Enabling Automatic Gunshot Detection and First Responders Dispatch for Safer Communities

A Technical Paper prepared for SCTE•ISBE by

Wael Guibene
Director – Wireless R&D
Charter Communications
6360 S Fiddlers Green Cir, Englewood, CO 80111
Wael.Guibene@charter.com

Hossam Hmimy
Sr. Director – Wireless R&D
Charter Communications
6360 S Fiddlers Green Cir, Englewood, CO 80111
Hossam.Hmimy@charter.com

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

1. Introduction

Gunshot detection has become a major request from many cities deploying smart city solutions. Current state of the art solutions requires a human interaction mechanism to detect the gunshot incident and number of shots through an operations center (OC). Based on this requirement, current solutions may introduce latencies ranging from 5-10 minutes from the moment a gunshot is detected to alerting of first responders.

According to the FBI, about 70% of active shooter situations are over in under five minutes [1], and the website of National Sheriffs' Association states that "shaving even seconds off the notification and response times can result in vastly different outcomes in these situations." [2]

The response time and latency introduced by existing solutions cannot be reduced as the OC in the loop introduces a human factor that needs to listen to the scene before calling and manually dispatching first responders.

In this paper we introduce a novel gunshot detection mechanism that is fully automated. Our paper describes the software platform along with algorithms that enable:

- Detecting, classifying and localizing gunshots (Audio)
- Recognizing the shooter(s) in the scene (Video)
- Recording the scene from different angles (Video)
- Sending messages in real time to first responders dispatch center as well as authorized personnel with the accurate location and short video recording of the scene.

Our approach applies machine learning (ML) algorithms at the edge to detect and discriminate gunshots from indoor/outdoor ambient sound. When gunshots are detected, the sound direction is also derived, and security cameras follow the direction of the sound to capture the scene, identify the shooter(s) and automatically inform responders about the incident, thus providing significantly improved situational awareness.

2. AI-based Gunshot Detection

2.1. Hardware Platform

In our approach, we rely on machine learning and artificial intelligence (AI) algorithms to isolate and identify the gunshots from other sounds.

Our approach consists of hardware (HW) and software (SW) platforms enabling automatic gunshot detection, localization of the active shooting incident and real-time communication to police and first responders.

Figure 1 below depicts the HW platform controlling the cameras.

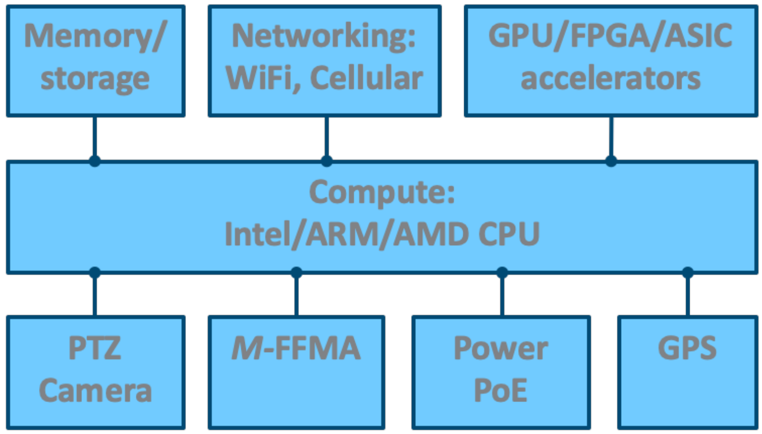

Figure 1 - HW platform for automatic gunshot detection

The cameras and their controllers are mounted on street fixtures, e.g. lighting poles or intersections poles.

The HW platform consist of the following components:

- Compute element (central processing unit (CPU) / microprocessor unit (MPU)
- Memory for internal storage
- Networking: either Ethernet, Wi-Fi, cable or cellular for backhauling
- Accelerators enabling different algorithms: graphic processor unit (GPU) for image processing, FPGA for audio and mic-array management, ASIC for image processing acceleration.
- *M*-element far-field microphone array (M-FFMA)
- Pan-Tilt-Zoom (PTZ) Camera: to enable dynamic scene acquisition
- PoE: to enable simultaneous communication and power delivery
- Global Positioning System (GPS): to localize the HW platform and fine timestamp events of active shooting

2.2. Software and AI Platform

Figure 2 below describes the end-to-end flow of the automated gunshot detection and reporting framework.

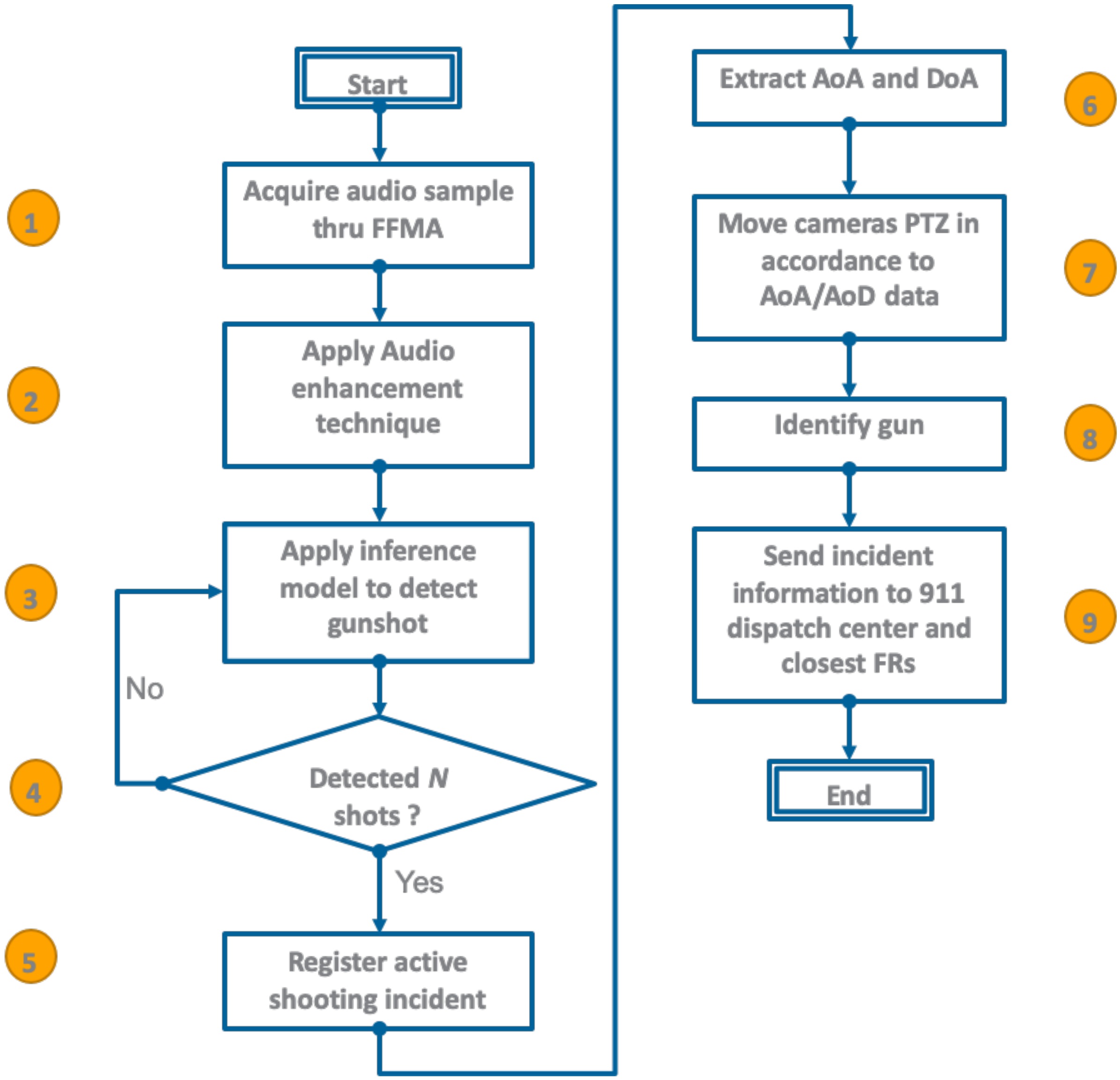

Figure 2 - SW and AI stack for gunshot detection

1. Audio samples are acquired through the M-element far field microphone array (FFMA). The FFMA allows active noise cancellation (ANC) and direction of arrival (DoA)/angle of arrival (AoA) algorithms to run simultaneously with high accuracy.
2. Given that the gunshots have spike-like behavior in the audio temporal domain, we introduce the algebraic detector (AD) to act as both a detector and an ANC for the audio captured by the microphone array. This will increase the accuracy of the ML inference model that detects the shots. The AD is a pre-processing step that locates the shot in time domain prior to applying our ML model. This will reduce the $P_{\text{False Alarm}}$ for the end-to-end (E2E) application.

One of the metrics that confirms the accuracy of the AD as a joint ANC and denoising technique is highlighted in the receiver operating characteristic (ROC) curve in figure 4 below and the example of gunshot detections in time domain. From the ROC curves, up to 70% accuracy at 1% false alarm rates are achieved using just 30 samples. This will scale to over 95% accuracy with a constant probability of false alarms at less than 1% at full MPEG Audio Layer-3 (MP3) audio quality (44100 samples).

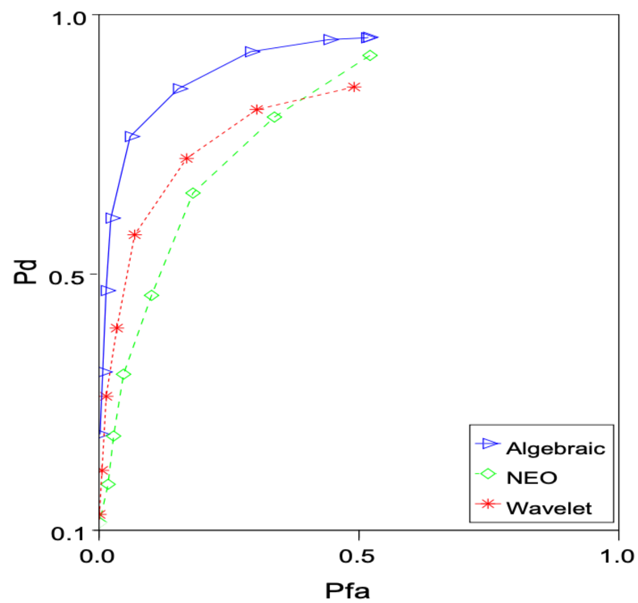

Figure 3 - ROC curves comparison SoTA vs. AD

We can see from the Figure above that the ROC curve associated with AD outperforms both NEO and Wavelet detection algorithms.

Another metric that helps assess the AD performance is an example of gunshot detection in time domain and the capability of the AD to denoise and separate the spike-like behavior of the gunshots from other noise at a location.

The Figure below shows an example of ground truth and isolated gunshots from city noise. The AD can run either offline or in real time to simultaneously reduce noise and locate spike-like shots in time domain.

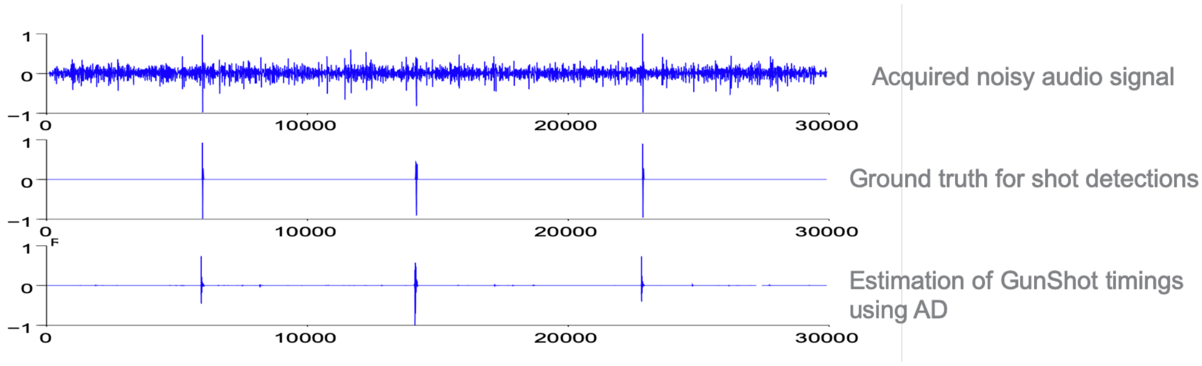

Figure 4 - Time domain gunshot detection and audio denoising

3. After detecting the likelihood timing of the shot, we apply a pre-trained model on gunshots over the cleaned audio signal. The pre-processing ANC through AD step allows for cleaner, better audio signal to feed into the ML classifier and allows better results with fewer false alarms for the E2E application.
4. If a gunshot is detected, a counter and a timer turn on. The counter is set to a value of N shots and the timer is set to P seconds. We set these conditions to further ensure robustness of the solution. As we do not trigger an alert at each shot, we need to aggregate N shots within P seconds to classify the event as an active shooting incident and trigger the rest of the algorithm. Shots fired from different people are processed concurrently.
5. If the algorithms register N shots within P seconds, it triggers an active shooting event. The algorithm will keep a counter increment of each shot it is detecting in the background.
6. Using the FFMA capabilities to run AoA and DoA, the system infers the angle and direction from which the shots are being fired.
7. The system issues pan-tilt-zoom (PTZ) “OnVif” commands to the camera to pan and tilt to the direction of the active shooting. The zoom is calibrated thru the Received Signal Strength Indicator (RSSI) of the audio track to estimate how far the shooting is from the camera and the microphone array. Depending on the deployment settings, the gunshot is most probably detected by more than one system. Each system is commissioned with its GPS coordinates and the shot is either identified by the closest detection system (if there is only one system deployed) or by triangulation of the different systems. The system starts recording and storing the scene in the camera’s internal storage as well as exposes the live stream thru real time streaming protocol (RTSP). The system also runs on-camera facial recognition and captures all faces in scene.
8. Another advantage of our approach is the capability to recognize guns by correlating the noise-free sound provided by the AD and ANC to an on-device database containing the sound signature of the most sold guns.
9. A standardized report is created by the system and sent to first responders. An example of the aggregated data sent by the system is as described by the following JavaScript Object Notation (JSON) meta-data, in Figure 5, with the following fields:
    - **NB_Shots_Per_Platform**: identifies each HW platform that detected the shooting as well as the number of shots detected per platform.
    - **AS_GPS_Coordinates**: either the trilateration GPS location of the shooting or the closest HW platform that detected the active shooting.
    - **Timestamp**: timestamp at which the alert was sent.
    - **Video_Streams**: links to the live-feeds from each HW platform that detected the active shooting for first responder and 911 network operations center (NOC) to evaluate in

real time the logistics needed to handle the active shooter (casualties, injuries, number of shooters, number of first responder units needed, etc.).

- **Faces_Captured_At_Scene**: as the camera is capturing the scene, we run in real time face detection of all people present at the scene and store and send images as part of the JSON in base64 encoding.
- **Gun_Type**: is the identified gun involved in the shooting alongside its presence likelihood


```
{
  "NB_Shots_Per_Platform": {
    "DeviceID_1" : "15",
    "DeviceID_2" : "10",
    "DeviceID_3" : "15"
  },
  "AS_GPS_Cordinates": {
    "longitude": "12.3456",
    "latitude" : "78.91011"
  },
  "timestamp" : "123456789",
  "Video_Streams": {
    "DeviceID_1_RT_STREAM" : "rtsp://user:password@DeviceID_1.IPaddress:554/access-media/media.amp?videocodec=h264&audiocodec=aac",
    "DeviceID_2_RT_STREAM" : "rtsp://user:password@DeviceID_2.IPaddress:554/access-media/media.amp?videocodec=h264&audiocodec=aac",
    "DeviceID_3_RT_STREAM" : "rtsp://user:password@DeviceID_3.IPaddress:554/access-media/media.amp?videocodec=h264&audiocodec=aac"
   },
  "Faces_Captured_At_Scene" : {
    "FID_1" : "9j4AAQSkZJRgABAgAAZABkAAD/7AARRHVja3kAAQAEAAAAPA+4ADkFkb2JlAGTAAAA
      ...XA5MNt0VAyYRlttHMq8cpEJZfHt7FoAoFAoFAoFAoFAoFAoFAoFAoFAoFAoFAoFAoFAoFAoFAoFAoFAoP/Z",
    "FID_2" : "....",
    "FID_3" : "..."
  },
  "Gun_Type" :{
    "guntype_1" :  "likelihood percentage"
  }

}
```


Figure 5 - Example meta-data of gunshot detection and first responder dispatch

3. Example Deployment of the Gunshot Detection System

In this section, we give concrete examples and results from a proof of concept (PoC) that is running the gunshot detection system indoors in an office setting.

Our HW platform is built on top of GPU-accelerated embedded single board computer and we also show the FFMA running the AD for noise cancelation and DoA/AoA algorithms in Figures (6) and (7), respectively.

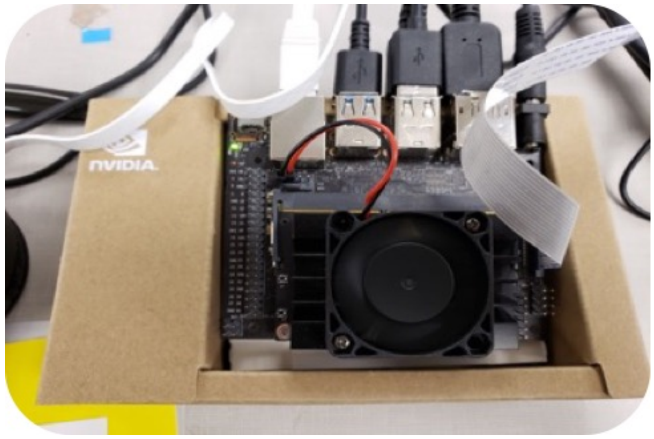

Figure 6 - SBC HW platform

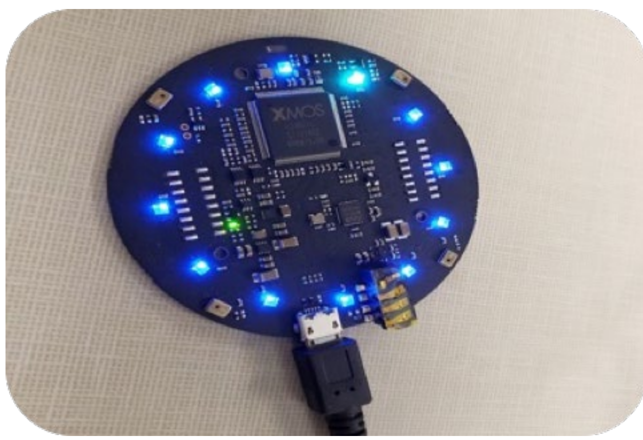

Figure 7 - FFMA running AD and AoA/DoA

Figure (8) shows one of the PTZ cameras used that is connected and controlled by the automatic gunshot detection system.

Figure 8 - PTZ camera example

4. Conclusion

In this paper, we presented a unique and novel approach to an AI-assisted automatic gunshot detection and reporting system. Our approach enables fully-autonomous isolation of gunshot sounds and recognition of gun types as well.

We have deployed and tested our solution in our lab using recorded gunshots to test the resiliency of the algorithms and test the end-to-end performance and latency of the solution. Our tests show conclusive results on the detection time: near real-time detection with under two seconds to detect an active shooting situation, take photos of the likely shooter and the scene and send them to first responders and police for dispatch.

Future work includes testing the system using real gun shots in the shooting range, as well as collecting more gun sound samples from the shooting range to enhance the model.

5. Bibliography & References

[1] ShotSpotter: Home [https://www.shotspotter.com/]

[2] Blair, J. Pete, and Schweit, Katherine W. A Study of Active Shooter Incidents, 2000-2013. Texas State University and the Federal Bureau of Investigation, U.S. Department of Justice, Washington D.C., 2014.

[3] "Embracing Technology to Decrease Law Enforcement Response Time", Feb. 18, 2016. National Sheriffs' Association (www.sheriffs.org).

