Shooter Localization Using Social Media Videos

Junwei Liang
junweil@cs.cmu.edu
Carnegie Mellon University

Jay D. Aronson
aronson@andrew.cmu.edu
Carnegie Mellon University

Alexander Hauptmann
alex@cs.cmu.edu
Carnegie Mellon University

ABSTRACT

Nowadays a huge number of user-generated videos are uploaded to social media every second, capturing glimpses of events all over the world. These videos provide important and useful information for reconstructing events like the Las Vegas Shooting in 2017. In this paper, we describe a system that can localize the shooter location only based on a couple of user-generated videos that capture the gunshot sound. Our system first utilizes established video analysis techniques like video synchronization and gunshot temporal localization to organize the unstructured social media videos for users to understand the event effectively. By combining multimodal information from visual, audio and geo-locations, our system can then visualize all possible locations of the shooter in the map. Our system provides a web interface for human-in-the-loop verification to ensure accurate estimations. We present the results of estimating the shooter's location of the Las Vegas Shooting in 2017 and show that our system is able to get accurate location using only the first few gunshots. The full technical report, all relevant source code including the web interface and machine learning models are available [1].

KEYWORDS

Event Reconstruction, Video Synchronization, Video Analysis, Audio Signal Processing, Gunshot Detection, Shooter Localization

ACM Reference Format:
Junwei Liang, Jay D. Aronson, and Alexander Hauptmann. 2019. Shooter Localization Using Social Media Videos. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3343031.3350536

1 INTRODUCTION

With the growing use of camera phones all over the world, public events can now be captured and shared via social media instantly. In a big public event with a large crowd of people, video recordings would capture different moments of the event at different positions from different perspectives. These large amount of videos enable research in semantic concept detection [10, 16–18], video captioning [5, 6], intelligent question answering [4, 9, 13], 3D event reconstruction [1, 11], and activity detection [2, 14]. These videos also provide important information for the authorities if a public safety event occurs. For example, Boston Marathon Bombing, Dallas Shooting and Las Vegas Shooting all have hundreds or even thousands of attendees upload videos of the event that could be useful for first-responders and investigators. However these consumer videos are captured "in the wild", often with few metadata that we could recover once they are uploaded to social media [3]. These videos are noisy and sometimes with low quality. Analysts often need to go over a large number of these videos as useful information about the event may spread across different segments of different videos.

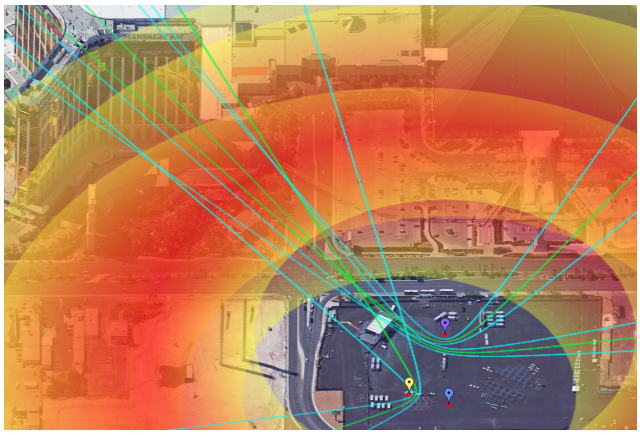

Figure 1: Visualization results of shooter localization using the our system for Las Vegas Shooting in 2017. It is computed based on only three video recordings as marked on the map and single gunshot. The red and yellow donut is the heatmap probability of the shooter location in horizontal distance. Our system also estimates that the shooter is likely to be within the light blue hyperbola lines. As we see, the overlapping area of all estimations points to the shooter's actual location - the north wing of the Mandalay Bay Hotel.

In this paper, we build the shooter localization system to solve this problem, which utilizes machine learning models like video synchronization [12] for event reconstruction [1, 8, 11] and gunshot detection [15, 19, 20] to help analysts quickly find relevant video segments to look at. we present how the our system can geo-localize the shooter given a few video recordings that only captures the gun shot sound, as shown in Figure 1. Our system first uses automatic video synchronization [12] and a web interface for manual refinements to put all unstructured videos into a global timeline. Our interface allows users to engage with multimodal information effectively to understand the event. Then our system performs automatic gunshot detection [15] to temporally localize the gunshot segments within each video. Based on the supersonic

[1] https://vera.cs.cmu.edu

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350536

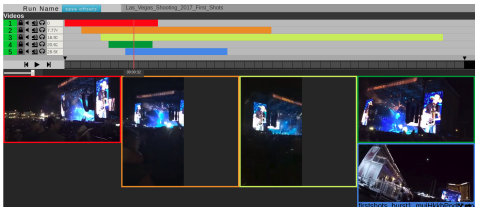

Figure 2: Video synchronization into a global timeline.

bullet physics [21] and a more general sound travel physics as described in Section 3, our system identifies the exact time of each video hearing the shockwave sound and muzzle blast sound, then computes the possible distances and directions of the shooter from each of the videos. After putting each video on the map, our system can visualize all possible areas of the shooter location.

Our system provides the first open source framework with human-in-the-loop paradigm to solve this problem. Our contributions are three folds. Firstly, we propose a framework that is physically sound and has been verified on real events for shooter localization using only social media videos without any metadata. Secondly, we provide a web interface that allows human verification at each step to ensure the accuracy of the estimations. Last but not least, we point out important areas within our framework that call for future researchers and engineers to build models to reduce human efforts and improve the system.

This paper is organized as follows. In Section 2, we describe the video synchronization module. In Section 3, we describe the gunshot detection system and how we can estimate shooter location based on gunshot sound. In Section 4 we explain our system architecture. In Section 5 we discuss the future directions to improve the system.

2 VIDEO SYNCHRONIZATION

We don't assume the videos from social media have any metadata like global time stamps or GPS that we could use. Therefore we will need put the videos into a global timeline first. After user uploads all relevant videos to our system, an automatic video synchronization model is utilized to organize the videos. Currently, the automatic system synchronizes the videos using sound. Please refer to [12] for more technical details. However, in order to use these videos for shooter localization, we need the videos synchronized based on visual cues, as light travels much faster than sound. Our system provides an easy-to-use interface, in which users can manually verify and synchronize pairs of videos as precise as at the frame-level. Assuming the users match the video pair to the exact frame pairs and the video FPS is 30, the error margin of the synchronization is within 33 milliseconds. The pairwise synchronization results are aggregated automatically into global results as shown in Figure 2. Users can play the videos in a global timeline to understand the events in a coherent manner.

3 SHOOTER LOCALIZATION

3.1 High-level Design

Now that we have all the videos put into a global timeline, we could estimate the shooter distances from each of the videos based on Section 3.2 if the bullet is supersonic and for each pair of the videos we could estimate the shooter directions and locations based on the

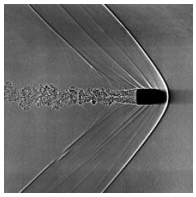

Figure 3: A shadowgraph of a supersonic bullet. Taken from wikipedia.

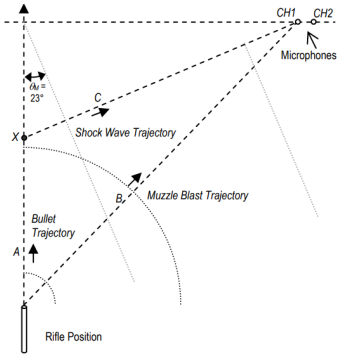

Figure 4: The physics model of how the shockwave sound and muzzle blast sound of a supersonic bullet reach the camera. Taken from [21].

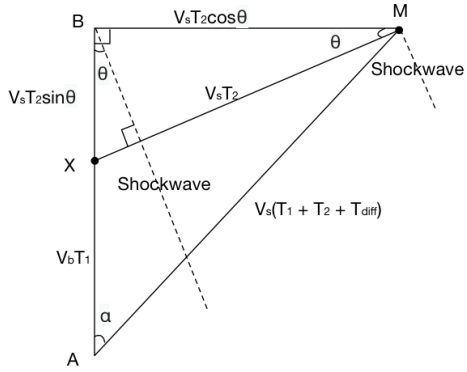

Figure 5: Method 1 math notation.

time differences of the muzzle blast sound reaching the two videos as explained in Section 3.3. These aforementioned estimations are computed for each video (Method 1) and each pair of videos (Method 2), for one single gunshot. Each estimation provides an area of possible locations and the area where all estimations overlap is the most likely location of the shooter. Users can apply such estimations to multiple gunshots to get even more accurate localization results.

3.2 Method 1

This method requires the bullet to be supersonic. The main idea is that a supersonic bullet creates two distinctive sounds, shockwave sound and muzzle blast sound, if identified temporally, one can use the time difference of the two sound reaching the camera to estimate the distance between the camera and where the bullet is fired from.

Figure 3 shows a shadowgraph of a supersonic bullet. The shadowgraph basically shows how the air looks like when a bullet is travelling beyond the speed of sound. As we see, the bullet creates a cone-like shockwave wall that expands as the bullet travels. When this wall arrives the camera, it records the shockwave sound. The physics model of how the shockwave sound and muzzle blast sound of a supersonic bullet reach the camera is shown in Figure 4. For more details, please refer to [21].

Based on the physics model, we can derive the computation graph as in Figure 5. Suppose V_s is the speed of sound, V_b is the speed of the bullet, α is the angle between the camera to the shooter and the bullet trajectory. T_{diff} is the time difference between the camera records the shockwave sound and the muzzle blast sound. For the camera to record the shockwave sound, after the bullet is fired, the bullet travels T_1 under V_b to point X, and then the shockwave travels at speed of sound V_s for time T_2 to reach the camera. We have:

$$\begin{aligned} AB &= V_b T_1 + V_s T_2 sin\theta = V_s(T_1 + T_2 + T_{diff})cos\alpha \\ BM &= V_s T_2 sin\theta = V_s(T_1 + T_2 + T_{diff})sin\alpha \end{aligned} \tag{1}$$

Hence, we have:

$$\begin{aligned} (V_b - V_s cos\alpha)T_1 + (V_s sin\theta - V_s cos\alpha)T_2 &= V_s T_{diff} cos\alpha \\ (-V_s sin\theta)T_1 + (V_s sin\theta - V_s sin\alpha)T_2 &= V_s T_{diff} sin\alpha \end{aligned} \tag{2}$$

In Eq 2, the unknown variables are T_1 and T_2. Based on the two equations we can solve T_1 and T_2. Then the distance from the camera to the shooter, i.e. AM, could be computed by $V_s(T_1 + T_2 + T_{diff})$. Currently in our system, we ask users to input the range of the speed of sound V_s, the speed of the bullet V_b and the angle α. In practise the range of α is usually set from zero to fifteen, which already covers 30 degrees of freedom since the graph could be flipped. θ is given by $arcsin(V_s/V_b)$ according to [21]. T_{diff} is currently marked by the users, aided by a spectrogram of the gunshot sound in the web interface. Since we have a range of V_s, V_b and α, we use the Monte Carlo method (random sampling) to uniformly sample a value for each variables at a time to get T_1 and T_2, repeat many times (10k for example) and then report back the minimum, maximum and mean of the distance D. Please refer to the code for more details. Note that the distance is direct distance from the camera to the shooter. In order to have accurate visualization on the map which requires **horizontal** distance D_h, users can enter the elevation of the shooter D_e, and the horizontal distance is computed by $D_h = \sqrt{D^2 - {D_e}^2}$. Hence we get a donut-like possible area of the shooter for each video as shown in Figure 7. In future work, we could automate the process of getting V_b by automatic gun type detection based on the gunshot sound. We could get the speed of sound if we can estimate the temperature of the event location.

It is important to note that we assume the bullet travels at constant speed V_b until the camera hear the shockwave sound. Clearly this assumption is bad since bullet speed may drop to its half after traveling for 700 meters. Currently in the our system, we recommend users to use a wider range of bullet speed to compensate for this assumption.

To sum up, Method 1 operates under the assumption that the bullet is supersonic, constant speed, and the users can reliably mark out the time of the shockwave sound and the muzzle blast sound on the spectrogram of the video.

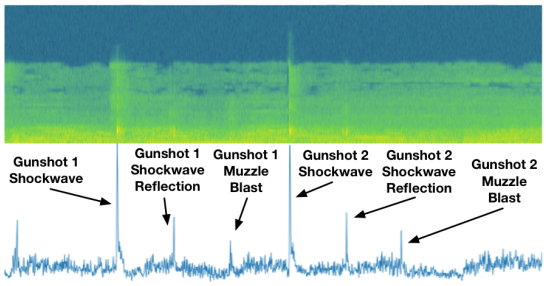

Figure 6: Example of gunshot spectrogram and power graph on the web interface.

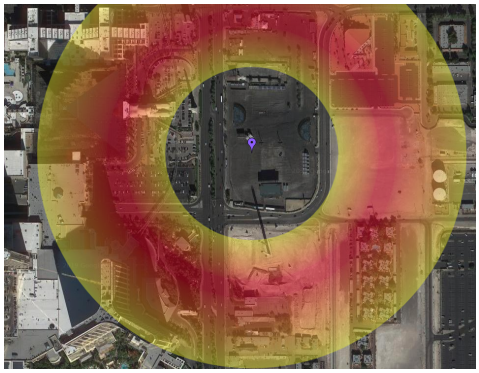

Figure 7: Example shooter localization using Method 1.

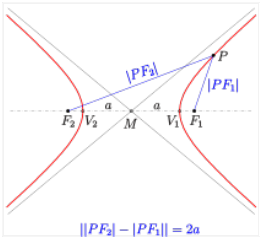

Figure 8: Hyperbola math notation. Taken from wikipedia.

3.3 Method 2

This method applies to a pair of videos that capture the muzzle blast sound of the gunshot, which includes all types of gunshot sound or sound in general. Method 2 makes use of the definition of a hyperbola as shown in Figure 8. The points (P) anywhere on the hyperbola satisfy that $||PF_2|-|PF_1|| = 2a$, and satisfy $|PF_2|-|PF_1| = 2a$ if we only consider points on the right part of the hyperbola. In the shooter localization case, for each pair of videos, since we have synchronized them and mark the muzzle blast sound in the videos' timeline, we know the time difference between video 1 and video 2 hearing the gunshot T_{diff}. Given the speed of sound V_s, essentially we can compute the value of $2a = V_s T_{diff}$. After we put the two video camera locations on the map (F_1 and F_2), we can draw a hyperbola, where the shooter is likely on.

As shown in Figure 9, we can see three hyperbola lines. Recall in Section 2 we mention that the error margin of the video synchronization is 33 milliseconds, given that the frame matching could be off by half a frame. Also, currently in our system we ask users to enter the range of the speed of sound. Therefore we draw three hyperbola lines using three different value of $2a$: $V_{s_{min}}(T_{diff} - 0.033)$,

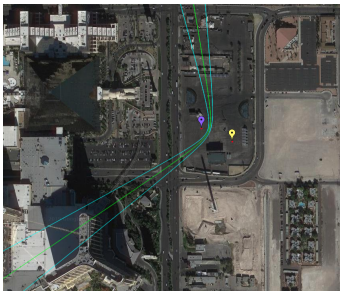

Figure 9: Example shooter localization using Method 2. $(V_{s_{min}} + V_{s_{max}})(T_{diff} - 0.033)/2.0$ and $V_{s_{max}}(T_{diff} + 0.033)$. The second hyperbola is green colored while the others are light blue. The shooter is possible to be within the light blue lines, with the most likely locations are on the green line.

To sum up, method 2 relies on accurate video synchronization, camera locations and markings of the muzzle blast sound.

3.4 Comparing Method 1 and 2

When testing the system in real-world scenario like the Las Vegas Shooting, we find that method 1 is sensitive to the timing of shockwave sound and the muzzle blast sound, i.e., method 1 requires that T_{diff} to be accurate, while method 2 estimation is sensitive to the camera locations. Meanwhile method 1 is not sensitive to the errors of the camera locations.

3.5 Camera Locations

Currently in our system, we provide a Google Map interface and ask the users to **manually** mark the camera locations **at the time of** the video hearing the gunshots. In future work, we could utilize Google Street View images or 45 degree view images to automatically match video frames to a GPS hence we get the camera locations without manual labor.

4 SYSTEM ARCHITECTURE

We utilize production-ready web server - Apache server for serving the web requests and a flexible back-end Python server to leverage multi-CPU and multi-GPU computing cluster. Future researchers could plug their machine learning components into the system seamlessly and efficiently. Please refer to the Github site and the technical report [7] for more details.

5 CONCLUSION AND FUTURE WORK

In this paper, we present our shooter localization system. We demonstrate that our framework combined with machine learning and physics models can help geo-localize the shooter using only unstructured videos from social media. We show that with only three videos, we are able to correctly localize the shooter location in the case of Las Vegas Shooting in 2017. Currently our shooter localization system has many parts that require human operation. In future work, we plan to make those parts automatic.

Acknowledgements This work was partially supported by the financial assistance award 60NANB17D156 from U.S. Department of Commerce, National Institute of Standards and Technology (NIST). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation/herein. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NIST, DOI/IBC, or the U.S. Government.

REFERENCES

[1] Jia Chen, Junwei Liang, Han Lu, Shoou-I Yu, and Alexander Hauptmann. 2016. Videos from the 2013 Boston Marathon: An Event Reconstruction Dataset for Synchronization and Localization. (2016).

[2] Jia Chen, Jiang Liu, Junwei Liang, Ting-Yao Hu, Wei Ke, Wayner Barrios, Dong Huang, and Alexander G Hauptmann. 2019. Minding the Gaps in a Video Action Analysis Pipeline. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. IEEE, 41–46.

[3] Po-Yao Huang, Junwei Liang, Jean-Baptiste Lamare, and Alexander G Hauptmann. 2018. Multimodal Filtering of Social Media for Temporal Monitoring and Event Analysis. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*. ACM, 450–457.

[4] Lu Jiang, Junwei Liang, Liangliang Cao, Yannis Kalantidis, Sachin Farfade, and Alexander Hauptmann. 2017. Memexqa: Visual memex question answering. *arXiv preprint arXiv:1708.01336* (2017).

[5] Qin Jin and Junwei Liang. 2016. Video description generation using audio and visual cues. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*. ACM, 239–242.

[6] Qin Jin, Junwei Liang, and Xiaozhu Lin. 2016. Generating Natural Video Descriptions via Multimodal Processing. In *Interspeech 2016*. 570–574.

[7] Junwei Liang, Jay D Aronson, and Alexander Hauptmann. 2019. Technical Report of the Video Event Reconstruction and Analysis (VERA) System-Shooter Localization, Models, Interface, and Beyond. *arXiv preprint arXiv:1905.13313* (2019).

[8] Junwei Liang, Susanne Burger, Alex Hauptmann, and Jay D Aronson. 2016. Video Synchronization and Sound Search for Human Rights Documentation and Conflict Monitoring. (2016).

[9] Junwei Liang, Liangliang Cao, Yannis Kalantidis, Li-Jia Li, Alexander G Hauptmann, et al. 2019. Focal Visual-Text Attention for Memex Question Answering. *IEEE transactions on pattern analysis and machine intelligence* (2019).

[10] Junwei Liang, Jia Chen, PY Huang, XC Li, Lu Jiang, ZZ Lan, PB Pan, HH Fan, Q Jin, J Sun, et al. 2016. Informedia@ Trecvid 2016. In *TRECVID 2016 Workshop. Gaithersburg, MD, USA*.

[11] Junwei Liang, Desai Fan, Han Lu, Poyao Huang, Jia Chen, Lu Jiang, and Alexander Hauptmann. 2017. An event reconstruction tool for conflict monitoring using social media. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[12] Junwei Liang, Poyao Huang, Jia Chen, and Alexander Hauptmann. 2017. Synchronization for multi-perspective videos in the wild. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1592–1596.

[13] Junwei Liang, Lu Jiang, Liangliang Cao, Li-Jia Li, and Alexander G Hauptmann. 2018. Focal visual-text attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6135–6143.

[14] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G. Hauptmann, and Li Fei-Fei. 2019. Peeking Into the Future: Predicting Future Person Activities and Locations in Videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[15] Junwei Liang, Lu Jiang, and Alexander Hauptmann. 2017. Temporal localization of audio events for conflict monitoring in social media. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1597–1601.

[16] Junwei Liang, Lu Jiang, and Alexander Hauptmann. 2017. Webly-supervised learning of multimodal video detectors. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[17] Junwei Liang, Lu Jiang, Deyu Meng, and Alexander Hauptmann. 2016. Exploiting multi-modal curriculum in noisy web data for large-scale concept learning. *arXiv preprint arXiv:1607.04780* (2016).

[18] Junwei Liang, Lu Jiang, Deyu Meng, and Alexander Hauptmann. 2017. Leveraging Multi-modal Prior Knowledge for Large-scale Concept Learning in Noisy Web Data. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. ACM, 32–40.

[19] Junwei Liang, Qin Jin, Xixi He, Gang Yang, Jieping Xu, and Xirong Li. 2014. Semantic Concept Annotation of Consumer Videos at Frame-Level Using Audio. In *Pacific Rim Conference on Multimedia*. Springer, 113–122.

[20] Junwei Liang, Qin Jin, Xixi He, Gang Yang, Jieping Xu, and Xirong Li. 2015. Detecting semantic concepts in consumer videos using audio. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2279–2283.

[21] Robert C Maher and Steven R Shaw. 2008. Deciphering gunshot recordings. In *Audio Engineering Society Conference: 33rd International Conference: Audio Forensics-Theory and Practice*. Audio Engineering Society.