PYRAMIC: FULL STACK OPEN MICROPHONE ARRAY ARCHITECTURE AND DATASET

Robin Scheibler[†]*, Juan Azcarreta*[‡]*, René Beuchat*[‡]*, and Corentin Ferry*[♯]

[†]Tokyo Metropolitan University, Tokyo, Japan
[‡]École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[♯]Univ Rennes, F-35000 Rennes, France

ABSTRACT

In this paper we introduce an open source and reproducible microphone array hardware design and an anechoic dataset recorded with this array. The Pyramic array has 48 microphones spread onto six identical modules connected to an FPGA-ARM combo. The arrangement of the six modules can be reconfigured to create a large number of geometries. We describe in detail the architecture of the array and make openly available all necessary hardware design files, VHDL code, and C libraries together with extensive documentation. This effectively enables replicability of part or all of the array.

The curated dataset of anechoic measurements done using the Pyramic array comprises source locations with dense azimuth sampling at multiple heights, playing both test and speech signals. The manual calibration of source and microphone locations is assessed and improved upon using time-difference of arrival methods. The array response to each source location is also provided. Finally, the dataset is used to assess the performance of two well-known direction of arrival estimation algorithms on the Pyramic architecture.

***Index Terms*—** Microphone array, FPGA, calibration, dataset, reproducibility.

1. INTRODUCTION

Microphone arrays have gained a lot of attention for audio processing due to their ability to leverage spatial cues from acoustic signals. Extensive literature has been written about microphone arrays [2, 3] for applications such as source localization [4], beamforming [5] and source separation [6, 7] among others. In addition, many techniques presented to recent speech processing research evaluation campaigns [8, 9] have focused on multichannel solutions. However, the development of novel applications of microphone arrays, such as Internet of Things (IoT) devices and wearables [10], requires to spend precious time building dedicated embedded platforms. An alternative is to have access to recordings from an existing platform suiting the application.

Many microphone array designs have been proposed, with varying trade-offs between number of microphones, processing capabilities and portability of the array. The Large acoustic Data Array [11], LOUD, is a 1020-microphones planar array with real-time processing capabilities. Nevertheless, such a large array lacks the flexibility to be tested and record datasets in different environments. To overcome these limitations, [12] combines microelectromechanical systems (MEMS) microphones and a Field Programmable Gate Array (FPGA) platform to create a compact and real-time 128-channel array for robotic applications. To the best of our knowledge, neither documentation nor code is available to build either of these systems. The Bela platform [13] is an exemplary open architecture, but does not scale up to more than 10 microphones.

At the same time, data collection efforts have led to a variety of audio datasets being released, especially targeting speech processing applications [14]. Table 1 summarizes some of the existing audio datasets according to the type of recording device, the total data length and the conditions of the recordings. While there exists many audio datasets, most are recorded with planar microphone array architectures in reverberant environments and near planar source placements.

We release a full stack design of the Pyramic microphone array [15, 16] and a curated dataset of 3D sources recorded with this array [1]. The proposed array has a modular architecture and is composed of 48 MEMS microphones spread across its edges, which form a tetrahedron. The system sends samples to an FPGA-ARM combo capable of real-time standalone processing of the sampled signals.

The main contributions of this paper are twofold. First, it provides a comprehensive reference design for the Pyramic array, including bill-of-material, hardware design files, VHDL and C code along with an extensive documentation. Having the full design available allows not only reproducibility of the exact same array, but also its modification to create new specialized architectures. For example, FPGAs are both necessary to scale up the number of channels beyond the typical eight microphones, and notoriously arduous to develop for. A reference implementation available under a permissive license can save up to hundreds of hours of development time. Second, we provide a curated dataset of anechoic recordings of audio sources. Through carefully recorded samples and impulse responses, such a dataset makes available the Pyramic architecture without having to build, buy, or even use the hardware itself. Thus, providing users at all levels, from hardware to algorithm designers, with a *full stack*, open, and reproducible microphone array architecture.

According to best practices and to maximize potential reuse, all the software (Python, C, and VHDL code) is released under MIT license. The data is released under CC-BY 4.0. Finally, the hardware is under a CC-BY-SA 4.0 license. Two notable caveats are the use of Altium for schematics and board development which can cause some problems [17] under academic license (we had a commercial one), and the VHDL which relies on some proprietary IPs from Altera (e.g. FIFO). The latter is not a problem in practice as these IPs are available at no cost through the Altera toolchain.

We demonstrate the performance of the Pyramic array architecture by showing the results for the well-known SRP-PHAT [18] and MUSIC [19] algorithms for direction of arrival. In addition, we display manual and optimized source and microphone locations calibra-

The research presented in this paper is reproducible. Code and design files are available at `http://github.com/LCAV/pyramic`. The dataset is available at `zenodo` [1] and `https://github.com/fakufaku/pyramic-dataset`.

978-1-5386-8151-0/18/$31.00

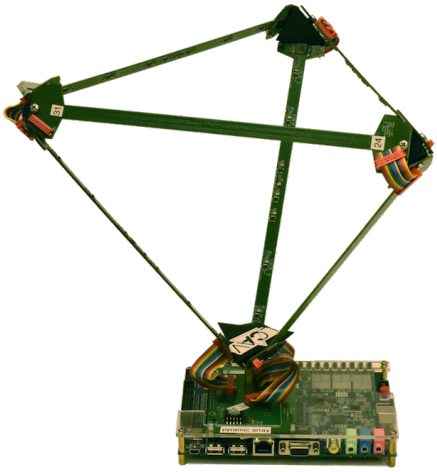


Fig. 1: The assembled Pyramic microphone array.

tion results that reduce the channel sensitivity of the array [20]. As well as this, miscellaneous applications for the Pyramic array [21] are discussed. This work continues our previous efforts to release open-source software libraries to reduce the overhead of deployment multichannel signal processing algorithms [22].

The rest of this paper is organized as follows. Section 2 covers the architecture of the full Pyramic microphone array stack, from the hardware and FPGA designs to the C application structure. Section 3 describes the curated dataset recorded with the Pyramic array. Section 4 presents the main applications of the array. Section 5 concludes the paper.

	# mics	Type	Length	Conditions
MMA [23]	40	Planar	10 h	Meeting
AMI [24]	4/8	Planar	100 h	Meeting
DEMAND [25]	16	Planar	22 h	Noise
IDIAP [26]	12	Planar	1.5 h	Meeting
CMU [27]	15/8/46	Planar	N/A	Noisy speech
CHiME-4 [28]	6	Planar	62.6 h	Noisy
Pyramic [1]	48	3D	2.5 h	Anechoic

Table 1: Brief comparison of some publicly available audio datasets.

2. MICROPHONE ARRAY ARCHITECTURE

The microphone array has a semi-modular design striking a balance between manufacturability, modularity, and ease of reconfiguration. The basic element is a linear sub-array of eight microphones. Up to two sub-arrays can be daisy-chained into a longer array, thus reducing the number of connections necessary with the processing unit.

While an arbitrary number of sub-arrays could be used, we have limited ourselves to six, thus forming a 48-channel array. We have chosen to arrange these sub-arrays to form a tetrahedron with the microphones lying on its edges. Because this shape looks pyramidal, we chose to name the microphone array thus formed Pyramic. A picture of the array is shown in Figure 1.

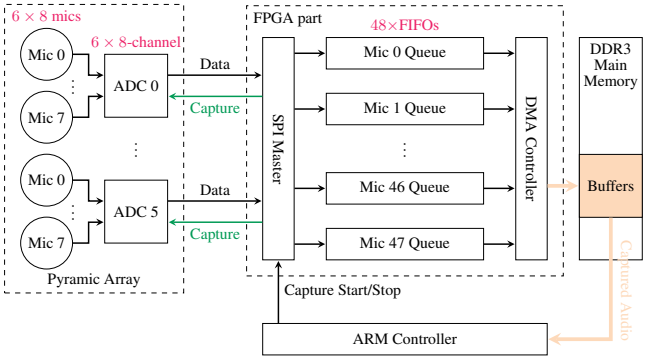


Fig. 2: Architecture and data flow of the Pyramic synchronous capture.

2.1. Hardware Architecture

Each sub-array is a 27 centimeters long printed circuit board (PCB) sporting eight analog INMP5404 MEMS microphones. On a single PCB, six out of eight microphones form a uniform linear array with the remaining two spaced by 8 mm at the center, thus avoiding spatial aliasing up to 21 kHz [29]. Figure 3 depicts one of the sub-arrays with the microphone locations. Each sub-array has an AD7606 analog to digital converter (ADC) from Analog Devices, which samples the eight microphones synchronously at 48 kHz and 16 bits, and communicates with the host FPGA through a serial peripheral interface (SPI) bus.

2.2. Acquisition and Processing Architecture

The Pyramic array feeds audio data to the field-programmable gate array (FPGA) part of an Altera DE1-SoC device through a SPI bus. Then, the audio samples can be stored in a 1GB DDR3 memory in the DE1-SoC device, which can also be accessed by a dual-core ARM processor integrated in the same die as the FPGA. A stereo signal at 48 kHz and 16 bits can also be sent directly to an audio output in the DE1-SoC through the FPGA.

Overall, the FPGA architecture is composed of (a) a master module controlling and capturing data from the ADCs through the SPI protocol, (b) a direct memory access (DMA) unit that writes data into the DDR3 memory, and, (c) a bus interface to manage the FPGA from the ARM processor. Figure 2 shows the data and control signal flow in the FPGA capture module. Note that the FPGA triggers the capture signal synchronously for all ADCs, which is important for multichannel processing. Additionally, there is a stereo output module, made of a DMA unit that reads back data from the DDR3 memory, and a sound controller driver that controls the audio digital to analog converter of the DE1-SoC.

A C application programming interface (API) is available to the application developer to control the FPGA design from the ARM processor. It is responsible for providing the application with buffers for the captured data and for the output audio, and for enabling the audio capture and output of the DE1-SoC device. The available interfaces for capture through this API are shown in pale red on Figure 2. The DMAs are set up to operate in a *double-buffering* mode, where the input buffer is split equally in two, and a function tells the client software which half is being written to. The buffer size is user-controlled, both for input and output, and is limited to a fixed size of 400 MB for capture and 100 MB for output.

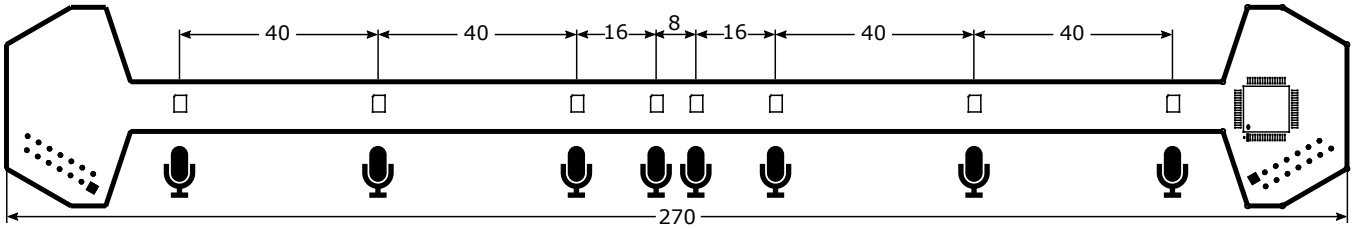


Fig. 3: Single array PCB layout composing the Pyramic array. Distances are in millimeters. Six microphones form a uniform linear array with the remaining two closely spaced at the center to avoid spatial aliasing [29]. The PCBs can be arranged arbitrarily forming new geometries.

female	FAKS0 / SA1	FCMR0 / SA2	FMGD0 / SX34
male	MDLD0 / SX13	MJLN0 / SX99	

Table 2: Speech samples from the TIMIT corpus [31] used for the recordings.

3. DATASET OF ANECHOIC RECORDINGS

The Pyramic array described in the previous section was used to collect a dataset of recordings with sound sources densely located in a circle around the array and at three different heights [1]. The signals played by the sources include calibration signals (sweeps), white noise, and natural speech. The initial motivation for collecting this dataset was to evaluate direction of arrival (DOA) algorithms [30]. Nevertheless, its usefulness naturally extends to the evaluation of audio array processing algorithms in general, e.g. beamforming, calibration, and source separation, to name but a few. The recording setup was carefully designed so that it can be used to evaluate any of linear, piece-wise linear, circular[1], planar, or fully 3D microphone arrays. This is achieved by taking well-chosen subsets of the 48 microphones of the Pyramic array. Along with the raw recordings, the dataset includes the segmented audio samples, the impulse response of the array to each of the source locations, and an accurate calibration of microphones and source locations refined using the dataset recordings themselves.

3.1. Collection

All the recordings took place in the anechoic chamber of EPFL, Switzerland. We placed the tetrahedral Pyramic array on a turn table with one face of the tetrahedron flat on top of the array. Three loud speakers were placed at three different heights between $3.5\,\mathrm{m}$ and $4\,\mathrm{m}$ away from the array. Each speaker played eight sounds successively: a linear and an exponential sweep, white noise, and five speech utterances extracted from the TIMIT database (see Table 2). In addition, there is a segment of silence at the beginning of the recording that can be used to estimate the microphones self-noise. The array was fully rotated in two degrees increments and the sounds were replayed for each speaker and rotation. This forms a total of 4380 recorded sound samples. The control of the turntable as well as the playback were fully computer controlled. The playback device and the Pyramic array were both playing and recording, respectively, at $48\,\mathrm{kHz}$ but were not synchronized. As a consequence, the global time of arrival of the sound at the array was lost. To avoid further loss of timing between the audio samples, they were concatenated and recorded all at once.

[1]But not *uniformly* circular.

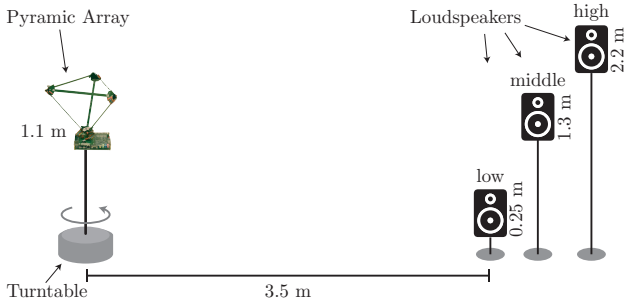


Fig. 4: The setup of Pyramic microphone array with the three loudspeakers in the anechoic chamber.

3.2. Calibration

All distances and locations were calibrated as finely as possible when performing the experiments. There remains however an imprecision unavoidable in manually setup experiments. For example, it turned out to be difficult to position the array so that its top is exactly flat. This leads to small changes in the elevation of the speakers with respect to the array as the latter rotates.

Thanks to the large number of measurements, it is nevertheless possible to improve upon this using numerical optimization techniques. Due to the compactness of the array and its distance from the loudspeakers, the far field approximation holds for our measurements. In this case, it is possible to use the blind calibration method from Thrun [20]. This method jointly recovers the locations of microphones and the direction of sound sources by exploiting the low-rank structure of the time-difference of arrival (TDOA) matrix. This is achieved by solving the following optimization problem

$$\min_{\mathbf{X},\mathbf{P}} \|\mathbf{X}^\top \mathbf{P} - c\boldsymbol{\Delta}\|_F^2, \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{3\times M}$ and $\mathbf{P} \in \mathbb{R}^{3\times N}$ contains the microphone and sources locations, respectively, in their columns. The entries of the TDOA matrix are $(\boldsymbol{\Delta})_{mk} = \tau_{mk}$, which correspond to the time-of-flight between the reference and m-th microphones along the direction of the k-th source, and c is the speed of sound. Thrun's method cleverly solves (1) by applying an SVD followed by a gradient descent to find the affine transformation that makes the columns of $\mathbf{P}$ closest to being unit norm.

To apply the method, we first used the generalized cross-correlation method with phase transform (GCC-PHAT) [32] on the white noise sequence recordings to recover the TDOA between the first microphone and all the others. In five cases, the method failed and returned a TDOA larger than three times the size of the

	Manual	GD	Thrun's [20]
RMSE	7.71 mm	1.86 mm	1.86 mm

Table 3: The root mean-squared fitting error between the microphones and sources locations and the TDOA measurements in millimeters.

array. Because Thrun's method does not consider outliers and a reliable manual calibration was available, we replaced the outliers by the value expected if the manual calibration were correct.

Because the manual calibration provides a good starting point, a simple gradient descent (GD) method can also be considered. We ran both Thrun's and the GD methods. No ground truth being available, we evaluate with the fitting error from (1). As can be seen in Table 3, both Thrun's and GD methods perform similarly, reducing the fitting error of TDOA four-fold compared to manual measurements. The source locations before and after calibration are compared in Figure 5a. The calibration code is released as part of the dataset.

3.3. Array Response

Using the exponential sweep signals recorded, the impulse response of the array for every microphone and source location was computed. The recordings of silence were used to compute the power spectral density of the microphone noise. The noise PSD was used in turn to do Wiener deconvolution of the excitation signal from the recorded data. A typical impulse response is shown in Figure 5b.

4. APPLICATIONS

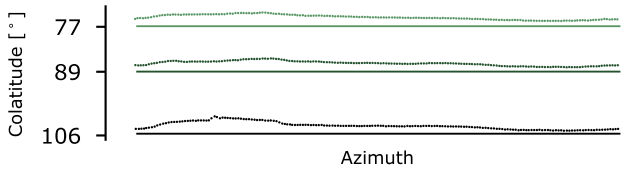
Having a reference hardware platform for array processing, as well as the corresponding dataset, provides opportunities to test a few applications that we outline in this section.

4.1. Live Demonstrations

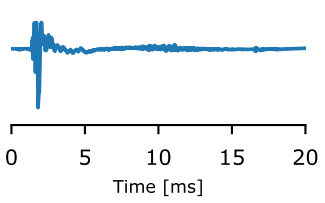
While dissemination of theory of algorithms is of paramount importance, a demonstration of said algorithms running live is a very convincing argument of their practical importance. The Pyramic array was used several times for this purpose. For example, as part of a demonstration at ICASSP 2017 [33], the array was mounted on a specially made lamp with 60 LEDs individually controllable. The data collected from the array was fed to the Bluebild DOA finding algorithm running in real-time on a laptop. The result from the algorithm was then used to create a 3D animated representation of the sound field as well as to illuminate the direction of sources on the LEDs. A similar demonstration was later performed at EPFL, Switzerland for the FRIDA [30] algorithm.

4.2. Evaluation of DOA Algorithms on 3D sources

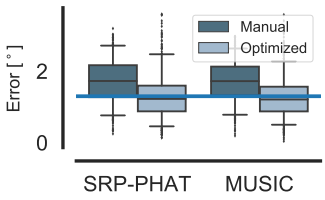
The impact of recording equipment non-idealities is rarely taken into account when evaluating the performance of DOA finding and beamforming algorithms. The Pyramic dataset provides an easy way to evaluate them. As an example, and a further test of the calibration from Section 3.2, we assess the localization performance of MUSIC [19] and SRP-PHAT [29] DOA algorithms with 30000 points on the sphere. Figure 5 shows the localization error of the finding algorithms with respect to source locations before and after optimization is applied. After calibration, both methods present an error close to the average gridding error. We can observe that while the manual calibration error is higher, it presents less outliers than the optimized



(a) Difference of manual (solid line) and optimized (dotted line) calibrations.



(b) A typical impulse response.



(c) DOA Error.

Fig. 5: The spherical localization error for SRP-PHAT and MUSIC DOA finding algorithms. Errors with respect to both manual and optimized calibrations are shown. The horizontal line is the average gridding error.

calibration method. The code used for this evaluation is provided and can serve as a template for benchmarking using the dataset.

4.3. Realistic Simulation of Multichannel Impulse Responses

To cope with the lack of available data, most room impulse response (RIR) generators are limited to a few microphone directional responses. Thanks to the Pyramic dataset, it is possible to combine for example an image source model [34] based RIR generator with the impulse responses from the Pyramic array to simulate indoor sound acquisition using the Pyramic array. The current limited number of heights in the measurements might restrict this application to the 2D room geometry case.

5. CONCLUSION

In this paper we presented a publicly available hardware design of a microphone array, namely the Pyramic array. In addition, we released a dataset composed of 48-channel anechoic recordings of 3D sources using the Pyramic array. Our objective is to reduce the development and testing time of multichannel audio processing algorithms. In the future, we plan to use the Pyramic array measurements to benchmark state-of-the-art calibration techniques [35]. Furthermore, we will extend the Pyramic array reference design and datasets to other audio applications such as source separation, diarization or to asynchronously distributed microphones processing scenarios.

6. ACKNOWLEDGMENTS

We warmly thank André Guignard for helping with the physical construction of the array. Sahand Kashani for reviewing the VHDL design. Hanjie Pan, Dalia El Badawy, Miranda Kreković, and Mihailo Kolundžija for helping during the collection of the dataset. Etienne Rivet and Hervé Lissek for help with the anechoic chamber. Special thanks to Lukas Drude for providing the TikZ code for Figure 2.

7. REFERENCES

[1] R. Scheibler, "Pyramic dataset : 48-channel anechoic audio recordings of 3D sources," Apr. 2018. [Online]. Available: http://dx.doi.org/10.5281/zenodo.1209563

[2] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Springer Science & Business Media, 2008, vol. 1.

[3] M. Brandstein and D. Ward, Eds., *Microphone arrays: signal processing techniques and applications*. Springer, 2013.

[4] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Process. Mag.*, vol. 13, no. 4, pp. 67–94, Jul. 1996.

[5] H. L. Van Trees, *Optimum Array Processing (Part IV of Detection, Estimation, and Modulation Theory)*. John Wiley & Sons, Apr. 2004.

[6] S. Makino, Ed., *Audio Source Separation*. Springer, 2018.

[7] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multi-microphone speech enhancement and source separation," vol. 25, pp. 692–730, 2017.

[8] K. Kinoshita, M. Delcroix, S. Gannot, E. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, A. Sehr, and T. Yoshioka, "A summary of the REVERB challenge: State-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, p. 7, Jan. 2016.

[9] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The CHiME challenges: Robust speech recognition in everyday environments," in *New era for robust speech recognition – Exploiting deep learning*. Springer, Nov. 2017, pp. 327–344.

[10] J. K. McElveen, "Wearable directional microphone array apparatus and system," Jul. 26 2016, US Patent 9.402,117 B2.

[11] E. Weinstein, K. Steele, A. Agarwal, and J. Glass, "LOUD: a 1020-Node Microphone Array and Acoustic Beamformer," in *Proc. ICSV*, Cairns, Australia, Jul. 2007.

[12] F. Perrodin, J. Nikolic, J. Busset, and R. Y. Siegwart, "Design and calibration of large microphone arrays for robotic applications," in *Proc. IEEE/RSJ IROS*, 2012, pp. 4596 – 4601.

[13] A. McPherson, "Bela: An embedded platform for low-latency feedback control of sound," *J. Acoust. Soc. Am.*, vol. 141, no. 5, pp. 3618–3618, 2017.

[14] J. Le Roux and E. Vincent, "A categorization of robust speech processing datasets," Mitsubishi Electric Research Labs, Cambridge, MA, USA, Tech. Rep. TR2014-116, Aug. 2014.

[15] J. Azcarreta, "Pyramic array: An FPGA based platform for multi-channel audio acquisition," École Polytechnique Fédérale de Lausanne (EPFL), Aug. 2016.

[16] C. Ferry, "Extension board for CycloneV multi microphone acquisition – signal analysis," École Polytechnique Fédérale de Lausanne (EPFL), 2017.

[17] F. Mondada, M. Bonani, F. Riedo, M. Briod, L. Pereyre, P. Rétornaz, and S. Magnenat, "Bringing robotics to formal education: The Thymio open-source hardware robot," *IEEE Robot. Automat. Mag.*, 2017.

[18] J. H. DiBiase, H. F. Silverman, and M. S. Brandstein, "Robust Localization in Reverberant Rooms," in *Microphone Arrays*. Berlin, Heidelberg: Springer, 2001, pp. 157–180.

[19] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, 1986.

[20] S. Thrun, "Affine Structure From Sound," in *NIPS*, 2006, pp. 1353–1360.

[21] R. Scheibler, "Rake, Peel, Sketch: The signal processing pipeline revisited," Ph.D. dissertation, EPFL, Switzerland, 2017.

[22] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A Python package for audio room simulations and array processing algorithms," in *Proc. IEEE ICASSP*, Calgary, CAN, 2018, pp. 351–355.

[23] E. Zwyssig, F. Faubel, S. Renals, and M. Lincoln, "Recognition of overlapping speech using digital MEMS microphone arrays," in *Proc. IEEE ICASSP*, Vancouver, CAN, 2013, pp. 7068–7072.

[24] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronentha, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner, "The AMI meeting corpus," in *Measuring Behavior '05*, vol. 88, 2005, pp. 100–103.

[25] J. Thiemann, N. Ito, and E. Vincent, "The Diverse Environments Multi-channel Acoustic Noise Database (DEMAND): A database of multichannel environmental noise recordings," *J. Acoust. Soc. Am.*, vol. 133, p. 3591, 2013.

[26] S. Duffner, P. Motlicek, and D. Korchagin, "The TA2 database – a multi-modal database from home entertainment," *Int. Journal of Comp. and Elec. Eng.*, vol. 4, no. 5, pp. 670–673, 2012.

[27] T. M. Sullivan and R. Stern, "Multi-microphone cross-correlation based processing for robust speech recognition," *J. Acoust. Soc. Am.*, vol. 93, p. 2319, 1993.

[28] E. Vincent, S. Watanabe, A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, vol. 46, pp. 535–557, 2017.

[29] I. J. Tashev, *Sound Capture and Processing*, ser. Practical Approaches. Chichester, UK: John Wiley & Sons, Jul. 2009.

[30] H. Pan, R. Scheibler, E. Bezzam, I. Dokmanić, and M. Vetterli, "FRIDA: FRI-Based DOA Estimation for Arbitrary Array Layouts," in *Proc. IEEE ICASSP*, 2017.

[31] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "TIMIT acoustic-phonetic continuous speech corpus," *Linguistic data consortium*, vol. 10, no. 5, p. 0, 1993.

[32] C. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 4, pp. 320–327, 1976.

[33] E. Bezzam, R. Scheibler, J. Azcarreta, H. Pan, M. Simeoni, R. Beuchat, P. Hurley, B. Bruneau, C. Ferry, and S. Kashani, "Hardware and software for reproducible research in audio array signal processing," *Proc. IEEE ICASSP*, pp. 6591–6592, March 2017.

[34] J. B. Allen and D. A. Berkley, "Image Method For Efficiently Simulating Small-room Acoustics," *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, 1979.

[35] M. Kreković, G. Baechler, I. Dokmanić, and M. Vetterli, "Structure from sound with incomplete data," in *Proc. IEEE ICASSP*, Calgary, CAN, 2018.