Audio Engineering Society

Convention e-Brief 401

Presented at the 143rd Convention
2017 October 18–21, New York, NY, USA

Sound Fields Forever: Mapping sound fields via position-aware smartphones

Scott H. Hawley[1], Sebastian Alegre[1], and Brynn Yonker[1]

[1] *Department of Chemistry & Physics, Belmont University, Nashville TN USA*

Correspondence should be addressed to Author (scott.hawley@belmont.edu)

ABSTRACT

Google Project Tango is a suite of built-in sensors and libraries intended for Augmented Reality applications allowing certain mobile devices to track their motion and orientation in three dimensions without the need for any additional hardware. Our new Android app, "Sound Fields Forever," combines locations with sound intensity data in multiple frequency bands taken from a co-moving external microphone plugged into the phone's analog jack. These data are sent wirelessly to a visualization server running in a web browser. This system is intended for roles in education, live sound reinforcement, and architectural acoustics. The relatively low cost of our approach compared to more sophisticated 3D acoustical mapping systems could make it an accessible option for such applications.

1 Introduction

Visual representations of sound patterns serve as useful education and diagnostic tools. Prior work on visualization of two-dimensional (2D) sound directivity measurements of loudspeakers and microphones using commodity smartphone architectures [1] inspired us to pursue the development of a tool for measuring and visualizing 3D sound fields.

The two main approaches to this involve the use of multichannel microphone arrays, or a single microphone which 'sweeps' through the space. The latter approach is less complicated experimentally, but is primarily only useful for steady-state sound sources. Regardless of which approach is used, an additional complication involves the careful (and at times laborious) 'bookkeeping' process of logging microphone positions.

Our method uses a special smartphone endowed with Google Tango [2], a technology intended for Augmented Reality but which we use for automated data acquisition, to perform real-time logging of location information for the sound measurements.

2 System Design

The system is built on a client-server model, whereby the smartphone app measures location and sound intensity information, and streams these data wirelessly via a WebSockets connection to a visualization server running in a web browser.

2.1 Tango Device.

The driving paradigm for the design is that of a live sound reinforcement application, during which one or more assistants can be "walking the venue" with the Tango devices, providing the front-of-house engineer a real-time map of the venue's sound.

Using a combination of cameras, infrared emitters of point clouds (similar to those produced by Microscoft Kinect), and inertial sensors (gyroscope and accelerometer), the Lenovo Phab 2 Pro shown in Figure 1 is able to synthesize an internal representation of the geometry of its surrounding environment (e.g., walls and objects) as well as the phone's position within it.

Figure 1. The Lenovo Phab 2 Pro, the first Google Tango device for the general public. The cameras and inertial sensors are used by the Tango software to provide a representation of the room and the phone's position within it. (Source: Lenovo.com)

The internal coordinate system established by the software is typically accurate to within 2 cm, however it can experience some 'drift' which is routinely corrected for. The primary system of drift correction is known as "Area Learning," in which an estimate of the environment's geometry is stored and compared against at later times. While such temporary trajectory errors have minimal effects on most AR applications such as gaming, for our scientific data-logging app the temporary lapses in correct position can produce data which are assigned the incorrect location. We account for this simply by averaging the nearby data in a nearest-neighbours smoothing operation within the visualization server.

Since the Tango device is able to operate untethered, i.e. without any "lighthouse" or motion-tracking camera systems (as in Virtual Reality systems such as Oculus and HTC Vive), and can operate in environments of essentially arbitrary size, we have named our combined client-server system "Sound Fields Forever," which is also the name of the Android client app [4].

The Area Learning feature also allows for saving the environment's environment data, both for later use as well as for sharing among multiple devices – as a means of synchronizing their internal coordinate systems. At time of press this feature has not been included in our app, so that only one device is usable at a time, however we plan to upgrade this soon.

2.2 Client App. The data-acquisition client is operated by sweeping the smartphone in a path throughout the sound field in question, e.g. by walking around the venue. At the time of development, only one specific smartphone equipped with Google Tango was available, namely the Lenovo Phab 2 Pro. The internal microphone of this device feeds into a compression and noise reduction circuit which is *not* bypassable, making it unsuitable for sound measurements. Instead, we use a small lavalier condenser microphone plugged into the Phab 2 Pro's analog audio input. After calibrating the app to account for the frequency response of this microphone, we are able to obtain values corresponding to C-weighted dB SPL.

The app applies a series of bandpass filters in octave bands ranging from 62.5 Hz to 4000 Hz, plus one additional filter at a user-specified frequency, and reports the time-averaged sound levels in these filter bands to the visualization server.

2.3 Visualization Server.

The visualization server is written in JavaScript, and relies on the Socket.io implementation of WebSockets for the Node.js server-side scripting package. The graphics are rendered in the web browser using the d3.js real-time data visualization library [6]. The data locations or "sites" are used to fill in a full map by means of a Voronoi diagram — i.e., it assigns each pixel color to be the same value as that of the nearest data site. In order to minimize the effects of location drift errors and/or transient "noise," the server can apply "smoothing" of the sound level values via a simple nearest-neighbour smoothing algorithm adapted for unstructured datasets. Future work may include a more sophisticated smoothing system using, e.g., radial basis functions. The web server can similarly produce contour plots, for which the (smoothed) data are first interpolated onto a rectangular grid via barycentric interpolation.

Since often only "head height" significant for music venues, for simplicity and speed the web visualization software produces only a 2D map of the sound field by disregarding the height information. Full 3D renderings can be made using "offline" with 3D data visualization applications such as ParaView [7], and may be possible in the web server in the future, pending upgrades to the visualization code.

3 Results

In Figure 2 we show sample output from the visualization server, for measurements at Columbia Studio A in Nashville TN, a 10m x 12m studio tracking room with reflective walls, which is often used for rehearsals by the live sound production students at Belmont University. The room is excited by pink noise played from a PA set up on one end of the room, and visualized using the bandpass filter in the 62.5 Hz octave band. The small black-in-white circles show the path swept by the smartphone client, as measured by the Tango system.

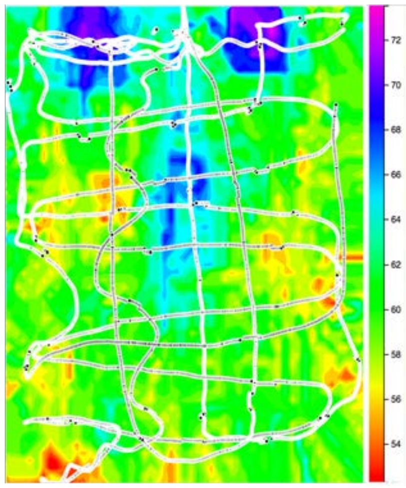

Figure 2. 2D contour plot of sound intensity level in Columbia Studio A, with pink noise played from a PA at the top of the figure.

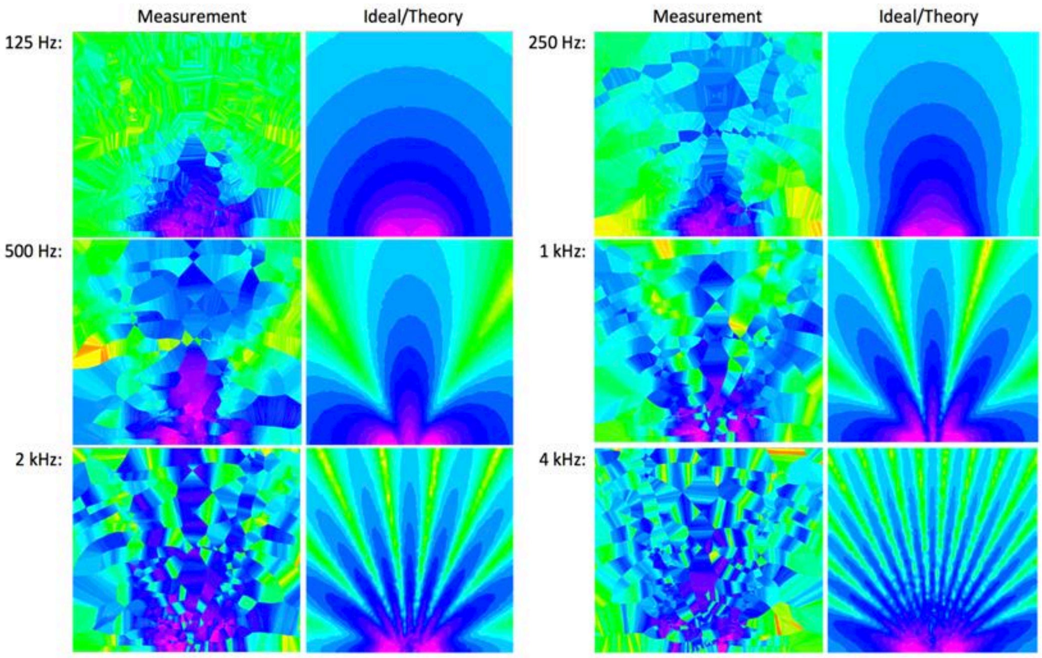

Figure 3. Sound intensity map for two-speaker interference, in various octaves. Left columns show measurements, right columns show simulated data (for idealized omnidirectional sources), where colors are in analogy to light energy: purple = loudest, red = quietest. Two loudspeakers placed 2 ft. apart in an anechoic environment played a single harmonic superposition of equal-amplitude steady-state tones of the frequencies shown. Bandpass filters in the app produced sound intensities for each octave band, which were streamed to the visualization server and plotted individually. Although some positional drift is evident, we see the expected angular dependencies of nodes.

Figure 3 shows a measurement of two-speaker interference for several frequency bands, obtained by sweeping the smartphone inside an anechoic chamber in a manner similar to that shown in Figure 2. The angular dependence shows good agreement with what one would expect from the simple interference due to two omnidirectional sources. The intensity fall-off with distance shows some variance from the ‘theory’ case, which we will continue to investigate.

These 2D results were obtained from the real-time visualization server by disregarding height information. By means of more versatile software such as ParaView [7], one can use the full 3D data provided by the client to obtain offline visualizations of full 3D datasets, such as the sound field in the neighbourhood (~1m) of a loudspeaker as shown in Figure 4. In this figure, the lack of symmetry and variability in the isosurfaces shown may be due to drift errors from the Tango, indicating that this method may be better suited for large, venue-scale measurements as opposed to close-range studies requiring high positional fidelity.

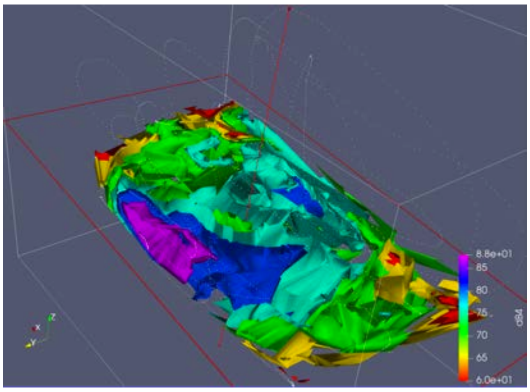

Figure 4. Set of 3D isosurfaces for sound in front of a Mackie HR824mk2 speaker playing a 1kHz tone, plotted using ParaView [7]. Here the 3D surfaces are sliced horizontally to reveal their structure. The variability and lack of symmetry may be due to drift errors or other location inaccuracies from the Tango system.

4 Conclusions

We have demonstrated a portable system for mapping sound fields in 2D and 3D, appropriate for acoustics education, room tuning, and live sound reinforcement. This mapping method relies entirely on internal smartphone hardware, and does not require additional scanning or beaming hardware to be placed around the environment.

The system provides a convenient and relatively inexpensive method for automating data acquisition of position-dependent measurements of steady-state sound fields, and for producing visualizations of sound fields in rooms. Due to drift errors in the AR system, however, location information can contain errors which can yield poor results for close-range measurements requiring high location accuracy. Methods for refining this are being investigated.

Since the initial preparation of this eBrief, AR frameworks have been released with do not require the Tango hardware (i.e. without so-called “depth-sensing cameras”), namely Google’s “ARCore” [8] and Apple’s “ARKit” [9]. While we look forward to investigating these technologies, it remains unclear what level of positional accuracy they will provide for scientific applications.

5 Acknowledgements

This project was supported by the Summer Undergraduate Research For the Sciences (SURFS) program at Belmont University. The authors also wish to thank Dr. Wesley Bulla and Dave Warburton for access to the anechoic chamber.

References

[1] S.H. Hawley and R.E. McClain Jr., “Visualizing Sound Directivity via Smartphone Sensors,” accepted for publication in *The Physics Teacher* (2017). https://arxiv.org/abs/1702.06072

[2] D. Keralia, K.K. Vyas and K. Deulkar, “Google Project Tango – A Convenient 3D Modeling Device,” *International Journal of Current Engineering & Technology*, **4** no. 5 (2014). https://get.google.com/tango

[3] "Area Learning," *Google Tango Developers Guide*, https://developers.google.com/tango/overview/area-learning, last accessed Aug 7 (2017).

[4] S.H. Hawley and S. Alegre, "Sound Fields Forever," *Google Play Store*, (2017).

[5] V. Pimentel and B.G. Nickerson, "Communicating and Displaying Real-Time Data with WebSocket," *IEEE Internet Computing*, **16**, 4, (2012).

[6] M. Bostock, V. Ogievetsky, J. Heer, "D3: Data-Driven Documents," *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2011). https://d3js.org/

[7] J. Ahrens, B. Geveci, C. Law, "ParaView: An End-User Tool for Large Data Visualization," *Visualization Handbook*, Elsevier (2005). https://www.paraview.org/

[8] Google ARCore software, https://developers.google.com/ar/

[9] Apple ARKit software, https://developer.apple.com/arkit/