Sourena @Home 2011 Team Description Paper

Seyed Mohammad Ghaffarian[1], Hadi Mahzarnia[1], Hedayat Vatankhah[1], Farshad Einabadi[1], Mahdi Nejadgholi[1], Alireza Shafaei[1], Seyed Mahdi Hosseini[2], Hamid Shahsavari[3], Amir Kiomarsi[3], and Saeed Shiry Ghidary[1]

[1] Computer Engineering Department, Amirkabir University of Technology,
[2] Electrical Engineering Department, Amirkabir University of Technology,
[3] Mechanical Engineering Department, Amirkabir University of Technology,
Tehran, Islamic Republic of Iran
{ s.m.ghaffarian, mahzarnia, hvatankhah, einabadi, nejadgholi, shafaei, sm-hosseini, h.shahsavari, shiry}@aut.ac.ir

Abstract. In this paper we introduce the "Sourena" robot for the 2011 RoboCup@Home competitions. We describe the hardware characteristics and software capabilities of Sourena which has made it reliable for different tasks in home environments.

Keywords: Sourena, @Home Robot, RoboCup, TDP.

1 Introduction

RoboCup@Home is an international competition for autonomous mobile robots aimed to be used in indoor environments. The league is designed to enhance the role of indoor robots in human's everyday life. Almost all the tests in this league require the robot to interact with humans. For a robot to successfully accomplish these tests, a wide range of techniques are required to be prepared and implemented for real-time usage.

The Sourena team is a three year robotic research team in the field of home robotics, which attempts to overcome the challenges in this field. We built our robot from scratch and named it Sourena after the Iranian army leader in the Parthian era. Three years ago, we took part in the 2008 RoboCup@Home competitions with our young robot and we reached the finals. After improving our robot in some aspects, we took part in the 2009 competitions with the second version of Sourena. One year later, we participated in the 2010 RoboCup@Home competitions with the third version of Sourena. According to the experience from previous competitions, we have improved Sourena to effectively perform the tasks required in RoboCup@Home 2011 by improving the robot's hardware and software. Improvements include a new hardware design plus new software components such as improved motion control software for navigation and localization, improved visual object recognition and a novel face recognition method. These improvements have provided the robot with abilities to perform:

- Navigation and obstacle avoidance in indoor environments
- Automatic map generation and self-localization (SLAM)
- Speech recognition and synthesizing
- Human recognition and tracking
- Object recognition and manipulation
- Special tasks in RoboCup@Home league

We have developed a novel software framework to facilitate the process of developing distributed robotic software, which is able to support multiple languages (e.g. Java, C++, C#, etc.) and multiple operating systems (e.g. Linux and Windows). In section 2 we describe the hardware characteristics of the robot. Section 3 describes the software architecture and the robot's A.I. capabilities in more detail. Finally, section 4 describes some novel systems and idea's to use in home robots.

2 Hardware Specifications

2.1 General Specification

Sourena (3rd version), as seen in Fig. 1, is a 3-wheeled differential drive robot with 2 active wheels. The robot consists of two parts, a base and a body. The base has a size of 40cm x 45cm x 60cm and the total height of the robot is 150cm. The motors have a total power of 200W and a maximum speed of 100-rpm; new wheels with a diameter of 20cm, which results a maximum speed of 1.05 m/s. We have used two 12V Lead-acid batteries with 18Ah each to supply the robot's power. Total weight of Sourena is about 30kg. For manipulation purposes, Sourena uses a 5-DOF robotic arm. The electronic boards of Sourena consist of a main board, motor drivers, sensor interfaces, communication interface, supply units, and control and protection unit.

We should also mention that for the 2011 competitions in Istanbul, we have designed a new and improved robotic platform (4th version of Sourena) which is much more suitable to our requirements.

2.2 Processing Unit

As mentioned before, the software framework enables the robot to use multiple computers for processing purpose; a laptop with a 2.6 GHz Intel Core-i7 processor is used as the central computing unit and another laptop with a 2.5 GHz Intel Core2Duo processor is used as a complementary computing unit. More laptops can be used (either on-board or off-board) to increase the computation power. A wide range of abilities and accessories come along with the laptops, such as speaker, voice communication tools (microphone), network communication facilities (LAN and WLAN), etc.

2.3 Sensors

To mention the robot's sensors, Sourena's vision system is a Bumblebee-2 stereo camera mounted on a pan/tilt unit. For localization, navigation and obstacle avoidance, Sourena uses a Lase laser scanner and shaft encoders. Sourena is also equipped with a microphone.

Fig. 1. Sourena (3rd version)

3 Software Specifications

3.1 A Distributed Software Framework

According to the experience of our robotic team, there are some important features that are missing in many of today's robotic toolkits and frameworks. These features constitute the design choices for our proposed framework. The required features are: *distributed-architecture*, *platform-independence*, *programming-language-neutrality*, *ease of use* and *the ability to cooperate with other toolkits and frameworks*. There are a few proposed frameworks that claim to realize these features, but the problem with those designs is that they use uncommon software technologies, especially in the field of robotics, that increase the complexity of the system and they are difficult to use for programming and developing robotic software. Other frameworks only realize a subset of the aforementioned features. The design proposed in this work is simple and very easy to use and effectively overcomes the challenges.

The proposed design is similar to the one presented in [2], but that design lacks the feature of supporting multiple programming-languages and it solely relies on the C++ language for development purposes. We introduce some changes to the design and create a new and improved system. Moreover we provide some additional tools with graphical user interfaces that improve the system's usability and functionality.

To meet the aforementioned requirements, the system was designed based on message-passing among software components. The system's middle-ware receives data from the application, adds a header consisting of some information about the data and sends it to the receiver using low-level mechanisms. To achieve platform-independence and language-neutrality, the communication mechanism must be supported by almost all operating-systems and programming-languages. Such mechanisms are the standard computer network protocols. The TCP protocol was chosen, because it's a connection-oriented protocol and it's essentially loss-less which leads to simpler parsers at the receiver [2].

As mentioned above, different components communicate with each other through message-passing. We call each component a "*service*". Each service is specialized for a certain task in the robot's software. Services listen for incoming messages and after receiving one, they process and respond appropriately, according to the message's content. The robot's software consists of several services, which they are distributed among some computers to run and the computers are connected by a network. To achieve programming-language-neutrality, modularity is enforced through the operating system process model: each service executes as a process on some CPU [2].

For the initiation of service execution, two programs named "*Launched-Pad*" and "*Service-Launcher*" were developed. The task of these programs is to automatically execute all desired services on different computers in the system. To do so, the configuration of the system must be described for the Launch-Pad program. This description is presented in the form of an XML file. The Launch-Pad program, has a graphical user interface and allows to initiate the execution of the entire system by the press of a button. This is possible by the help of the Service-Launcher program. The Service-Launcher must be installed on every computer in the system. The Service-Launcher constantly listens on a specific TCP port and waits for execution requests from a Launch-Pad program. Fig. 2 is an illustration of this process.

Using this framework, we were able to use almost any hardware and software we desired; ranging from custom-built electronics to high-quality manufactured devices and ranging from our own developed services to third-party robotic toolkits. Currently we are using a combination of Windows and Linux computers and the services are developed by multiple programming-languages, such as Java, C++ and C#. We were also able to easily integrate some third party programming toolkits in the robot's software.

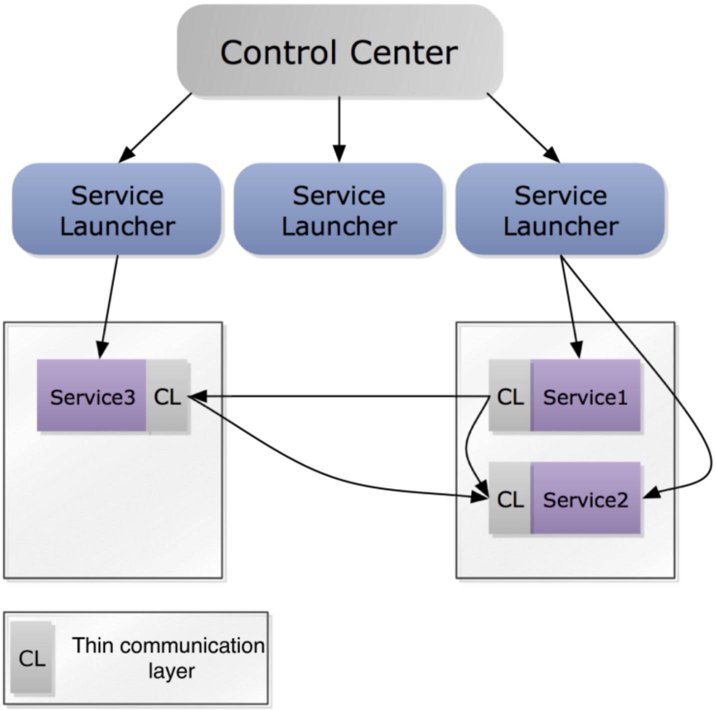

Fig. 2. The framework's architecture

3.2 Speech

For speech synthesis, we use CMU's Festival, which offers a general framework for building speech synthesis systems and it offers full text to speech through a number of APIs. For speech recognition, we use Microsoft's Speech SDK which is free software and includes a simple and reliable speech recognition system.

3.3 Human Tracking

Human tracking is an important issue in robot vision. There are lots of real-time applications including autonomous mobile service robots, human-robot interaction and perceptual user interfaces which require human tracking. Present tracking algorithms regularly apply color based methods [3], [4]. These methods have low performance in cluttered background and chaotic environments. To overcome this problem we used an algorithm based on intensity and range image fusion in combination with SIFT [5]. Initially Camshift [6] is applied to carry out similarity search via color histograms. Then peak analysis of disparity histogram of range images [7] are used to determine the region of interests (ROI) across frames. Then features extracted from the target are searched within the selected region to locate the human. This mutual support mechanism can lead to robust human tracking. We have implemented our method using the Bumblebee-2 stereo-vision system mounted on Sourena for real-time tracking of humans. Experimental results show that the proposed Camshift/range data analysis algorithm improves the tracking performance of the conventional Camshift algorithms in a complex scenario.

3.4 Face Detection and Recognition

Many services provided by robots (e.g. human recognition) depend on face detection. We employ a novel face detection algorithm which uses depth data to improve efficiency of a boosted classifier on 2D data for reduction of false positive (FP) alarms. We use two levels of cascade classifiers (see Fig. 3). The classifiers of the first level work on 2D data and classifiers of the second level use depth data captured by our stereo camera. Each camera frame is divided to many sub windows. The first level employs conventional cascade of boosted classifiers which eliminates many non-faces sub-windows. The remaining sub-windows are used as input for the second level. The second level calculates the corresponding depth model of the sub-windows. A heuristic classifier in combination with Linear Discriminant analysis (LDA) classifier is applied on the depth data to reject remaining non-face sub-windows. In the heuristic classifier, we check the validity of a simple fact. The fact is that near faces will have a bigger size and far faces will have smaller sub-window size.

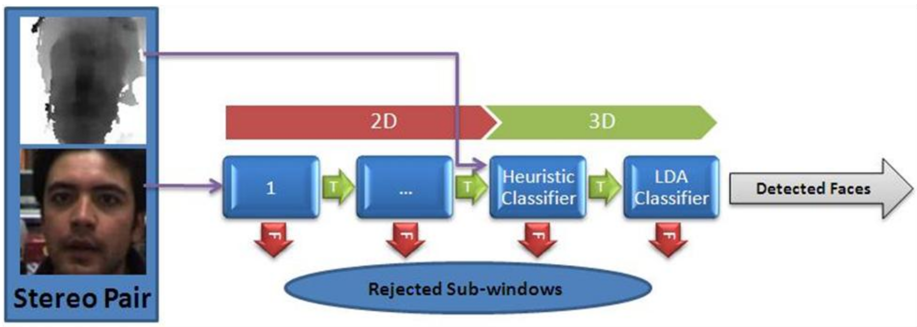

Fig. 3. Two levels of cascades classifiers used for face detection. 2D data is fed into a boosted classifier. After completion of the 2D layer, corresponding depth model of all remaining sub-windows are computed. These depth models are used as the input to the 3D classifiers. Sub-windows which verify through 3D classifiers would be determined as face.

We implemented our method using the Bumblebee-2 stereo vision system on Sourena for real-time detection of human faces in natural cluttered environments. Using this method, false positive alarm of 2D face detection is significantly reduced.

A Principal Component analysis (PCA) classifier in conjunction with a multi-layer Perceptron (MLP) is applied for the recognition of localized faces. This algorithm is able to incrementally learn new samples which enables the robot to recognize new people. Invariance to change in illumination is achieved using local normalization mechanism [8].

3.5 Object Recognition

Object Recognition is a major challenge in @Home competitions. We use combination of SIFT [5] and SURF [9] local features to find the object. Our experiments imply that SIFT is much robust than any other local feature algorithm available, but it costs a lot of resources. On the other hand SURF is much faster than SIFT. As a result, we came up with a hybrid solution to achieve improved object recognition. For training phase we simply extract object features and store them as object properties among other special features. For recognition, a series of steps are taken to extract objects available in the scene for further processing. Description of each object in the scene is calculated, and matched against the objects database. A mutual match is ensured by several measurements.

3.6 Object Manipulation

For manipulation purposes, after the process of object recognition, coordinates of the target object are obtained by a visual object tracking system and the coordinates are then given to the inverse kinematics system of the arm, that guides the gripper to approach the target object to grab it.

3.7 Localization, Navigation and Mapping

Any mobile robot should be able to navigate in its target environment. When it comes to home robots, this task will become more vital and challenging considering the complexity of the environment and safety requirements for such robots. Considering the navigation task of a home robot, it is clear that it should be able to build a map of its environment; and to make this map usable, it should be able to localize itself in the map. Therefore, these areas should be treated with special care.

Sourena is equipped with a Lase laser scanner and also a stereo camera. Last year our focus was to make a navigation system up and running using CARMEN Robot Navigation Toolkit[1] with the laser range scanner of Sourena. For this year, we are considering a shift to Mobile Robot Navigation Toolkit (MRPT)[2] which has a more active upstream and is potentially more suitable for our needs. As an additional positive point, this library better integrates into our robot's software framework.

Additionally, we are investigating the use of visual information to extend the capabilities of our navigation, localization and mapping system to allow the robot to navigate more effectively in home-like environments.

---

[1] http://carmen.sourceforge.net

[2] http://babel.isa.uma.es/mrpt/index.php/Main_Page

4 Novel Systems

Sourena includes some novel sensors that are not yet required in the @Home competitions. These systems include an electronic nose and a sound source localization system. These systems are among the many research topics of our laboratory which currently they are implemented and used in experiments.

For additional information on novel research topics in our laboratory, refer to the "*Research*" section of the team website (http://ceit.aut.ac.ir/athome).

5 Conclusion

In this paper the software and hardware characteristics of the Sourena robot were introduced. In order to succeed in the RoboCup@Home league, we provided Sourena with capabilities such as speech recognition, object recognition, face detection and recognition, navigation and obstacle avoidance, map-generation and self-localization, object manipulation, etc. With these abilities and an improved set of hardware, hopefully Sourena will have a good demonstration in the 2011 @Home competitions.

References

1. @Home Technical Committee: RoboCup@Home Rule-book 2009. RoboCup@Home Official Website (2009)
2. M. Quigley, E. B.: STAIR: Hardware and Software Architecture. Advancement of Artificial Intelligence (AAAI) Robotics Workshop (2007)
3. Comaniciu D., Ramesh V., Meer P.: Kernel-based object tracking. In: PAMI, 25(5):564–577 (2003)
4. Ramanan D., Forsyth D.: Finding and tracking people from the bottom up. In: ICCV03, pages II: 467–474 (2003)
5. Lowe, D.: Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2) (2004) 91–110
6. Bradski, G., Kaehler A.: Learning OpenCV. Oreilly Press (2008).
7. WANG, J.-G., E. T. LIM, et al.: Real-time Stereo Face Recognition by Fusing Appearance and Depth Fisherfaces. VLSI Signal Processing 49: 409–423 (2007)
8. Xie, X. and K.-M. Lam: An efficient illumination normalization method for face recognition. Pattern Recognition Letters 27: 609–617 (2006)
9. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Computer Vision and Image Understanding 110(3) (2008) 346–359