XR-PALS: XR Tool for Loco Positioning System

Dominik Grzelak[0000−0001−6334−2356] and Victor Victor[0009−0005−2434−6318]

Chair of Software Technology, Dresden University of Technology, Germany
{dominik.grzelak,victor.victor}@tu-dresden.de

Abstract. Localization is crucial for the safe and reliable operation of mobile robots in dynamic indoor environments. The present study focuses on ultra-wideband (UWB)-based tracking systems, specifically utilizing the Loco Positioning System of the Crazyflie drone platform. Unlike other tracking technologies, UWB-based systems rely on precisely configured real-world positions for accurate operation. Commonly, laser-based measurement tools are used for this task; however, they require manual calculations, increasing wrongdoing and cognitive load. Therefore, we propose XR-PALS, a mixed reality application that provides an intuitive interface for managing these cyber-physical spaces. Our tool enables seamless synchronization between virtual and physical setups of tracking systems by integrating UWB modules mounted on movable tripods. A user study was conducted to evaluate the efficiency, usability, and accuracy of XR-PALS compared to a traditional laser-based approach in configuring UWB-based systems. Results indicate that XR-PALS significantly reduces task completion time, eliminates measurement errors, and enhances user confidence, particularly in complex spatial layouts.

Keywords: Extended reality · Mixed reality · Ultra-wideband positioning · Indoor positioning · Cyber-physical space · Loco Positioning System · Crazyflie

1 Introduction

Mobile robots are becoming increasingly essential for real-world applications in indoor environments, such as monitoring and inspection tasks in warehouse logistics. At the same time, the growing affordability of mobile robots has made them ideal for educational purposes, providing opportunities to explore programming concepts, such as trajectory planning and swarm behavior experimentation. In this context, numerous augmented reality (AR) and virtual reality (VR) studies focus on understanding and controlling robot collectives through virtual representations, for example, [9, 11, 14, 15, 20]. However, accurate positional awareness is crucial for developing reliable and safe robotic applications. Although an onboard sensor such as Inertial Measure Unit (IMU), which combines accelerometers, gyroscopes, and magnetometers, provides basic motion estimation, their accuracy degrades over a long time (see, for example, [18, 23]), rendering them insufficient for long-term positional tracking. Addressing this limitation requires integrating external systems capable of delivering accurate, real-time positioning information. In this regard, several positioning systems offer varying solutions, such as optically-based lighthouse systems, ultra-wideband (UWB)-based systems, and camera-based motion capture systems. Thus, with the growing interest in mixed reality (MR) and the need to configure such cyber-physical space (CPSp), simplifying the deployment process is vital. That is, designers and engineers require tools that simplify the creation and synchronization of virtual and physical environments of the CPSp.

Nevertheless, few studies have focused on such methods to effectively enhance user experience utilizing MR applications for mobile robotics and positioning systems. To address this gap, we introduce XR-PALS, a novel MR application for the dynamic deployment of the Loco Positioning System (LPS), a UWB-based positioning system designed for the Crazyflie drone platform. LPS not only enables accurate position detection of drones in indoor environments but also enhances the user experience in extended reality (XR) applications for robotics.

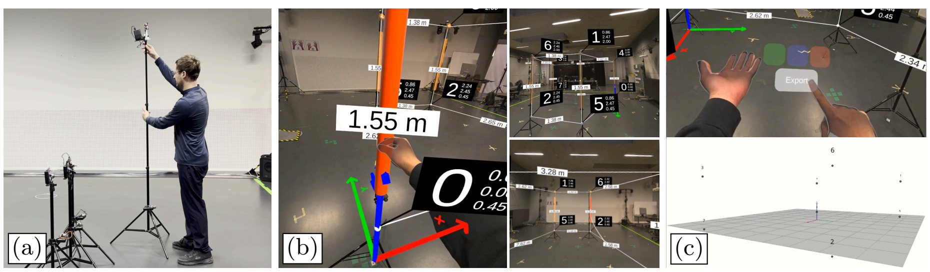

Fig. 1. Creation and synchronization of a CPSp. (a) The boundaries of the virtual environment can be designed before or after the physical setup. (b) The virtual overlay makes fine-tuning the placement of the tripod easy. (c) The virtual anchor positions are used to set the position data of the physical anchors by pressing the export button, effectively synchronizing both worlds.

Traditionally, deploying the LPS involves a static, fixed placement of Loco Positioning nodes or anchors within a room. This process involves manually measuring the anchor positions in space for accurate calibration, typically using tools such as stick, tape, or laser measurement devices. After measurement, these values must be manually entered into the configuration software provided by the Crazyflie platform (refer to Fig. 4b). This method is labor-intensive, particularly when adapting the system to new environments.

Contribution. To address the challenge of managing dynamic positioning systems, we introduce XR-PALS for the LPS from Bitcraze AB. XR-PALS employs a two-step process: first, the CPSp is virtually modeled, and second, it is physically deployed by synchronizing the physical positions of anchors with their virtual representations (refer to Fig. 1). Therefore, our approach employs anchors mounted on tripods for flexible repositioning within the environment. The 3D virtual overlays assist in aligning and synchronizing the virtual and physical positions of each anchor, effectively automating the transfer of the virtual anchor layout to its physical counterparts. This process eliminates the need for manual measurements by automatically updating the anchor positions remotely, significantly simplifying LPS deployment. As a result, a single operator can efficiently set up a cyber-physical layout with minimal effort. A demonstration video is available online.[1] That is, by employing structurally flexible anchors, XR-PALS enhances the efficiency of defining and deploying CPSp, paving the way for broader applications in robotics, education, and collaborative robotics. Additionally, XR-PALS is broadly adaptable to other UWB-based systems and mobile robot platforms, further demonstrating its versatility and potential impact.

Outline. Section 2 provides the context and rationale behind XR-PALS, outlining the challenges of deploying UWB-based positioning systems and the need for an intuitive, mixed-reality approach. Section 3 explores real-world applications of UWB in robotics. In Section 4, we present the design and implementation of XR-PALS, detailing its features, system architecture, and interoperability with existing localization frameworks. Section 5 describes the experimental setup and user study conducted to evaluate XR-PALS against a traditional laser-based method with respect to task completion time and measurement accuracy. Section 6 concludes the paper with a summary of key findings and potential directions for future work.

2 Context and Rationale of XR-PALS

XR-PALS forms a critical component of a broader approach to programming CPSp. Before deploying Cyber-Physical System (CPS) applications, it is necessary to first define and establish a CPSp. The primary focus of our work is enabling the seamless definition and deployment of a CPSp.

Creation of Cyber-Physical Spaces. CPSps integrate physical environments (e.g., rooms, warehouses, or labs) with digital systems, where sensors and actuators interface with computational models. The definition and design of such spaces involve considering physical constraints, communication infrastructure, and the interaction between hardware, software, and humans. But firstly, this involves creating a functional and logical representation of the environment where cyber (digital) and physical (real-world) systems coexist. This process involves a conceptual and technical dimension. Conceptually, it includes defining the boundaries of the space, mapping anchor locations, ensuring adequate coverage, and calibrating the system to align digital models with real-world coordinates.

[1] https://videocampus.sachsen.de/video/cyber-physical-spaces-xr-deployment-uwb-positioning-systems/f16fb2b1154ca24d7ffb1f4ecb6e0b70

Technically, deploying a positioning system such as the LPS entails preparing the anchors, configuring transmission power, determining optimal anchor placement, and aligning physical anchors with their virtual counterparts to ensure accurate spatial referencing for mobile robots.

Setup of UWB-Based Systems. Most UWB-based systems require fixed anchor positions to triangulate the location of moving robots. For instance, the Time Difference of Arrival (TDoA) system employed in our work depends on accurately positioned anchors to determine the location of a transmitting UWB module. Consequently, stationary setups are predominantly used to control robot applications. As noted in [19], most existing localization systems are constrained by limited coverage areas due to the reliance on stationary anchors. Additionally, in computer science education, positioning systems often need to be quickly set up and dismantled for short-term exercises or demonstrations. This highlights the need for dynamic positioning systems, which offer greater flexibility in such contexts. To emphasize, while related work has focused on improving UWB-based systems with stationary setups, we aim to explore a novel direction by investigating mobile UWB-based systems with structurally free, reconfigurable anchors. Given the dynamic nature of our proposed setup, manual measurement methods such as stick, tape, or laser measurement devices are no longer practical. Therefore, it is imperative to develop a method that simplifies and accelerates the deployment of UWB-based systems while ensuring accurate 3D coordinate synchronization. To the best of our knowledge, mobile UWB-based systems with dynamic anchors have not yet been thoroughly investigated (cf. also [19]). Additionally, no prior work has fully realized an XR tool to create CPSp and synchronize anchor positions in real time.

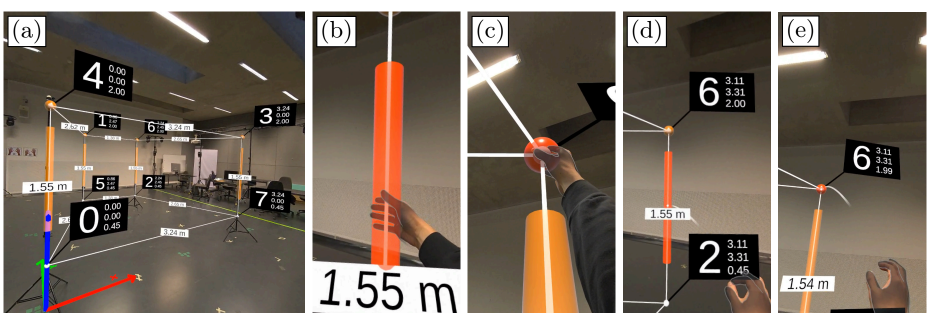

Fig. 2. Screenshots of some features of XR-PALS. (a) The 3D overlay shows each anchor ID and position, as well as the distance between anchors. (b)/(c) Moving a virtual tripod or a top anchor using a grab gesture. (d)/(e) Moving a virtual tripod or a top anchor using a distance grab gesture.

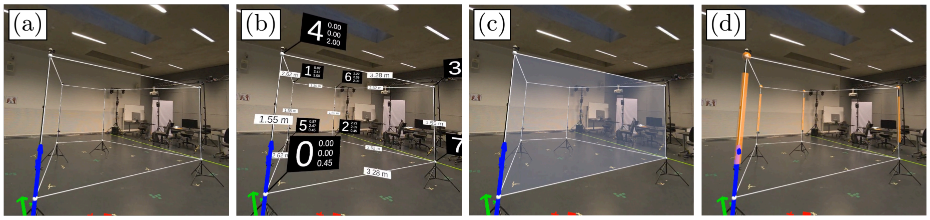

Fig. 3. Several overlay info layers that can be toggled on/off individually, allowing several possible combinations of (a) CPSp edges, (b) anchor ID with position and edge length information, (c) CPSp faces depicting volume, and (d) affordance overlay as interaction indicators for editing the CPSp.

3 Real-World Applications with UWB

XR-PALS is developed around the Crazyflie platform. The Crazyflie, a nano quadcopter developed by the Swedish company Bitcraze AB, was designed primarily as an open-source and open-hardware developmental platform. The Crazyflie itself, without additional sensors, lacks the ability to determine its position in space over the long term. The LPS, an external system that provides accurate positional information, addresses this limitation. According to the documentation, UWB modules (i.e., DWM1000) are used to measure the distance between each module. With a minimum of four, preferably eight, anchors and a Loco Positioning deck mounted on the Crazyflie, it can calculate its position onboard. Fig. 4a exemplarily shows the placement of the anchor in our work. The optimal placement requires certain conditions: the anchors should be evenly distributed around the flying volume, with at least 2 meters of separation between them. They must also have an unobstructed line of sight to the flying area and be positioned at least 15 centimeters away from walls, ceilings, or metal objects to avoid interference from reflections. Considering these requirements and the advantages of mobile tracking systems (such as flexible repositioning and temporary mobile setups, as noted in Section 2), a more robust design for dynamic anchors setup is necessary.

In our work, the LPS is a low-cost, lightweight tracking system using UWB technology. It has already been employed in our case studies to allow precise control of drone swarms.[2] UWB has recently been widely used in robotics, demonstrating robust and scalable deployment. Due to its affordability, UWB enables the development of lightweight systems optimized for real-world industries, including smart manufacturing and robotics research [3, 5, 6, 12, 13, 21, 22, 24].

It has been explored not only in resource-constrained environments, such as single-anchor UWB configurations paired with IMUs for accurate tracking in simpler setups (e.g., [3, 6, 13, 24]), but also in more complex indoor environments.

[2] See [7, 8] and the video demonstration on programming drone swarms: https://videocampus.sachsen.de/video/programming-drone-collectives/dcaa90c5c4023641b62223d8d5f5f515.

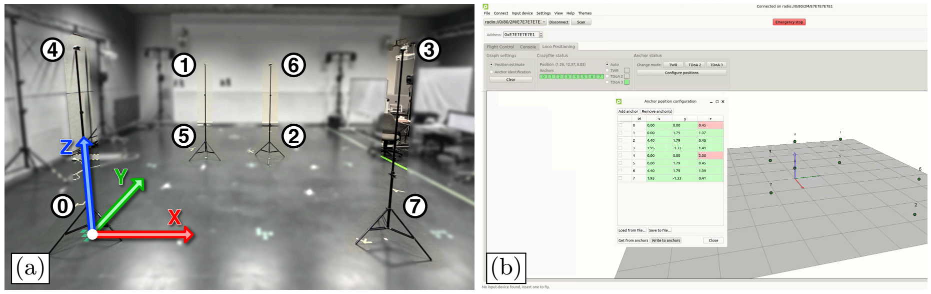

Fig. 4. Loco Positioning System. (a) Placement of anchors in our 8-anchor setup. (b) Screenshot of the LPS configuration tool `cfclient`.

UWB is typically not used as a standalone tracking technology in these scenarios. As noted by [5], "UWB only measures distance and cannot provide an actual reference point on the indoor environment, it is often used in combination with visual inertial odometry (VIO)." Thus, UWB-VIO systems are particularly more effective in low-texture or non-line-of-sight (NLOS) environments, such as warehouses or tunnels, where they deliver consistent performance where vision-based systems or SLAM alone may fail [12, 17, 21, 22].

There are some works that integrate XR with UWB but focus on either using UWB for various XR applications or adding XR to increase the accuracy through hybrid methods [1,4,10,16,17]. There is a lack of research on using XR to simplify the setup process of UWB systems, which is the focus of our work.

4 XR-PALS: Design and Implementation

4.1 Feature of XR-PALS

Setting up the LPS is a complex task requiring precise measurements and configurations of anchors. As mentioned in Section 3, preferably, eight anchors must be used for precise positioning. The user must calculate the exact position of each anchor in a 3D Cartesian coordinate system and then manually enter the data into the `cfclient` software, as seen in Fig. 4b. To add to the complexity, the ID of each anchor, sequentially numbered 0 through 7, cannot be changed and its placement must follow the layout displayed in Fig. 4a. This complexity is bound to have human errors that can occur at several stages of the process, such as calculating the Cartesian coordinates, entering them into the `cfclient`, or even placing the anchor IDs in the correct placement. All this is becoming even worse when the CPSp layout is not a perfect rectangle.

XR-PALS is designed to simplify the setup process, make it more efficient, and eliminate human errors as much as possible. The main design idea of XR-PALS is to allow the user to work in the virtual and the physical world simultaneously, allowing the user to synchronize the two worlds by simply superimposing and matching the virtual layout to the physical one. This is done using the two separate components of XR-PALS: the MR application and the middleware.

MR Application. The MR application runs on an MR headset, which is a VR headset with passthrough capabilities, with core features that can be seen in the requirement specifications in Table 1. The core features are divided into four categories: info layer overlays (**REQ-1xx**), interactions (**REQ-2xx**), export/import (**REQ-3xx**), and support (**REQ-4xx**).

It is essential to have a visual representation of the CPSp layout in the MR environment. The CPSp origin (**REQ-111**) must be permanently rendered since it is used as the anchor ID 0 position in the XY-axes and the origin of all other anchors when calculating their Cartesian coordinates. CPSp edges (**REQ-112**) and faces (**REQ-115**) are very helpful in visualizing the boundary or volume of the CPSp concerning the available space in the physical room. Anchor information (**REQ-113**) is helpful to identify the correct ID placement and to know the exact position in Cartesian coordinates relative to the origin, and the edge length information can be used to ensure that the anchors are placed at the correct distance from each other. This information should be automatically updated whenever an anchor moves (**REQ-114**) and should be done lag-free (**REQ-122**). When setting up the CPSp in a small room, showing all the info layers can be overwhelming, so the user should be able to toggle on/off each info layer (**REQ-121**) using the hand menu (**REQ-411**). The interaction indicators (**REQ-116**) are essential as affordance ovelays for the user.

The core functionality of the MR application is the ability to move the virtual anchors around to set up the CPSp layout. This can be done by grabbing or distance-grabbing (grabbing an object from a far distance) an anchor or a pole that combines two anchors simulating a tripod and moving it around (**REQ-211-214**). The origin will automatically be moved and rotated when anchor ID 0 or 7 is moved (**REQ-215,216**), either by moving said anchor or the pole related to it.

Export/import functionality is straightforward, allowing the user to export and import a CPSp layout to and from the middleware (**REQ-311-313**). An important feature is the ability to also save the YAML file locally as a backup, ensuring the user does not lose any progress when the middleware is down (**REQ-321**).

Since the user works in an MR environment, moving around in the physical space is necessary to set up the anchors and ensure that all virtual and physical anchors are superimposed precisely. This is why we designed a hand menu (**REQ-411**), a hand-attached menu that only appears when the user raises their left hand with the palm facing upwards. This gesture is natural and does not interfere with other hand-based interactions. A stationary menu that stays in place where it opened would make it less accessible for quick access. The hand menu consists of buttons to toggle on/off all the info layers (**REQ-121**) and export the anchor positions to the middleware (**REQ-311,312**).

Lastly, providing visual (**REQ-221,421**) and auditory feedback (**REQ-222,422**) for hand interactions is essential to inform the user that the interaction was successful, aside from increasing the immersive experience. This is because we use hand interaction instead of controller interaction, making haptic feedback impossible. The feedback is available for default, hover, and select state to inform the user about the current interaction state, eliminating false affordance (i.e., phantom click or illusory button press) when taking action.

Middleware. The middleware is a service that connects to the MR application and `cfclient`, passing through anchor positions data to each other, and must be installed on the same machine as the `cfclient`. It is a REST API service developed in Rust and has endpoints for handling HTTP requests from the MR application.

Workflow. XR-PALS makes two workflows possible when setting up the LPS. The first workflow is to map the virtual layout to the physical one. This workflow is possibly a more common use case in the dynamic deployment of LPS since any new room the can be different in size and shape. So, the easiest way is first to put the physical tripods in the four corners of the new room, then use XR-PALS to adjust the virtual layout to match the physical one. After the virtual layout is correctly superimposed to the physical one, the user can export the virtual anchor positions to the `cfclient` and, consequently, update the physical anchors.

The second workflow is the other way around, mapping the physical layout to the virtual one. For example, this use case may occur when the user has a specific CPSp layout for a specific robotics experiment that has to be precisely replicated in the new room. In this case, the user can first load the virtual layout, adjust the origin position and orientation so the virtual layout fits the new room, and then manually put all physical anchors to match the virtual ones.

After setting up, it is also possible to have a real-time update of the CPSp. This is done by automatically exporting the new anchor positions only if changed. No longer need to wait for any user's action. An example use case is a drone swarm experiment in a vast room, where the drones are flying, and the anchors are moved around the room to follow the drones since we only want to monitor the detailed change in swarm formation and not the whole room.

4.2 Implementation Aspects

The overall system consists not only of two core components described in Section 4.1 but of multiple interconnected components, as illustrated in the system architecture diagram in Fig. 5:

- XR-PALS MR app: The user interacts with the system via an MR interface, which allows for the swift and intuitive management of CPSp. The user can export the anchor positions data and send it to the `cfclient` through XR-PALS middleware.
- XR-PALS middleware: This middleware is installed in the same machine as `cfclient` and acts as an intermediary, handling API requests from the XR-PALS MR app and passing data to the `cfclient`.
- Crazyflie PC client (`cfclient`): The LPS software responsible for issuing flight commands and processing sensor data. It runs on a local machine and

Table 1. Some of the requirement specifications of XR-PALS, specifically for the MR application, cf. Figs. 2 and 3. In the Category column, [F] are functional requirements, describing system functionalities, and [N] are non-functional requirements, describing system qualities and constraints.

ID	Category	Description
REQ-111	[F] Origin	Render the CPSp origin.
REQ-112	[F] Info Layer	Render the CPSp edges and update the corresponding edges if an anchor moves.
REQ-113	[F] Info Layer	Render anchor ID, position, and edge length.
REQ-114	[F] Info Layer	When an anchor moves, recalculate and display the correct position relative to CPSp origin for all connected anchors and the correct length for all connected edges.
REQ-115	[F] Info Layer	Render the CPSp faces and update the corresponding faces if an anchor moves.
REQ-116	[F] Info Layer	Render the interaction indicators.
REQ-121	[N] Info Layer	Allow mix-and-match combination of which overlay info layers to render.
REQ-122	[N] Performance	When an anchor moves, ensure the recalculation and rerendering of all connected CPSp components and info layers are lag-free.
REQ-211	[F] Edit	Moving an anchor by grabbing.
REQ-212	[F] Edit	Moving an anchor by distance-grabbing.
REQ-213	[F] Edit	Moving two anchors by grabbing the corresponding pole.
REQ-214	[F] Edit	Moving two anchors by distance-grabbing the corresponding pole.
REQ-215	[F] Edit	Moving anchor ID 0 (through an anchor or pole) will move the CPSp origin.
REQ-216	[F] Edit	Moving anchor ID 7 (through an anchor or pole) will rotate the CPSp origin to point the X-axis towards it.
REQ-221	[N] Usability	Add visual feedback to anchor and pole interaction indicators.
REQ-222	[N] Usability	Add auditory feedback to anchor and pole interaction indicators.
REQ-311	[F] Export	Manually export anchor positions in YAML format and send it to the XR-PALS middleware.
REQ-312	[F] Export	Auto-export and auto-update the `cfclient` in real-time through XR-PALS middleware.
REQ-313	[F] Import	Load and generate a CPSp layout from the XR-PALS middleware.
REQ-321	[N] Usability	Export a YAML backup copy to the local device when XR-PALS middleware is unavailable.
REQ-411	[F] Menu	Hand-attached menus only appear when the user raises their left hand with the palm facing upwards.
REQ-421	[N] Usability	Add visual feedback to all button presses.
REQ-422	[N] Usability	Add auditory feedback to all button presses.

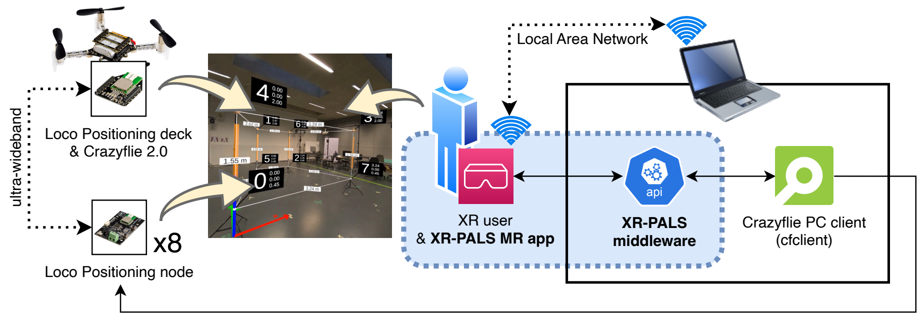

Fig. 5. System architecture (figure contains elements from Bitcraze AB [2]). The two core components, XR-PALS MR application and XR-PALS middleware, are bolded and highlighted in blue.

interacts with the drone hardware. In our case, it communicates with the XR-PALS middleware to receive anchor position data from the XR-PALS MR app and then sends it to the Loco Positioning nodes.
- Loco Positioning node (x8): The anchors that are mounted on moveable tripods, two anchors per tripod. The anchor IDs must be placed in a specific placement layout, as displayed in Fig. 4a.
- Loco Positioning deck (mounted on the Crazyflie 2.0 drone): This UWB module communicates with the Loco Positioning nodes, enabling precise positioning for the mounted drone.

Roughly speaking, our system architecture can be regarded as an “extended plugin” for the LPS tab in the `cfclient` (the standard configuration tool for the LPS of the Crazyflie platform, also refer to Fig. 4b), but with enhanced capabilities.

4.3 Interoperability

Most UWB-based systems require the actual positions of the anchors to be known. To ensure interoperability, the architecture of XR-PALS is designed to allow the updating of anchor positions for systems beyond the LPS. That is, the interactive modeling of coordinates remains consistent, and the import functionality is platform-dependent. We adopt a component-based approach, where developers only need to provide a new implementation that adheres to a specified interface, enabling seamless integration with other tracking systems.

Manual Configuration. The node positions can be directly read within the MR application and are then sequentially entered into the `cfclient` application, which, in our case, is the standard software for the LPS (refer to Fig. 4b).

Configuration by File Import. XR-PALS supports importing position data by exporting a file tailored to the tracking system currently used in the MR application. Fig. 6 illustrates the YAML inline format we use, structured as a dictionary where the keys (0, 1, 2, and so forth) are integers representing unique anchor identifiers. The values are nested dictionaries containing the anchor's x, y, and z coordinates in 3D space, stored as floating-point numbers. The file can be directly imported into the `cfclient` (refer to Fig. 4b).


```
0: {x: 0.0, y: 0.0, z: 0.41}
1: {x: -1.51, y: 1.8, z: 0.15}
2: {x: 1.78, y: 1.76, z: 2.41}
3: {x: 1.30, y: -1.94, z: 0.27}
4: {x: -1.50, y: -1.46, z: 2.50}
5: {x: -1.56, y: 1.81, z: 2.38}
6: {x: 1.82, y: 1.80, z: 0.15}
7: {x: 1.29, y: -1.83, z: 2.53}
```


Fig. 6. YAML inline format for storing anchor positions.

Remote Configuration. The service is capable of automatically setting the position data in the anchors. Therefore, XR-PALS calls an external service on the host computer. The source code of this service is based on the original LPS node firmware.[3] This requires a system in TDoA mode and a Crazyflie with the LPS deck. A web service runs on another machine, accepting one, multiple, or all positions for the anchors. The data is provided in the same YAML format depicted in Fig. 6.

5 Evaluation

To evaluate the effectiveness of XR-PALS compared to a traditional laser-based approach, we conducted a statistical analysis using Analysis of Variance (ANOVA) on two key performance metrics: *task completion time* and *measurement accuracy*. The goal was to determine whether the choice of measurement device significantly influenced these outcomes while controlling for potential confounding factors such as age, sex, and spatial layout complexity.

5.1 Participants

Eighteen participants (14 male, 4 female), aged between 26 and 41 years, participated in the study (refer to Table 2). Participants had varying levels of prior

[3] Refer to https://github.com/bitcraze/lps-node-firmware/blob/master/tools/lpp/set_positions.py.

AR/MR/VR experience, rated on a scale from 1 (beginner) to 5 (expert). The distribution of experience levels was as follows: four participants were beginners (1), two had limited experience (2), one had intermediate experience (3), and two were experts (5).

Table 2. Summary of descriptive statistics of our user study variables.

Variable	Min	1st Quartile	Median	Mean	3rd Quartile	Max
Age	23.00	27.25	29.00	30.78	33.50	41.00
Experience Level*	1.00	1.00	2.00	2.33	3.00	5.00
Completion Time (s)	70.0	111.0	372.5	391.9	627.2	960.0
Instruction Time (s)	180.0	240.0	300.0	330.0	360.0	600.0

* This includes 9 missing values because 9 participants were using the laser measurement device in the control group. Here, we did not ask for the experience level.

Table 3. Distribution of spatial layouts (R, T) across groups (XR-PALS, Laser). R means rectangular, and T means trapezoidal.

Group (Tool)	Anchor Layout	Group Size n
XR-PALS	R	4
XR-PALS	T	5
Laser	R	5
Laser	T	4

5.2 XR-PALS (Test Group) and Laser (Control Group)

Participants were divided into two equal groups (refer to Table 3):

- XR-PALS group ($n = 9$): Used our XR-PALS application for the measurement task.
- Laser-based group ($n = 9$): Used a conventional laser measurement device for the measurement task.

Both groups performed the same task to configure the LPS, with the only variation being the layout.

5.3 Experimental Design

Task Description. Each participant was tasked with measuring the physical positions of the anchors using either the XR-PALS in an MR environment or a conventional laser measurement device. The layouts varied in complexity, ranging from a simple rectangular configuration to a more challenging trapezoidal one, requiring additional spatial calculations. For the XR-PALS group, the task was completed when the user pressed the "Export" button inside XR-PALS after finishing the measurements. In contrast, participants using the laser measurement device completed the task when the measurement values were manually entered into the `cfclient` application, the standard software for the LPS in the Crazyflie platform (cf. Fig. 4b).

Tutorial Session. Participants completed a tutorial session before the measurement task, ensuring familiarity with their assigned tool. Instruction time ranged from 180 to 600 seconds (mean ± SD: 330 ± 111.3024, refer to Table 2). Only one participant required approximately 10 minutes of instruction on how to use the laser measurement device.

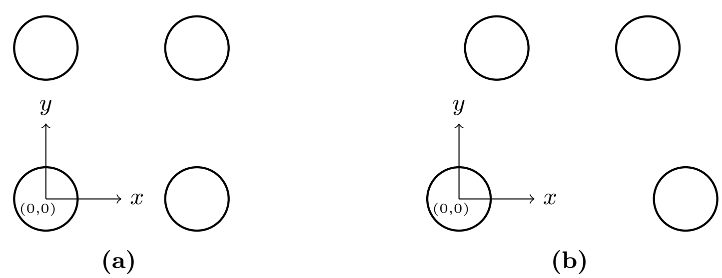

Fig. 7. Two spatial arrangements of the UWB system: (a) rectangular layout and (b) trapezoidal layout.

Tracking System Layout. The study varied the physical layout of the UWB system (refer to Fig. 7 and Table 3):

– Rectangular Layout ($n = 9$): A regular, straightforward spatial configuration.
– Trapezoidal Layout ($n = 9$): A more complex, irregular configuration requiring additional calculations.

The height of each anchor was provided in advance and was set to a default value (cf. with Fig. 4a): For the lower anchors (IDs 0, 2, 5, 7), the height was set to 45 cm, and for the top anchors (IDs 1, 3, 4, 6), the height was set to 2 m. In other words, participants were required to measure only the x and y coordinates for each anchor.

5.4 Results and Discussion

In the following, we test *task completion time* and *measurement accuracy*. Task completion time serves as an essential indicator of efficiency, measuring how quickly participants could complete the configuration of the LPS using either XR-PALS or a laser-based approach. Similarly, measurement accuracy assesses the precision of the recorded anchor positions.

Task Completion Time. The bar plots depicted in Fig. 8 illustrate the *mean task completion times* for each group, showing a clear advantage for XR-PALS. To model the relationship between task completion time and the predictor variables, we used the following linear equation: $Y_{\text{time}} = \beta_0 + \beta_1\text{Group} + \beta_2\text{Age} + \beta_3\text{Sex} + \beta_4\text{Layout} + \epsilon$. Here, Y_{time} represents the task completion time, ϵ the residual error, while β_1 to β_4 denote the estimated coefficients for the measurement tool (Group), participant Age, Sex, and Layout complexity.

The results of the ANOVA indicate a significant effect of the measurement tool (XR-PALS vs. laser-based group) on task completion time ($F(1, 13) = 44.426, p < 0.001$), suggesting that participants using XR-PALS completed the measurement task significantly faster than those using the laser-based method. The non-significant effects of Age ($F(1, 13) = 0.579, p = 0.460$), Sex ($F(1, 13) = 0.182, p = 0.677$), and Layout ($F(1, 13) = 0.008, p = 0.930$) suggests that performance differences were not driven by demographic factors or the spatial arrangement of the anchors but were instead primarily influenced by the *measurement tool* itself.

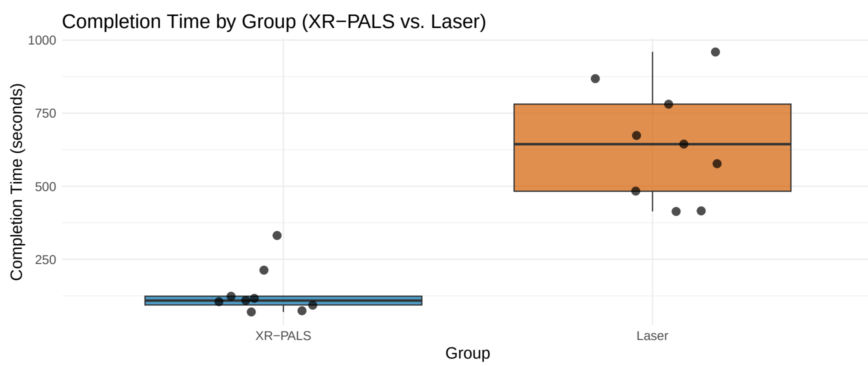

Fig. 8. Mean task completion times for each group, showing an advantage for XR-PALS over the laser-based method.

Accuracy. The accuracy results are depicted in Fig. 9 for all participants. The linear model used to describe the relationship between accuracy and the predictors is: $Y_{\text{accuracy}} = \beta_0 + \beta_1\text{Group} + \beta_2\text{Age} + \beta_3\text{Sex} + \beta_4\text{Layout} + \epsilon$. Here, Y_{accuracy} represents accuracy, ϵ the residual error, and β_1 to β_4 are the coefficients for the respective predictor variables.

The results of the ANOVA show that Age affected accuracy ($F(1, 13) = 7.794, p = 0.0153$), indicating that accuracy differed across age groups. However, the effects of Group ($F(1,13) = 2.036, p = 0.1772$), Sex ($F(1,13) = 0.569, p = 0.4642$), and Layout ($F(1,13) = 1.830, p = 0.1992$) were not significant. This suggests that Layout and demographic factors, such as Sex and Age, did not significantly influence accuracy, while Age was the main factor impacting accuracy. However, we observed in our study that the laser-based system resulted in measurement errors for some participants. On the other hand, no fundamental errors were observed in the XR-PALS group (refer to the large error bars in Fig. 9). This can be attributed to the more intuitive design of XR-PALS.

In the following, we discuss how the mean error was measured and some observations regarding the impact of both layouts.

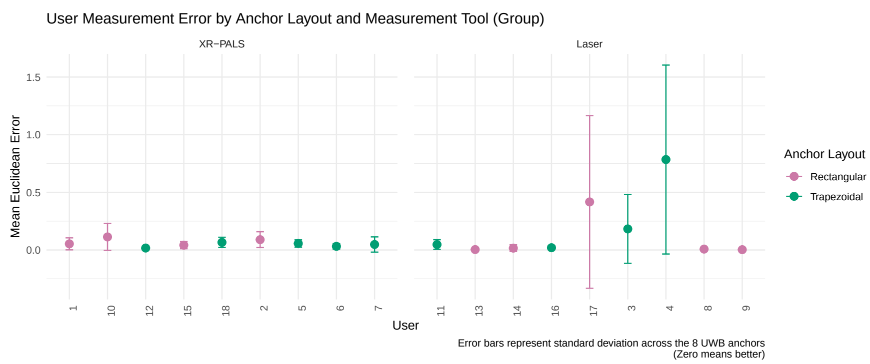

Fig. 9. Mean measurement error for each group when determining the physical positions of the anchors.

Computing the Error. For each layout, we first measured the positions of all 8 anchors once. These measurements served as our baseline data set for both the rectangular and trapezoidal layouts. Participants were then tasked with "recording" the positions of the anchors, as described in Section 5.3. To compute the accuracy of their measurements, we calculated the Euclidean distance between each pair of corresponding points. Specifically, for each anchor, we compared its measured position to the baseline position of the respective layout, computing the Euclidean distance as the difference between the measured and actual coordinates in 3D space.

Effect of the Layout. The shape of the layout had a small yet notable impact on laser users. The trapezoidal layout increased the difficulty of manual calculations (refer to the green error bars on the right-hand side of Fig. 9). However, the sample size was insufficient to compute a p-value. In contrast, XR-PALS users showed no significant difference between rectangular and trapezoidal layouts, as XR-PALS handled spatial calculations automatically (refer to the pink and green error bars, respectively, on the left-hand side of Fig. 9).

(a) **(b)**

Fig. 10. Bigram word cloud generated from user feedback from (a) the XR-PALS group and (b) the laser-based group.

Usability. After completing the study, participants were asked for feedback on their experiences. Specifically, we inquired about aspects such as what they found difficult or easy, what features they would like to see, and what could be improved for greater ease of use. A bigram word cloud for each tool was generated from their comments (see Fig. 10), which provides an overview of their feedback. Participants using XR-PALS reported a higher level of ease and intuitiveness compared to the laser-based group. In contrast, some laser users expressed concerns about potential mistakes, leading to hesitancy in decision-making. Some participants found the laser measurement device more challenging to use, especially when the layout was trapezoidal, even though it was perceived as relatively easy for the rectangular layout. This was also evident in the data, as can be derived from the higher completion time of the task and, in some cases, lower accuracy. Overall, participants provided more positive feedback for XR-PALS than the laser-based one. The primary reasons for this preference were the longer time required for the laser method, the reduced likelihood of making mistakes, and the lower cognitive load associated with computing positions manually with pencil and paper. Participants also agreed that using the laser measurement device multiple times for mobile setups would not be viable, suggesting that the laser method is less practical in such contexts.

6 Conclusion

The proposed MR application, XR-PALS, offers a comprehensive solution for the virtual creation and physical deployment of the Bitcraze LPS. Specifically, it enables the seamless virtual management of CPSp for Crazyflie tiny drones. While designed for the Bitcraze platform, this approach is equally applicable to other UWB-based systems, expanding its potential for broader use.

The integration of MR technology simplifies the deployment process by reducing configuration time significantly. As a result, there is an alternative to traditional measurement tools, such as laser devices or tape measures on the floor. Unlike conventional approaches that rely on default configuration tools, XR-PALS provides an immersive, interactive experience, making the management of CPSp more intuitive and user-friendly.

Our findings suggest that XR-PALS offers significant advantages over traditional laser-based methods: it reduces task completion time, eliminates errors, and improves user confidence. These results indicate that our MR-based application can significantly enhance the efficiency of configuring UWB-based tracking systems, especially in non-standard layouts. The immersive nature of MR also contributes to better decision-making and lower cognitive load, which can be particularly beneficial in complex setups.

Future Work. The scalability of XR-PALS in larger, more complex environments requires further investigation, particularly in scenarios involving setups with more than eight anchors. Additionally, while XR-PALS outperformed the laser-based method in terms of task completion time and was less error-prone in usage, the long-term adoption and integration of the system into professional settings still require further evaluation. Furthermore, feedback from the user study highlighted areas for improvement, which will be addressed in future work.

Acknowledgments. Funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden. The European Social Fund (ESF Plus) and the German Federal State of Saxony have funded this work within the project ProSECO (100687967).

References

1. Arun, A., Saruwatari, S., Shah, S., Bharadia, D.: XRLoc: Accurate UWB Localization to Realize XR Deployments. In: Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems. p. 459–473. SenSys '23, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3625687.3625810
2. Bitcraze AB. bitcraze.io (2025). https://www.bitcraze.io/products/loco-positioning-node/, https://www.bitcraze.io/products/loco-positioning-deck/, https://www.bitcraze.io/products/old-products/crazyflie-2-0/. Accessed 9 Jan 2025
3. Cao, Y., Yang, C., Li, R., Knoll, A., Beltrame, G.: Accurate position tracking with a single UWB anchor. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 2344–2350 (2020). https://doi.org/10.1109/ICRA40945.2020.9197345
4. Choi, H.B., Lim, K.W., Ko, Y.B.: Improved virtual anchor selection for ar-assisted sensor positioning in harsh indoor conditions. In: 2020 Global Internet of Things Summit (GIoTS). pp. 1–6 (2020). https://doi.org/10.1109/GIOTS49054.2020.9119679
5. Choi, H.B., Lim, K.W., Ko, Y.B.: LUVI: Lightweight UWB-VIO based relative positioning for AR-IoT applications. Ad Hoc Networks **145**, 103132 (2023). https://doi.org/10.1016/j.adhoc.2023.103132

6. Elsanhoury, M., Nieminen, J., Välisuo, P., Siemuri, A., Koljonen, J., Elmusrati, M., Kuusniemi, H.: Precise Indoor Positioning System for Mobile Robots via Smoothed UWB/IMU Sensor Fusion. In: 2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN). pp. 1–6 (2023). https://doi.org/10.1109/IPIN57070.2023.10332542
7. Grzelak, D.: Programming Drone Collectives: Towards Safe Plug-And-Play Modularity. In: Proceedings of the 1st German Robotics Conference (GRC). Robotics Institute Germany, Nürnberg, Germany (Mar 2025), https://ras.papercept.net/conferences/conferences/GRC25/program/GRC25_ContentListWeb_1.html#thga_58, Standalone video contribution (demo video, abstract, and poster)
8. Grzelak, D., Lindner, M., Belov, M., Aßmann, U., Husak, O., Fricke, H.: A Bigraphical Framework for Modeling and Simulation of UAV-based Inspection Scenarios. Preprint (Revised). Technische Universität Dresden (2024), https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-908655
9. Jang, I., Hu, J., Arvin, F., Carrasco, J., Lennox, B.: Omnipotent Virtual Giant for Remote Human & Swarm Interaction. In: 2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN). pp. 488–494. IEEE Press (2021). https://doi.org/10.1109/RO-MAN50785.2021.9515542
10. Lee, C., Shim, K., Seo, S., Ryu, G., Choi, Y.: Never Tell the Trick: Covert Interactive Mixed Reality System for Immersive Storytelling. In: 2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW). pp. 1009–1010 (2024). https://doi.org/10.1109/VRW62533.2024.00301
11. Li, X., Wang, J.D., Dudley, J.J., Kristensson, P.O.: Swarm Manipulation in Virtual Reality. In: Proceedings of the 2023 ACM Symposium on Spatial User Interaction. pp. 1–11. SUI '23, Association for Computing Machinery (2023). https://doi.org/10.1145/3607822.3614519
12. Lin, H.Y., Yeh, M.C.: Drift-Free Visual SLAM for Mobile Robot Localization by Integrating UWB Technology. IEEE Access **10**, 93636–93645 (2022). https://doi.org/10.1109/ACCESS.2022.3203438
13. Liu, Z., Li, J., Wang, A., Cheng, X., Wang, A.: Design and Implementation of UWB/MIMU Tightly-coupled System for Indoor Positioning. In: 2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS). pp. 1–7 (2018). https://doi.org/10.1109/UPINLBS.2018.8559945
14. Macchini, M., De Matteïs, L., Schiano, F., Floreano, D.: Personalized Human-Swarm Interaction Through Hand Motion. IEEE Robotics and Automation Letters **6**(4), 8341–8348 (2021). https://doi.org/10.1109/LRA.2021.3102324
15. Makhataeva, Z., Varol, H.A.: Augmented Reality for Robotics: A Review. Robotics **9**(2), 21 (2020). https://doi.org/10.3390/robotics9020021
16. Miller, J., Soltanaghai, E., Duvall, R., Chen, J., Bhat, V., Pereira, N., Rowe, A.: Cappella: Establishing Multi-User Augmented Reality Sessions Using Inertial Estimates and Peer-to-Peer Ranging. In: 2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). pp. 428–440 (2022). https://doi.org/10.1109/IPSN54338.2022.00041
17. Miller, J., Soltanaghai, E., Duvall, R., Chen, J., Bhat, V., Pereira, N., Rowe, A.G.: Multi-user augmented reality with infrastructure-free collaborative localization. ArXiv **abs/2111.00174** (2021), https://api.semanticscholar.org/CorpusID:240354122
18. Örnhag, M.V., Persson, P., Wadenbäck, M., Åström, K., Heyden, A.: Trust Your IMU: Consequences of Ignoring the IMU Drift. In: Proceedings of the IEEE/CVF

Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 4468–4477 (June 2022)

19. Park, N.J., Kim, K.H., Ma, J.M., Son, J.H., Ahn, H.S.: Implementation of a multi-drone system for UWB-based onboard localization with free structural anchors. Mechatronics **102**, 103234 (2024). https://doi.org/10.1016/j.mechatronics.2024.103234
20. Seiger, R., Aßmann, U., Grzelak, D., Belov, M., Riedel, P., Podlubne, A., Zhao, W., Kerber, J., Mohr, J., Espinosa, F., Schwartz, T.: Immersives verteiltes robotic co-working. Informatik Spektrum **43**(6), 425–435 (2020). https://doi.org/10.1007/s00287-020-01297-w
21. Wang, J., Gu, P., Wang, L., Meng, Z.: RVIO: An Effective Localization Algorithm for Range-Aided Visual-Inertial Odometry System. IEEE Transactions on Intelligent Transportation Systems **25**(2), 1476–1490 (2024). https://doi.org/10.1109/TITS.2023.3317408
22. Xu, H., Wang, L., Zhang, Y., Qiu, K., Shen, S.: Decentralized Visual-Inertial-UWB Fusion for Relative State Estimation of Aerial Swarm. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 8776–8782 (2020). https://doi.org/10.1109/ICRA40945.2020.9196944
23. Zhao, J.: A Review of Wearable IMU (Inertial-Measurement-Unit)-based Pose Estimation and Drift Reduction Technologies. Journal of Physics: Conference Series **1087**(4), 042003 (Sep 2018). https://doi.org/10.1088/1742-6596/1087/4/042003
24. Zhong, Y., Liu, T., Li, B., Yang, L., Lou, L.: Integration of UWB and IMU for precise and continuous indoor positioning. In: 2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS). pp. 1–5 (2018). https://doi.org/10.1109/UPINLBS.2018.8559718