ONLINE FIRST February 10, 2026

Computer Assisted Methods in Engineering and Science
***, https://doi.org/10.24423/cames.2025.1966

Combining Indoor and Outdoor Positioning for Navigation in AR Environments

Krzysztof Skabek*, Dominika Rola, Wojciech Zamarski

Faculty of Computer Science and Mathematics, Cracow University of Technology, Kraków, Poland

* *Corresponding Author: krzysztof.skabek@pk.edu.pl*

This article presents a comparative analysis of augmented reality (AR) technologies – Vuforia, Immersal, MultiSet, and the ARCore Geospatial Application Programming Interface (API) – in terms of performance, accuracy, and interference tolerance for indoor and outdoor positioning and navigation. Two test environments were used: an indoor (laboratory) setup enabling detailed module testing, and a hybrid deployment on the Cracow University of Technology (CUT) campus to illustrate the feasibility of AR navigation in diverse environmental conditions. The research was conducted according to six scenarios. One involved outdoor GPS navigation, while the others concerned indoor navigation. Based on the measurements, recommendations are provided for selecting AR localization platforms for mixed navigation. As part of the detailed testing, an AR navigation system was implemented on the CUT campus as a combination of indoor and outdoor approaches. The final implementation was developed in the Unity environment. Software tests were conducted with particular emphasis on transitions between indoor and outdoor navigation.

Keywords: augmented reality (AR), 3D reconstruction, photogrammetry, LiDAR (light detection and ranging), indoor and outdoor positioning, geolocalization.

1. Introduction

Accurate positioning is a fundamental requirement for compelling augmented reality (AR) experiences, as it enables virtual objects to be meaningfully integrated with the physical environment. Depending on the context – indoors or outdoors – AR systems employ a variety of localization techniques. Indoor positioning relies on technologies such as visual SLAM (simultaneous localization and mapping) [16, 18], fiducial markers, Wi-Fi [22–24, 44], Bluetooth beacons [26, 27], and UWB (ultra-wideband) [28], while outdoor positioning is primarily based on satellite navigation systems such as GPS (global positioning system), often enhanced with sensor fusion or geospatial visual data. Recent advancements have focused on hybrid solutions that leverage the strengths of multiple technologies, aiming to achieve higher accuracy, robustness, and a seamless user experience across diverse environments [12, 38–40].

Accurate positioning technologies are key enablers for a wide range of AR applications. From indoor navigation in complex environments like airports and museums [1], to outdoor experiences in tourism, urban planning [2], and gaming [3], the ability to precisely locate and anchor virtual objects unlocks new forms of interaction. Positioning in AR also supports innovative solutions in fields such as industrial maintenance [4], medical guidance and training [5, 6], as well as education, marketing, and emergency response, where real-time spatial information enhances both user engagement and operational efficiency. As AR positioning methods continue to advance, they open the door to increasingly immersive and practical experiences across diverse sectors.

Contemporary AR-based navigation solutions transfer the experience of human interaction with the real environment into the virtual world. Inspiring examples include Live View [36] in Google Maps, Sygic GPS Navigation [45], and many others [46–48]. Despite achievements in AR navigation, the Cracow University of Technology campus lacks a dedicated navigation system based on this technology. Therefore, this article focuses on developing an AR mobile application that supports users in finding their way to selected destinations on the campus and within its buildings.

The pre-development stage of the AR navigation system involved a comparative analysis and selection of AR development platforms, with particular emphasis on Vuforia, Immersal, MultiSet, and Geospatial API. Considerable attention was devoted to assessing the suitability of these systems for developing navigation applications by examining their performance, accuracy, and interference tolerance. Experimental studies were conducted in both controlled indoor and outdoor environments to provide a comprehensive understanding of the capabilities and limitations of the tested solutions.

The novelty of this work lies in an end-to-end, scenario-based comparison of four practical AR localization solutions (Vuforia, Immersal, MultiSet, and ARCore Geospatial API) under a single, unified evaluation app and consistent criteria spanning performance, accuracy, and robustness to interference. In addition, we demonstrate a hybrid campus-scale AR navigation prototype that performs automatic indoor–outdoor switching (Area Target vs. Geospatial anchoring) while maintaining a consistent navigation layer (navigation mesh (NavMesh) and points of interest (POIs)) and user experience. The resulting recommendations are grounded in repeatable measurements across six scenarios in both controlled and real-world environments.

2. Software for AR localization

Several commercial and research-driven software platforms facilitate AR positioning in both indoor and outdoor scenarios. Vuforia is a widely adopted AR software development kit (SDK) that utilizes marker-based and markerless tracking for robust object registration in varying environments [7, 9, 10]. Immersal provides solutions for large-scale indoor AR experiences, integrating visual positioning and sensor fusion for precise localization in enterprise settings [8]. MultiSet offers advanced multi-sensor data integration for spatial tracking, supporting diverse use cases from interactive exhibitions to industrial applications [13]. These tools represent the state-of-the-art in enabling reliable AR positioning and serve as foundations for many current and emerging AR applications. Other commercial VPS (visual positioning system) options include Lightship VPS by Niantic [34].

2.1. Vuforia

Vuforia Engine developed by Parametric Technology Corporation (PTC) is a mature AR platform for Android, iOS, the Universal Windows Platform (UWP), Unity, and headsets (HoloLens 2, Magic Leap 2). It offers robust tracking and recovery after brief target loss, supporting seamless AR experiences. A key feature in this study is Area Targets, which use full 3D scans (e.g., via Vuforia Creator and LiDAR (light detection and ranging) devices) to localize the camera within indoor spaces [10]. This turns interiors into spatial references for placing AR content. Area Targets are intended for indoor use; outdoor performance is limited by lighting and occlusions. Other tracking modes include: Image Targets – 2D images as anchors, Model Targets – 3D object detection by geometry, VuMarks – custom markers encoding data and *Ground Plane* – horizontal surface detection. Vuforia supports multiple simultaneous targets and many devices. Main limitations include: no native GPS/geolocation for outdoor navigation, performance depends on sensor quality and target preparation, and the free license limits the number of trackables.

2.2. Immersal

The Immersal SDK provides markerless spatial mapping and 6 degrees of freedom (DoF) localization via point cloud maps [37], with reported centimeter-level precision [16, 17]. Mapping works on ARKit/ARCore smartphones or with 360° cameras or LiDAR capture; spatial data can be exported as meshes or embedded for offline use. Localization runs via the cloud (online) or using on-device maps (offline). Immersal has been used in both indoor and outdoor environments (e.g., malls, industry, campuses). Multiple datasets can be merged to support building- or campus-scale navigation. Accuracy depends on careful image acquisition and environmental stability (lighting, motion), but deployment is simplified by using standard mobile cameras. Unity integration and sample scenes speed up prototyping. A free non-commercial tier (e.g., map/image limits, branding) is available, while larger projects require paid plans.

2.3. MultiSet

MultiSet AI is a VPS using deep learning and high-resolution 3D mapping for 6DoF localization with centimeter-level precision. It targets scales from rooms to large facilities and processes LiDAR scans into vector representations, enabling robust localization. Computation is primarily cloud-based, requiring internet connectivity. A dedicated iOS app (LiDAR) generates 3D meshes for object placement and occlusion; multiple datasets can be fused for complex sites. The system is suitable for indoor navigation and industrial scenarios, including visualization of building information modeling (BIM) and Internet of Things (IoT) data. A basic free plan (with limits) supports commercial use, while extended plans cater to enterprise deployments.

2.4. Geospatial API

Google's ARCore Geospatial API [11, 36] enables global-scale AR by combining GPS, device sensors, and Google's visual positioning. Developers place anchors using WGS84 (latitude, longitude, altitude) on outdoor surfaces. SDKs support Android/iOS and Unity (via AR Foundation, ARCore Extensions) with localization performed via the cloud. The API refines GPS using visual matches to Street View imagery [17, 20, 40], achieving accuracy within tens of centimeters under favorable conditions [15, 21]. ARCore adds SLAM tracking, plane detection, lighting estimation, depth-based occlusion [41], and Cloud Anchors for multiuser experiences. Limitations include: unsuitability for indoor environments (GPS degradation, lack of Street View coverage) and performance depends on imagery quality, lighting, and weather. Access is free within quotas, making it effective for wide-area outdoor AR applications (navigation, tourism, urban information).

2.5. Selecting the solutions

The examined AR platforms differ in their underlying positioning technologies, environmental adaptability, and deployment workflows, which influences their applicability in both indoor and outdoor AR scenarios.

Vuforia is a well-established AR engine offering multiple tracking modes, including Area Targets and Model Targets, based on preprocessed images and

3D scans. It provides robust indoor localization on a broad range of devices but lacks support for GPS-based positioning and is not officially intended for outdoor use. Its performance depends heavily on the quality of input assets and camera sensors, and the free license imposes restrictions on the number of supported targets.

Immersal utilizes a markerless VPS that builds point cloud maps from captured images, supporting centimeter-level accuracy in both indoor and outdoor environments. It allows for online or offline localization, and map creation can be performed using mobile devices or dedicated scanners. Its flexibility, combined with wide hardware compatibility, makes it suitable for large-scale spatial mapping without specialized equipment.

MultiSet also operates on visual positioning principles but enhances spatial understanding through deep learning and vectorized 3D representations. It offers high accuracy and contextual scene analysis, although it currently supports only iOS devices with LiDAR for map generation. Localization requires active internet connectivity, and cloud-based processing is central to its operation.

Geospatial API combines GPS, sensor data, and Google's visual localization to provide outdoor positioning without prior mapping. It supports large-scale deployment and enables fast prototyping, though performance is dependent on environmental conditions and the availability of Street View imagery. The system does not support indoor use and requires a constant internet connection.

All platforms offer integration with Unity and provide tools for 3D content alignment, yet they differ in licensing terms, mapping requirements, and environmental robustness. Immersal and MultiSet offer detailed localization in controlled or complex spaces, while Geospatial API excels in rapid deployment across urban outdoor environments. Vuforia remains a reliable solution for structured indoor contexts where predefined assets are available. It is worth noting that some previously popular services have been discontinued (e.g., Azure Spatial Anchors was retired in 2024) [35].

3. Preparing the environment

3.1. Unity and libraries

To develop an AR application enabling the testing of positioning technologies in both indoor and outdoor environments, the Unity engine was used – one of the most widely adopted platforms for creating immersive applications.

The AR functionality was implemented using the AR Foundation package [19], which provides a unified interface for the native ARCore (Android) and

ARKit (iOS) libraries. The project also integrated the following libraries and extensions:

- ARCore extensions – enabling the use of features such as Geospatial API and Cloud Anchors,
- ARKit plugin – providing full support for devices running iOS,
- ARCore – supporting Android devices and offering core tracking and localization functionality in AR environments,
- Vuforia Engine – the project utilized the Area Target feature, allowing for scanning and subsequent recognition of physical spaces based on previously generated 3D models. This solution enabled precise user positioning within known environments, even in the absence of a GPS signal,
- Immersal SDK – leveraging visual SLAM mechanisms for localization and spatial mapping in both indoor and outdoor environments,
- MultiSet plugin – a cloud-based AR positioning system that utilizes LiDAR scans and deep learning to enable precise 6DoF localization in complex indoor environments.

The source code was developed using the Visual Studio 2022 development environment. Builds were prepared for both Android and iOS platforms, and the application was tested on mobile devices.

The application configuration also included appropriate system permissions, such as access to GPS and inertial sensors. When enabling advanced Wi-Fi RTT (Wi-Fi round-trip time) features, privacy-preserving approaches should be considered [25].

This development environment enabled efficient integration and comparison of various positioning solutions in the context of AR applications, while maintaining high cross-platform compatibility and operational performance.

4. Positioning accuracy

All experiments were conducted using a unified testing app built in Unity. The primary indoor testing site was a controlled $3\,\mathrm{m} \times 3\,\mathrm{m}$ room with standardized lighting and reference markers, illustrated as in Fig. 1. Outdoor measurements were conducted on a paved square at the university campus.

To prepare the AR models, 3D scans (or photos for the Immersal platform) were created using dedicated applications (Vuforia Creator, Immersal Mapper, MultiSet) on an iPhone 15 Pro 128GB, which is equipped with a LiDAR sensor. The Unity environment views of the prepared scenes for each platform are shown in Fig. 2. At this stage of development, the position of each cube is correct, meaning that the base of each object aligns with the drawn reference square.

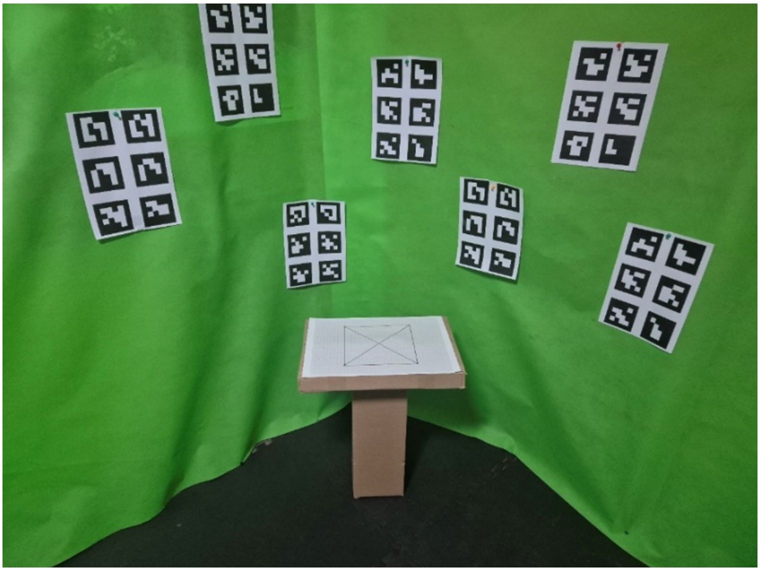


FIG. 1. Testing room.

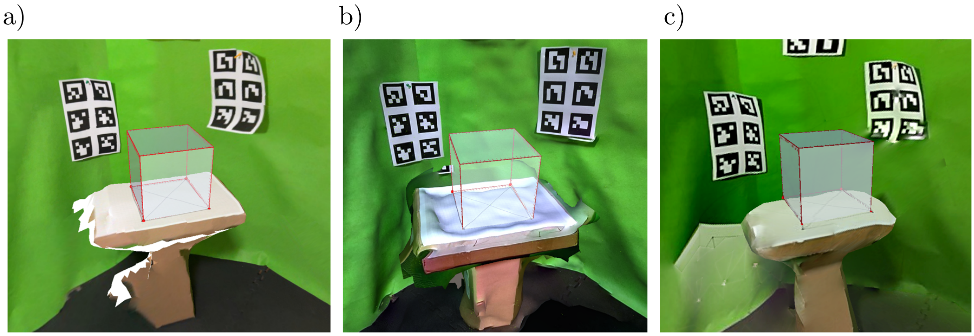
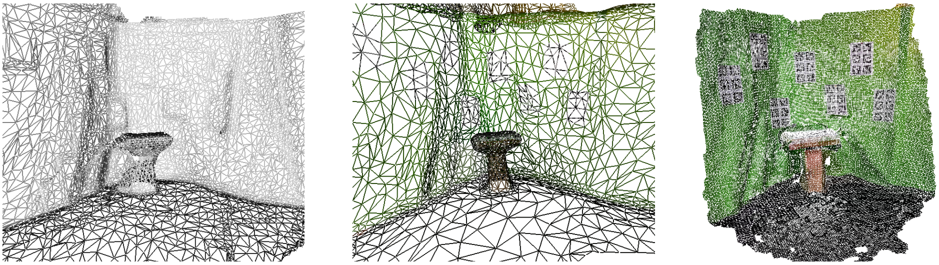


FIG. 2. Views from Unity showing the placement of the virtual cube in the mapped space for the AR platforms: a) Vuforia, b) Immersal, c) MultiSet.

4.1. Initial model accuracy

At first, we compared the accuracy of three mesh models prepared for embedding virtual objects in augmented environments: Vuforia, Immersal and MultiSet. Vuforia and MultiSet use LiDAR range measurements to obtain the initial model of the surrounding area. Immersal, on the other hand, establishes the model based on photogrammetry [37]. The numbers of vertices obtained for our indoor scene are indicated in Fig. 3. It appears that the densest model was created using photogrammetry. In the case of LiDAR measurements, the initial point clouds are more regular and dense, but the final mesh representation stored in the AR application is optimized [31–33].



Vuforia 6346 vertices | MultiSet 3455 vertices | Immersal 24 740 vertices

Fig. 3. Initial meshes with their numbers of vertices.

4.2. Comparison criteria

To assess the positioning capabilities of the AR technologies tested (Vuforia, Immersal, MultiSet, and Geospatial API), a comprehensive set of evaluation criteria was established and grouped into three categories: performance, accuracy, and resistance. Most criteria were quantitative, except for two qualitative ones: the effect of colored lighting (3.2) and tolerance to projected patterns (3.3).

1. Performance:
   - 1.1. Scene recognition time: time (in seconds) from app launch to the initial appearance of an AR object.
   - 1.2. Battery consumption: average percentage drop in battery level per minute of continuous use.
   - 1.3. Memory usage: random-access memory (RAM) usage measured with the PSS (proportional set size) metric.
2. Accuracy:
   - 2.1. Indoor initial positioning error: mean distance between AR and real-world marker vertices in a 20 cm reference square.
   - 2.2. Outdoor positioning error: same as 2.1, measured against a 150 cm reference square in an outdoor environment.
   - 2.3. Extended tracking error: positioning error of a secondary virtual cube located 5, 10, and 30 meters from the originally mapped scene. This criterion evaluates the system's ability to maintain accurate spatial tracking despite moving through unmapped areas or losing visual contact with the reference scene.
   - 2.4. Positional drift over time: maximum deviation recorded during a 10-minute session of continuous AR tracking in a static position.
3. Resistance:
   - 3.1. Low-light threshold: maximum percentage reduction in brightness at which the AR system still functions.

3.2. Colored lighting impact: binary value indicating whether AR initialization succeeds under colored red–green–blue (RGB) lighting conditions.

3.3. Pattern robustness: binary value indicating whether the AR system can recognize scenes under projected high-contrast black-and-white patterns.

4.3. Measurement methodology

Six testing scenarios were defined:

Scenario I: measured scene recognition time, indoor positioning error, and positional drift. The smartphone was placed at three distinct positions (A, B, and C), each representing a different level of marker visibility and spatial challenge:

- position A: all reference markers were fully visible, creating optimal tracking conditions,
- position B: approximately half of the markers were occluded, representing moderate difficulty for AR tracking,
- position C: only a few markers were visible, presenting a minimal-information scenario and testing the boundary of reliable tracking.

Scenario II: evaluated battery and memory consumption over a 30-minute session of uninterrupted app operation. Data were logged using diagnostic scripts.

Scenario III: assessed extended tracking accuracy by placing a secondary AR cube at 5, 10, and 30 meters from the original scene. Users moved through these locations while maintaining line-of-sight camera input. The error was calculated relative to a reference square at each distance. In scenario III, extended tracking across 5 m, 10 m, and 30 m relies on persistent visual localization and place recognition [17] to mitigate drift outside the originally mapped area.

Scenario IV: determined the minimum ambient light required for successful recognition by incrementally increasing brightness from total darkness.

Scenario V: evaluated resistance under RGB lighting and projected black-and-white patterns. Recognition success under altered visuals was recorded.

Scenario VI: measured outdoor accuracy using a 150 cm square on the pavement. Positioning error was calculated after stabilization of AR tracking during walking.

Tests were conducted on a Samsung Galaxy S24 (Exynos 2400, 8 GB RAM, Android 14). Each measurement was repeated ten times. The device was mounted on a tripod for all tests except those requiring user movement.

4.4. Performance

Performance was evaluated in terms of scene recognition time (scenario I), battery consumption, and RAM usage (scenario II). In the recognition speed test, both Vuforia and Immersal demonstrated rapid initialization across all positions (A–C), with mean detection times ranging from 1.65 to 2.51 seconds. Vuforia was particularly consistent across conditions, showing minimal variance even under reduced marker visibility. In contrast, MultiSet exhibited a considerably higher and less stable recognition times, peaking at 9.90 ±1.30 seconds in position A, and only slightly improving in position C (see Table 1).

Table 1. Scene recognition time in seconds in scenario I.

Position	Vuforia		Immersal		MultiSet	
	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
A	1.69	0.36	1.65	0.31	9.90	1.30
B	1.66	0.30	1.65	0.29	9.66	2.28
C	1.87	0.75	2.51	0.29	7.72	0.39

Battery and memory efficiency were assessed over a 30-minute continuous operation (scenario II). Vuforia showed the highest battery consumption at 0.45 %/min, while Immersal consumed slightly less (0.433 %/min) but used the most RAM with 717 MB PSS. MultiSet was the most resource-efficient, consuming just 0.367 %/min of battery and 458 MB of memory, making it a better choice for power-constrained or long-duration mobile AR tasks.

4.5. Accuracy

Accuracy testing spanned several scenarios and criteria, including indoor and outdoor initial positioning, tracking drift, and extended tracking at various distances. In scenario I (initial indoor positioning), Vuforia and Immersal delivered stable results with positioning errors generally below 4 cm. MultiSet matched this performance in optimal conditions (positions A and B) but deteriorated severely in position C, where error rose dramatically to over 61 cm, indicating poor robustness to reduced feature visibility (see Table 2).

Table 2. Initial AR positioning error in indoor scenario [cm].

Position	Vuforia		Immersal		MultiSet	
	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
A	3.55	0.57	3.35	1.88	2.84	1.45
B	2.57	0.29	1.21	0.43	2.84	1.66
C	3.61	0.24	1.43	0.92	61.69	3.46

In extended tracking (scenario III), where AR targets were located at 5 m, 10 m, and 30 m away from the origin, Vuforia yielded the lowest errors at short distances, while Immersal performed better at 30 m, suggesting better spatial persistence at scale. MultiSet showed relatively high and fluctuating errors across all distances. These observations align with recent advances in VIO (visual-inertial odometry) robustness and event-based fusion for odometry [29, 30]. These results are also in line with patterns observed on public benchmarks for VO (visual odometry)/SLAM [42, 43].

In outdoor testing (scenario VI), Immersal again led with 9.4 cm average error. Vuforia and MultiSet followed with 16.2 cm and 19.1 cm, respectively. Geospatial API performed poorly due to GPS limitations, with average error exceeding 3.2 meters [15, 21].

Drift testing over a 10-minute period confirmed the stability of Vuforia and Immersal (maximum drift below 12 cm), while MultiSet's error in position C reached nearly 89 cm, reinforcing earlier findings of instability in complex environments.

4.6. Resistance

The resistance of AR systems to environmental challenges was tested in low-light conditions (scenario IV) and under visual interference (scenario V). In reduced lighting, Immersal retained functional tracking down a 92 % luminance reduction, outperforming Vuforia (90 %) and MultiSet (82 %), indicating better sensor robustness and image processing under poor visibility (see Table 3).

TABLE 3. Maximum luminance reduction tolerated before failure.

Platform	Vuforia		Immersal		MultiSet	
Reduction [%]	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
	90	0.89	92	1.64	82	6.46

Under RGB lighting, all systems functioned correctly, demonstrating color-independence. However, in the presence of high-contrast projected patterns, only Vuforia was able to consistently initialize and maintain AR tracking. Immersal and MultiSet failed under three of the four tested patterns, exposing a lower tolerance to structured visual noise.

5. Implementation of AR navigation system

The presented application was developed as part of a master's thesis [14] with the goal of testing and evaluating the effectiveness of AR-based positioning systems in real-world educational settings. The prototype serves as a proof of concept for a system that assists users in locating specific destinations across the Cracow University of Technology (CUT) campus, combining indoor and outdoor localization methods in a single interface.

The system was implemented using the Unity game engine and integrates two complementary AR technologies. For indoor spaces, the Vuforia Engine was used, leveraging 3D scans of building interiors to provide accurate camera tracking and virtual content placement. For outdoor areas, the application employs the Google Geospatial API, which uses a combination of GPS and visual localization through Street View data to estimate device position on a global scale.

Figure 4 summarizes the runtime workflow in three cooperating layers. First, based on the current user position in the localization system, the appropriate tracking method is selected: indoor via Vuforia Area Target or outdoor via Geospatial API. Once localization is successful, the user position is passed to the navigation system to compute the route on a navigation mesh (NavMesh). Then, the user moves to another localization following turn-by-turn guidance

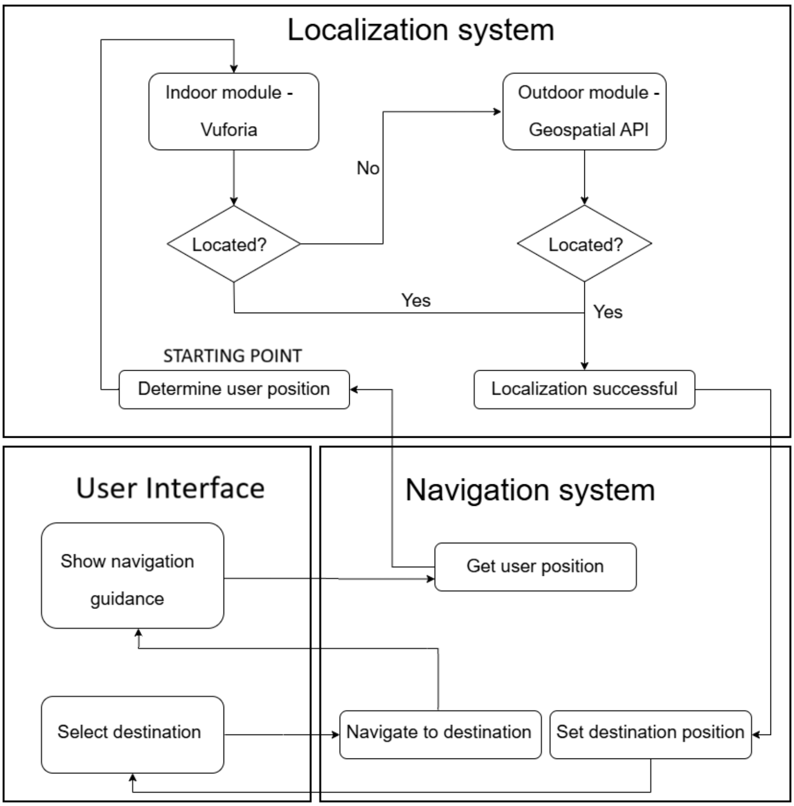


Fig. 4. System architecture and data flow.

provided through the user interface (UI). This separation simplifies automatic indoor/outdoor mode switching and keeps a consistent user experience regardless of the underlying localization method.

Outdoor positioning, illustrated in Fig. 5, exhibited lower precision. Virtual objects were occasionally offset from their intended locations, which affected the clarity of spatial feedback. This behavior was most noticeable in areas with limited satellite visibility or outdated visual data. Nevertheless, the system remained functional and responsive, providing general orientation cues.

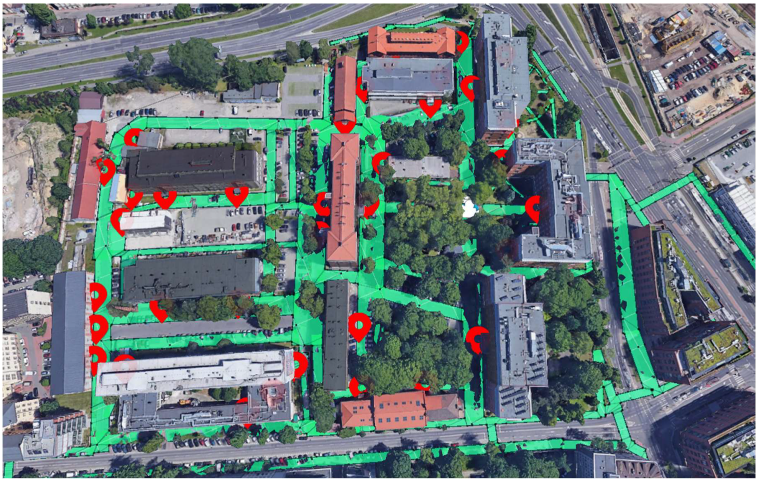


Fig. 5. Outdoor navigation map: NavMesh (green) and POIs (red markers) on the campus grounds.

To make navigation possible, a NavMesh (Unity AI navigation) was prepared to cover continuous, walkable surfaces–outdoors (walkways, plazas) and indoors (corridors, stairwells). On top of this navigation layer, POIs are placed as Unity objects used as destinations and reference points. Outdoors, each POI is georeferenced with WGS84 (World Geodetic System 1984) coordinates (latitude/longitude/altitude) and anchored as an ARCore Geospatial Anchor (with heading). Indoors, POIs are defined in the local frame of the Vuforia Area Target scan (3D coordinates in model space). Mode switching is automatic: when stable Area Target tracking is present, the app enables Indoor mode (geospatial anchors are disabled and cleared); otherwise, if geolocation conditions are met, Outdoor mode is activated. Examples of the outdoor and indoor configurations of the NavMesh and POIs are shown in Figs. 5 and 6, respectively.

In practical use, the application automatically switches between indoor and outdoor localization depending on the user's current context. This hybrid model enables continuous positioning across diverse spatial environments. Indoor localization was generally accurate and stable, especially in the scanned areas of

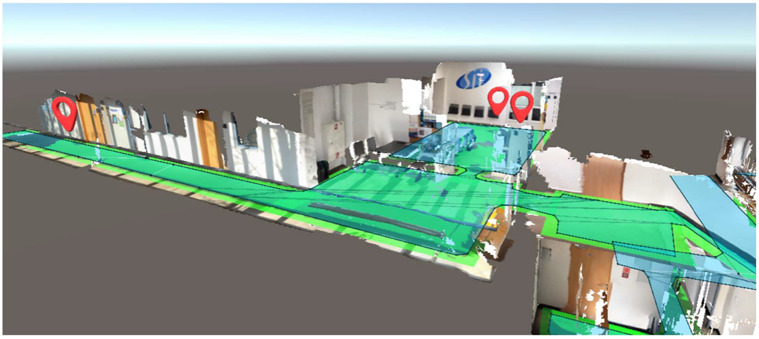


FIG. 6. Indoor navigation map: NavMesh (green) and POIs (red markers) inside the building (Area Target).

the WIiTCH building. As shown in Fig. 7, AR elements such as arrows and lines were correctly positioned and maintained spatial consistency during use.

a) b)

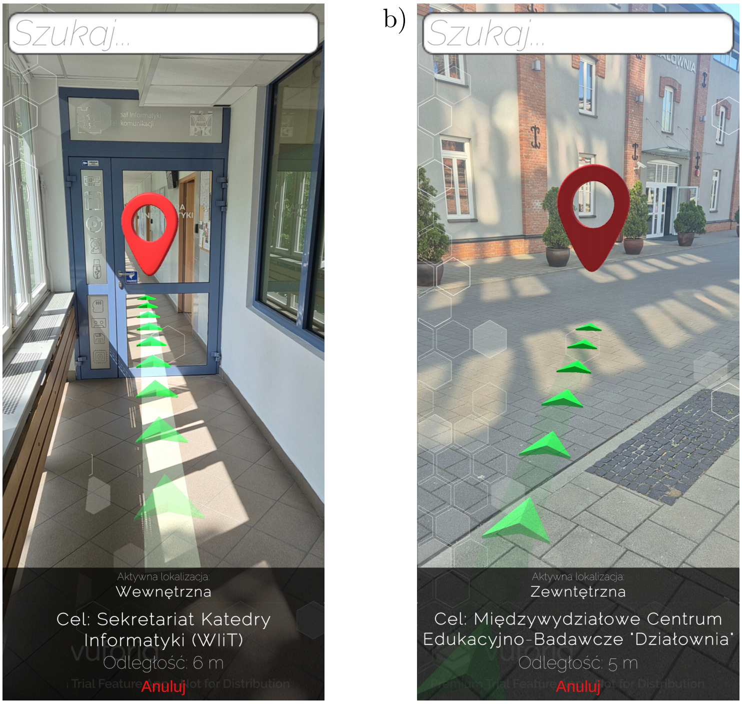


FIG. 7. Screenshots from our AR system: a) indoor navigation, b) outdoor navigation.

Overall, testing confirmed that the application operates in accordance with the initial design objectives. It provides a working demonstration of AR-based positioning across both indoor and outdoor environments using off-the-shelf technologies. While the system performs well in structured indoor spaces, its outdoor performance may benefit from the integration of alternative localization methods or the use of more detailed visual mapping. The current version offers a solid foundation for future development and practical deployment in academic or public environments.

6. Summary

In this work, we benchmarked multiple widely used AR localization technologies under unified scenarios and demonstrated a hybrid navigation pipeline with seamless indoor–outdoor transitions implemented in Unity. Two test environments were created as part of this work: first, an experimental environment, enabling detailed module testing, and the other, an implementation environment, realized on the Cracow University of Technology campus. Considerable attention was devoted to detailed testing of performance, accuracy, and resistance to interference, including six test scenarios. One of these scenarios involved outdoor GPS navigation, while the others involved indoor visual navigation.

The experiments demonstrated that the tested technologies are generally suitable for use in navigation applications based on AR. Regarding the quality of the initial mesh representation, there is a certain advantage of the LiDAR-based technology providing more regular mesh structures while photogrammetric technology allows for wider use on smartphones equipped only with cameras and in outdoor environments beyond the measurement range of LiDAR sensors.

Analyzing the resistance of the systems to adverse environmental conditions, it was revealed that Vuforia performed particularly well in the presence of visual interference in the form of structured lights. On the other hand, Immersal showed the greatest tolerance to low lighting levels, making this technology particularly attractive for night applications or in poorly lit indoor spaces. In terms of performance, there was a noticeable difference in the time of initializing space tracking, because MultiSet needed significantly more time to recognize the scene than competing solutions, while it had the lowest consumption of system resources. Battery consumption was relatively high for all tested technologies, which could negatively affect the comfort of practical use of AR applications.

The resulting AR application makes it possible to navigate in both indoor or outdoor environments, adapting its navigation routines based on available scope of data, which makes the navigation possible under all conditions considered in this study.

References

1. Alarifi M.A. *et al.*, Ultra wideband indoor positioning technologies: Analysis and recent advances, *Sensors*, **16**(5): 707, 2016, https://doi.org/10.3390/s16050707.

2. Reitmayr P., Schmalstieg D., Location Based Applications for Mobile Augmented Reality, [In:] *Proceedings of the Fourth Australasian User Interface Conference on User Interfaces 2003*, Vol. 18, pp. 65–73, Australian Computer Society, 2003.

3. Rauschnabel M., Rossmann A., tom Dieck T., An adoption framework for mobile augmented reality games: The case of Pokémon GO, *Computers in Human Behavior*, **76**: 276–286, 2017, https://doi.org/10.1016/j.chb.2017.07.030.

4. Nee A.Y.C., Ong S.K., Chryssolouris G., Mourtzis D., Augmented reality applications in design and manufacturing, *CIRP Annals*, **61**(2): 657–679, 2012, https://doi.org/10.1016/j.cirp.2012.05.010.

5. Vávra M. *et al.*, Recent development of augmented reality in surgery: A review, *Journal of Healthcare Engineering*, **2017**: 4574172, 2017, https://doi.org/10.1155/2017/4574172.

6. Buttiglione M.D., Guerrera F., Piazzolla P., Colombo G., Ruffini E., Gribaudo M., Collaborative virtual reality framework for surgical training and simulation, [In:] *39th Proceedings of ECMS 2025*, Vol. 39, Iss. 1, pp. 661–667, Catania, Italy, 2025, https://doi.org/10.7148/2025-0661.

7. Billinghurst M., Clark A., Lee G., A survey of augmented reality, *Foundations and Trends in Human-Computer Interaction*, **8**(2–3): 73–272, 2015, https://doi.org/10.1561/1100000049.

8. Immersal, Immersal SDK Documentation, 2023, https://developers.immersal.com/docs/ [accessed: 29.05.2025].

9. PTC, Vuforia Engine Library, 2024, https://library.vuforia.com [accessed: 29.05.2025].

10. PTC, *Area Targets*, Vuforia Engine Library, 2021, https://library.vuforia.com/features/area-targets.html [accessed: 10.09.2025].

11. Google, ARCore Geospatial API Documentation, 2024, https://developers.google.com/ar [accessed: 29.05.2025].

12. Guo X., Ansari N., Hu F., Shao Y., Elikplim N.R., Li L., A survey on fusion-based indoor positioning, *IEEE Communications Surveys & Tutorials*, **22**(1): 566–594, 2020, https://doi.org/10.1109/COMST.2019.2951036.

13. MultiSet AI, MultiSet AI Developer Documentation, 2025, https://docs.multiset.ai [accessed: 29.05.2025].

14. Zamarski W., *Navigation methods in augmented reality on the example of the CUT campus*, Master Thesis, Cracow University of Technology, Cracow, 2025.

15. Sattler T. *et al.*, Benchmarking 6DOF outdoor visual localization in changing conditions, [In:] Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8601–8610, Salt Lake City, USA, 2018, https://doi.org/10.1109/CVPR.2018.00897.

16. Mur-Artal R., Tardós J.D., ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras, *IEEE Transactions on Robotics*, **33**(5): 1255–1262, 2017, https://doi.org/10.1109/TRO.2017.2705103.

17. Arandjelović R., Gronat P., Torii A., Pajdla T., Sivic J., NetVLAD: CNN architecture for weakly supervised place recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**(6): 1437–1451, 2018, https://doi.org/10.1109/TPAMI.2017.2711011.

18. Hess W., Kohler D., Rapp H., Andor D., Real-time loop closure in 2D LIDAR SLAM, [In:] *IEEE ICRA Workshop*, 2016, https://research.google.com/pubs/archive/45466.pdf.

19. Unity Technologies, AR Foundation Documentation, 2025, https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@latest/ [accessed: 10.09.2025].

20. Reinhardt T., Using Global Localization to Improve Navigation, *Google Research*, Feb 11, 2019, https://research.google/blog/using-global-localization-to-improve-navigation/ [accessed: 10.09.2025].

21. Hakamäki S., *Assessing Localization Accuracy of Google ARCore Geospatial API*, MSc Thesis, Tampere University, 2024, https://trepo.tuni.fi/bitstream/handle/10024/157627/Hakam%C3%A4kiSaku.pdf.

22. Qiao J., Yang F., Liu J., Huang G., Zhang W., Li M., Advancements in indoor precision positioning: A comprehensive survey of UWB and Wi-Fi RTT positioning, *Network*, **4**(4): 545–566, 2024, https://doi.org/10.3390/network4040027.

23. Kosek-Szott K., Szott S., Ciezobka W., Wojnar M., Rusek K., Segev J., Indoor positioning with Wi-Fi location: A survey of IEEE 802.11mc/az/bk fine timing measurement research, *Computer Communications*, **247**: 108400, 2026, https://doi.org/10.1016/j.comcom.2025.108400.

24. Hashem O., Harras K.A., Youssef M., Accurate indoor positioning using IEEE 802.11mc round trip time, *Pervasive and Mobile Computing*, **75**: 101416, 2021, https://doi.org/10.1016/j.pmcj.2021.101416.

25. Schepers D., Ranganathan A., Privacy-preserving positioning in Wi-Fi fine timing measurement, *Proceedings on Privacy Enhancing Technologies*, **2022**(2): 325–343, 2022, https://crysp.petsymposium.org/popets/2022/popets-2022-0048.pdf.

26. Bai L., Ciravegna F., Bond R., Mulvenna S., A low cost indoor positioning system Using Bluetooth Low Energy, *IEEE Access*, **8**: 136858–136871, 2020, https://doi.org/10.1109/ACCESS.2020.3012342.

27. Bencak P., Hercog D., Lerher T., Indoor positioning system based on Bluetooth Low Energy for intralogistics, *Electronics*, **11**(3): 308, 2022, https://doi.org/10.3390/electronics11030308.

28. Cho J., Jeong S., Kim J.-Y., Kim G.-H., Lee J., Lee B., Real-time indoor localization and safety applications using UWB, *KSCE Journal of Civil Engineering*, **29**(8): 100164, 2025, https://doi.org/10.1016/j.kscej.2025.100164.

29. Minervini A., Carrio A., Guglieri G., Enhancing Visual–Inertial Odometry robustness and accuracy in challenging environments, *Robotics*, **14**(6): 71, 2025, https://doi.org/10.3390/robotics14060071.

30. Zhang J., Yu X., Sier H., Zhang H., Westerlund T., Event-based sensor fusion and application on odometry: A survey, *arXiv*, 2024, https://arxiv.org/abs/2410.15480.

31. Teo T.-A., Yang C.-C., Evaluating the accuracy and quality of an iPad Pro's built-in LiDAR for BIM, *Developments in the Built Environment*, **14**: 100169, 2023, https://doi.org/10.1016/j.dibe.2023.100169.

32. Abdel-Majeed H.M., Shaker I.F., Abdel-Wahab A.M., Awad A.A.D.I., Indoor mapping accuracy comparison between Apple pro devises' LiDAR sensor and terrestrial laser scanner, *HBRC Journal*, **20**(1): 915–931, 2024, https://doi.org/10.1080/16874048.2024.2408839.

33. Sheshtar F.M., Alhatlani W.M., Moulden M., Kim J.H., Comparative analysis of LiDAR and photogrammetry for 3D crime scene reconstruction, *Applied Sciences*, **15**(3): 1085, 2025, https://doi.org/10.3390/app15031085.

34. Niantic, Visual Positioning System (VPS) Documentation, 2025, https://lightship.dev/docs/ardk/features/lightship_vps/ [accessed: 10.09.2025].

35. Microsoft, Azure Spatial Anchors – Product Lifecycle (Retired Nov 20, 2024), 2025, https://learn.microsoft.com/en-us/lifecycle/products/azure-spatial-anchors [accessed: 10.09.2025].

36. Google, Build global-ccale, immersive, location-based AR experiences with the ARCore Geospatial API, 2025. https://developers.google.com/ar/develop/geospatial [accessed: 10.09.2025].

37. Schönberger J.L., Frahm J.-M., Structure-from-motion revisited, [In:] *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016, https://openaccess.thecvf.com/content_cvpr_2016/papers/Schonberger_Structure-From-Motion_Revisited_CVPR_2016_paper.pdf.

38. DeTone D., Malisiewicz T., Rabinovich A., SuperPoint: Self-supervised interest point detection and description, [In:] *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 337–349, 2018, https://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w9/DeTone_SuperPoint_Self-Supervised_Interest_CVPR_2018_paper.pdf.

39. Sarlin P.-E., DeTone D., Malisiewicz T., Rabinovich A., SuperGlue: Learning feature matching with graph neural networks, [In:] *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4938–4947, 2020, https://openaccess.thecvf.com/content_CVPR_2020/papers/Sarlin_SuperGlue_Learning_Feature_Matching_With_Graph_Neural_Networks_CVPR_2020_paper.pdf

40. Sarlin P.-E., Cadena C., Siegwart R., Dymczyk M., From coarse to fine: Robust hierarchical localization at large scale, [In:] *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, 2019, https://openaccess.thecvf.com/content_CVPR_2019/papers/Sarlin_From_Coarse_to_Fine_Robust_Hierarchical_Localization_at_Large_Scale_CVPR_2019_paper.pdf.

41. Du R. *et al.*, DepthLab: Real-time 3D interaction with depth maps for mobile augmented reality, [In:] *UIST '20: Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pp. 829–843, 2020, https://doi.org/10.1145/3379337.3415881.

42. Sturm J., Engelhard N., Endres F., Burgard W., Cremers D., A Benchmark for the Evaluation of RGB-D SLAM Systems, [In:] *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573-580, Vilamoura-Algarve, Portugal, 2012, https://doi.org/10.1109/IROS.2012.6385773.

43. Geiger A., Lenz P., Urtasun R., Are we ready for autonomous driving? The KITTI vision benchmark suite, [In:] *Conference on Computer Vision and Pattern Recognition*, 2012, https://www.cvlibs.net/publications/Geiger2012CVPR.pdf.

44. IEEE, *IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 4: Enhancements for Positioning*, 2022, https://standards.ieee.org/ieee/802.11az/7226/.

45. Sygic, Real View Navigation, https://www.sygic.com/what-is/real-view-navigation.

46. Huang B.-C., Hsu J., Chu E.T.-H., Wu H.-M., ARBIN: Augmented Reality Based Indoor Navigation System, *Sensors*, **20**(20): 5890, 2020, https://doi.org/10.3390/s20205890.

47. Sato F., Indoor navigation system based on augmented reality markers, [In:] Barolli L., Enokido T. [Eds.], *Innovative Mobile and Internet Services in Ubiquitous Computing. IMIS 2017*, Advances in Intelligent Systems and Computing, Vol. 612, pp. 266–274, Springer, Cham, 2018, https://doi.org/10.1007/978-3-319-61542-4_25.

48. Jiang J.R., Subakti H., An indoor location-based augmented reality framework, *Sensors*, **23**(3): 1370, 2023, https://doi.org/10.3390/s23031370.

Received October 30, 2025; revised December 23, 2025; accepted December 23, 2025; available online December 23, 2025; version of record February 10, 2026.