Embedded and Integrated Audition for a Mobile Robot

S. Brière, D. Létourneau, M. Fréchette, J.-M. Valin, F. Michaud
Université de Sherbrooke
Sherbrooke (Québec) CANADA J1K 2R1
{simon.briere,dominic.letourneau,maxime.frechette,jean-marc.valin,francois.michaud}@USherbrooke.ca

Abstract

"Aurally Informed Performance' for mobile robots operating in natural environments brings difficult challenges, such as: localizing sound sources all around the robot; tracking these sources as they or the robot move; separate the sources as a pre-processing step for recognition and processing, in real-time; dialogue management and interaction in crowded conditions; evaluating performances of the different processing components in open conditions. In this paper, we present how we address these challenges by describing our eight microphone system for sound source localization, tracking and separation, our on-going work on its DSP implementation, and the use of the system on Spartacus, our mobile robot entry to AAAI Mobile Robot Competitions addressing human-robot interaction in open settings.

Introduction

To deal with the problems of auditory scene analysis on a mobile robot, we have developed a system that uses an array of eight microphones to localize, track and separate sound sources, in real-time and in noisy and reverberant environments (Valin, Michaud, & Rouat 2006; Valin, Rouat, & Michaud 2004). The system was first validated on Spartacus, a robot participant to the AAAI 2005 Mobile Robot Challenge, making the robot attend the conference as a regular attendee (Michaud *et al.* 2005). Spartacus uses three onboard computers to implement its software architecture, with one dedicated to the audition capabilities of the robot. Software integration is done using MARIE, our middleware framework for distributed component processing (Cote *et al.* 2006). Using its audition skills, Spartacus is able to position its pan-tilt-zoom camera in the direction of sound sources, to navigate by following a sound source, to understand specific vocal requests (using NUANCE[1]) and to respond accordingly (using a preprogrammed dialogue manager and Festival for speech generation).

[1]http://www.nuance.com

Since then, we have pursued our work to extend the capabilities of the system by: 1) porting the audio extraction and sound source separation algorithms on a floating-point DSP (Digital Signal Processor), providing specialized processing resources to free the onboard computer for other decisional components, and also allowing the system to be used on different robots and in various settings; and 2) improving the audition modalities on the robot for more natural interaction with people. In this paper, we present these two trusts and explain how Spartacus audition modalities (localization, tracking and separation of sound sources; dialogue management and interaction) have improved for its participation to AAAI 2006 Mobile Robot Open Interaction competition (held in July 2006).

DSP Implementation of our Sound Source Localization, Tracking and Separation (SSLTS) System

Our SSLTS system is represented in Figure 1. It uses an array of eight microphones, spatially positioned on the front and the back of the robot's torso. The system is made of three main modules: the localization module, the tracking module and the sound separation module.

The localization module is used to localize multiples sources in the environment surrounding the robot. It uses a steered beamformer (Valin *et al.* 2004; Valin, Michaud, & Rouat 2006) approach (also known as SRP-PHAT) to find the position of the sources. By computing the weighted cross-correlation between each microphone pair in the system (28 pairs when using 8 microphones), the beamformer determines the direction of arrival of multiple sources using 1024-sample frames at a 48 kHz sampling frequency.

When the source positions are found, the tracking module is used to follow moving sources. By using particle filters (Arulampalam *et al.* 2002) for each source already being tracked, the system updates the state (position and velocity) of the detected sources as they move based on a probabilistic representation of the problem. This module also manages which sources are being tracked by determining if the localized source is a real source (not a false detection) and if the

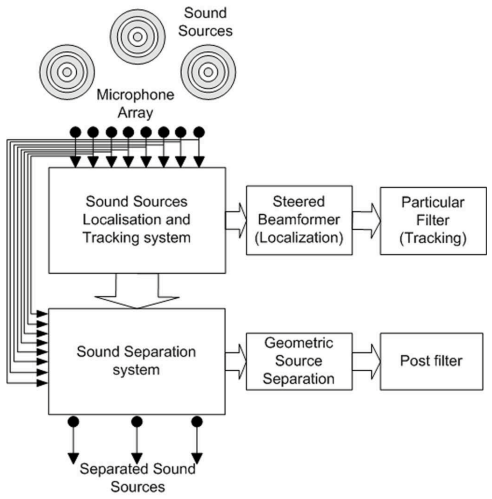

Figure 1: Block diagram of our SSLTS system.

sources already tracked are still present.

The information provided by the tracking module is used by the sound separation module to separate the detected sources. The separation is necessary for analyzing the audio content of a single source, for example when using a speech recognition system. In order to separate the audio sources, the separation module uses an modified version of the geometric sound source (GSS) separation algorithm. The modifications to the GSS were made necessary to ensure that the algorithm could be used in real-time. A multi-source post-processing algorithm is used to remove interference and noise from the GSS-separated audio streams. The post-processing is optimized for speech recognition accuracy (and not for human audibility quality).

Evaluation and usability of this system will be limited by its portability. For instance, a small mobile robot to be used to study human-robot interaction with autistic children (Michaud, Duquette, & Nadeau 2003) needs to be small and cannot require computer-intensive processing. Currently, the system requires 25% of the processing power of a 1.6 GHz Pentium M laptop for the localization and tracking modules (Valin, Michaud, & Rouat 2006) and 25% of the same processor for the separation module. So that is why we ported the implementation of these modules on a DSP, as explained in the following subsection.

DSP Implementation

Our SSLTS-DSP implementation is done on a TMS3206C713 Texas Instruments DSP. This processor is a floating-point 225 MHz processor with 256 kbytes of internal RAM memory with L1 and L2 cache support. According to the specifications, the processor is rated at 225 MIPS and 1500 MFLOPS, and its architecture is optimized for audio processing. The DSP is built on a Lyrtech[2]) CPA-II board shown in Figure 2. This board has 24 bits analog-to-digital converters (ADC) able to run at frequencies from 32 kHz to 192 kHz. These ADC are used to capture the microphone outputs. The board also provides 64 Mbytes of external memory (SDRAM) and 8 Mbytes of flash ROM memory. It also provides an USB2 interface that can be used to control the system and to transfer the separated sound sources to another computer for post-processing (e.g., speech recognition).

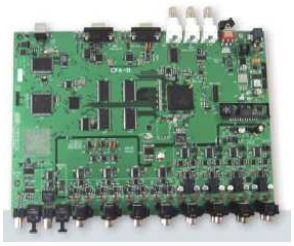

Figure 2: Lyrtech CPA-II, our development board.

The transfer of the system on a dedicated processor is motivated by some reasons. First of all, we want to make a system that is better suited to robotics applications in term of power consumption and size than a laptop. Having an external board removes particular software and OS dependency, thus allowing the system to be used in many different configurations and systems. Because the processing would be done by a different processor than the embedded robot one, we will have more CPU time available to other complex algorithm (e.g. image recognition). We also hope that such a system will cost less than a laptop.

Work is currently in progress to port the actual system on the DSP board. The first phase of the project was to port the system from a Linux C++ based environment to a TI DSP C-based environment. Using recorded microphones outputs, we were able to obtain similar results between the DSP and the original system. We used eight simulated microphones inputs at a sampling frequency of 48 kHz with an overlap of 50% and 1024 sample frames. However, there was no memory management and the code was not optimized for DSP processing, which uses Very Large Instruction Words (VLIW) for parallel processing.

To meet the original specifications of the system, the processing needs to be done under 21.33 ms for a frame size of 1024 samples frames with no overlap, and 10.66 ms for a frame size of 1024 samples with 50% overlap which is the case in the original system. We are trying to build the system using similar parameters as the original system: four simultaneous sources at 48 kHz and the same localization resolution. Currently, without optimization, the localization and tracking modules on the DSP require between 13 ms and 106 ms of processing cycle time, depending on the number of sound sources being tracked. The separation module requires from 5 ms (for one sources) and 20 ms (for four sources) of the DSP processing cycle time. The worst case scenario is then 126 ms, which is about 12 times slower than the real-time requirement of 10.66 ms.

We are currently working on different solutions to improve the system processing cycle time, and the optimiza-

[2]http://www.lyrtech.com

tion process is currently on-going. One is to make the necessary changes to the code in order to take advantage of the DSP parallel architecture. The DSP optimization tools are helping us pinpoint critical elements in the processing done by the DSP. Improving memory management will also bring improved results. The SSLTS has high memory requirements because it uses eight microphones, with 1024 sample frames for each one, represented using floating point complex numbers. The DSP external memory is slow compared to its internal memory, and a more efficient use of the internal memory may substantially improve the processing cycle time. However, it appears evident at this point that it will be impossible to build the original system on the DSP with the exact same parameters, because of memory limitations. The algorithm uses an array of 71736 floating-point elements in order to perform an acurate localization on a 2562 point grid around each microphone. Each elements in the array needs to be accessed one at the time in a loop, making the benefit of using the cache minimal because the values are read but not reused afterward. Because the array is about 280 kilobytes in size and that we have 256 kilobytes of internal memory on the processor, we have to put the array in external memory, resulting in slower performance. To successfully build the system on that particular processor, we probably will have to reduce sampling rate, reduce the resolution of the localization system and reduce the maximum number of simultaneous sources in order to reach the targeted time requirements.

Dialogue Management and Integration on a Mobile Robot

Spartacus integrates planning and scheduling, sound source localization, tracking and separation, message reading, speech recognition and generation, and autonomous navigation capabilities onboard a custom-made interactive robot. Integration of such a high number of capabilities revealed interesting new issues such as coordinating audio/visual/graphical capabilities, monitoring the impacts of the capabilities in usage by the robot, and inferring the robot's intentions and goals. Spartacus development is ongoing and aims at adding new capabilities to the robot and improving our software and computational architectures, addressing issues of human-robot interaction and the integrated challenges of designing an autonomous service robot. Reported issues are: 1) the robust integration of different software packages and intelligent decision-making capabilities; 2) natural interaction modalities in open settings; 3) adaptation to environmental changes for localization; and 4) monitoring/reporting decisions made by the robot (Gockley *et al.* 2004; Smart *et al.* 2003; Maxwell *et al.* 2004; Simmons *et al.* 2003).

For our participation to the AAAI 2006 Mobile Robot Competition, Spartacus is designed to be a scientific robot reporter, in the sense of a human-robot interaction research assistant. The objective is to have Spartacus provide understandable and configurable interaction, intention and information in unconstrained environmental conditions, reporting the robot experiences for scientific data analysis. Interactions occurring in open settings are rapid, diverse and context-related. Having an autonomous robot determining on its own when and what it has to do based on a variety of modalities (time constraints, events occurring in the world, requests from users, etc.) also makes it difficult to understand the robot's behavior just by looking at it. Therefore, we concentrated our integration effort this year to design a robot that can interact and explain, through speech and graphical displays, its decisions and its experiences as they occur (for on-line and off-line diagnostics) in open settings.

The robot is programmed to respond to requests from people, and with only one intrinsic goal of wanting to recharge when its energy is getting low (by either going to an outlet identified on the map or by searching for one, and then ask to be plugged in). Spartacus' duty is to address these requests to the best of its capabilities and what is 'robotically' possible. Such requests may be to deliver a written or a vocal message to a specific location or to a specific person, to meet at a specific time and place, to schmooze, etc. The robot may receive multiple requests at different periods, and will have to manage on its own what, when and how it will satisfy them.

Spartacus is shown in Figure 3. Spartacus is equipped with a SICK LMS200 laser range finder, a Sony SNC-RZ30N 25X PTZ color camera, the microphone array placed on the robot's body, a touchscreen interface, an audio amplifier and speakers, a business card dispenser and a LEDs electronic display. High-level processing is carried out using an embedded Mini-ITX computer (Pentium M 1.7 GHz), and two laptop computers (Pentium M 1.6 GHz) installed on the platform. One is equipped with a RME Hammerfal DSP Multiface sound card using eight analog inputs to simultaneously sample signals coming from the microphone array. It is also connected to the audio amplifier and speakers using the audio output port. The other laptop does video processing and is connected to the camera through a 100Mbps Ethernet link. Communication between the three on-board computers is accomplished with a 100Mbps Ethernet link. All computers are running Debian GNU Linux.

Figure 4 illustrates Spartacus' software architecture. It integrates Player for sensor and actuator abstraction layer (Vaughan, Gerkey, & Howard 2003), and CARMEN (Carnegie Mellon Robot Navigation Toolkit) for path planning and localization (Montemerlo, Roy, & Thrun 2003). Also integrated but not shown on the figure is Stage/Gazebo for 2D and 3D simulators, and Pmap library[3] for 2D mapping, all designed at the University of Southern California. RobotFlow and FlowDesigner (FD) (Cote *et al.* 2004) are also used to implement the behavior-producing modules and SSLTS. For speech recognition and dialogue management, we interfaced this year the CSLU toolkit[4]. Software integration of all these components are made possible using MARIE, a middleware framework oriented towards developing and integrating new and existing software for robotic systems (Cote *et al.* 2006). NUANCE speech recognition software is also available through MARIE.

[3] http://robotics.usc.edu/ ahoward/pmap/

[4] http://cslu.cse.ogi.edu/toolkit/

Figure 3: Spartacus (front view, back view), with the microphones located near the *.

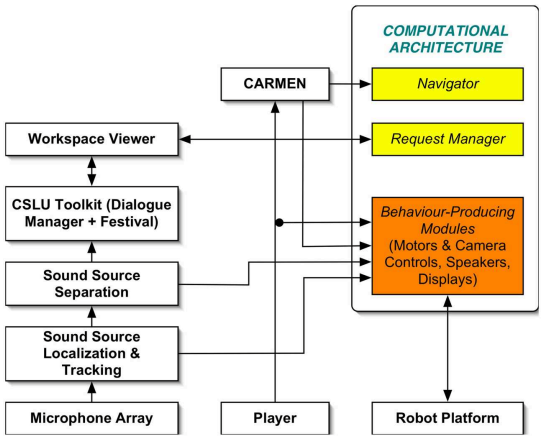

Figure 4: Spartacus software architecture.

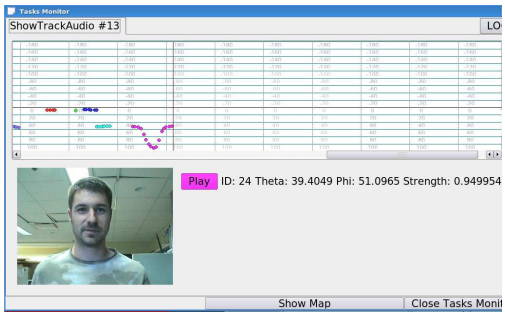

Figure 5: Track audio interface.

Regarding the auditory and interaction capabilities of the robot, we came up with a set of modalities to improve our understanding of integrated modalities (vision, audition, graphical, navigation), as follows:

- Visualization of the SSLTS results. In 2005, the SSLTS system was implemented on Spartacus and was directing the camera toward the loudest sound source around the robot. To fully represent what the robot is hearing, we needed a more appropriate interface. Figure 5 illustrates the interface developed. The upper part shows the angle of the perceived sound sources around the robot, in relation to time. The interface shows in real-time the sound sources perceived, using dots of a distinct color. The sound sources are saved, and the user can select them and play back what the SSLTS generated. This interface reveals to be very valuable for explaining what the robot can hear, to construct a database of typical audio streams in a particular setting and to analyze the performance of the entire audio block (allowing to diagnose potential difficulties that the speech recognition module has to face). More than 6600 audio streams were recorded over the twelve hours of operation of the robot at the AAAI 2006 conference, held at Seaport Hotel and World Trade Center in Boston. By constructing such database of audio streams, we will be able to evaluate in a more diverse set of conditions the performance of speech recognition algorithms.
- Develop contextual interfaces and interaction capabilities according to the state and intentions of the robot. It is sometimes difficult to know exactly which task is prioritized by the robot at a given time, making the users unable to know what the robot is actually doing and evaluate its performance. So we made the robot indicate its intention verbally and also graphically (in case, for whatever reasons, the interlocutor is not able to understand the message). Figure 6 illustrates the case where Spartacus is requesting assistance to find a location in the convention center.

  The graphical interfaces of the robot were made so that users can indicate through push buttons (using the touch screen) what they want Spartacus to do. We also made it possible for the users to make these requests verbally. To facilitate the recognition process, a specific grammar and dialogue manager are loaded in CSLU for each graphical

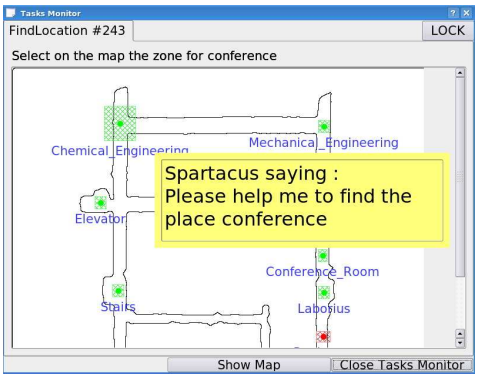

Figure 6: Graphical display for an assistance request to find a location.

mode. The audio and graphical interaction are therefore tighly integrated together.

These modalities are functional and were demonstrated at AAAI 2006 Conference, showing our implementation of integrated and dynamic representations of localization, vocal requests and menus. However, it is only a first step in coming up with the most efficient integration of these modalities. Our intents are to explore what can be learned using this integration, and to use our setup to evaluate speech recognition (e.g., NUANCE, Sphinx) and dialogue management (e.g., CSLU, Collagen) packages, to study how they can be interfaced to the visual and mobile capabilities of the robot, and to assess how a stronger coupling between SSLTS and speech recognition algorithms can help improve speech recognition. We are also interested in adding new aural capabilities (e.g., speaker identification, sound recognition), and evaluate how the robot could be able to listen while speaking.

Acknowledgments

F. Michaud holds the Canada Research Chair (CRC) in Mobile Robotics and Autonomous Intelligent Systems. Support for this work is provided by the Natural Sciences and Engineering Research Council of Canada, the Canada Research Chair program and the Canadian Foundation for Innovation.

References

Arulampalam, M.; Maskell, S.; Gordon, N.; and Clapp, T. 2002. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing* 50(2):174–188.

Cote, C.; Letourneau, D.; Michaud, F.; Valin, J.-M.; Brosseau, Y.; Raievsky, C.; Lemay, M.; and Tran, V. 2004. Code reusability tools for programming mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1820–1825.

Cote, C.; Brosseau, Y.; Letourneau, D.; Raievsky, C.; and Michaud, F. 2006. Using marie in software development and integration for autonomous mobile robotics. *International Journal of Advanced Robotic Systems, Special Issue on Software Development and Integration in Robotics* 3(1):55–60.

Gockley, R.; Simmons, R.; Wang, J.; Busquets, D.; DiSalvo, C.; Caffrey, K.; Rosenthal, S.; Mink, J.; Thomas, S.; Adams, W.; Lauducci, T.; Bugajska, M.; Perzanowski, D.; and Schultz, A. 2004. Grace and george: Social robots at aaai. Technical Report WS-04-11, AAAI Mobile Robot Competition Workshop. pp. 15-20.

Maxwell, B.; Smart, W.; Jacoff, A.; Casper, J.; Weiss, B.; Scholtz, J.; Yanco, H.; Micire, M.; Stroupe, A.; Stormont, D.; and Lauwers, T. 2004. 2003 aaai robot competition and exhibition. *AI Magazine* 25(2):68–80.

Michaud, F.; Brosseau, Y.; Cote, C.; Letourneau, D.; Moisan, P.; Ponchon, A.; Raievsky, C.; Valin, J.-M.; Beaudry, E.; and Kabanza, F. 2005. Modularity and integration in the design of a socially interactive robot. In *Proceedings IEEE International Workshop on Robot and Human Interactive Communication*, 172–177.

Michaud, F.; Duquette, A.; and Nadeau, I. 2003. Characteristics of mobile robotic toys for children with pervasive developmental disorders. In *Proceedings IEEE International Conference on Systems, Man, and Cybernetics*.

Montemerlo, M.; Roy, N.; and Thrun, S. 2003. Perspectives on standardization in mobile robot programming: The carnegie mellon navigation (carmen) toolkit. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2436–2441.

Simmons, R.; Goldberg, D.; Goode, A.; Montemerlo, M.; Roy, N.; Sellner, B.; Urmson, C.; Schultz, A.; Abramson, M.; Adams, W.; Atrash, A.; Bugajska, M.; Coblenz, M.; MacMahon, M.; Perzanowski, D.; Horswill, I.; Zubek, R.; Kortenkamp, D.; Wolfe, B.; Milam, T.; and Maxwell, B. 2003. Grace : An autonomous robot for the aaai robot challenge. *AI Magazine* 24(2):51–72.

Smart, W. D.; Dixon, M.; Melchior, N.; Tucek, J.; and Srinivas, A. 2003. Lewis the graduate student: An entry in the aaai robot challenge. Technical report, AAAI Workshop on Mobile Robot Competition. p. 46-51.

Valin, J.-M.; Michaud, F.; Hadjou, B.; and Rouat, J. 2004. Localization of simultaneous moving sound sources for mobile robot using a frequency-domaine steered beamformer approach. In *Proceedings IEEE International Conference on Robotics and Automation*, 1033–1038.

Valin, J.-M.; Michaud, F.; and Rouat, J. 2006. Robust 3d localization and tracking of sound sources using beamforming and particle filtering. In *Proceedings International Conference on Audio, Speech and Signal Processing*, 221–224.

Valin, J.-M.; Rouat, J.; and Michaud, F. 2004. Enhanced robot audition based on microphone array source separation with post-filter. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Vaughan, R. T.; Gerkey, B. P.; and Howard, A. 2003. On device abstractions for portable, reusable robot code. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2421–2427.