BahiaRT@Home 2017 Team Description Paper

Josemar R. de Souza, José Diôgo da Silva Carneiro, Ramon C. Mercês, Ricardo S. Matos, Samuel Santiago Carvalho, Icaro Ariel Carneiro Leite, Oberdan R. Pinheiro, and Marco A. C. Simões

State University of Bahia (UNEB/ACSO), Salvador, BA, Brazil
josemar@uneb.br,josejuniordsc@gmail.com rcmerces@gmail.com, rikrdomattos@gmail.com, sam.scarvalho17@gmail.com, icaaro.leite@gmail.com oberdan.pinheiro@gmail.com, , msimoes@uneb.br
http://www.acso.uneb.br/bill/

Abstract. The center for Computer Architecture and Operating Systems (ACSO) at the State University of Bahia (UNEB), represented by Bahia Robotics Team (BahiaRT), introduces **BILL** ((**B**ot **I**ntelligent **L**arge capacity **L**ow cost). Aiming to develop solutions for service and assistive robotics, RoboCup@Home works as the perfect test environment to validate such solutions. This paper describes the development of BILL's basic features.
Keywords = BILL, assistive robotics, RoboCup@Home.

1 Introduction

BILL (**B**ot **I**ntelligent **L**arge capacity **L**ow cost) (Fig.1) is the idealization of a project born in 2008 aimed at research in the area of service robots. Bill was designed and built to meet the requirements league RoboCup@home as part of the RoboCup initiative.

Fig. 1. BILL's conceptual art on the left and his current look on the right.

In the RoboCup@Home league, a set of benchmarks is used to evaluate the performance of the robots in a non-standardized and realistic home environment. Thus, the tasks are focused on the areas of interaction and cooperation between Human-Robot, navigation and mapping in dynamic environments, recognition of objects and faces in natural lighting conditions, artificial intelligence, standardization and system integration, etc.

In 2015 and 2016, BILL has participated in the RoboCup at Home league, and in the last three years he's got the second place in the Latin-American Robocup.

2 Group Background

BahiaRT is a scientific cooperation group which aims to insert the state of Bahia into scientific research in Robotics and Artificial Intelligence. The initiative was created by ACSO in August 2006 by researchers and students from UNEB and other institutions.

The main objective of BahiaRT is to actively participate in the international research initiative known as RoboCup. The goal is to strengthen the Brazilian participation in this important initiative on Robotics and Artificial Intelligence.

We are competitors in the RoboCup since 2007, in the first year, we have participated in 2D Soccer Simulation League and also in the Mixed Reality demonstration competition (formerly Physical Visualization). In Mixed Reality, BahiaRT won the third place in RoboCup 2009 and the fourth place in RoboCup 2010. BahiaRT also has developed the MR-SoccerServer: the main module of MR software infrastructure. In 3D Soccer Simulation, BahiaRT ranked the fourth place in the last two years in RoboCup.

3 Functionalities

3.1 Face and Gender Recognition

The vision module has been totally redesigned, this package is responsible for performing any image processing every robot needs: people recognition, face detection, gender, objects, gestures, etc. The idea here was to actually build a module responsible for receiving, processing and responding to external stimulus from image capture. The following diagram describes at a high level the basic operation of this module:

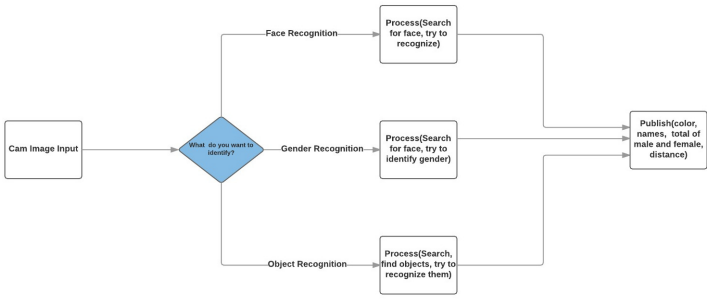

Fig. 2. High level diagram vision module

Currently, the facial and gender recognition modules, which follow the pipeline proposed by [8], are already in operation, where we have four steps to do: Face Detection, Face Preprocessing, Training a machine-learning algorithm from collected faces and Face Recognition.

Face Detection The first step is to detect all faces in the image without worrying about recognizing it, just detecting human faces. This step is possible thanks to the Haar-Cascade classification method, which is based on the cascade classification algorithm [2].
The algorithm Haar Cascade Classifiers runs an image from the pixel (0,0) with sub-windows, passing through each image region using a classification. Thereby, it follows line by line using the X and Y axis, repeating itself every time a search is completed and restarting with higher sub-window scales until the sub-window meets the size of the entire image. All sub-windows pass through several classifiers, only the ones that keep a positive result after each and every analysis, are considered faces. If in any step of the detection until the end of the cascade, one window is rejected, the classifier understands there are no faces. When this algorithm finds a face, it will separate the face from the rest of the image.

Face preprocessing This step is essential for increasing the accuracy of facial recognition, as it attempts to reduce the differences that can be found from environment to environment, such as light conditions, face orientation, background removal and so on. But to increase reliability in real-world conditions, it is necessary to include methods of detecting facial features such as eye detection (which is being used in this project), these classifiers are also based on the cascade classification [2].

Collecting faces and learning from them After all the pre-processing the collected faces are saved so that it is possible to perform a match in the next phase. At this stage also a label is saved to identify each distinct pre-processed face, so that it can be re-identified. It is very important that faces have position variability, because saving them in single positions can make it difficult to recognize them, since people's expressions and behavior are not pre-established.

Face recognition With preprocessed and saved images, these are trained in the Eigenfaces [3] machine learning algorithm, which creates a model that allows us to pre-tell who is in front of the webcam. With the algorithms of the class FaceRecognizer merber of OpenCv library, return a number responsible for measuring the reliability of being a particular person, we use a threshold number based on tests done indoors, which allows us to confirm that the result returned is correct.

Gender Recognition For gender recognition we use a dataset with great variability of images of men and women we assign a label to each genre and apply

two phases of facial recognition: Preprocessing (for the face to match the characteristics of the dataset), and Face Recognition to compare the proximity that an input face corresponds to the dataset.

3.2 Speech Recognition and Synthesizing

The voice is the form of man machine interaction most used to give commands to the robot in a more natural way, either through specific commands or natural language.

For recognition, we use the CMU PocketSphinx [4], which features greater flexibility for adaptation and personalization, allowing to adapt the dictionary and acoustic models to the problem of context. After the speech recognition, use the output of pocketsphinx to feed the state machine using the boost Regex library. With that we construct a grammar able to interpret the commands and fulfill the assigned tasks.

Pocketsphinx uses a statistical approach based on hidden Markov models (HMM) [5], and its architecture is defined in 5 modules: front-end, phonetic dictionary, acoustic model, language model and decoder.

In order to speak with people, we use the ros package sound play [7] witch can translate a ROS [1] topic into sounds. Supporting since built-in sounds until OGG/WAV files. In the process of synthesizing, we use Festival[6], a software that allows us to change the various aspects of voice, such as tone, speed of speech, among others, in order to ensure better understanding by the listener, allowing us to generate better interaction experience.

3.3 Navigation

Navigation is the keystone for efficient execution and enviroment interaction for robots. The components used by BILL are; encoders output, odometry, gmapping(ROS), move_base(ROS), amcl(ROS), map_server(ROS) and 360 degrees laser scanner. The pipeline will be discribed below. The encoder data is used by odometry module to estimate the moviments of the robot in space, further, the odometry data is used to trace trajectory to a desired target by the move_base. The trajectory and odometry feedback are displayed in a map provided by the map server. Once all data is being published, the simultaneos mapping and localization using the adaptative Monte Carlo localization[10] is activated integrating the 360 laser scan data.

3.4 Bill's GUI

This module was designed in an efort to achieve dynamic utilization and optmal initialization of ROS nodes. Using QT[9] integrated with ROS, through the already existing message communication provided by the Robot Operating System, was possible to create this module just as a visualization tool, it means, all the processing necessary as face recognition or voice syntesis, are done in their own packages, and all information provided by them is sent to the interface which displays and send requisitions to each respective package.

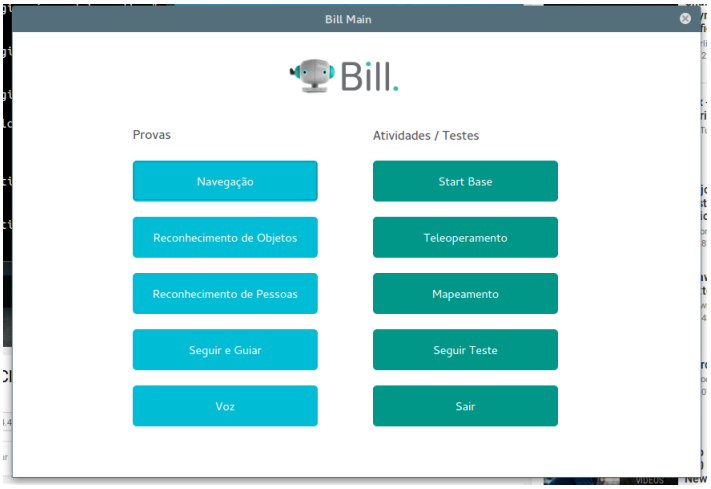

Fig. 3. BILL GUI

4 Experiments and results

We have made lots of experiments with Bill to measure and prove its abilities in different environments. On face recognition field we only tested in indoor environments. Seven people were used in this experiment, which was divided into two steps: The first, only one person in front of the robot and the second with two people. In the first step 50 rounds were done for each person, in which the person appeared in front of the robot for 300 cycles and the robot had to recognize it. We obtain an average of 72.64% accuracy. In the second one, 7 rounds were done combining the people in different pairs for each round. The process was made as the first step, for every round, the robot had to recognize the people during 300 cycles. BILL scored the average result of 66.67% accuracy. The speech recognition was based on robocup's 2016 speech recognition & audio detection test, where the robot has to understand and answer a serie of questions. 4 people in two different environments, indoor and outdoor, were used. For each person the questions were asked 100 times. The indoors environment tests were done with the operator 75 cm distant from the robot. BILL obtained an average of 61.25% accuracy in this requirement. The outdoors tests, using the same settings as previous, scored an average of 56.67% accuracy in this task.

5 Conclusions and future work

During development and competition tests, BILL has proven to be an efficient product assisting humans in their daily chores, thus perfect BILL's abilities and introduce new funcionalities are essential to all future work. Our major development intentions it's make BILL more inteligent using a deep learning into

a bill core to recognize patterns and uses the feature to improve bill capacites about speech and face recognition and allow bill to manipulate objects and recognize then.

6 Module Description

In order to provide completely autonomous operation, BILL owns two main modules of control: The High-level control, which includes algorithms to solve functionalities such as global task planning, navigation and tracking, recognition faces, user-interaction, among others. And a low-level to control sensors and actuators in the real world.

.

6.1 BILL Hardware Description

Based on characteristics of robots we built for IEEE Open and Robocup, in addition to observation of equipments used around the world, we were able to come to a new motion base for the robot BILL, presenting higher mobility using a round base, 2 differential drive wheels and 2 free wheels -one in the front and other in the rear for maintaining balance. All the electronic parts were carefully check in order to avoid short-circuits and increase power.

- Base: One Arduino Mega 2560; Two motors IG32P 24VDC 190 RPM Gear Motor with Encoder; One Notebook Dell Inspiron 15R-4470, intel core i7; One digital buzzer. One RPLIDAR 360 degrees laser scaner. Three Sabertooths controlers. One LM35 linear temperature Sensor. three batterys 11.1 volts and 2800 mAh; One digital push button;
- Torso: Mini actuator Firgelli Automations; One Emergency switch;
- Arm: five Dynamixel-ax-12A; One ArbotiX-M; Maximum load: 1kg.
- Head: One Dynamixel-ax-12A; One Microsoft Kinect sensor; Two Microsoft life Cam HD-3000; One Rode Videomic Pro.
- Tablet: Motorola tablet.

6.2 BILL Software Description

The low level is composed of a proportional control running on arduino boards, although simple, very effective compared to the previous versions of our own codes. The communication and high level system is composed of tools developed by our team and open source applications of the Robot Operating System (ROS).

- Navigation, localization and mapping: Hector mapping.
- Face recognition: OpenCV library.
- Speech recognition: PocketSphinx library; Boost library.
- Speech generation: Festival.
- Grapichal User Interface to control and start ROS Node: QT Creator.

7 References

1. Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng. Ros: an open-source robot operating system. In ICRA workshop on open source software, volume 3, page 5, 2009.

2. Michael J. Jones Paul A. Viola. Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition. Proceedings of the 2001 IEEE Computer Society Conference, 1:I–511 – I–518, 2001.

3. Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. Eigenfaces fisherfaces: Recognition using class specific linear projection, 1997.

4. D Huggins-Daines. Pocketsphinx api documentation, 2010.

5. Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, et al. Spoken language processing, volume 18. Prentice Hall Englewood Cliffs, 2001.

6. Alan Black, Paul Taylor, Richard Caley, Rob Clark, Korin Richmond, Simon King, Volker Strom, and Heiga Zen. The festival speech synthesis system, version 1.4. 2. Unpublished document available via http://www. cstr. ed. ac. uk/projects/festival. html, 2001.

7. Sound_Play. Retrieved January 18, 2017, from http://wiki.ros.org /sound_play.

8. Lélis Daniel. Mastering OpenCV with Practical Computer Vision Projects, 2012.

9. Qt Corporation, Qt Creator. Retrieved March 08, 2017, from https://www. qt.io/ide/.

10. Frank Dellaert, Dieter Fox, Wolfram Burgard, Sebastian Thrun. "Monte Carlo Localization for Mobile Robots." Proc. of the IEEE International Conference on Robotics and Automation Vol. 2. IEEE, 1999.

Name of the Team
BahiaRT@Home
Contact information
Josemar - josemarsbr@gmail.com
Ramon - rcmerces@gmail.com
Website
http://www.acso.uneb.br/bill/
Team members
Josemar Rodrigues de Souza
Ramon Campos Mercês
José Diôgo da Silva Carneiro
Ricardo Silva Matos
Samuel Santiago de Carvalho
Icaro Ariel Carneiro Leite
Oberdan Rocha Pinheiro
Marco Antônio Costa Simões
Romero Mendes Freire de Moura Júnior

Hardware:

- Two motors IG32P 24VDC 190 RPM Gear Motor with Encoder
- One Arduino Mega2560.
- One Bluno Mega2560.
- One IO Expansion Shield for Arduino V7.
- One Mega Sensor Shield V2.4.
- One Sabertooth 2 X 12.
- Two r LV-MaxSonar-EZ Sonar Sensors.
- Mini actuator Firgelli Automations.
- One Emergency switch.
- One Microsoft Kinect sensor.
- Two Microsoft life Cam HD-3000.
- One Rode Videomic Pro.
- Motorola Xoom Tablet.

Software:

- slam_gmapping: Navigation, localization and mapping.
- PocketSphinx library and Boost library:Speech recognition.
- Point cloud library(PCL)