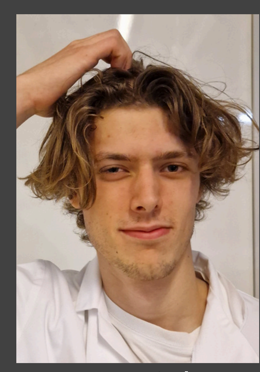
24 **Triangular Rods:**

Capturing Acoustic Patterns with an Acoustic Camera


Atom Levison


Camiel van der Marel


Sam Moerdijk


Jasper Nierse

1. Introduction

It can be derived mathematically that a triangle mostly vibrates at its end points (see QR code). We have tried to demonstrate this result experimentally using an acoustic camera (AC). The AC is a tool made up of a microphone array and a camera that is able to localize a sound and visualize it on a picture or video. This is where the AC has the advantage over the conventional approach where it is also able to visualize it.

2. Theory

An AC can be seen in figure 1. It can be used to locate sound sources under two assumptions:

1. All microphones collect the same data if we ignore the phase-difference.
2. The sound radiated from the source can be interpreted as a plane wave, which is a fair assumption for $r \gg 2\lambda$, where *r* is the distance from the source to the AC and λ is the wavelength of the incoming soundwave [1].

The difference in path length ($v\,\Delta t$) can then be used to calculate Θ using *formula 1*, where $v\,\Delta t$ and Θ are as given in *figure 3*. This method is commonly known as time-based analysis.

$$\theta = \arccos\frac{v\Delta t}{d}$$

Formula 1: *v* is the speed of sound, *Δt* is the time difference and *d* is the distance between individual microphones.

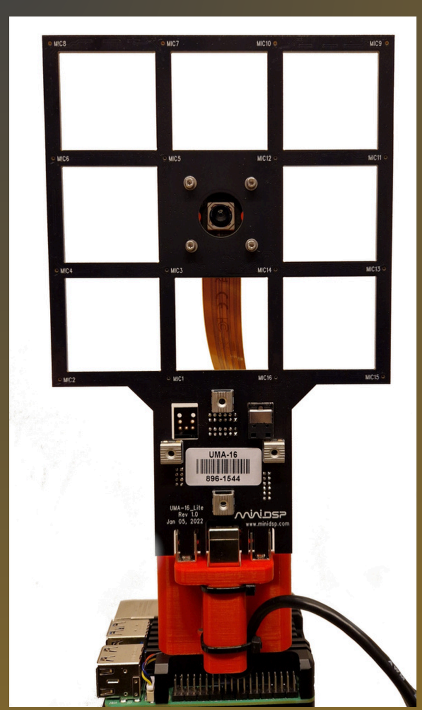
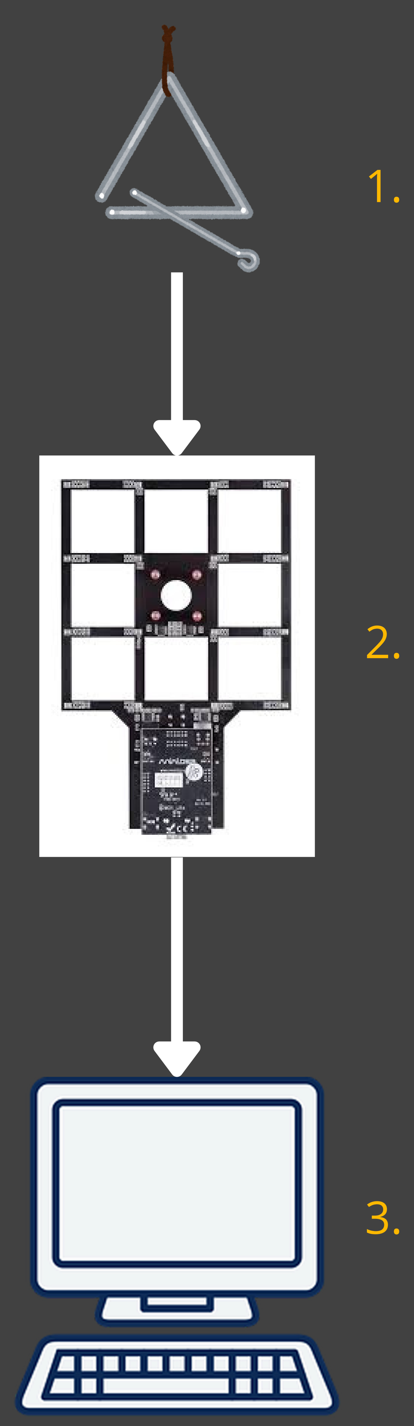


Figure 1: An AC consists of an array of microphones, with a camera in the center. This is the AC we used: a miniDSP UMA-16 v2 microphone array and a Raspberry Pi Camera Module 3 Wide camera operated by a Raspberry Pi 5 computer. The microphones are 0.042 ± 0.001m apart.

3. Method

Figure 2: Test setup



1. The lowest frequency with a major spike is located at 1911Hz, as can be seen in *figure 4*. This corresponds to a wavelength of 0.18m. Since all higher frequencies will have shorter wavelengths, setting the AC and the source 0.90m apart allows us to use $r \gg 2\lambda$.
2. The AC that will be used is shown in *figure 1*. Audio will be detected at a sampling frequency of 44000Hz for a duration of 2 seconds. The picture will be taken at a resolution of 1920x1440.
3. Python collects the data of all 16 microphones and calculates an average phase difference by use of the correlation function in the scipy.signal library [3]. The location of the source is then found by use of *formula 1*. Finally, this location is converted to pixel coordinates and shown in the picture. [2]

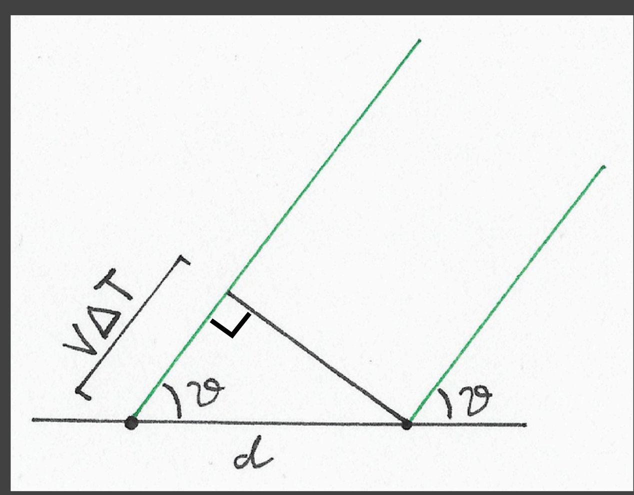


Figure 3: *v Δt* is the difference in path length, *d* is the distance between microphones and *Θ* is the incoming angle.

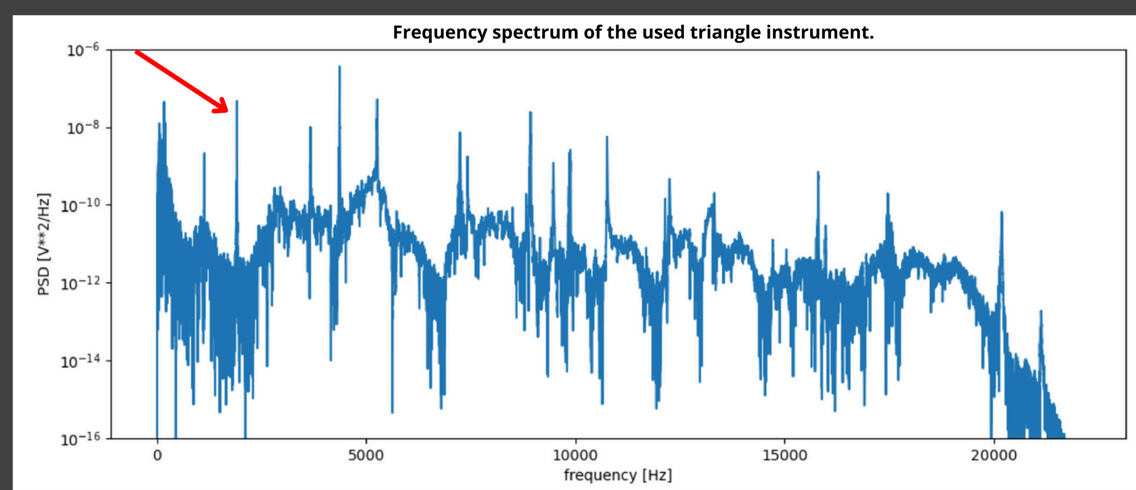


Figure 4: Frequency spectrum of the used triangle instrument. The red arrow shows the spike at 1911Hz.

4. Results and Discussion

The sound source was localized at ((9.8 ± 1.9) × 10^{2} ; (8.4 ± 1.7) × 10^{2}). The error was propagated based on the error on the distance between the microphones of 0.001m, as well as an error on Δt of 1/44000s. The result is shown in *figure 5*, which is a cropped version of the original figure.

There are three important discussion points:

1. The errorbars span over the entire triangle, meaning that the resolution of our setup is too low to conclude anything meaningful.
2. Most of our measurements were influenced by background noise. It would thus be naive to think that this measurement was fully accurate.
3. It is possible that we measured the sound of the wooden hammer hitting the triangle, rather than the vibrations of the triangle.

A few things that can be done to further improve and expand upon this setup and technique are:

1. Use beamforming techniques to filter the undesirable noise. [4]
2. Use beamforming techniques to improve accuracy. [4]
3. Use a bigger non symmetrical array for increased accuracy. [4]
4. Merge the time based audio with a video recording. [5]

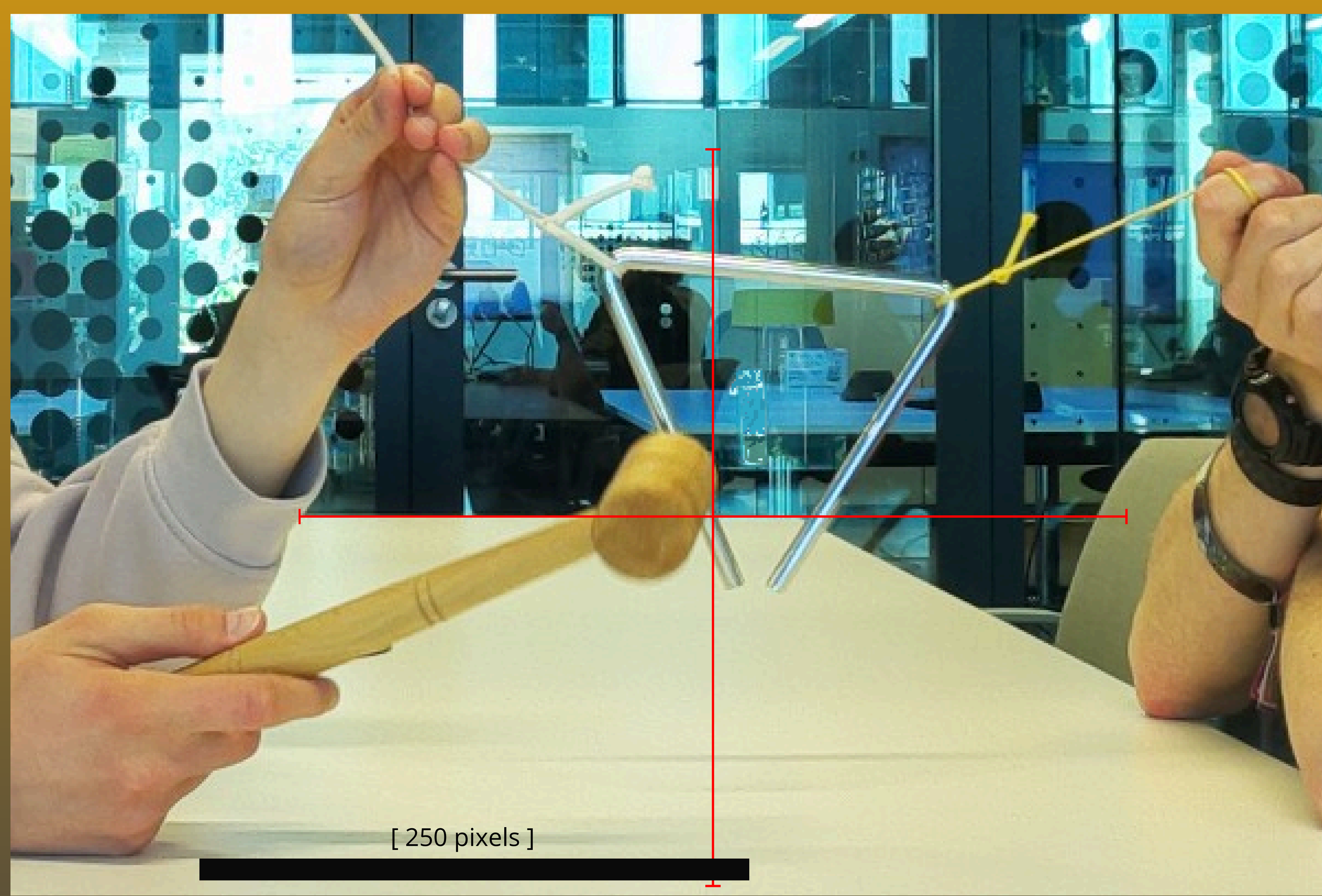


Figure 5: A cropped version of the result of our final measurement. The original 1920x1440 picture was cropped for better visibility. The distance between the triangle and the array was 0.9m. The red cross indicates the found coordinates with its error-margin.

5. References

[1]. de Lucia, D. (2021). *Implementation of a low-cost acoustic camera using arrays of MEMS microphones* [Master's Thesis, POLITECNICO DI MILANO], https://www.politesi.polimi.it/retrieve/84d6b8b9-be88-4655-8f34-f15b6699fe08/2021_12_De%20Lucia.pdf

[2]. van der Marel, C. (2024). *Group-24-DIYacousticcamera* [Software]. GitHub. https://github.com/Camiel023/acousticcamera-group24

[3]. Virtanen, et al. (2020). *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python* [Software]. Nature Methods, 17(3), 261-272. https://pypi.org/project/scipy/

[4]. Chiariotti, et al. (2019). *Acoustic beamforming for noise source localization – Reviews, methodology and applications* [article]. Mechanical Systems and Signal Processing, 120, 422-448. https://www.sciencedirect.com/science/article/abs/pii/S088832701830637X?via%3Dihub

[5]. Navat, M. (2021). *DIY Acoustic Camera using UMA-16* [article]. Mike's Newsletter. https://navat.substack.com/p/diy-acoustic-camera-using-uma-16

University of Amsterdam
VU Vrije Universiteit Amsterdam