Audio Engineering Society

Convention Paper

Presented at the 153rd Convention
2022 October, New York, NY

Determining the Source Location of Gunshots From Digital Recordings

Robert B. Calhoun[1] and Scott Lamkin[1]

[1] *ShotSpotter, Inc.*

Correspondence should be addressed to Robert B. Calhoun (rcalhoun@shotspotter.com)

ABSTRACT

Digital recordings, in the form of photographs, audio, or video play an important evidentiary role in the criminal justice system. One type of audio recording that experts in acoustic forensics may encounter are those produced by the ShotSpotter Respond gunshot location system. This technical note introduces open-source software for extracting the location and timing metadata stored in these recordings and for performing acoustic multilateration on a set of time differences of arrival determined from ShotSpotter WAV files or any other digital audio file for which sufficiently accurate timing and location metadata are available. The process of determining the source location of a gunshot from three or more digital recordings is further explained via a set of worked examples, including a simple single-shot incident and an incident in which seven shots were fired from a moving vehicle.

1 Introduction

Digital recordings, in the form of photographs, audio recordings, and video recordings play an important evidentiary role in the criminal justice system [1]. Digital recordings of gunshot muzzle blasts provide valuable evidence regarding the timing and number of shots fired; multiple timestamped recordings can be used to estimate the source of a gunshot muzzle blast [2].

This technical note focuses on digital audio files generated by the ShotSpotter Respond gunshot location system. ShotSpotter deploys networks of acoustic sensors to provide real-time notification of gunfire incidents [3]. The systems are typically installed on behalf of public safety agencies in neighborhoods with a documented history of gun violence. The system measures the arrival time of impulsive audio sounds on multiple sensors deployed at known locations on buildings or utility poles and computes the discharge time and discharge location using the mathematical technique of multilateration. Following the multilateration step, audio is retrieved from participating sensors to classify the located sound event as one of concern—such as gunfire—or as a similar-sounding impulsive noise event generated by fireworks or vehicular backfires. Incidents flagged as plausible gunfire by the machine classifier are routed to trained operators for validation before publication to the subscribing public safety agency, who respond in accordance with the policies and procedures of their department.

ShotSpotter's sensors digitize audio and perform all digital signal processing necessary to detect impulsive sounds characteristic of gunfire and measure arrival time and other acoustic properties. The data is transmitted to the location engine as a sequence of individual pulse packets. Audio is not retrieved until a possible shot is localized using data from multiple sensors. Remote signal processing permits analysis of multiple high-sampling rate audio channels while minimizing wireless bandwidth usage, but it does not facilitate independent forensic analysis. In contrast, the minimally-processed digital audio recordings produced by ShotSpotter Respond can support independent review and may qualify as substantive digital evidence [4]. The present work describes the audio format used in ShotSpotter WAV files and introduces open-source software tools to extract recording location and timing metadata from these files.

2 ShotSpotter Digital Recordings

2.1 Digital Audio System

ShotSpotter model "Scepter 2" sensors are Linux-based single-board computers equipped with a cellular radio, an on-board GPS receiver[1], and a digital audio system comprising analog microphones and a 24-bit analog-to-digital converter[2]. The system clock is disciplined by the pulse-per-second signal from an on-board GPS chipset using `chrony` [5], a program similar to the well-known `ntpd` [6] optimized for embedded system.

Digital audio is acquired using Linux ALSA kernel timestamping functions [7] with corrections applied for the (fixed) latency associated with digital filtering on the codec and transfer of data via I2S. In standard recording mode, 48 kHz audio is downsampled to single-channel, 12 kHz, 16-bit FLAC [8] cache files of one second duration, with each file starting at the top of the second and comprising a number of samples equal to the sampling rate. While the Linux system clock is GPS-disciplined, the 24.576 MHz oscillator used for audio acquisition is not; occasionally a sample must be dropped or repeated in order to keep the audio properly synchronized with the system clock. The one-second cache files are stored on the remote sensor in a circular buffer that is only retrieved as necessary to fulfill server-side requests for audio files of a specific length and duration, and deleted approximately 24 hours after creation.

[1]MAX-7C GPS module from u-blox (Thalwil, Switzerland) coupled with a GLA.01 ceramic loop antenna from Taoglas (Enniscorthy, Ireland).

[2]KECG2742WBL-25L electret microphones from Transducers USA (Elk Grove Village, IL) digitized by a Cirrus Logic (Austin, TX) WM8737 stereo audio codec.

2.2 Digital Audio Formats

The digital audio recordings available from a ShotSpotter Respond incident comprises one or more PCM-coded WAV (audio/x-wav) files, plus embedded metadata that documents the time and location when each file was recorded. These files are constructed by transcoding one-second duration FLAC files that are downloaded from the remote sensors as required. Audio files associated with real-time alerts ("reviewer audio") are automatically generated such that each file starts one second before the predicted arrival time of the first impulse. When an incident is published as a result of the review process, a set of "clock-aligned audio" files are created. While each reviewer audio file has a different starting timestamp in order to focus attention on the impulsive noise, sets of clock-aligned audio files share the same start time and are aligned to one-second boundaries. Clock-aligned audio files are preferred for forensic work because they clearly indicate the relative times of arrival of the gunshot sounds on different sensors and because—unlike reviewer audio—they do not imply a discharge location. Both sets of files have the same timing accuracy and reviewer audio can be used for forensic work if clock-aligned files are not available.

2.3 File Integrity

Audio file download requests are tracked in a database. In line with best practices for handling digital audio [10], a digital hash (MD5) of each WAV file is computed and recorded along with the timestamps for when the data was requested and when the request was filled. Completed WAV files are stored in AWS S3 with bucket versioning [11] enabled to protect files from accidental deletion or modification. The AWS S3 "ETag" field (also an MD5 hash [12]) can be used to verify that the file was stored correctly in AWS S3. While MD5 is no longer considered a cryptographically secure hash, the combination of request logs, bucket versioning and file hashes can be used to verify that the original PCM WAV file has not been changed.

```
{
  "serialNumber": "SCP-00-BNG-8319",
  "startTimeUTC": "2021-09-29T15:37:47.000Z
    ",
  "duration": 8.0,
  "utcOffset": -14400.0,
  "spoolFormat": 10,
  "recordStatus": -1,
  "friendlyName": "SCP-00-BNG-8319",
  "friendlyNumber": -1,
  "sensorType": "Scepter2",
  "firmwareVersion": "ECO1774rc24
    -9.1.0.77147-0p5",
  "geolocation": {
    "latitude": 41.290138,
    "longitude": -82.226468,
    "elevation": 257.88016086756
  },
  "zone": {
    "name": "OberlinOH",
    "zoneId": 404,
    "host": "SSI-EASTERN-DEV",
    "instanceId": 40,
    "instanceVersion": "9.8.12.77116",
    "displayName": "OberlinOH",
    "localTzid": "America/New_York"
  },
  "weather": {
    "medium": "air",
    "temperature": 18.3,
    "relativeHumidity": 58.68,
    "salinity": 0.0,
    "speed": 0.0,
    "direction": 0.0,
    "precipitation": "clear",
    "observationTime": "2021-09-29T14
    :51:00.000Z",
    "observationDetails": {
      "provider": "METAR station",
      "station": "KCLE"
    }
  },
  "transducers": [
    {
      "name": "Microphone (SCP Standard)",
      "channel": 0,
      "fullScale": 93.09,
      "noise": 0.0,
      "active": true
    }
  ]
}
```


Fig. 1: Sample smjx metadata object embedded in ShotSpotter audio recordings. Schema in [9].

2.4 Recording Metadata

Acoustic multilateration requires knowing the location and the sound arrival time at each acoustic receiver. Embedding the time and location of the recording as file metadata helps ensure they are transferred together. ShotSpotter WAV files contain a custom `smjx` RIFF chunk, where "x" is the version character, currently "0". This chunk is an unencrypted json object that has been compressed with the zlib compression library. The format is documented in `json-schema` [13] format and the open-source tool `smjx_reader.py` is available extract and decompress it [9].

For acoustic multilateration, the primary data of interest are `startTimeUTC` (the timestamp of the first sample in the recording) and `geolocation` (the WGS84 [14] latitude, longitude, and elevation above MSL of the sensor at the start of the recording). The `smjx` object also contains a weather object that documents the air temperature and wind conditions in use by the ShotSpotter system at the time of the recording, but, as historical data on local weather conditions can be obtained from multiple providers, it is not necessary to use the weather data embedded in the file.

A sample of `smjx` file metadata is given in Figure 1.

2.5 Time Synchronization Accuracy

Source location algorithms take *relative* times-of-arrival as input but measuring absolute arrival times (relative to UTC) avoids the challenges associated with using unsynchronized receivers [15]. Scepter 2 is configured to use five reference clocks: a kernel pulse-per second (KPPS) refclock that ingests timepulse signals from the GPS chipset via a GPIO; GPS navigation messages transmitted via a serial port; and up to three network time protocol (NTP) servers. The NTP refclocks allow quick synchronization on device boot but their accuracy is limited by the relatively high latency of the cellular network. The refclocks are managed by `chronyd`, which evaluates the quality of each refclock and disciplines the system clock. As noted by Mills [16], the magnitude of the maximum timing error is of greater interest than the mean offset. Per RFC 5905 [17], this maximum timing error λ due to all causes is:

$$\lambda = |\theta| + \varepsilon + \delta/2$$

where θ is the system time offset, ε is the root dispersion, and δ the root delay. These timing statistics

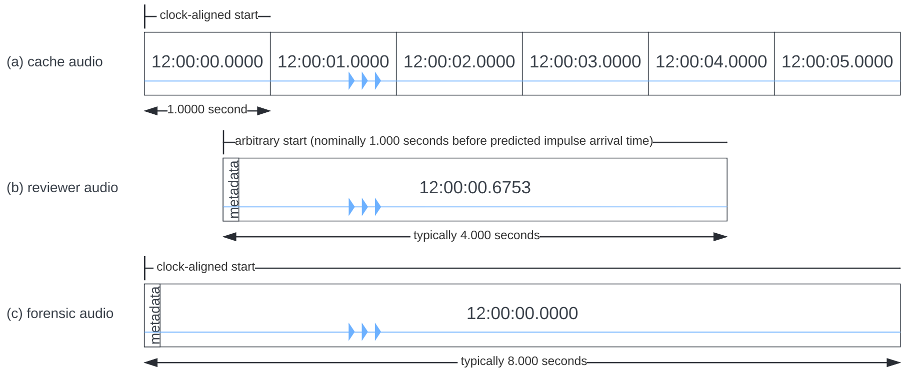

Fig. 2: Types of audio produced by the ShotSpotter system. Cache files (a) of one-second duration are recorded by the remote sensor independently of any triggering activity. Reviewer audio files (b) are constructed such that the file starts 1.0 seconds before the computed arrival time of the gunshot; forensic audio files (c) are constructed on one-second boundaries and of duration sufficient to contain all shots on all sensors.

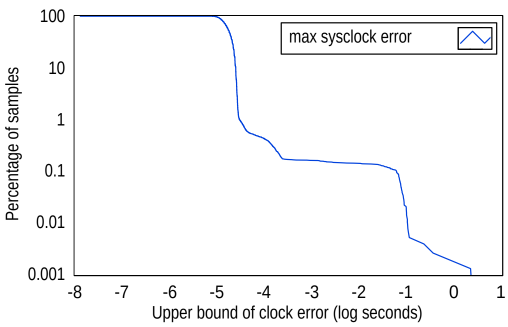

Fig. 3: Cumulative distribution function of maximum sysclock error λ from a survey of Scepter 2 sensors. Plotted on a log-log plot following Mills [6]. Two populations are visible, devices using KPPS refclocks (λ between 10^{-5} and 10^{-4} sec) and devices using NTP refclocks (λ between 10^{-3} and 10^{-1} sec); NTP refclocks are inadequate for acoustic multilateration. A small number of observations (34 out of 79119) made on devices using KPPS refclocks exhibited anomalously poor time synchronization, suggesting that timing error should be continuously monitored and recorded in audio metadata.

are kept by `chronyd`; statistics from four surveys of 21,000 fielded devices conducted during August and September of 2022 are shown in Figure 3. A reasonable cutoff for the maximum allowable timing error is 1 ms. We found that 99.955% of sampled devices using a KPPS refclock met this threshold, for which $\bar{\lambda} = 18.7 \pm 10.5\mu$s. Devices using NTP refclocks due to recent startup or damaged GPS[3] did not achieve sufficient timing accuracy to use in multilateration calculations ($\bar{\lambda} = 71.1 \pm 16.1$ms). A small number (0.045%) of devices using a KPPS refclock exhibited time errors in excess of 1 ms, an unexpected result that suggests a broken control loop. We conclude that KPPS refclocks generally yield excellent results, NTP refclocks are inadequate, and that the current value of error estimate λ should be both continuously monitored for anomalous behavior and included the file metadata.

2.6 Sensor Position Accuracy

Scepter 2 sensors use an on-board GPS chipset to determine their location. After a GPS lock is acquired, the GPS subsystem provides an updated position fix each second. Unlike the realtime ShotSpotter system, which uses a long-term average of position [18], the geolocation reported in file metadata is as-reported by the GPS system. This is in keeping with the philosophy of providing "minimally-processed" data in the digital audio recordings, but it limits the position accuracy of geolocations coded in the wav files to the published datasheet accuracy (50% CEP of 2.0 m per [19].)

[3] Functional GPS is required for realtime location, so such devices are not used to detect gunshots.

3 Source Location

The goal of the source location problem is to estimate the origin of an acoustic signal from measurements on multiple receivers. This is done by comparing the predictions of a sound propagation model with actual signal measured on each receiver and determining the location that is consistent with the physics of acoustic propagation and best fits the available data.

The simplest model for acoustic propagation is one that assumes straight-line travel in a stationary homogeneous medium, for which the arrival time t_i on sensor i satisfies the relationship

$$t_i = \frac{1}{c} d_i + t_o \tag{1}$$

where t_o is the discharge time, d_i is the shooter–sensor distance, and c is the speed of sound. An acceptable expression for c as a function of temperature is $c = 20.03\sqrt{T + 273.15}$ m/s.

We evaluated the accuracy of the model using the public data set collected during a 2018 live fire exercise in Pittsburgh, PA [20]. This exercise comprised nine single-shot and nine three-shot incidents fired at nine different firing locations around the city. A plot of measured travel time $(t_i - t_o)$ vs expected travel time d_i/c is shown in Figure 4. Arrival times t_i are measured arrival times of the first shot in each incident; d_i is the straight-line 3D distance from the surveyed firing point location to each sensor. Discharge time t_o was estimated for each shot using linear regression, and c was computed using air temperature measurements from local meteorological stations[4]. For these data the correlation coefficient is 0.99976, indicating that the straight-line propagation model is appropriate.

Several efficient solutions for locating a source from multiple time differences of arrival (TDoAs) under the straight-line propagation model are available [21, 22]. The number of TDoAs required is at least one more than the solution dimension. When the solution is over-determined, these algorithms return the best-fitting source location in the least-squares sense. As an inverse problem, the accuracy of source location solutions strongly depends on geometry of the receiver array, a topic we will address in future work. Monte Carlo simulation is a useful way to quantify the effect of uncertainty in the inputs on the uncertainty in the computed location [23].

[4]KPIT (18–22 km distant) and KAGC (8–12 km distant).

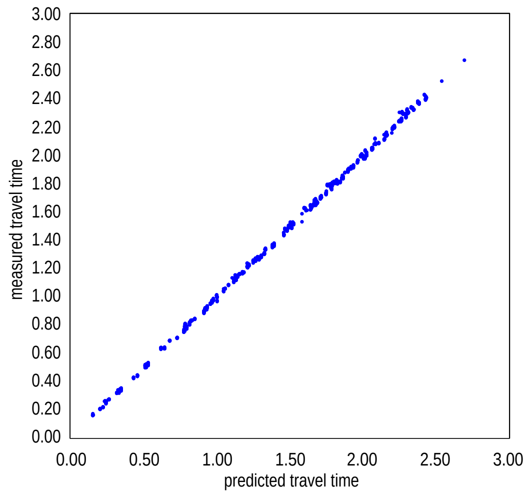

Fig. 4: Measured muzzle blast travel time $(t_i - t_o)$ vs predicted travel time d_i/c.

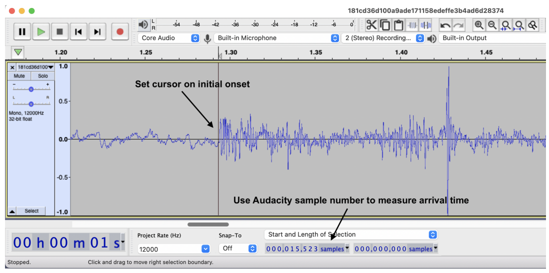

Fig. 5: Using Audacity to determine arrival time of the muzzle blast by selecting a region that starts with the first discernible increase in signal above background.

4 Examples

4.1 Software Tools

As part of the present work, we have released open-source tools for reading ShotSpotter audio files and generating source location estimates from muzzle blast arrival times[5]. Python script `smjx_reader.py` extracts the `smjx` metadata from a ShotSpotter WAV file passed as an argument, while `find_pulses.py` takes a directory of WAV files as an argument and steps the user through the steps necessary to determine a location solution. The directory's audio files are opened

[5]These programs share some data structures and algorithms with production ShotSpotter Respond software, but they are not extracted from or used in that system.

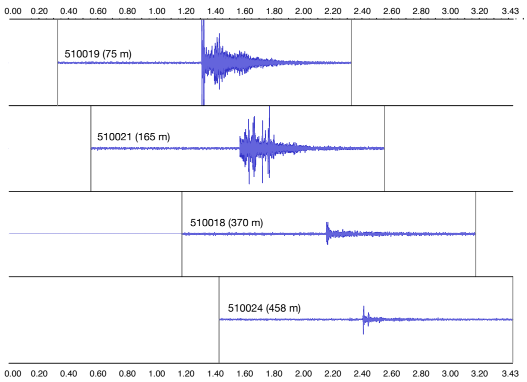

Fig. 6: Single-shot example (#783-556267) with waveforms sorted by distance. Audio is aligned to one second before the onset arrival time on each sensor.

in Audacity [24] one at a time, permitting the user to enter the sample number associated with the start of the impulse for the shot under investigation (Figure 5). After selecting a pulse (or none) for each audio file in the directory, `find_pulses.py` outputs a JSON file containing the arrival time and position data necessary for multilateration. This output is passed to `find_location.py`, which solves the multilateration problem using Reddi's algorithm [21].

Several sets of sample data recorded in active coverage areas are included with the software. The output of the `find_pulses.py` script is easily parsed and can be used with the provided multilateration code or used in the development of other source-location algorithms.

4.2 Single Shot Example

Figure 6 shows reviewer audio from a simple single-impulse incident, #783-556267. The low amplitude of the onset of the impulsive noise on sensor 510021 implies there is an occluded path between the source of the noise and the sensor that detected it. Despite the occluded path, it is easy to identify the initial arrival time of the impulsive sound and the location computed from these four sensors is within 2.5 m of a location computed using 11 reporting sensors from forensic audio data set. This shows that mildly occluded paths can still generate good location results.

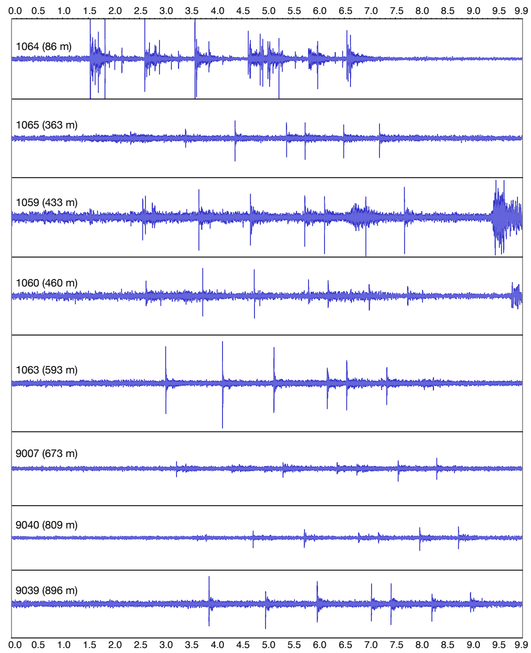

Fig. 7: Seven-shot moving shooter example (#312-780375) with waveforms sorted by distance. Audio is clock-aligned, showing the relative arrival time of the shots on each sensor.

4.3 Moving Shooter Example

Figure 7 shows waveforms from an alleged drive-by shooting incident #312-780375 that occurred in Chicago, IL. This example illustrates clock-aligned audio files generated by the ShotSpotter system. Manual analysis of this incident requires locating the seven shots one at a time. When working with multiple-shot incidents, care must be taken to ensure that echoes from earlier shots are not mistaken as the onset arrival time from later shots. A plot that shows both waveform and spectrogram views (Figure 8) can be helpful in identifying near line-of-sight impulses. Another useful technique is to use the shot spacing from more distant, echo-free recording as a guide when identifying pulses as the shot spacing is invariant with distance for a stationary shooter. The shot spacing will change when the shooter is moving, and the opening or closing of space between shots is invaluable in estimating shooter direction.

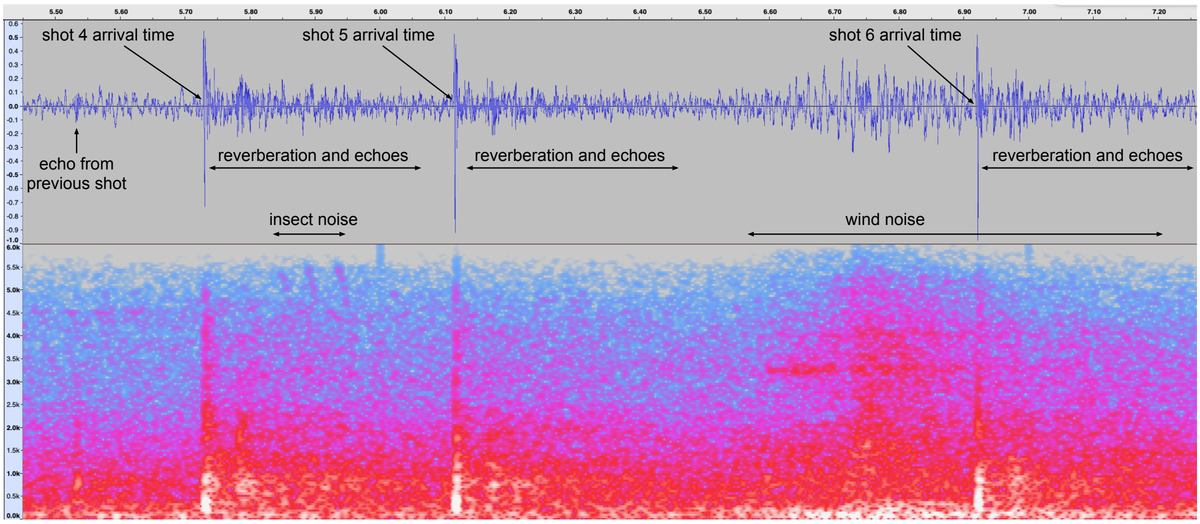

Fig. 8: Closeup waveform/spectrogram view of shots 4, 5, and 6 from sensor 1059 of #312-780375 showing the broad bandwidth of gunshot signals compared with other ambient noises. Screenshot from Audacity [24].

5 Conclusion

The ShotSpotter Respond system creates single-channel digital audio recordings of potential gunshot incidents in the ordinary course of business. System clocks on these devices generally maintain excellent synchronization with UTC via GPS KPPS refclocks, but the presence of a small number of anomalous timing measurements suggests tighter monitoring and reporting of timing statistic in file metadata is appropriate. Similarly, the file metadata should include the positional error estimate associated with instantaneous GPS measurements.

We show it is feasible to estimate the initial arrival time of an impulsive noise using digital audio workstation software. These measurements can be used as inputs into an open-source multilateration program. With suitable receiver geometry it is possible to get excellent source location accuracy via these semi-automated techniques.

References

[1] Global Justice Information Sharing Initiative Advisory Committee, "Video Evidence: A Primer for Prosecutors," Technical report, Bureau of Justice Assistance, Office of Justice Programs, U.S. Department of Justice, 2016, `https://bja.ojp.gov/library/publications/video-evidence-primer-prosecutors`.

[2] Maher, R. C., *Principles of Forensic Audio Analysis*, Springer, 2018.

[3] *ShotSpotter Respond Datasheet*, ShotSpotter, Inc., 2021.

[4] Hagy, D., "Digital evidence in the courtroom: A guide for law enforcement and prosecutors," Technical Report NCJ 211314, National Institute of Justice, Office of Justice Programs, U.S. Department of Justice, 2007.

[5] "Chrony," `https://chrony.tuxfamily.org/`, version 2.1.1.

[6] Mills, D. L., "Internet time synchronization: the network time protocol," *IEEE Transactions on communications*, 39(10), pp. 1482–1493, 1991.

[7] "Linux Sound System Documentation," `https://www.kernel.org/doc/html/latest/sound/index.html`, version 4.4.10.

[8] "FLAC: Free Lossless Audio Codec," `https://xiph.org/flac/`, version 1.3.3.

[9] "ShotSpotter Metadata Specification," `https://github.com/ShotSpotter/research.source-location-shotspotter-audio`, version 1.0.

Fig. 9: Plot of individual shot locations of an alleged moving shooter incident in Chicago, IL made using Google Earth [25].

[10] Scientific Working Group on Digital Evidence, "SWGDE Best Practices for Forensic Audio," `https://drive.google.com/file/d/1zF0KzO5Xe5jAO2CgLxISIm8fLSFJ46ax/view`, 2022-06-09.

[11] "Using versioning in S3 buckets," `https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html`, version 2006-03-01.

[12] "Amazon Simple Storage Service (S3) API Reference," `https://docs.aws.amazon.com/AmazonS3/latest/API/s3-api.pdf`, version 2006-03-01.

[13] "JSON Schema," `https://json-schema.org`, 2020-12.

[14] Bossler, J. D., Campbell, J. B., McMaster, R. B., and Rizos, C., *Manual of geospatial science and technology*, CRC Press, 2010.

[15] Beck, S. D., "Who fired when: Associating multiple audio events from uncalibrated receivers," in *Audio Engineering Society Conference: 2019 AES International Conference on Audio Forensics*, Audio Engineering Society, 2019.

[16] Mills, D. L., "On the Accuracy and Stability of Clocks Synchronized by the Network Time Protocol in the Internet System," *ACM SIGCOMM Computer Communication Review*, 20(1), pp. 65–75, 1989.

[17] Mills, D. L., Martin, J., Burbank, J., and Kasch, W., "Network Time Protocol Version 4: Protocol and Algorithms Specification," Technical report, RFC 5905, 2010.

[18] Calhoun, R. B., Dunson, C., Johnson, M. L., Lamkin, S. R., Lewis, W. R., Showen, R. L., Sompel, M. A., and Wollman, L. P., "Precision and accuracy of acoustic gunshot location in an urban environment," 2021, doi:10.48550/ARXIV.2108.07377.

[19] *MAX-7: u-blox 7 GNSS modules Data Sheet*, u-blox, 2014, rev. 05.

[20] "ShotSpotter Sample Live Fire Test Data," `https://github.com/ShotSpotter/research.accuracy-of-gunshot-location`, 2019.

[21] Reddi, S. S., "An Exact Solution to Range Computation with Time Delay Information for Arbitrary Array Geometries," *IEEE Transactions on Signal Processing*, 41(1), pp. 485–486, 1993.

[22] Mathias, A., Leonardi, M., and Galati, G., "An efficient multilateration algorithm," in *Digital Communications-Enhanced Surveillance of Aircraft and Vehicles, 2008. TIWDC/ESAV 2008. Tyrrhenian International Workshop on*, pp. 1–6, IEEE, 2008.

[23] Maher, R. C. and Hoerr, E., "Audio forensic gunshot analysis and multilateration," in *Audio Engineering Society Convention 145*, Audio Engineering Society, 2018.

[24] "Audacity: free, open source, cross-platform audio software," `https://www.audacityteam.org`, version 2.4.2.

[25] "Google Earth," `https://www.google.com/earth/versions/`, version 7.3.3.7786.