UM2212

User manual

Getting started with AcousticSL real-time sound source localization middleware

Introduction

The AcousticSL software lets you implement a real-time sound source localization algorithm using two or four signals acquired from digital MEMS microphones to estimate the direction of arrival of the main audio source.

It is based on three different DOA algorithms exploiting cross correlation in the time domain, generalized cross correlation with phase transform, and a matching pursuit routine using sparse representation framework.

The angle can be estimated over a 180 or 360 degree range, depending on the number of channels adopted and the microphone placement.

The resolution of the computed value can be chosen at runtime, allowing you to determine the best tradeoff between localization precision and resource consumption.

The AcousticSL library is provided both in source code and in binary format inside the X-CUBE-MEMSMIC1 software package with sample applications running on the X-NUCLEO-CCA02M2 expansion board connected to a NUCLEO-F401RE development board and on the X-NUCLEO-AMICAM1, when connected to a NUCLEO-L4R5ZI.

The AcousticSL is also part of the FP-AUD-SMARTMIC1 function pack.

The package is based on STM32Cube technology and can be easily ported to any microcontroller with an FPU.

1 Licensing information

Licensed under Software License Agreement SLA0077, (the "License"). You cannot use this package except in compliance with the License. You may obtain a copy of the License at www.st.com.

Some of the library code is based on the CMSIS DSP software library by ARM®, a suite of common signal processing functions to be used on ARM® Cortex®-M processor-based devices. Licensing terms are available in the release_note.html file included in the software package, in the next lines of this document and on the web at https://www.keil.com/pack/doc/CMSIS/DSP/html/index.html.

2 AcousticSL software library

This library uses audio signals from microphones to execute sound source localization and estimate the angle of arrival of the audio source. It embeds three algorithms with differing RAM and MIPS footprints and overall performance (Knapp, C. H. (1976). The generalized correlation method for estimation of time delay, IEEE Transactions on Acoustics, Speech and Signal Processing ASSP. 24(4), 320-327):

- XCORR cross correlation algorithm:
  - performed in the time domain
  - less resource intensive
  - lower resolution
  - requires microphones to be placed at greater distances
- GCC-PHAT algorithm
  - performed in the frequency domain
  - relatively resource intensive
  - higher resolution
  - performance should not be impacted by microphone distance
- BMPH algorithm
  - performed in the frequency domain
  - more resource demanding than XCORR, but less than GCC-PHAT
  - performance should not be impacted by microphone distance
  - performs a hangover step to enhance source stability

The basic versions of these algorithms use two audio signals to estimate an angle in a 180 degree range. A 360 degree range requires four audio streams and appropriate library configuration. The algorithm and the number of microphones are chosen by the user during library initialization. The software is designed to run in a real-time environment.

2.1 AcousticSL software library

This library uses audio signals from microphones to execute sound source localization and estimate the angle of arrival of the audio source. It embeds three algorithms with differing RAM and MIPS footprints and overall performance (Knapp, C. H. (1976). The generalized correlation method for estimation of time delay, IEEE Transactions on Acoustics, Speech and Signal Processing ASSP. 24(4), 320-327):

- XCORR cross correlation algorithm:
  - performed in the time domain
  - less resource intensive
  - lower resolution
  - requires microphones to be placed at greater distances
- GCC-PHAT algorithm
  - performed in the frequency domain
  - relatively resource intensive
  - higher resolution
  - performance should not be impacted by microphone distance
- BMPH algorithm
  - performed in the frequency domain
  - more resource demanding than XCORR, but less than GCC-PHAT
  - performance should not be impacted by microphone distance
  - performs a hangover step to enhance source stability

The basic versions of these algorithms use two audio signals to estimate an angle in a 180 degree range. A 360 degree range requires four audio streams and appropriate library configuration. The algorithm and the number of microphones are chosen by the user during library initialization. The software is designed to run in a real-time environment.

2.1.1 Angle resolution

Angle output resolution mainly depends on which algorithm is used and how the result is computed.

2.1.1.1 XCORR cross-correlation algorithm

Cross-correlation is based on time-domain computation and the resolution depends on both the sampling frequency and the distance between microphones. For each combination of these parameters, the maximum delay between two microphones (in terms of samples) is:

$$DelayMax = Floor\left(\frac{(Distance) * (Sampling\ Frequency)}{(Sound\ Speed)}\right) \quad (1)$$

This value is strictly related to the maximum number of angles that can be deduced by the routine and thus determines the output resolution.

The minimum detectable angle over 180 degrees is:

$$Minimum\ Detectable\ Angle = Floor\left(\frac{180}{2*DelayMax}\right) \quad (2)$$

The larger the microphone distance, the higher the possible resolution, at a higher MIPS cost. If you choose this algorithm, the initialization function automatically computes the achievable resolution. In this case, the initialization function ignores the resolution parameter set by the user.

2.1.1.2 GCC-PHAT algorithm

The GCC-PHAT algorithm works with frequencies and generally offers a better resolution even in low intermicrophone spacing configurations. You can set the desired resolution with a dedicated parameter in the initialization structure (see the chm help file in the Documentation folder). In this case, the higher the resolution, the higher the MIPS consumption.

Note: *The resolution value is related to the use of the library with two channels; when the library is used with four channels to perform source localization over a 360 degree range, two detected values are merged internally and the overall resolution may be different from the initial setting.*

2.1.1.3 BMPH algorithm

The block-matching pursuit with hangover (BMPH) algorithm also works with frequencies. You can set the desired resolution with a dedicated parameter in the initialization structure.

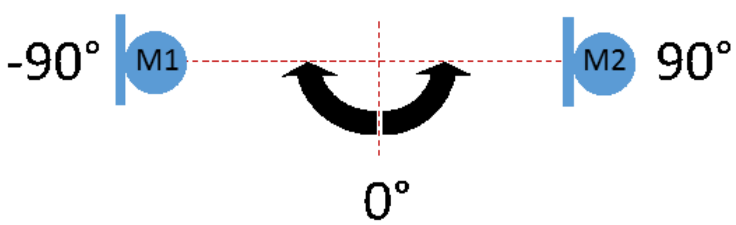
The best resolution available for the BMPH algorithm is four degrees. While MIPS consumption and memory requirements of the BMPH algorithm are generally lower than for the GCC-PHAT algorithm, the algorithms exhibit similar performance as the algorithm adaptively analyzes the most relevant frequencies in the frame.

2.1.2 Microphone geometry

2.1.2.1 Two-microphone scenario

The figure below shows the plan view of M1 and M2 microphones.

Figure 1. Two-microphone arrangement



The minimum inter-microphone spacing depends on the algorithm:

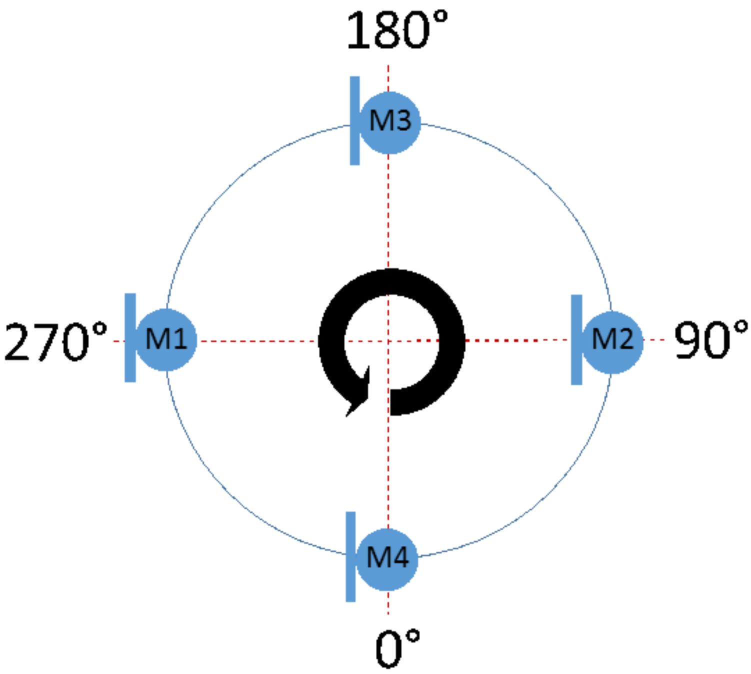
- for XCORR, the distance must allow at least one sample delay between them;
- GCC-PHAT has potentially no distance constraints.

The estimated angle is measured by taking as a reference the imaginary line that passes through the middle point of the segment, which joins the two microphones and is orthogonal to it.

2.1.2.2 *Four-microphone scenario*

The figure below shows the arrangement of M1, M2, M3 and M4 microphones. There are the same distance constraints of the two-microphone scenario, but measured between opposite microphones, that is, between M1-M2 and between M3-M4.

Figure 2. Four-microphone arrangement



3 Library profiling

Profiling helps to evaluate the library resource consumption in terms of MIPS, RAM, and Flash. You can find detailed information in the AcousticSL_Package.chm compiled HTML file of the Documentation folder.

4 References

- Knapp, C. H. (1976). The generalized correlation method for estimation of time delay, IEEE Transactions on Acoustics, Speech and Signal Processing ASSP. 24(4), 320-327.

Revision history

Table 1. **Document revision history**

Date	Version	Changes
18-May-2017	1	Initial release.
26-Oct-2021	2	Replaced references to X-NUCLEO-CCA02M1 expansion board with X-NUCLEO-CCA02M2 expansion board. Minor text changes throughout the document.

Contents

List of tables

List of figures