Pyramic: Full Stack Open Microphone Array Architecture and Dataset

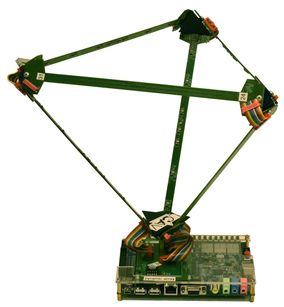
Robin Scheibler*, Juan Azcarreta, Corentin Ferry*, and René Beuchat*†
*Tokyo Metropolitan University *Univ Rennes, F-35000 Rennes, France †École Polytechnique Fédérale de Lausanne, Switzerland

Full-stack microphone array design

Abstract – We present **Pyramic**[1], a *full-stack open* microphone array design. Full-stack open means that every step of the array design from the hardware up to data collection and algorithms is open and documented. The result is a collection of templates under **permissive licenses** covering the whole design process and that can be reused as needed.

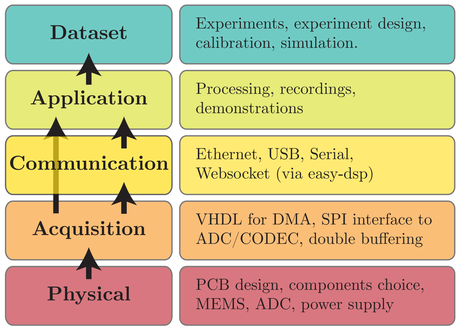
Targets – Compact arrays, embedded arrays, robotics, IoT, wearables

The Pyramic array



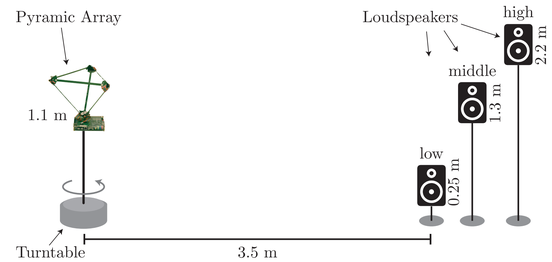
- **48 microphones** distributed on 6 edges
- 2 output channels
- Sampling frequency @ 48 kHz
- Bit depth of 16 bits per channel
- FPGA reads samples into shared memory
- Dual-core ARM CPU @ 800 MHz

Full-stack design



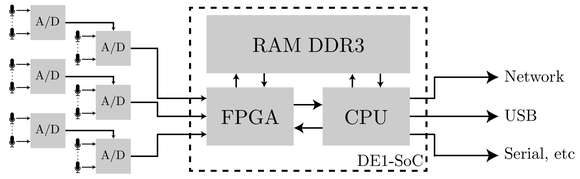
Dataset layer: Anechoic recordings[2]

- Recordings done in anechoic chamber
- Array fully rotated in increments of **2 degrees**
- Three loudspeaker heights
- Total of 540 positions
- 8 samples per position (2x sweeps, 1x noise, 5x speech)



Communication/Application layer

- System-On-Chip with FPGA/ARM, 1GB RAM
- Linux system
- Ethernet connection to the network
- USB/Serial communication
- Easy-DSP[3]: Browser based processing interface



Physical layer: PCB design[1]

- Each sub-array is 27 cm long
- **Microphones**: Analog INMP5404 MEMS
- **Analog to Digital converter (ADC)**: AD7606

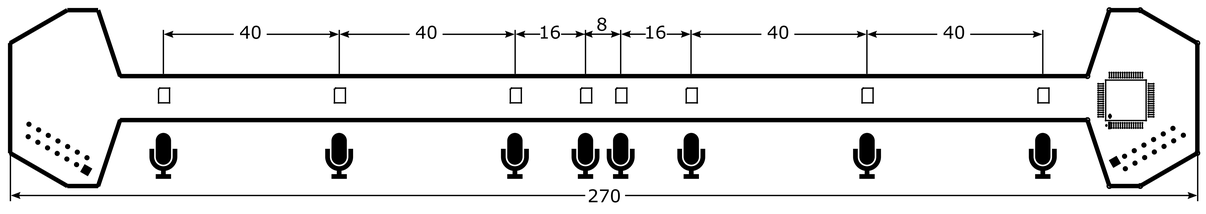


Figure 1: In a single PCB, six microphones form a uniform linear array, with the remaining two closely spaced at the center to avoid spatial aliasing. Distances are in millimeters.

Calibration and Code Samples

- Manual calibration improved with self-localization [Thrun 2006]
- Dataset wrapper and code samples provided in Python[2]

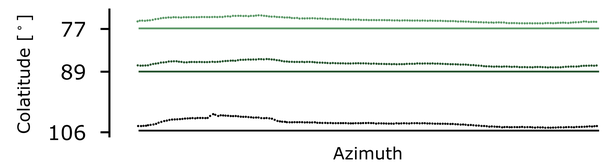
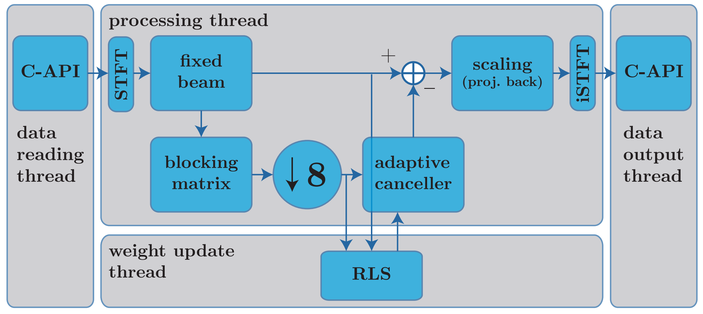


Figure 2: Difference of manual (solid line) and optimized (dotted line) calibrations.

Tonight: Demo! Real-Time Beamforming[4]

Demo session at 19:30

- Fixed 48-channel beamformer computed during calibration step
- Number of channels reduced to 6 in the adaptive branch
- Recursive least squares (RLS) computes the adaptive weights
- Multi-threaded implementation



Publications using the Pyramic array

H. Pan, R. Scheibler, E. Bezzam et al., *FRIDA: FRI-based DOA Estimation with Arbitrary Array Layout*, ICASSP, 2017

E. Bezzam, R. Scheibler, J. Azcarreta, H. Pan et al., *Hardware And Software For Reproducible Research In Audio Array Signal Processing* , ICASSP, 2017

R. Scheibler, D. Horiike, N. Ono, *Blinkies: Sound-to-light conversion sensors and their application to speech enhancement and sound source localization*, APSIPA, 2018

Get it all!

Free, as in *free speech*, and as in *free beer*!!
[1]Pyramic hardware and software: `https://github.com/LCAV/Pyramic`
[2]Dataset: `https://zenodo.org/record/1209563#.W5qOkkxuKUk`
[3]Easy-DSP: `https://github.com/LCAV/easy-dsp`
[4]Demo software: `https://github.com/fakufaku/pyramic-demo`

We would like to acknowledge Hanjie Pan, Dalia El Badawy, Miranda Kreković, and Mihailo Kolundžija for helping during the collection of the dataset. Etienne Rivet and Hervé Lissek for their assistance with the anechoic chamber. Special thanks to André Guignard for his kind collaboration with the physical construction of the Pyramic array. We also thank Sahand Kashani for his contributions to the VHDL design.