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 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 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 in binary format inside a software package with sample applications running on the X-NUCLEO-CCA02M1 expansion board connected to a NUCLEO-F401RE board.

The package is based on STM32Cube technology and is easily ported to any STM32F4 microcontroller with an FPU.

Information regarding STM32Cube is available on www.st.com at http://www.st.com/stm32cube.

Contents

List of figures

1 Licensing information

Some of the library code is based on the CMSIS DSP software library by ARM, a suite of common signal processing functions for use 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

http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html.

ARM license note:

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[a]:

- 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 Angle resolution

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

2.1.1 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:

$$DelayMax = Floor\left(\frac{(Distance) * (SamplingFrequency)}{(SoundSpeed)}\right)$$

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

The minimum detectable angle over 180 degrees is:

$$MinimumDetectableAngle = Floor\left(\frac{180}{2 * DelayMax}\right)$$

[a] 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.

The larger the microphone distance, the higher the possible resolution, at a higher MIPS cost. If this algorithm is chosen, the achievable resolution is computed automatically by the initialization function and cannot be set by the user. In this case, the resolution parameter is ignored by the initialization function.

2.1.2 GCC-PHAT algorithm

The GCC-PHAT algorithm works with frequencies and generally offers a better resolution even in low inter-microphone spacing configurations. The user 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.

The resolution value relates 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 to the initial setting.

2.1.3 BMPH algorithm

The block matching pursuit with hangover (BMPH) algorithm also works with frequencies and the user 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 because the algorithm adaptively analyzes the most relevant frequencies in the frame.

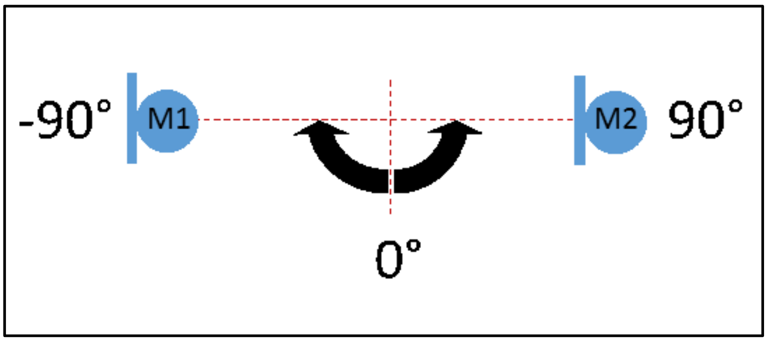
2.2 Microphone geometry

2.2.1 Two-microphone scenario

Figure 1: "Two-microphone arrangement" shows the plan view of microphones M1 and M2. 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

Figure 1: Two-microphone arrangement

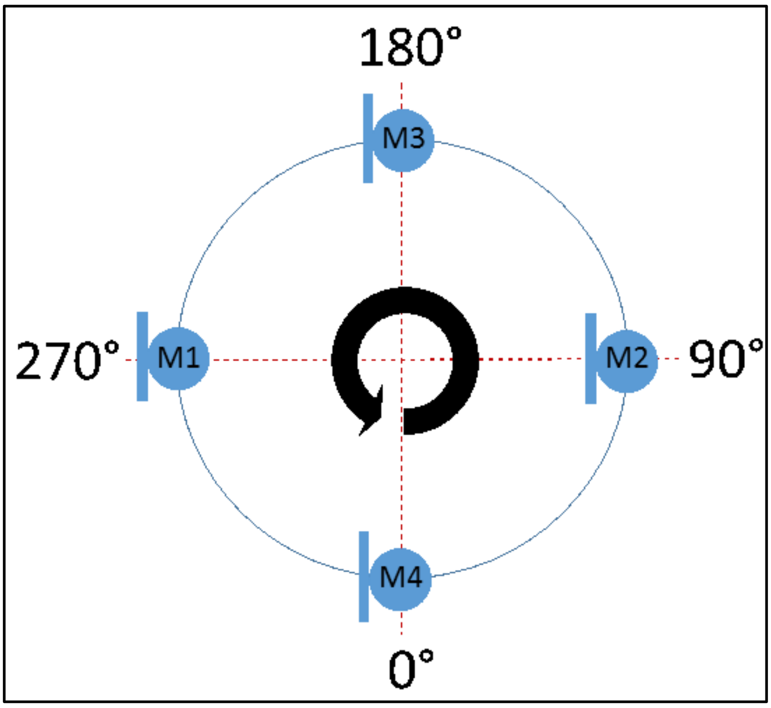


The estimated angle is measured taking as a reference the imaginary line passing through the middle point of the segment joining the two microphones and orthogonal to it, as depicted in the picture.

2.2.2 4-microphone scenario

Figure 2: "Four-microphone arrangement" shows the arrangement of microphones M1, M2, M3 and M4. The same distance constraints as in the two-microphone scenario must be observed, but they are measured between opposite microphones; i.e., between M1-M2 and between M3-M4.

Figure 2: Four-microphone arrangement



3 Library profiling

Profiling was performed in order to evaluate the library resource consumption in terms of MIPS, RAM and Flash. Detailed information can be found in the AcousticSL_Package.chm compiled HTML file located in the Documentation folder.

4 References

1. 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.
2. User manual, UM1900 - Getting started with the digital MEMS microphone expansion board based on MP34DT01-M for STM32 Nucleo. STMicroelectronics

5 Revision history

Table 1: Document revision history

Date	Version	Changes
18-May-2017	1	Initial release.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.