

Multipath Mitigation via Clustering for Position Estimation Refinement in Urban Environments

Julian Gutierrez, Russell Gilabert, Evan Dill, *NASA Langley Research Center*
Guillermo Hernandez, David Kaeli, Pau Closas, *Northeastern University, Boston MA*

BIOGRAPHY

Julian Gutierrez is a computer research engineer in the Safety Critical Avionics Systems Branch at NASA Langley Research Center. In 2017, Julian received his MSc in electrical engineering from Northeastern University. Julian is currently pursuing a PhD in computer engineering at the same university focused on high performance computing development within dependable navigation for autonomous aerial vehicles.

Russell Gilabert is a computer research engineer in the Safety Critical Avionics Systems Branch at NASA Langley Research Center. Russell received his MSc in electrical engineering from Ohio University in 2018. His research is currently focused on GNSS augmentation techniques and dependable navigation for autonomous aerial vehicles.

Evan Dill is a research scientist in the Safety Critical Avionics Systems Branch at NASA Langley Research Center. In 2014, Evan received his Ph.D. in electrical engineering from Ohio University. Currently, his research is focused on dependable navigation for autonomous aerial vehicles and the design of assured autonomous systems.

Guillermo Hernandez is a PhD candidate in the Department of Electrical and Computer Engineering at Northeastern University, Boston, MA. His research interests include signal processing, applied cryptography, and GNSS positioning.

David Kaeli is a College of Engineering Distinguished Professor. He received his PhD in Electrical Engineering from Rutgers University. Prior to joining Northeastern in 1993, he spent 12 years at IBM, the last 7 at T.J. Watson Research Center, Yorktown Heights, NY. His research spans a range of areas including microarchitecture and high performance accelerators, to hardware security and back-end compilers. Dr. Kaeli is a Fellow of both the IEEE and ACM.

Pau Closas is an Associate Professor at Northeastern University, Boston, MA. He received his MSc and Ph.D in electrical Engineering from UPC in 2003 and 2009. He also holds a MSc in advanced mathematics from UPC, 2014. His primary areas of interest include statistical signal processing, robust stochastic filtering, and machine learning. He is the recipient of multiple awards including the 2014 EURASIP Best PhD Thesis Award, the 2016 ION Early Achievements Award, 2019 NSF CAREER Award, and 2022 IEEE AESS Harry R. Mimno Award.

ABSTRACT

Position estimation using global navigation satellite systems (GNSS) suffers from poor accuracy within urban canyons due to significant signal disruption caused by tall buildings. This issue can be attributed to the GNSS signals reflecting off buildings resulting in severe multipath reflections which degrade the receiver's performance. In this paper, we introduce an innovative approach to filter GNSS satellite measurements to improve the accuracy of the estimated position by leveraging a clustering algorithm. This approach utilizes a predictive GNSS availability service to filter out non-line-of-sight measurements. Then, a subset of line-of-sight satellite measurement combinations are evaluated using a clustering algorithm. When combined, results show these techniques can reduce the mean horizontal error measured in an urban canyon by nearly an order of magnitude, from ~ 18 meters to ~ 2 meters when using a single point positioning solver.

I. INTRODUCTION

High accuracy positioning will become a necessity to enable future UAS operations (Young et al., 2020). This is especially true for urban settings, where global navigation satellite systems (GNSS) positioning (Morton et al., 2021) will play a pivotal role in optimizing routes for real-time traffic management and ride-sharing services (Williams et al., 2022), as well as powering location-based mobile apps that facilitate finding nearby restaurants or tracking delivery drivers (Zangenehnejad and Gao, 2021). Urban Air Mobility (UAM) envisions a future where aerial vehicles navigate complex urban landscapes to provide efficient transportation solutions (Straubinger et al., 2020). Accurate and reliable position estimation is a critical factor in ensuring safe and efficient UAM operations. However, the errors caused by multipath in dense urban environments can severely impact the accuracy of position estimates for GNSS users (Amin et al., 2016), as well as future airborne vehicles (Xie and Petovello, 2014).

One popular method to improve the reliability of GNSS positioning is Receiver Autonomous Integrity Monitoring (RAIM) (Hewitson and Wang, 2006). RAIM is a system that ensures the integrity of GPS measurements for accurate position, navigation, and timing (PNT) solutions by detecting and mitigating errors caused by erroneous GPS measurements or other anomalies. While RAIM systems play a crucial role in ensuring the integrity of GPS measurements, the effectiveness of RAIM algorithms

decreases in urban canyons due to multiple satellite signals simultaneously experiencing high levels of multipath, and signal obstructions which reduce the redundancy of correct signals (Angrisano et al., 2012).

Systems, such as the Satellite-Based Augmentation Systems (SBAS) (Van Diggelen, 2009) and Ground-Based Augmentation Systems (GBAS) (Pullen et al., 2011), have also been developed to correct for GPS signal errors. Unfortunately, these systems are unsuitable to correct for multipath errors in urban environments. Newer technologies, such as advanced RAIM (ARAIM) algorithms, have also been developed to use multiple GNSS constellations in an effort to further reduce the positioning error, and can be useful in urban environments (Tran and Presti, 2019; El-Mowafy et al., 2022). Using ARAIM may be sufficient to remove signals affected by multipath, as long as the number of affected signals is small. Furthermore, ARAIM paired with 3D models have been shown to be effective at removing NLOS multipath effects (Hsu et al., 2015). Additionally, combining GNSS with other sensor data, such as inertial sensors or Wi-Fi positioning, can enhance accuracy and reliability in urban navigation (Ang, 2018). However, it's important to recognize that these methods aren't universally available or applicable in every situation. Therefore, developing new methods of error mitigation, such as selective satellite usages techniques, as explored in this research, is crucial in realizing reliable Urban Air Mobility operations.

In this work, we propose a novel clustering technique which involves evaluating a subset of combinations of satellites measured by the receiver when determining a position solution, moving beyond the conventional single-satellite assessment employed by RAIM. By concurrently evaluating the position solution from multiple satellite combinations, we can identify single problematic measurements, and detect and address the complexities that arise when significant multipath is present on multiple measurements. This new approach is computationally intensive due to its combinatorial nature, but can now feasibly run in real-time due to the increased compute power available in modern consumer computer systems. This computational power allows us to analyze larger amounts of data using our technique to provide a better position solution in a reasonable amount of time.

Clustering algorithms offer a promising avenue for identifying similarities between position solutions within a set of combinations of satellites. Clustering algorithms are a type of unsupervised machine learning algorithms designed to group similar data points together in order to discover patterns within the data (Xu and Wunsch, 2005). Common clustering methods include K-means, hierarchical clustering, and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Xu and Tian, 2015), each employing different approaches to assign data points to clusters based on predefined criteria. By grouping satellite combinations with similar position solutions, we can identify anomalies that deviate from these established clusters. Outliers, which may represent erroneous satellite measurements or unique error patterns, often stand out as distinctive clusters or isolated data points. By leveraging clustering techniques, we can gain insights into these irregularities and remove erroneous measurements from our final position estimation. This approach could be used to enhance the detection of problematic satellite measurements and contribute to improved accuracy and reliability in navigation and positioning systems.

However, even with the advances in computation, iterating through all possible combinations is computationally impractical. For this reason, being able to choose which combinations to iterate through is important. The state space of possible solutions can be reduced by using NavQ (Dill et al., 2021) to determine which satellite measurements are predicted to be NLOS and remove them prior to running the clustering algorithm. The development of NavQ is an ongoing effort to achieve near real-time GNSS simulations by leveraging high performance computing techniques (Dill et al., 2021; Gutierrez et al., 2022a) to improve GNSS reliability in urban canyons (Gutierrez et al., 2022b; Moore et al., 2023; Gilabert et al., 2023). In this work we use the lower elevation angle LOS satellites to create the combinations of satellites for which we compute position solutions as these satellites typically have a higher probability of having larger multipath error components (Hofmann-Wellenhof et al., 2007). The higher elevation angle satellites remain fixed and are appended to each combination. The position solution for each satellite combination is then computed and the clustering algorithm determines the densest collection of position solutions. Finally the combination of satellites closest to the centroid of the cluster is selected as the algorithm's solution. Further elaboration is provided in the following sections of this paper.

II. OBJECTIVES

This work advances three research objectives:

1. **Enhancing Position Estimation Accuracy in Urban Environments:** The primary objective of this research is to develop and prototype an innovative approach to improve the accuracy of a GNSS position estimation in urban canyons.
2. **Exploring Advanced Satellite Signal Selection:** This research seeks to explore advanced satellite signal selection methods, moving beyond conventional RAIM techniques. The clustering algorithm determines and removes erroneous measurements. It is hypothesized that even when there are many erroneous measurements, there often still exist a subset of measurements that will produce a more accurate solution than using all the available signals measured by the receiver.
3. **Leveraging Machine Learning for Multipath Mitigation:** This research aims to leverage clustering algorithms to mitigate the effects of LOS multipath on positioning accuracy.

III. METHODOLOGY

1. Physical Setup

For this experiment, we employed a SparkFun u-blox F9R breakout GNSS receiver, for its ability to measure L1 and L2 signals from various satellite constellations (including GPS/QZSS, Galileo, GLONASS, and Beidou), generate a GNSS/inertial integrated solution, and provide raw GNSS ephemeris and signal measurements. To complement this receiver, we chose the ArduSimple simpleANT2B multiband GNSS antenna, which is fully compatible with the satellite constellations and frequencies used by the u-blox F9R GNSS receiver. The antenna was affixed to the vehicle's roof using a magnetic mount with the receiver installed inside the vehicle. To facilitate data collection, the receiver was connected to a laptop, which powered the receiver and logged the GNSS data.

2. Location

Data for this experiment was collected around the downtown area of Columbus, Ohio shown in Figure 1. The city of Columbus was a good candidate due to its moderately sized urban canyon. The vehicle was driven through the same route in the downtown area of Columbus, Ohio 8 times over 2 days, for a total of 4 test drives each day, collecting 5445 epoch samples. As part of this analysis, we also focus on a specific segment of the path. This segment of the path, which will be referenced as the *urban canyon segment*, is characterized by the dense urban canyon topography we target in this work. The *urban canyon segment* of the path contained 1883 epochs, out of the 5445 total epochs we collected.

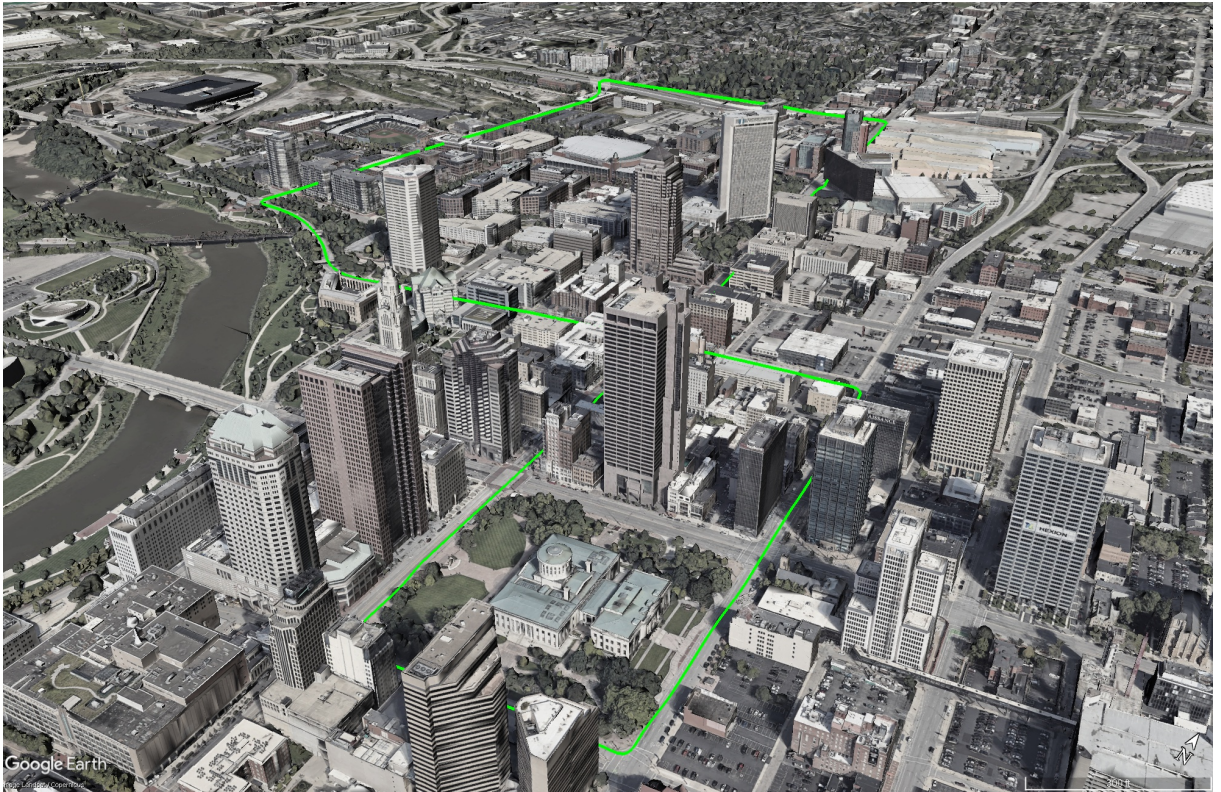


Figure 1: Google Earth view of the location with the ground-truth displayed in green.

3. Ground-truth Reference

The ground-truth reference in this experiment was derived from the fused GNSS/inertial solution generated by the F9R receiver during the data collection process as depicted in Figure 1. Although this integrated solution is anticipated to possess greater accuracy compared to either a standalone GNSS or inertial solution, it remains susceptible to inaccuracies, particularly within areas of higher urban density. Nevertheless, this fused solution suffices for the purpose of comparing the raw and processed GNSS solutions.

IV. CONSENSUS CLUSTERING ALGORITHM

A novel clustering algorithm was developed to enhance GNSS-based position estimation. A block diagram describing this algorithm is shown in Figure 2. The algorithm is divided into three main stages: 1) “NavQ execution”, 2) “Satellite combinations and positions computation”, and 3) “Cluster determination”. This algorithm is designed to use RTKLIB’s single point positioning (SPP) solver. An SPP solver is a navigation technique that estimates the location of a single receiver by using the pseudorange measurements to different GNSS satellites. Additionally, the algorithm does not rely on previously computed solutions to reduce the error (i.e., each solution is independently computed). Using these techniques allows for the treatment of each epoch in isolation to determine the effectiveness of the algorithm.

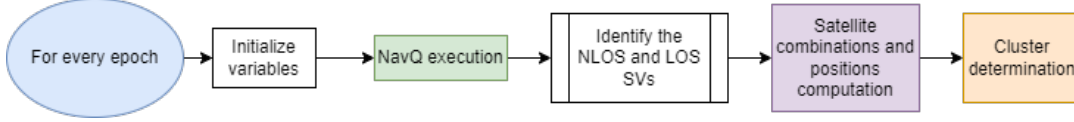


Figure 2: Block diagram of our proposed consensus clustering algorithm.

For each epoch, the NavQ GNSS performance monitor (Dill et al., 2021) is used to predict which satellites are in LOS, and which are NLOS. The algorithm then computes the combinations of the lowest satellites (in terms of elevation angle) and obtains the position solution for each distinct combination of satellites. If the algorithm finds a sufficient number of valid position solutions, the clustering algorithm identifies the densest cluster of position solutions. The combination of satellites that generates the position solution closest to the centroid of the identified cluster is selected as the final set of satellites.

1. Input Parameters and Variables

The proposed algorithm uses two input parameters:

- N_T : represents the target number of satellites used to create combinations of satellites and compute position solutions. The value selected for the results described in this paper is 10, unless stated otherwise.
- $R_{\min}$: ensures a minimum number of valid position solutions are obtained so that the clustering algorithm can work effectively. The value selected for the results described in this paper is 0.1.

Additionally, the algorithm uses the following variables:

- N : indicates the number of satellites being evaluated. For each epoch, this variable is initialized to be equal to N_T .
- B_{NLOS} : boolean variable indicating the use of NLOS satellites. For each epoch, this variable is initialized to *false*.
- V_C : equal to the number of valid combinations of satellites.
- R_V : defined as $V_C/2^N$. R_V represents the ratio of the number of valid combinations of satellites, versus the total combinations of satellites.
- C_{count} : indicates the number of clusters found by the clustering algorithm.

2. NavQ Execution

In this stage, the fused GNSS/inertial solution is used to provide an estimated position to NavQ (Gilabert et al., 2023). This position is processed by NavQ to determine the relative geometry between satellite positions, local buildings and the receiver. The output of NavQ is a list of satellites that are considered in LOS. The measurements observed by the sensor at this epoch are then divided into two sets: 1) the measurements coming from LOS satellites, and 2) the measurements coming from NLOS satellites.

3. Satellite Combinations and Positions Computation

Figure 3 depicts the block diagram of the logic used to produce the results from the “combinations of satellites and position solutions” stage from Figure 2. In this stage, we determine which combinations of satellites (and their corresponding position solutions) are passed on to the clustering algorithm. The algorithm has two priorities when creating the combinations of satellites. Priority 1 uses LOS satellites only. Priority 2 incorporates the use of high elevation angle NLOS satellites when there are not enough LOS satellites to meet the target $R_{\min}$. The algorithm then determines the lowest N -LOS satellites and creates a total of 2^N combinations. Any remaining LOS or NLOS satellites are added to all of the combinations. These combinations are

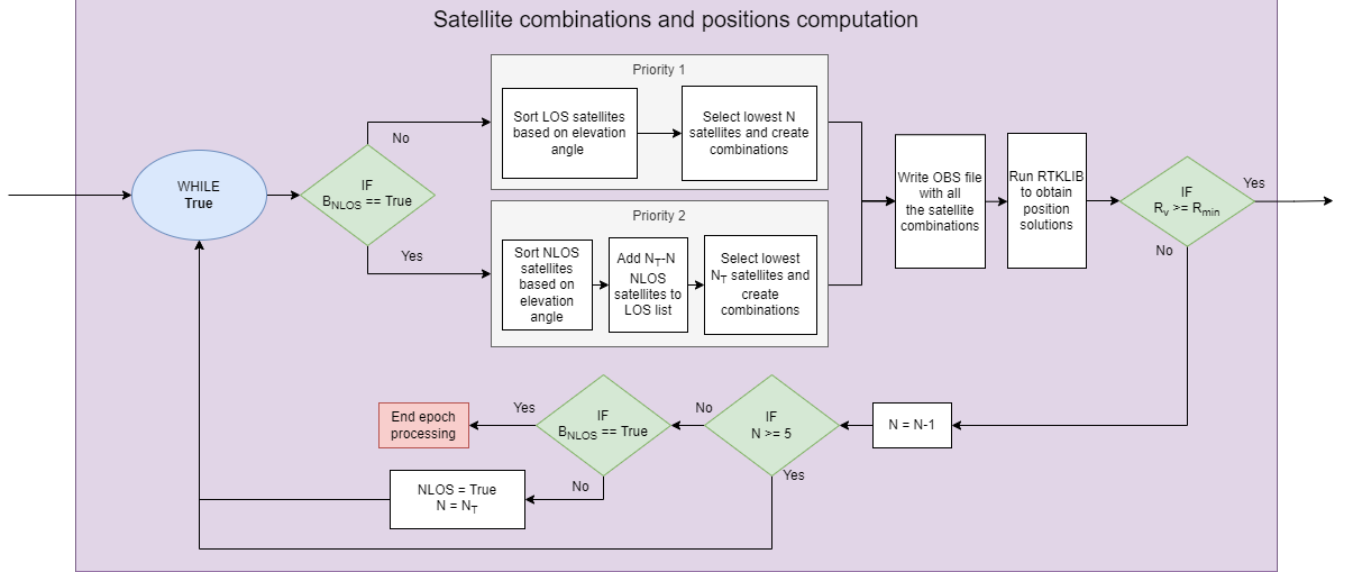


Figure 3: Block diagram of the satellite combinations and positions computation stage of the consensus clustering algorithm.

written to a RINEX file where each combination is treated as a different epoch in RTKLIB¹. This modified version of RTKLIB is used to obtain the position solutions for all the combinations of satellites. Some of these combinations will not generate a valid position solution for one of two reasons: 1) the combination does not have enough satellites to produce a solution, or 2) the solver is unable to find a position solution that meets a predetermined threshold, therefore marking that combination of satellites as invalid.

If $R_v \geq R_{min}$, the algorithm continues to the next block. The condition $R_v < R_{min}$ can happen in situations where there is a low satellite count and the combination of satellites do not produce enough valid position estimates. If this happens, the algorithm decreases the value of N by one and retests the new set containing combinations of satellites. Decreasing the value of N results in a smaller combination count (2^N), which increases the likelihood of obtaining enough position solutions to meet R_{min} , while reducing the need to use NLOS satellites. If the algorithm reaches a value of N lower than 5, it will begin to incorporate NLOS satellites. If B_{NLOS} is false, the algorithm sets B_{NLOS} to true, and resets the value of N to be equal to N_T . Once these changes are made, the algorithm restarts. If B_{NLOS} was already set to true, the epoch process is terminated because there are not enough satellites available to produce solutions for our algorithm to work. This situation was never encountered during this study, but is certainly a possibility in GNSS degraded environments.

4. Cluster Determination

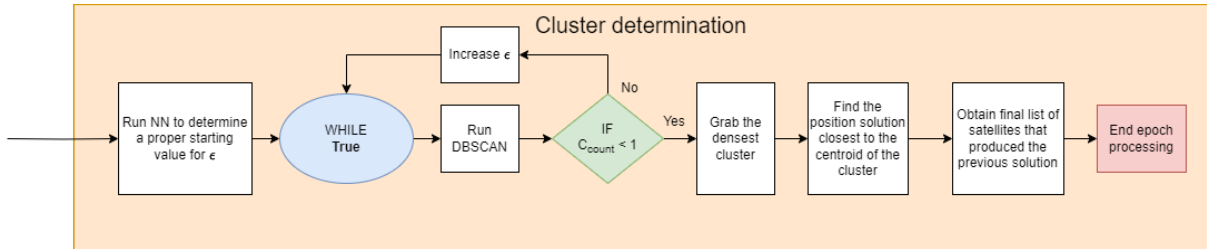


Figure 4: Block diagram of the cluster determination stage of the consensus clustering algorithm.

Figure 4 shows the final stage of the algorithm, where a clustering algorithm is used to identify the densest cluster of the position solutions. In this work, different hierarchical clustering techniques have been tested, as well as the aforementioned DBSCAN method, where DBSCAN was selected based on the observed results. DBSCAN works by grouping together data points that are closely packed and have a sufficient number of neighbors (also referred to as core points), defining clusters based on the density

¹The demo5 version of RTKLIB (Takasu et al., 2007) was modified to process multiple epochs as if they were the same epoch. This modification allows multiple combinations of satellites to be processed within the same execution program, and therefore, improving the execution time of the algorithm.

of the data, while identifying points that are isolated as noise. The two main variables used with DBSCAN are "epsilon" (ϵ), representing the radius within which to search for neighboring points, and "minPts" ($K_{\min}$), the minimum number of points required to form a dense region or cluster (Xu and Tian, 2015). Our algorithm initializes the value of ϵ to the smallest Haversine distance² between the closest position solutions. $K_{\min}$ is defined as: $K_{\min} = R_{\min} \times V_C$.

The algorithm then runs DBSCAN with the previous parameters on the valid position solutions. If a cluster is not found ($C_{\text{count}} < 1$), then the value of ϵ is increased. DBSCAN will rerun this step until the algorithm is able to find at least one cluster. This technique allows for the systematic identification of the densest cluster. The premise of this approach is that the densest cluster will contain position solutions from satellite combinations which have not been affected by multipath. The proximity of the position solutions indicates that any position solution that is not found within the dense cluster is likely the result of a combination of satellites that contained one or more outliers. Finally, the algorithm selects the combination of satellites that resulted in the position solution closest to the centroid of the cluster.

5. Example

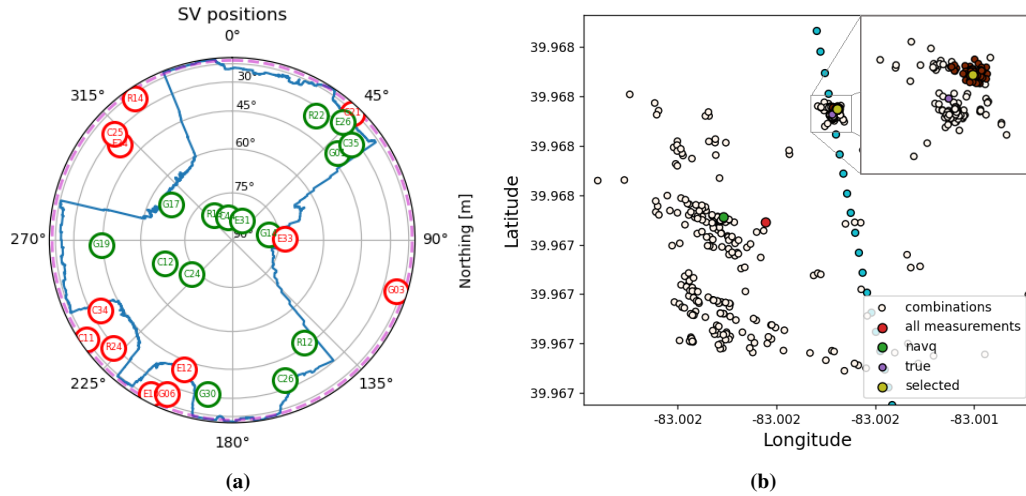


Figure 5: Example epoch with a moderate sky blockage due to buildings to the east and northwest of the position of the vehicle as shown in the sky-plot (a), and (b) the resulting position solutions from the satellite combinations including the selected position solution.

Figure 5 shows an example of how the algorithm works for a datapoint from the collected data. The input parameters to the consensus clustering algorithm were set to the following values: $N = 10$, and $R_{\min} = 0.1$. In this example, the sensor observed measurements from 17 satellites, where 14 of those satellites were predicted to be in LOS by NavQ. The algorithm then created combinations of satellites for the satellites: C12, C24, C26, C35, G01, G17, G19, G30, R12, and R22; while keeping fixed the satellites: C44, E31, G14, and R13. All of the 1024 possible combinations were tested using RTKLIB and only 457 combinations generated valid position solutions.

Figure 5(b) shows the valid position solutions together with the solution produced from the use of all measurements, and the solution produced by NavQ. It is clear there are a large number of position solutions near the true reference position of the vehicle (marked in purple), as seen in the zoomed image in the top right. The consensus clustering algorithm is able to find a dense cluster near the true position, resulting in the selection of a combination of satellites that yields a horizontal positioning error of 1.61 m, as compared to 67.17 m and 74.56 m of horizontal error for the position solutions, using all measurements and NavQ, respectively.

V. EXPERIMENTAL RESULTS

We assess the performance of our proposed algorithm by measuring the horizontal error between the estimated ground truth and the computed position solution. Measurements are collected in meters. Figure 6 shows the ground track of the various position solution methods along with the truth reference. The blue rectangle highlights the *urban canyon segment* of the path. In Figure 6, we can observe considerable variability in the solutions produced using all measurements from the sensor, particularly those in the denser portion of the urban canyon around the taller buildings and narrow corridors. On the other hand, both

²The Haversine distance is the shortest distance between two points on the surface of a sphere. This formula is commonly used in navigation and geospatial applications applied to the spherical shape of the Earth.

NavQ and our novel approach find position solutions closer to the true path; however, specific segments of the trajectory pose challenges for both approaches.

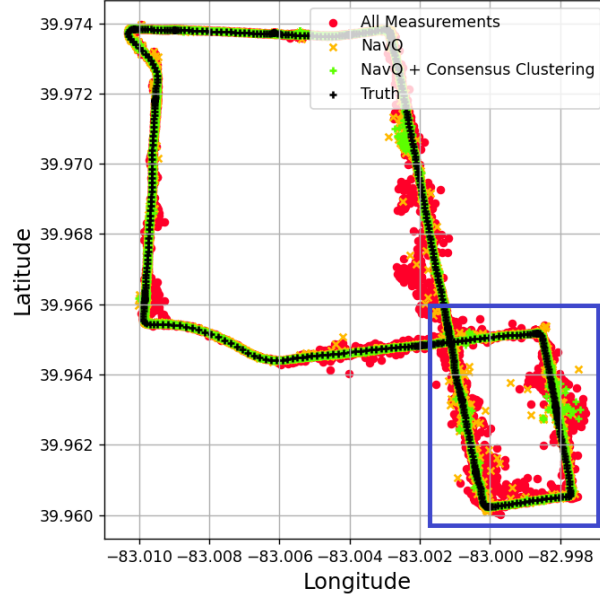


Figure 6: Position solutions obtained with our approach (green), as compared to using only NavQ (yellow) and all measurements (red). The blue rectangle marks the *urban canyon segment* of the full path.

Table 1 summarizes the results derived from using all eight datasets. We additionally processed all measurements and the filtered measurements from NavQ using RTKLIB’s fault detection and exclusion (FDE) system based on RAIM. We will refer to this system as RTKLIB-FDE. In this system, RTKLIB retries the estimation by excluding one by one of the available satellites. After all of the combinations are tested, the estimated receiver position with the lowest normalized squared residuals is selected as the final solution. This system is capable of identifying and excluding an outlier caused by multipath, but this feature will not be as effective with two or more invalid measurements. Those results can also be found in Table 1. These results were collected using $N_T = 10$, allowing 1024 potential combinations of satellites to be analyzed. Our method showed marked improvements across all metrics as compared to alternative methods. We were able to achieve 100% PVT availability, whereas using NavQ alone resulted in 0.64% of the epochs with no available position solution due to an insufficient number of satellites. This effect was also reflected with the RTKLIB-FDE outcome from NavQ.

Table 1: Horizontal error and PVT availability of the position solutions generated by all the raw measurements, NavQ, RTKLIB-FDE, and our consensus clustering algorithm with $N_T = 10$, for all eight datasets of the full path.

Method	Mean error [m]	Maximum error [m]	Standard Deviation [m]	PVT Availability
All measurements	11.81	206.14	16.44	100%
All measurements + RTKLIB-FDE	7.93	147.99	12.9	100%
NavQ	3.23	149.45	6.26	99.36%
NavQ + RTKLIB-FDE	3.71	6163.61	84.66	99.36%
NavQ + Consensus Clustering	2.10	57.97	3.25	100%

Additionally, RTKLIB-FDE improved the overall accuracy of the position solutions for both, all measurements and NavQ. In NavQ + RTKLIB-FDE, RTKLIB-FDE was executed after NavQ had already filtered out many satellites it had identified as NLOS, allowing for RTKLIB-FDE to isolate the remaining problematic measurements. Only 5 epochs resulted in a horizontal error of more than 200 m, and as high as 6 km, as shown in Table 1. These outliers skewed the mean error, although RTKLIB-FDE performed adequately most of the time.

Table 2 provides a similar summary of the outcomes, specifically for the *urban canyon segment* of the path. When utilizing

Table 2: Horizontal error and PVT availability of the position solutions generated by all the raw measurements, NavQ, RTKLIB-FDE, and our consensus clustering algorithm with $N_T = 10$, for all eight subsets of the datasets for the *urban canyon segment* of the path.

Method	Mean error [m]	Maximum error [m]	Standard Deviation [m]	PVT Availability
All measurements	18.52	206.14	18.73	100%
All measurements + RTKLIB-FDE	12.75	147.99	15.37	100%
NavQ	4.49	149.45	8.61	99.57%
NavQ + RTKLIB-FDE	7.03	6163.61	144.07	99.57%
NavQ + Consensus Clustering	2.39	57.97	4.19	100%

all measurements and employing RTKLIB’s SPP solver, the average horizontal error amounted to 18.52 m, with a standard deviation of 18.73 m, and a maximum error of 206.14 m. In contrast, employing NavQ alongside the clustering algorithm, which generates combinations from the 10 lowest elevation angle satellites, resulted in a notably reduced average horizontal error of 2.39 m, with a standard deviation of 4.19 m, and a maximum error of 57.97 m, when using RTKLIB’s SPP solver. The reduced maximum errors provides evidence regarding the efficacy of our approach.

Across the entire dataset, only 17 position solutions provided by our method had a horizontal error exceeding 10 m, as compared to using all measurements, constituting a 0.3% of the overall collected data. Finally, our algorithm is able to handle situations where LOS satellites are insufficient to produce a PVT solution (as shown by NavQ’s PVT availability). In these cases, the consensus clustering algorithm incorporates NLOS satellites to produce enough valid position solutions for DBSCAN to work effectively.

1. Horizontal Error Analysis

We compare the methodologies using an empirical cumulative distribution function (ECDF) graph to capture the horizontal error. The ECDF, a statistical visualization tool, offers a comprehensive representation of the cumulative distribution within a dataset. Figure 7 shows the ECDF’s for both: (a) the complete path dataset, and (b) the *urban canyon segment*. The x-axis delineates the horizontal error, while the y-axis denotes the cumulative distribution. In Figure 7(a), the 95th percentile is highlighted for the five methods, signifying the 95th percentile of the measured horizontal error. For the entire path dataset, the use of all measurements resulted in a 95th percentile error of 45.54 m. In contrast, employing NavQ reduced this error to 11.19 m, and our proposed method further decreased it to 5.00 m, demonstrating a substantial reduction in the overall error versus using NavQ alone. When compared to RTKLIB-FDE, our method was still able to provide a lower error, even when RTKLIB-FDE was paired with NavQ.

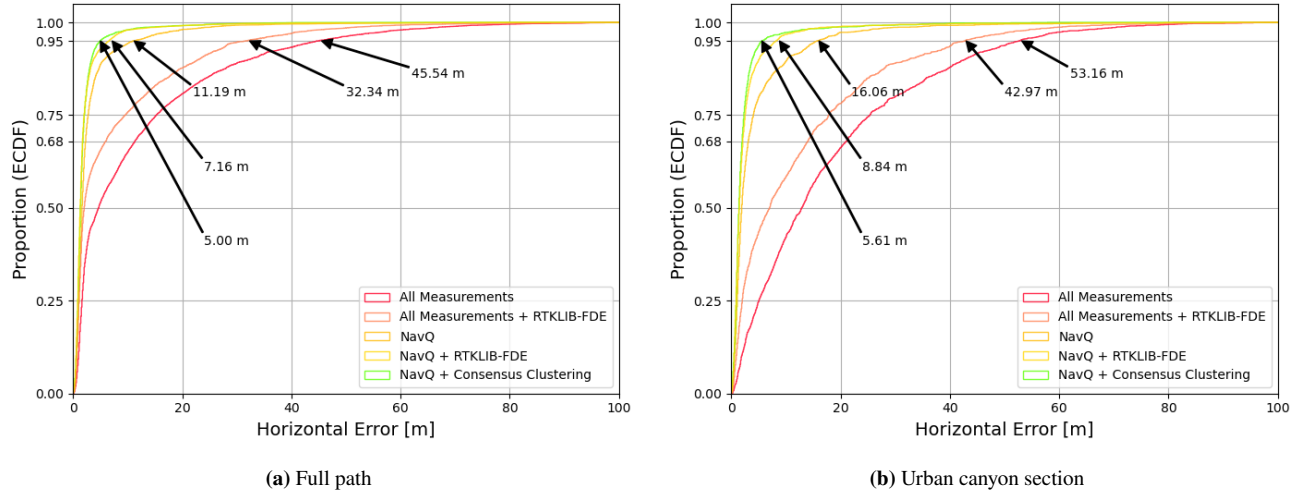


Figure 7: Empirical cumulative distribution function of the horizontal error for different algorithms. (a) results for the full 8 datasets, which include 5445 epochs, and (b) results for the *urban canyon segment* of the path for the 8 datasets, which includes 1883 epochs.

Figure 7(b) illustrates the ECDF for the *urban canyon segment* of the path. Here, the 95th percentile error increases across all methods, emphasizing the challenging nature of urban canyon environments. The horizontal errors observed with all measurements seem to correlate with the dense urban canyon. Notably, our proposed method exhibits a marginal degradation,

from 5.0 m to 5.61 m in the 95th percentile of the horizontal error, suggesting that our method is capable of handling the intricacies of dense urban canyons. We were able to significantly mitigate errors associated with faulty signal measurements. In this segment of the path, RTKLIB-FDE paired with NavQ was also able to reduce the 95th percentile error, although not as significantly as our method. The consistency in performance across varied environments underscores the resilience of our method. The following subsection aims to quantify the scenarios in which our method may exhibit limitations, specifically examining the results based on sky visibility.

2. Sky Visibility Effect

An insightful approach for characterizing the complexities of this problem can be illustrated by organizing the data based on the percentage of sky visibility. This visibility metric was computed using NavQ on the ground truth position and estimating the average sky visibility from the data used to generate sky plots. Figure 8 plots the average horizontal error against the percentage of sky visibility, with data grouped into bins of 5% visibility increments. The error bars represent the maximum observed error within each bin for each method, and the histogram at the top of the plot illustrates the percentage of observations falling within each visibility bin. For the entire path segment as shown in Figure 8(a), epochs predominantly exhibit high visibility, attributable to extended path segments with limited obstacles around the ground truth positions. Conversely, Figure 8(b) shows a reduced sky visibility for the *urban canyon segment*, averaging approximately 75% visibility, with instances as low as 45%. Notably, the maximum horizontal error across multiple bins is significant when using all measurements. NavQ is capable of reducing the maximum error in each bin considerably, with our method exhibiting the most significant improvement. The lower average error at higher visibility aligns with expectations, indicating a favorable trend in terms of reducing errors in scenarios with better visibility.

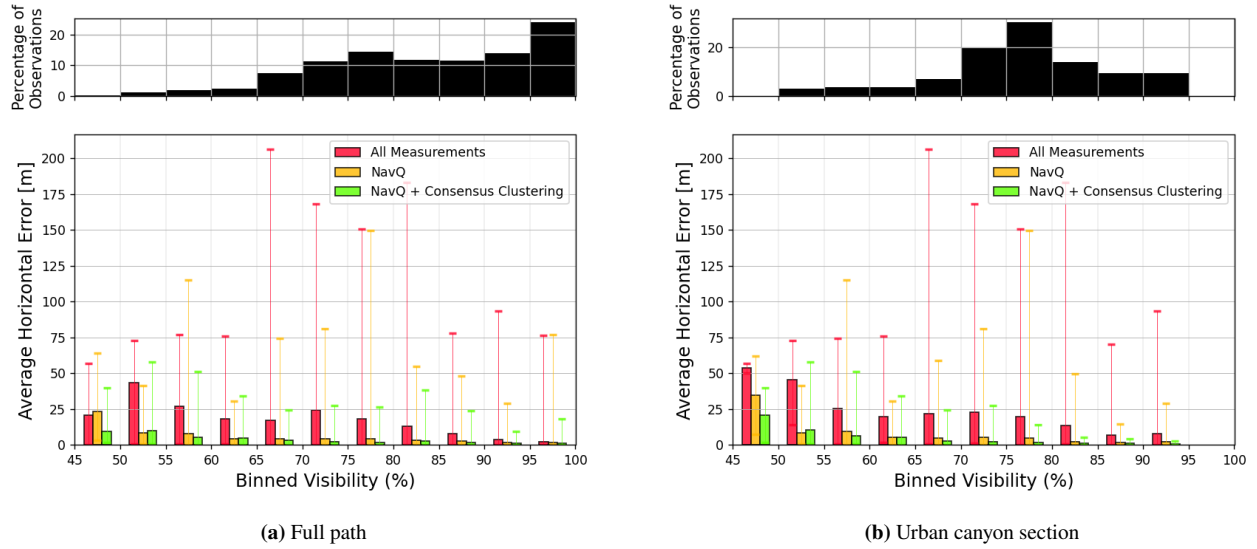


Figure 8: Horizontal error as a function of the visibility of the sky percentage. (a) results for the full 8 datasets, which include 5445 epochs, and (b) results for the *urban canyon segment* of the path for the 8 datasets, which includes 1883 epochs.

a) Low Sky Visibility Analysis

At low sky visibility, we observed two distinct error categories. The first category of errors are false positives arising from trees near the path erroneously obstructing the view of the sky. These false positives are a result of the vehicle traversing close to a tree, prompting NavQ to filter out numerous satellites due to the tree blocking their view of the sky. This error resulted in degraded position solutions for NavQ, and occasionally for the consensus clustering algorithm, due to an increased Dilution of Precision (DOP) from degraded satellite geometry. Figure 9 shows a representative example of this phenomenon.

Figure 9(a) shows the satellites used in NavQ's solution (shown in green) predominantly occupy the third quadrant of the sky. Despite 10 of these satellites being in LOS, RTKLIB was only able to identify 77 valid position solutions from these combinations, culminating in a suboptimal position solution selected by our algorithm, as compared to one utilizing all available measurements. While this issue aligns with the challenges associated with NavQ discussed in prior work (Moore et al., 2023), the occurrence in this specific environment is infrequent, representing less than 0.5% of the observations. It is important to note that, in this context, the magnitude of the error is typically not large. The limited error can be attributed to the fact that the

signals employed are not affected by reflective surfaces; rather, the issue arises from the use of satellites in a concentrated area of the sky, resulting in an unfavorable DOP for the position solution.

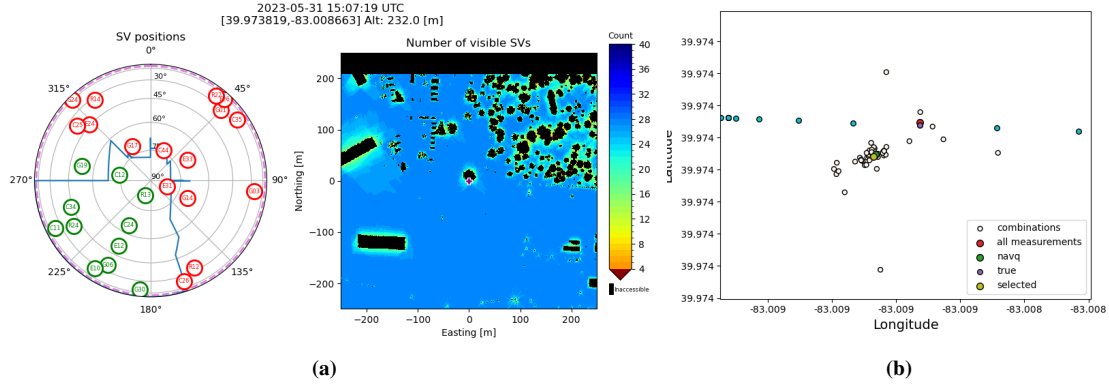


Figure 9: Example epoch of an incorrect assessment due to a tree blockage decreasing the sky visibility. (a) shows the sky-plot observed at the location of the vehicle, and (b) the resulting position solutions from the satellite combinations including the selected position solution by our consensus clustering algorithm.

The second category of errors observed during epochs with low sky visibility is attributed to poor visibility due to tall buildings in densely urban environments. In such areas, the constrained sky visibility limits the number of LOS satellites to an extent where obtaining an accurate position solution using only LOS measurements becomes infeasible, as evidenced by 35 occurrences where NavQ removed NLOS, and as a result, was unable to estimate a position with only LOS satellites. Alternatively, while there might be a sufficient number of valid LOS satellites to compute a single solution, their count may still be too small to generate valid combinations of position solutions, thereby compromising the clustering algorithm's ability to conduct a comprehensive analysis.

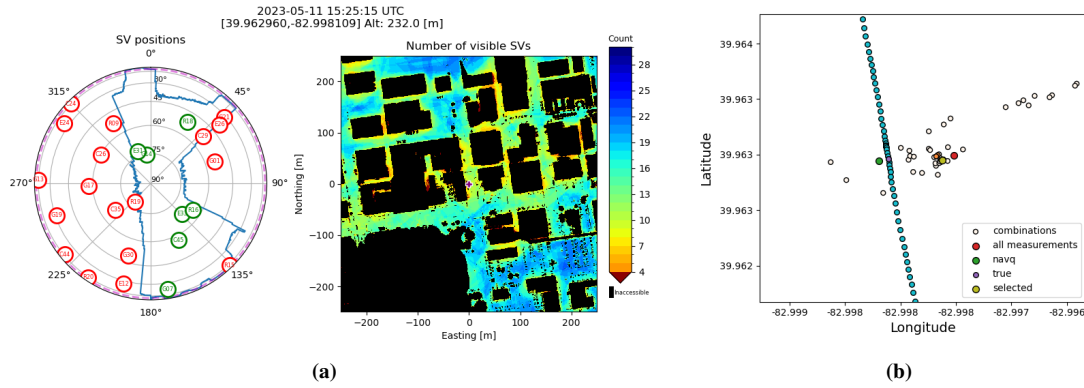


Figure 10: Example epoch of an incorrect assessment by the consensus clustering algorithm due to low sky visibility. (a) shows the sky-plot observed at the location of the vehicle, and (b) the resulting position solutions from the satellite combinations, including the selected position solution generated by our consensus clustering algorithm.

An instance of the second category of errors is shown in Figure 10(a), where only 7 satellites were in LOS, yielding just 1 valid position solution among all possible satellite combinations. Consequently, our algorithm attempts to lower the value of N , a strategy that proves insufficient in this case. Therefore, our approach incorporates NLOS satellites into the satellite sets to bolster the pool of valid position solutions eligible for the clustering algorithm. As illustrated in Figure 10(b), this corrective measure is, again, insufficient to rectify the skew in position solutions, but does yield more valid solutions. Intriguingly, in this specific case, the sole valid position solution derived from using only LOS satellites still turned out to be the best solution. Consequently, our algorithm selects a subset of satellites resulting in a position solution inferior to that of NavQ. Fortunately, such corner cases are infrequent, and in the majority of situations, our algorithm demonstrates superior performance.

b) High Sky Visibility Analysis

Across the majority of high sky visibility epochs, our algorithm demonstrated exceptional performance. A representative example of our algorithm's performance is presented in Figure 11(a). Figure 11(b) shows that our approach correctly identifies

a reliable position solution, which resulted in a minimal horizontal error. In this specific scenario, 14 satellites were in LOS, and the 10 satellites with the lowest elevations contributed to a normalized distribution of position solutions, with our algorithm converging near the truth position. This outcome implies the absence of significant multipath components in the signals received from the LOS satellites.

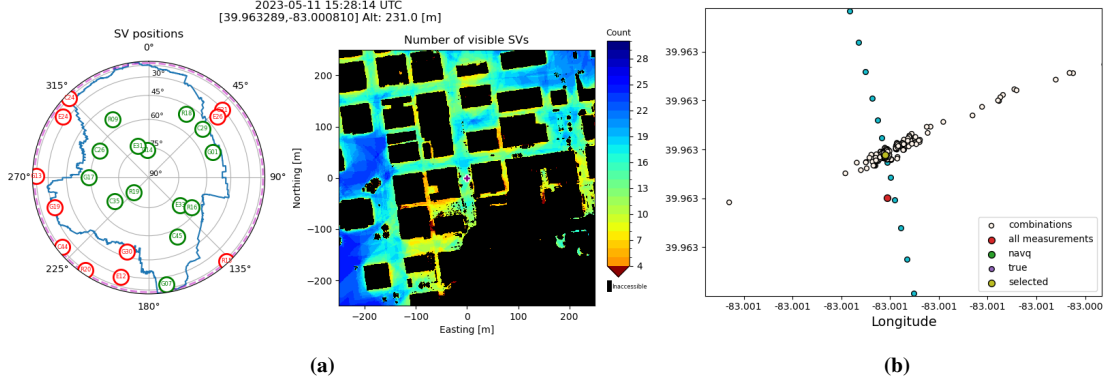


Figure 11: Example epoch of a correct assessment by the consensus clustering algorithm. (a) shows the sky-plot observed at the location of the vehicle, and (b) the resulting position solutions from the satellite combinations, including the selected position solution by our consensus clustering algorithm.

VI. COMPUTATIONAL PERFORMANCE ANALYSIS

We have also conducted an analysis of the algorithm’s execution time. Table 3 shows the mean epoch processing times when targeting different values of N_T , along with the associated horizontal error. Increasing the value of N_T above 10 drastically increases the execution time of the algorithm, making it infeasible for near real-time applications using today’s modern hardware. We do observe an improvement across all metrics when increasing the value of N_T , achieving the lowest mean error of 2.08 m with a maximum error of 55.56 m, with N_T equal to 12.

Table 3: Comparison of the average epoch processing time for the consensus clustering algorithm, for different values for N_T and the average and maximum horizontal error and PVT availability, across all datasets.

N_T	Mean epoch processing time [s]	Mean error [m]	Maximum Error [m]	PVT Availability
6	0.69	2.42	104.99	99.96%
8	1.69	2.15	105.19	100%
10	6.25	2.10	57.97	100%
12	28.86	2.08	55.56	100%

Finally, it should be noted that while the execution time of the algorithm is considered in our results, the primary goal of this research was to improve GNSS positioning accuracy. Additional work could be done to further optimize the algorithm’s performance.

VII. CONCLUSIONS

In this work we explore a unique approach to mitigating the impact of multipath in urban environments. We proposed a new algorithm to enhance position estimation accuracy in the presence of multipath on multiple satellite measurements. By combining an unsupervised machine learning algorithm and the NavQ GNSS performance monitor, our framework refines position estimates and offers a more reliable and accurate navigation and localization system. This technique yielded significant improvements to the horizontal error measured, decreasing the mean error from 11.81 m to 2.10 m when evaluating the full path and 18.52 m down to 2.39 m when evaluating the urban canyon path.

The work described in this paper addresses the critical issue of degraded position estimation performance within urban environments using GNSS. While existing methods, such as ARAIM and various augmentation systems, are generally effective in open sky environments with few erroneous satellite measurements, new solutions need to be developed for the unique challenges that arise when operating in urban environments. This research introduces a novel approach by considering a large set of potential combinations of satellite measurements, leveraging advanced clustering algorithms to address multipath.

While our methods consistently reduced errors effectively, there were areas in our analysis that could be improved. While a reasonable horizontal ground truth was constructed from the fused GNSS/inertial measurement unit (IMU) solution, it will inherently contain a small degree of error, particularly in denser portions of the urban canyon. Additionally, epochs within areas of dense urban canyon are challenging to process with the clustering algorithm due to reduced satellite visibility and poor geometry. In these circumstances, our algorithm is unable to produce an adequate number of valid combinations in order to make an accurate assessment. Our algorithm is able to address one of the key issues discussed by Gilabert et al. (Gilabert et al., 2023), which is when 3DMA removes too many satellites, the receiver is unable to produce a position solution. We addressed this PVT availability concern by adding high elevation angle NLOS satellites in order to produce enough position solutions for the clustering algorithm.

Future work in this area will include the development of a prototype of the framework, capable of real-time execution on a local platform with the sensor. Additionally, this improved position solution could be coupled with an IMU for even better performance. Finally, from the analysis of the multiple position solutions collected for each combination of satellites, we have found that in most cases with adequate GNSS satellite availability, there exists a combination of satellites that yields the lowest horizontal error. Future work will consider a way to identify the satellite combinations that produce this optimal position solution. This new method can involve the use of weighted clustering to give higher priority to position solutions that have specific performance characteristics (e.g., low DOP, low standard deviation of residuals, etc.).

ACKNOWLEDGMENTS

This work has been partially supported by the National Science Foundation under Awards ECCS-1845833 and CCF-2326559.

REFERENCES

- Amin, M. G., Closas, P., Broumandan, A., and Volakis, J. L. (2016). Vulnerabilities, threats, and authentication in satellite-based navigation systems [scanning the issue]. *Proceedings of the IEEE*, 104(6):1169–1173.
- Ang, C.-W. (2018). Vehicle positioning using WIFI fingerprinting in urban environment. In *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pages 652–657. IEEE.
- Angrisano, A., Gaglione, S., and Gioia, C. (2012). RAIM algorithms for aided GNSS in urban scenario. In *2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS)*, pages 1–9. IEEE.
- Dill, E., Gutierrez, J., Young, S., Moore, A., Scholz, A., Bates, E., Schmitt, K., and Doughty, J. (2021). A Predictive GNSS Performance Monitor for Autonomous Air Vehicles in Urban Environments. In *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, pages 125–137.
- El-Mowafy, A., Xu, B., and Hsu, L.-T. (2022). Integrity monitoring using multi-GNSS pseudorange observations in the urban environment combining ARAIM and 3D city models. *Journal of Spatial Science*, 67(1):91–110.
- Gilabert, R., Gutierrez, J., and Dill, E. (2023). Improvements to GNSS Positioning in Challenging Environments by 3DMA Lidar Informed Selective Satellites Usage. In *Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2023)*, pages 2606–2615.
- Gutierrez, J., Kaeli, D., Dill, E. T., and Closas, P. (2022a). Performance and Accuracy Assessment of Line Marching Algorithm Computations Utilizing GPUs Within a Predictive GNSS Quality Service. In *AIAA SCITECH 2022 Forum*, page 0029.
- Gutierrez, J., Neogi, N. A., Kaeli, D., and Dill, E. T. (2022b). A High-Performance Computing GNSS-aware Path Planning Algorithm for Safe Urban Flight Operations. In *AIAA AVIATION 2022 Forum*, page 3461.
- Hewitson, S. and Wang, J. (2006). GNSS receiver autonomous integrity monitoring (RAIM) performance analysis. *Gps Solutions*, 10:155–170.
- Hofmann-Wellenhof, B., Lichtenegger, H., and Wasle, E. (2007). *GNSS—global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media.
- Hsu, L.-T., Gu, Y., and Kamijo, S. (2015). NLOS correction/exclusion for GNSS measurement using RAIM and city building models. *Sensors*, 15(7):17329–17349.
- Moore, A., Gutierrez, J., Dill, E., Logan, M. J., Glover, S., Young, S., and Hoege, N. (2023). Accuracy Assessment of Two GPS Fidelity Prediction Services in Urban Terrain. In *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*.
- Morton, Y. J., van Diggelen, F., Spilker Jr, J. J., Parkinson, B. W., Lo, S., and Gao, G. (2021). *Position, Navigation, and Timing*

- Technologies in the 21st Century: Integrated Satellite Navigation, Sensor Systems, and Civil Applications*. John Wiley & Sons.
- Pullen, S., Walter, T., and Enge, P. (2011). SBAS and GBAS integrity for non-aviation users: moving away from "specific risk". In *Proceedings of the 2011 International Technical Meeting of The Institute of Navigation*, pages 533–545.
- Straubinger, A., Rothfeld, R., Shamiyeh, M., Büchter, K.-D., Kaiser, J., and Plötner, K. O. (2020). An overview of current research and developments in urban air mobility—Setting the scene for UAM introduction. *Journal of Air Transport Management*, 87:101852.
- Takasu, T., Kubo, N., and Yasuda, A. (2007). Development, evaluation and application of RTKLIB: A program library for RTK-GPS. In *GPS/GNSS symposium*, pages 213–218.
- Tran, H. T. and Presti, L. L. (2019). Kalman filter-based ARAIM algorithm for integrity monitoring in urban environment. *ICT Express*, 5(1):65–71.
- Van Diggelen, F. S. T. (2009). *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech house.
- Williams, N., Darian, P. B., Wu, G., Closas, P., and Barth, M. (2022). Impact of positioning uncertainty on connected and automated vehicle applications. *SAE International Journal of Connected and Automated Vehicles*, 6(12-06-02-0010).
- Xie, P. and Petovello, M. G. (2014). Measuring GNSS multipath distributions in urban canyon environments. *IEEE Transactions on Instrumentation and Measurement*, 64(2):366–377.
- Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2:165–193.
- Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678.
- Young, S., Ancel, E., Moore, A., Dill, E., Quach, C., Foster, J., Darafsheh, K., Smalling, K., Vazquez, S., Evans, E., et al. (2020). Architecture and information requirements to assess and predict flight safety risks during highly autonomous urban flight operations. Technical report, NASA Langley Research Center.
- Zangenehnejad, F. and Gao, Y. (2021). GNSS smartphones positioning: Advances, challenges, opportunities, and future perspectives. *Satellite navigation*, 2:1–23.