunci când oamenii interacționează și sunt aproape unul de altul sau când există o lumină puternică în spate. Pentru performanțe bune, acest modul necesită o imagine de minimum 20 de pixeli în imagine și o rotație de maximum 45^0. Eroarea pentru acest modul este aceeași cu eroarea de detectare umană și, pentru a fi mai precis, eroarea este aproximativ 1 față, având în vedere numărul de fețe din imagine.

7.2.2. Detectarea Facială cu OpenCV

Metoda pe care o folosim în mod curent pentru detectarea feței se bazează pe utilizarea clasificatorilor in cascadă bazați pe trăsături Haar din biblioteca OpenCV. După testare, s-a dovedit ca metoda da rezultate destul de bune în timp real. Această metodă nu are aceleași limitări ca cea încorporată, dar uneori oferă unele poziții false, care pot afecta rezultatele. Metoda clasificatorului Haar se bazează pe câteva caracteristici specifice, care sunt extrase din imagini: caracteristici ale marginilor, caracteristici linii și caracteristici dreptunghiulare. Este

instruit pe seturi mari de imagini pozitive și negative și, în funcție de setul de antrenament, poate detecta nu numai fețe, ci și alte tipuri de obiecte.



Figură 5. Detectie faciala OpenCV

Am testat funcționalitatea OpenCV în aceleași condiții ca și pentru modulul NAOqi. Rezultatele utilizând biblioteca OpenCV depind foarte mult de claritatea și direcția feței și, așa cum se poate vedea în imaginea de mai sus, rezultatele sunt influențate de unghiul de rotație a feței. Are probleme cu fețele cu rotație de aproximativ 90 de grade, dar detectează corect fețele care privesc direct camera. Distanța maximă de detecție între fețe și cameră depinde foarte mult de cât de clare sunt caracteristicile feței definite în imagine. Cu toate acestea, nu ne interesează detectarea fețelor atunci când oamenii nu privesc direct la robot, deoarece folosim modulul de detectare a persoanelor în paralel. Trebuie doar să recunoaștem oamenii și, pentru o predicție încrezătoare, avem nevoie de toată fața.

Având în vedere testele pe care le-am făcut și capacitățile robotului, precizia detectării feței folosind biblioteca OpenCV este de aproximativ 80%.

7.3. Recunoașterea Facială

Pentru recunoașterea feței, am încercat, de asemenea abordări multiple. Cele mai relevante sunt funcționalitatea cadru încorporată și modulul bibliotecii OpenCV. Am testat, de asemenea, un API online pentru recunoașterea feței, dar este incomod să se integreze in robotul Pepper, datorită timpilor lungi de răspuns și numărului limitat de apeluri API.

7.3.1. Recunoașterea Faciala cu NAOqi

Modulul încorporat este foarte sensibil la lumină și la scală (dimensiuni), prin urmare este nevoie de condiții bune de mediu pentru a funcționa acceptabil. Etapa de învățare este una sensibilă. Robotul nu va învăța o față până când aceasta nu este definită foarte clar, fără umbre sau lumină de fundal.

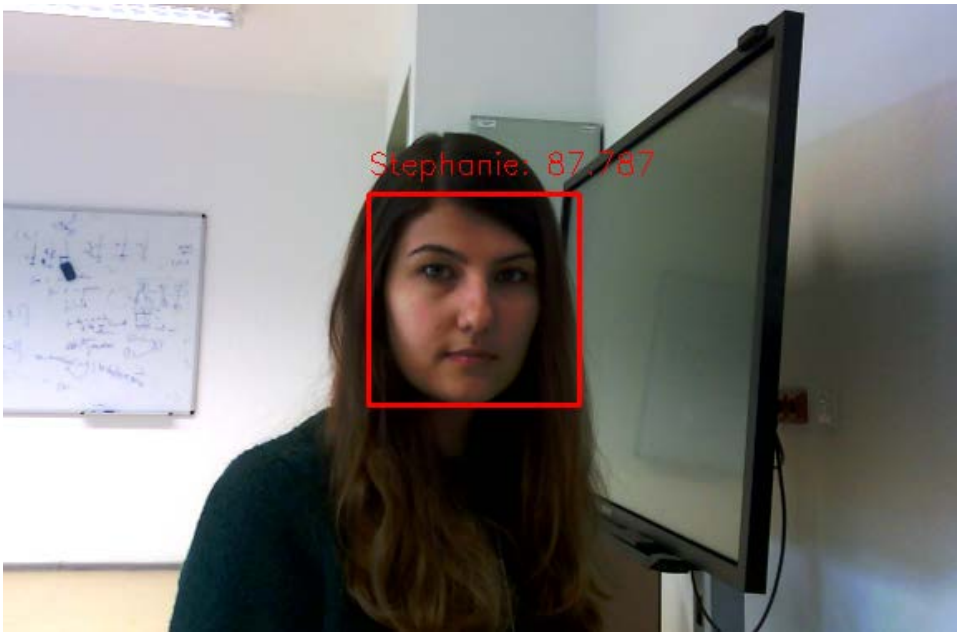
Recunoașterea feței în cadrul NAOqi funcționează bine, deoarece nu dă atât de multe rezultate false. Încearcă să recunoască persoana pentru fiecare față detectată și întoarce o predicție a unei persoane numai dacă încrederea este peste un anumit prag. Performanțele sale sunt limitate în ceea ce privește distanța, ceea ce înseamnă că nivelul de recunoaștere este bun dacă fața este relativ apropiată de cameră. Cu toate acestea, în condiții bune, rezultatele

acestui modul sunt foarte exacte. Pe baza testelor pe care le-am făcut, am determinat ca precizia predicției este de aproximativ 70%. Această precizie este obținută în condiții bune în ceea ce privește luminozitatea, ocluziile și distanța față de cameră.

7.3.2. Recunoașterea facială cu OpenCV

În comparație cu metoda anterioară, recunoașterea feței în OpenCV este mai precisă și mai puțin riguroasă. Am testat mai mulți algoritmi: Histograms, Eigenfaces, Fisherfaces și Local Binary Patterns Histograms. Având în vedere setul mic de fețe și rezoluția camerei, am obținut cele mai exacte rezultate în ceea ce privește recunoașterea feței folosind Local Binary Patterns Histograms.

Problema cu această metodă este nivelul de încredere, pentru că avem nevoie de filtrarea rezultatelor, dar nu este foarte bine definită. Recunoașterea feței în OpenCV funcționează foarte bine pentru fețele introduse în baza de date, dar problema este aceea că returnează fețe din baza de date chiar și pentru persoanele necunoscute. Am încercat să abordăm problema luând în considerare încrederea predicțiilor și am stabilit un prag pentru previziuni, astfel încât, chiar dacă avem mai puține predicții, precizia este mai bună.



Figură 6. Recunoasterea faciala OpenCV

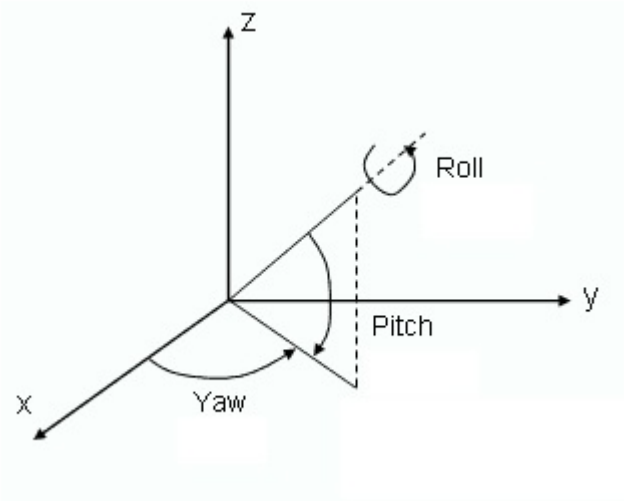
7.4. Cartografierea Fețelor Detectate pentru Detectarea Persoanelor

Înainte de a utiliza arhitectura YOLO, am folosit funcționalitatea incorporate a robotului, deoarece acesta se potrivea cel mai bine cerințelor noastre. Acesta a oferit o mulțime de informații utile, cum ar fi: distanța față de robot, unghiuri, poziția în coordonatele mediului, înălțimea reală etc., cu o eroare mică și din acest motiv, este puțin probabil să fie înlocuit pentru moment. Cu toate acestea, au existat mai multe tehnici oferite pentru detectarea și analiza feței decât cele ale robotului. De aceea, am avut nevoie de un modul pentru a cartografia persoanele

detectate de modulul de detectare umană cu chipurile identificate printr-o metodă externă, cum ar fi funcția OpenCV.

7.4.1. Abordare și Implementare

Modulul de detectare a persoanelor declanșează cartografierea, deoarece dacă robotul nu detectează nici o persoană din imagine, nu ar trebui să existe nici o față. Am folosit informațiile oferite de modulul de detectare umană pentru a obține unghiurile yaw și pitch ale capului oamenilor în imagine.

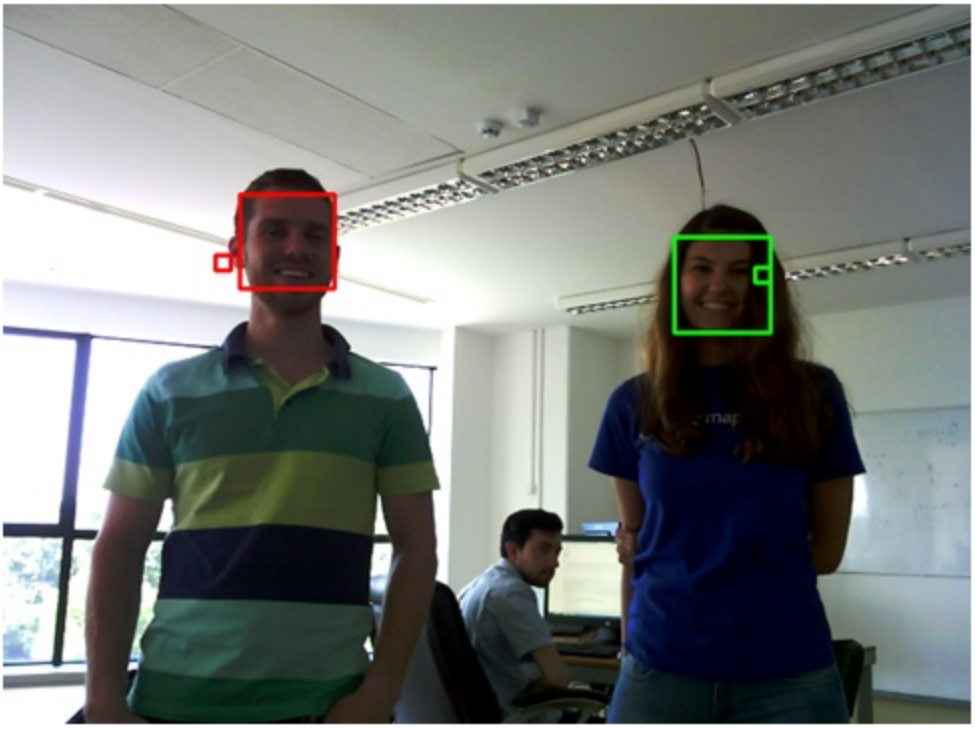


Figură 7. Unghiurile capului

Am folosit o funcție de transformare pentru a transforma unghiurile în coordonatele imaginii (pixeli), deoarece funcția OpenCV pentru detectarea feței returnează poziția feței în imagine (pixeli). După procesarea imaginilor, avem atât oamenii, cât și chipurile din imagine, așa că începem să le potrivim. O problemă foarte frecventă este că numărul de fețe și de oameni din imagini este diferit. Deci, am rezolvat această problemă considerând cea mai mică matrice (chipuri sau persoane). Dacă în imagine sunt mai puține chipuri detectate, potrivim fețele cu oamenii, altfel potrivim oamenii cu fețele. Informația pe care o avem de la modulul de detectare a persoanelor este un punct relativ al poziției de sus a capului, în timp ce din detecția feței avem regiunea în care a fost detectată fața. Pentru a le potrivi, am folosit distanța Euclidiană, unde un match presupune o distanță minimă Euclidiană între punctul de cap și centrul orizontal al feței.

7.4.2. Rezultate

Rezultatele depind în mare măsură de modulul de detectare umană și, mai precis, de pozițiile capului persoanelor. Un rezultat al potrivirii pentru două persoane poate fi văzut în imaginea de mai jos.



Figură 8. Recunoasterea faciala OpenCV

În această imagine, pătratele mai mari sunt fețele detectate cu biblioteca OpenCV, în timp ce cele mici reprezintă pozițiile capului persoanelor din imagine, detectate cu cadrul NAOqi. O potrivire pentru persoane-față este reprezentată de aceeași culoare în imagine. După cum puteți vedea în imagine, există o persoană suplimentară în imagine, totuși datorită luminozității și poziției sale nu este detectată.

7.5. Utilizarea Datelor de Adâncime pentru Recunoașterea Activității

După ce a fost construit un modul care detectează și identifică persoanele în imediata vecinătate a lui Pepper, următorul pas natural pentru creșterea nivelului de înțelegere, de către robot a lumii (și, în mod specific, pentru a si cum sa interacționeze cu oamenii) este de a recunoaște cel puțin unele dintre activitățile de bază ale persoanei țintă. De exemplu, acest lucru ar permite lui Pepper să înțeleagă dacă utilizatorul este inactiv și poate fi abordat, sau dacă este angajat și nu trebuie deranjat. Există multe alte cazuri în care Pepper ar putea să folosească înțelegerea a ceea ce este acțiunea instantanee a utilizatorului, cunoașterea care poate acționa ca declanșator al diferitelor sale comportamente.

Având în vedere capacitățile senzoriale ale lui Pepper, este imediat evident că, pentru a aplica metode de recunoaștere a activității umane, putem folosi fie fluxul de imagine RGB, fie fluxul de imagine 3D furnizat de senzorii 3D. Comparând atribute precum rezoluția sau numărul de cadrele pe secundă între cele două tipuri de senzori se preciza faptul că fluxul RGB funcționează mai bine. Cu toate acestea, deși calitatea senzorului 3D este satisfăcătoare pentru cazurile in care le utilizează Pepper, am decis să folosim date de adâncime din mai multe considerații:

- Datele de adâncime măresc șansele unei segmentări mai bune a unei persoane.

- Datele de adâncime sunt mai solide la schimbările de iluminare și sunt mai mult sau mai puțin invariabile pentru situații legate de lumină slabă la contralumina.
- Seturile de date supravegheate ale activităților umane potrivite pentru utilizarea cu metode de învățare mecanică sunt rare și de obicei nu foarte extinse; Prin urmare, datele de adâncime se pot dovedi a fi mai discriminatorii decât datele RGB din motive practice.

Secțiunile care urmează descriu, in detaliu, capacitățile senzoriale ale datelor de adâncime ale lui Pepper, stadiul actual al tehnicii privind recunoașterea activității din datele de adâncime. Vom descrie, de asemenea, două dintre abordările noastre și rezultatele intermediare obținute și vom discuta concluziile de până acum și o foaie de parcurs pentru viitoarele evoluții.

7.5.1. Capabilitățile de Vedere 3D ale lui Pepper

Robotul robot este echipat cu o cameră de profunzime 3D situată în cap. Este de fapt o versiune adaptată a unei camere de adâncime Asus Xtion cu un câmp de vedere de 58^0 orizontal, 45^0 vertical și 70^0 în diagonală. Distanța de utilizare este cuprinsă între 0,8 și 3,5 metri, focalizare fixă (40cm ~ 8m), care oferă imagini de adâncime la o rezoluție maximă de 320 × 240 pixeli la 20 de cadre pe secundă. În ceea ce privește precizia, camera de adâncime oferă un interval dinamic de 68.2db, în timp ce raportul semnal-zgomot este de 45dB.

Comparat cu cel mai comun senzor Kinect de la Microsoft, Asus Xtion este mai puțin performant, dar probabil o alegere mai bună pentru integrarea într-un robot. Rezoluția Kinect este ușor mai bună (512 × 424 pixeli), cu o distanță ceva mai mare în domeniul de utilizare (0,5 - 4,5 metri) și un câmp vizual crescut (70 ° orizontal și 60 ° vertical). Mai mult, senzorul Kinect a fost utilizat mult mai des în cercetarea academică, în special în colectarea seturilor de date pentru pozițiile și activitățile umane. Luând în considerare acest lucru, am decis să folosim setul de date NTU RGB + D [6], un set de date despre activitatea umană care se bazează pe înregistrări Kinect. Acesta este una dintre cele mai mari baze de date bazate pe adâncime disponibile astăzi și multe dintre tehnicile de ultimă oră îl folosesc pentru a raporta performanța. Clasele de activitate pe care le urmează sunt similare cu cele pe care am dori să le putem detecta folosind Pepper. Detaliile tehnice referitoare la acest set de date sunt descrise în secțiunile care vor urma.

7.5.2. Abordări Bazate pe Vederea în Adâncime

În domeniul computer vision, detectarea și recunoașterea acțiunii umane rămâne o problemă provocatoare, care facilitează o gamă largă de aplicații practice. În trecut, datele video RGB au fost studiate pe scară largă ca elemente de intrare pentru recunoașterea acțiunii umane, dar, comparativ cu datele RGB, modalitatea de adâncime este invariabilă pentru variațiile de lumina și oferă informații structurale 3D ale scenei. După lansarea programului Microsoft Kinect [2], care utilizează senzori de adâncime și algoritmi pentru estimarea foarte precisă a pozițiilor comune din harta de adâncime, recunoașterea acțiunii bazate pe schelet a devenit un subiect activ de cercetare. Comparând cu datele vizuale 2D, informațiile suplimentare de adâncime oferă mai multe avantaje. Caracteristicile extrase din îmbinările cu puncte 3D sunt solide pentru variațiile de scală și rotație, iar reprezentările umane bazate pe

informații scheletice 3D oferă o alternativă foarte promițătoare. În prezent, progresele tehnologiilor de imagistică in adâncime au adus multe beneficii, recunoașterea activității umane devenind realizabilă fără a atașa marcatori optici sau orice alt senzor de mișcare asupra corpului uman.

Detectarea și eliminarea fundalului folosind doar imaginea RGB este dificilă și ineficientă, dar cu caracteristicile imaginii de adâncime, această operație poate fi ușor, deoarece putem folosi faptul că subiectul este întotdeauna la o anumită distanță față de pixelii de fundal. Trebuie să găsim o valoare a adâncimii pragului pentru toti subiectii de deasupra cărora clasificăm toți pixelii ca fundal și obținem cu ușurință silueta imaginii in adâncime.

Problema recunoașterii acțiunii umane a fost studiată în ultimii ani și au fost propuse mai multe metode. Scopul principal este de a crea un model care să poată extrage caracterele principale ale fiecărei acțiuni care rezolvă problema variabilității intra-clase și a similarității între acțiuni. La început, sa pus un accent deosebit pe utilizarea modelelor Hidden Markov (HMM).

Ahmad Jalal și colab. [3] a propus o nouă metodă de recunoaștere a activităților utilizând secvența de imagini de adâncime. În timpul etapelor de preprocesare, ei au extras siluete umane in adâncime folosind tehnici de fundal și de îndepărtare a pardoselii și siluete umane urmărite prin luarea în considerare a cutiei dreptunghiulare având măsurători ale formei corpului pentru a ajusta dimensiunea casetei. În timpul tehnicilor de extracție a caracteristicilor spațio-temporale, s-a luat in calcul: un set de caracteristici ca istoric secvențial de adâncime, care conține un flux optic, identificator de mișcare, unghiurile de îmbinare și caracteristicile de localizare a articulațiilor. Aceste trăsături sunt aplicate peste media K pentru clusterizare și sunt introduse într-un HMM cu patru stări de la dreapta la dreapta pentru instruirea / testarea activităților umane.

Chen Chen și colab. [4] a introdus o metodă interesantă prin utilizarea hărților de mișcare in adâncime. Fiecare cadru dintr-o secvență video in adâncime este proiectat pe trei axe ortogonale carteziene. Sub fiecare vedere de proiecție, diferența absolută dintre două hărți proiectate consecutive este acumulata printr-o întreagă secvența video in adâncime care formează o hartă a mișcării in adâncime. Un clasificator de reprezentare colaborativă regularizata l2 cu o matrice ponderată pe distanțe Tikhonov este apoi folosit pentru recunoașterea acțiunii. Metoda dezvoltată se dovedește a fi eficientă din punct de vedere al calculului, permițându-i să ruleze în timp real.

Cele mai multe metode existente de recunoaștere a acțiunii bazate pe schelete modelează în mod explicit dinamica temporală a legăturilor scheletice prin utilizarea modelelor ascunse Markov, dar este foarte dificil să se obțină secvențele aliniate temporal și distribuțiile corespunzătoare ale emisiilor. Modelele HMM sunt modele probabiliste care sunt în general aplicabile seriilor de timp ale secvențelor liniare. HMM-urile au avantajul că pot genera o aliniere pentru un număr mare de secvențe de aceeași acțiune fără a calcula mai întâi toate aliniamentele în perechi. De asemenea, pot fi utilizați algoritmi de programare dinamică standard, numiți Forward (pentru notare) și Viterbi (pentru aliniere), cu toate probabilitățile unei secvențe fiind calculate și generate de profilele HMM și pot fi folosite pentru a clasifica secvențe necunoscute pentru care model.

Cercetătorii au încercat, de asemenea, sa găsească o arhitectura adecvata a rețelei neuronale cu capacitatea de recunoaștere a acțiunii. Au fost propuse sisteme bazate pe o harta asociativă auto-organizatoare, o variantă a hărții auto organizatoare (SOM), care învață să-și

asocieze activitatea cu elemente suplimentare, astfel ca această soluție a fost capabilă să reprezinte în mod parsimonic acțiunile umane. În [5], Johnsson și colab. a prezentat o arhitectură compusă din trei niveluri. Primul nivel este un SOM care învață să reprezinte posturile în funcție de intrarea în sistem. Al doilea nivel este un alt SOM care reprezintă reprezentarea activității superimpuse în primul nivel SOM în timpul acțiunii (învață să reprezinte acțiuni). Al treilea nivel este o rețea neurală artificială supravegheată, care învață să eticheteze acțiunea.

Recent, metodele deep learning au atins performanțe remarcabile în diferite activități tip computer vision. Prin urmare, s-a încercat utilizarea acestei metode în problema recunoașterii acțiunilor umane. Acum, o modalitate comună de a captura informații spațio-temporale în secvențele de schelet este posibila prin utilizarea de rețele neuronale convoluționale sau rețele neuronale recurente.

Dat fiind faptul că modelele bazate pe deep learning au reușit să codifice și să învețe date secvențiale în diverse aplicații, au fost propuse mai multe metode bazate pe RNN pentru recunoașterea acțiunilor bazate pe schelet. Du și colab. [5] a propus o rețea ierarhică RNN prin utilizarea mai multor RNN-uri bidirecționale într-un mod ierarhic nou. Structura scheletului uman a fost împărțită la cinci grupuri majore comune. Apoi fiecare grup a fost introdus în RNN bidirecțional corespunzător. Ieșirile RNN-urilor au fost concatenate pentru a reprezenta corpul superior și corpul inferior, apoi fiecare a fost introdus în continuare într-un alt set de RNN-uri. Prin concatenarea rezultatelor a două RNN-uri, a fost obținută reprezentarea corporală globală, care a fost alimentată la următorul strat RNN. În cele din urmă, un clasificator softmax a fost folosit în [6] pentru a efectua clasificarea acțiunilor.

7.5.3. Abordări de Deep Learning

Setul de date NTU RGB + D conține în total 60 de clase de acțiuni, care sunt împărțite în trei grupe majore: 40 acțiuni zilnice (băut, mănâncă, citit etc.), 9 acțiuni de sănătate (strănut, împiedicare, cădere etc.) și 11 acțiuni reciproce (punching, kicking, îmbrățișare, etc.). 40 de persoane au fost invitate să facă acest set de date, iar vârsta subiecților este cuprinsă între 10 și 35 de ani. Pentru a deveni invariabil în perspectivă, au fost folosite trei camere în același timp. În prezent, acesta este cel mai mare set de date de recunoaștere a acțiunii bazate pe adâncime, furnizând coordonate 3D a 25 articulații colectate de Kinect v2. Acest set de date conține peste 56 000 de secvențe și 4 milioane de cadre, capturate în diferite condiții de fond.

Analiza activității Bazate pe Schelet

Conceptul de reprezentare pe bază de schelet poate fi urmărit până în cercetarea primară a lui Johanson (1973), care a demonstrat că un număr mic de poziții comune pot reprezenta în mod eficient comportamente umane. Proiecțiile bazate pe schelet 3D oferă performanțe promițătoare în aplicațiile din lumea reală, inclusiv in jocurile bazate pe Kinect, deoarece reprezentările bazate pe schelet 3D pot să modeleze relația articulațiilor umane și să codifice configurația întregului corp.

Corpul uman poate fi descompus, la modul grosier, în cinci părți: două brațe, două picioare și un trunchi; și acțiunile umane pot fi interpretate ca interacțiuni ale diferitelor părți ale

corpului. Articulațiile fiecărei părți ale corpului se mișcă mereu împreună și combinația traiectoriilor lor 3D formează modele de mișcare mai complexe.

Analizând metodele pe care le-am putea folosi pentru extragerea scheletului din informațiile furnizate de Pepper, am descoperit că există o modalitate de a conecta datele lui Pepper privind adâncimiea pentru deschiderea modulului ROS, care poate produce un schelet. După aplicarea acestei operații, putem folosi coordonatele articulațiilor pentru a determina acțiunea.

Folosind setul de date prezentat mai sus, am incercat o rețea neurală capabilă să identifice acțiunile folosind coordonatele articulațiilor scheletului. Pentru a putea dezvolta acest clasificator, a trebuit să analizăm informațiile furnizate de acest set de date pentru a produce o serie de statistici pe baza cărora am decis ce trebuie să facem în etapa de preprocesare. Următoarea fază a constat în instruirea și testarea mai multor arhitecturi de retele neuronale.

Pentru a elimina echiparea excesivă și pentru a ajuta rețeaua să învețe mai repede caracteristicile fiecărei acțiuni, s-au propus o serie de transformări care trebuie aplicate pe schelet în etapa de preprocesare. Pentru recunoașterea acțiunii bazate pe schelet, datele de intrare, care sunt secvențe de tip coordonate 3D ale articulațiilor și rețelellor neuronale necesită adesea o mulțime de date pentru a îmbunătăți generalizarea și pentru a preveni suprapunerea. Astfel, un set de date nu poate conține suficiente informații, astfel ca pentru a utiliza cât mai bine oferta limitată de date de antrenament, pot fi folosite tehnicile de augmentare a datelor bazate pe transformarea 3D.

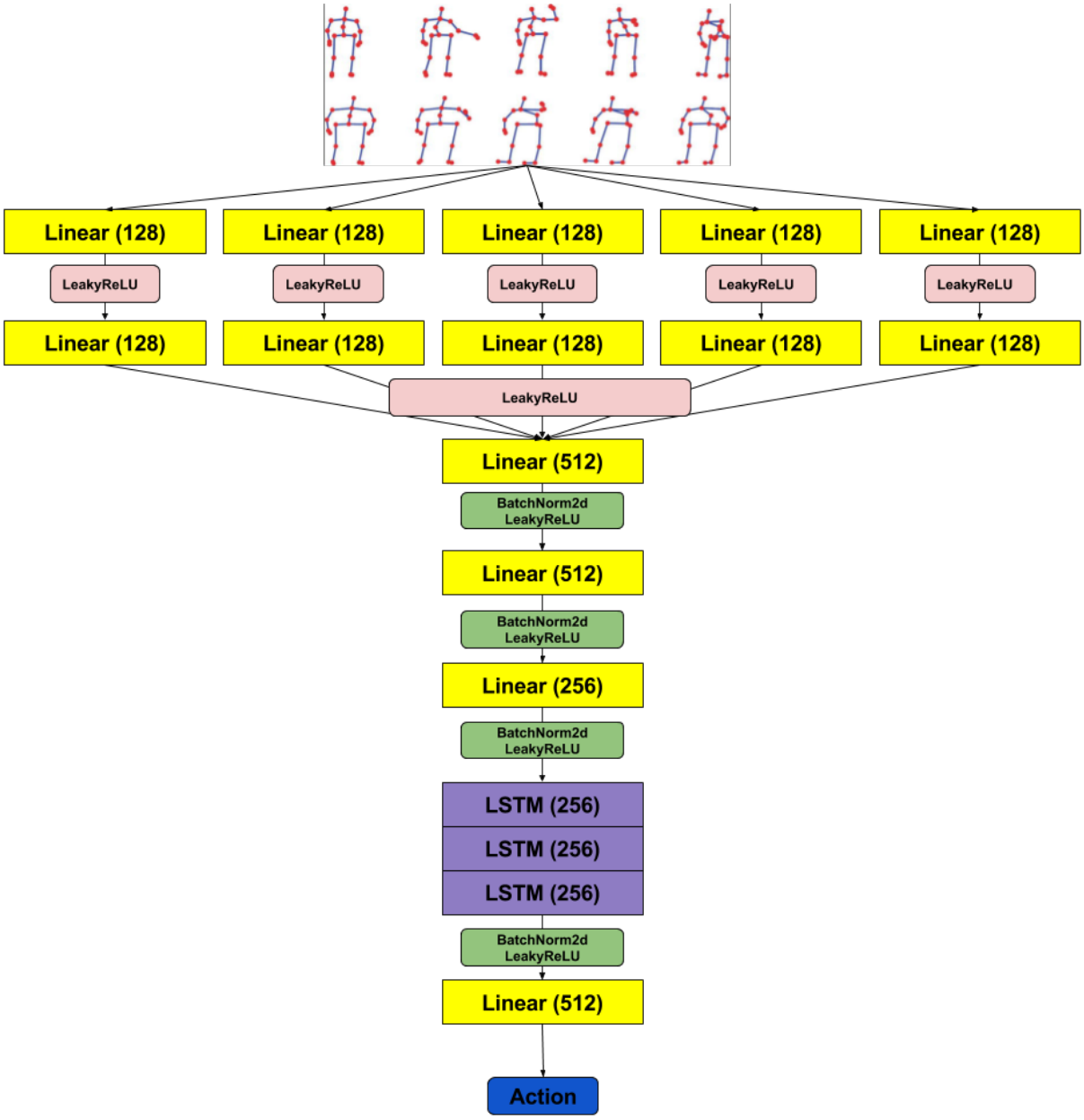
Pentru acțiunile care implică mai mult de un schelet, a fost important să se găsească o modalitate de a selecta cel mai activ schelet. Într-o primă încercare, am încercat să alegem cel mai activ schelet pentru fiecare acțiune, dar rezultatele au arătat că această transformare nu este cea corectă. Apoi am decis ca, în faza de antrenament, să folosim fiecare schelet separat și, în faza de testare, să calculam media rezultatelor pentru fiecare schelet înainte de a determina clasa.

Schemele trebuie să fie normalizate, iar acest lucru se poate face prin scăderea articulației centrale, care este media coordonatelor 3D ale centrului șoldului, șoldului stâng și șoldului drept. Pentru acțiuni care implică mai mult de un schelet, este important de găsit o modalitate de a selecta cel mai activ schelet.

Având în vedere că acțiunile din acest set de date au avut un număr variabil de cadre, a trebuit să găsim o modalitate de a asigura o dimensiune constantă a intrării rețelei neuronale în vederea utilizării tehnicilor mini-batching. Pentru aceasta, am implementat o transformare care asigură extragerea unui număr constant de cadre pentru eșantioane mai mari sau adăugarea de zerouri pentru cele mai mici. Pentru a alege cadrele stocate, am folosit o distribuție generată uniform.

Arhitectura rețelei neuronale utilizată pentru recunoașterea acțiunii este prezentată în Figura 9.

Primul nivel conține 5 linears care primesc secvența de cadre care descriu o acțiune. Pentru datele lor de ieșirile lor, este folosit LeakyReLU ca funcție de activare, iar rezultatul este folosit ca intrare pentru alte 5 linears, situate pe al doilea nivel. Al treilea nivel conține, de asemenea, un element liniar și primește ieșirile celui de-al doilea nivel concatenat și trecut printr-o funcție de activare. Următoarele două nivele conțin, de asemenea, o linie și rezultatul este în cele din urmă trimis la o memorie pe termen lung (LSTM).



Figură 9. Arhitectura rețelei de recunoaștere a activității bazată pe schelet

Setul de date NTU RGB + D are două protocoale standard de evaluare [7]: primul este protocolul de evaluare cu subiect încrucișat, în care jumătate dintre subiecți sunt folosiți pentru instruire, iar restul subiecților pentru testare, iar al doilea este protocolul de evaluare cu vedere încrucișată, în care 2/3 din puncte de vedere sunt folosite pentru instruire și 1/3 pentru testare. Folosind protocolul cu subiect încrucișat și arhitectura de rețea prezentată anterior, precizia obținută pentru testare este de 65,62%, fiind obținută după 861 de epoci.

Activitatea de Recunoaștere pe Bază de Adâncime

În timpul experimentelor, folosind datele din schelet din setul de date, am descoperit o serie de erori consecvente cauzate de Kinect și modul în care extrage scheletul:

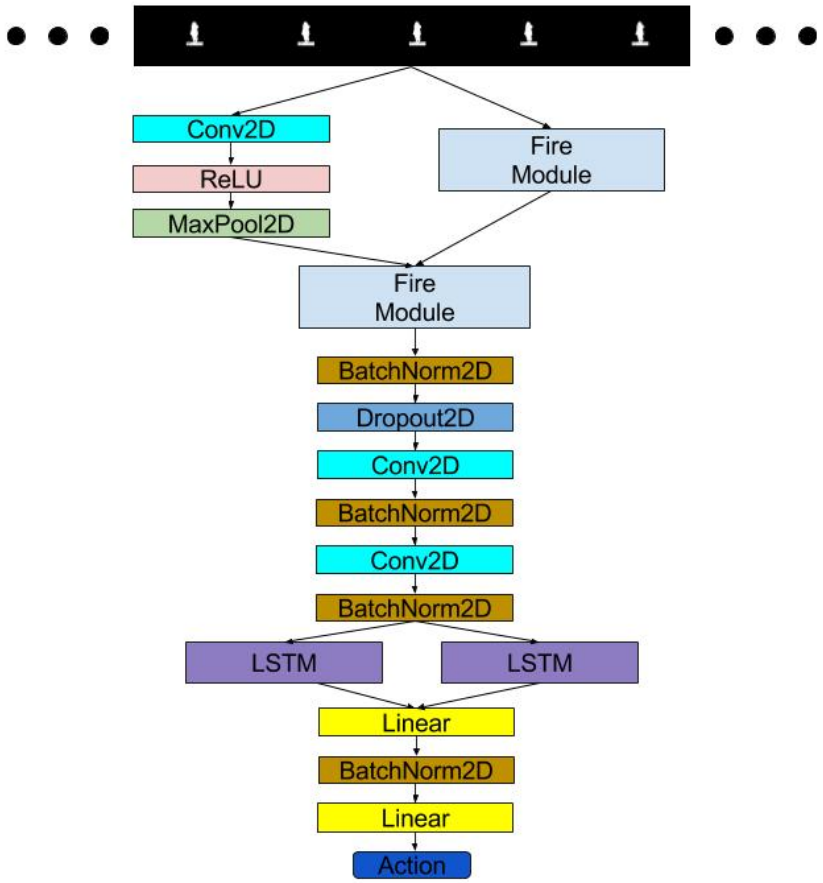
- Reflecțiile oamenilor pe unele suprafețe generează schelete de tip "fantomă"
- Ocluziile parțiale ale persoanelor cauzate de obiecte (ex: scaune) conduc la schelete incorecte
- Ocluziile parțiale ale unor părți ale oamenilor (de exemplu, poziții ale persoanelor perpendiculare pe vederea camerei) conduc la schelete incorecte
- Obiectele adiacente uneori conduc la schelete incorecte
- Situațiile în care persoanele se află foarte aproape (de exemplu, îmbrățișarea) duc uneori la schelete incorecte
- În unele secvențe, uneori identificatorii scheletului se schimbă aleatoriu;
- În unele secvențe pentru unele cadre scheletul nu este detectat deloc și lipsește;

Datorită tuturor acestor probleme am decis să proiectăm și să implementăm o arhitectură de tip rețea neuronala care va funcționa direct pe datele de adâncime, sărind peste scheletul Kinect în întregime. Această abordare presupune că rețeaua va putea extrage toate caracteristicile necesare din datele brute astfel încât să poată construi o reprezentare internă pentru secvența de acțiuni.

Pentru a reduce ușor spațiul de căutare și pentru a evita suprasolicitarea din cauza particularităților datelor, am decis, de asemenea, să folosim imaginile de adâncime de fond mascate, care sunt de asemenea furnizate în setul de date. În acest tip de secvențe, în fiecare dintre cadre este disponibilă numai o zonă activă constantă, în timp ce zona inactivă este setată la 0. Zona este determinată o dată pe baza unei uniri a dreptunghiurilor interesante (cu activitate în prim plan) peste toate cadrele.

Abordarea cea mai evidentă este utilizarea tehnicii deja demonstrate a rețelelor neuronale convoluționale (CNN) direct pe datele de intrare. Cu toate acestea, inspirat de SqueezeNet [8], am decis să folosim modulele Fire în cadrul rețelei. Această alegere a fost guvernată de dorința de a utiliza o rețea mai slabă pentru a procesa imaginile de adâncime. Având o întreagă secvență de imagini care trebuie procesate și faptul că ar trebui să putem produce rezultate în timp real a fost un factor important în această alegere, in tinp ce modulele de incendiu au un număr redus de parametri, dar mențin o performanță acceptabilă.

Una dintre arhitecturile testate este prezentată în Figura 10. Precizia obținută pentru această arhitectură specifică a fost de 59,47% față de întregul set de date. Cu toate acestea, această valoare nu este direct comparabilă cu rezultatele pentru abordarea bazată pe schelet, deoarece testarea bazată pe adâncime nu a urmat același protocol de testare. Următoarea noastră dezvoltare include testarea conform protocolului, a unor reglaje de hiperparametre și câteva explorări pe diferite arhitecturi.



Figură 10. Arhitectura rețelei de recunoaștere a activității pe bază de adâncime

7.5.4. Concluzii si Perspective de Viitor

Realizarea noastră s-a axat pe fezabilitatea utilizării capacității de 3D vision in adâncime a lui Pepper, ca o contribuție senzorială pentru metodele de clasificare a recunoașterii activității umane. Am folosit o combinație de rețele neuronale convoluționale și recurente procesate fie ca elemente de intrare de tip schelet, fie ca imagini din adâncime brute. Precizia pe care am obținut-o este satisfăcătoare, chiar dacă nu am ajuns la cele mai înalte standarde. Acest lucru se datorează faptului că numărul de parametri pe care îl folosim este destul de scăzut, ajutându-ne să implementăm un modul bazat pe rețele neuronale pentru Pepper, care rulează aproape în timp real.

8. Abordarea Navigării și Cartografierii

În partea de navigare, obiectivul nostru este de a crea un model care să funcționeze bine pe Pepper, pentru ca robotul să poată oferi îndrumare celor din jur. Pentru a face acest lucru, robotul trebuie să știe cum să navigheze între locația lui și alte locații predefinite. Această problemă poate fi împărțită în cartografiere, localizându-se în mediul înconjurător, urmata de navigarea între diferite locații.

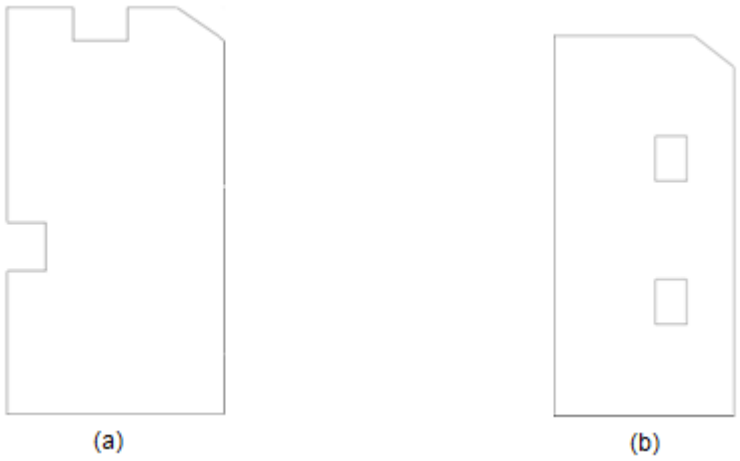
Testele au fost efectuate în două spații de lucru diferite pentru a evalua pachetul de cartografiere din cadrul ROS și din cadrul standard NAOqi. Rezultatele sunt prezentate la sfârșitul acestei cercetări.

Rezultatele preliminare utilizând senzorul Kinect cu Kinect Fusion standard și versiunea îmbunătățită a bibliotecii de pachete Kinfu numită remake Kinfu au fost construite pe secțiuni ale unui laborator la scară mare, pentru a vedea dacă reconstrucția 3D poate fi folosită pentru a forma un model de învățare mai in profunzime pentru navigarea prin robot.

8.1. Abordarea Slam

Metodele clasice SLAM folosesc măsurătorile laser, adâncimea, structura de la mișcare sau elementele de intrare de tip sonar pentru a crea o hartă și a localiza robotul pe baza anumitor repere detectate în timpul procesului. Harta este reprezentată de obicei ca o rețea de ocupare, care modelează mediul ca o matrice de probabilitate, fiecare celulă conținând probabilitatea de a fi ocupată sau nu.

Cadrele existente pentru SLAM au fost evaluate într-un mediu închis. Mediul are o lungime de 8 metri și o lățime de 4 metri. Au fost făcute două teste, unul cu obstacole plasate pe marginea zonei de lucru și unul cu obstacole în mijlocul zonei de lucru, atât in cadrul cartografierii ROS cât și in cadrul NAOqi standard. Primul spațiu de lucru conține două obstacole pe partea laterală a zonei, ca un test standard pentru a vedea calitatea cartografierii colțurilor și pentru a face mai ușor măsurătorile liniilor drepte prin măsurarea distanțelor dintre cutii. Cel de-al doilea spațiu de lucru conține cele două casete din mijlocul spațiului de lucru, pentru a vedea cât de bine se completează cele două cadre în locurile oarbe, unde fasciculele laser nu pot ajunge.



Figură 11. Spatii de lucru cu obstacole pe o parte (a) si in centru (b)

8.1.1. Cadrul de Lucru ROS

Cadrul de lucru ROS include un wrapper peste toți senzorii, ca elemente de intrare și comenzile pentru mutarea lui Pepper. Pachetul de cartografiere (gmapping) a fost utilizat pentru a efectua SLAM. Pachetul preia norul de puncte laser de la cei trei senzori laser ca intrare și

ieșire dintr-o rețea de ocupare. Rezoluția hărții este de 0,05 metri / pixel. De asemenea, pachetul permite setarea intervalului de actualizare a hărții, care a scăzut de la 5 la 1 secunde, pentru a îmbunătăți calitatea rezultatului. Intervalul laserului a fost setat la 3 metri, ceea ce reprezintă intervalul maxim al senzorilor de pe robotul Pepper. Scorul minim pentru a considera rezultatul scanării potrivite suficient de bun pentru a fi adăugat pe hartă a fost mărit la 300 deoarece robotul are probleme majore cu estimarea odometriei. Numărul de particule din filtrul de particule a fost de 100 în timpul rularii primului test și 300 în timpul rularii celui de-al doilea. Pachetul utilizează numai citirile de odometrie și laser pentru a construi o hartă în timp ce robotul este mutat manual în jurul mediului.

Așa cum se poate vedea din figura de mai jos, există unele dezangajări și puncte lipsă. Acestea se datorează erorilor din odometria robotului Pepper, în timp ce punctele lipsă se datorează în principal faptului că fasciculele laser ale robotului nu au putut ajunge în anumite locații. De exemplu, în imaginea de mai jos, în partea dreaptă, marginile cutiilor nu au fost atinse de fasciculele senzorilor laser, prin urmare algoritmul nu a reușit să închidă marginea obstacolelor, dar pentru partea dreaptă a mediului a fost parțial calculat presupunând că peretele este o linie continuă.

Figură 12. Spatii de lucru cu obstacole pe o parte (a) si in centru (b)

8.1.2. Cadrul de Lucru NAOqi

Cadrul de lucru NAOqi oferă diferite interacțiuni cu robotul Pepper printr-un proxy care permite comenzile către robotului. SDK-ul standard oferă metode pentru deplasarea robotului într-un sistem de coordonate cartezian prin trimiterea coordonatelor pe axele X și Y. Coordonatele se referă la poziția inițială a robotului. În același timp, oferă tehnici de evitare a obstacolelor pentru a ajunge la coordonatele solicitate. Versiunea NAOqi 2.5.5 a introdus, de asemenea, metode pentru cartografiere și navigare. Cartografia se face prin solicitarea unei raze de la poziția inițială a robotului, iar robotul va începe să se miște în jurul mediului și va începe procesul de cartografiere. Pachetul nu oferă niciun mijloc de a efectua SLAM în timp ce mutați robotul manual.

După cum se poate vedea din figura de mai jos, rezultatele sunt diferite de rezultate obtinute din rularea pachetului gmapping. Punctele sunt reprezentate mai dens, iar colțurile sunt interpolate, rezultând o figură rotunjită. Acest lucru se datorează faptului că robotul nu a reușit să se apropie de colțuri, deoarece trebuie să păstreze o distanță de siguranță față de toate obstacolele în timpul mișcării sale pentru a efectua cartografierea. Deși prima imagine arată similar cu mediul real, cea de-a doua interpolează între obstacolele centrale și pereții laterali. Pragul se bazează pe distanța de siguranță pe care trebuie să o țină robotul de orice obstacol, deci în partea de jos a imaginii, robotul nu va putea oricum să se miște.

Figură 13. Spatii de lucru cu obstacole pe o parte (a) si in centru (b)

8.2. Recunoașterea Scenelor 3D

Metodele Visual SLAM si cele de reconstrucție 3D cu dispozitive portabile oferă rezultate din ce in ce mai bune astăzi. Construirea unei reconstrucții 3D dense a mediului poate avea mai multe utilizări în navigarea robotului, cum ar fi simulările algoritmilor de rulare. Motivul principal pentru realizarea reconstrucției 3D in această cercetare este crearea unui spațiu virtual de lucru, foarte aproape de mediul real, în care să existe capabilitatea de rulare a algoritmilor vizuali SLAM, care pot fi apoi reglați pentru a lucra în mediul real. Nu este posibilă formarea unei rețele de tip deep reinforcement learning direct pe fluxul primit de la robot datorită faptului că timpul necesar pentru ca robotul sa acționeze la comandă combinat cu timpul necesar rețelei de a se antrena nu poate conduce la nici un rezultat bun, într-o perioadă decentă de timp.

Dispozitivul Kinect oferă o calitate convingătoare de detectare a adâncimii, menținând un cost redus și în timp real a naturii procesului. Acesta utilizează o tehnică de lumină structurată pentru a genera hărți adânci cu valori discrete ale mediului fizic. De fapt, dispozitivul Kinect furnizează de multe ori măsurători de adâncime care fluctuează și numeroase puncte în

mediile fizice fără citiri. Acestea sunt principalele probleme care trebuie depășite pentru a realiza o reconstrucție a modelului 3D al mediului fizic.

Generarea de modele 3D folosind camere de adâncime, cum ar fi dispozitivul Kinect, trebuie făcută prin deducerea geometriei suprafeței de la pe baza de noisy point-based data. Modelul 3D va fi furnizat sub formă de rețea. Norul de puncte generat de Kinect va fi folosit ca noduri care formează modelul 3D. Pentru a crea o rețea trebuie introdusă o metodă de legare a punctelor. O abordare simplă poate fi făcută presupunând că punctele dintr-o mică vecinătate sunt legate și prin urmare vor fi conectate în rețeaua obiectului generat. Acest lucru duce, de obicei, la o reconstrucție incompletă, după cum se vede dintr-un singur punct de vedere.

Biblioteca KinectSDK oferă metode pentru realizarea reconstrucției 3D într-un pachet numit Kinect Fusion. Acest lucru se realizează prin urmărirea continuă a poziției de 6 grade de libertate a camerei și prin fuziunea unor noi puncte de vedere ale scenei într-o reprezentare globală. De asemenea, include metode de segmentare a obiectului de interes din restul mediului fizic. SDK poate fi folosit cu ambele dispozitive Kinect 1, utilizând versiunea 1.8 Kinect SDK și cel mai nou dispozitiv Kinect One utilizând versiunea 2.0 a SDK. Un dezavantaj major este faptul că SDK permite doar realizarea reconstrucției 3D într-o rețea voxel 512 512 512, care poate atinge o suprafață maximă de 3x3 m utilizând cea mai mică rezoluție. Aceasta duce la necesitatea de a efectua o coasere a reconstrucțiilor pentru a avea un spațiu de lucru complet 3D virtual.

Aceste limitări spațiale sunt deja cunoscute, iar pentru a le depăși avem alternativa derivatului KinFu cu o retea voxel în mișcare. Acest lucru, combinat cu optimizarea rafului de poziție și închiderea buclei depășește limitele pachetului Kinect Fusion pentru a realiza o reconstrucție completă a camerei 3D cu dispozitivul Kinect. O versiune ușoară, refăcută și optimizată a Kinfu a fost utilizată pentru a efectua o reconstrucție 3D utilizând senzorul Kinect 1. Aceasta utilizează OpenNI pentru achiziționarea de imagini în profunzime. Performanța sa este de 1,6 ori mai mare decât versiunea PCL a Kinfu.

Pentru realizarea reconstrucției 3D au fost utilizați atât senzorii Kinect 1, cât și senzorii Kinect 2. A doua versiune oferă îmbunătățiri în ceea ce privește rezoluția și calitatea camerei , in ce ace privește adâncimea. Diferențele legate de rezoluția dintre cei doi senzori pot fi văzute în tabelul de mai jos. De asemenea, este important ca Kinect 2 să ofere un câmp vizual orizontal mai mare de 70^0 și un câmp vizual vertical mai mare de 60^0, comparativ cu primul dispozitiv cu 57^0, respectiv 43^0.

	Kinect 1		Kinect 2	
	Resolution	Frame Rate [Hz]	Resolution	Frame Rate [Hz]
color	640x480	30	1920x1080	30
depth	640x480	30	512x424	30
infrared	640x480	30	512x424	30

Tabel 2. Diferențele de rezoluție dintre senzorii Kinect 1 și 2

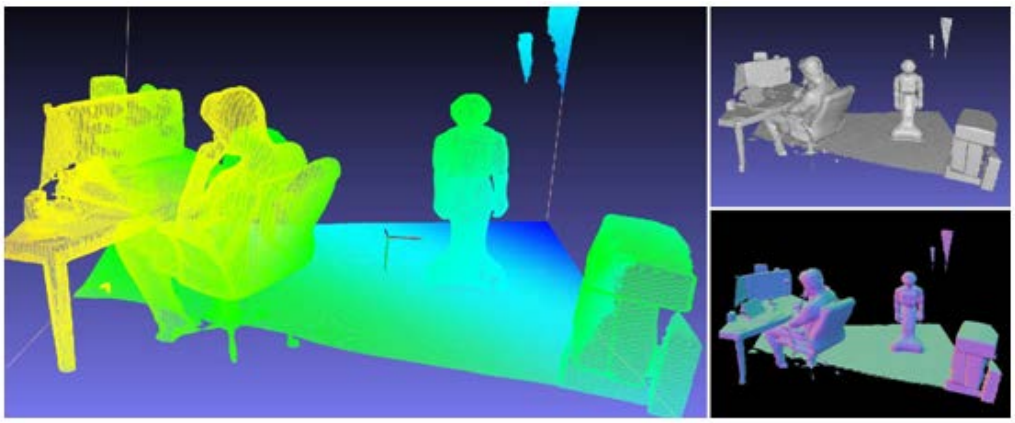
Ambele dispozitive Kinect au fost utilizate pentru a compara diferențele dintre rezultatele bibliotecii Kinfu și ale Kinect Fusion din SDK standard Kinect. Rezultatele sunt foarte similare din punct de vedere al preciziei, dar cel de-al doilea dispozitiv oferă o modalitate mai bună de umplere a petelor goale în reconstrucție.

8.2.1. Rezultatele Kinfu

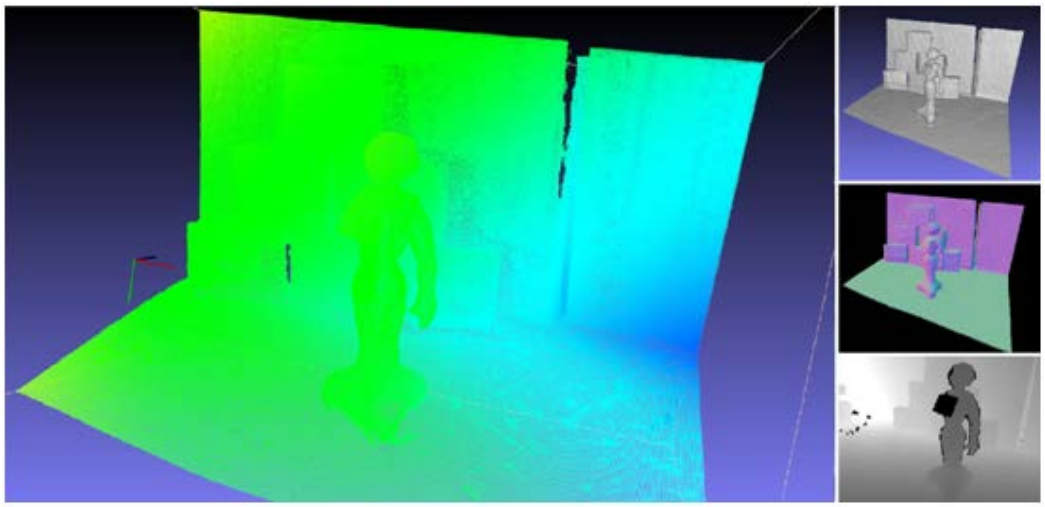
Biblioteca Kinfu Remake folosește dispozitivul Kinect 1 pentru a realiza reconstrucția 3D, utilizând elementul de intrare de adâncime a senzorului. Pepper a fost plasat în mijlocul spațiului de lucru și s-a efectuat o rotație de 360 de grade în jurul acestuia. În fundal, au fost plasate mai multe cutii pentru a vedea mai bine calitatea reconstrucției.

În figurile de mai jos, în partea stângă, se poate observa discretizarea finală reconstruită. Discretizarea este foarte densă, iar numărul de vârfuri poate fi redus, deoarece podeaua poate fi modelată ca un singur plan. În a doua imagine există pe peretele din spatele robotului niște pete negre. Robotul nu a fost complet reconstruit datorită reflexiei din materialul acestuia. Acest lucru poate fi mai bine văzut în ultima imagine, în care mai multe puncte de vedere din reconstrucția lui Pepper arată că partea robotului nu este perfect reconstituită. Mai mult decât atât, dispozitivul Kinect poate captura imagini cu adâncimea de peste 50 cm, de aceea unele părți ale încăperii nu vor fi reconstruite, cum ar fi partea din spate a robotului, deoarece distanța dintre perete și robot este mai mică decât limita de 50 cm. Din fericire, pentru sarcina la îndemână, robotul nu va putea naviga între spații înguste ca acesta, deci nu va impune restricții.

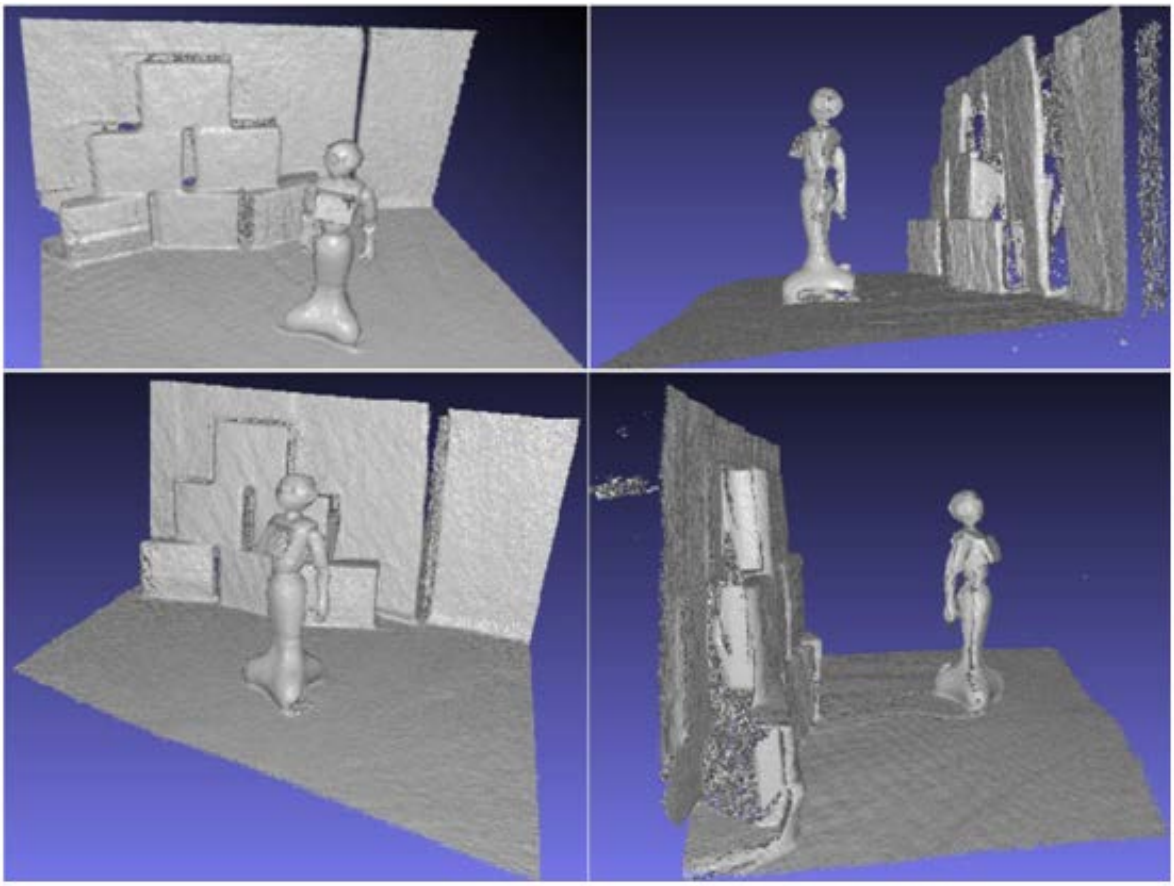
In același timp, datorită reflexiilor senzorului de adâncime cauzate de robot, reconstrucția trebuia făcută de mai multe ori.



Figură 14. Reteaua reconstruita Kinfu 1



Figură 15. Reteaua reconstruita Kinfu 2

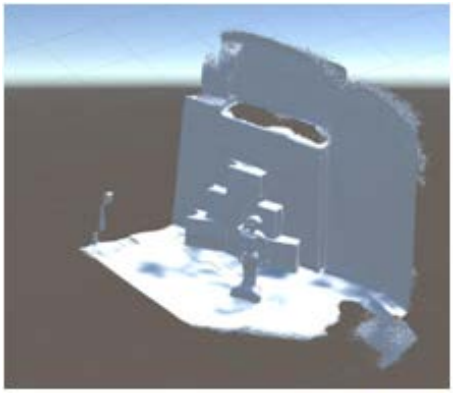


Figură 16. Reteaua reconstruita Kinfu 3

8.2.2. Rezultatele Fuziunii Kinect

Același scenariu de cameră prezentat în imaginile de mai sus, realizat cu biblioteca Kinfu Remake, a fost realizat cu una Kinect Fusion standard, utilizând dispozitivul Kinect 2. Deși detaliile reconstrucțiilor sunt în general mai bune decât utilizarea Kinect 1, așa cum se poate

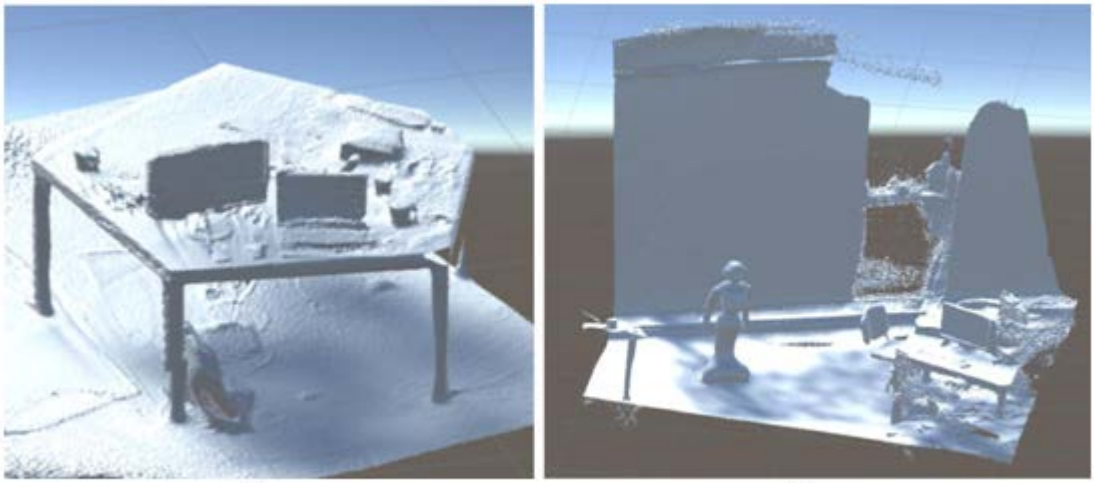
vedea în figura de mai jos, cu Kinfu, sistemul are aceleași limitări în ceea ce privește distanța minimă față de subiect și reflecții.



Figură 17. Rezultatele fuziunii kinetice

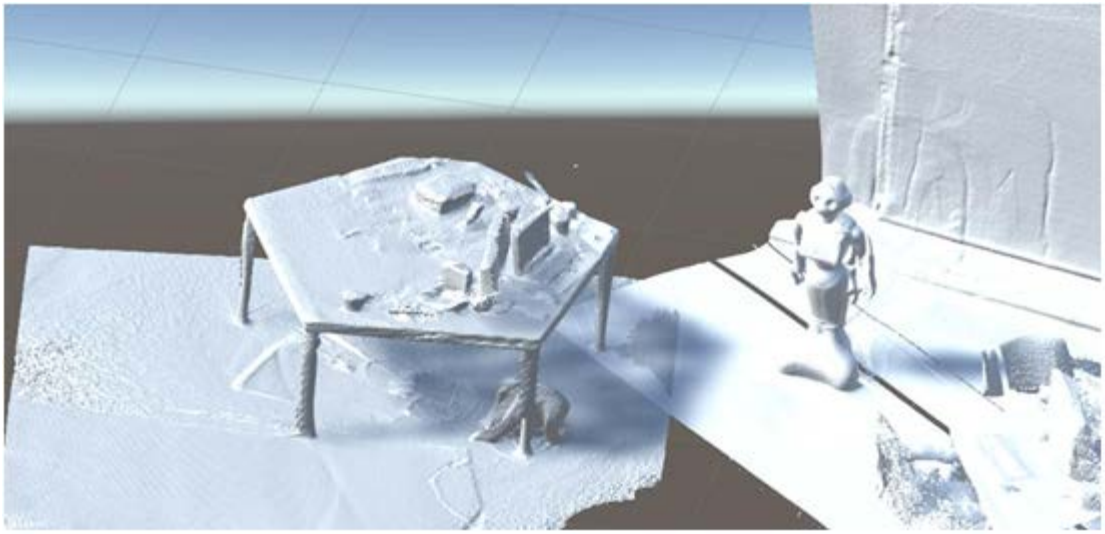
În figura de mai jos, în partea dreaptă (b), fereastra nu este detectată deoarece imaginea de profunzime este puternic influențată de reflexiile din fereastră.

În figura de mai jos (c) cele două ochiuri sunt conectate pentru a arăta că este posibilă o reconstrucție 3D a unei încăperi pe scară largă folosind reconstrucții multiple de 3x3x3 metri.



(a) (b)

Figură 18. Reconstructie birou (a) si reflexia in fereastra (b)



(c)

Figură 19. Modele legate (c).

9. Comportamente Implementate

Această secțiune descrie comportamentele implementate pe robot, începând cu cele de bază și urmate de cele mai complexe.

9.1. Comportamente de Baza

Așa cum am spus anterior, am avut nevoie de niște comportamente de bază de a începe cu scopul de a crea cele mai complexe. Toate comportamentele prezentate în această secțiune utilizează cadrul NAOqi.

9.1.1. Recunoașterea Persoanei

Comportamentul de recunoaștere a persoanei este un modul care încearcă să recunoască persoana din fața sa. Acesta este activat de comanda vocală "Cine sunt eu?". Pentru a ști că acest comportament a început, am adăugat o fraza suplimentară înainte de recunoaștere: "Lasă-mă să ma gândesc". După aceea, robotul va căuta în baza sa de date și după ce se va găsi un rezultat, acesta va spune numele persoanei dacă este cunoscut. Dacă există mai multe potriviri, robotul va spune toate numele potrivite. Dacă nu există nici un nume potrivit, acesta va răspunde că nu cunoaște persoana.
Acest comportament este implementat pe baza modulului de recunoaștere a feței din cadrul NAOqi.

9.1.2. Învățarea Feței

Acest modul este util pentru a adăuga o persoană in baza de date a robotului. Aceasta este declanșată de comanda "Remember me". După aceea, utilizatorul trebuie să-și spună

numele și să stea în fața robotului timp de câteva secunde, pentru a permite fotografierii clare ale feței. Utilizează faza de învățare a modulului de recunoaștere a feței de la NAOqi.

9.1.3. Comenzile Mișcării

Am adăugat un comportament simplu pentru mișcare, astfel încât robotul să poată fi mutat pe baza comenzilor vocale. Are 4 opțiuni: "Deplasare înainte", "Deplasare înapoi", "Deplasare în stânga" și "Deplasare la dreapta". Robotul se va deplasa în direcția specificată cu un metru și numai dacă nu există niciun obstacol. Acesta este implementat utilizând modulul de navigație de la NAOqi.

9.1.4. Redarea Muzicii

Am implementat acest comportament în partea de cercetare a proiectului nostru, când am încercat să înțelegem cum funcționează robotul. Aceasta implică mai multe module, cum ar fi recunoașterea vorbirii, animația și redarea audio. Este activată de o comandă vocală "Redă muzică". După aceasta, utilizatorul poate alege între 3 tipuri diferite: "disco", "rock" și "thai". Robotul va reda o melodie bazată pe opțiunea selectată și va începe să danseze în consecință.

9.1.5. Realizarea de Fotografii

Un al test comportamental este Acela de realizare a fotografiilor. In cadrul acestuia, am învățat cum sa folosim tableta atașata robotului sau senzorii acestuia. Comportamentul se activează prin rostirea “Realizează o poza” “Take a picture”, iar pentru aceasta utilizatorul trebuie sa atinga mana robotului. Cand robotul sesizeaza atingerea, va face poza si o va afisa pe tablet.

9.2. Urmărirea Persoanei

Un comportament foarte util este cel de urmare a unei persoane. Acest comportament are mai multe cazuri de utilizare. Un exemplu ar putea fi situația în care robotul este situat într-un magazin și un client dorește să știe câteva detalii despre un obiect pe care nu-l cunoaște. Deci, clientul va spune robotului să-l urmeze până la destinație.
Acest comportament utilizează modulele de recunoaștere vocală, detective a persoanei și urmărirea persoanelor de la NAOqi API. Este activat de comanda vocală "Urmeaza-mă". Pentru a ști că robotul a început comportamentul potrivit, acesta va spune "OK" sau "Da maestre", după care va începe urmărirea persoanei. Comportamentul este oprit când utilizatorul va spune "Stop".

Problemele pe care acest comportament le are au legătură cu urmărirea. Deoarece elemental de urmarire din cadrul NAOqi este foarte sensibil la schimbările bruște în ceea ce privește poziția luminii și a persoanei, poate pierde persoana pe care o urmărește. De aceea, pentru a avea o bună funcționare a comportamentului, utilizatorul ar trebui să încerce să se miște mai încet decât de obicei, astfel încât modulul de urmărire să poată înțelege că acea persoana este cea care trebuie urmarita.

9.3. Ghidajul Reperelor Locale

Acest comportament permite robotului să asiste o persoană pana la locația unui punct marcat în apropierea vecinătății și să ofere informații suplimentare despre acesta în funcție de un comportament predeterminat. Acest scenariu este util pentru a oferi orientare printr-un spațiu, cu o precizie bună, numai prin detectarea imaginii vizuale 2D.

Obiectele sau punctele dorite pot fi etichetate cu ușurință prin atașarea unei pictograme imprimate predefinite cu modele specifice numite "repere" (vezi Figura 1). Aceste etichete pot fi generate prin sistemele care oferă un cod unic de identificare. Acest lucru poate fi plasat în locații diferite în domeniul de acțiune al robotului. În funcție de punctul de reper detectat de catre robot, acesta poate obține informații despre locația și obiectul identificat de către robot. De asemenea, acest sistem, împreună cu alte informații senzoriale, poate ajuta la crearea unui modul de localizare mai solid.

Printr-o bază de date predefinită pe robot, reperele pot fi corelate cu orice fel de informații textuale. Folosind aceste informații, comportamentul este capabil, la sosirea la țintă, să se reangajeze cu persoana care a inițiat comportamentul și să-i prezinte faptele stocate. Această bază de date poate fi ușor întreținută și extinsă.

Procesul intern al comportamentelor urmează schema prezentată mai jos în figura 2. Mai întâi, comportamentul poate fi declanșat prin vorbire prin solicitări de la o persoană angajată, potrivirea unor propoziții precum "Arata-mi ...", "Du-mă la ...". În continuare, robotul va iniția un proces obiectiv de confirmare. Robotul poate obține fie intenția specifică prin vorbire, fie va ghida utilizatorul pentru a selecta ținta dorită prin tableta atașată. Această opțiune va deschide o aplicație pe ecran prezentând opțiunile posibile. Acest lucru oferă un avantaj dublu: reprezentarea mai bună a utilizatorului pentru toate opțiunile disponibile și o selecție mai precisă (de exemplu: comenzile vocale sunt mai greu de interpretat în medii zgomotoase, atunci când există un context mic, când intenția este reprezentată de un nume).

După stabilirea țintei dorite, robotul va începe să caute prin spațiul din jur, făcând rotații complete de 360^0 și scanând și într-un interval vertical de +/- 30^0. Atunci când obiectivul a fost detectat vizual, folosind algoritmul de detectare a punctului de referință, robotul inițiază o procedură de navigare pentru a ajunge cat mai aproape de distanta de 0,5 metri de acesta. În funcție de informațiile și procedurile stocate pentru acel identificator, robotul va indica spre ținta și va decupla după ce a spus: "Aici este <numele țintă dorit>" sau urmează următorul comportament. Acest comportament va încerca să se reintegreze vizual cu inițiatorul și, dacă reușește, va prezenta informațiile stocate din baza de date.

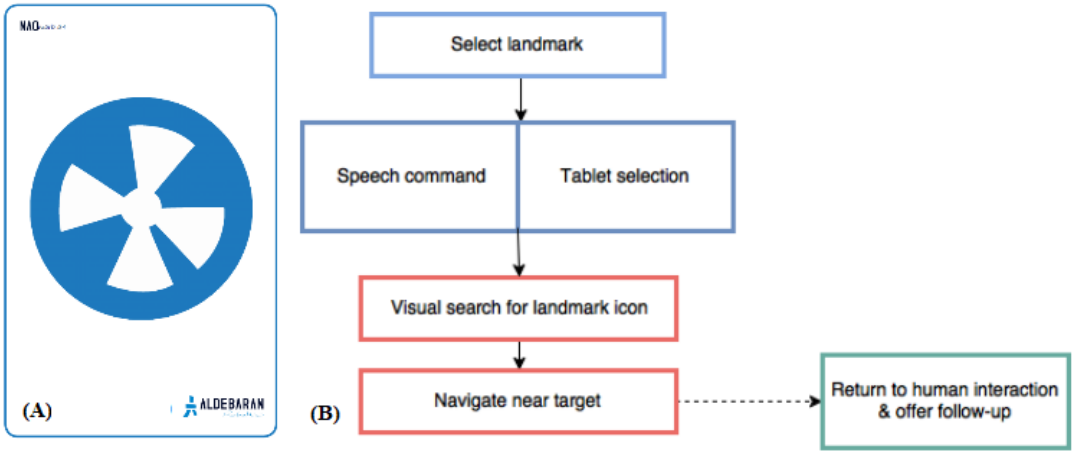


Figura 1. Exemplu de iconita reper (a) Argitectura de ghidare a reperului local (b).

Performanțe și Limitări

Punctele de reper constau în cercuri negre, cu palete triunghiulare albe centrate în mijlocul cercului. Locația specifică a diferitelor palete ai triunghiului este folosită pentru a distinge un punct de reper de celelalte. Detectarea reperelor a fost testată în condiții de iluminare de birou - adică sub 100 lux, pana la 500 lux.

Deoarece detectarea în sine se bazează pe diferențe de contrast, ar trebui să se comporte bine atât timp cât semnele din imaginile de intrare sunt în mod rezonabil contrastate. Software-ul de îmbunătățire a detectării poate fi obținut prin activarea auto-gain al camerei sau prin reglarea contrastului prin intermediul interfeței monitor. Posibilitatea de detectare a acestor repere este între 2^0 și 23^0, ceea ce corespunde cu 14-160 de pixeli în imaginea QVGA. De asemenea, există un interval al unghiului de înclinare de +/- 60^0 din direcția ortogonală a camerei. Un alt avantaj este că această detectare este invariantă de rotație, astfel încât etichetele să poată fi plasate cu ușurință.

Evoluții viitoare. Acest comportament poate fi extins cu o hartă de localizare a unei zone mai largi folosind algoritmi de localizare simultană și de cartografiere. Acest lucru va necesita doar o procedură care precede căutarea vizuală. De asemenea, îmbunătățirea bazei de date textuale cu posibilitatea de a face referire la alte canale de comportament ar oferi un alt nivel de complexitate.

9.4. Identificarea Utilizatorului

Identificarea utilizatorului este un comportament util pentru situația în care robotul va rămâne într-o singură poziție și va fi într-o stare pasivă care așteaptă sa fie abordat de clienți. Abordarea propusă pentru această problemă este identificarea difuzorului folosind un detector uman combinat cu un detector de fețe. Informația despre distanța față de persoană și unghiul capului față de robot este folosită pentru a înțelege dacă persoana este orientată către robot. Dacă există mai multe persoane orientate spre robot, folosim un mecanism de postprocesare pentru a determina care dintre ele este cea mai apropiata sau care se adresează direct robotului. Mecanismele de postprocesare care pot fi utilizate în această situație se pot baza pe mișcarea buzelor, pe direcția sunetului, pe focalizarea ochilor etc. Pentru moment, folosim

distanța față de persoana detectată. Cu toate acestea, aceasta nu este cea mai bună opțiune, deoarece senzorul 3D al robotului nu este foarte precis.

Arhitectura pentru scenariul propus este reprezentată de următoarea imagine:

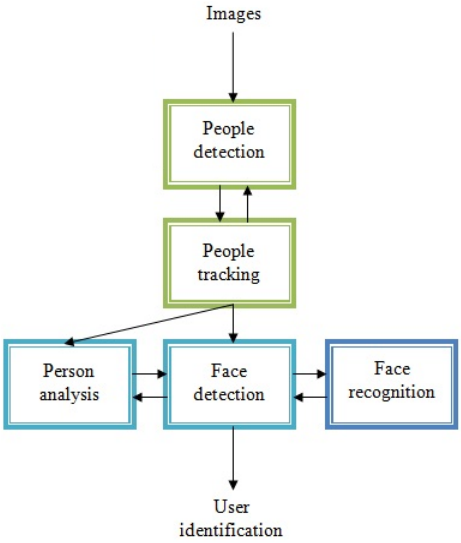


Figura 2. Arhitectura de identificare a utilizatorului

Aplicația este declanșată atunci când o persoană este detectată în câmpul vizual al robotului. Declanșarea se face cu ajutorul modulului de detectare a persoanelor. Când sunt detectate una sau mai multe persoane, robotul începe să le analizeze, verificând dacă persoanele se uită la el. Această parte implică o combinație între modulele de detecție a persoanelor și modulele de detectare a feței, pentru a afla dacă o față este detectată pentru o anumită persoană detectată. Am analizat mai multe scenarii, după cum urmează:

- Dacă există o singură persoană detectată și persoana se uită la robot timp de mai mult de 5 secunde, atunci robotul va ieși din starea pasivă și se va apropia de persoana întrebând: "Bună ziua! Pot sa te ajut?". Persoana poate respinge interacțiunea, caz în care robotul va reveni la starea pasivă.
- În cazul în care există mai multe persoane în imagine, robotul va analiza distanțele de la acele persoane la robot.
    - Dacă există doar o singura persoană mai apropiată de robot și care se uită la acesta timp de mai mult de 5 secunde, atunci consideră că poate acea persoană dorește să interacționeze cu ea și se apropie de el cu aceeași întrebare ca în cazul precedent.
    - Un alt caz, care este foarte probabil să apară, este atunci când un grup de oameni stau în fața robotului și toți, sau cel puțin o parte dintre ei, se uită la robot. În acest caz, deoarece robotul nu știe pe cine să se concentreze, va cere unei persoane să se apropie mai mult de ea dacă vrea să interacționeze cu acesta.
    - Similar cazului precedent este cazul în care mai mulți oameni stau în fața robotului, dar numai unul privește spre robot. În acest caz, robotul se va

concentra pe acesta. Acest caz este declanșat numai atunci când nu sunt îndeplinite condițiile prealabile unui alt caz.

- Când o persoană este foarte aproape de robot, la o distanță mai mică decât un prag specific, caz în care robotul nu va verifica dacă persoana îl privește, dar îl va întreba direct dacă vrea să interacționeze cu el. Am considerat acest caz, deoarece o persoană poate rămâne prea aproape din greșeală, în timp ce oamenii care se află în spatele lui ar putea dori să interacționeze cu robotul.

Detectarea persoanelor din fața robotului se face folosind modulul de detectare a persoanelor și detectare a feței. Modulul de recunoaștere a feței este folosit atunci când robotul abordează o persoană. Dacă robotul cunoaște persoana cu care se adresează acesteia, schimbând formula de adresare si introducând numele persoanei. De exemplu: "Bună ziua, Ștefania! Pot sa te ajut?".

Pentru a implementa acest comportament, am folosit numai funcționalitățile încorporate disponibile în cadrul NAOqi, deoarece oferă informații suplimentare, cum ar fi rotația capului, direcția privirii, ceea ce este foarte util pentru a înțelege dacă persoana se uită la robot sau nu.
Pentru a evalua corectitudinea acestui comportament, am numărat de câte ori robotul a gestionat corect scenariile menționate anterior. Rezultatele sunt prezentate în tabelul de mai jos.

<table>
<tr><td>Doar o persoana in fata robotului</td><td colspan="3">Mai multe persoane in fata robotului</td><td>Persoana prea aproape de robot</td></tr>
<tr><td rowspan="2">20 din 20</td><td>O persoana mai aproape fata de restul</td><td>Aceeași distanta</td><td>Aceeași distanță și o singură persoană care privește spre robot</td><td rowspan="2">10 din 20</td></tr>
<tr><td>16 din 20</td><td>11 din 20</td><td>8 din 20</td></tr>
</table>

Tabel 3. Rezultatele testului de identificare a utilizatorului

Cele mai bune rezultate se obțin în cazul în care în imagine există doar o singură persoană, care se comportă corect de fiecare dată. Rezultatele pentru scenariul în care o persoană este mai aproape decât restul nu sunt foarte clare definite.

Deoarece chiar și atunci când există două persoane în imagine și una dintre ele este mai aproape, robotul poate pierde persoana din spate și poate trata situația așa cum o face în scenariul cu o singură persoană. Cu toate acestea, se comportă așa cum se presupune, așa că am inclus ca rezultate corecte și acele cazuri. Rezultatele pentru scenariul în care oamenii din fața robotului se află la aceeași distanță și doar o persoană care privește spre robot nu este atât de bună, deoarece acest scenariu este strâns legat de "aceeași distanță". Analiza feței robotului durează mai mult și va merge mai repede la cazul "Aceleași distanțe". În plus, rezultatele pentru "Persoana prea apropiată de robot" sunt în jur de 50%, deoarece poate pierde persoana când

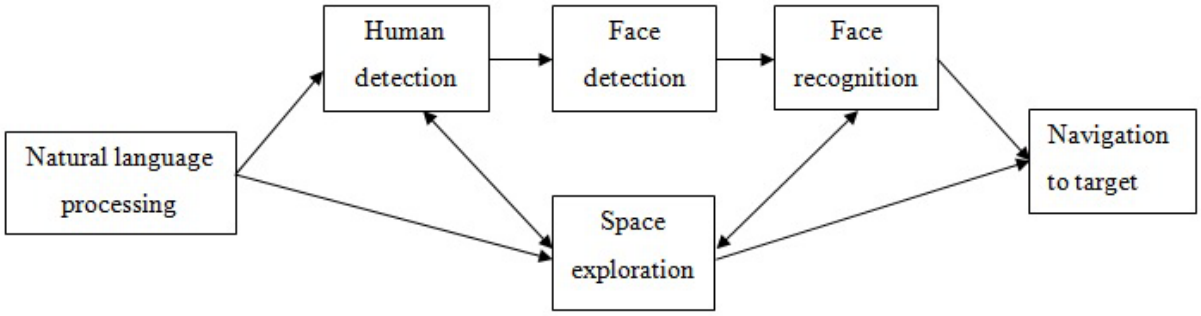
acesta este prea aproape de ea. Acest lucru se întâmplă deoarece, dacă persoana este prea apropiată, robotul nu vede întreaga persoană și nu o poate înțelege ca persoană. Circumstanțele în care acest caz poate funcționa sunt atunci când robotul urmărește persoana în avans.

9.5. Căutarea și Interacțiunea cu o Persoană

Un alt comportament pe care îl considerăm important pentru proiectul nostru este abilitatea robotului de a căuta o persoană. În acest context, am implementat o metodă care permite robotului să navigheze aliator prin mediul înconjurător până când detectează persoana dorită. Comportamentul este activat de comanda vocală "Căuta pe cineva". După ce robotul aude numele persoanei, începe să o caute până când o găsește.

9.5.1. Descriere Arhitecturală și Implementare

Arhitectura pe care am considerat-o pentru a rezolva căutarea persoanei cu ajutorul explorării spațiale implică mai multe module pentru diferite probleme. Folosim un modul de procesare a limbajului natural pentru identificarea numelui persoanei. După aceasta, folosim în paralel un modul responsabil cu navigația și un modul de detectare a persoanei, a feței și de recunoaștere a feței. După ce un om este recunoscut, folosim un alt modul de navigație pentru ca robotul să ajungă la destinație. Arhitectura este prezentată în schema de mai jos:



Figură 20. Arhitectura de cautare si interactiune

Pentru modulul de prelucrare a limbajului natural am folosit funcționalitatea incorporata a lui Pepper pentru recunoașterea vorbirii. Funcționalitatea încorporată are limitări, deoarece este sensibilă când există prea mult zgomot sau când există mulți oameni care vorbesc simultan. Această limitare este o problemă obișnuită atunci când vorbim de platforme robotizate și depinde de capacitățile hardware ale robotului. Pentru scenariul nostru, funcționalitatea încorporată funcționează bine și o putem îmbunătăți în viitor prin tehnici mai avansate dacă avem nevoie de ele.

Partea principală a arhitecturii constă în două module mari: detectare-recunoaștere umana și navigare.

Modulul de detectare-recunoaștere umană implică mai multe sub-module, așa cum se poate vedea în figură. Acesta va realiza imagini 2D sau 3D ca elemente de intrare, în funcție de natura algoritmului, și va genera informații despre locația oamenilor în imagini. Algoritmul va detecta persoane sau grupuri de persoane; în cazul detectării grupurilor de persoane trebuie să

existe un mecanism suplimentar pentru a le împărți. Rezultatele trebuie să contina informații suplimentare despre poziție față de robot, pentru a ști care dintre persoanele detectate este relevanta în ceea ce privește identificarea de către utilizator.

Această parte poate fi văzută ca un canal de tehnici de tip computer vision. Este necesar mai întâi un algoritm de detectare umană pentru a identifica dacă există sau nu persoane în câmpul vizual al robotului. Dacă există oameni în imagini, atunci trebuie să aplicăm câteva tehnici mai avansate pentru analiza umană. Pentru a face acest lucru, avem nevoie de o reprezentare mai exactă a omului, așa că am ales să analizăm fețele. Prin urmare, dacă există persoane detectate în imagine, extragem fețele folosind niște algoritmi de detectare a feței. Pentru fiecare față identificată în imagine, aplicăm un mecanism pentru a prezice dacă robotul cunoaște fața sau nu. Dacă se întâmplă acest lucru, el transmite trecerea la modulul următor. Pentru modulul de tip computer vision, am folosit funcțiile încorporate din NAOqi, deoarece acestea sunt compatibile cu modulul de navigație.

În paralel cu modulul de recunoaștere-detectare umană există modulul pentru explorarea spațiului, care permite robotului să exploreze propriul mediu.

Pentru partea de navigare am testat implementări NAOqi multiple. Dacă nu există nici o activare vocală pentru robot, acesta ar explora mediul în căutarea oamenilor și va stoca ultima locație cunoscută, în timp ce cartografiază mediul. Robotul are funcționalități integrate pentru explorarea mediului, crearea unei rețele de ocupare și localizarea pe hartă. Aceste funcționalități sunt o versiune beta, însă au oferit rezultate bune pentru scenariul de față. Dacă apare declanșatorul și robotul nu are cunoștințe anterioare despre ultima poziție cunoscută sau daca persoana s-a mutat din locația cunoscută, robotul își va mișca capul lateral și se va roti pentru a încerca să găsească persoana respectivă. Dacă nu reușește acest lucru, explorarea este rulată din nou pentru a găsi persoana respectivă. Dacă acest lucru nu reușește, robotul consideră că persoana a părăsit mediul înconjurător și utilizează textul pentru a comunica acest lucru.

Ultima parte a sistemului este responsabilă cu mișcarea robotului spre anumită locație din mediu. Atunci când o persoană este identificată, fie printr-o cunoaștere prealabilă a locației sale, fie prin descoperirea în timpul fazei de explorare, robotul îi obține poziția de la modulul de identificare a persoanei și se deplasează la acesta. Navigarea oferă mijloace pentru a evita obstacolele, dar modulul are nevoie de îmbunătățiri pentru a fi utilizat într-un scenariu real.

Figură 21. Test navigatie

9.5.2. Rezultate

Rezultatele pe care le-am obținut folosind sistemul prezentat sunt satisfăcătoare pe ambele părți, detectarea-recunoașterea umană și navigația. Ca măsurătoare de precizie am numărat de câte ori robotul a detectat corect persoana pe care o căuta și a ajuns la destinația corectă. Pentru a face acest lucru, am improvizat o mică incintă pentru robot, după cum puteți vedea în imaginea de mai jos, pentru a ști exact care este mediul robotului și pentru a ști dacă acesta se mișcă corect prin el. Am numărat un număr total de 22 de mișcări corecte ale robotului de la 30 de încercări, cu 2 persoane înregistrate în memoria sa.

Considerând sistemul ca o compoziție de module independente, putem prezenta rezultatele pe recunoașterea și cartografierea mediului. Pentru cartografierea mediului am folosit noile caracteristici adăugate în NAOqi legate de performanța SLAM. Apoi am localizat cele două persoane pe hartă și am măsurat rezultatele. De asemenea, am rulat simultan modulul de cartografiere în ROS pentru a avea o linie de bază a cartografierii.

Rezultatele privind r detectarea-recunoașterea umană sunt influențate de numărul de persoane din imagine, de pozițiile lor, de condițiile de iluminare și de distanța dintre persoană și robot.

Modulul de detectare umană este important, deoarece îl folosim pentru a declanșa detectarea feței. Începem detectarea și numai după detectarea unui om se ia în considerare verificarea fețelor din imagine. Având în vedere testele pe care le-am făcut și numărul de persoane din imagini, eroarea globală a detectării se află în jurul unei persoane, care lipsește de cele mai multe ori. Această eroare este aceeași pentru modulul de detectare a feței, pentru o medie a datilor cand aceasta fata lipseste. Totuși, ceea ce am observat este că fețele sunt detectate mai rapid și că urmărirea este mai solidă. Pentru rezultate bune de recunoaștere a

fetei sunt necesare condiții bune in care se face aceasta recunoastere, totuși rezultatele sunt satisfăcătoare pentru cazul nostru de utilizare.

Rezultatele navigației în timpul fazei de cartografiere cu NAOqi API pot fi văzute în următoarele figuri:

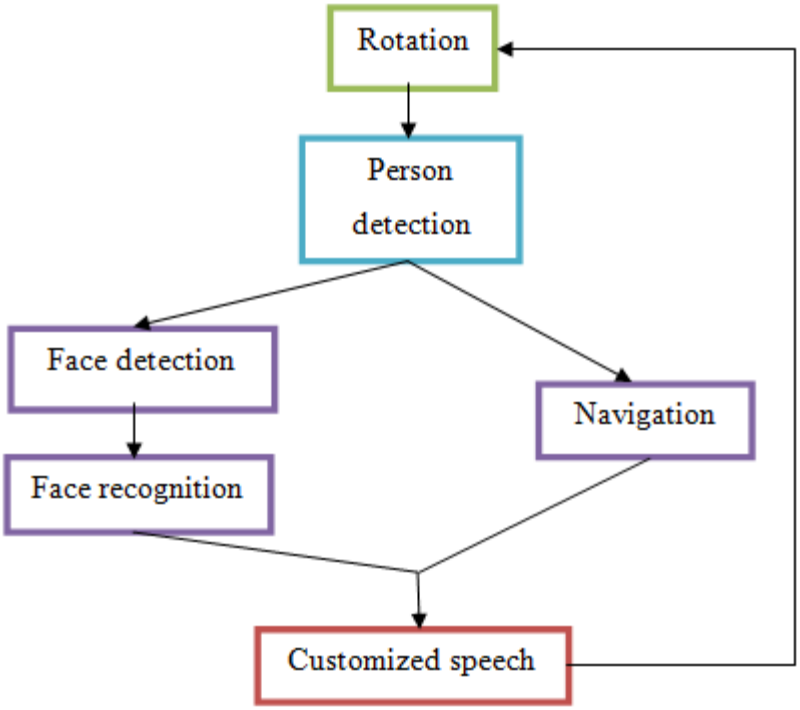
Figură 22. Rezultatele navigarii

Rezultatele cartografierii arată că API-ul NAOqi poate rezolva problema de cartografiere, dar API nu poate localiza corect robotul pe hartă. Acest lucru se datorează unei erori mari în măsurătorile senzorilor. Luăm în considerare îmbunătățirea acestui modul, încercând să corectăm eroarea senzorilor în funcție de odometrie.

9.6. Căutarea Persoanei Ghidate

Căutarea persoanei ghidata este o alternativă mai solida pentru metoda anterioară. Robotul nu navighează aleatoriu prin mediul înconjurător, ci este ghidat de locația persoanelor detectate în cameră.

Arhitectura este compusă din mai multe submodule, pentru navigație, mișcare, vorbire și vizualizare. Aceasta este prezentat în figura de mai jos:



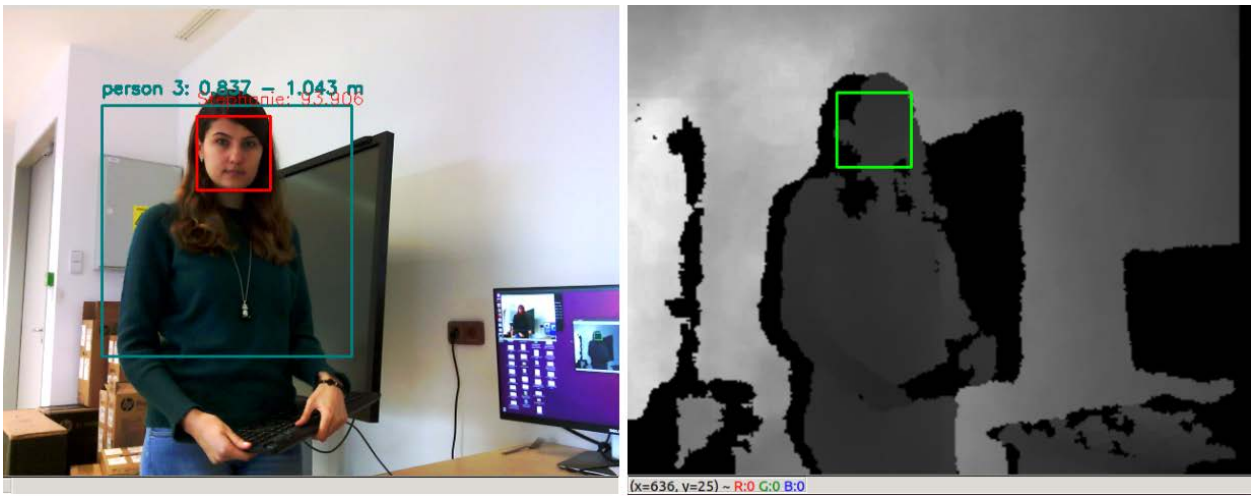
Figură 23. Arhitectura de cautare persoana ghidata

Robotul este situat într-o poziție inițială și începe rotația, mai întâi cu capul și apoi cu întregul corp. Mișcarea capului este de la dreapta sus la stânga sus, apoi de la stânga jos la dreapta. Am ales o astfel de mișcare pentru a inspecta toate locațiile posibile în poziția actuală a robotului. Dacă nu există nici o persoană detectată, atunci se va roti cu întregul corp cu 120 de grade. Când se detectează o persoană, mișcările capului și ale corpului sunt oprite, iar modulele următoare sunt activate, mai precis, recunoașterea navigației și a feței, care se vor desfășura în paralel. Robotul va urmări persoana detectată și va merge pana la o anumita distanta de aceasta. Dacă robotul a atins ținta, dar persoana nu este recunoscută, îi va spune persoanei să privească direct la el pentru a o identifica. Dacă persoana este recunoscută, Pepper îi va spune informațiile dorite, de exemplu unele notificări sau mementouri. Dacă nu, va continua căutarea. Dacă nu detectăm nimic în 3 rotații, ne oprim din comportament, deoarece considerăm că am acoperit toate posibilitățile. Continuarea după detectarea unei persoane și înlocuirea și îmbunătățirea criteriilor de oprire.

Am folosit rețeaua YOLO2 pentru detecția umană, deoarece oferă detectări foarte bune chiar dacă oamenii sunt situați la o distanță mai mare și am combinat rezultatele oferite de YOLO cu rezultatele oferite de senzorul 3D pentru a obține distanțele. Pentru navigare am folosit funcționalitatea încorporată de la NAOqi, în timp ce pentru detectarea și recunoașterea feței am folosit cascadele Haar din biblioteca OpenCV.

Rezultatele acestui comportament depind de navigația și urmărirea persoanei, combinate cu senzorul 3D. După cum am explicat anterior, navigația are unele erori, pe care nu le putem controla, cauza fiind senzorii. Senzorul 3D are, de asemenea, unele probleme, deoarece nu returnează întotdeauna distanța potrivită pana la persoana. Urmărirea este

următorul nostru modul de îmbunătățire și sperăm să obținem rezultate mai bune după aceasta. Detectarea și recunoașterea persoanei funcționează foarte bine, oferind rezultate consecvente. Un exemplu de modul în care robotul percepe și analizează imaginile este în următoarea imagine:



Figură 24. Perceptia robotului si analiza unui exemplu de imagine

Dacă consideram (ca distanta in metri) de cate ori Pepper ajunge la o persoană, în condiții identice, am spune că acuratețea este de aproximativ 80% (16 ture din 20). Acest rezultat a fost obținut, experimentând cu o persoană stand pe un scaun, fără obstacole în fața acesteia. Nu putem încă să evităm obstacolele deoarece nu exista o reprezentare a mediului, deci dacă robotul detectează o persoană care se află în spatele unui obstacol, nu va reuși să ajungă la destinație. Acesta este un pas încă nedepășit pana la aceasta data, folosind doar capabilitățile robotului. Considerăm utilizarea dispozitivelor externe, Kinect, Lidar, pentru a crea o hartă a mediului, pe care o vom încărca pe robot.

10. Stagii masteranzi

Pe parcursul acestei etape, s-au realizat stagii de practica efectuate de 3 masteranzi la agentul economic, Centrul IT pentru Știință și Tehnologie SRL (CITST), un IMM orientat spre cercetare și dezvoltare de produse inovatoare.
Stagiile de practică s-au desfășurat precum urmează.

Alexandra Ghită, actual studentă la masterat în anul II la masteratul Artificial Intelligence din UPB, a desfășurat un stagiu în perioada Mai – Iunie 2017. În timpul stagiului s-a familiarizat cu mediul de afaceri și cu infrastructura disponibilă la agentul economic și a efectuat teste și dezvoltări legate de identificarea persoanelor de către robot.

Alexandru Gavril, actual student la masterat în anul II la masteratul Artificial Intelligence din UPB, a desfășurat un stagiu în perioada Iulie – August 2017. În timpul stagiului s-a familiarizat

cu mediul de afaceri și cu infrastructura disponibilă la agentul economic și a efectuat teste și dezvoltări legate de navigarea robotului (SLAM).

Mihai Nan, actual student la masterat în anul I la masteratul Artificial Intelligence din UPB, a desfășurat un stagiu în perioada Iulie – August 2017, deci pe perioada verii duă susținerea examenului de licentă. În timpul stagiului s-a familiarizat cu mediul de afaceri și cu infrastructura disponibilă la agentul economic și a efectuat teste și dezvoltări legate de recunoașterea activităților utilizatorului.

In anexele la acest raport se găsesc atestatele stagiilor de practică efectuate în compania CITS.

11. Concluzii

Proiectul SPARC are ca scop proiectarea și implementarea unei platforme care să permită o flexibilitate sporită și o ușurință în definirea comportamentelor specifice clienților pentru roboți de asistenta.

Abordarea se bazează pe conceptul de programare la nivel de obiectiv. Această abordare ajută dezvoltatorul să determine cu ușurință acțiunile unui robot folosind compoziția comportamentelor de bază, de exemplu, detectarea unui utilizator, răspunsul la interogarea utilizatorului, urmărirea utilizatorului, mutarea în locație, detectarea obiectului, indicarea unui obiect.

Prima parte a proiectului a constat în colectarea de informații despre roboții Tiago și Pepper și despre modul în care putem implementa caracteristicile dorite. În urma analizei am hotarât să orientăm eforturile către robotul Pepper și să dezvoltăm un set de module și comportamente de bază proprii care să poată fi portat și pe alte tipuri de roboți.

În etapa curentă am dezvoltat arhitectura platformei de comportamente, respectiv de gestiune pentru un set de comportamente “de bază”, am realizat un modul de interacțiune vocală în limba engleză pentru ca în viitoarea etapă să îl adaptăm pentru limba română, am dezvoltat un set de module de computer vision pentru detectarea și recunoașterea persoanei, răspunzând și provocării detectării persoanei care se adresează robotului într-un grup de persoane. Detecția persoanei am realizat-o întâi investigând capabilitățile robotului și constatând că acestea sunt limitate în condiții reale, am utilizat rețeaua YOLO2, o metodă metodă foarte solidă, care este aproape invariabilă de poziționare și luminozitate. Am realizat apoi detecția fețelor și recunoașterea persoanelor pe baza NAOqi, Open CV. În plus am realizat un prim modul pentru recunoașterea activităților permite lui Pepper să înțeleagă dacă utilizatorul este inactiv și poate fi abordat, sau dacă este angajat și nu trebuie deranjat, bazat pe vedere de adâncime și rețele de convoluție (deep learning). Pentru orientarea robotului în mediu și navigare am folosit o abordare SLAM cu recunoșterea scenelor 3D cu biblioteca Kinfu Remake care folosește dispozitivul Kinect 1 pentru a realiza reconstrucția 3D, utilizând elementul de intrare de adâncime a senzorului.

Pe baza celor dezvoltate am realizat un set de comportamente de bază care pot fi combinate pentru a dezvolta comportamente complexe, de tipul celor descrise in secțiunea 4, de exemplu căutarea și interacțiunea cu o persoană, ghidarea persoanei către o anumită

locație, căutarea persoanei ghidate, interacțiunea prin voce cu persoana din fața robotului dintr-un grup de persoane sau cu persoana ghidată.

Ca urmare a activităților acestei etape, expuse anterior, s-a elaborat o tehnologie **Detecția, recunoașterea și urmărirea persoanei** pentru detecția, recunoașterea și urmărirea unei persoane de către robotul Pepper în condițiile existenței mai multor persoane în jurul robotului, tehnologie care poate fi aplicată și altor tipuri de roboți sociali.

Implementare la realizator si transfer tehnologic la agentul economic in etapa 3.

Diseminare

Proiectul SPARC a fost prezentat în cadrul workshop-ului ***Robotics, Automation, and Society*** organizat de Academia Română pe data de 29 November 2017, în cadrul Secției pentru Știința și Tehnologia Informației.

In prezent suntem in curs de finalizare a 3 lucrări științifice care vor fi trimise spre acceptare la următoarele conferințe:

- 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018), 1-5 Oct 2018, Madrid, Spania (deadline 1 martie 2018), IEEE
- 19th Towards Autonomous Robotic Systems (TAROS) Conference, 25-27 iulie 2018, Bristol, UK (deadline 2 feb 2018), Springer
- 12th International Symposium on Intelligent Distributed Computing, Oct 2018, Bilbao, Spain (deadline 9 aprilie 2018), Springer

La sfârșitul proiectului avem în plan elaborarea unei lucrări pentru a fi trimisă spre acceptare la Journal of Field Robotics, Wiley sau Robotics and Autonomous Systems, Elsevier.

Bibliografie

[1] Stuckless, R., “Developments in real-time speech-to-text communication for people with impaired hearing”, In M. Ross (Ed.), “Communication access for people with hearing loss”, pp. 197-226, MD: York Press, Baltimore, 1994.

[2] A Prochazka et.al. Microsoft Kinect Visual and Depth Sensors for Breathing and Heart Rate Analysis, Sensors, 16(7), 2016, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4970046/ *accesat dec 2017*

[3] Ahmad Jalal, Shaharyar Kamal, and Daijin Kim, Human Depth Sensors-Based Activity Recognition Using Spatiotemporal Features and Hidden Markov Model for Smart Environments**,** Journal of Computer Networks and Communications, Volume 2016
https://www.hindawi.com/journals/jcnc/2016/8087545/, *accesat dec 2017*

[4] Vennila Megavannan, Bhuvnesh Agarwal, R. Venkatesh Babu, Human Action Recognition using Depth Maps,
http://www.serc.iisc.ernet.in/~venky/Papers/kinect_action_recognition_spcom12.pdf, *accesat dec 2017*

[5] Miriam Buonamente, Haris Dindo, and Magnus Johnsson, Action Recognition based on Hierarchical Self-Organizing Maps, 2015, http://ceur-ws.org/Vol-1315/paper7.pdf *accesat dec 2017*

[6] Yong Du, Wei Wang, Liang Wang, Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition
https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Du_Hierarchical_Recurrent_Neural_2015_CVPR_paper.pdf *accesat dec 2017*

[7] Amir Shahroudy et.al. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis, 2015
https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Shahroudy_NTU_RGBD_A_CVPR_2016_paper.pdf *accesat dec 2017*

[8] Forrest N. Iandola, et.al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, 2016
https://arxiv.org/abs/1602.07360 *accesat dec 2017*

Lista de Figuri

Lista de Tabele