

Battery Efficiency in Vision-Based Indoor Navigation: Energy Considerations in NaVIP*

Jun Yu^{1,*}, Yitian Yang¹, Mushu Li¹ and Vinod Namboodiri^{1,2,*}

¹Department of Computer Science and Engineering, P.C. Rossin College of Engineering and Applied Science, Lehigh University, 113 Research Drive, Bethlehem, PA 18015, USA

²Department of Biostatistics and Health Data Science, College of Health, Lehigh University, 124 East Morton Street Suite 155, Bethlehem, PA 18015 USA

Abstract

Wayfinding in unfamiliar indoor environments is a challenging task for visually impaired persons (VIPs) due to the limitations of mapping and satellite-based positioning. Recent indoor localization solutions have emerged with good results in terms of positioning accuracy. Not much is known, however, about how they compare in terms of energy consumption, an important metric for use on mobile devices. This study presents a systematic comparison of battery consumption across three core localization technologies—computer vision (CV)-based, inertial measurement unit (IMU)-based, and ARKit-based—within the context of our proposed iOS application, NaVIP. Using the commercial app GoodMaps as a comparison, energy usage across navigation, exploration, and idle modes are evaluated under standardized testing conditions. Results show that while CV and ARKit-based methods offer accurate localization, they incur substantially higher power usage than IMU-only solutions. Notably, NaVIP's server-assisted CV approach consumes significantly less energy on the user's device compared to ARKit's fully on-device computation pipeline, even with live camera feed rendering enabled. These findings highlight the benefits of offloading compute-heavy tasks to backend servers. Subsequent analysis provides actionable design insights for developing energy-efficient, accessible navigation tools for durable use by mobile end users.

Keywords

Indoor navigation, smartphone application, battery life, energy efficiency, offloading computation

1. Introduction

Vision-based localization has become a prominent approach in recent indoor navigation systems, offering improved spatial awareness and robustness through the use of real-time camera input and computer vision algorithms. However, the increased computational demands of these methods raise important concerns regarding battery life—especially for applications deployed on mobile devices. This issue is particularly critical for blind and visually impaired (BVI) users who depend on such applications for independent mobility and cannot just switch off the application when remaining battery levels are lower. This population is also less likely to detect low-battery warnings or respond quickly to sudden device shutdowns [1]. Energy efficiency is a critical design consideration for mobile applications, particularly in location-based services, where minimizing battery consumption is essential for maintaining usability, ensuring user satisfaction, and supporting prolonged usage [2]. In this context, energy efficiency is not just a technical metric but a practical requirement for ensuring safety, accessibility, and continuous operation in real-world navigation scenarios.

While existing vision-based solutions address many of the fundamental navigational challenges for visually impaired persons (VIPs)—including infrastructure independence [3], real-time performance, and localization accuracy—their energy efficiency remains largely underexplored. In assistive navigation, battery longevity is not merely a technical concern but a user-centered requirement for trust and long-term usability. Many vision-based solutions rely on continuous camera input and real-time processing, both of which can be power-intensive by nature, particularly during extended navigation tasks. This makes energy efficiency a crucial consideration in the design of scalable and dependable navigation

IPIN-WCAL 2025: Workshop for Computing & Advanced Localization at the Fifteenth International Conference on Indoor Positioning and Indoor Navigation, September 15–18, 2025, Tampere, Finland

*Corresponding author.

✉ juy220@lehigh.edu (J. Yu); yiy523@lehigh.edu (Y. Yang); mul224@lehigh.edu (M. Li); vin423@lehigh.edu (V. Namboodiri)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

systems. To address this, we developed NaVIP¹ (Navigation for VIPs), a deep learning-based absolute pose regression (APR) system [5] that directly regresses the user’s camera position and orientation from a single image captured on their phone. Compared to traditional vision-based approaches such as retrieval-heavy [6] or 3D-2D feature matching methods [7, 8], NaVIP offers a more lightweight and flexible alternative, enabled by its end-to-end APR framework. In contrast, ARKit and structure-from-motion (SfM) methods can achieve high localization accuracy but often impose substantial computational and memory demands on the client device, making offloading challenging and accelerating battery drain during real-world use. Although NaVIP can alleviate this through a streamlined end-to-end architecture, it remains unclear how its energy consumption compares with ARKit-based or hybrid localization methods. To fill this gap, we present an empirical study measuring battery usage across multiple infrastructure-free navigation approaches—including core APR-based solution of NaVIP, IMU-only solutions, and ARKit-enabled systems—under realistic task-oriented conditions. By analyzing these trade-offs, we aim to inform the design of more energy-efficient and reliable navigation tools for VIPs, where power efficiency is a core component of overall usability.

NaVIP adopts a client-server architecture that offloads the majority of computation, including camera pose regression and path planning, to the server, thereby alleviating the energy burden on user devices. This design significantly reduces on-device processing and leads to lower battery consumption during use. In our experiments, we systematically evaluate battery usage on user devices and compare NaVIP’s consumption patterns with those of alternative localization approaches. Notably, our findings show that APR-based localization in NaVIP conserves battery life more effectively than ARKit-based solutions, even when live camera feed rendering is enabled on the client side. Further energy savings can be achieved by disabling on-screen rendering while continuing to transmit video frames to the server—an approach that maintains localization accuracy while significantly reducing power draw. The underlying cause of this discrepancy lies in architectural differences: ARKit performs all vision-based localization and sensor fusion directly on the device, leading to consistently higher energy usage and thermal load. In contrast, NaVIP leverages server-side inference with lightweight client-side operations, enabling a more power-efficient experience that is well-suited for long-duration indoor navigation tasks in real-world settings.

In summary, this paper presents the first systematic study of battery consumption in indoor navigation systems designed for visually impaired users, focusing on a comparison of CV-, IMU-, and ARKit-based approaches. Our contributions are three-fold: (1) we propose a lightweight, server-assisted APR framework for vision-based localization that minimizes client-side power consumption; (2) we conduct a comprehensive, task-oriented evaluation of battery usage across CV-, IMU-, and ARKit-based methods in real-world scenarios; and (3) we identify key architectural decisions, such as offloading inference from the client, that significantly improve energy efficiency without compromising usability or accuracy. Together, these contributions offer actionable insights for designing sustainable, inclusive, and power-aware indoor navigation solutions for the visually impaired.

2. Related Work

Indoor navigation systems have become a critical area of research, particularly in efforts to support blind and visually impaired (BVI) users in independently navigating complex indoor environments such as transit stations, airports, and university buildings [9], where traditional physical signage is only accessible for sighted persons. Existing solutions employ a variety of technologies, including Wi-Fi fingerprinting [10], Bluetooth low energy (BLE) beacons [11], augmented reality (AR)-based tracking [12], visual feature matching [6], and inertial sensors [13]. These systems aim to deliver real-time guidance, improve spatial awareness, and foster autonomy for VIPs, but they vary in terms of cost, scalability, and infrastructure dependence. Traditional approaches like WiFi and BLE beacons that rely on installing external tags or transponders face challenges in scalability and long-term maintenance, especially due to the need for frequent recalibration and their impact on building aesthetics. As a result, these infrastructure-heavy solutions have gradually fallen out of favor. Depending on the choice

¹More details are provided in our accepted work submitted to ICCV Workshop ACVR 2025 [4].

of transponders, their physical deployment, the localization algorithm, and the characteristics of the indoor space, these systems report different levels of accuracy [14, 15, 16].

Recent studies have evaluated indoor localization approaches based on criteria such as environmental robustness and usability. ARKit-based systems have demonstrated strong performance in responsiveness and spatial awareness by fusing camera and IMU data through visual-inertial odometry (VIO)[17, 12], and applications like GoodMaps have been adopted for commercial use. Vision-based methods leveraging image retrieval[6] or 3D-2D feature matching [7, 8] have gained popularity due to their infrastructure independence and reliable localization in visually complex environments. Overall, both approaches offer viable solutions for indoor localization, each with distinct strengths and limitations. Retrieval methods prioritize simplicity and speed but often struggle with scalability, while 3D-2D techniques offer higher accuracy and robustness at the cost of greater computational demands. The choice between them typically depends on the specific requirements and constraints of the navigation application. To address these trade-offs, we developed NaVIP, a novel navigation system that uses an APR approach to regress the user’s location directly from any single image. However, both ARKit- and vision-based methods often involve substantial computational overhead and may pose battery challenges for mobile navigation systems. IMU-only and SLAM-based methods are lightweight and infrastructure-free, but they tend to suffer from drift over time and difficulty with initial localization. As a result, they are commonly integrated into hybrid models to correct accumulated errors [13].

While previous studies have extensively addressed accuracy, robustness, and usability, they rarely examine energy consumption—an equally critical factor in real-world deployment, especially for accessibility applications. Battery life is a key determinant of user experience in mobile computing [2], and for BVI users, sudden device shutdowns can compromise safety, navigation continuity, and user trust. Navigation aids that demand continuous camera access, real-time sensor fusion, or AR rendering can quickly deplete battery resources, making them unsuitable for long-duration tasks. Despite this, few studies offer empirical measurements of energy usage across different localization strategies, leaving a major gap in our understanding of their practical viability. Our work addresses this gap by systematically evaluating the battery consumption of three leading infrastructure-free localization methods—vision-based (APR), ARKit-based, and IMU-based—within our assistive navigation system, NaVIP. We integrate these modules into a unified app framework and apply a consistent evaluation protocol across navigation, exploration, and idle states. Additionally, we use APR as a universal initializer to support fair comparison across modalities. Our findings aim to inform not only the localization community but also researchers and developers focused on building power-aware, reliable assistive technologies for real-world deployment.

3. Procedures

3.1. Study Objective

This study aims to evaluate and compare the battery consumption of four different indoor navigation variants for visually impaired users: (1) NaVIP, a vision-based navigation system developed by our group; (2) an ARKit-based navigation solution leveraging Apple’s augmented reality framework; (3) an IMU-based system relying on inertial sensors; and (4) GoodMaps, a commercial navigation app. The primary objective is to assess each method’s energy efficiency using standardized testing conditions under two under realistic scenarios below.

- **Navigation Mode:** Users specify a destination, and the system computes the shortest path, providing continuous voice prompts and real-time guidance based on position and orientation.
- **Exploration Mode:** As the default mode, it offers audio feedback on nearby points of interest (PoIs), supporting spatial awareness without a set destination.

Vision-Based (APR) Navigation. APR uses camera input and a deep neural network to directly predict 6-DoF poses from image frames, eliminating the need for traditional visual localization steps such as feature matching and optimization. Offloading inference to a backend server enables lightweight client-side operation, making APR particularly suitable for visually impaired users due to its infrastructure-free, map-free, and energy-efficient design.

ARKit-Based Navigation. ARKit fuses visual-inertial odometry, SLAM, and motion sensors to estimate position entirely on-device. It delivers accurate, low-latency localization but is resource-intensive, leading to high battery usage and thermal buildup during extended use. While effective for route guidance, its energy demands limit prolonged operation.

IMU-Based Navigation. IMU navigation estimates motion via accelerometers and gyroscopes, offering fast, energy-efficient tracking without external input. However, it suffers from drift and lacks spatial awareness. In our setup, it relies on APR to initialize the starting pose before switching to inertial tracking for short-term navigation.

3.2. Test Sites and Participants

One of the biggest challenges of comparative evaluations across indoor navigation systems is the lack of access to setup each system at the site. For example, GoodMaps is deployed at a few locations around the U.S. based on commercial agreements that provided them access to deploy their system. Similarly, academic research typically has access to only collaborative academic environments such as university buildings. Thus, it is important to carefully construct testing scenarios that support a fair comparison across systems deployed at different sites. Therefore, all tests were conducted over pre-defined routes of the same length and requiring similar duration of time to complete regardless of site location.

For this work, NaVIP—including its isolated Vision-, ARKit-, and IMU-based variants—were tested in the Health, Science, and Technology (HST) Building at Lehigh University (Bethlehem, PA), while GoodMaps was tested at the Newark Liberty International Airport (EWR) Terminal B (Newark, NJ). All tests were conducted by a single facilitator or trained sighted volunteer to ensure consistent handling and movement across trials. No human subject data was collected, and no participation from visually impaired users was required during battery profiling.

3.3. Device Configuration

All tests were conducted on the same device model—an iPhone 13 Pro—to eliminate hardware variability. Prior to each test run, the device settings were standardized in Table 1.

Table 1: Standardized Device Settings for Battery Test

Setting	Value
Battery Level at Start	100%
Screen Brightness	100%
Volume	25%
Wi-Fi	ON
Bluetooth	OFF
Airplane Mode	ON
Do Not Disturb Mode	ON
Low Power Mode	OFF

3.4. System-Level Battery Profiling

Each app was evaluated in both navigation and exploration modes. The navigation mode guides the user from a designated starting point to a predefined indoor destination (e.g., check-in), while the exploration mode is intended for users who simply want to understand their surroundings without specifying a destination—similar to viewing a plain map. A key distinction is that exploration mode does not require rendering a path on the floor plan or in 3D scenes. Each test run followed the same duration and measurement protocol.

- The test app was launched, and the navigation/exploration task was initiated.
- The tester followed a predefined route while holding the phone at chest height.
- Upon reaching the destination, the test was concluded, and the following data were recorded:
 - Start and end battery percentages;
 - Any anomalies (e.g., thermal warnings, app crashes)

In addition to navigation and exploration modes, each app was also tested in an idle state, left on the home screen, to serve as a baseline for comparison. To ensure consistency and capture measurable battery consumption, each test session was fixed at 30 minutes in duration.

3.5. Module-Level Battery Profiling

Since the two apps are built on distinct technologies—GoodMaps relies heavily on ARKit for localization, whereas NaVIP streams video frames to a backend server for camera pose estimation—we aim to identify which specific components contribute most to battery consumption. To enable a fair comparison, we

Table 2
Battery Usage Across Different Navigation Apps

Application	Test Site	Mode	Functionality	Battery Drop	Notes
GoodMaps	HST	Idle	Home screen	5%	Normal
	EWR	Navigation	ARKit (+ 3D rendering), map rendering, voice prompt, haptics	23%	Overheating
	EWR	Exploration	ARKit, map rendering	20%	Slightly overheating
NaVIP	HST	Idle	Home screen	5%	Normal
	HST	Navigation	Upstream video streaming, voice prompt, haptics, live camera feed rendering	13%	Normal
	HST	Exploration	Upstream video streaming, voice prompt, haptics, live camera feed rendering	14%	Normal

also integrated all major localization modules (server-side APR-based, ARKit-based, and IMU-based) into the NaVIP app².

For the test of ARKit module, we designed to use the `ARWorldTrackingConfiguration` but avoided leveraging any additional ARKit features beyond basic user tracking. Notably, ARKit-based localization may utilize `CoreMotion` for refining camera orientation predictions; therefore, we also report results for an ARKit + IMU configuration.

Each module was tested under identical conditions for a fixed duration. During each module test:

- NaVIP was run in exploration mode to avoid confounding from path planning or rerouting.
- The tester followed a predefined route while holding the phone at chest height.
- After 30 minutes, the test was concluded, and the following data were recorded:
 - Start and end battery percentages
 - Any anomalies (e.g., thermal warnings, app crashes)

To ensure consistency, all tests were conducted within the same app and in the same environment.

4. Experimental Results

To evaluate the energy efficiency of different indoor navigation strategies, we conducted battery profiling experiments under realistic usage scenarios. We compared the power consumption of ARKit-based tracking, IMU, and NaVIP’s video streaming architecture during typical user interactions. Experiments were structured at two levels: system-level tests (covering full app activity across navigation, exploration, and idle modes) and module-level tests (isolating individual components in controlled settings). All tests ran for 30 minutes on identical hardware and in the same environment for fair comparison. Results and analysis from both levels are presented below.

4.1. System-Level Results

The results presented in Table 2 reveal several important insights into the battery efficiency of different indoor navigation solutions under varying modes of operation—navigation, exploration, and idle.

Idle Mode. In the idle condition, where each app was left running on its home screen without active navigation or exploration tasks, both NaVIP and GoodMaps exhibited minimal battery consumption, each with a 5% drop over 30 minutes. This low energy usage confirms that neither app imposes significant background overhead when inactive, making it an appropriate baseline for comparison.

Navigation Mode. Navigation mode, which involves full system functionality, showed the highest battery drain across all configurations. GoodMaps recorded the most substantial battery drop at 23%, accompanied by signs of overheating. This can be attributed to its use of ARKit for localization, combined with intensive 3D rendering, continuous map updates, voice prompts, and haptic feedback. In contrast, NaVIP’s navigation mode consumed 13%, demonstrating better power efficiency. Although NaVIP uses continuous upstream video streaming and client-side camera feed rendering, it offloads pose computation and path planning to the server, reducing on-device processing demands and helping avoid thermal issues. The difference in battery drain between these two systems underscores the energy-saving benefits of NaVIP’s client-server architecture.

Exploration Mode. In exploration mode, which provides spatial awareness without destination-based guidance, both apps again showed different power profiles. GoodMaps consumed 20%, while

²Note that NaVIP functions effectively without relying on ARKit and IMU; they are included solely for the purpose of module-level battery profiling.

NaVIP used 14%. This reflects a moderate reduction from navigation mode for both systems, likely due to the absence of continuous path planning and rerouting. However, GoodMaps still showed signs of slight overheating, suggesting that its ARKit-driven rendering pipeline remains active even without explicit navigation tasks. NaVIP, while maintaining upstream video streaming and rendering, benefits from offloaded computation and showed no thermal concerns. Notably, the comparison between navigation and exploration modes also suggests that specialized audio prompts and haptic feedback for VIPs do not introduce additional battery overhead.

Functionality-Level Impact. A breakdown of each app’s functional components—ARKit-based localization, video streaming, 3D rendering, voice prompts, and haptics—highlights the dominant role of visual localization and rendering in battery consumption. GoodMaps performs real-time ARKit tracking and local 3D scene rendering on-device, placing sustained load on the CPU, GPU, and motion co-processor. This results in higher power draw and thermal buildup during extended use. In contrast, NaVIP reduces on-device load by offloading pose estimation to a remote server. It relies on lightweight upstream video streaming and omits local 3D visualization, minimizing GPU usage and leveraging the smartphone’s energy-efficient video pipelines. While both apps include voice and haptic feedback for accessibility, our tests show these features have minimal battery impact due to their low-power, intermittent use. Overall, the comparison emphasizes how architectural choices—especially offloading—can significantly influence energy efficiency in mobile navigation systems.

Summary. The results validate that NaVIP achieves a favorable trade-off between usability and energy consumption. Its architecture enables continuous localization and user guidance while avoiding the battery drain and overheating risks observed in fully on-device ARKit-based systems. These findings demonstrate the viability of server-assisted vision-based navigation systems for visually impaired users, especially in scenarios requiring prolonged usage.

4.2. Module-Level Results

To further isolate the battery impact of individual localization components, we conducted module-level tests by embedding all major localization technologies—server-side APR, ARKit, IMU, and ARKit+IMU—within the NaVIP app. This controlled setup eliminated external variables such as app-specific

UI rendering or back-end integration differences, ensuring a consistent evaluation environment. Each module was tested independently under identical conditions for 30 minutes. It is worth noting that all the methods above rely on the APR model for initial localization and for correcting accumulated errors when the navigation path deviates from expected use. Since APR is invoked infrequently during typical operation, its overall energy contribution is minimal and can be reasonably considered negligible. The results in Table 3 reveal clear differences in battery usage among modules:

Upstream Video Streaming. The upstream video streaming module, representing NaVIP’s default visual input system, consumed 12% of the battery during the 30-minute test session. This moderate energy usage is expected given the continuous activation of the camera and the real-time transmission of video frames to the backend server for processing. Notably, the energy cost of this module does not include any on-device inference, as all computationally intensive tasks such as pose estimation and scene understanding are performed on the server side. This architectural choice not only reduces battery consumption but also limits device overheating and preserves system responsiveness. The thermally stable performance throughout the test suggests that upstream video streaming can be a viable long-term localization strategy for mobile applications when paired with efficient server-side infrastructure. Moreover, since upstream video streaming consumes constant bandwidth in most cases, future work might also examine its trade-off with network availability and data plan constraints in real-world deployment.

ARKit. When the ARKit module was activated using `ARWorldTrackingConfiguration`, the device experienced a 16% battery drop over 30 minutes. This increase, relative to upstream video streaming,

Table 3: Battery Usage Across Different Module Components

Module Type	Test Site	Battery Drop	Notes
Upstream Video Streaming	HST	12%	Normal
ARKit	HST	16%	Warming
IMU	HST	5%	Normal
ARKit + IMU	HST	17%	Warming

reflects the higher computational demands of ARKit’s visual-inertial odometry (VIO) pipeline. ARKit continuously fuses camera frames with gyroscope and accelerometer readings to track motion and orientation, requiring substantial CPU and GPU cycles for real-time feature tracking, depth estimation, and pose updates. While thermal throttling was not triggered, the device exhibited mild warming, indicating a sustained processing load. Unlike NaVIP’s offloaded approach, ARKit performs all computations locally, which may lead to further performance degradation over extended usage due to thermal accumulation. This result highlights a trade-off: although ARKit offers precise short-term tracking and responsiveness, it incurs a heavier energy burden that may not scale well for long navigation sessions in resource-constrained scenarios.

IMU. As expected, the IMU-based module demonstrated the lowest battery consumption at only 5%. This efficiency is a result of the system’s reliance solely on low-power inertial sensors, such as accelerometers and gyroscopes, which operate independently of camera input and require minimal computation. These sensors can estimate user motion through dead reckoning, using integrated velocity and orientation changes over time. However, the inherent drawback is cumulative error—drift increases with time and distance traveled, leading to degraded positional accuracy unless external correction (e.g., APR or visual landmarks) is applied. In our testbed, the IMU module still relied on NaVIP’s APR component to establish the initial pose, after which dead reckoning was used to track motion. While this approach is ideal for conserving power during short sessions or in environments where camera use is infeasible, its standalone viability for accurate indoor navigation remains limited without periodic recalibration.

ARKit + IMU. Combining ARKit with IMU integration resulted in the highest energy consumption across all modules, with a 17% battery drop. This configuration leverages Apple’s CoreMotion framework to further refine ARKit’s pose estimation by fusing high-frequency inertial sensor data with the visual odometry pipeline. While this improves motion tracking robustness—particularly in fast movements or low-light conditions where visual cues are less reliable—it significantly increases computational load on the device. The combined system must continuously synchronize multi-modal sensor streams, maintain real-time frame processing, and update environmental maps, all of which place sustained demand on both CPU and GPU. The warming observed during the test, though not critical, suggests a growing thermal footprint that may affect long-term usability and device longevity. This setup, while offering the most accurate short-term localization, may not be practical for energy-constrained accessibility applications without adaptive switching or load balancing strategies.

Summary. These module-level insights confirm that ARKit-based localization is inherently more power-intensive than upstream video streaming or IMU-only alternatives, particularly when paired with additional sensor fusion. NaVIP’s architectural choice to isolate visual input and delegate processing to the server demonstrates a strategic advantage in minimizing client-side battery drain. Future designs may consider hybrid approaches that dynamically activate or deactivate modules based on usage context to optimize power efficiency without compromising localization performance.

System vs. Module Observations. Comparing module-level results to system-level evaluations (Table 2), we observe that GoodMaps’ 23% battery drop in navigation mode is consistent with the behavior of the ARKit + rendering pipeline when deployed in a full-featured app. NaVIP’s 13–14% consumption across navigation and exploration modes aligns closely with the 12% attributed to upstream video streaming, confirming that its energy efficiency stems from offloading compute-intensive tasks to the server.

5. Discussion and Limitations

5.1. Architectural Trade-offs and Energy Implications

Our findings underscore key architectural trade-offs in mobile indoor navigation, particularly around computation placement. Offloading localization to a remote server, as in NaVIP, preserves accuracy while significantly reducing battery usage. This client-server split avoids on-device pose regression and path planning, supporting longer usage—especially important in accessibility contexts. This design reflects a broader shift toward lightweight clients that stream sensor data to cloud-based AI models

for inference. In contrast, ARKit-based solutions offer decent localization with low latency and no infrastructure needs, but perform all computation—vision tracking, sensor fusion, and 3D mapping—on-device. This leads to faster battery drain and occasional thermal issues during prolonged use, limiting their viability for continuous assistive navigation. We also observe that pairing ARKit with IMU sensors further increases energy consumption due to real-time sensor fusion. In comparison, NaVIP’s video streaming frontend remains relatively efficient—even with live rendering—making it well-suited for vision-based, client-light localization. These results highlight that energy efficiency depends not only on the localization algorithm but also on system architecture. As mobile apps adopt larger models and complex pipelines, offloading will likely be key to ensuring performance and usability.

5.2. Limitations and Future Work

This study has several limitations. First, evaluations were conducted at different sites: NaVIP in the HST building and GoodMaps at Newark Airport. Although route length, hardware, and usage time were controlled, environmental differences may introduce noise. These variations don’t inherently favor one system but stress the need for standardized benchmarking environments. Second, we relied on app-level energy profiling, which, while informative, lacks the precision of system-level instrumentation. Deeper profiling wasn’t feasible due to proprietary restrictions in commercial apps like GoodMaps. Third, we measured battery usage over fixed durations rather than per unit of localization accuracy. This was due to the absence of ground-truth pose data in real-world settings. We assume both systems reliably navigate users from start to destination. For detailed accuracy benchmarking, we refer to our companion work [18] using pseudo ground truth in curated datasets. Lastly, there’s a need for long-term, in-situ studies involving diverse users. These would capture real-world variation in usage patterns and device behavior. However, such efforts are limited by the site-specific nature of most systems and lack of access to multiple commercial platforms. Future work should pursue cross-app collaborations to enable large-scale evaluations of energy, accuracy, usability, and user satisfaction in realistic deployments.

6. Conclusion

This paper presents the first empirical analysis of battery consumption across key indoor localization technologies—CV-based, IMU-based, and ARKit-based—evaluated within a unified testbed using NaVIP and GoodMaps. By conducting both system-level and module-level profiling under standardized conditions, we demonstrate that localization accuracy alone is not a sufficient metric for real-world deployment in accessibility contexts; energy efficiency must also be prioritized. Our results show that ARKit, while providing reliable on-device localization, imposes a substantial energy cost that may hinder its practicality for long-term use. In contrast, the proposed NaVIP’s server-assisted architecture enables energy-efficient navigation without sacrificing localization fidelity. The ability to maintain functionality while reducing device-side computation aligns with broader trends in edge-cloud computing and the integration of foundation models in mobile applications. These findings point toward a design path for future accessible navigation tools: leveraging lightweight client interfaces paired with backend intelligence to deliver high performance with low energy consumption. Such architectures are well-suited to support visually impaired users who rely on continuous, dependable assistance throughout their indoor journeys.

Acknowledgments

This work was partially supported by the U.S. National Science Foundation through award #2345057.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT-4 in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and

take(s) full responsibility for the publication's content.

References

- [1] W. C. Simões, G. S. Machado, A. M. Sales, M. M. de Lucena, N. Jazdi, V. F. de Lucena Jr, A review of technologies and techniques for indoor navigation systems for the visually impaired, *Sensors* 20 (2020) 3935.
- [2] K. Hasan, N. Tom, M. R. Yuce, Navigating battery choices in iot: An extensive survey of technologies and their applications, *Batteries* 9 (2023) 580.
- [3] S. Winter, M. Tomko, M. Vasardani, K.-F. Richter, K. Khoshelham, M. Kalantari, Infrastructure-independent indoor localization and navigation, *ACM CSUR* 52 (2019) 1–24.
- [4] J. Yu, Y. Yang, V. Namboodiri, Navip: A low-cost, infrastructure-free indoor navigation solution for visually impaired persons, in: *ICCV Workshop on ACVR*, Honolulu, USA, 2025.
- [5] A. Kendall, M. Grimes, R. Cipolla, PoseNet: A convolutional network for real-time 6-dof camera relocalization, in: *Proceedings of the IEEE ICCV*, 2015, pp. 2938–2946.
- [6] A. Yang, M. Beheshti, T. E. Hudson, R. Vedanthan, W. Riewpaiboon, P. Mongkolwat, C. Feng, J.-R. Rizzo, Unav: An infrastructure-independent vision-based navigation system for people with blindness and low vision, *Sensors* 22 (2022) 8894.
- [7] C. Chen, Y. Chen, J. Zhu, C. Jiang, J. Jia, Y. Bo, X. Liu, H. Dai, E. Puttonen, J. Hyypä, An up-view visual-based indoor positioning method via deep learning, *Remote Sensing* 16 (2024) 1024.
- [8] M. Mahmoud, Z. Zhao, W. Chen, M. Adham, Y. Li, Automated scan-to-bim: A deep learning-based framework for indoor environments with complex furniture elements, *Journal of Building Engineering* 106 (2025) 112596.
- [9] M. H. Abidi, A. N. Siddiquee, H. Alkhalefah, V. Srivastava, A comprehensive review of navigation systems for visually impaired individuals, *Heliyon* (2024).
- [10] S. Wu, X. Zeng, M. Zhang, K. Cumanan, A. Waraiet, Z. Chu, K. Xu, Lcvae-cnn: Indoor wi-fi fingerprinting cnn positioning method based on lcvae, *IEEE Internet of Things Journal* (2025).
- [11] S. A. Cheraghi, A. Almadan, V. Namboodiri, Cityguide:a seamless indoor-outdoor wayfinding system for people with vision impairments, in: *In Proceedings of the 21st International ACM ASSETS*, Poster Session, 2019.
- [12] R. Cervenak, P. Masek, Arkit as indoor positioning system, in: *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, IEEE, 2019, pp. 1–5.
- [13] C. H. Tsai, F. Elyasi, P. Ren, R. Manduchi, All the way there and back: Inertial-based, phone-in-pocket indoor wayfinding and backtracking apps for blind travelers, *ACM transactions on accessible computing* 17 (2024) 1–35.
- [14] V. Nair, C. Tsangouri, B. Xiao, G. Olmschenk, W. H. Seiple, Z. Zhu, A hybrid indoor positioning system for blind and visually impaired using bluetooth and google tango, *Journal on Technology and Persons with Disabilities* 6 (2018).
- [15] V. Namboodiri, Towards accessible and inclusive navigation and wayfinding, in: *Workshop on Hacking Blind Navigation at The ACM CHI Conference on HFCS*, 2019.
- [16] S. A. Cheraghi, V. Namboodiri, G. Aarsal, Cityguide: A seamless indoor-outdoor wayfinding system for people with vision impairments, in: *Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, IEEE, 2021, pp. 105–110.
- [17] D. Sato, U. Oh, K. Naito, H. Takagi, K. Kitani, C. Asakawa, Navcog3: An evaluation of a smartphone-based blind indoor navigation assistant with semantic features in a large-scale environment, in: *Proceedings of the 19th International ACM ASSETS*, ASSETS '17, 2017, pp. 270–279.
- [18] J. Yu, Y. Yang, V. Namboodiri, Large-scale benchmarking of vision-based indoor localization using absolute pose regression, in: *2025 15th IPIN*, IEEE, 2025, pp. 1–7.