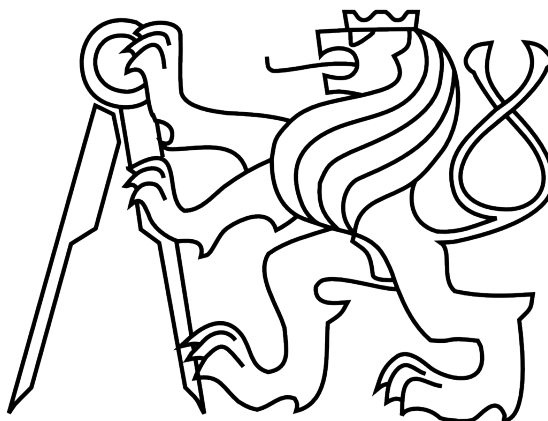


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra kybernetiky



Bakalářská práce

Interakce mezi Nao robotem a člověkem

Michaela Kavková

Vedoucí práce: RNDr. Daniel Průša, Ph.D.

Studijní program: Kybernetika a robotika

Obor: Robotika

21. května 2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Michaela Kavková
Studijní program: Kybernetika a robotika (bakalářský)
Obor: Robotika
Název tématu: Interakce mezi Nao robotem a člověkem

Pokyny pro vypracování:

1. Seznamte se s robotem Nao [1]. Získejte přehled o cílech projektu HUMAVIPS, který s uvedeným modelem robotu pracuje [2].
2. Seznamte se s knihovnou pro detekci lidských tváří poskytnutou pro projekt HUMAVIPS firmou Eyedea Recognition.
3. Navrhněte aplikaci pro interakci Nao robotu s lidmi. Zjistěte, které způsoby komunikace s Nao robotem jsou z technického hlediska spolehlivé, návrh aplikace tomuto přizpůsobte. V implementaci využijte knihovnu uvedenou v předchozím bodě (namísto modulu ALFace Detection).
4. Implementaci doplňte programátorskou dokumentací.
5. Aplikaci vyzkoušejte na vhodné skupině lidí. Vypracujte studii o průběhu testování a chování účastníků.

Seznam odborné literatury:

- [1] Aldebaran Robotics Academics forum. <http://academics.aldebaran-robotics.com/>
[2] HUMAVIPS web page. <http://humavips.inrialpes.fr/>
[3] Harms, D.; McDonald, K.: Začínáme programovat v jazyce Python. Computer Press, Brno, 2006.

Vedoucí bakalářské práce: RNDr. Daniel Průša, Ph.D.

Platnost zadání: do konce zimního semestru 2012/2013


prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 9. 11. 2011

Čestné prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne . . . 21.5.2012

. *Michaela Kavalová*

podpis

Poděkování

Chtěla bych poděkovat vedoucímu práce RNDr. Danielu Průšovi, Ph.D. a Prof. Ing. Václavu Hlaváčovi, CSc. za zapůjčení robota Nao, a dále oponentovi RNDr. Janu Preclíkovi, Ph.D. a jeho studentům programování za pomoc při testování. Rovněž bych chtěla poděkovat rodině za trpělivost.

Abstrakt

Tato práce se zabývá interakcí mezi humanoidním robotem Nao a člověkem. Popsáno je hardwarové a softwarové vybavení. Diskutovány jsou detektory zajišťující rozpoznávání lidských obličejů. Jedná se o modul ALFaceDetection, který je k dispozici na robotu Nao, a o detektor od firmy Eyedea Recognition. S využitím detekovaných tváří je navržena interakce s člověkem, která je vytvořena podle cílů projektu HUMAVIPS. Realizována je aplikací Vědomostní kvíz. Robot pokládá lidem otázky s předem známými odpověďmi a lidé na ně odpovídají. Důraz je kladen na přirozenost robota. Dosažené výsledky a některé problémy zjištěné během testování aplikace na skupině studentů jsou shrnuty v závěru práce.

Abstract

This bachelor thesis deals with the interaction between the humanoid robot Nao and human. Hardware and software of the robot is described. Two detectors of human faces are discussed. It is ALFaceDetection, a module which is available on Nao, and a detector developed by Eyedea Recognition company. Benefiting from detected faces, interaction with a human is proposed, fitting into objectives of European project HUMAVIPS. A part of the work is the application named Knowledge Quiz. Robot asks questions with predefined answers, people respond to them. Emphasis is placed on the nature of the robot in the proposed scenario. Achieved results and some problems identified during testing the application in a group of students are documented.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 1 |
| 2 | Robot Nao | 3 |
| 2.1 | Základní parametry | 3 |
| 2.2 | Klouby a ramena | 5 |
| 2.3 | Senzory | 9 |
| 2.4 | Framework NAOqi | 11 |
| 2.5 | Vývojové prostředí Choregraphe | 13 |
| 3 | Projekt HUMAVIPS | 17 |
| 4 | Knihovny pro detekci lidských tváří | 19 |
| 4.1 | Modul ALFaceDetection | 19 |
| 4.2 | Knihovna od firmy Eyedea Recognition | 20 |
| 4.3 | Porovnání modulů | 20 |
| 5 | Aplikace - Vědomostní kvíz | 23 |
| 5.1 | Popis aplikace | 23 |
| 5.2 | Specifikace programu | 23 |
| 5.3 | Změny specifikace | 24 |
| 5.4 | Implementace | 25 |
| 5.5 | Testování | 31 |
| 5.6 | Diskuze výsledků | 33 |
| 6 | Závěr | 35 |
| | Seznam obrázků | 37 |

OBSAH

| | |
|-----------------------|-----------|
| Seznam tabulek | 39 |
| Literatura | 41 |
| A Obsah DVD | 43 |
| B NAOqi moduly | 45 |

Kapitola 1

Úvod



Obrázek 1.1: Robot Nao

Robot Nao je malý humanoidní robot vyráběný francouzskou firmou Aldebaran Robotics. Současnou podobu má robot od roku 2005 jen s drobnými vylepšeními. Prodává se v několika variantách a jeho výroba je sériová. Aldebaran Robotics se snaží na trhu uspět s univerzálním robotem pro výzkumné a akademické účely. Cílem firmy je, aby se robot dostal na univerzity jako řádný prostředek k výuce robotiky, aby byl dostupný všem těmto

studentům a aby se studenti sami podíleli na jeho vývoji. I přesto, že robot v nejvyšší výbavě je velmi drahý, daří se tento cíl zčásti plnit. Používá se na více než 350 univerzitách po celém světě. Robot je také využívám v mezinárodní soutěži v robotické kopané RoboCup, kde v roce 2007 vystřídal psíka Aibo. V pořadí již druhého robota vlastní i Centrum strojového vnímání z Katedry kybernetiky FEL, ČVUT v rámci projektu HUMAVIPS.

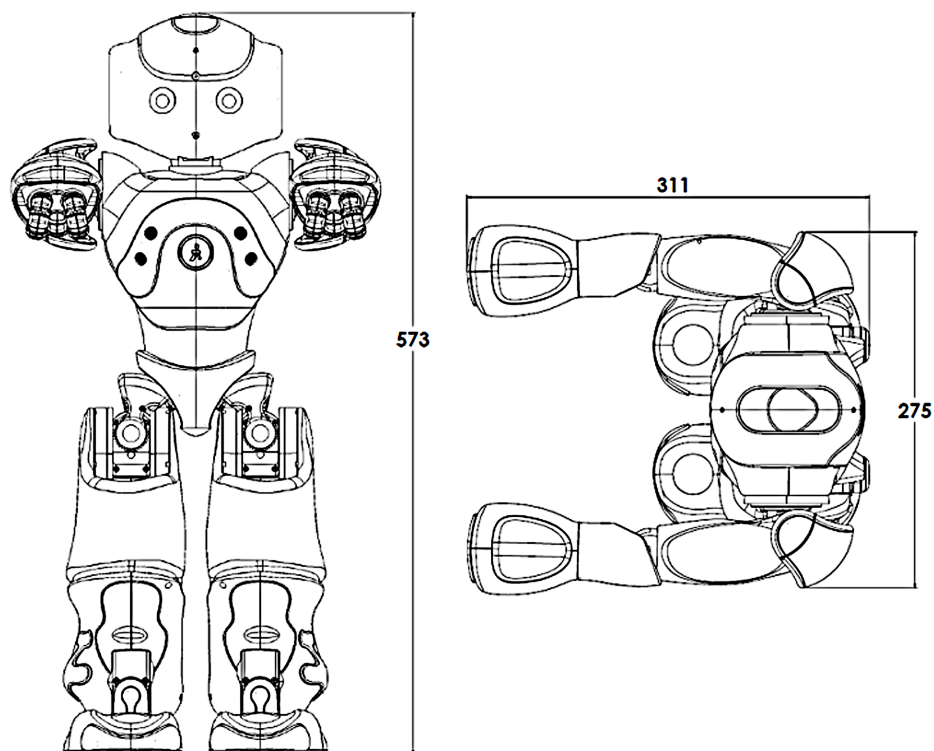
Cílem této práce je naprogramovat robota Nao tak, aby byl schopen interakce s lidmi. Tímto problémem se zabývá projekt HUMAVIPS. K interakci s okolím robota slouží mnoho senzorů a několik vyjadřovacích prvků včetně softwarových knihoven. V této práci zdokumentuji hardwarové a softwarové vybavení robota. Zaměřím se hlavně na moduly pro rozpoznávání lidských tváří ALFaceDetection. K dispozici mám i modul pro rozpoznávání lidských tváří od Firmy Eydea Recognition, se kterým mohu porovnávat. Součástí dokumentu je implementace aplikace v jazyku Python, která demonstruje možnosti modulu, včetně jejího testování. K aplikaci je připojena i programátorská dokumentace.

Kapitola 2

Robot Nao

2.1 Základní parametry

V kapitole o robotu Nao čerpám z dokumentace [1]. Tělo robota je vysoké 573 mm, široké 275 mm a hluboké 311 mm. Váží necelých 5 kg a je vyrobeno z technického plastu. Je napájeno šesti článkovou Li-Ion baterií s kapacitou 2Ah.

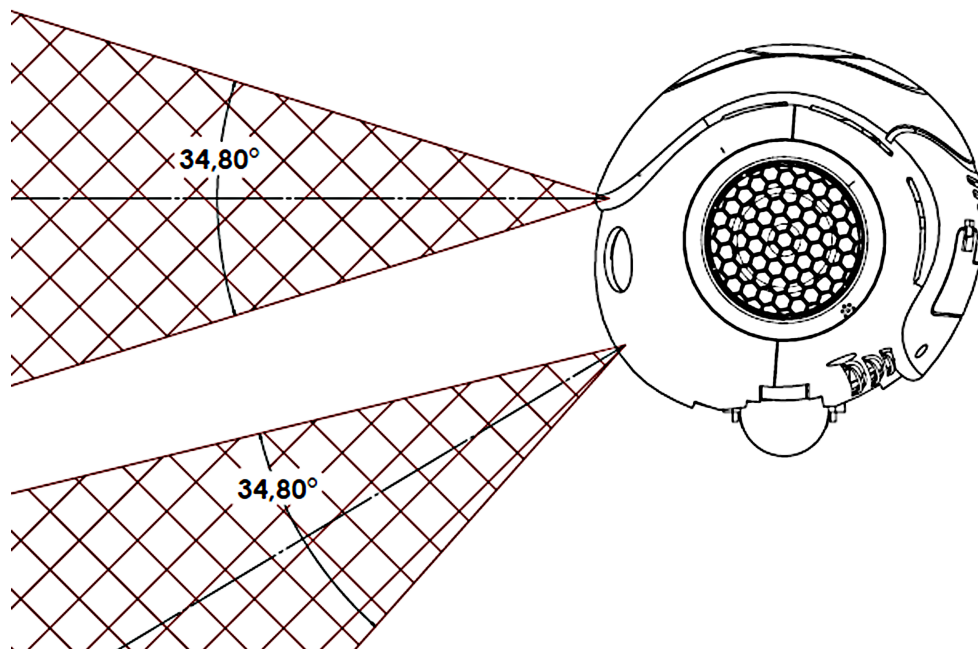


Obrázek 2.1: Základní parametry robota

Mozkem robota je procesor AMD GEODE s frekvencí 500MHz, kterému asistuje 256MB SDRAM nebo 2GB flash paměti. Robot má vestavěný operační systém Linux založený na OpenEmbedded distribuci. K počítači se připojuje přes Ethernet nebo Wi-Fi.

Robot se ovládá trojicí dotykových tlačítek na hlavě, které jsou podsvícené LED diodami. Na hrudi má robot umístěno hlavní tlačítko, které slouží zejména k zapínání a vypínání robota. Podsvícení tohoto tlačítka indikuje stav baterie. Další LED podsvícení je umístěno v očích a uších.

Robot vidí pomocí dvou VGA kamer umístěných na hlavě. Obě kamery jsou stejné, jejich diagonální zorné pole je 58° , horizontální $34,8^\circ$. Dokáží zaostřit na vzdálenost 30 cm a dál. Mohou snímat obraz s rozlišením 640×380 bodů s frekvencí 30 snímků za sekundu.



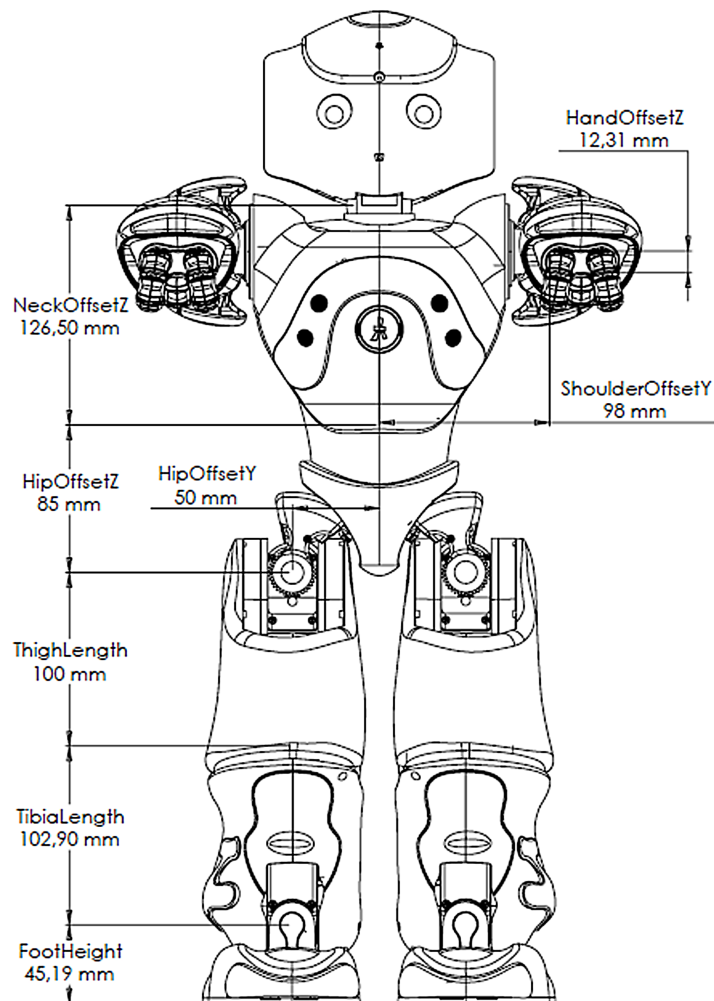
Obrázek 2.2: Znázornění detekčních úhlů kamer

Pro audio komunikaci mu slouží dva reproduktory o výkonu 2×2 W, které jsou umístěny na uších a jsou podsvícené, a dále mu slouží čtyři mikrofony umístěné na hlavě vpředu, vzadu a na uších.

Na robotu je umístěno několik desítek senzorů, které podrobněji rozepteší v kapitole 2.3. Robot má celkem 25 stupňů volnosti. Stupeň volnosti, anglicky degrees of freedom (DOF), je počet nezávislých parametrů, které jednoznačně popisují polohu tělesa. Dva DOF připadají na pohyb hlavou, jeden DOF na pohyb v pánvi, čtyři DOF na pohyb paže, dva DOF na pohyb dlaně a pět DOF na pohyb jedné nohy.

2.2 Klouby a ramena

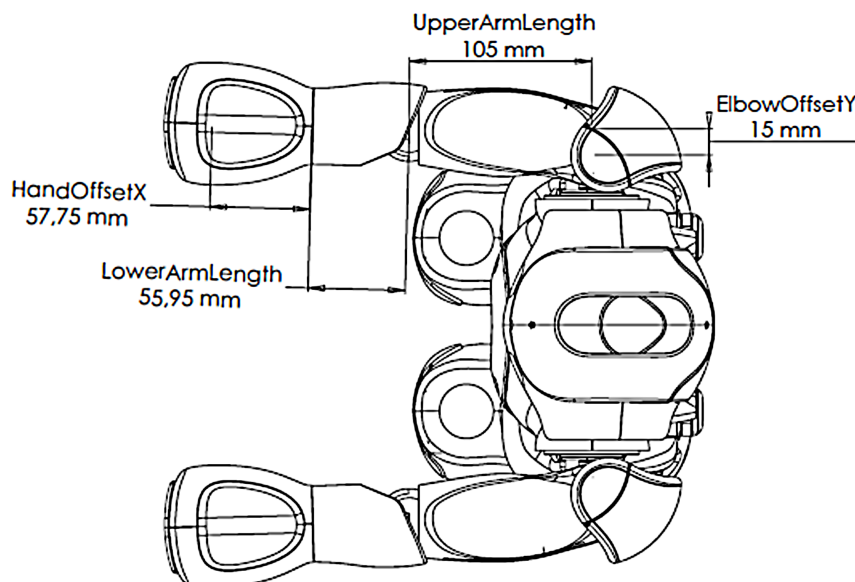
Robot má celkem 25 motorů, které ovládají pohyby v kloubech. Motory jsou ovládány přes 36 Hallových senzorů s přesností na 12 bitů. Tomu odpovídá přesnost přibližně $0,1^\circ$. Motory jsou dvojího druhu a každý tento druh má i různé převodové stupně. Specifikace motorů lze nalézt v dokumentaci robota [2], pro tuto práci však nejsou důležité. Délky jednotlivých ramen robota a omezení pohybu v kloubech je vyznačeno na následujících obrázcích a přehledně zpracováno do tabulek.



Obrázek 2.3: Délky ramen robota a další důležité parametry

| Jméno (anglická verze) | Jméno (česká verze) | Délka [mm] |
|------------------------|---|------------|
| NeckOffsetZ | Vzdálenost krku od středu robota v ose z | 126,50 |
| ShoulderOffsetY | Vzdálenost ramene od osy robota | 98,00 |
| HandOffsetZ | Vzdálenost zápěstí od středu ruky v ose z | 12,31 |
| HipOffsetZ | Vzdálenost kyčle od středu robota v ose z | 85,00 |
| HipOffsetY | Vzdálenost kyčle od středu robota v ose y | 50,00 |
| ThighLength | Vzdálenost kolene od kyčle | 100,00 |
| TibiaLength | Vzdálenost kotníku od kolene | 102,90 |
| FootHeight | Výška chodidla | 45,19 |

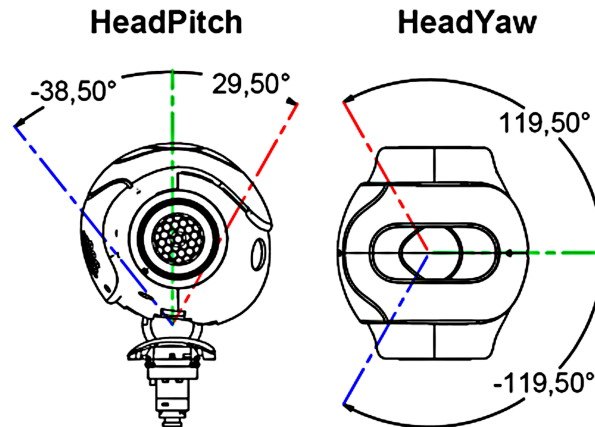
Tabulka 2.1: Rozměry robota k obrázku 2.3



Obrázek 2.4: Délky ramen robota a další důležité parametry

| Jméno (anglická verze) | Jméno (česká verze) | Délka [mm] |
|------------------------|---|------------|
| UpperArmLength | Vzdálenost od ramene k lokti | 105,00 |
| LowerArmLength | Vzdálenost od lokte k zápěstí | 55,95 |
| HandOffsetX | Vzdálenost zápěstí od středu ruky v ose x | 57,75 |
| ElbowOffsetY | Vzdálenost ramene od osy ruky | 15,00 |

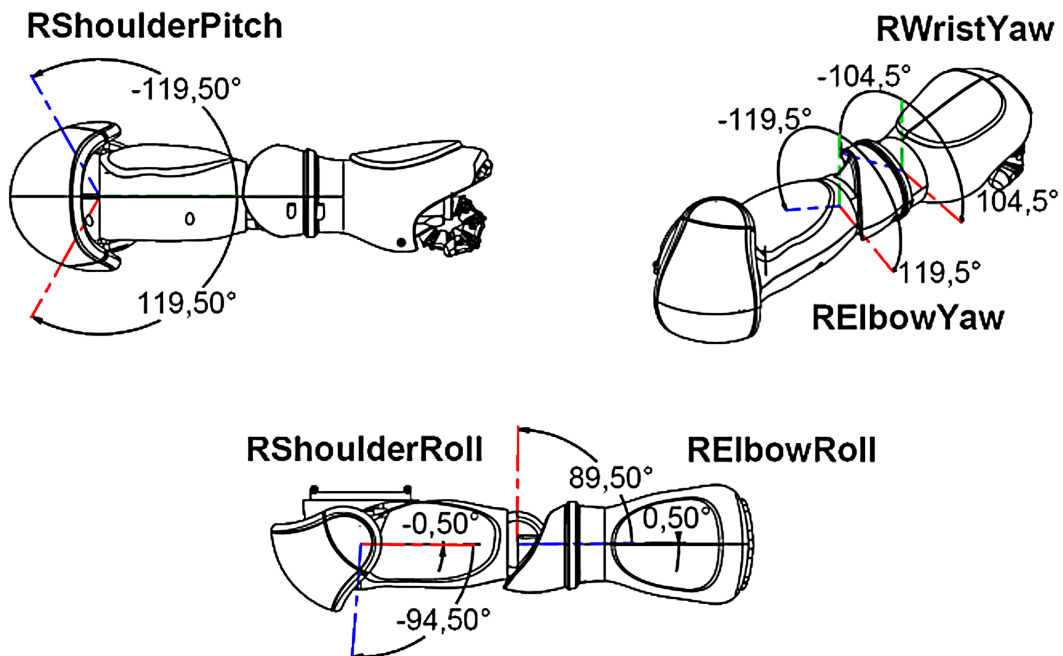
Tabulka 2.2: Rozměry robota k obrázku 2.4



Obrázek 2.5: Zobrazení krčního kloubu robota

| Jméno (anglická verze) | Jméno (česká verze) | Rozsah [°] |
|------------------------|-------------------------|-----------------|
| HeadPitch | Pohyb hlavy kolem osy y | -38,5 až 29,5 |
| HeadYaw | Pohyb hlavy kolem osy z | -119,5 až 119,5 |

Tabulka 2.3: Rozměry robota k obrázku 2.5

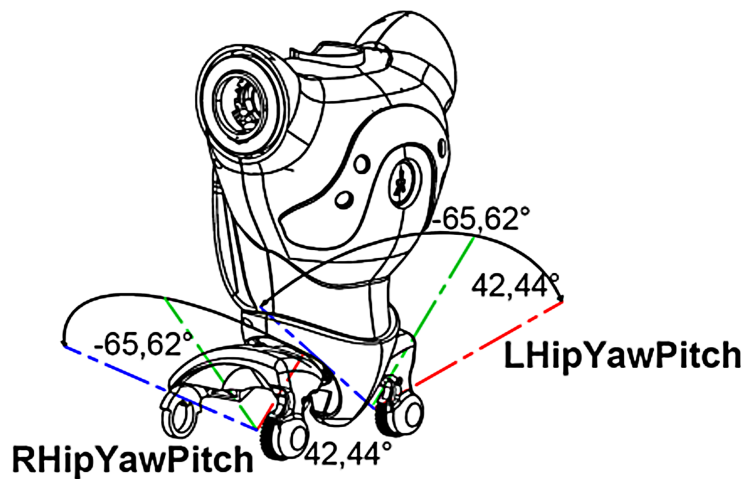


Obrázek 2.6: Zobrazení ruky robota

| Jméno (anglická verze) | Jméno (česká verze) | Rozsah [°] |
|------------------------|---------------------------|-----------------|
| RShoulderPitch | Pohyb ramene kolem osy y | -119,5 až 119,5 |
| RShoulderRoll | Pohyb ramene kolem osy x | -94,5 až -0,5 |
| RElbowYaw | Pohyb lokte kolem osy z | -119,5 až 119,5 |
| RElbowRoll | Pohyb lokte kolem osy x | 0,5 až 89,5 |
| RWristYaw | Pohyb zápěstí kolem osy z | -104,5 až 104,5 |
| RHand | Pohyb prstů v dlani | otevřít, zavřít |

Tabulka 2.4: Rozměry robota k obrázku 2.6

Pro pravou i levou ruku platí stejné rozsahy.

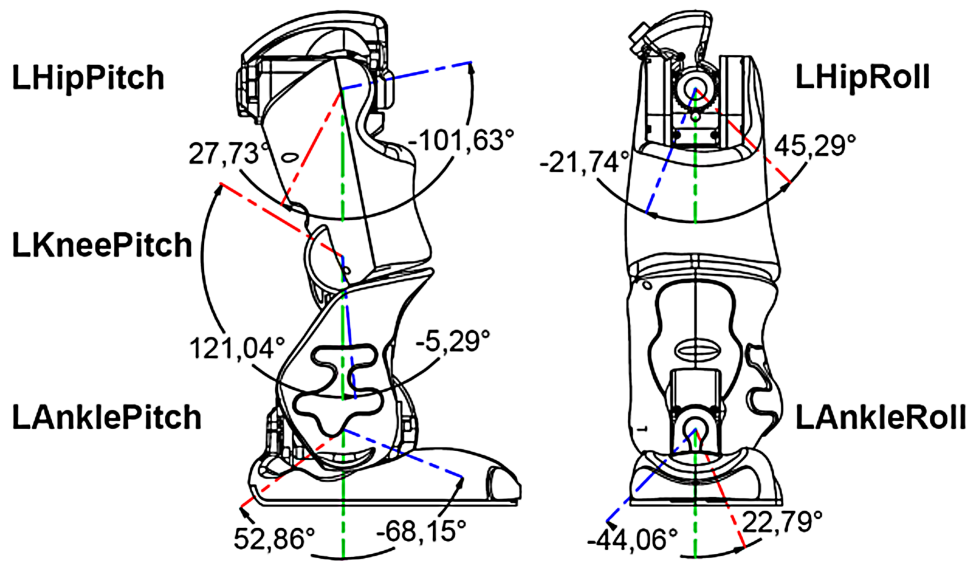


Obrázek 2.7: Zobrazení kyčelního kloubu robota

| Jméno (anglická verze) | Jméno (česká verze) | Rozsah [°] |
|------------------------|---------------------------|-----------------|
| LHipYawPitch | Pohyb trupu v levé kyčli | -65,62 až 42,44 |
| RHipYawPitch | Pohyb trupu v pravé kyčli | -65,62 až 42,44 |

Tabulka 2.5: Rozměry robota k obrázku 2.7

LHipYawPitch a RHipYawPitch jsou ve skutečnosti jeden motor, proto se mohou pohybovat pouze společně.



Obrázek 2.8: Zobrazení nohy robota

| Jméno (anglická verze) | Jméno (česká verze) | Rozsah [°] |
|------------------------|---------------------------|------------------|
| LHipRoll | Pohyb kyčle kolem osy x | -21,74 až 45,29 |
| LHipPitch | Pohyb kyčle kolem osy y | -101,63 až 27,73 |
| LKneePitch | Pohyb kolene kolem osy y | -5,29 až 121,04 |
| LAnklePitch | Pohyb kotníku kolem osy y | -68,15 až 52,86 |
| LAnkleRoll | Pohyb kotníku kolem osy x | -44,06 až 22,79 |

Tabulka 2.6: Rozměry robota k obrázku 2.8

Pro pravou i levou nohu platí stejné rozsahy.

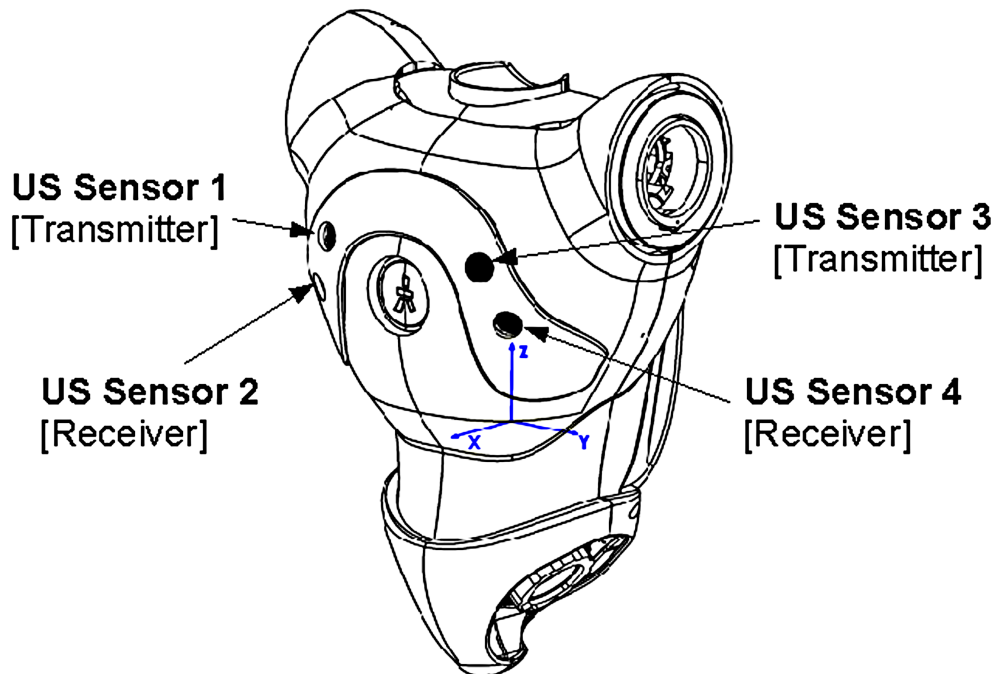
2.3 Senzory

Robot obsahuje:

- 36 Hallových senzorů
- 2 gyrometry
- 1 akcelerometr
- 2 tlačítka na nohách

- 2 dvojice sonarů
- 2 infračervené senzory
- 2 senzory zatížení na chodidlech

Hallovy senzory slouží k měření pohybu motorů v kloubech. Tlačítka na nohách slouží k detekci překážek, ale až v okamžiku, kdy do nich robot narazí. K lepší detekci překážek slouží dvojice vysílačů a přijímačů ultrazvukových vln umístěných na hrudi. Měření vzdálenosti je spolehlivé na 15-70 cm. Předměty bližší než 15 cm je robot schopen zaznamenat jen s neznámou vzdáleností. Ultrazvuk má frekvenci 40 kHz.

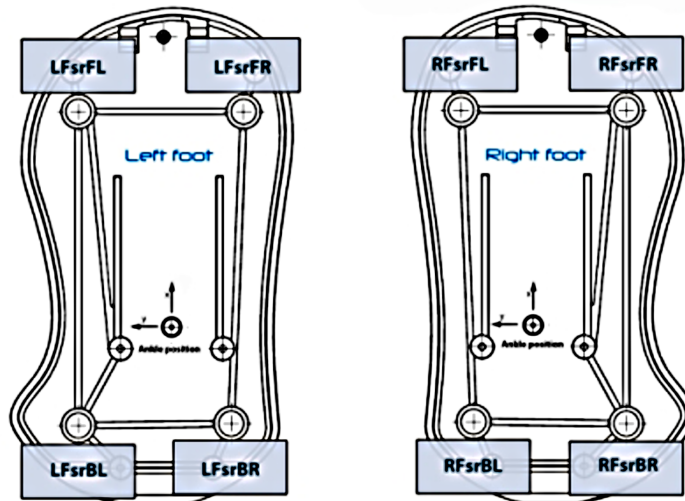


Obrázek 2.9: Dvojice sonarů

Infračervené senzory jsou použity pro komunikaci dvou NAO robotů, na použití NAO robota jako dálkového ovladače nebo na ovládání robota dálkovým ovladačem.

Centrum rovnováhy je umístěno v hrudi a má vlastní procesor. Zpracovává data ze dvou gyrometrů a jednoho akcelerometru. Pokud je robot v klidu, úhel naklonění je počítán z akcelerometru. Pokud je však robot v pohybu, lépe se uplatňují gyrometry. Spojení gyrometrů však způsobuje odchylku, proto se při pohybu robota úhel naklonění počítá i z akcelerometru, který tuto odchylku vyrovnává.

Senzory na chodidlech měří zatížení robota, ale dokáží nám i říci, zda je robotova noha na podlaze. Jsou zde použity FSR senzory (Force sensitive resistors). Dokáží změřit sílu 0 až 25 N.



Obrázek 2.10: FSR senzory

2.4 Framework NAOqi

Framework NAOqi byl vytvořen přímo pro robota NAO. Lze ho použít v Linuxu, Windows i Mac OS. Poskytuje API, které je možno volat v jazyku C++, Python a dokonce Urbi. Dostupné API lze pozorovat ve webovém prohlížeči zadáním IP adresy a portu robota. Funkce od výrobce se dají doplnit o vlastní.

V principu se NAOqi moduly volají třemi způsoby. První způsob je klasické volání dílčích úkonů za sebou. Až se dokončí jeden úkol, zavolá se druhý. Druhý způsob volání je paralelní. Všechny dílčí úkony jsou konány zároveň. Třetím způsobem je volání jako událost. Pokud je splněna podmínka, je zavolána obslužná procedura.

Struktura NAOqi začíná u brokeru. Broker (česky zprostředkovatel) provádí instrukce, které přicházejí na IP adresu a port. Při startu robota sám naváže spojení přes Ethernet nebo Wi-Fi. Pro spuštění našeho vlastního modulu se používá vzdálený broker, který předává instrukce hlavnímu brokeru. Do hlavního brokeru vstupují data z programu Telepathe a Choregraphe. Dále do něj vstupují instrukce ze senzorů, modulů, video kamery, mikrofonů, paměti atd.

Z celkových 36 modulů zde vyberu 6 důležitých, které v této práci používám. Kompletní seznam modulů včetně popisu je uveden v Příloze B.

- **ALFaceDetection**

- dokáže detekovat lidské obličej v obraze
- díky tomuto modulu může robot interagovat s lidmi

- **ALLandMarkDetection**

- rozpoznává NAO marky (sada znaků, které robot zná od výroby)

- **ALMemory**

- nejdůležitější modul robota
- implementuje událostmi řízenou paměť
- s ALMemory pracuje skoro většina modulů

- **ALMotion**

- poskytuje metody k ovládání kloubů
- jsou mezi nimi například chůze, ovládání rukou, zajištění kloubů proti pohybu

- **ALTextToSpeech**

- tvoří ze zadaného textu syntetizovanou řeč, kterou robot přehrává z reproduktorů
- bez tohoto modulu by se zřejmě neobešla žádná interakce s okolím, protože řeč je nejdůležitější dorozumívací prostředek současnosti

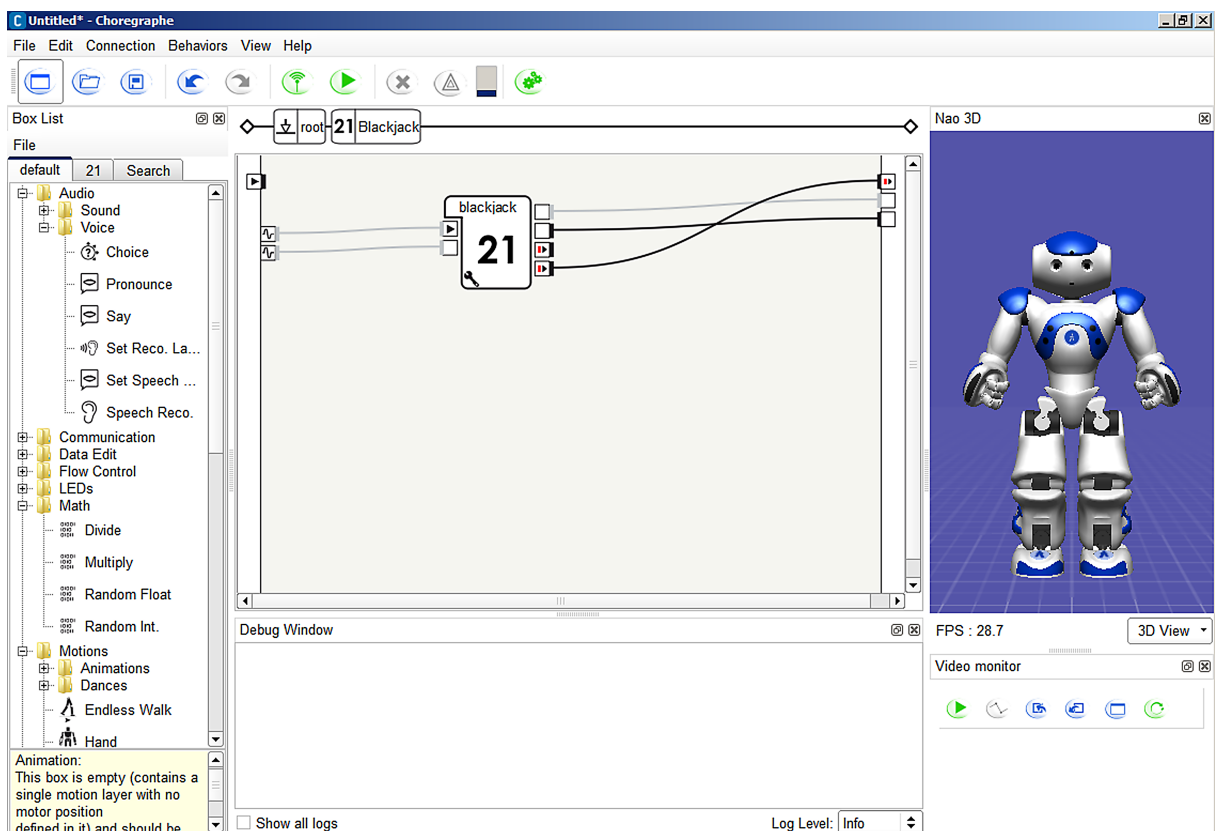
- **ALVisionRecognition**

- rozpoznává naučené objekty
- robota musíme nejprve naučit tyto objekty rozpoznávat

2.5 Vývojové prostředí Choregraphe

2.5.1 Choregraphe

Grafické vývojové prostředí od společnosti Aldebaran Robotics, Choregraphe slouží k programování robota Nao. Je určené pro širokou škálu lidí od neznalce programování až po experty. Výrobce dodává prostředí s knihovnou několika modulů zajišťujících základní funkcionalitu. Tyto moduly lze přepisovat a ukládat. Začátečníci moduly v podobě malých boxů pouze spojují a tím utváří program. Jsou zde naimplementovány i základní programátorské struktury jako podmínky a cykly. Pokročilí boxy upravují nebo si mohou vytvořit vlastní. Experti mohou vytvářet nové moduly v C++ a používat je v Choregraphe.



Obrázek 2.11: Vývojové prostředí Choregraphe

Grafické uživatelské rozhraní je intuitivní. V hlavní nabídce najdeme klasické ovládání známé ze všech programů. V uživatelské liště jsou důležitá tlačítka na propojení robota s Choregraphe, spuštění vytvořeného projektu na robotu, zastavení projektu, debugging a tlačítko na uvolňování a zastavení motorů robota. V sekci Behaviours se otevře okno se

správou vlastností nahraných v robotu. Z výroby to je například sedání, vstávání, Taichi apod.

2.5.2 Box list

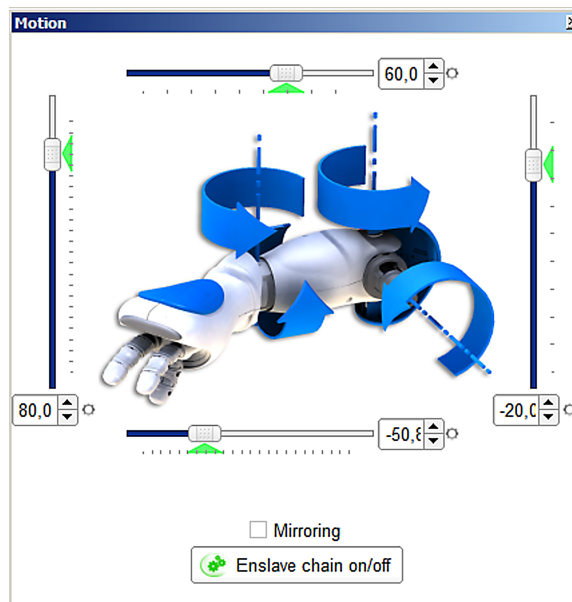
Nachází se na levé straně Choregraphe a ukazuje nám nabídku boxů. Jednoduchým drag and dropem můžeme boxy přetáhnout do šedé oblasti, kde se tvoří projekt. Spojováním boxů vznikne spustitelný program. Boxy jsou řazeny do kategorií a lze v nich i vyhledávat.

2.5.3 Video Monitor

Slouží k učení objektů, nahrávání databází naučených objektů a také ke sledování, co robot právě vidí. Databáze se ukládá samostatně mimo projekt.

2.5.4 Nao 3D

Toto okno slouží k reprezentaci robota ve 3D prostoru. Klikáním na části jeho těla můžeme jednoduše měnit polohu jeho kloubů. Pokud je robot připojen, tyto úpravy jsou ihned přenášeny na robota.



Obrázek 2.12: 3D zobrazení ruky robota

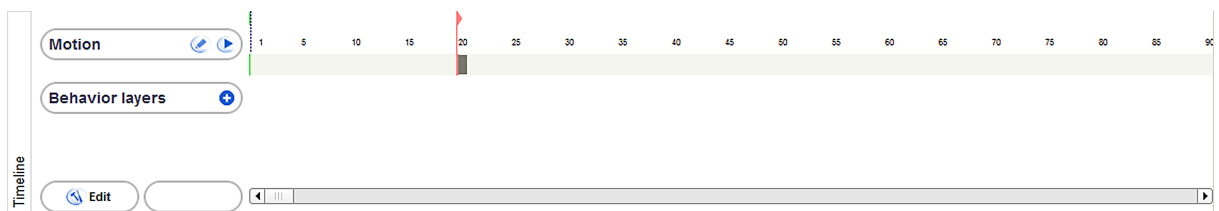
2.5.5 Debug Window

Zde se zobrazují chybové a varovné hlášky, ale programátor sem může vypisovat i svoje vlastní výpisy přidáním následujícího příkazu do kódu:

```
self.log("Ahoj světe!").
```

2.5.6 Tvorba pohybu

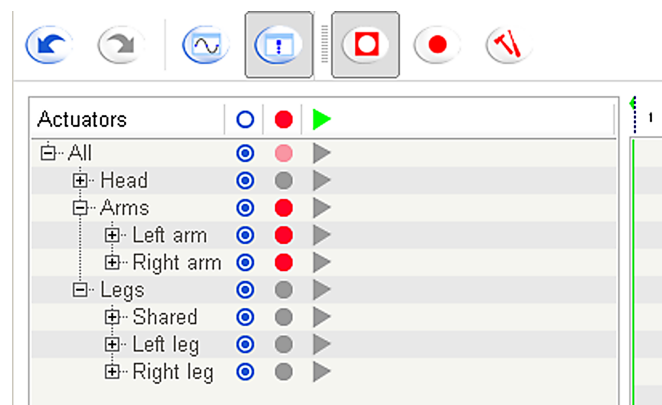
Souvislý pohyb robota se dá vytvořit pomocí časové osy (Time Line editor).



Obrázek 2.13: Časová osa

Klikneme na časovou osu, kde chceme provést pohyb, dále klikneme na část robota, kterou chceme hýbat, a všechny pozice se automaticky ukládají. Takto si jednoduše vytvoříme potřebné pohyby. Pohyby na časové ose můžeme vyexportovat do Pythonu nebo C++ a vložit do vlastního kódu.

Pohyby lze ještě nahrát tlačítka v liště, jako vidíte na obrázku 2.14. Reálné pohyby robota se přenáší do Choregraphe a automaticky ukládají. Po nahrání lze tyto pohyby dále měnit.



Obrázek 2.14: Seznam kloubů a značení jejich akcí

Klouby lze ovládat i jednotlivými příkazy v našem skriptu. Pro pohyb kloubu *název_kloubu* o úhel φ rychlostí *v* píšeme

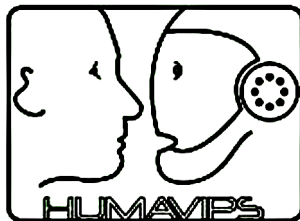
```
ALMotion.changeAngles(["název_kloubu1", "název_kloubu2"], [phi_1, phi_2], v).
```

Kapitola 3

Projekt HUMAVIPS

V této kapitole čerpám z technického dodatku projektu HUMAVIPS [3]. Název HUMAVIPS je zkratka pro *Humanoids with auditory and visual abilities in populated spaces*, což znamená Humanoidní roboti se schopností slyšet a vidět v zalidněných prostorách. Projekt je zaměřen na interakci s lidmi s důrazem na přirozenost. Humanoidní robot by měl být schopný prozkoumat zalidněné prostředí, lokalizovat lidi a určit jejich stav (nálada, na co je člověk zaměřený). Měl by být schopen navázat dialog. K tomu potřebuje senzory, díky nimž může pozorovat své okolí.

Doba trvání projektu je 3 roky a zapojilo se do něj 5 subjektů. Koordinátorem projektu je Národní institut výzkumu pro informatiku a automatizaci z Francie. Dodavatel robotů je Aldebaran Robotics. Další subjekty jsou České vysoké učení technické v Praze, IDIAP výzkumný institut ze Švýcarska a Univerzita Bielefeld z Německa. Na ČVUT pracuje na projektu Centrum strojového vnímání (CMP).



Obrázek 3.1: Logo projektu HUMAVIPS

Kapitola 4

Knihovny pro detekci lidských tváří

Rozpoznávání lidských tváří v obraze je jedna ze základních dovedností robotů schopných interagovat s okolím. Čím je detekci obličejů v obraze přesnější, rychlejší a robustnější, tím lepší výsledky robot může podat. Součástí API knihovny je i modul na rozpoznávání lidských obličejů `ALFaceDetection`. V této kapitole také popíšu Face detektor od firmy Eyedea Recognition.

4.1 Modul `ALFaceDetection`

Modul `ALFaceDetection` slouží k rozpoznávání lidských obličejů. Modul vrací všechny detekované tváře v jeden časový okamžik. Modul sám detekuje obličej v určitém intervalu, zpravidla 30 ms. Z výstupu modulu se dozvíme identitu tváře, velikost, polohu vůči kameře robota a souřadnice důležitých bodů obličej. Těmi jsou oči, obočí, nos a ústa.

Rozpoznávání je rychlé, ale v závislosti na robotově další činnosti se zpomaluje. Obvyklý interval detekce v praxi je 500 až 1000 ms. Rozpoznávací algoritmus bych hodnotila jako spolehlivý i při horším osvětlení. Tváře lze detekovat i z mírného úhlu natočení hlavy do všech směrů nebo při osvětlení tváře z jedné strany. Překvapivě dobrý výsledek podává robot při detekci osob s brýlemi.

Jako méně spolehlivé bych hodnotila rozpoznávání identity lidí. Modul je schopný zapamatovat si jednu tvář jako deset různých tváří a přesto při rozpoznávání identitu tváře nepřihadí. Je to dáno hlavně tím, že pro jednu osobu si nelze zapamatovat více dat. Pokud chceme rozpoznávání tváře přeučit na lepších datech, smažeme tím ta stará, a proto celý efekt přeučení ztrácí smysl. Pokud robotovi ukážeme fotografii osoby, kterou si robot zapamatuje, tak jméno osoby přiřadí správně. Člověk však neumí držet stálý výraz a natočení

hlavy, proto robot chybuje.

4.2 Knihovna od firmy Eyedea Recognition

Knihovna od firmy Eyedea Recognition využívá open-source algoritmus rozpoznávání lidských tváří v obraze. K tomu ještě optimalizuje chybu a čas rozpoznávání. Optimalizace rozhodovací strategie na základě času je založena na sekvenční analýze poměru pravděpodobnosti (Wald's sequential probability ratio test - SPRT), do kterého vstupuje chyba klasifikace. Tento algoritmus se nazývá WaldBoost, což je sjednocení algoritmu AdaBoost se sekvenčním testem poměru pravděpodobnosti. O klasifikátoru AdaBoost se můžeme více dočíst v [4] a o klasifikátoru WaldBoost v [5].

4.3 Porovnání modulů

K porovnání modulů jsem použila výsledky práce [6], která zkoumala *false positive rate* a *true positive rate* těchto modulů. *False positive rate* je poměr detekovaných obličejů v místech, kde ve skutečnosti nejsou, ku všem detekovaným obličejům. Obdobně *true positive rate* je poměr detekovaných obličejů, které v obraze jsou, ku všem detekovaným obličejům. Testovalo se na třech databázích. První z nich je MIT+CMU databáze obličejů vyfocených zepředu. Zbylé dvě databáze jsou obrázky o dvou rozlišeních lidí, kteří se většinou dívali do kamery. Výsledek porovnání je zaznamenán v tabulce 4.1.

| | MIT+CMU databáze | |
|---------------------|------------------------|--------------------|
| | počet špatných detekcí | true positive rate |
| ALFaceDetection | 0 | 3.5 % |
| Eyedeia Recognition | 0 | 59.7 % |
| | Obrázky 320 × 240 | |
| | počet špatných detekcí | true positive rate |
| ALFaceDetection | 2 | 36.4 % |
| Eyedeia Recognition | 2 | 90.5 % |
| | Obrázky 640 × 480 | |
| | počet špatných detekcí | true positive rate |
| ALFaceDetection | 0 | 5.9 % |
| Eyedeia Recognition | 0 | 91.2 % |

Tabulka 4.1: Porovnání modulů na třech databázích obrázků

Z tabulky 4.1 je vidět, že modul od Eyedeia Recognition má lepší *true positive rate*, což dokazuje optimalizaci tohoto modulu.

Kapitola 5

Aplikace - Vědomostní kvíz

5.1 Popis aplikace

Aplikace Vědomostní kvíz slouží ke zkoušení jednoho člověka z připravených otázek s možnými odpověďmi A, B, C. Zkoušející je robot Nao. Robot položí zkoušenému několik otázek ze zvoleného oboru. Zkoušený odpovídá ukázáním písmene správné odpovědi. Nakonec robot člověka oznámkuje.

Aplikace je navržena podle cílů projektu HUMAVIPS. Robot s člověkem komunikuje hlasem, kamery používá k hledání člověka a k detekování odpovědi. Robot používá i LED diody k signalizaci správné či špatné odpovědi. Umí se za člověkem otáčet. To všechno působí na zkoušeného člověka lidským dojmem.

5.2 Specifikace programu

Kvízové otázky jsou uloženy v souboru formátu `.csv`. Každý řádek souboru obsahuje obor otázky, text otázky, možné odpovědi a písmeno správné odpovědi. Tyto údaje jsou vždy odděleny středníkem. Soubor může vypadat takto:

```
PR;What isn't basic programming structure?;Cycle;Command;Feedback;C  
G;What is the capital of the Czech Republic?;Pilsen;Brno;Prague;C
```

Otázky jsou v angličtině, protože robot Nao neumí syntetizovat český text. Databáze otázek je uložena v robotově paměti a uživatel ji může kdykoliv zaměnit za jinou.

Program začíná kontrolou přítomnosti databáze a jejího načtení. Dále se načítá databáze

statistiky, která má podobný tvar jako databáze otázek. Pokud robot bude zkoušet člověka, kterého už někdy dříve zkoušel, použije statistiku, ve které má uloženy špatně zodpovězené otázky. Ty pak zkouší jako první.

Po načtení všech důležitých dat z paměti se robot uvede. Mezitím se již detekují tváře před kamerou a robot zkoumá, zda-li je již někdy neviděl. Pokud tvář robot nezná, zapamatuje si ji pod názvem `humanx`, kde `x` je pořadové číslo tváře. Robot se také za tvář otáčí, aby ji měl vždy naproti sobě.

Poté robot náhodně vybere sekvenci otázek, které bude pokládat. Po vyslovení otázky robot čeká na odpověď zkoušeného. Zkoušený má k dispozici pět různých karet s odpovědmi. Tři z nich jsou možné odpovědi A, B, C a zbylé dvě slouží k zopakování otázky (znak vykřičníku) a k vyjádření, že odpověď dotyčný nezná (znak otazníku). Aby byla odpověď uznána, musí se nacházet pod obličejem člověka. Tím předejdeme omylu, kdyby se karta nacházela někde v místnosti, kde se provádí kvíz. Za správnou odpověď člověk získá jeden bod, za špatnou nula bodů a za nezodpovězenou otázku také nula bodů. Otázku lze opakovat nejvíce pětkrát, pak robot automaticky přejde na další otázku. Po detekci odpovědi robot bliká očima. Zeleně zabliká, pokud detekoval správnou odpověď, červeně pokud špatnou, modře pokud člověk chce zopakovat otázku a nakonec žlutě pokud člověk odpověď nezná.

Mezi jednotlivými otázkami robot zkoumá, která tvář byla detekována nejčastěji a podle toho aktualizuje jméno osoby. Může se tedy stát, že zkoušené subjekty se při běhu programu vymění, a robot si ukládá statistiku pod jiným jménem osoby.

Po položení všech otázek robot sečte body a podle výsledku člověka slovně ohodnotí. Uloží si do souboru získanou statistiku a program se ukončí.

Pokud kdykoliv během běhu programu lidská tvář zmizí z obrazu na déle než minutu, robot se začne rozhlížet, jestli člověka znovu nezahledne. Tuto proceduru opakuje dvakrát, a pokud ani poté člověka nenajde, program se sám ukončí.

5.3 Změny specifikace

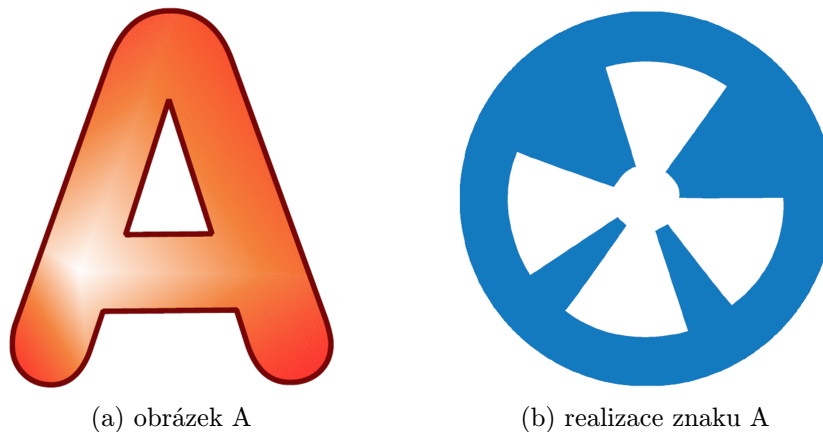
Během práce na úloze bylo nutné vytvořit karty, které bude robot schopný rozpoznat s nejmenší chybou a v co nejkratším čase. Zpočátku jsem používala modul `ALVisionRecognition` na rozpoznávání naučených obrázků v obraze. Tento modul se však projevil jako nevhodný, protože robot nestíhal detekovat a zpracovávat tolik dat v rozumném čase. Někdy detekce trvala i 30 sekund, navíc robot často chyboval a detekce byla velmi závislá na

míře osvětlení.

Zkoušela jsem i rozpoznávání hlasu modulem ALSpeechRecognition. Robot slova dokázal rozpoznat s nízkou chybovostí, ale bylo nutné mluvit robotovi přímo do mikrofону z krátké vzdálenosti. Proto jsem tento modul taktéž určila jako nevhodný pro moji aplikaci.

Nakonec jsem pro rozpoznávání odpovědí použila modul k detekci tzv. NAO marků ALLandMarkDetection. Ten funguje velmi spolehlivě a hlavně rychle. Je také závislý na osvětlení, ale ne tolik jako ALVisionRecognition. V průběhu práce jsem však zjistila, že robot také chybuje a NAO marky si občas plete. Ze 70 detekcí se spletl třikrát, což odpovídá 4,3 %. Chyba se vždy objevovala jen při první detekci daného NAO marku.

Aplikaci jsem naprogramovala jak pro obrázky s písmeny, tak i pro NAO marky. Detekce obrázků je sice pomalá, ale vypadá více lidsky. Aplikace pracující s NAO marky je však rychlá, ale pro lidi není tolik zajímavá. Obě varianty jsou však možné použít. V následujících sekcích budu popisovat aplikaci používající NAO marky.

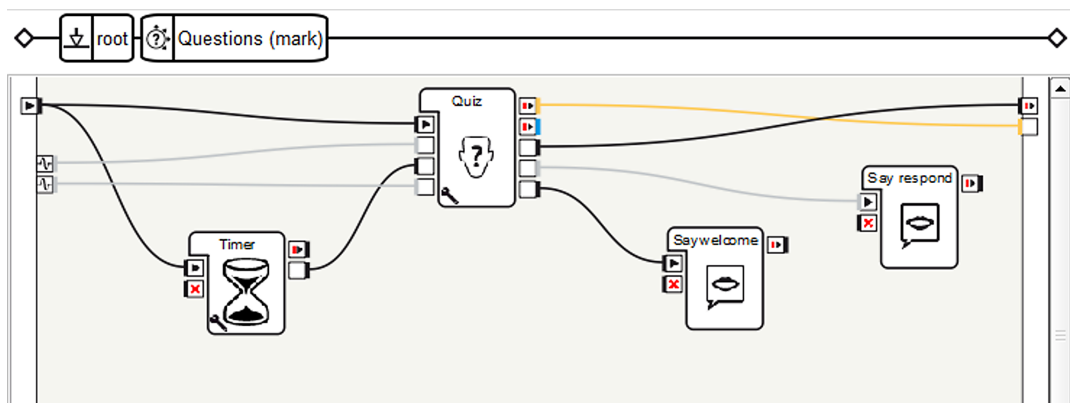


Obrázek 5.1: Karty s odpovědí A

5.4 Implementace

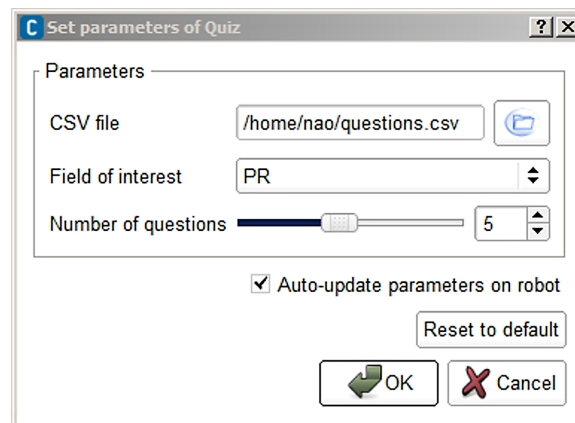
Program je napsán v programovacím jazyku Python. Využilo se grafického vývojového prostředí pro robota Nao Choregraphe 1.12.0.62. Vytvořila jsem si tedy box v Choregraphe s názvem *Questions (mark)*. V tomto boxu se nachází další čtyři boxy a dva výstupy z ALMemory, a to výstupní proměnná modulu pro rozpoznávání tváří a výstupní proměnná modulu pro rozpoznávání NAO marků. Tyto data vedou do boxu Quiz, ve kterém běží hlavní program. Dále se zde nachází box Timer, kterým se v pravidelných časových inter-

valech aktivuje vstup onTimer. Zbylé dva boxy *Say welcome* a *Say respond* slouží pouze k mluvení. Jelikož robot není schopen zároveň mluvit a vykonávat jinou činnost, musela jsem mluvení v úvodu a mluvení při detekci odpovědi volat paralelně. Vytvoření paralelního boxu mi přišlo rychlejší a bezpečnější, než programování vláken v Pythonu. Při úvodní řeči může robot zároveň detekovat obličej a zapamatovávat si je, při mluvení po detekci odpovědi robot může blikat očima.



Obrázek 5.2: Struktura boxu Questions (mark)

V parametrech boxu Quiz si můžeme nastavit cestu k databázi otázek, obor otázek a počet otázek.



Obrázek 5.3: Vlastnosti boxu Quiz

Ve vlastnostech boxu Timer lze nastavovat pouze periodu, která je defaultně nastavená na 5 s.

Pro načítání databáze otázek jsem se rozhodla použít balíček `csv`, který je součástí Pythonu. Soubor typu `.csv` se v Pythonu snadno načítá pomocí kódu:

```
questions = csv.reader(open(file, 'rb'), delimiter=';')
for row in questions:
    data = data + [row]
```

Stejně snadno se do `.csv` souboru i zapisuje:

```
statistic = csv.writer(file, delimiter=';')
for k in range(len(newStatistic)):
    statistic.writerow(newStatistic[k])
```

Kód boxu Quiz obsahuje jednu třídu s názvem `MyClass`. Ta obsahuje 29 metod. Následuje jejich výčet, a co konají.

- `def __init__(self)`
 - inicializace třídy
 - definice použitých proměnných
 - zavádění proxy spojení (`ALTextToSpeech`, `ALVideoDevice`, `ALFaceDetection`, `ALMotion`)
- `def onLoad(self)`
 - nastavení počáteční barvy očí
- `def onUnload(self)`
 - při ukončování programu uvolňuje klouby
- `def onInput_onStart(self)`
 - metoda volaná jednou na začátku programu
 - nejprve se načítají data ze souborů
 - poté se robot představí a uvede (zavolá box *Say welcome*)

- dokud robot nezdetekuje žádného člověka, tak čeká; pokud detekuje, kontroluje jeho jméno, hledá v databázi a připravuje otázky
- následuje kód, kde robot vyslovuje otázku, čeká na odpověď a vyhodnocuje odpověď
- nakonec je zavolána metoda `finish()`
- `def onInput_onTimer(self)`
 - metoda, která vyhodnocuje, zda byla tvář spatřena před méně než minutou
 - pokud nebyla, robot se rozhlíží na obě strany a snaží se člověka najít
- `def onInput_onFace(self, p)`
 - metoda je volaná vždy, když přijde nějaká informace z `ALMemory` o detekci tváře
 - vstupem je proměnná `p`, která obsahuje mimo jiné identitu tváře, pozici detekované tváře v obraze, souřadnice očí, obočí, nosu a úst
 - součástí této metody je i pohyb hlavy robota při každé třetí detekci ve směru člověka
 - každou pátou detekci se aktualizuje název detekované tváře
- `def onInput_onMark(self, p)`
 - metoda, která zpracovává data z `ALMemory` o detekovaných NAO marcích
 - vstupem je proměnná `p`, ve které je číslo NAO marku a pozice značky v obraze
- `def onInput_onStop(self)`
 - volá metodu `onUnload()`
- `def writeAboutDetectedFace(self, p)`
 - z proměnné `p` vypisuje počet detekovaných tváří a jméno tváře do debug logu
- `def readCSV(self, file)`
 - načítá databázi otázek do proměnné

- `def setZeroHeadPosition(self)`
 - metoda, která nastaví hlavu robota do nulové pozice
- `def isMarkBelowFace(self, mark, face, timem, timef)`
 - kontroluje, zda čas detekce tváře a odpovědi je rozdílný nejvýše o 1 s
 - kontroluje, zda odpověď je pod nebo nad obličejem
- `def turnHeadToFace(self, heads)`
 - metoda, která nastavuje natočení hlavy robota směrem k člověku
 - vstup `heads` jsou průměrované úhly pozice člověka vůči kameře robota
- `def findQuestions(self, interest, file, number)`
 - metoda, která slouží k sestavení sady otázek
 - nejdříve se mezi otázky zařadí ty, které se vyskytovaly ve statistice špatných odpovědí
 - za ně se zařadí náhodně seřazené otázky
 - pokud je počet dostupných otázek menší než požadovaný, počet požadovaných otázek se automaticky sníží
- `def sayQuestion(self, file, number)`
 - slouží k vyslovování znění otázek
- `def countScore(self, answer, definedAnswer, k)`
 - vrací skóre odpovědi dané otázky
 - v této metodě je také implementováno blikání očí robota
- `def finish(self)`
 - robot ohodnotí člověka na základě jeho celkového skóre
 - poté se rozloučí, uloží statistiku a vypne program

- `def questionRecursive(self, k)`
 - slouží k rekurzivnímu volání vyslovení otázky, pokud je detekován znak vykřičníku
 - je omezen na pět opakování
- `def greenBlinking(self)`
 - robotovy oči třikrát zeleně zablikají
- `def redBlinking(self)`
 - robotovy oči třikrát červeně zablikají
- `def yellowBlinking(self)`
 - robotovy oči třikrát žlutě zablikají
- `def blueBlinking(self)`
 - robotovy oči třikrát modře zablikají
- `def readStatistic(self)`
 - slouží k načítání statistiky
- `def findUser(self, names)`
 - vybírá z detekovaných jmen nejčastěji detekované
- `def checkUser(self, user)`
 - hledá ve statistice jméno tváře
- `def writeStatistic(self)`
 - zapisuje statistiku do souboru
- `def readSerialNumber(self)`
 - načítá konstantu ze souboru, která udává pořadové číslo tváře

- `def writeSerialNumber(self)`
 - zapisuje pořadové číslo tváře do souboru
- `def learnFace(self)`
 - slouží k zapamatování neznámého obličeje pod jménem `humanx`, kde `x` je pořadové číslo tváře

5.5 Testování

Součástí každého softwarového projektu (i menšího) je testování. V průběhu programování aplikace jsem ji testovala na sobě. Po dokončení aplikace jsem navštívila Jiráskovo gymnázium v Náchodě, kde jsem měla domluvené testování s cca. 10 studenty semináře Programování pod vedením RNDr. Jana Preclíka, Ph.D..

Na testování se dostavilo 8 studentů 3. ročníku. Pro studenty jsem si připravila krátkou prezentaci o robotu, aby měli představu o jeho hardwarovém a softwarovém vybavení. Poté započalo vlastní testování. Studenty jsem poučila, jakou aplikaci budou testovat a co se od nich očekává. Při testování se neobjevila žádná výrazná chyba v kódu programu. Studenti však zjistili, že si robot plete identity jejich tváří a že si plete NAO marky. To studenti ohodnotili negativně, ale na druhou stranu přiznali, že chybovat je lidské. Experimentálně bylo zjištěno, že si robot NAO marky splete v 4,3 % případů, a proto naprostá většina kvízů byla detekována správně.

Testeři nebyli spokojeni s častějšími výpadky robota. Robot se za dobu testování dvakrát zasekl tak, že musel být restartován a jednou došlo k výpadku spojení s kamerou. Očekávali, že za vysokou nákupní cenu robota dostanou kvalitu provedení a hlavně spolehlivost. Tyto výpadky a problémy s chodem robota jim kazily dojem z testovaného programu.

Poslední z věcí, které se studentům nelíbily, byla hlasitost robota. Přestože robot mluvil na maximální hlasitost, v místnosti plné lidí a běžících počítačů ho nebylo příliš slyšet. Proto studenti spíše otázky četli z obrazovky notebooku spojeného s robotem. Výslovnost robota jim příliš nevadila, i když nějaká slova vyslovoval špatně.

Bohužel těmto konkrétním studentům asi angličtina moc nejde a proto jsem jim některé věty musela překládat. Robotem to zřejmě nebylo, protože jsem aplikaci zkoušela i na mých přátelích, kteří bez problému rozuměli.



Obrázek 5.4: Studenti pozorující robota

Studentům se líbilo, že se za nimi robot otáčel, a pokud nikoho nedetekoval, začal je hledat. Dále se jim líbilo slovní hodnocení, které robot studentům na konci sdělil. Taktéž se zasmáli, když po robotovi chtěli po šesté zopakovat otázku a robot je odmítl s tím, že už tuto otázku nebude nikdy opakovat.

Myslím, že robot Nao zanechal na studentech dobrý dojem. Chybování v detekcích je sice zklamalo, ale robot jim zpestřil seminář. Jedni studenti říkali, že jim chyby v detekci úplně nevadily, druzí zase říkali, že je tím robot nepoužitelný. Obě strany měly pravdu. Detekci však moje aplikace nijak nezmění, proto aplikaci Vědomostní kvíz považovali za odladěnou. Robotiku hodnotili jako perspektivní obor. Možná, že jim i moje prezentace umožnila nahlédnout do oboru, který doteď neznali. A třeba bude i inspirací pro volbu dalšího studia po absolvování gymnázia.



Obrázek 5.5: Student odpovídá na otázku

5.6 Diskuze výsledků

I přes změny specifikace programu během práce se mi podařilo dokončit aplikaci v podobě, jakou jsem deklarovala. Vědomostní kvíz je kompletní a nenapadá mě žádné podstatné vylepšení. Jelikož jsou API moduly nepřístupné, nemohu se pokusit o vylepšení detekce. Přínosem by bylo, kdyby se Aldebaran zabýval zdokonalením modulu na rozpoznávání hlasu a také syntetizací českého textu. Pak by komunikace s robotem byla mnohem přirozenější.

Kapitola 6

Závěr

Seznámila jsem se s robotem Nao, s jeho hardwarem, softwarem a frameworkem NAOqi. Popisala jsem robotovy rozměry a schopnost pohybu v jednotlivých kloubech. Věnovala jsem se robotovým sensorům a audio/video vstupům a výstupům. Seznámila jsem se s NAOqi API knihovnou. Prostudovala jsem modul na rozpoznávání lidských tváří ALFaceDetection a modul od firmy Eyedea Recognition. Zhodnotila jsem jejich odlišnosti a chybovost. Byla jsem schopna implementovat aplikaci, která ukazuje, jak spolehlivě pracuje ALFaceDetection a jaké jsou jeho hranice. Aplikaci jsem otestovala na skupině studentů gymnázia. Aplikaci jsem doprovodila programátorskou dokumentací.

Při návrhu aplikace jsem se zaměřila hlavně na moduly rozpoznávání lidských obličejů. Bez nich by nebyla možná interakce robota s člověkem. Při testování aplikace si studenti příliš nestěžovali na detekci obličejů. Více je zaujala rozpoznávání identity tváře, která není příliš spolehlivá. Přesto jim robot připadal věrohodný. V aplikaci také využívám rozpoznávání obrázků z obrazu a detekci NAO marků. Vzhledem k náročnosti rozpoznávání obrázků byla více používána detekce NAO marků. Ta sice tolik nepřispěla k polidštění robota, ale byla spolehlivější.

S trochou nadsázky můžeme říct, že chyby v detekcích jsou také lidské, protože lidé nejsou dokonalí a pletou se. Proto si myslím, že se mi podařilo vytvořit aplikaci, ve které robot působí realisticky.

Seznam obrázků

| | | |
|------|---|----|
| 1.1 | Robot Nao | 1 |
| 2.1 | Základní parametry robota | 3 |
| 2.2 | Znázornění detekčních úhlů kamer | 4 |
| 2.3 | Délky ramen robota a další důležité parametry | 5 |
| 2.4 | Délky ramen robota a další důležité parametry | 6 |
| 2.5 | Zobrazení krčního kloubu robota | 7 |
| 2.6 | Zobrazení ruky robota | 7 |
| 2.7 | Zobrazení kyčelního kloubu robota | 8 |
| 2.8 | Zobrazení nohy robota | 9 |
| 2.9 | Dvojice sonarů | 10 |
| 2.10 | FSR senzory | 11 |
| 2.11 | Vývojové prostředí Choregraphe | 13 |
| 2.12 | 3D zobrazení ruky robota | 14 |
| 2.13 | Časová osa | 15 |
| 2.14 | Seznam kloubů a značení jejich akcí | 15 |
| 3.1 | Logo projektu HUMAVIPS | 17 |
| 5.1 | Karty s odpovědí A | 25 |
| 5.2 | Struktura boxu Questions (mark) | 26 |
| 5.3 | Vlastnosti boxu Quiz | 26 |
| 5.4 | Studenti pozorující robota | 32 |
| 5.5 | Student odpovídá na otázku | 33 |

Seznam tabulek

| | | |
|-----|--|----|
| 2.1 | Rozměry robota k obrázku 2.3 | 6 |
| 2.2 | Rozměry robota k obrázku 2.4 | 6 |
| 2.3 | Rozměry robota k obrázku 2.5 | 7 |
| 2.4 | Rozměry robota k obrázku 2.6 | 8 |
| 2.5 | Rozměry robota k obrázku 2.7 | 8 |
| 2.6 | Rozměry robota k obrázku 2.8 | 9 |
| 4.1 | Porovnání modulů na třech databázích obrázků | 21 |

Literatura

- [1] Dokumentace robota Nao
<http://www.aldebaran-robotics.com/documentation/index.html>
- [2] Motor a převodovka v každém kloubu robota v. 3.3
<http://www.aldebaran-robotics.com/documentation/nao/hardware/kinematics/motors.htmlmotor-and-reduction-type-for-each-joint-3-3>
- [3] Technický dodatek projektu HUMAVIPS
<http://humavips.inrialpes.fr/files/2011/03/HUMAVIPS-annexI-final.pdf>
- [4] AdaBoost - cvičení z RPZ
<http://cmp.felk.cvut.cz/cmp/courses/recognition/Labs/adaboost/adaboost.pdf>
- [5] WaldBoost – Learning for Time Constrained Sequential Detection
<ftp://cmp.felk.cvut.cz/pub/cmp/articles/sochman/Sochman-TR-2004-15.pdf>
- [6] Preliminary empirical comparison of the face detector used in the humanoid robot NAO, Vojtěch Franc, Daniel Průša, 2011

Příloha A

Obsah DVD

- Aplikace - Soubory potřebné ke spuštění aplikace Vědomostní kvíz
 - marks.pdf, pictures.pdf - značky k vytištění
 - questions.csv, serialNumber.txt - soubory musí být nutně přítomny v robotu Nao v adresáři /home/nao/
 - questions_mark.crg, questions_picture.crg - projekty v programu Choregraphe
 - Questions.png - obrázek boxu, je třeba jej vložit do cesty programu Choregraphe, např.: c:\Program Files (x86)\Aldebaran\Choregraphe 1.12.0.62\share\choregraphe\media\images\box\interaction\
- Dokumentace - Obsahuje tuto práci ve formátu .pdf
- Literatura - Odborná literatura, ze které jsem čerpala
- Multimedia - Video a foto z testování

Příloha B

NAOqi moduly

NAOqi API obsahuje 36 modulů. Následuje jejich úplný výčet, vlastnosti a použití.

- **ALAudioDevice**
 - slouží k získání zvuku z mikrofonů
 - posílá zvuk do reproduktorů
- **ALAudioPlayer**
 - slouží k přehrávání MP3 a wav souborů z reproduktorů
- **ALAudioRecorder**
 - slouží k nahrávání zvuku
- **ALAudioSourceLocalization**
 - počítá pozici (azimut a výšku) zvukového zdroje detekovaného pomocí modulu ALSoundDetection
- **ALBehaviorManager**
 - spravuje vlastnosti (behaviours) NAO robota
- **ALBonjour**
 - slouží k obsluze robota přes Bonjour

- **ALFaceDetection**
 - detekuje lidské obličej
- **ALFaceTracker**
 - sleduje lidské obličej získané z ALFaceDetection
- **ALFileManager**
 - spravuje uživatelská data ve sdíleném adresáři
- **ALFrameManager**
 - používá se k přehrávání projektů z programu Choregraphe v NAOqi
- **ALFs**
 - přenáší data ze senzorů zatížení na nohách
- **ALInfrared**
 - slouží k přijímání/odesílání rozkazů z infračervených senzorů
- **ALLandMarkDetection**
 - rozpoznává NAO marky (sada znaků, které robot zná od výroby)
- **ALLaser**
 - zprostředkovává práci s laserovou hlavou (není součástí akademického robota Nao)
- **ALLauncher**
 - povoluje dynamické napojování na knihovny, spouští je a odpojuje je
- **ALLeds**
 - ovládá LED podsvícení robota
- **ALLogger**
 - vypisuje chyby, varování a informace o systému

- **ALMemory**
 - nejdůležitější modul robota
 - implementuje událostmi řízenou paměť
 - s ALMemory pracuje skoro většina modulů
- **ALMotion**
 - poskytuje metody k ovládání kloubů
 - jsou mezi nimi například chůze, ovládání rukou, zajištění kloubů proti pohybu
- **ALMotionRecorder**
 - poskytuje metody, které umožňují zapamatování pozic kloubů
- **ALPreferences**
 - povoluje připojení na xml soubory, ve kterých jsou nastavené předvolby
- **ALPythonBridge**
 - překládá Python za chodu
- **ALRedBallDetection**
 - detekuje červené balónky na základě barevné saturace
- **ALRedBallTracker**
 - sleduje červené balónky na základě dat z ALRedBallDetection
- **ALResourceManager**
 - spravuje robotovy zdroje akcí (pohyb, LED, mluvení, přehrávání zvuku)
- **ALRobotPose**
 - vrací robotovu pózu
- **ALSensors**
 - detekuje zmáčknutí senzorů dotyku na nohách a zmáčknutí hlavního tlačítka na hrudi

- **ALSentinel**
 - hlídá nabití baterie, správnou funkci NAOqi, přehřátí motorů a hlavy
- **ALSonar**
 - používá se k detekci překážek pomocí sonaru
- **ALSoundDetection**
 - detekuje zvuk hlasitější než je daný práh
- **ALSpeechRecognition**
 - zpracovává příchozí zvuky a vyhledává v nich lidskou řeč, dokáže identifikovat slova
- **ALTextToSpeech**
 - tvoří ze zadaného textu syntetizovanou řeč, kterou robot přehrává z reproduktorů
- **ALVideoDevice**
 - zpracovává informace z videokamer a předává je do modulů na rozpoznávání obrazu (obličejů, NAO marků, červených balónků a naučených objektů)
 - nastavuje videokamery (například rozlišení, počet snímků za minutu, přepínání kamer)
- **ALVisionRecognition**
 - rozpoznává naučené objekty
 - robota musíme nejprve naučit tyto objekty rozpoznávat
- **ALVisionToolbox**
 - dokáže nahrávat video, ukládat jednotlivé snímky, vyrovnávat bílou barvu a další užitečné úkony
- **DCM**
 - spravuje spojení se senzory a motory