IS-313: Location Intelligence

CDE3301 Final Design Report

AY 2025/26

Submitted by

Lim Ji Yong, A0282723A

Lin Hong Yi, A0307436E

Anitej Datta, A0284708U

Supervisors

Dr. Yen Shih Cheng

Graham Zhu Hongjiao

Acknowledgements

The team would like to express our sincere gratitude to B&O for their insights on the importance of location information as well as the ideas they had on potential implementations, which inspired us to decide on developing this middleware.

The team would also like to thank our amazing studio masters Dr Yen and Graham for their support and guidance throughout the past semester, Darren for joining us onn this venture and integrating our packages with a graphical user interface, as well as Tse Han for sharing his preliminary results on UWB accuracy.

We would also like to thank our peers from our studio group – Givson, Cheng Yi, Tse Han, Annanya, Kaveri, Siddhi, Tanmay, Rishi, Avaneesh, Teng Wai, Kashfy – for working with us and generously extending help and guidance throughout the semester.

Executive Summary

Bang & Olufsen (B&O), a Danish luxury audio company, envisions future sound systems that respond intuitively to a listener's position—creating seamless, spatially adaptive audio experiences across rooms. Achieving this requires precise, room-scale indoor localisation.

This project develops a UWB (Ultra-Wideband)–based middleware that bridges raw radio-frequency ranging data to a usable, high-resolution spatial grid. The middleware fuses data from multiple distributed anchors to determine a user's live position. Our design follows a three-layer architecture:

1. **Edge layer:** Multiple NXP Type-2BP UWB modules collect Time-of-Flight and Angle-of-Arrival data from an iPhone acting as a UWB transmitter.
2. **Communication layer:** A distributed MQTT Pub-Sub framework allows each anchor to publish its data over Wi-Fi to a central broker, ensuring scalability and resilience across rooms.
3. **Processing layer:** A Pose Graph Optimisation (PGO) algorithm fuses all incoming measurements into a globally consistent position estimate. Outlier rejection and a sliding-window filter reduce noise and reject erroneous readings in real time.

Across our controlled test rig, the middleware consistently improved position accuracy by ~32% over the worst-anchor baseline, while rejecting roughly 10% of noisy data before fusion. The system not only reduced mean error by 3-4x, but also improved reliability. This demonstrates that even imperfect UWB readings can, through sensor fusion, yield stable, high-fidelity location data suitable for responsive multi-room sound experiences.

Table of Contents

1. Introduction

1.1 Problem Situation

Bang & Olufsen (B&O) is a Danish high-end audio solutions provider. They are renowned for creating iconic audio and home entertainment products to the highest standards of sound, craft and design.

In recent years, B&O has been keen to develop more “seamless” user experiences. B&O has been exploring ways to enhance its network of devices with location awareness. The vision is for users to effortlessly manage their devices and unite multiple speakers into a cohesive system for an immersive audio experience.

More specifically, the focus is on adapting to the user's movement to integrate this spatial context with the system, eliminating the need for manual device control. Currently, B&O’s integration methods for their audio collection are the Beolink system [1], [2], and a physical external hub BeoConnect core [3]. However, they still rely on manual control methods of using remote controls, mobile applications, or physically touching the speaker. In addition, they are designed with only B&O devices in mind and are a closed system that other brands cannot integrate with.

1.2 Technical Review

1.2.1 Background Context on Localisation

At present, indoor localisation is most commonly achieved using WiFi and Bluetooth by measuring the Received Signal Strength Indicator (RSSI) from multiple fixed anchors. A method of trilateration then pinpoints a single location in the area. However, Wi-Fi and Bluetooth are accurate to 5 m and 1 m, respectively. There is no inherent directional data, and location fingerprinting (prerecording of signals across the map) is also needed, which makes it sensitive to future environment modifications [4]

GPS is an established method of localisation in outdoor settings, but offers a resolution of over 1-5m at best, and physical obstructions from buildings cause significant errors.

Another possibility is using vision-based solutions, but they fail when users face away or with poor lighting. Location information is also challenging to piece together in multi-room contexts, and camera captures raise privacy issues for home use.

In line with the vision for their devices to gain spatial context awareness, B&O has already embedded the relevant hardware in their production line of speakers. The project therefore leverages this technology, Ultrawide-Band, to be the edge device that obtains location data.

1.2.2 Ultrawide-Band (UWB)

UWB is a radio technology that transmits signals across a wide frequency spectrum (typically 3.1-10.6 GHz), enabling centimetre-level accuracy. Unlike Wi-Fi or Bluetooth, UWB measures time-of-flight (how long a pulse takes to travel between devices) and can also determine the angle-of-arrival, giving both range and direction. This makes UWB robust to multipath effects (signal reflections) in cluttered indoor environments.

1.2.3 Existing companies

Two of the few companies using UWB are Pozyx and Kinexon, which offer UWB-based Real-Time Location Systems (RTLS). Implementations are often limited to asset tracking rather than automations. Their systems are typically point-to-point to a high-power central anchor, and do not offer real-time positional data in a dynamic environment, nor do they provide the scalability and adaptability needed for B&O’s seamless audio experience.

Apple's U1 chip, used in devices like iPhones (11 and later), AirTags, and some HomePods, contain a NXP UWB chip [5]. However, they have only been used for point-to-point measurements (device tracking) rather than a full networked solution. Google Pixel’s UWB applications are similar. These solutions work well for static or small-scale use cases (e.g. locating a lost item, pairing devices), but don't offer the open, flexible, real-time indoor location awareness necessary for coordinated devices to provide “Follow-me” or contextual audio across multiple areas and rooms.

1.2.4 Company-side

B&O possesses a mastery of acoustic design, which allows them to create precise audio “sweet spots” and directional sound zones through their stereo and beam-forming speaker systems. In addition, built-in UWB hardware within many of their existing products can power centimetre-level spatial awareness.

While NXP continues to refine low-level signal acquisition capabilities and B&O focuses on user-facing experiences, the middleware layer remains underdeveloped. This project bridges these two technological advantages by integrating multiple streams of sensor data with real-time computation to form a dynamic spatial grid.

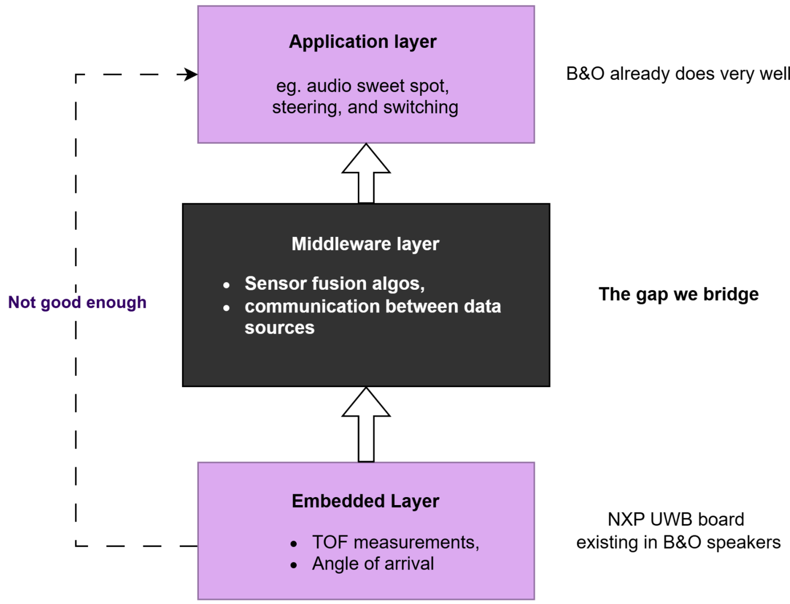


Fig 1: Visual representation of the gap between B&O's capabilities and applications

The black-box middleware potentially exposes a programmable interface, such as an API, which lets other edge devices access live location data. This enables systems, such as B&O's, to trigger spatially-aware automation experiences, like seamless audio room transitions, which BeoLink and BeoConnect Core cannot offer.

2. Design Overview

2.1 B&O's application layer goals

Product designers and engineers at B&O have expressed interest in a few features for future products and updates. Some examples that the project team are targeting are:

1. "Follow-Me" audio sweet-spot
   When the listener moves from the sofa to their yoga mat, the stereo image auto-pans so the listener always stays within the acoustic sweet-spot.

2. Adaptive multi-room hands-free speaker switch
   When the listener crosses the doorway, music crossfades gradually from the living room to the kitchen exactly with no manual taps nor voice commands.

3. Localisation-based Playlist Transitions
   Suppose a listener habitually plays classical music on the couch and rock in the kitchen. This pattern can be learned. After a few days of this routine, when the listener sits on the couch after cooking, the speakers will seamlessly pan towards the couch and automatically change the next queued song to classical music.

These scenarios underscore the recurring requirement of real-time location knowledge of speaker and user positions, at the centimetre level and multi-room scale.

2.2 Technical Requirements and General Performance

The developmental UWB module used is the Type 2BP. Run on NXP's Trimension SR150 UWB chipset, it offers Time of Flight (ToF) and Angle of Arrival (AoA) capabilities to measure the ranging distance and azimuth angles.

The distance accuracy is specified as ±10 cm [6]. The AoA accuracy was reported to be 10 degrees in the azimuth plane [7]. This translates to a translation error of over 60 cm in the azimuth plane.

AoA1		Elevation										
	[deg]	-60	-48	-36	-24	-12	0	12	24	36	48	60
Azimuth	-60	-59.07	-55.91	-54.28	-58.83	-59.44	-60.00	-56.82	-58.59	-59.67	-59.23	-59.61
	-48	-47.65	-48.60	-49.06	-47.75	-47.67	-48.50	-46.38	-48.27	-47.69	-48.81	-46.83
	-36	-36.11	-36.24	-35.30	-36.03	-35.89	-36.20	-35.65	-35.92	-36.25	-35.29	-36.43
	-24	-23.30	-23.62	-23.75	-23.62	-23.36	-24.22	-23.83	-23.30	-23.79	-23.84	-24.62
	-12	-12.96	-11.75	-11.97	-12.20	-11.46	-11.88	-12.14	-11.81	-11.83	-12.67	-12.47
	0	0.05	-0.35	0.23	-0.26	0.56	0.09	0.15	0.01	0.09	-0.12	-0.37
	12	11.58	13.14	12.90	11.51	12.54	11.78	12.04	12.00	11.65	12.21	12.17
	24	23.30	24.80	24.60	23.74	24.10	23.75	23.93	24.42	23.38	24.28	24.70
	36	35.34	34.26	35.30	36.21	35.39	36.09	35.70	37.17	35.86	36.16	36.25
	48	47.22	52.25	48.41	47.02	49.19	49.02	47.22	48.25	48.86	47.59	47.43
	60	59.92	53.25	57.31	59.46	59.50	59.15	58.57	58.58	59.77	59.47	58.11

Fig 2: Type2BP EVK Product Brief showing AoA readings varying with Azimuth [7]

Device performance verification by the team revealed that the single-device error exceeded 120 cm, averaging across 20 UWB readings. “more on this later.”

The proposed middleware aims to achieve a sub-20 cm accuracy in the overall resolution of output measurements. This target accuracy references expected audio sweet spot sizes. Where KEMAR dummy models sit 1.4 m away from the loudspeakers, a head shift by ±3 cm translates to just-noticeable changes in the sound image perceived [8].

We aim to provide location accuracy down to the resolution sufficient to perceive a stable virtual acoustic image, shifting the sweet spot alongside the users. Given the high absolute error provided by the hardware, as well as the fact that listeners in a real room are likely to be further than 1.4 m away, which increases the audio sweet spot size, ±10 cm was set as our target location output accuracy.

2.3 Physical Prototype Overview:

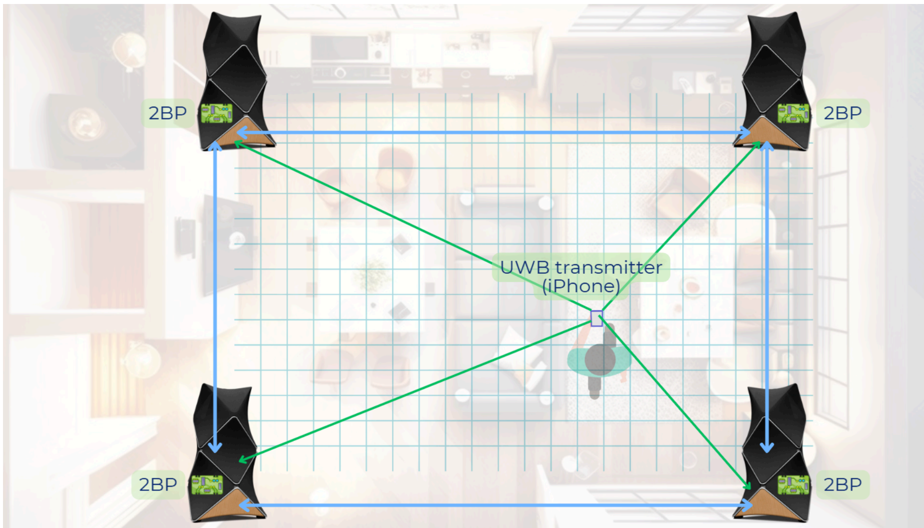


Fig 3: Model of a home with four B&O speakers

The project was built on the presumption of the ideal customer having multiple B&O devices around the home. For simplicity, the rectangular test rig represents a home.

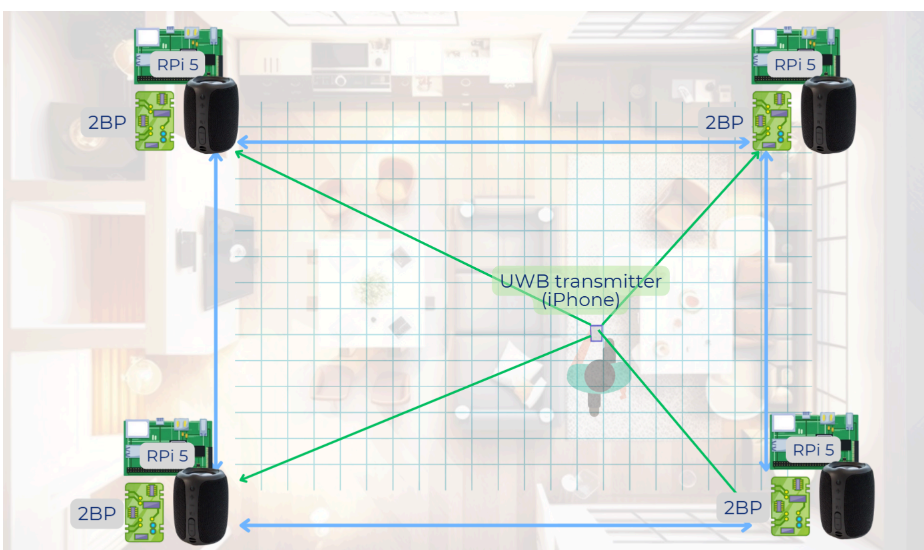


Fig 4: Model of home with speakers - ground truth prototype

Each B&O speaker is prototyped by a cluster of devices consisting of a Raspberry Pi 4, an NXP 2BP UWB module, and a speaker. The phone acts as the UWB transmitter and is a good proxy for the user's position.

2.4 Final design overview:

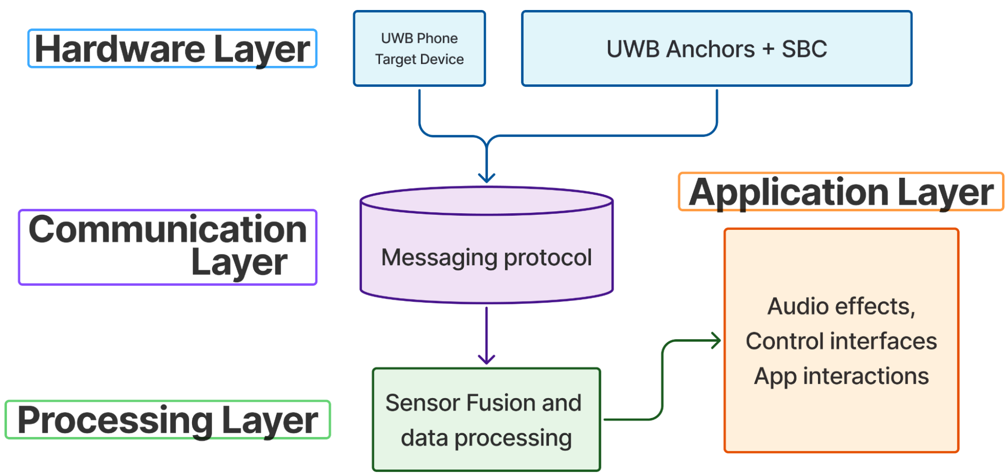


Fig 5: System design (abstracted version)

There are 3 distinct sub-processes. All UWB devices in range collect raw location readings at the hardware edge layer. This rolling data is constantly consolidated and redistributed at the communication layer. The processing layer applies our sensor fusion process that synergises the multiple streams of UWB readings to pinpoint where each UWB node is located. This live information is exposed to an application layer as a shared resource to power automations and beyond.

System requirements for a resilient, scalable, and extensible system architecture:

1. The hardware layer must take in measurements from a variable number of UWB anchors.
2. Processing software must be easily deployable and modular, such that each layer can be integrated with future modifications.

3. Establish a reliable testing process by fixing a physical rig, logging methods and a data collection procedure, to verify performance gains.

3. Development and Testing

3.1 Iterative design changes

Due to the nature of our development, most of which was parallelised, most of the improvement happened in incremental steps. These learnings all accumulated to form the final test rig as well as middle-ware design.

Prototype	Description	Learning points
Pure iPhone ranging	● One iPhone ranges with another iPhone. ● Built a custom app in Swift to display the moving iPhone on the map ● Location data enabled audio-switching on Bluetooth speakers	● iPhone can function as the UWB transmitter in 360° ● iPhone as a receiver is subject to a limited conical range. This results in huge variance when a receiver iPhone is rotated. ● Audio switching on Bluetooth commands has significant latency
Phone to 2BP	● iPhone is used as the transmitter, and the 2BP is the receiver. ● Reading from laptop UART, azimuth and elevation readings of the iPhone can be read from an RPi.	● The 2BP binary codes can exclusively pair with an iPhone or another 2BP board. This number of connections is insufficient for our middleware to work.

Tried embedded to enable multiple 2bp-iphone	● Attempted to rewrite 2BP firmware to enable simultaneous ranging with other 2BPs and iPhones, so as to perform edge-to-edge detection	● Resource conflicts prevent simultaneous connections with multiple 2BPs from being formed ● For MVP, shift focus to using multiple boards and sensors instead of splitting resources on each board
Phone to multiple 2bp	● iPhone range with multiple 2BP boards. We only take readings from the 2BP boards ● Each 2BP board is linked to an RPi to read and publish data to the middleware on the same network for processing.	● Parallel ranging is possible
Prelim test rig	● 2BP boards were ball-jointed mounts in the ceiling ● iPhone is moved around by hand	● Wobbly board mounting caused significant variance in the translational vectors, which then affects the location data ● Multipath (signal reflections) caused high jitter since the boards were mounted vertically

Test rig ceiling mounting, 45deg, phone on floor	● Ceiling-mounted UWB boards were securely fixed with a 45° downwards pitch ● Data collected at 4 fixed coordinates with no translational equivalents (see below)	● iPhone transmitter on the ground gives the most stable readings ● Metal objects and people walking around the UWB transmitter induce higher variance in readings and are visible even though our middleware.

Table 1: System design iteration milestones

3.2 Test rig

To verify our system design iterations, the team set up a fixed test rig in EA-04-04 i-Lounge. This enables us to ensure repeatability and eliminate confounding variables.

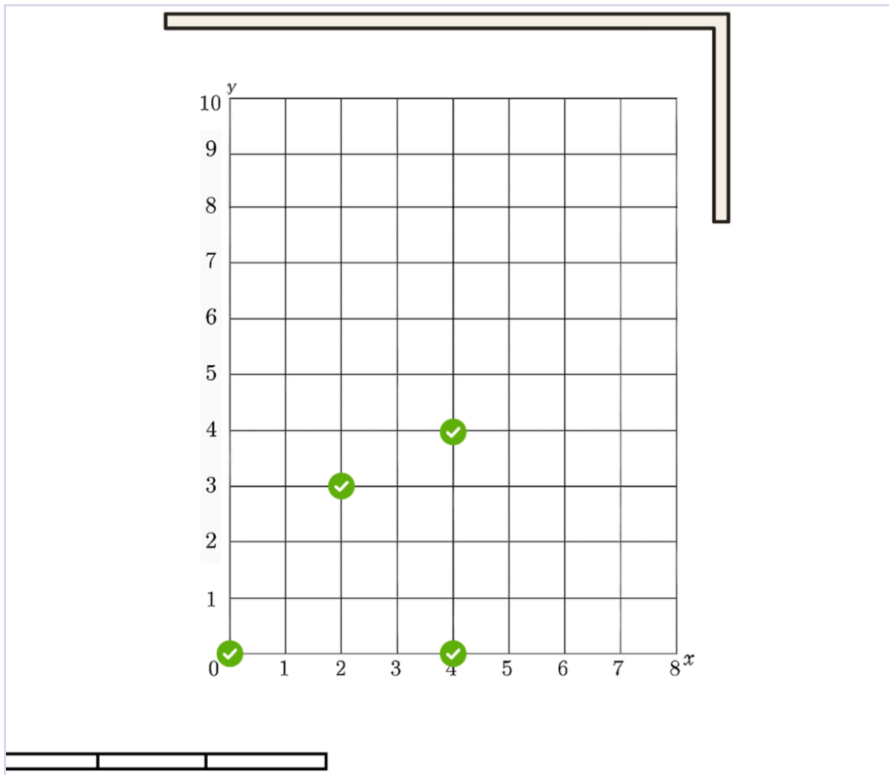


Fig 6: Test rig system and coordinates for data collection

The rig above was set up permanently to ensure that all hardware would not change throughout the weeks of development.

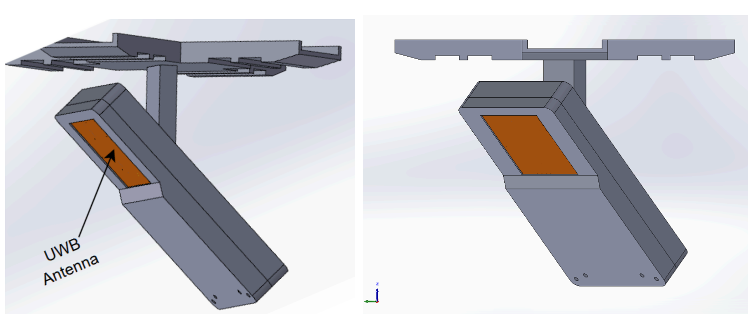


Fig 7: Fixed 45° UWB module ceiling mounts

The main results that B&O was keen to see were:

1. How data changed with movements of the user throughout the area (2BP characterisation with distance and angle changes) (static positions).
2. How data changed with rotations of the UWB device tracking the user (iPhone)

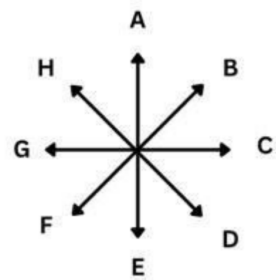


Fig 8: Directions of the phone

Arrows correspond to the direction of the back face of the phone, with an additional rotation titled U being the phone being back camera facing up at the ceiling.

3. How data varies with every addition of a UWB anchor to the system
4. How data output from the middleware compares with the raw coordinates.

B&O expressed that the quality of data is more important than the quantity of locations collected. Given team resource constraints, the team leveraged rotation symmetry and collected data from only one quadrant of the grid, as marked in Fig 6.

4. Final System design

This section presents the finalised system design. Beginning with an outline of the communication framework within the distributed system, followed by a detailed discussion of the Data Processing Layer. During which, the characteristics of the input data are examined to explain how they inform the design choices. The section concludes with an overview of the application layer.

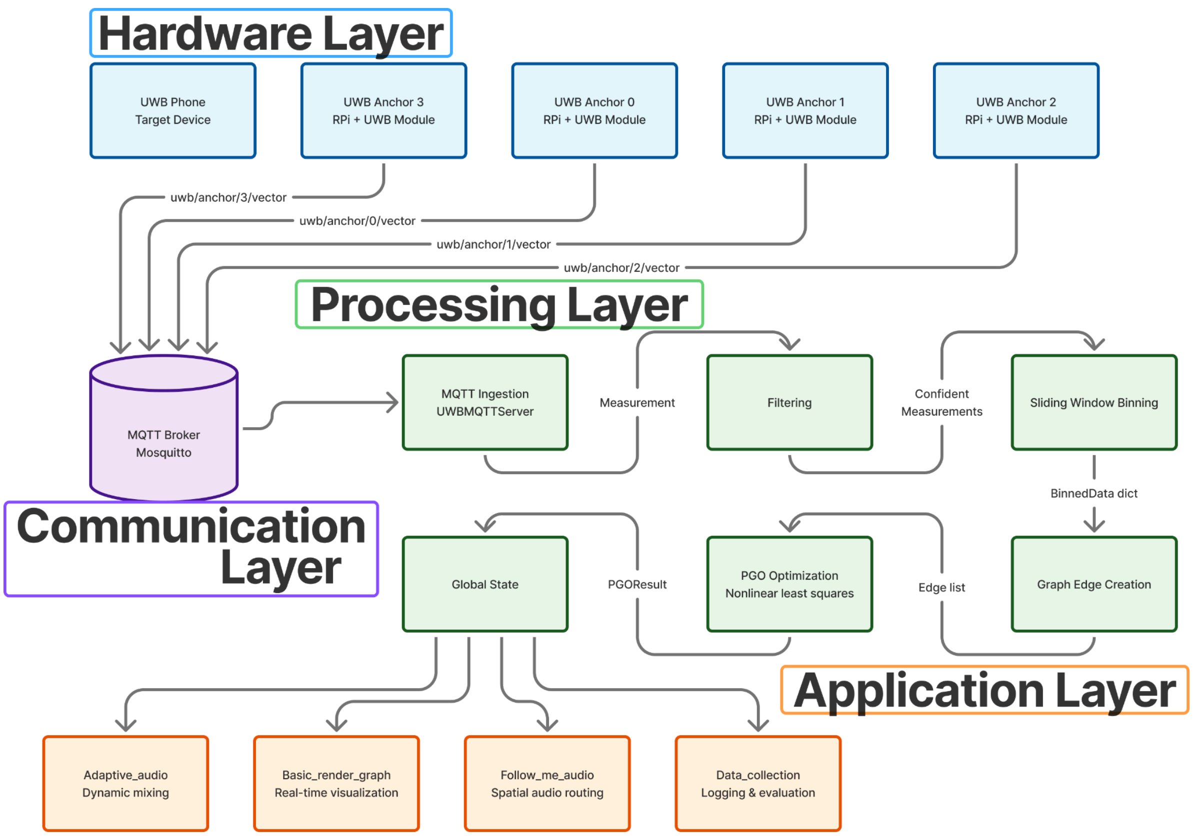


Fig. 9: Final system design (full version)

4.1 Communication

Given that a real system will have a dynamic number of anchors in physical space, reliable and flexible connectivity to distribute the data produced is required. The application is to bootstrap off the speakers' WiFi connectivity, guaranteeing that all speakers in the network can be mapped on the global home grid even without direct LOS. In multi-room setups, for instance, Bluetooth connections cannot span.

MQTT was chosen as the communication protocol for the system due to its lightweight Pub-Sub architecture, which provided the flexibility to easily add and edit nodes in the distributed system.

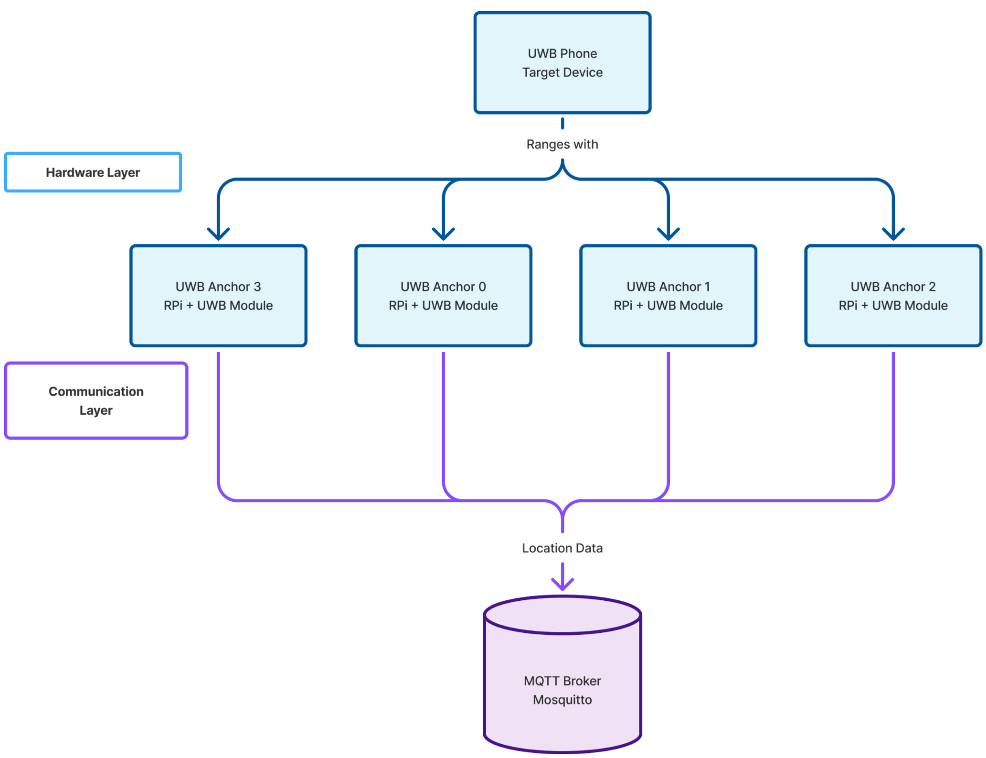


Fig. 10: Communication overview

The Mosquitto MQTT broker is run on a centralised computer, for instance, a laptop, which could represent B&O's Beolink that serves as their communication and compute hub. Similarly, all the timing controls related to the time-series location data are also processed at this central hub.

4.2 Data Processing layer

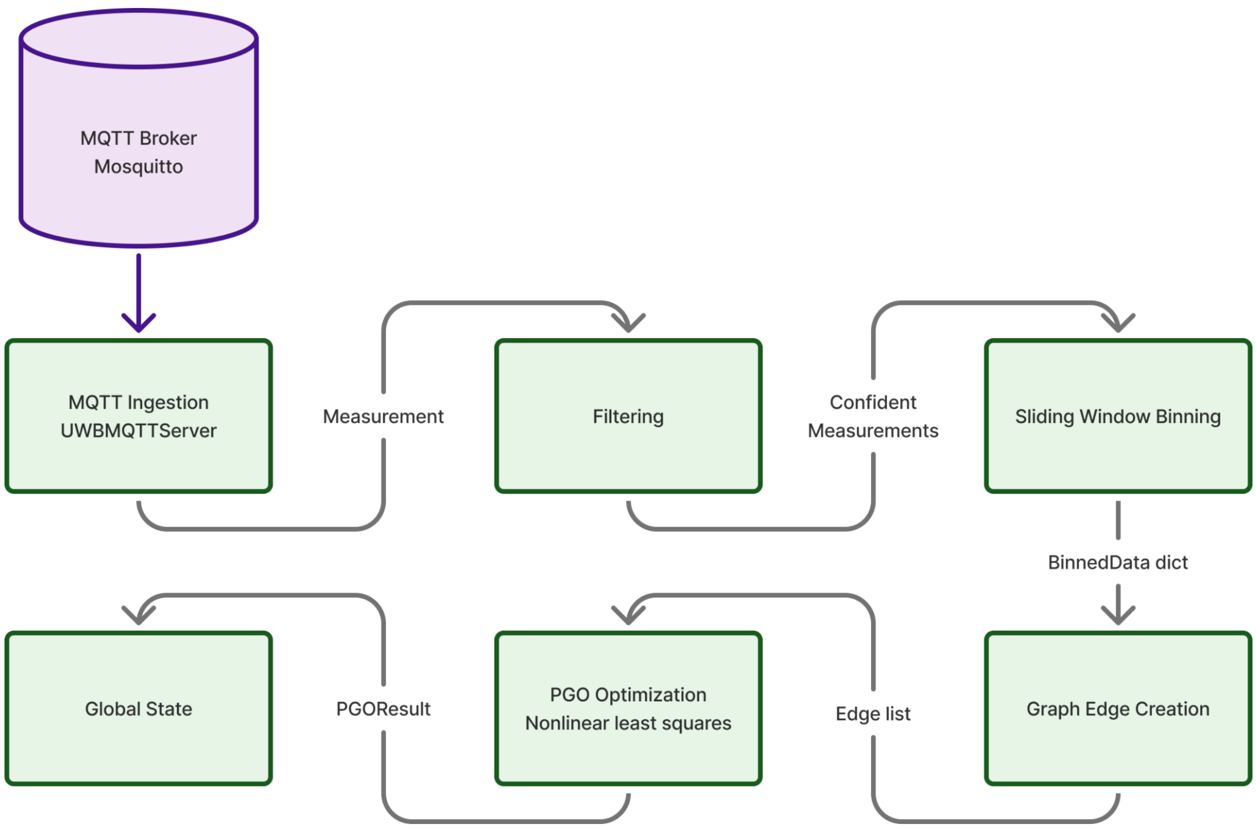


Fig. 11: Data processing flow

The data processing represents the bulk of our novel approach. It takes all the data from our MQTT service and turns it into something that can be handed off to the application layer.

4.2.1 Handling the data input

The output of each of the evaluation boards is essentially a vector signifying the direction to the transmitter (in this case, an iPhone). This vector is relative to the UWB antenna on the board.

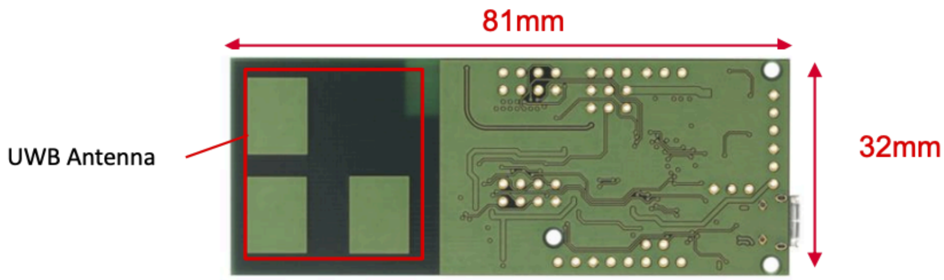


Fig. 12: Board layout

This introduces to us the first problem, which is a local vector relative to the board. Functionally, we have to work in global coordinates. Hence, we apply a rotation based on the actual placement in the test rig.

$$R_{\text{anchor}} = R_z(\text{yaw}) \times R_y(+45^\circ)$$

Benchmarking this, we made a critical observation that it is often far from the ground truth.

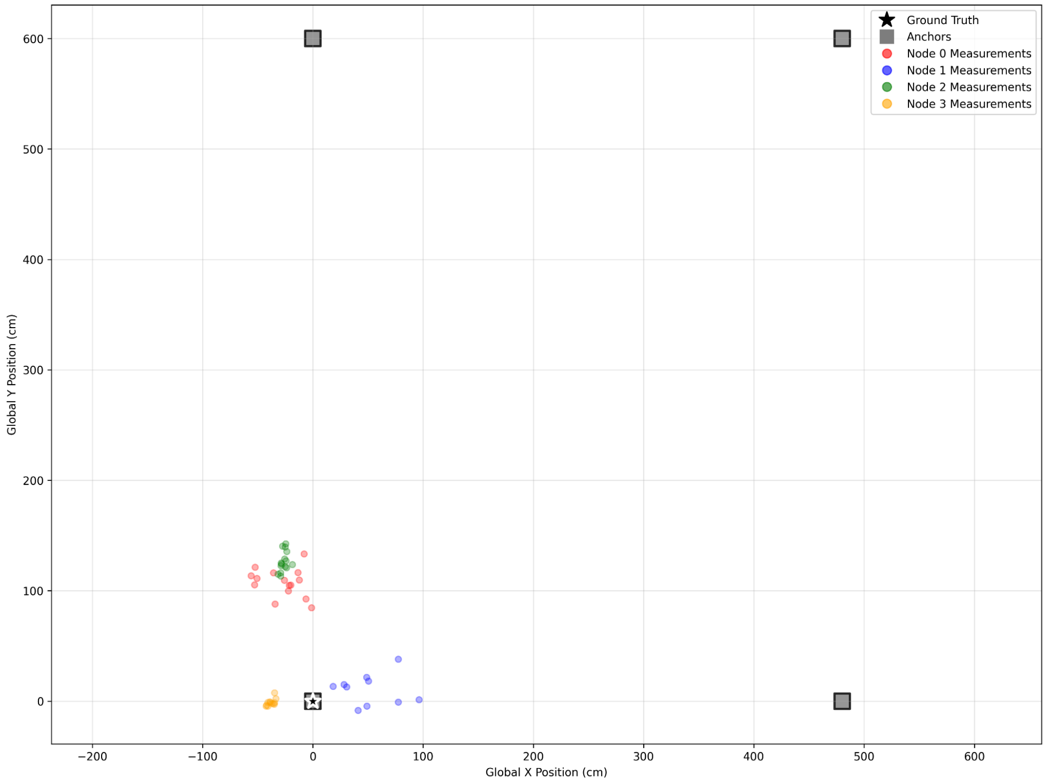


Fig.13 : Raw measurements from UWB Board

There is noticeable spatial dispersion and noise. Even in segments where the data points appear relatively well-clustered, a non-negligible deviation from the ground truth can persist.

4.2.2 The Algorithm (Pose Graph Optimisation)

We can view the problem as an optimisation problem on a graph, with anchors as nodes and measurements as edges. We can minimise and distribute all the errors by minimising the error functions.

Since we expect each edge to be slightly inaccurate, the vector sum of

$$B\text{-}\vec{\text{Transmitter}} + \vec{\text{Transmitter}}\text{-A} \neq \vec{BA}$$

We can then add errors in red and green to ensure that the sum does add up. This is called forming a loop closure; the minimising of errors across the entire graph allows us to merge information from the different anchors without internal conflicts.

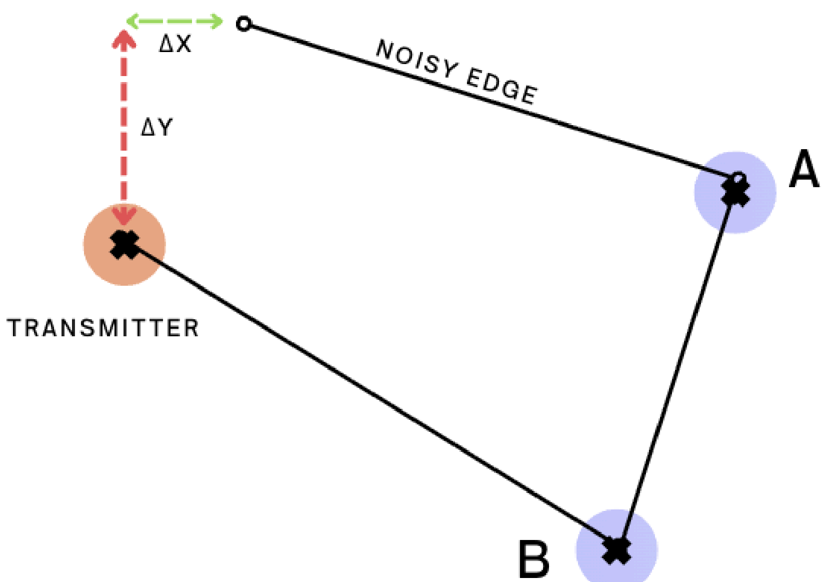


Fig. 14 : Visual representation of Pose Graph Optimisation forming a loop closure

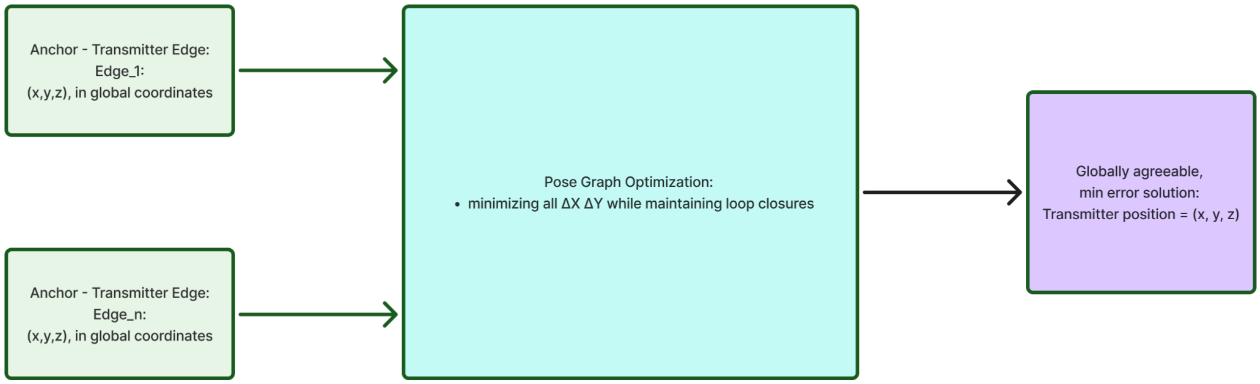


Fig. 15: PGO input and output

A naive approach would be to take the latest data point from each node to pass into the PGO algorithm, essentially rerunning PGO whenever new data arrives.

However, this is rather sensitive to the noise differences between sequential measurements. It is also rather hard to orchestrate as the boards don't all get readings at the same time.

4.2.3 Optimising PGO inputs

At this point, the performance of the PGO system is mostly dependent on the quality of the input edges. So our next goal is to ensure the best quality input edges.

1. Although each measurement has some noise, it averages out, there is also no need to do PGO for every individual datapoint update (5hz), we can update datapoints in larger steps. This led us to adopt a sliding window of 2s. (Appendix A)

2. There exist points which are “very wrong” which sometimes skew the sliding window average as well. So we implemented additional checks to reject likely erroneous readings. (Logic flow in Appendix B)

3. At some positions, one anchor might just perform unusually poorly. When the variance of an anchor is too high, we can disable it from inputting into PGO.

After implementing this filter, we rejected 10.05% of the data on the basis that they are statistical outliers.

4.3 Application layer

The application layer must be able to interface with our packages easily, we have built the PGO and supporting packages to be side-effect free and simple to use. It only takes a few lines of code to bring up our system after importing the packages.

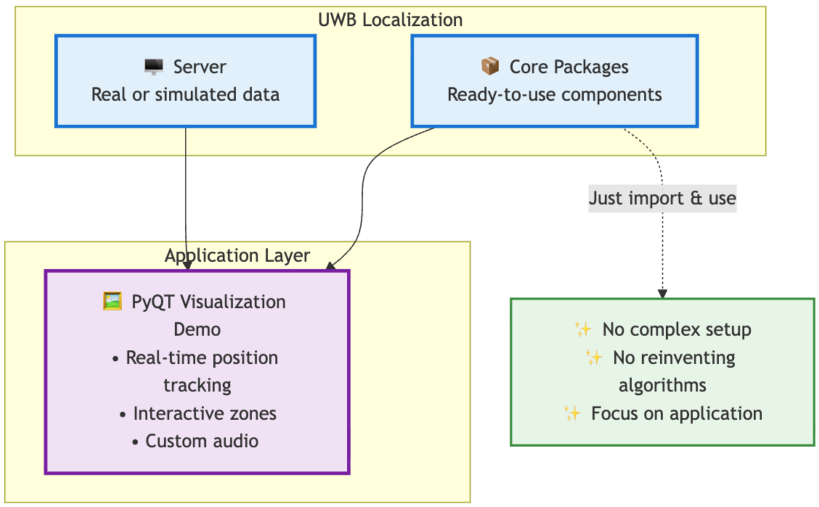


Fig. 16: Package usage

```
# Configure MQTT with your laptop's IP
mqtt_config = MQTTConfig(
    broker="192.168.68.66",  # Replace with your laptop's IP, or tbh it doesnt matter
cuz we add the laptop ip as an argument when calling the file
    port=1884
)

# Start server
server = ServerBringUp(
    mqtt_config=mqtt_config,
    window_size_seconds=1.0
)

try:
    server.start()
    print("Server started. Waiting for anchor connections...")
    while True:
        if server.user_position is not None:
            print(f"Current position: {server.user_position}")
        time.sleep(1)
except KeyboardInterrupt:
    server.stop()
```

Fig. 17: Running of package code

5. Final Evaluation

In this section, we will break down our results of our final system against the baseline of reading raw data from 1 UWB board. The focus here is on the performance of the system across a few different metrics:

1. Different positions in the grid (corresponding to different local displacements from each UWB board)
2. Different orientations of the phone (Fig 8, Page 19)

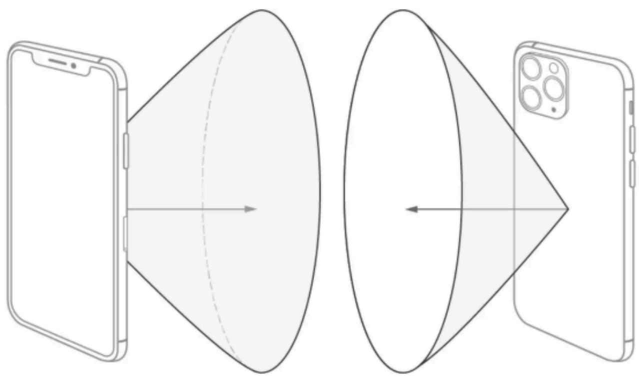


Fig. 18 Optimal iPhone UWB receiver spatial range [9]

3. Increasing amount of anchors in use for ranging

5.1 Data Handling

The raw data collected for the evaluations look like the following

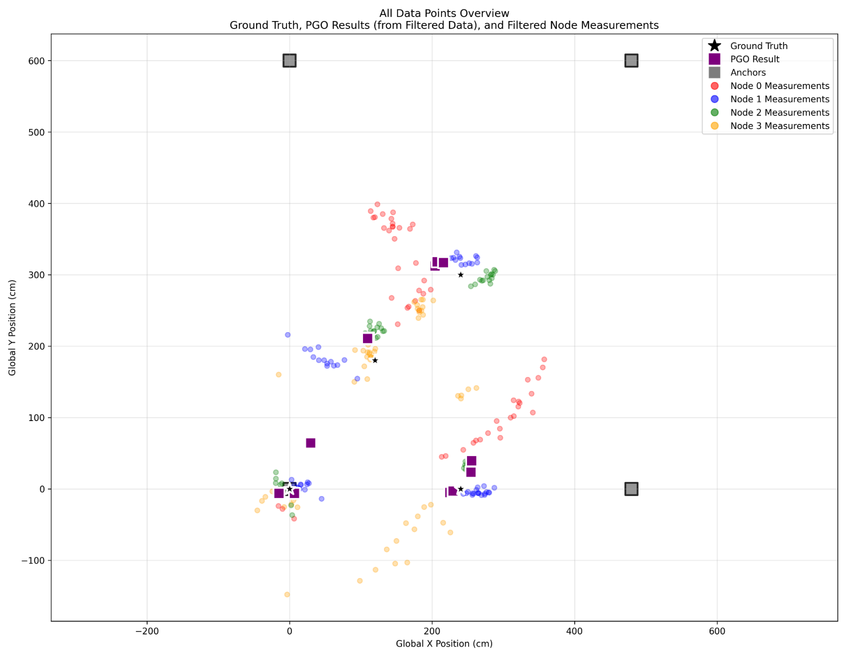


Fig 19. Raw Data collection

Another notable characteristic is that due to the different local displacements from the transmitter, the precision and innate errors affecting each anchor are different.

5.2 Overall metrics

This shows the overall impact of the system, with more anchors in the distributed system increasing the accuracy across the board.

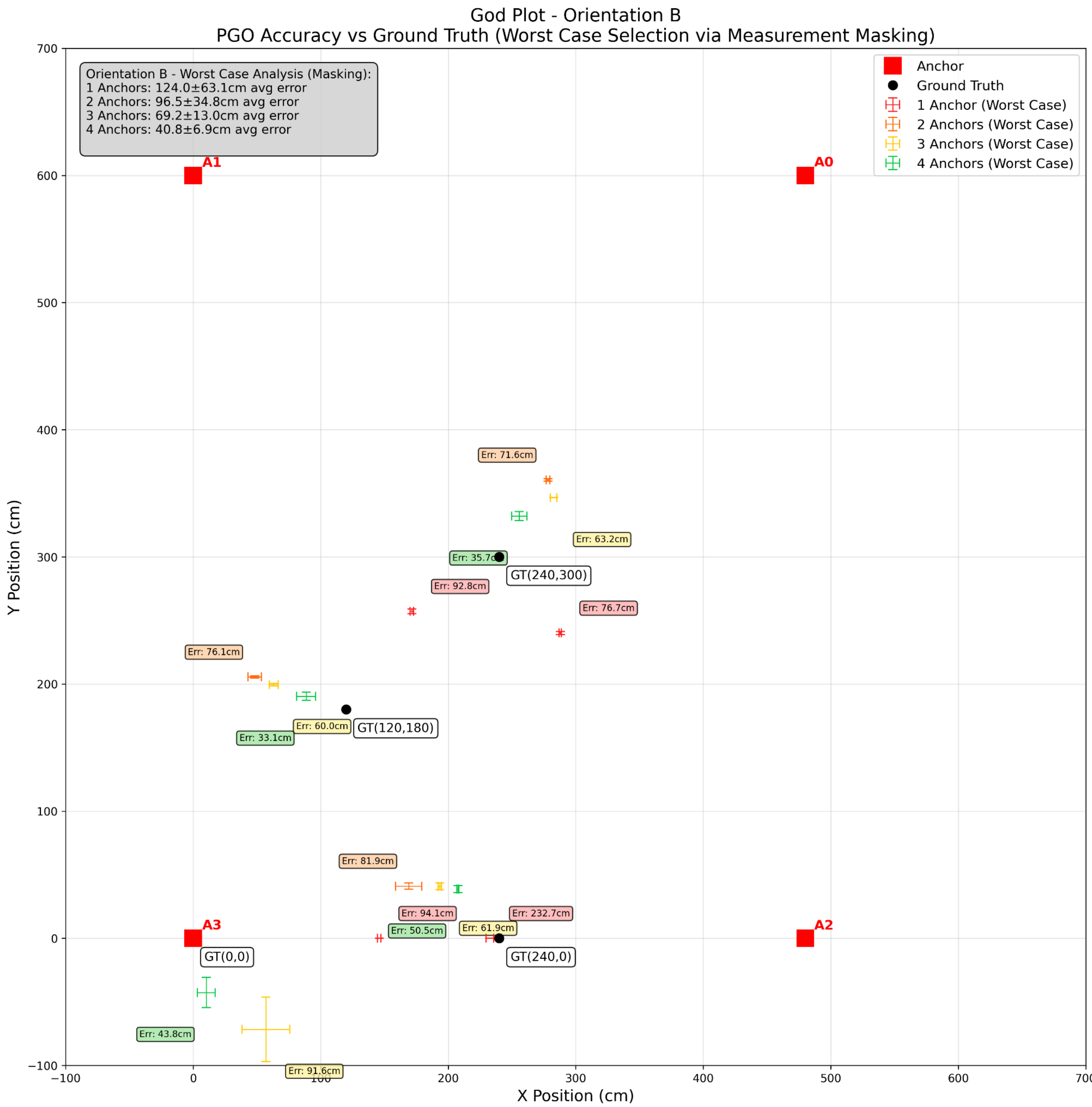


Fig. 20 Overall improvement

This is a set of results from one phone in one direction, increasing in the amount of anchors being used for the PGO algorithm, the length of the whiskers represents the Std Error from our repetitions at the same spot.

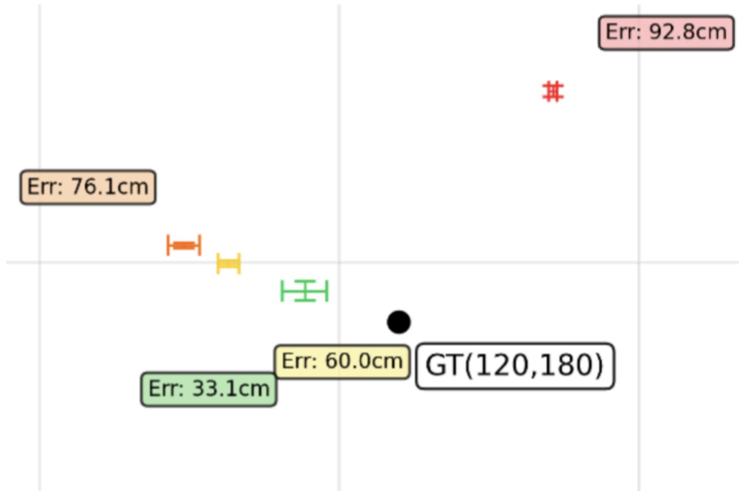


Fig. 21 Zoomed in view

Using this point as a sample, we observe a consistent increase in accuracy against the ground truth as the number of anchors being used increases. Quantitatively for the above case, we observe 32.3% across all locations compared to the single worse anchor scenario.

However, we do observe an increase in variability in the PGO solutions, however this is expected, as when we increase more anchors we inherently increase the amount of noise in the system.

5.3 Resilience against sensor variability

We noticed that the boards do not share consistent inaccurate zones or fixed errors. As such switching the dev-boards in anchor 3 and anchor 1 may give completely different readings.

Despite that, even when there is subpar performance, there is a reduction in error as it gets absorbed into the rest of the system.

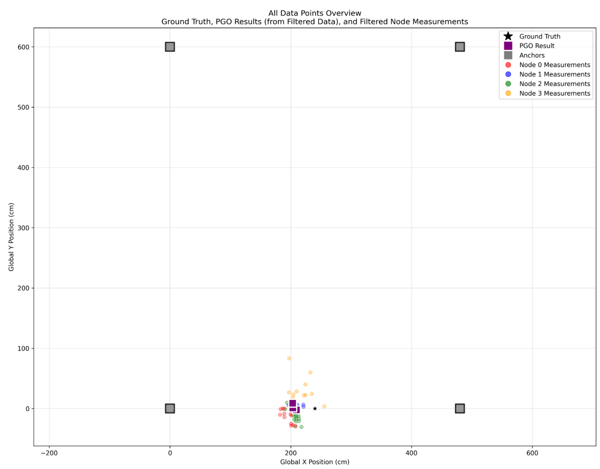


Fig 22. Alternative board config

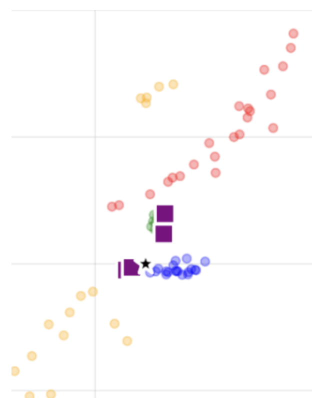


Fig 23. Original board config

This might suggest that there are additional methods to handle the data, we will discuss that in (6. Future iterations).

5.4 Overall accuracy overview

Position (X, Y, Orient)	PGO X	PGO Y	Orient	Count	PGO Error	Worst Anchor	Worst Error
(0, 0, A)	7.3	17.2	A	3	32.1	3	84.4
(0, 0, B)	24.2	-48.8	B	3	54.5	2	285.3
(0, 0, C)	-15.1	50.2	C	3	52.6	2	129.3
(120, 180, A)	110.2	210.8	A	3	32.3	0	107.6
(120, 180, B)	90.4	185.2	B	3	30.3	2	128.1
(120, 180, C)	91.0	200.3	C	3	35.8	2	91.8
(120, 180, U)	99.0	190.5	U	3	23.4	3	54.1
(240, 0, A)	241.1	13.6	A	4	24.2	0	132.5
(240, 0, B)	201.8	18.8	B	3	42.6	0	145.8
(240, 0, C)	237.4	89.3	C	3	116.5	0	458.2
(240, 0, U)	244.8	4.0	U	3	7.1	3	41.9
(240, 300, A)	209.2	315.7	A	3	34.9	0	124.5
(240, 300, B)	263.2	336.8	B	3	43.5	1	82.0
(240, 300, C)	292.0	328.9	C	3	59.5	2	93.9
(240, 300, U)	209.7	325.5	U	3	39.6	1	101.7

Fig 24: Testing accuracy overview table

This summarizes the accuracy across our testing area, we are able to consistently beat the worst case scenario.

Apart from the decrease in error, this increase in reliability is of high importance to B&O and other indoor localization applications as it prevents systems from “losing track” or being grossly wrong about the user’s location and potentially ruining the whole experience.

6. Future iterations

For future efforts in enhancing the middleware system, there are a few methods the team would like to explore.

6.1 Remove UWB source nodes when detected as unreliable

The system performance is highly dependent on the input location readings. The team observes that, depending on the orientation of the boards, some UWB nodes deliver readings that are “confidently wrong” with an offset. This lack of variance renders the data in such occurrences to pass through the k-means filter.

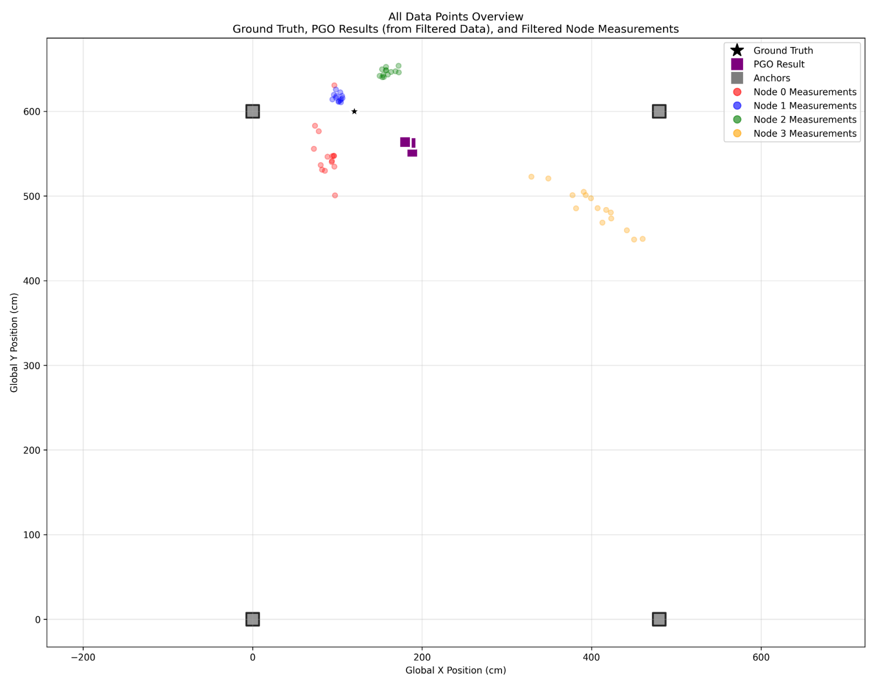


Fig 25. Bad anchor skewing PGO

A possible solution worth exploring is to transform vector data from each anchor to the global frame, and pre-process to get a “consensus” across all anchor sources.

With this consensus, one or a few anchors can be deemed unreliable and essentially turned off before passing to PGO. In the case of the example above, this would result in a solution much closer to the ground truth.

6.2 NXP UWB board improvements

Currently, NXP boards are underperforming in comparison to their specifications, especially in certain directions. The team suspects that signal fingerprinting could be done to the board with the help of NXP's calibration.

Additionally, 2BPs should allow for simultaneous ranging with multiple other 2BP boards and iPhones. The team is currently using the 2BP for Murata to keep in line with B&O's hardware integration. However, we are unable to get the 2BP to work following Murarta's setup guide. UWB modules from other manufacturers could be tested in place of the 2BP as well.

7. Conclusion

This project set out to deliver adaptable room-scale locationing that is packaged as adaptable middle-ware. The intention to drastically lower the pain points such as precision and connection drop with UWB have been met by adopting and leveraging the multi-device environment that B&O owners often have. Concretely, the design of this middleware has blended communication in a distributed system, sensor fusion using PGO as well as outlier detection to produce measurements as a single global estimate.

Across our fixed test rig, the middleware consistently reduced error relative to single-anchor baselines and absorbed anchor-specific inaccuracies. In a specific case, fusing more anchors yielded a ~32% accuracy improvement over the worst-anchor scenario, while our filters rejected ~10% of bad readings before they could skew the optimisation. This addresses a core need, not only lowering mean error by 3-4x, but increasing system reliability and eliminating grossly wrong measurements and deadzones. This presents itself as a large step towards more granular positioning data requirements.

To further build upon this MVP, the intermediate next step include anchor-health gating (temporarily disable biased sources pre-PGO), calibration of UWB boards to tame directional biases, support for simultaneous multi-device ranging. With these iterations, we expect further tightening of accuracy and variance toward the ±10 cm target referenced by acoustic perceptual thresholds.

To finish off, by deploying a novel approach to B&O’s unique edge of having multiple devices capable of ranging in a home, we have piloted that the imperfect UWB signals can be improved upon greatly when combined to form a greater system. This technique and middleware stack sits in a position that can leverage on future UWB improvements, lowering errors as raw data input quality increases as well.

8. Reflection and Lessons Learned

Individual reflections

Ji Yong

This project deepened my appreciation for software as much as hardware. Through the iterations of developing the middleware, I learnt how to write cleaner, modular code (avoiding side effects, using packages, and managing version control properly). A hardware problem can be optimised as a software challenge, and that was an interesting observation on my end. It was rewarding to watch abstract code translate into a working, responsive network. Going forward, I look forward to implementing design and build systems that are both dependable and elegant.

Anitej Datta

The 2nd semester of the design project enabled us to realize our vision in the form of a functional prototype. My initial involvement was in the test setup using the existing UWB-iPhone handshake-based firmware, which helped us identify various pain points, including the errors obtained from ceiling reflections and lack of a systemic data collection process. This eventually led to the redesign of the whole software package, including the integration of MQTT server, function packages and demos.

Key highlights for me would be the continuous iterations in the test setup and data collection, with each cycle leading to a better understanding of the underlying problems and corresponding approach to solutions. Next, the exposure to SWE principles as an EE major, which allowed me to appreciate the benefits of a comprehensive software architecture to handle a system with distributed functionality. Throughout the whole project, I have gained experience with the Raspberry Pi and Linux, which I will continue to use in future projects. Last but not the least, I enjoyed collaborating with my team to solve engineering problems together, and I have learnt a lot from our discussions and brainstorming sessions.

Hong Yi

As my longest running project, I felt that this provided an interesting opportunity in learning how real products are developed and what it really means to know the end user and understand the problem deeply before simply diving into prototyping. The greatest insight to me was when we really understood the B&O customer base and realised we don't need to focus on purely improving UWB readings from the board to get better data and in fact can leverage on their ecosystem and unique distributed system in a user's home.

On a more abstract note, I really observed the non-linear nature of the progress during this project. We had steps that were completely blocking, eg. building good modular software and a good stable rig came before all else. Without either there was no PGO tuning, no iterations on different data handling strategies.

This really made it obvious how "Separation of concerns" isn't just a software concept but one that can be applied to all levels of the stack. Different subsystems and stacks should be able to be developed independently of each other instead of being tightly coupled.

<u>How this informs future designs</u>

We'll keep designing for dynamism first: a decentralized pub-sub backbone (MQTT) lets anchors come and go without brittle coordination, and an API boundary cleanly hands off location to applications. That's now our default architecture pattern for indoor context systems.

On estimation, we won't "stack vectors and pray"—Pose Graph Optimization becomes a standard tool to distribute error globally when edges disagree, with clear weighting hooks for trust calibration.

References

[1] “What is Beolink Multiroom?,” Bang & Olufsen Support. Accessed: Apr. 21, 2025. [Online]. Available: https://support.bang-olufsen.com/hc/en-us/articles/4411572883089-What-is-Beolink-Multiroom

[2] “The History of Beolink (B&O Multiroom).” Accessed: Apr. 21, 2025. [Online]. Available: https://www.bang-olufsen.com/en/sg/story/beolink-history

[3] “Beoconnect Core, Turntable and TV Speaker Hub - Bang & Olufsen.” Accessed: Apr. 21, 2025. [Online]. Available: https://www.bang-olufsen.com/en/sg/accessories/beoconnect-core

[4] G. Retscher, “Indoor Navigation,” in *Encyclopedia of Geodesy*, Springer, Cham, 2016, pp. 1–7. doi: 10.1007/978-3-319-02370-0_9-1.

[5] Apple Inc., “Nearby Interaction with UWB,” Apple Developer. Accessed: Apr. 21, 2025. [Online]. Available: https://developer.apple.com/nearby-interaction/

[6] “NXP UWB IoT Solutions SR150 and SR040.” Accessed: Nov. 07, 2025. [Online]. Available: https://www.nxp.com/docs/en/fact-sheet/UWBIOTFSA4.pdf

[7] “Type2BP | LBUA0VG2BP| EVK product brief.” Accessed: Nov. 07, 2025. [Online]. Available: https://www.murata.com/-/media/webrenewal/products/connectivitymodule/ultra-wide-band/nxp/mtd-apn-081-a-type2bp-evk-product-brief.ashx?la=ja-jp&cvid=20250516024226000000

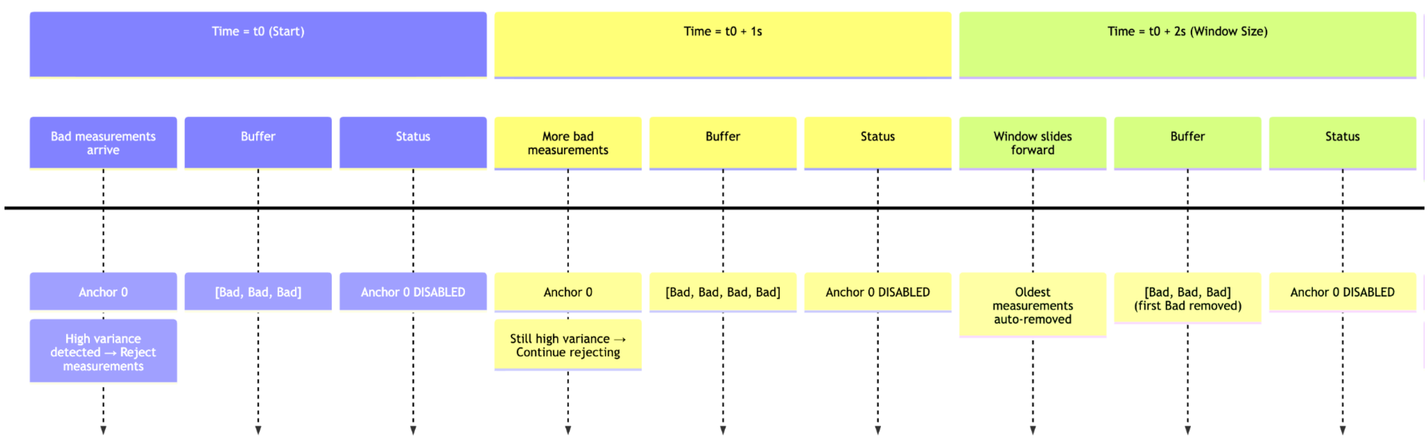
[8] J. Rose, P. Nelson, B. Rafaely, and T. Takeuchi, “Sweet spot size of virtual acoustic imaging systems at asymmetric listener locations)".

[9] “Initiating and maintaining a session,” Apple Developer Documentation. Accessed: Apr. 21, 2025. [Online]. Available: https://developer.apple.com/documentation/nearbyinteraction/initiating-and-maintaining-a-session

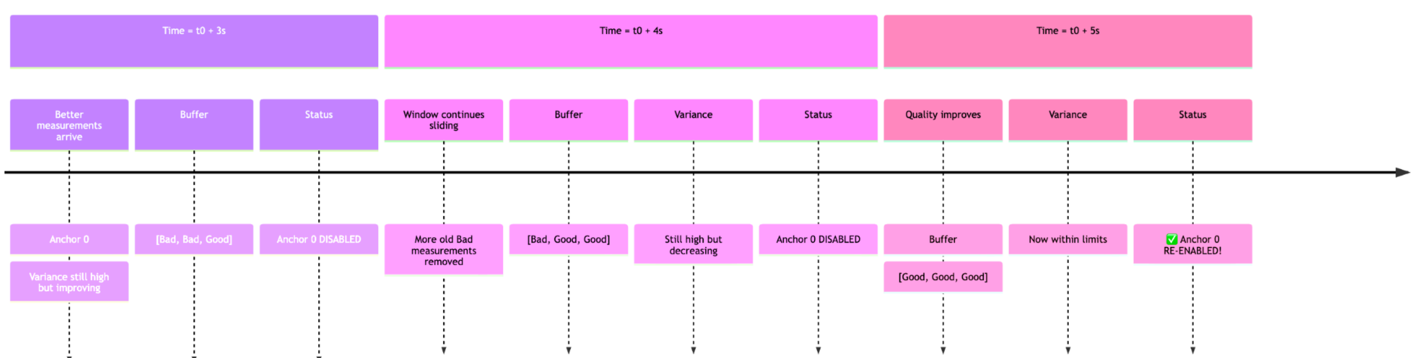
Appendix (A, B, C)

[A] Examples of the continuous anchor variance analysis + data binning in sequential sliding window(s)

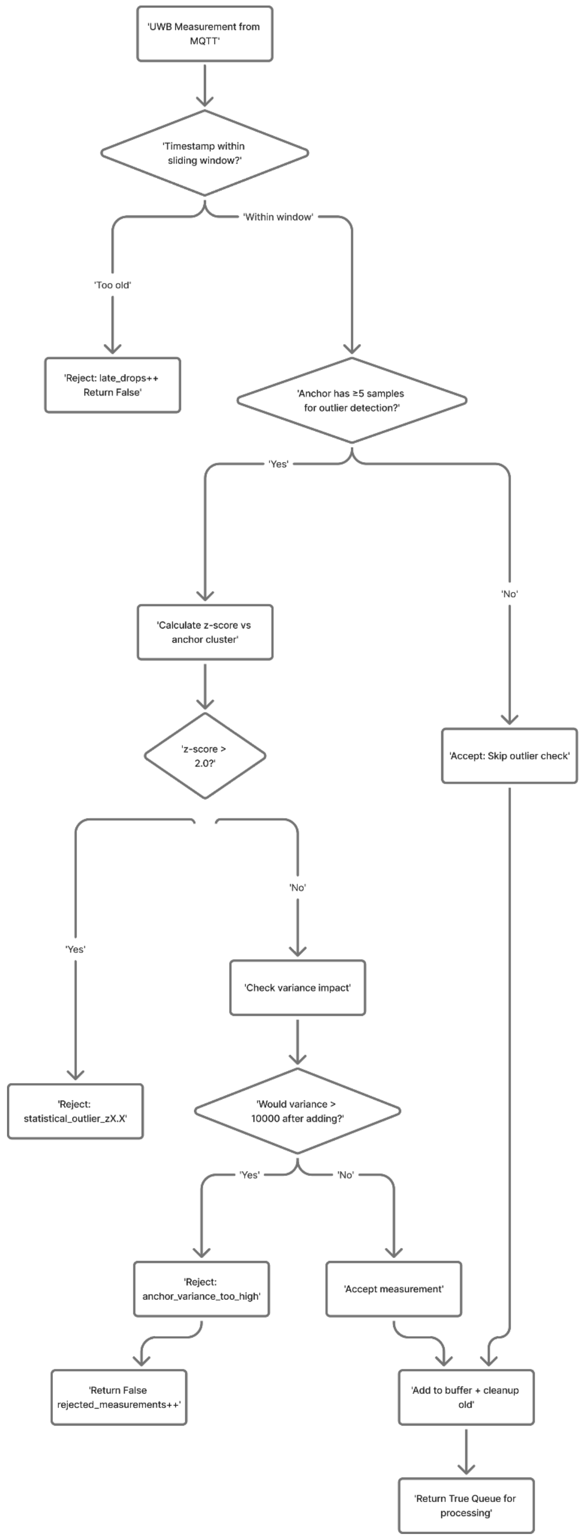
(1)



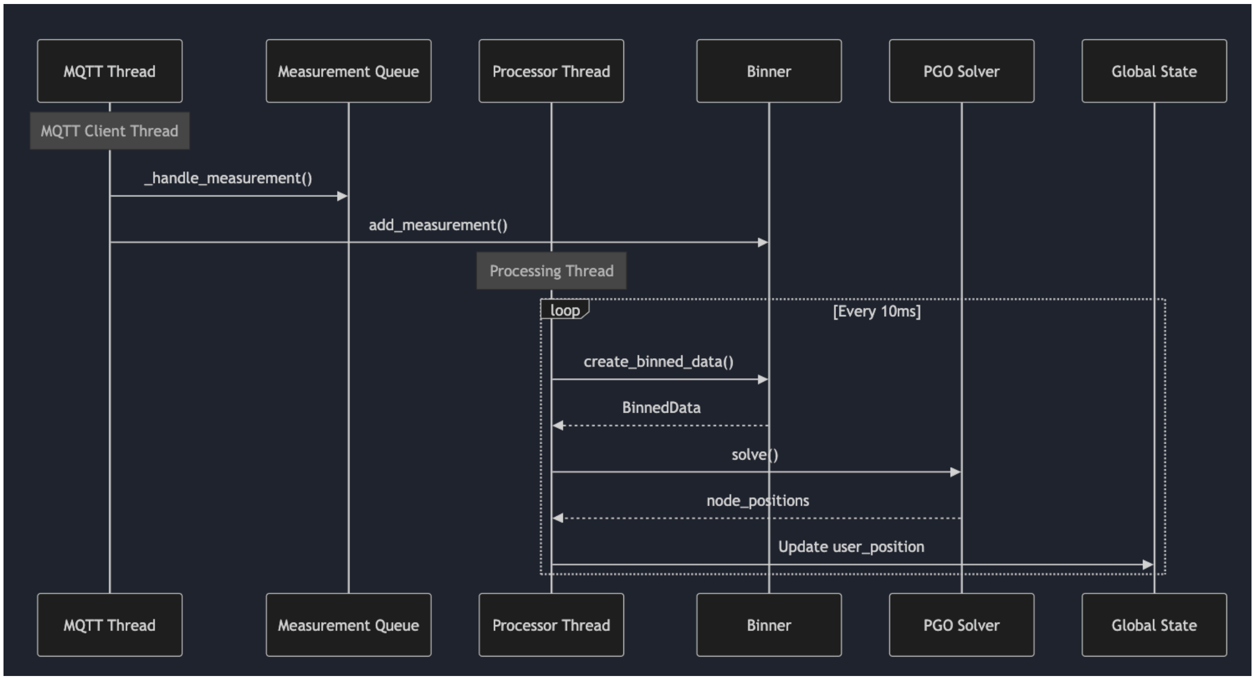
(2)



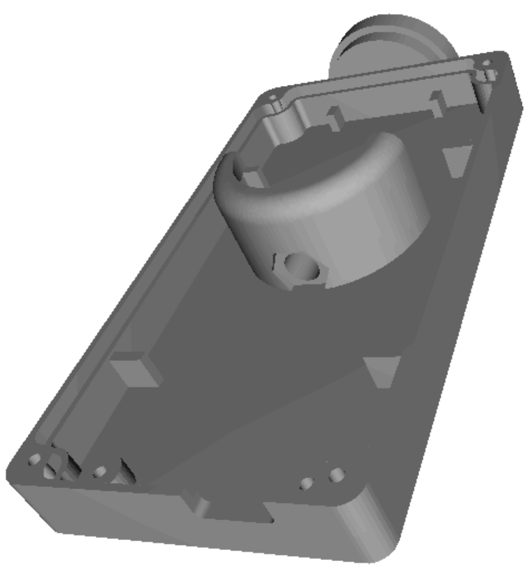
[B] PGO Input Logic Tree (Overarching Decision Diagram for accepting a data vector into the sliding window)



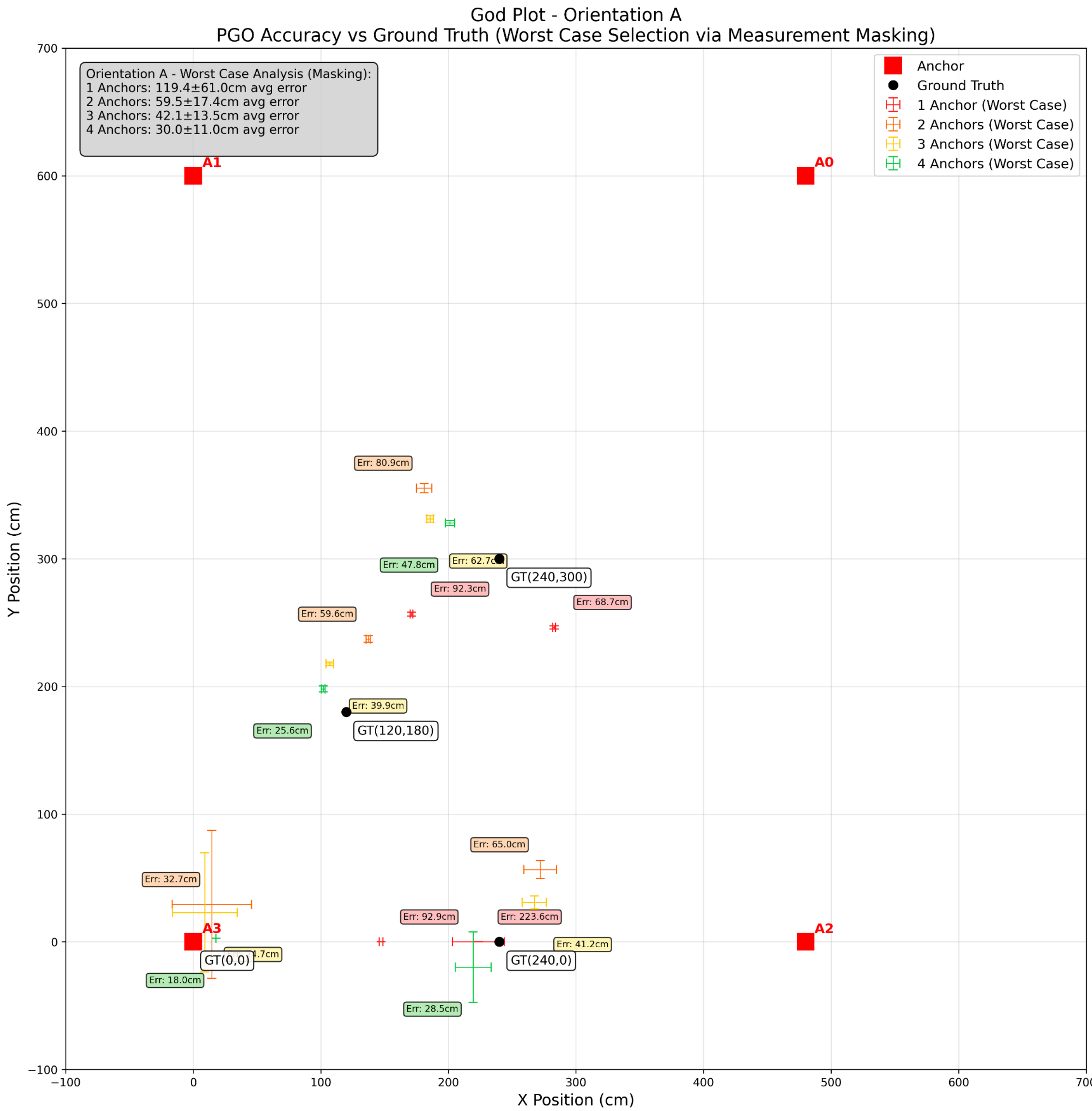
[C] MQTT UML Diagram for processes

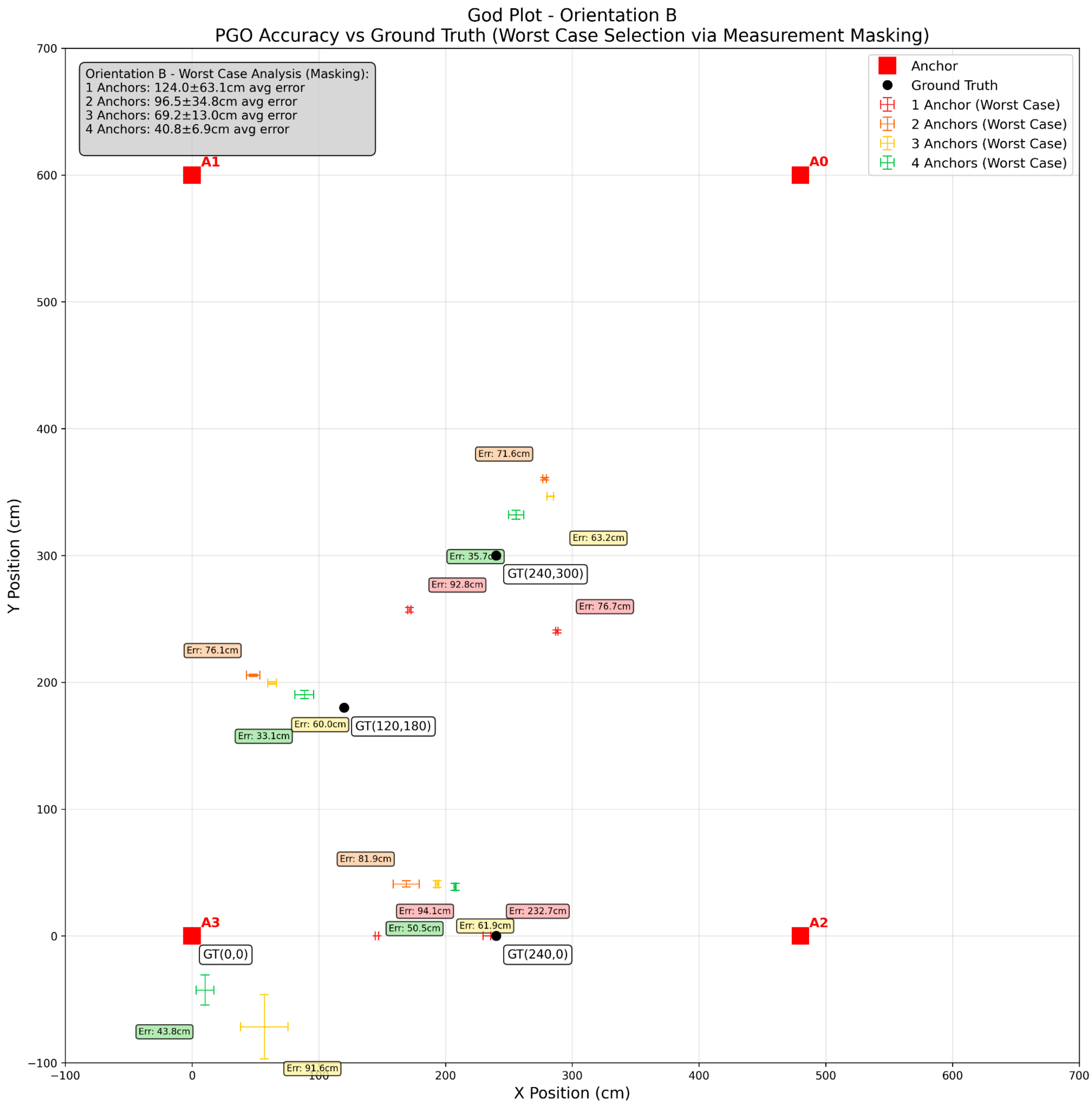


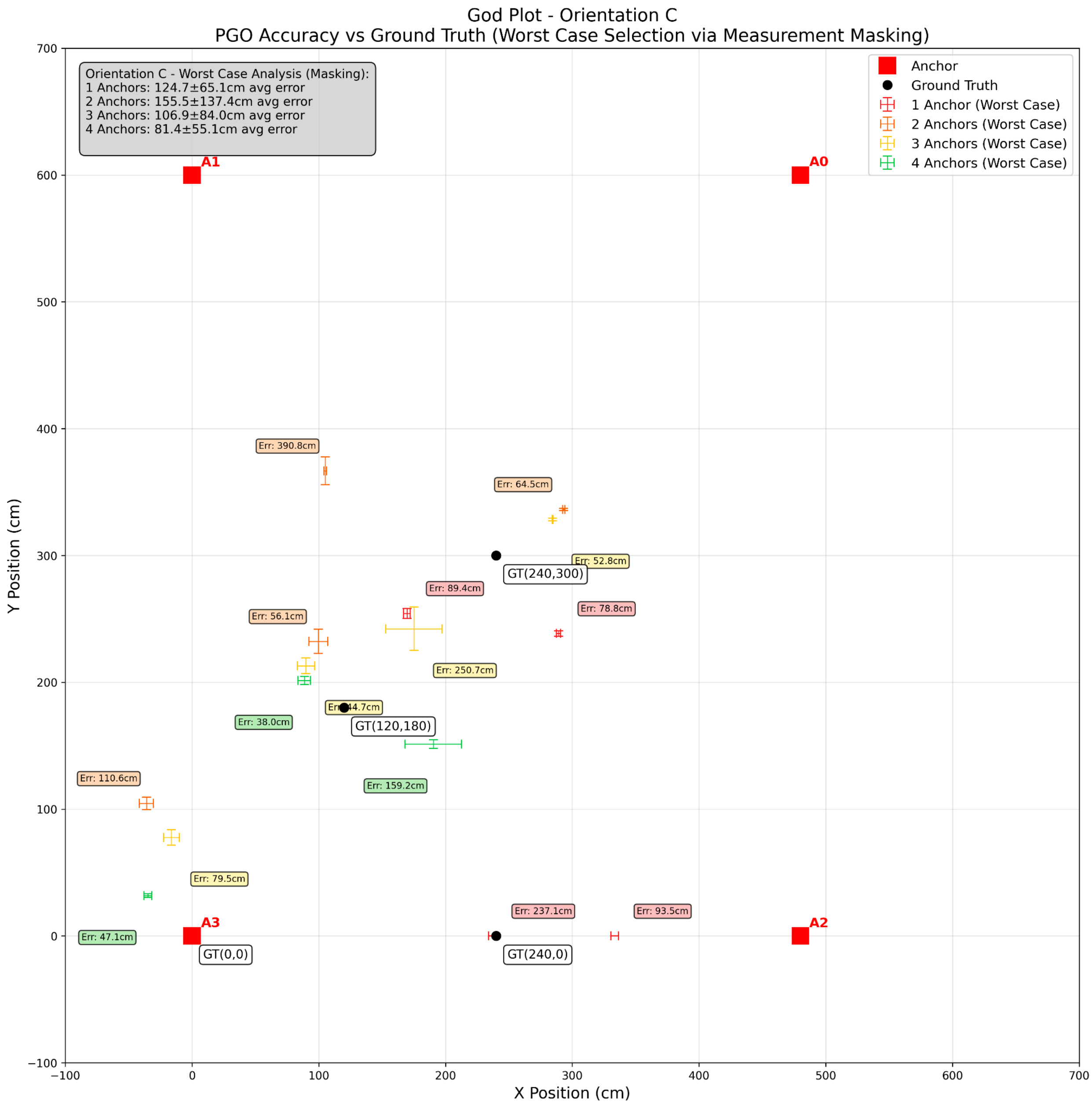
[D] UWB Board mounts from original test setup (straight 90° mount)

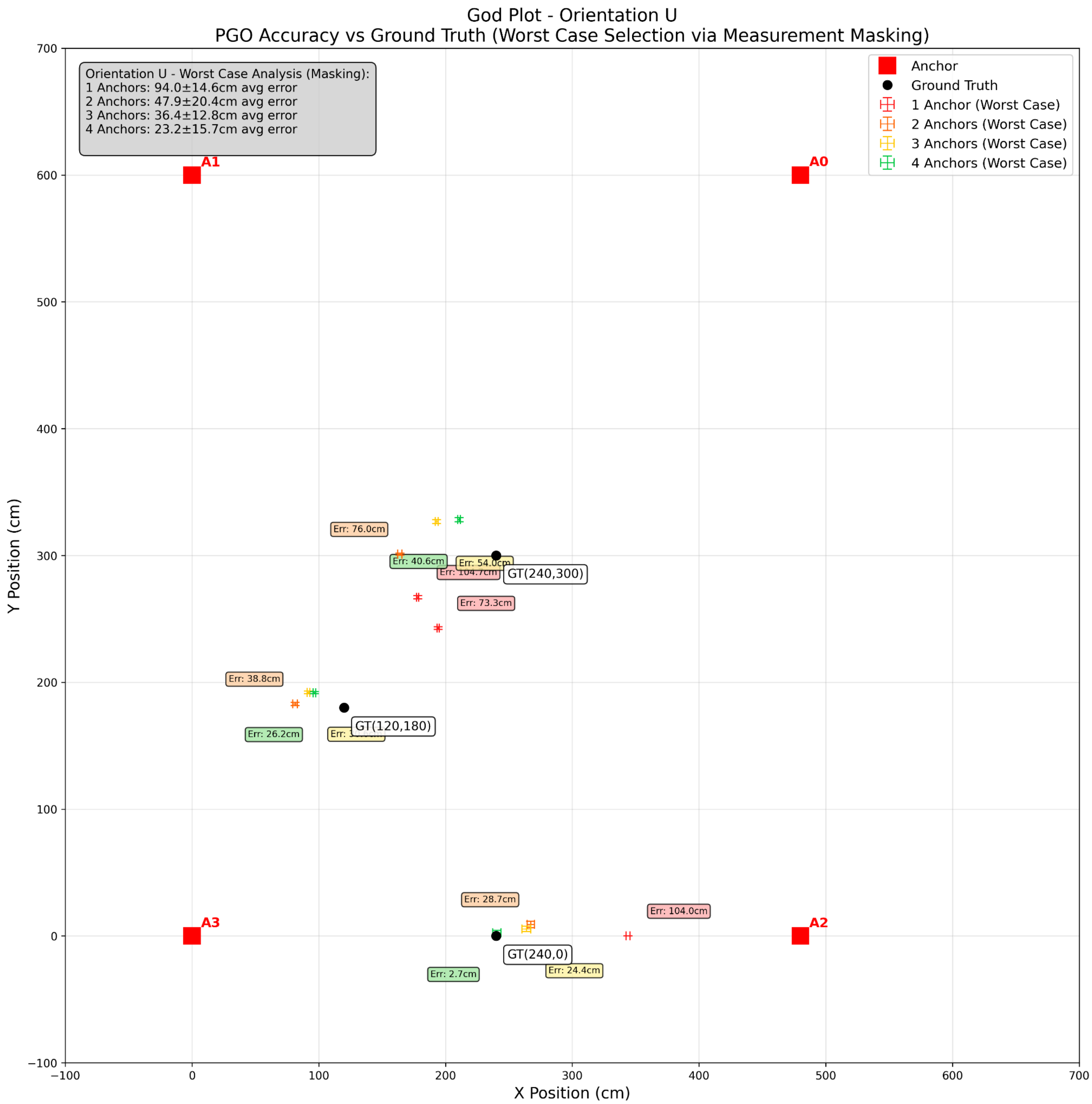


[E] Performance improvements across all directions




God Plot - Orientation B
PGO Accuracy vs Ground Truth (Worst Case Selection via Measurement Masking)
Orientation B - Worst Case Analysis (Masking):
1 Anchors: 124.0±63.1cm avg error
2 Anchors: 96.5±34.8cm avg error
3 Anchors: 69.2±13.0cm avg error
4 Anchors: 40.8±6.9cm avg error
Anchor
Ground Truth
1 Anchor (Worst Case)
2 Anchors (Worst Case)
3 Anchors (Worst Case)
4 Anchors (Worst Case)
A0
A1
A2
A3
GT(240,300)
GT(120,180)
GT(0,0)
GT(240,0)
Err: 71.6cm
Err: 63.2cm
Err: 35.7cm
Err: 92.8cm
Err: 76.7cm
Err: 76.1cm
Err: 60.0cm
Err: 33.1cm
Err: 81.9cm
Err: 94.1cm
Err: 232.7cm
Err: 50.5cm
Err: 61.9cm
Err: 43.8cm
Err: 91.6cm
Y Position (cm)
X Position (cm)


God Plot - Orientation C
PGO Accuracy vs Ground Truth (Worst Case Selection via Measurement Masking)
Orientation C - Worst Case Analysis (Masking):
1 Anchors: 124.7±65.1cm avg error
2 Anchors: 155.5±137.4cm avg error
3 Anchors: 106.9±84.0cm avg error
4 Anchors: 81.4±55.1cm avg error
Anchor
Ground Truth
1 Anchor (Worst Case)
2 Anchors (Worst Case)
3 Anchors (Worst Case)
4 Anchors (Worst Case)
A1
A0
A3
A2
Err: 390.8cm
Err: 64.5cm
Err: 52.8cm
GT(240,300)
Err: 89.4cm
Err: 78.8cm
Err: 56.1cm
Err: 250.7cm
Err: 44.7cm
GT(120,180)
Err: 38.0cm
Err: 159.2cm
Err: 110.6cm
Err: 79.5cm
Err: 237.1cm
Err: 93.5cm
Err: 47.1cm
GT(0,0)
GT(240,0)
Y Position (cm)
X Position (cm)
700
600
500
400
300
200
100
0
−100
−100
0
100
200
300
400
500
600
700


God Plot - Orientation U
PGO Accuracy vs Ground Truth (Worst Case Selection via Measurement Masking)
Orientation U - Worst Case Analysis (Masking):
1 Anchors: 94.0±14.6cm avg error
2 Anchors: 47.9±20.4cm avg error
3 Anchors: 36.4±12.8cm avg error
4 Anchors: 23.2±15.7cm avg error
Anchor
Ground Truth
1 Anchor (Worst Case)
2 Anchors (Worst Case)
3 Anchors (Worst Case)
4 Anchors (Worst Case)
A0
A1
A2
A3
GT(240,300)
GT(120,180)
GT(240,0)
Err: 76.0cm
Err: 40.6cm
Err: 54.0cm
Err: 104.7cm
Err: 73.3cm
Err: 38.8cm
Err: 26.2cm
Err: 28.7cm
Err: 104.0cm
Err: 2.7cm
Err: 24.4cm
X Position (cm)
Y Position (cm)

[F] Performance improvements across all directions