BLIND GPS LOCALIZATION: ENHANCING POSITIONING ACCURACY IN GPS-DENIED ENVIRONMENTS

Cismas Alexandru[1], Cismas Ioana[2], Popescu Decebal[3]

This paper presents a method for locating a subject using the triangulation method. Three drones were used to transmit signals to a point, with the research focusing on identifying and mitigating potential delays that affect the determination of Time of Arrival (ToA). The study details, through precise measurements, the delays caused by communication protocols, the execution time of instructions on microcontrollers, and the modulation time at the radio module level. The obtained results highlight the feasibility of performing measurements while acknowledging the trade-off in precision they entail.

Keywords: localization, drone, Time of Arrival (ToA), processing delays, microcontrollers (MCU), communication protocol

1. Introduction

We are going through a difficult period full of wars and weapons that are more and more sophisticated. The UAV industry is in continuous development, and one of the technological directions is the identification of new positioning methods that are independent of satellites. During combat, one of the main objectives is to destroy communications. An enemy that is not able to communicate or convey its position will not be able to follow effective tactics or battle strategies. Organization before and during a fight ensures a predictable course and mitigates the loss of human life. Information from the field is vital to ensure this organization. One of these pieces of information is the position of specific equipment or soldiers, and to be able to determine the position, we have several means at our disposal. The primary way of positioning is GPS, but with the reception of satellite signals as its operating principle, there is equipment that is capable of jamming exactly these signals.

[1]Eng., Faculty of Computer Science, UNST Politenica Bucuresti, Romania, email: cismas.alex12@gmail.com

[2]Eng, Military Technical Academy, Bucharest, Romania, email: ioanamtei06@gmail.com

[3]Professor Doctor Ing., Faculty of Computer Science, UNST Politenica Bucuresti, Romania, email: decebal.popescu@upb.ro

Localization, in essence, is a fundamental aspect of navigation, enabling UAVs to perform a diverse range of complex tasks, including mapping, surveillance, search, rescue, and precise delivery. The challenge of localization is not only to determine geographical coordinates but also to understand the orientation, speed, and trajectory of the UAV in a three-dimensional space. The complexity of this task is amplified by the dynamic environments in which UAVs often operate, requiring robust and adaptive localization solutions.

The motivation for research in this area is driven by the growing demand for drones that can operate autonomously in diverse environments. As UAVs become more prevalent, accurate and reliable localization becomes essential to ensure safety, efficiency, and compliance with regulatory standards. In addition, as drones are increasingly integrated into the airspace with human-crewed aircraft, the accuracy of location systems becomes crucial for maintaining the integrity of air traffic control systems and avoiding collisions.

Advances have influenced the evolution of location techniques in sensor technology, signal processing, and machine learning. Traditional localization methods, such as GPS-based systems, are being complemented or replaced by sophisticated algorithms that exploit visual, acoustic, and inertial cues. These emerging approaches offer increased resilience to signal degradation, environmental obstacles, and interference, which are limitations of GPS-dependent methods.

In addition, integrating artificial intelligence and computer vision has introduced the possibility of semantic localization, where UAVs can understand and interact with their environment based on visual recognition of landmarks and features. This capability improves location accuracy and extends the operational range of UAVs into areas without GPS signals, such as indoor spaces or dense urban environments.

Given the strategic importance of UAV localization and the rapid pace of technological innovation in this area, it is timely and relevant to explore the current state of research and development.

2. **State of the art**

Uncrewed aerial vehicles (UAVs) have emerged as a promising solution for time-critical location-based applications such as search and rescue operations, environmental monitoring, and security surveillance [1, 2, 3]. However, designing UAV networks that can perform complex missions while keeping complexity low presents significant challenges. A vital issue is UAV intelligence (UAV-I), defined as its ability to process information and make decisions [4]. Deploying UAV-I across the network can help reduce on-board complexity but also affects autonomy and latency [5].

Recent work has studied various architectures for low-complexity UAV networks. One popular approach involves UAV-to-UAV (U2U) communication, in

which each UAV operates with local information [6, 7]. While this approach enables fast decisions, limited context awareness can reduce estimates' accuracy. An alternative is integrating UAVs with cellular infrastructure for increased awareness through edge/cloud support [8, 9]. However, this reduces autonomy and reliance on stable connectivity.

Over the past decade, Simultaneous Localization and Mapping (SLAM) techniques have enabled significant advances in UAV navigation capabilities. Early implementations of SLAM relied primarily on global navigation satellite system (GNSS) signals that suffer from errors and unavailability in specific environments [17]. However, recent work has demonstrated the feasibility of visual-inertial SLAM using onboard cameras and inertial measurement units to perform location and mapping without GNSS [18, 19].

The integration of multiple sensors has also improved UAV perception abilities. Multimodal sensor fusion addresses concatenated data from cameras, light detection and range estimation (LiDAR) scanners, and other exteroceptive sensors to build dense 3D representations of environments [20, 21]. Techniques such as iterative point clouds match points from different spatial positions to obtain consistent maps [22, 23]. Dempster-Shafer theory and fuzzy logic have been applied to deal with uncertainties arising from sensor noise and imperfect data association [24, 25].

Advances in visual odometry now allow UAVs to track their position incrementally using sequences of camera images [26, 27]. When combined with on-board inertial measurements, visual-inertial odometry provides robust state estimates even during rapid motion [19, 18]. The resulting trajectories can then be merged with maps constructed using structure from motion or simultaneous localization and mapping further to refine localization accuracy [28, 29].

Recent work on the mapping front has focused on semantic mapping, which labels objects and properties in 3D maps [30, 31]. Deep learning approaches use large datasets to train neural networks for tasks such as object detection, segmentation, and classification directly from sensor data [32]. Labelled maps aid scene understanding and enable applications involving object interaction and manipulation.

More recently, research has focused on the use of emerging wireless technologies. Cellular networks show promise for control beyond line-of-sight and long-distance operation. Moreover, with the launch of 5G, beyond 5G (B5G) and 6G networks, cellular localization may see widespread use for SUAV [33]. Simultaneously, artificial intelligence/machine learning (AI/ML) techniques are gaining attention due to their ability to learn environmental properties and adapt to dynamic conditions [33].

3. **Proposed solution and context description**

All these techniques seek to ensure an efficient way of positioning an object in space. In terms of positioning, especially when talking about outdoor locations, the most widely used technique is GPS. There are so many applications where GPS is used, and we could say that any device that requires outdoor use adopts the technology. Emerging technologies that aim to define new methods of outdoor location promise to create redundant solutions that will allow, on the one hand, more accurate locations but also provide a backup for devices where positioning is vital.

In our scenario, the need for a backup for positioning is essential because, in multiple war cases, the GPS signal is jammed, and devices can no longer receive it. Jamming devices perform very well when targeted to specific frequencies, technologies, or protocols. For our proposed system, a backup solution is triangulation using drones.

In a scenario where the GPS signal is affected by jamming, and the drone is used to locate a point by triangulation, several viable methods exist for determining the drone's precise position. Three promising solutions for the future are the relative positioning of the drone within a swarm, position determination by interpretation of the surrounding space and fixed point flight mode.

If we have a group of drones operating cooperatively, they can communicate with each other and exchange information about position and other parameters. Using this data, each drone can calculate its relative position to the others. By using distributed localization algorithms, drones can collaborate to obtain precise positions in a coordinated way.

Drones can be equipped with advanced sensors, such as stereo, lidar, or radar cameras, which allow them to collect information about their surroundings. By analyzing the data obtained, drones can identify landmarks or unique features in the landscape and determine their position about them. Image processing algorithms or machine learning techniques can be used to interpret the data and estimate the drone's position with significant accuracy.

If drones are equipped with advanced radio modules, they can communicate with the device to exchange information. The new generation of radio modules provides parameters that allow accurate distance determination using a single transmitter and receiver by calculating the time it takes for the radio signal to travel between the drone and the device.

4. **Implementation**

Three drones that may also have surveillance or reconnaissance functions can be used to achieve a triangulation system for the devices. We conducted a test scenario using 3 DJI Mavic Pro drones on which we mounted end devices using LoRa communication technology. These were mounted on LoRa using external

phone charging batteries and powered via a USB cable. These were attached to the drone legs.

Determining location using Time of Arrival (ToA) poses several challenges related to device time synchronization. LoRa, as a communication protocol, does not provide a ToA determination mechanism, so to set up such a system, a clock synchronization mechanism must first be implemented. Initially, the first communication tests via the LoRa protocol were using a 8-bit microcontroller. While the microcontroller is powerful enough for actual communication, it lacks the granularity for precise clock synchronization. Thus, it was necessary to switch to a more powerful microcontroller.

To determine the timestamp granularity required to achieve 10-meter accuracy in distance measurements, we can use the distance and time of flight formula:

$$\text{Distance} = \text{ToA} \times v_{\text{signal}} \tag{1}$$

Using radio waves at a speed of about $3 \times 10^8\,\text{m/s}$, we substitute the known values:

$$\Delta t = \frac{10\,\text{m}}{3 \times 10^8\,\text{m/s}} \approx 33\,\text{ns} \tag{2}$$

So, to achieve 10-meter accuracy in distance measurements using radio waves at a speed of about $3 \times 10^8\,\text{m/s}$, a granularity of about 33 nanoseconds would be required during the timestamp. These calculations are based on ideal assumptions and may vary in practice due to factors such as jitter, interference, and other environmental factors.

For the tests conducted, we used a Raspberry Pi, taking into account the errors caused by clock drift. Since we did not have the possibility to use GPS for real-time synchronization, it was necessary to mitigate some of these errors.

To determine the error, we used a DS3231 real-time clock, which according to the datasheet, has an accuracy of 2 ppm when the operating temperature is between 0 and 40 degrees, meaning 0.17 s/day. We synchronized the Raspberry Pi's clock with the DS3231 clock and let it run for 12 days. At the end of the 12 days, we compared the time value read on the Raspberry Pi with the time value on the DS3231. The time difference between the Raspberry Pi and the real-time clock was about 40 seconds. Given that the real-time clock has a deviation over a 12-day period of about 2 seconds, we get a deviation of the Raspberry Pi clock of 42 seconds over 12 days, which is an error of about 40 ppm.

The 40 ppm error can translate to an error of 3.5 seconds/day or 40 µs/s. A signal travels a distance of 10 m, thus:

$$\text{Time} = \frac{\text{Distance}}{v} = \frac{10\,\text{m}}{3 \times 10^8\,\text{m/s}} = 3.34 \times 10^{-8}\,\text{s} = 33\,\text{ns} \tag{3}$$

If we do the calculation at this granularity and relate the error to this unit of time, respectively to this measurement accuracy, we conclude that the error can be ignored.

5. **Experimental results and discussion**

The initial synchronization is crucial, and we achieve this by utilizing the signal from a GPS. This method proves effective, with many of the modules exhibiting an error of less than 1 ppm. Subsequently, we perform clock synchronization using a Python update script. This script initializes the GPS module, waits for the module to be calibrated with the satellites, and then synchronizes the clock.

Determining the Time of Arrival (ToA) is a complex process that involves calculating a series of delays. The time measured from when the time is read from the hardware device to when the departure timestamp is compared to the arrival timestamp is not simply the time the signal travels through the air.

We identify a series of delays that must be considered to accurately determine the time a signal travels through the air: delay due to the sequence of instructions, delay in transmitting the packet via the SPI protocol, delay in processing the packet, time the signal spends in the air.

To determine these delays, we reduced the time the signal spends in the air to 0 by shortening the distance to 20 cm. This means that the measured time reflects only the internal delays of the system.

- T_{seq} - Delay due to the sequence of instructions
- $T_{\text{SPI\_TX}}$ - Delay in transmitting the packet via the SPI protocol at transmission
- $T_{\text{proc\_TX}}$ - Delay in processing the packet at transmission
- $T_{\text{mod\_TX}}$ - Delay in modulating the packet at transmission
- T_{air} - Time the signal spends in the air (which tends to 0 in the test case)
- $T_{\text{demod\_RX}}$ - Delay in demodulating the packet at reception
- $T_{\text{proc\_RX}}$ - Delay in processing the packet at reception
- $T_{\text{SPI\_RX}}$ - Delay in transmitting the packet via the SPI protocol at reception

The formula for calculating the total time, T_{total}, is:

$$T_{\text{total}} = T_{\text{seq}} + T_{\text{SPI\_TX}} + T_{\text{proc\_TX}} + T_{\text{mod\_TX}} + T_{\text{air}} + T_{\text{demod\_RX}} + T_{\text{proc\_RX}} + T_{\text{SPI\_RX}} \tag{4}$$

In the case where the distance is very short (20 cm), T_{air} can be considered negligible. Thus, the formula becomes:

$$T_{\text{total}} \approx T_{\text{seq}} + T_{\text{SPI\_TX}} + T_{\text{proc\_TX}} + T_{\text{mod\_TX}} + T_{\text{demod\_RX}} + T_{\text{proc\_RX}} + T_{\text{SPI\_RX}} \tag{5}$$

Measuring the time between reading the clock from the transmitter and reading the clock from the receiver in this scenario allows us, under real test conditions, to isolate only the time the signal spends in the air. This measurement will be made using an auxiliary device, an oscilloscope, because it has very high precision and can measure signals with very fine granularity. The oscilloscope is crucial in accurately measuring the time the signal spends in the air, as it can

capture and display the signal's waveform, allowing us to precisely determine the time it takes for the signal to travel from the transmitter to the receiver. We will generate an external interrupt when the transmitter's clock is read and another external interrupt when the receiver's clock is read.

The external interrupt is actually an instruction that raises a pin from logical 0 to logical 1. The first pin is on the transmitting device, and the second on the receiving equipment. This way, we will calculate the total time.

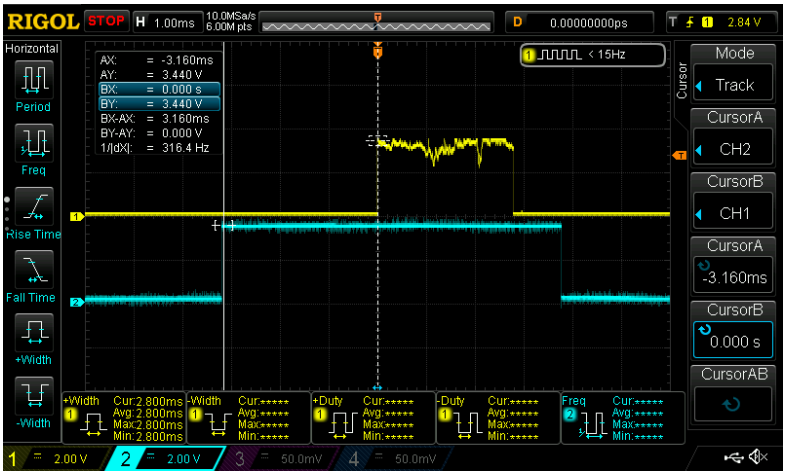

FIGURE 1. Diagram of Delays in Determining Time of Arrival (ToA)

In the Figure 1, it can be seen that the time measured with the oscilloscope is 3.160 ms. We can thus extrapolate the measurements taken at greater distances, considering this as the constant component given by the aforementioned delays. This constant component, represented by the 3.160 ms measurement, is crucial in our calculations as it allows us to isolate the time the signal spends in the air by subtracting it from the total time, thereby providing a more accurate measurement of the ToA.

To improve accuracy and highlight the delays, we conducted a series of additional measurements with utmost thoroughness. We measured the time between two instructions of raising the microcontroller pin from logical 0 to logical 1 and then back to logical 0 as shown in Figure 2. This time allows us to subtract from the total time to isolate the delay between the two clock readings for synchronization.

In our case, this time is 12.90 µs, resulting in the effective time between the two clock readings being:

$$3.160\,\text{ms} - 12.90\,\mu\text{s} = 3.160\,\text{ms} - 0.01290\,\text{ms} = 3.1471\,\text{ms}$$

Thus, the time remaining between the clock readings at transmission and reception is 3.1471 ms.

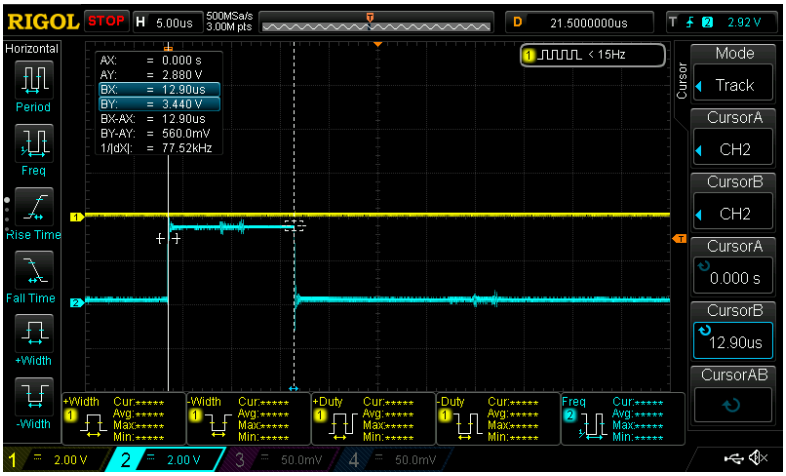

FIGURE 2. Measurement of Instruction Time for Pin State Change

After time synchronization on the 4 Raspberry PIs, 3 being used on the drone and 1 being used on the device, the data packet transmission script was executed. The data packets are composed of an ID, by which the source of the signal was known, the timestamp that allows the Time of Arrival (ToA) calculation, and the coordinates of the drones, which were known at the time of launch.

The 3 devices were mounted on drones, as shown in Figure 3, and raised to a height of 150 meters, with a distance between the drones' launch points and the point where the device is located, as shown in Table 1.

TABLE 1. Drone/device coordinates

Drone	**Latitude**	**Longitude**	**Distance**
First drone	44°59'47"N	26°12'23"E	3778 m
Second drone	44°57'56"N	26°14'03"E	2951 m
Third drone	44°57'57"N	26°16'48"E	1890 m
Device	44°58'56"N	26°14'18"E	

The initial data was erroneous because the system was turned on before the drones were lifted into the air, as the drones and the device were very close together. The received data, as mentioned above, consists of an ID, the timestamp, which we need to calculate distances, and the coordinates for reference. The data for each of the 3 drones are shown in Table 2.

After receiving the timestamp, we calculated the distances from the drone to the point we want to determine. In Table 3, we include the distances calculated using the timestamp and the actual distances according to the coordinates determined in the field.

TABLE 2. Received string data

Nr. crt.	Drone	Received string
1	Drone 1	#01;12:30:23672827335; 44.965833N;26.280011E
2	Drone 2	#02;12:30:23957723884; 44.996431N;26.206430E
3	Drone 3	#03;12:30:24272529325; 44.965634N;26.234218E
4	Drone 1	#01;12:30:27204857344; 44.965839N;26.280002E
5	Drone 2	#02;12:30:27675357285; 44.996434N;26.206418E
6	Drone 3	#03;12:30:28172847225; 44.965634N;26.234223E
7	Drone 1	#01;12:30:29372427542; 44.965842N;26.280020E
8	Drone 2	#02;12:30:29465543693; 44.996419N;26.206445E
9	Drone 3	#03;12:30:30172827142; 44.965634N;26.234229E

TABLE 3. Calculated distance vs accurate distance

Nr. Crt.	Drone	Calculated distance	Real distance
1	Drone 1 - Device	3323 m	3778 m
2	Drone 2 - Device	2961 m	2951 m
3	Drone 3 - Device	1750 m	1890 m
4	Drone 1 - Device	3812 m	3778 m
5	Drone 2 - Device	2563 m	2951 m
6	Drone 3 - Device	1852 m	1890 m
7	Drone 1 - Device	3523 m	3778 m
8	Drone 2 - Device	2438 m	2951 m
9	Drone 3 - Device	1452 m	1890 m

According to Table 3, the distances obtained are varied, and the results are not according to our initial estimates. This is due to the hardware device's inability to retain the time very precisely. The largest distance difference obtained is 513 meters, which means that from our point we can determine an error zone of approximately 500 meters, and defined as a circle with the radius being the difference mentioned above, as shown in Figure 3.

The errors that arise are difficult to mitigate because the challenges we face stem from the quality of the hardware devices used, particularly the Raspberry Pi, which is not a real-time device. According to these results, we need to understand the source of the significant variation in distances. To achieve this, we must comprehend the errors that can affect the transmission.

Our results can suggest different possibilities. If there was a synchronization error between the devices and assuming there were no other errors, such as no delays in the processes executed on the Raspberry Pi and no transmission errors, we can consider the following scenarios:

If there was an advanced desynchronization error, where the drone had an advanced timestamp compared to the device, the measured distances would be

smaller than the actual ones. This is because the transmission time would be reduced by the desynchronization time, indicating that the signal had a shorter Time of Arrival (ToA).

On the other hand, if there was a delayed desynchronization error, where the drone had a delayed timestamp compared to the device, the measured distances would be greater than the actual ones. This is because the transmission time would be increased by the desynchronization time, indicating that the signal had a longer ToA.

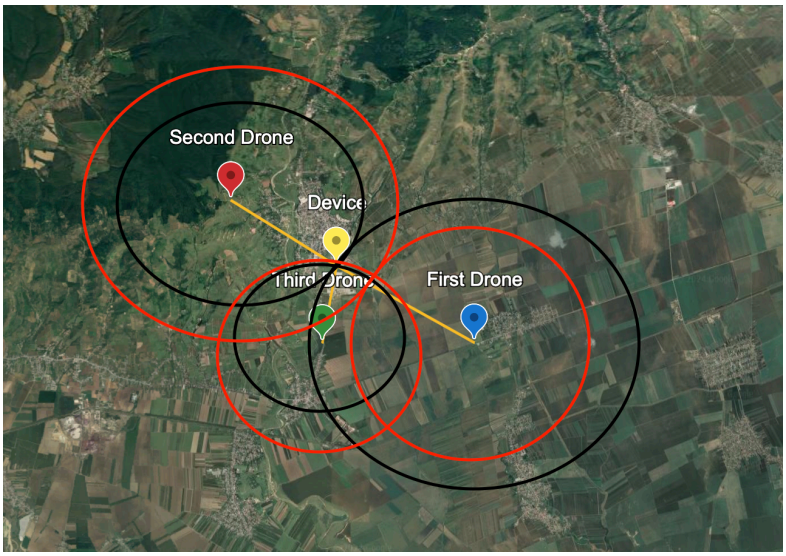

FIGURE 3. Device error area

In Figure 3, only one of the errors is depicted. If we were to represent multiple measurements, we must understand that the point determined by the intersection of the red circles will move within an error zone relative to the point labeled Device.

6. **Conclusions**

Determining distances using Time of Arrival (ToA) is a highly useful technique when GPS signals are unavailable. However, it is crucial to account for additional delays such as device processing time and transmission delays that may arise. Our objective was to determine the location of a subject using drones. Prior to field testing with drones, laboratory tests were conducted to measure these additional delays that can occur beyond the time the signal spends in the air. By isolating these delays, we can achieve greater accuracy in the measured distances. Laboratory tests identified issues related to device desynchronization and potential delays caused by multipath propagation. To address these errors in future developments, we propose a communication protocol that eliminates the need for

device synchronization, thereby limiting errors to processing times and transmission delays due to protocols and other factors. This protocol, once implemented, will enhance the reliability and accuracy of the distance measurements, paving the way for more precise drone localization. This approach will ensure more precise localization by minimizing synchronization issues and accurately accounting for the various processing and transmission delays.

REFERENCES

[1] *A. Guerra et al.*, "Networks of UAVs of Low–Complexity for Time–Critical Localization," arXiv, 2022.

[2] *F. Guidi et al.*, "Distributed inference for collaborative localization in networks of drones," IEEE Sensors Journal, vol. 19, no. 23, pp. 11409-11420, 2019.

[3] *D. Dardari et al.*, "Airborne sensing systems for UAVs," Proceedings of the IEEE, vol. 108, no. 11, pp. 1770-1792, 2020.

[4] *P. M. Djurić*, "UAV intelligence in networks: A tutorial," IEEE Communications Surveys & Tutorials, 2023.

[5] *A. Guerra and D. Dardari*, "When UAVs meet wireless communications: Design and optimization of flying networks," IEEE Proceedings, 2023.

[6] *D. Dardari et al.*, "Airborne communications using drones: Will flying by provide [D2D] gains?"; IEEE Transactions on Vehicular Technology, vol. 66, no. 12, pp. 10927-10938, 2017.

[7] *C. Zhang et al.*, "Cooperative localization for UAV networks using factor graphs," IEEE Internet of Things Journal, vol. 5, no. 4, pp. 2740-2751, 2018.

[8] *A. Al-Hourani et al.*, "UAV-based clouds for communication networks: Potentials and challenges," IEEE Communications Magazine, vol. 57, no. 4, pp. 128-135, 2019.

[9] *Q. Wu et al.*, "Federated learning-based intelligent fog networking for IoT in 6G," IEEE Wireless Communications, vol. 27, no. 5, pp. 120-127, 2020.

[10] *S. Zhang et al.*, "CVM-Net: Cross-View Matching Network for Ground-to-Aerial Geolocalization," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 1568-1577.

[11] *Z. Zhuang et al.*, "MSBA-Net: Multi-Scale Bipartite Graph Attention Network for Cross-View Matching," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11586-11595.

[12] *A. Dutta and J. K. Murthy*, "GeoNet: Geopositioning of Aerial Imagery Using Cross-View Localization," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8364-8373.

[13] *A. Vaswani et al.*, "Attention Is All You Need," Advances in Neural Information Processing Systems, 2017, vol. 30.

[14] *F. Preparata and M. Shamos*, Computational Geometry: An Introduction. New York, NY, USA: Springer-Verlag, 1985.

[15] *J. Wei and A. Yilmaz*, "A Visual Odometry Pipeline for Real-Time UAS Geopositioning," Drones, vol. 7, no. 9, pp. 1-18, 2023.

[16] *P. E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich*, "Superglue: Learning feature matching with graph neural networks," in Proceedings of the IEEE/CVF Conference on

Computer Vision and Pattern Recognition, Seattle, WA, USA, 19–13 June 2020, pp. 4938-4947.

[17] *A. Gupta and X. Fernando*, "Simultaneous Localization and Mapping (SLAM) and Data Fusion in Unmanned Aerial Vehicles: Recent Advances and Challenges," Drones, vol. 6, no. 4, p. 85, 2022.

[18] *J. Delaune et al.*, "VINS-Fusion: A tightly-coupled multi-sensor visual-inertial odometry using point, line and plane features," in IEEE Int. Conf. Intel. Rob. Syst., 2019, pp. 5761-5768.

[19] *T. Qin, P. Li, and S. Shen*, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," IEEE Transactions on Robotics, vol. 34, no. 4, pp. 1004-1020, Aug. 2018.

[20] *F. Endres et al.*, "3-D mapping with an RGB-D camera," IEEE Trans. Robot., vol. 30, no. 1, pp. 177-187, Feb. 2014.

[21] *M. Burri et al.*, "The EuRoC micro aerial vehicle datasets," Int. J. Rob. Res., vol. 35, no. 10, pp. 1157-1163, 2016.

[22] *P. J. Besl and N. D. McKay*, "A method for registration of 3-D shapes," IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 2, pp. 239-256, Feb. 1992.

[23] *H. Yang and L. Carlone*, "A Polynomial-Time Solution for Robust Registration with Intelligent Outlier Rejection," Int. J. Comput. Vis., vol. 128, no. 1, pp. 190-214, Jan. 2020.

[24] *G. Shafer*, A Mathematical Theory of Evidence. Princeton University Press, 1976.

[25] *L. A. Zadeh*, "Fuzzy sets," Inf. Control, vol. 8, no. 3, pp. 338-353, Jun. 1965.

[26] *J. Engel et al.*, "Direct sparse odometry," IEEE Trans. Robot., vol. 34, no. 4, pp. 1611-1626, Aug. 2018.

[27] *C. Forster et al.*, "SVO: Fast semi-direct monocular visual odometry," in IEEE Int. Conf. Robot. Autom., 2014, pp. 15-22.

[28] *M. Bloesch et al.*, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," Int. J. Robot. Res., vol. 36, no. 10, pp. 1053-1072, 2017.

[29] *C. Forster et al.*, "Continuous on-manifold preintegration for real-time visual-inertial odometry," Int. J. Robot. Res., vol. 33, no. 2, pp. 244-264, 2014.

[30] *D. M. Rosen et al.*, "Semantic flying cars: Visually guided flying vehicles for urban environments," in 2018 IEEE/RSJ Int. Conf. Intel. Rob. Syst. (IROS), 2018, pp. 511-518.

[31] *A. Handa et al.*, "gvnn: Neural network library for geometric computer vision," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 8620-8629.

[32] *K. He et al.*, "Mask r-cnn," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961-2969.

[33] *M. Khelifi and I. Butun*, "Swarm Unmanned Aerial Vehicles (SUAVs): A Comprehensive Analysis of Localization, Recent Aspects, and Future Trends," Journal of Sensors, vol. 2022, no. 8600674, Feb. 2022.