Evaluating the Usability and Performance of the LighthouseV2 Ground Truth System

Lukas Furtner, Maximilian Schuh, Markus Schuß, Carlo Alberto Boano
Institute of Technical Informatics, Graz University of Technology, Austria
E-Mail: lukas.furtner@student.tugraz.at, {schuh, markus.schuss, cboano}@tugraz.at

***Abstract*—As the IoT gained increasing importance in the last decade, the need for testing facilities is also rising. One requirement for many IoT systems is location awareness. In order to test the accuracy of location estimates, developers commonly need a reference system in their testbed. Such a reference system should be accurate, precise, and scalable. It should offer a high and constant update rate, and be affordable at the same time. In 2018, bitcraze introduced LighthouseV2, a reverse-engineered optical indoor localization system initially designed for virtual reality gaming. According to bitcraze, their LighthouseV2, which they use for the localization of nano drones, should offer all the requirements for a reference system. However, most of the work that evaluates the performance of the lighthouse system refers to LighthouseV1, which uses a different kind of hardware.**

In this work, we present our evaluation of LighthouseV2. We show how the LighthouseV2 can be a ground truth system that provides accuracy better than 10cm and scalability in a dynamic domain. In our experimental setups, we use up to six base stations and show the differences between the position estimation algorithms in the crazyflie nano drone. In addition, we also mention some weaknesses and knowledge to ensure LighthouseV2's usability as a ground truth system in a large-scale deployment.

***Index Terms*—LighthouseV2; Indoor Localization; Crazyflie; Testbed**

I. Introduction

Due to the growing importance of the Internet of Things (IoT), the need for testing and benchmarking facilities for these systems is on the rise [1] [2] [3]. The recent advance of ultra-wideband (UWB) in the IoT allowed for new, context-aware applications due to its excellent localization performance. This development fostered the need for new testbeds that allow the testing and benchmarking of UWB systems in large-scale, real-world environments [4] [5]. However, to effectively benchmark highly-mobile UWB-based systems, ground truth systems are necessary to track the position of the devices within the testbed. Such reference systems come with several requirements.

Accuracy and precision. The main requirement of a reference system is its accuracy and precision. To be able to benchmark a new system, the reference system has to provide at least the same accuracy of 5-15cm. Otherwise, the measurement results are unusable, as the evaluation system introduces bigger errors than the developed system. Many works on UWB show that this technology can achieve an accuracy of 5-15cm [6]. Therefore, for testing the accuracy of UWB, the reference system should also be in the domain of 10cm. An accuracy higher than 5cm would be even better, ensuring the system performs better than UWB. For precision, the reference system should at least be better than 1cm to provide reproducible measurements.

Scalability. The scalability of the reference system is also an essential characteristic. A scalable system can adapt to changing testing methods and requirements without losing performance. That includes, for example, the usage of 5 clients simultaneously, which is needed for testing systems with several mobile agents that are present at the same time. Serving multiple objects at the same time can speed up testing or even enable tests that are based on interferences with other clients. Another essential requirement is the scalability of the measurement area. That serves the ability to start with tests on smaller areas and still be able to expand later when a system needs to be tested for areas of up to 250m^2.

Update rate. Another point that the reference system should fulfill is tracking the position of a mobile device. As UWB technology can be used inside products that perform movements, the testbed is also required to support the evaluation of mobile systems. Tests within dynamic domains require a reliable update rate of at least 50ms for the reference system. Therefore, the updates' latency must be within 20ms to support motions. Otherwise, it can occur that the received positioning values do not match the current position, as a movement has changed it already. An additional requirement for mobility is the constancy of updates. Dropped updates lead to a temporary increase in latency, which again introduces an even worse position estimation during movement.

Low cost. In order to keep the price low, the system should consist of low-cost, off-the-shelf components, so the total price is not more than 1,000€. Especially for large-scale testbeds, this requirement is critical to cover the entire facility without costs of 10,000€. Furthermore, open-source hardware and components are desirable as they provide a solid basis for future expansion.

So far, there does not exist a reference system that fulfills all these requirements. One option is a motion-capturing system like OptiTrack [7]. This system works by placing cameras around the area, which should be tracked. With the help of infrared light and reflective markers, it is then possible to capture the motions performed by the markers, leading to the markers' position and orientation. This system provides a localization accuracy within 0.2mm and latencies lower than 9ms [8]. However, motion-capturing systems have a downside, as they are very expensive.

The lighthouse system is entirely different, initially designed for virtual reality gaming. According to its manufacturer, the newer Lighthouse version, LighthouseV2, together with the crazyflie nano drone, should fulfill all the previously mentioned requirements. The off-the-shelf hardware is cheaply priced and should be able to perform 10cm accurate position measurements. Furthermore, it offers scalability and mobility support. Especially the price of 800€ makes this system an attractive candidate to use as ground truth.

Our contributions. In this work, we evaluate if LighthouseV2 is a feasible candidate for a reference system. First, we check the latency distribution across consecutive measurements to support high update rates for an accurate tracing of mobile systems. Then, we evaluate the system's accuracy within different setups. In addition, we show the system's precision and evaluate its robustness, where we test what happens when the system gets influenced by different disturbance sources. We also try different system setups and configurations to evaluate whether this affects precision or accuracy. After that, we provide measurements that show the optimal system setup before we finally perform additional measurements on a scaled-up system.

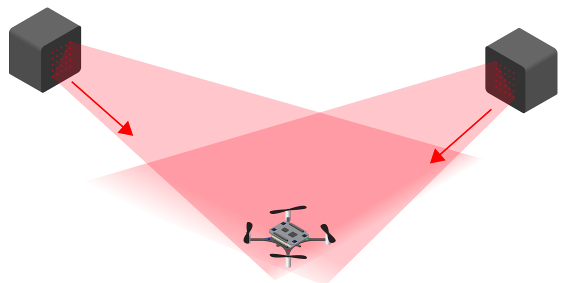


Fig. 1: General idea of the lighthouse system [9]. The lighthouse can work with 1-16 base stations. The crazyflie nano drone requires all four sensors of the positioning deck to receive the beams from a base station in order to calculate its current position.

II. Background

In this section, we first describe the working principle of the lighthouse system. Therefore, we give an overview of the components and explain the base stations. After that, we deal with the extension board on top of the crazyflie, cover the encoding of the laser beams, and present the software used for positioning. Then, we present the limitations of the lighthouse system before we mention related evaluations at the end of the background. Finally, we present the existing testbed.

A. Lighthouse System

The lighthouse system is an optical indoor positioning system, initially built for tracking in virtual reality gaming [9]. Due to its low price of 800€, it also found use cases in the research domain, especially for localization of the crazyflie nano drones [10]. The lighthouse system consists of two parts, the base station [11] and the positioning deck [12], which is mounted on top of the crazyflie nano drone. So far, two versions of the lighthouse system exist, LighthouseV1 and LighthouseV2. The difference between them is that their base stations are built differently. However, the general idea (see Figure 1) of the different versions is similar. We show this working principle in Figure 2: every base station contains lasers that are mounted on a rotating drum. Therefore, the emitted laser beams, which have a wavelength of 850nm (infrared spectrum), sweep through the room and contain information about the angle to the base station. The infrared signal is received by four photodiodes which are mounted on top of the positioning deck that sits on top of the crazyflie. These photodiodes are connected to a Field Programmable Gate Array (FPGA), which also sits on top of the positioning deck. The FPGA decodes the signal and sends the information about the angles via Universal Asynchronous Receiver Transmitter (UART) to the STM32F4 chip, which is the Central Processing Unit (CPU) of the crazyflie. With the help of a config file containing the base stations' positions and headings, this chip finally calculates the current position out of the angles as explained in Section II-E. We explain next the individual components in more detail.

B. Base Station

As stated above, the different lighthouse versions use different base stations.

LighthouseV1. The base stations from this version use two spinning drums that rotate with a constant frequency. One drum spins vertically, whereas the other one spins horizontally. On each of the drums, a laser is mounted so it can sweep through the room. The drums are shifted in a way that there is only one laser that emits light into the room every time. A synchronization light in the base station pulses every time just before a laser emits light to the outside of the case. This synchronization light is encoded with information containing the base station number and which beam (horizontal or vertical) will follow. With this, measuring the time between the synchronization light and the actual laser sweep is possible. This time represents an angle to the base station that can be used for position estimation. This synchronization light can also be used to synchronize with a second station to avoid two laser sweeps through the room simultaneously. Alternatively, the stations can be synchronized via cable. The drawback of LighthouseV1 is that it only supports up to two base stations. That is due to the system's design, as only the synchronization flashes are encoded with information, making it impossible to distinguish between the sweeping lasers if there are more than two stations.

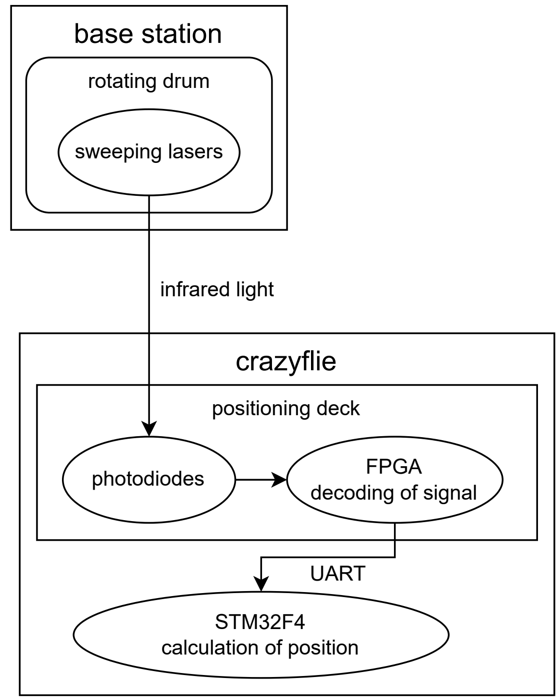


Fig. 2: Working principle of the LighthouseV2.

LighthouseV2. This newer version from 2018 has some severe advantages in its design. Inside the LighthouseV2 there is one horizontal spinning drum (see Figure 3) that contains two lasers (circled in blue). One laser is mounted with an angle of 45 degrees, whereas the other is mounted at -45 degrees. In this newer version, each laser emits a modulated signal. This modulated signal contains the station number, the laser index, and the current angle of the drum. Because of the two 90-degree shifted laser planes from the base station, the receiver of the beams can then calculate the elevation and azimuth angle to the base station. Due to the different modulation of the lasers, they can be distinguished from each other, which no longer requires a synchronization light. With this new identification method of the sweeps, it is possible to use more than two base stations, which provides a bigger working area. The further explanation in the background will refer to this version of the lighthouse.

C. Lighthouse Positioning Deck

For the localization on the receiving side, bitcraze designed a rectangular Printed Circuit Board (PCB) that can be mounted on top of the crazyflie (shown in Figure 4). This board is called the

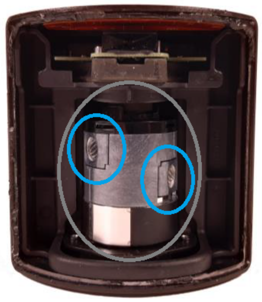


Fig. 3: Insides of LighthouseV2 [13]. The two 45° and -45° mounted lasers (blue mark) are mounted on the horizontal rotating drum (gray mark).

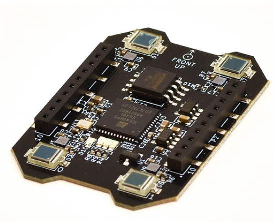


Fig. 4: Lighthouse positioning deck [12]. The phototransistors are located at the corners of the board. An FPGA sits on top of the board as well.

lighthouse positioning deck and contains a BPW34S photodiode on each of its four corners [14], which can detect light from 400 to 1100nm [15]. The best sensitivity of the photodiodes is at a wavelength of 850nm, which matches the wavelength of the lasers in the base stations. Because of the four photodiodes, four different angles (azimuth and elevation angle to every photodiode) to the base station are received every update cycle. That allows positioning and calculating the heading of the drone with a single base station. To handle the decoding of the laser beams, the positioning deck contains an FPGA. After decoding, the FPGA sends the decoded time via UART to the CPU of the crazyflie.

D. Encoding of the laser beam

The manufacturer has not published how the information is encoded to send from the base station to the receiver. However, previous work did some reverse engineering [16]. Researchers state that the LighthouseV2 uses a 17-bit Linear Feedback Shift Register (LFSR) for each laser. In a LFSR, the following input bit is derived from its current state within a feedback loop. This feedback loop takes specified bit positions of the current state and Exclusive Or (XOR)s them together. The result of the XOR becomes the new input bit. The bits, which are XORed, are specified by the LFSR polynomial. In the LighthouseV2, the polynomial from each laser depends on the base station number and the laser index. With timed state updates, it is then possible to produce a bitstream representing the spinning drum's current angle. The state-update frequency of the LFSR is 6MHz, and together with 17 bits, it is possible to produce a bitstream that covers 21.8ms, which fits into the update rate of 20ms for LighthouseV2. The generated bitstream is then encoded in Differential Manchester (DM) before it gets emitted by the laser.

E. Positioning Software

As mentioned in Section II-C, the FPGA sends the decoded information from the beam to the CPU. The CPU then derives the angles and the current position from this information. There exist two different methods for the calculation of the position.

1) Sweeping Method: The sweeping method, which is performed by the extended Kalman filter on the one side directly uses the angles inside the existing Kalman filter to adapt the estimated position. One received base station is enough for this positioning method to calculate the current position.

2) Crossing Method: The crossing beam method calculates the intersection of the received light planes from two base stations. Based on this, the CPU calculates the position of every sensor individually and then takes the average of the positions for the current location. This method needs at least two base stations that are received by the sensors.

To calculate the angles and later the position out of the received data, the CPU needs to know the positions and headings of the base stations. Therefore, the system has to be configured beforehand. The crazyflie offers two different configuration methods.

3) Simple Configuration: The first method, *Estimate geometry simple*, is the faster method. It only requires placing the crazyflie at a position where it receives the base station. This placement point is then used as the origin, whereas the current heading of the drone defines the coordinate system. The position of the base stations is then calculated with the help of the received angle information. However, this configuration method only supports up to two stations.

4) Normal Configuration: The second method, *Estimate geometry*, needs at least two base stations and uses at least three positions for the configuration. These positions are the origin, a 1m distance point into x direction, and a third point on the ground plane. It is furthermore possible to use additional points, which should result in more accurate base station positions. With the three minimum required points, it is possible to set the coordinate system as needed. Therefore, aligning the coordinate system to a global reference is way easier with the second method, as the yaw information of the first method is not very precise. The second method also scales better and supports the configuration of up to 16 base stations.

F. Limitations

According to bitcraze, the second version of the lighthouse positioning system has the following specifications [9]. In general, the lighthouse system is designed for indoor usage. As an optical system, the positioning deck requires a direct line of sight to at least one base station. The base stations should be mounted at least 2m above ground and angled down by 30-45°. The drone should then be operated from the ground up to 50cm below the base stations. Bitcraze states that they observe an absolute accuracy better than 10cm and a relative precision better than a millimeter in their 5x5m working area. The relative precision means the drone can fly back to any saved position and will not be off by more than a millimeter. Other technical characteristics are the update rate of 20ms and that up to 16 base stations are supported in hardware. The range to a base station can be up to 6m, and the view angles of the base stations are 150 degrees vertically and 110 degrees horizontally.

G. Related Evaluations

In this section, we focus on the previous evaluation of the LighthouseV2. Taffanel et al. [17] evaluate both lighthouse versions with a motion-capturing system. Their work presents an average 2-3cm accuracy for the crossing beam method with external motion. The outliers are up to 12cm away from the ground truth position. When flying with the drone, they measure an accuracy better than 5cm for all measurement points and estimation methods, while over 75% of

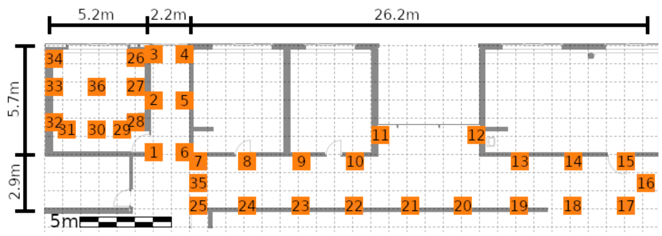


Fig. 5: UWB testbed. The map shows the currently installed nodes. The testbed is divided into two sections, the HALLWAY (nodes 1-25, 35) and the OFFICE (nodes 26-34, 36).

the measured points are within a range of 1.5cm. For precision, they observe all measurement values within 0.7mm. However, this work only uses two base stations, meaning that it does not cover extensive areas. Furthermore, the work does not investigate any source of disturbance that affects the system. Another point that they do not cover is the influence of different configurations and setups.

H. ITI UWB Testbed

This section describes the UWB testbed deployed at our premises. The testbed consists of 53 UWB nodes deployed at 36 positions within an office building. Several nodes (DW1000 and DW3000) for UWB are co-located to allow the benchmarking of the performance. All these nodes cover an area of about 270m^2. The UWB modules are mounted on a rail that sits at the height of 2.7m from the ground. In Figure 5, we present the locations of the nodes. The whole testbed area is split into two regions. One of them (OFFICE configuration) contains nodes 22-34 and 36, whereas the other one (HALLWAY configuration) uses the remaining ones. Our aim is to let the lighthouse system provide localization data in these two regions.

III. Experimental Evaluation

In this section, we first present our experimental setup. That contains the hardware that we use, as well as software parts that we use for further analysis. In Section III-B, we present the completed experiments. Therefore, we describe the individual setups for the measurement before we present and classify the results.

A. Experimental Setup

In order to receive localization data, we use the crazyflie with the lighthouse positioning deck. Therefore, we use the logged position values of raw sweeps directly from the extended Kalman filter. For crossing, we get the values from the internal lighthouse x/y/z variables, representing the calculated position. We establish the connection from a Personal Computer (PC) to the crazyflie via a micro-Universal Serial Bus (USB) cable. To get the localization information to the PC, we access the crazyflie via its Python Application Programming Interface (API). Furthermore, we do any additional information processing with Python. For accuracy measurements, we use a previously-generated 1m grid as a reference. In order to place the crazyflie on the desired points, a mounting has been designed, which was manufactured with a 3D printer. This mounting fits on top of the two available tripods, covering heights of up to 2.5m. In addition, we use a white headlamp with a strength of 700lm and an infrared lamp with a strength of 100W for disturbance measurements.

B. Experiments

This section deals with the performed measurements and their results.

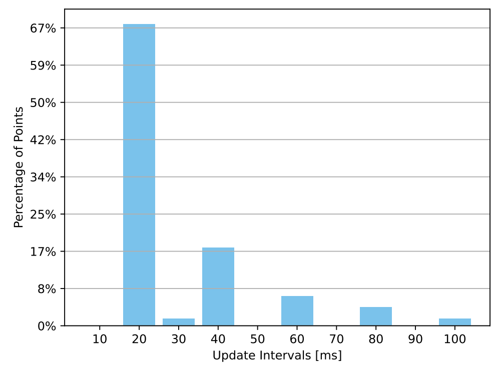
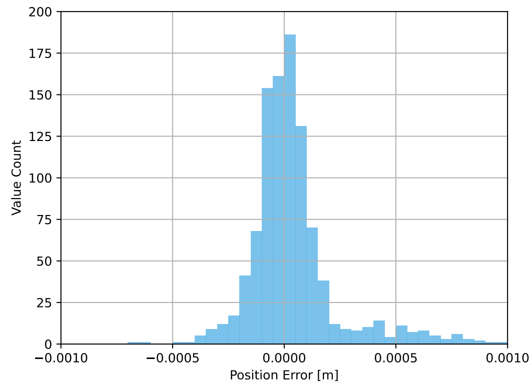


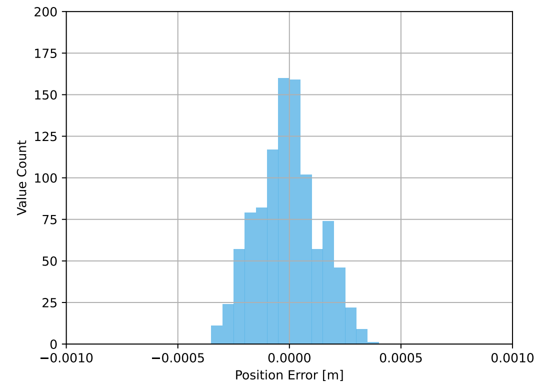
Fig. 6: Update latency. 68% of all updates happen at 20ms. Additionally, 90% of the updates are received within 40ms.

1) Update frequency: This experiment should verify the 50Hz update rate that the manufacturer mentions. After initial experimental investigations, we introduce the following setup for this measurement. A perfect term for measuring the frequency is the position from the crossing method. That is because the corresponding variables, lighthouse x/y/z, do not change if no beams are received and decoded, which is similar to no position update. The fact that this variable is a float, which furthermore means that it is unlikely that the exact position is measured twice, makes it a perfect candidate for this measurement. After configuring the lighthouse system with the normal configuration, we place the crazyflie in the middle of two visible base stations. Then we configure the logging to be in 10ms intervals to expect a value change every second value (20ms) that we receive from the drone. After the setup, we record 1000 values.
We look at the received values for the evaluation and observe that this 20ms update rate is correct but not constant. Figure 6 presents the measured update latencies. It shows that the 20ms update rate is only reached at 68% of all the updates. Furthermore, the figure shows that the system receives about 90% of the updates within 40ms. A race condition during transferring the data from the crazyflie to the PC should be the reason for the occurrence at 30ms. During the measurement, we also discovered one outlier at 430ms which we can not explain. To sum this experiment up, the system does not have a constant update rate of 20ms. However, 90% of the updates are within 40ms, and there are hardly any outliers above 100ms which is still good enough for the reference system.

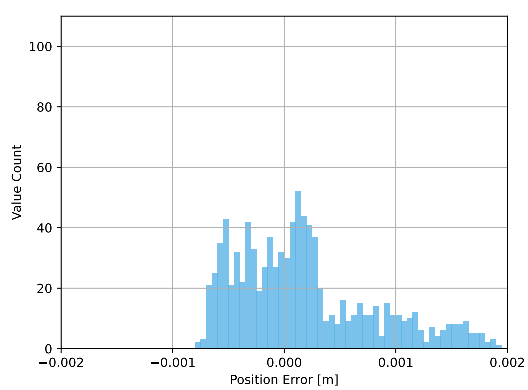
2) Precision: In this experiment, we evaluate how the received position values are distributed. The setup is nearly the same as in the previous experiment. We place the drone in the middle of two lighthouse base stations and start logging with a 10Hz update rate to ensure new position values every time. To get our dataset, we log 1000 values for each positioning method. Additionally, we do the same with three base stations.
We analyze the dataset by norming the values to an average error of 0m, as we are not interested in that for this experiment. In Figure 7, we show the corresponding results. With two stations, the measurement values for sweeping and crossing resemble a Gaussian distribution. For crossing, the distribution is within a range of 1mm. This is not the case for the sweeping method, as it has some outliers 0.5-1mm away from the origin. When using three base stations, we observed different results than expected. The sweeping values for three base stations are distributed as shown in Figure 7c. The reason for this has to be the extended Kalman filter that processes three



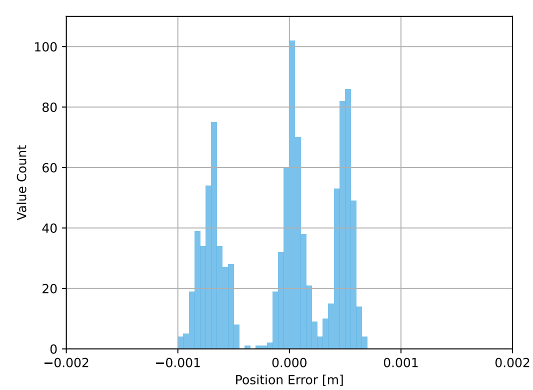
(a) Sweeping distribution of two stations.



(b) Crossing distribution of two stations.



(c) Sweeping distribution of three stations.



(d) Crossing distribution of three stations.

Fig. 7: Distributions of positioning values. The values follow a Gaussian distribution when we use two base stations. The standard deviations for this case are 0.20mm for sweeping and 0.14mm for crossing. When using three stations, the values are no single Gaussian distribution anymore. However, the standard deviation with 0.84mm for sweeping and 0.49mm for crossing is still very precise.

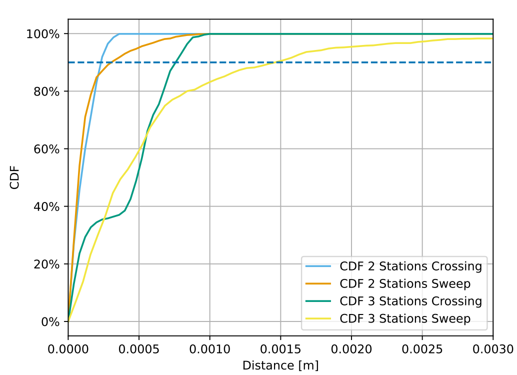


Fig. 8: Distribution of the positioning values. For two base stations, 90% of the received positioning values lie within 0.22mm for crossing and 0.30mm for sweeping. With three base stations, the 90% mark is at 0.75mm for crossing and 1.50mm for sweeping.

slightly different position values from the three stations. We discover three Gaussian populations with different offsets for crossing with three stations (shown in Figure 7d). This is due to the three different pairings of the three stations, where every pairing (1,2; 1,3; 2,3) results in a slightly different position. The corresponding standard deviations are 0.14mm for Figure 7a, 0.20mm for Figure 7b, 0.49mm for Figure 7c, and 0.84mm for Figure 7d. In addition to this, we present the absolute localization errors in Figure 8, where the dashed line marks the 90th percentile value. The plot shows that a setup with two base stations has its 90% mark at 0.22mm for crossing and 0.30mm for sweeping. For a setup with three base stations, the corresponding 90% mark is 0.75mm for crossing and 1.5mm for sweeping. The precision matches the work in Section II-G when only two stations are used. When using a third station, the precision drops slightly but is still in an acceptable range, which is good enough for the reference system.

3) Accuracy with one base station: This experiment aims to show the accuracy achieved with a single base station. Due to the single station, only the sweeping method is available. We perform two different measurements, using both configuration methods mentioned in Section II-E3 and Section II-E4 once. To configure with the normal estimation method that requires two base stations for configuration, we use a second station just for the configuration and turn it off afterward. For both measurements, we measure positions in a 6x3m grid, where the base station is located in one corner, about 1m away, and 2.8m above the configuration point. Due to reachability issues with the second station, we use the (1,1,0) coordinate as the configuration point for the normal estimation method. For the

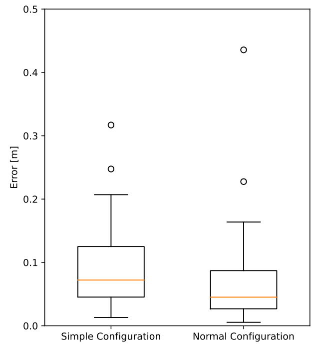
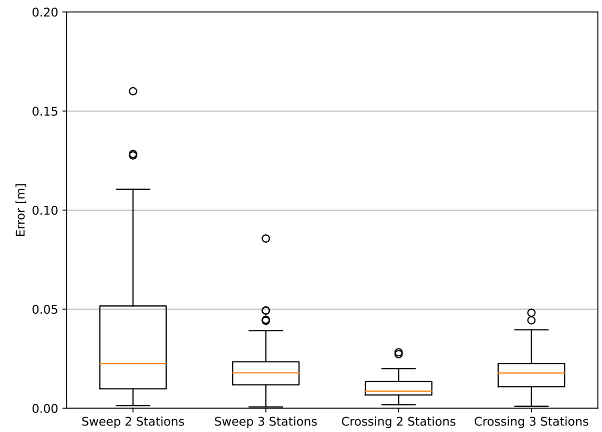
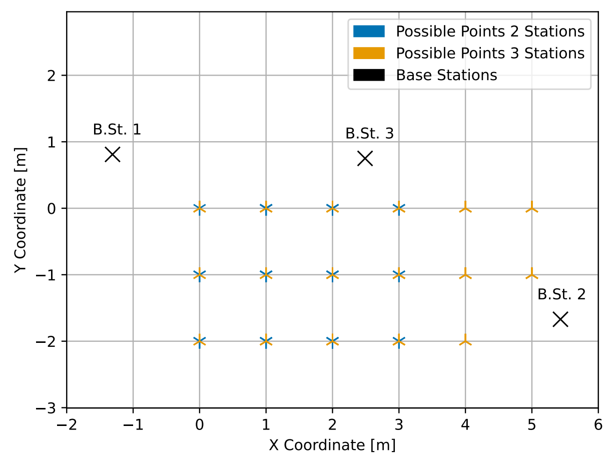


Fig. 9: Accuracy with one station. The normal configuration method, with an average error of 4.5cm, performs better than the simple configuration method, which has an average error of 7.3cm. The 75th percentiles are 12.6cm and 8.7cm.



(a) Accuracy with more stations.



(b) Range with more stations.

Fig. 10: Accuracy and range with more stations. Figure (a) shows the accuracy for two and three stations. It can be seen that sweeping gets better with a third station, whereas this is not the case for crossing. Figure (b) shows the range improvement for crossing when a third base station gets added.

measurement values, we record 1000 values on every meter point of the grid. We take the recorded values' average for the evaluation, subtract the reference point, and create the Euclidean norm for the absolute position.
The results in the boxplots of Figure 9 show an average positioning error of 7.3cm for the simple configuration and 4.5cm for the normal configuration. 75% of the measured points are within a range of 12.6cm for the first configuration. For the second configuration, the 75% mark is at 8.7cm. Especially measurement points more than 4m away from the base station get very inaccurate. The reason for the simple configuration being worse than the normal configuration is likely twofold. First, only one specific position is measured in the configuration. The system derives the base station's location from this position, so there is no opportunity for correction if this is a bad point. Second, there may be a bad alignment with the reference coordinate system. As described in Section II-E, aligning the simple configuration method to a reference system is hard. However, by creating a pixel map showing the measured values compared to the reference points, we ensure this is not the case for the measurement. There we see that the measurement points do not follow a rotation of the coordinate system. It looks more like a distortion and does not depend on a configuration error. To summarize, a system with only one base station does not fulfill our accuracy requirement, as 25% of the positioning points have errors of 12.5-20.6cm for the simple configuration and 8.7-16.3cm for the normal configuration, which exceeds the 10cm accuracy requirement.

4) Accuracy and range with more base stations: This paragraph shows the accuracy of two and three base stations. Additionally, we show how the area of the lighthouse system changes when an additional station is present. For our measurements, we use the same setup as above, a 6x3m grid where the first two base stations are located in corners opposite each other. When we use three base stations, we locate the third base station in the middle of the 5m edge to see if this improves accuracy or covered area. We record 1000 values on each point for the measurements and use the mean values for the boxplots. To get height information, we measure at 0, 0.6, 1.5, and 2m. In addition to that, we also log the minimum and maximum values on each point.
We show the results for these measurements in Figure 10. In Figure 10a, the two left boxplots correspond to the sweeping method with two and three base stations, whereas the two right boxplots present the results for crossing. In relation to the results from one station (Figure 9), the accuracy is better, especially when using the crossing method. This method's mean error is 0.8cm for two stations and 1.7cm for three. Additionally, all points are measured with at least 3cm accuracy. Sweeping with two stations also shows an improvement compared to one station. The mean error reduces from 4.5cm for one station to 2.2cm for two. Adding a third station is also beneficial for the sweeping method, as it makes the system even more accurate and achieves a mean error of 1.7cm. However, crossing with a third station does not improve the system anymore. That might be because of stations that hardly reach some measurement points and then introduce an additional crossing calculation. As shown in Figure 7d, this results in a Gaussian distribution with a bigger offset than the actual reference point. Therefore, filtering the received crossing points based on the nearest two stations could improve the system for crossing with more stations. Figure 10b presents the crossing range improvement from the third station. This plot presents

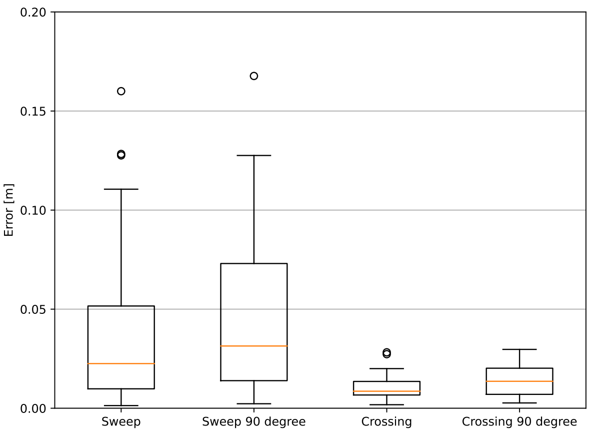


Fig. 11: Impact of the measurement angle. A 90° rotation of the drone does not really influence the accuracy, as the points are still in the same domain of 17cm for sweeping and 3cm for crossing.

the points that can be measured with two or three base stations. The evaluation is combined for all heights that have been measured, so a green marker without a red marker means that for all the different heights, no value for two base stations is measurable, but it is for three base stations. The plot shows additional measurable points on the right side, which are not reachable from the first base station. With the added station three and the existing station two, these points can be evaluated for crossing. To sum up, the accuracy with more stations is better than with one. Adding a station can extend the range, and the sweeping accuracy improves. The crossing method is accurate enough for the reference system, as its accuracy is within 3cm for two and 5cm for three stations.

5) Impact of the measurement angle: With this experiment, we want to show if the accuracy of a 90-degree rotated drone matches that of a drone with no rotation. The reason for this experiment is that the measurement with two base stations received more accurate x-coordinates than y-coordinates. Therefore, this leads to the observation that the asymmetric positioning deck could impact the accuracy of either the x or y direction. As a setup, we use the same as for the measurement with two base stations.

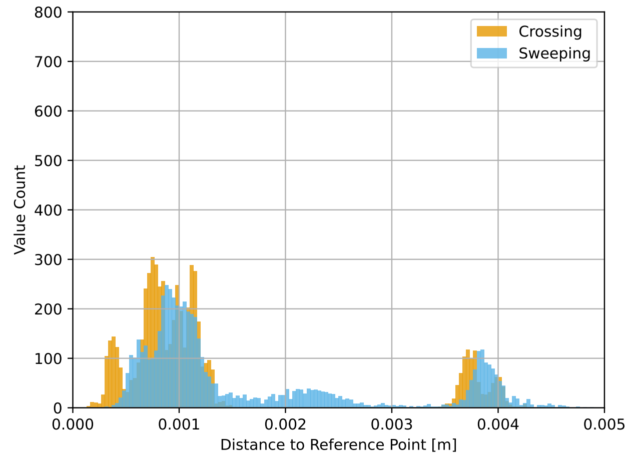
The results show that the resulting measurement values do not differ much from each other, no matter if the drone is 90-degree rotated or not. We show the absolute errors in Figure 11: We observe a mean error of 2.50cm for sweeping without rotation and a mean error of 3.13cm when the drone is rotated by 90 degrees. For crossing, the corresponding mean errors are 0.86cm and 1.35cm. Further investigation of our measurement values shows that the source for x-accuracy being not the same as y-accuracy in this specific measurement setup has nothing to do with the drone's rotation. To summarize, the drone's yaw does not impact the accuracy, as all measured values stay within 13cm for sweeping and 3cm for crossing, no matter which direction the drone faces.

6) Precision Depending on Configuration: As we use the normal estimation method for configuration for most of our experiments, we want to show the reproducibility of a configuration. Therefore, we look at the distribution of the same measured points with different configurations. We use three base stations for the experimental setup, configure the system six times, and save the configurations in six different files. For each of the configurations, we use the same origin and 1m point, as this is the base for scaling and positioning. Within the six configurations, we vary the number of location points and their locations on the ground plane. For the measurement, we log 1000 positioning values for every configuration on the same point.

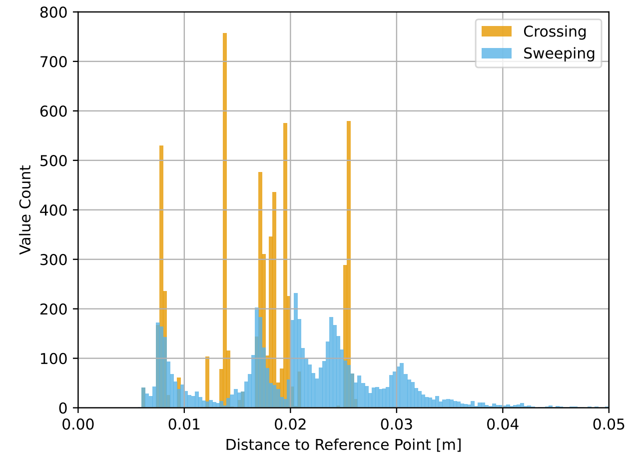
We show the distributions of the points (0,0,0) in Figure 12a and (1,0,1.5) in Figure 12b. For the origin, the whole distribution is within 5mm. However, this is not the case for the point (1,0,1.5), as we observe a distribution in a range of 2cm for crossing and even worse for sweeping, where the measurement values are mainly distributed in a range of 4cm. The reason for the distribution being that high might be because of our approach to use bad configurations as well, where the plane is hard to calculate, as the configuration point for the plane is nearly on the same axis as the origin and 1m point. We show the corresponding Cumulative Distribution Function (CDF) for the origin point in Figure 12c, which shows that the sweeping method has a worse precision than crossing. One of the configurations (Config3 from Figure 12c) has an offset of about 3mm to the other ones. However, this range is not crucial as it is still very precise. Figure 12d shows the CDF for the point (1,0,1.5). It shows that every spike in the figure above belongs to an individual configuration. In general, the crossing methods behave more accurately than sweeping, as the distances of the solid lines are nearer to the reference point. To sum it up, the precision of different configurations is within a range of 5cm for sweeping and 2.5cm for crossing, demonstrating that a reconfiguration of the system can result in positioning values that are shifted by these amounts.

7) Robustness (disturbance from light): In this experiment, we want to show if the system suffers from surrounding light sources. In the presentation of LighthouseV1 [18], it gets mentioned that on a surface 5m away from the base station, the laser's signal is approximately 1000 times weaker than the infrared light from the sun. This fact restricts the system to indoor usage, but there can also be some disturbance sources. In our setup for this experiment, we use two base stations that are about 2.5m away from the drone. As disturbance sources, we use a 700lm headlamp and a 100W infrared lamp. We place the disturbance sources approximately 10cm away from the drone and emit light to the top of the receiving board. First, we start with a reference measurement and then use one disturbance at one time. We record 1000 localization values for every measurement. Figure 13 presents our results. The headlamp disturbance does not have any impact on the accuracy. That is because the lighthouse uses 850nm light, which is out of the visible light area (400-700nm). On the other side, infrared light impacts the system's accuracy. The sweeping method disturbed with infrared light (Figure 13a) calculates positions that are about 1mm off. Additionally, there are some outliers in the range of 1cm, but they occur seldomly (8 out of 1000). For the crossing method, the results are slightly different. We show them in Figure 13b. For this method, the infrared light leads to two different localization positions. One of the two positions is about 0.5mm shifted, whereas the other one is 8mm away from the reference measurement. Additionally, the update rate drops. We show this in Figure 14, which shows that the configured 100ms updates only occur 83% of the time. 97% of updates with this disturbance happen within 300ms, and we also discover outliers of up to 2.6s. To sum up this section, it is important to pay attention to possible infrared disturbances, as these can affect the accuracy and update rate of the measurements.

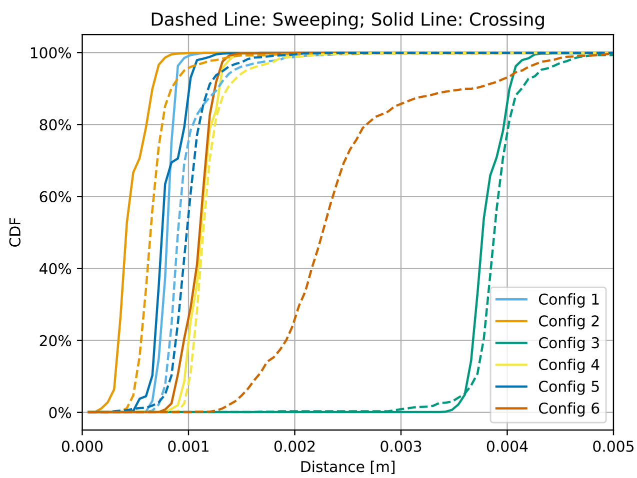
8) Robustness (optical disturbances): This part shows how the system behaves when something disturbs the line of sight connection from the base station to the receiving photodiodes. To discover this, we perform two experiments. In the first experiment, we showcase a situation when something shuts off one base station during operation.



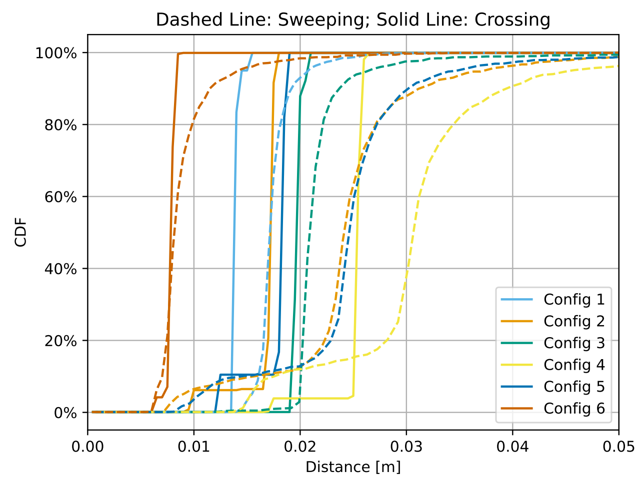
(a) Configuration deviation of origin.



(b) Configuration deviation of (1,0,1.5).



(c) CDF of origin.



(d) CDF of (1,0,1.5).

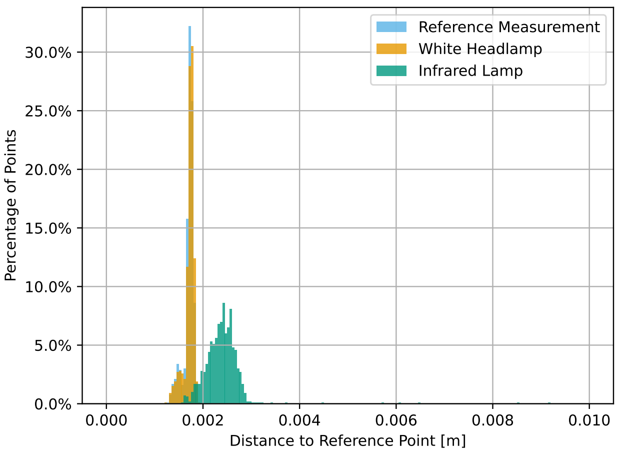
Fig. 12: Precision of different configurations. The different measurement values of all the configurations are within 5mm for the origin point and 4cm for the point (1,0,1.5). Sweeping performs worse than crossing. Figures (c) and (d) represent the corresponding CDFs for (a) and (b). They show that each spike of the upper plot corresponds to one configuration.

For this, we build a setup with two base stations and then disconnect one base station from its power supply during operation. We log the positioning data with the `cfclient`, which is the Graphical User Interface (GUI) for the crazyflie [19].

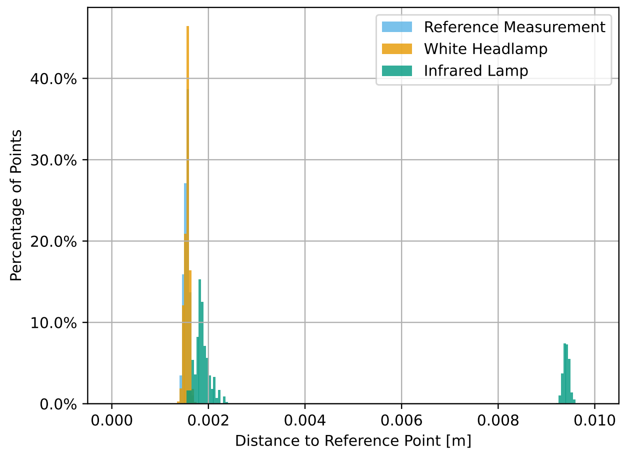
When disconnecting the second base station, we discover that the localization data gets more imprecise, which means that the single positioning values suffer from more noise. We show this in Figure 15, where the received data gets noisier after the dashed line, representing the disconnection point. Before disconnecting, we receive localization data with noise in the range of about 1mm. When only one station is received, this noise range increases to 1cm. Furthermore, the accuracy also gets worse, as the received average shifts by about 5mm. When we reconnect the station, the system improves in accuracy and precision like before the disconnection.

In the second experiment, we cover one receiving photodiode with dark adhesive tape, which could happen if something smaller disturbs the line of sight. We used two base stations for this experiment and started covering one photodiode. The system no longer works when we do this, as the crazyflie performs no positioning updates. That leads to the fact that the lasers from the base stations have to be received by all four diodes. If no updates on the position happen when not flying, the extended Kalman filter becomes unstable and is no longer usable for localization information. To summarize, receiving one instead of two stations does not break the system. It just lowers accuracy and precision. However, no further positioning is possible if one diode gets covered somehow. During flight, this could happen if several drones are operated in the same area, which could then lead to small position drifts, until all 4 diodes receive data again. Therefore, using a second station for this case would remedy this case.

9) Optimization Measurements: We perform next measurements with different system setups that use two base stations. Together with the manufacturer's recommended setup, we tried additional setups. Some of them can also be used in bigger systems. We use a 3x3m grid for this experiment and measure at three different heights. For the



(a) Disturbance of sweeping.



(b) Disturbance of crossing.

Fig. 13: Light disturbances. The system is not affected by a 700lm white light headlamp. A 100W infrared light influences the system, which results in an accuracy shift of 1mm for sweeping. Two shifts of 0.5mm and 8mm occur for crossing.

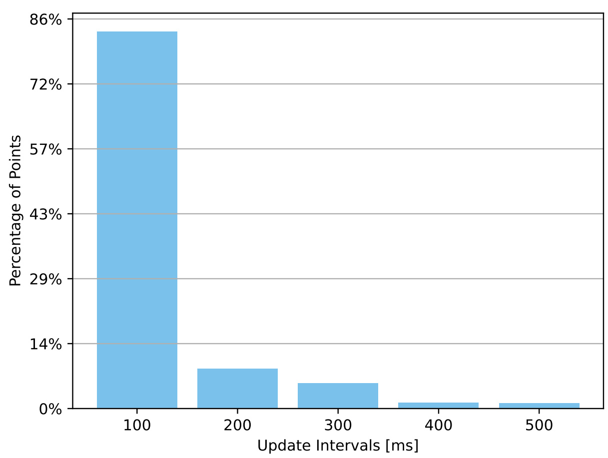


Fig. 14: Update rate for infrared disturbance. The update rate worsens when infrared light shines on the receiving board.

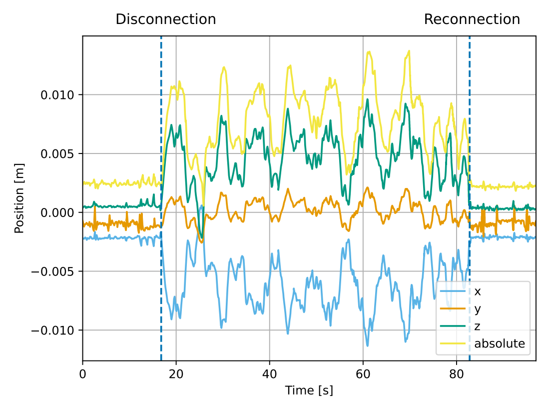


Fig. 15: Received positioning values after disconnecting and reconnecting the second base station. Precision and accuracy get worse after the disconnection: this event is marked by the first dashed line. After reconnecting the second station, the precision and accuracy get better, as before the disconnection. The second dashed line marks the reconnection.

evaluation, we use 1000 values on each point and take their average. We present the individual setups in Figure 16. The base stations are located at the height of approximately 2m. All base stations are tilted by about 45 degrees to the ground, and the orientation is the same as the lines starting from the stations in Figure 16. Only the base stations from Setup5 in Figure 16 are tilted down by 90°, facing the ground underneath them. The top of these stations points to the positive y direction. According to the different distributors of the lighthouse, Setup1 or Setup2, where the base stations are located in corners next to each and in opposite corners other should achieve the best results. We present our results in Figure 17. The corresponding 25th percentiles, mean errors, and 75th percentiles of the boxplots are represented in Table I. For these measurements, sweeping and crossing are quite similar. That could be because the base stations are just a short distance away from each other. Setup1 and Setup5 perform best for sweeping and crossing. The setups have a mean localization error of 1.5cm and 1.6cm for sweeping and 1.4cm and 1.5cm for crossing. Therefore, the recommended usage for two base stations is to place the base stations in two corners next to each other. That matches the recommendation of the manufacturer. Setup3 and Setup4 perform worst, which leads to the fact that the base stations should not be placed half a meter above or next to each other. To configure a bigger working area, Setup5, Setup6, or Setup7 can be used, as these do not focus on the same point in the middle. That makes configuring more stations that are lined up easier, and dead spots can be avoided too. Out of these three setups, Setup5 performs best. It has an average error of 1.6cm for sweeping and 1.5cm for crossing. Additionally, all points we measure with this setup have an error of less than 3cm. To summarize this part, Setup5, where the base stations face the ground from above, is generally an easy and good setup.

10) Accuracy in large scale deployments: We perform two measurements using more than three base stations to cover a larger area. First, we use the best multi-station setup from Section III-B9 and want to verify the accuracy when hopping over stations during

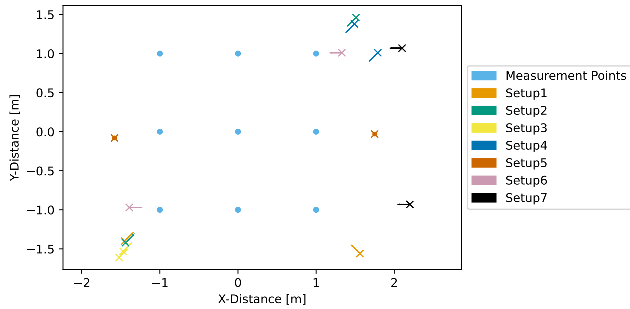
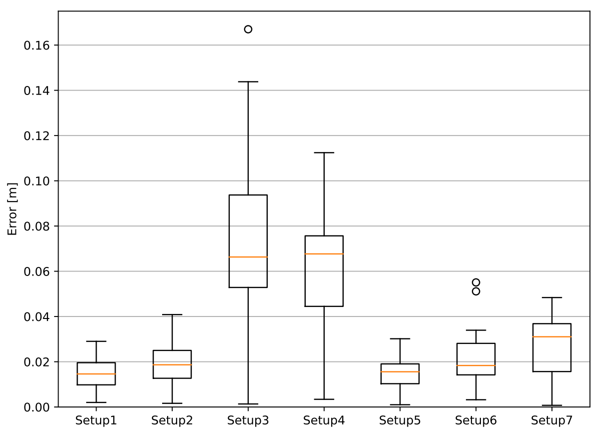
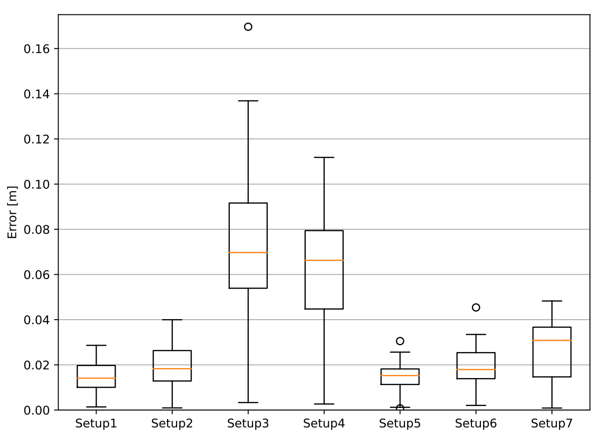


Fig. 16: Setups for optimization measurements. It contains the measurement points and the different setups we use for finding the optimal one. The heading of the stations is represented by the lines pointing away from the stations.



(a) Optimization for sweeping.



(b) Optimization for crossing.

Fig. 17: Optimization measurement results. Setup1 and Setup5 from Figure 16 perform best. Setup3 and Setup4 should be avoided as they produce the worst results.

configurations. Therefore, we place four base stations in a straight line with 2.5m between them. We mount the stations at a height of 2.1m, and all of them are adjusted to face the ground underneath. The origin configuration point for this measurement is located under

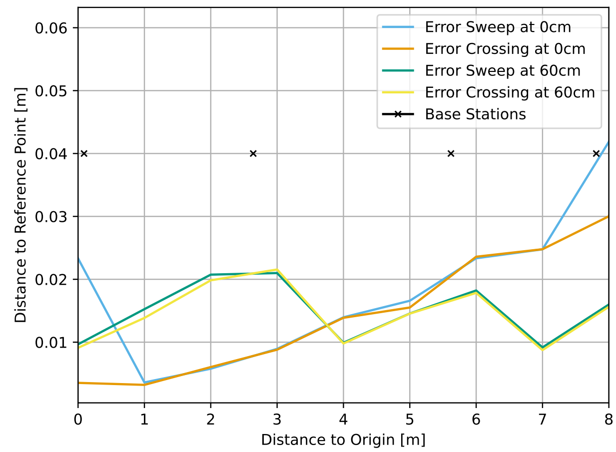


Fig. 18: Four stations in a line. It shows how the accuracy behaves when the distance to the configuration point gets higher. It can be seen that the accuracy gets slightly worse with more distance to the origin.

one of the outer base stations. We record 1000 localization values for the measurement and take the mean value for the evaluation.

We present the results in Figure 18. It can be seen that the error rises when the distance to the origin gets bigger. The plot also shows that the crossing method performs marginally better. The black markers in the plot show the x-location of the base stations. In general, even if the accuracy drops over distance, the error is still within an acceptable range and does not increase significantly.

In our second experiment, we used six base stations to cover a part of the HALLWAY from Figure 5. Due to placement issues, we initially tried to use Setup6 from Figure 16, as Setup5 is impossible because the HALLWAY is not equipped with a mounting rail in the middle. However, we discovered additional placement restrictions during the setup, which led to using a combination of Setup6 and Setup7 from Figure 16. We show the final locations of the base stations in Figure 19. All the stations are tilted down by about 60 degrees.

During configuration, we observed some further issues. When receiving more than four base stations at one time, the crazyflie can, by design, not receive any of them. Therefore, no position update is possible in these areas, as it is not supported in the crazyflie

Setup/Method	25% Mark [m]	Mean Error [m]	75% Mark [m]
Setup1/sweeping	0.010	0.015	0.020
Setup1/crossing	0.010	0.014	0.020
Setup2/sweeping	0.013	0.019	0.025
Setup2/crossing	0.013	0.018	0.026
Setup3/sweeping	0.053	0.066	0.094
Setup3/crossing	0.054	0.070	0.092
Setup4/sweeping	0.045	0.068	0.076
Setup4/crossing	0.050	0.066	0.079
Setup5/sweeping	0.010	0.016	0.019
Setup5/crossing	0.011	0.015	0.018
Setup6/sweeping	0.014	0.018	0.028
Setup6/crossing	0.014	0.018	0.025
Setup7/sweeping	0.016	0.031	0.037
Setup7/crossing	0.015	0.030	0.037

TABLE I: Optimization measurement results.

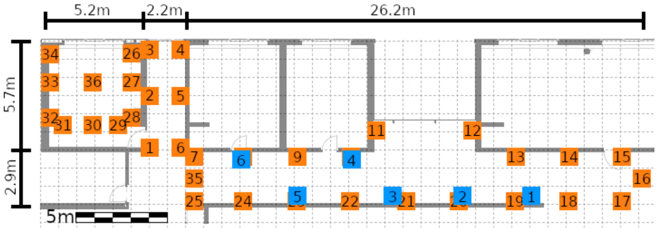


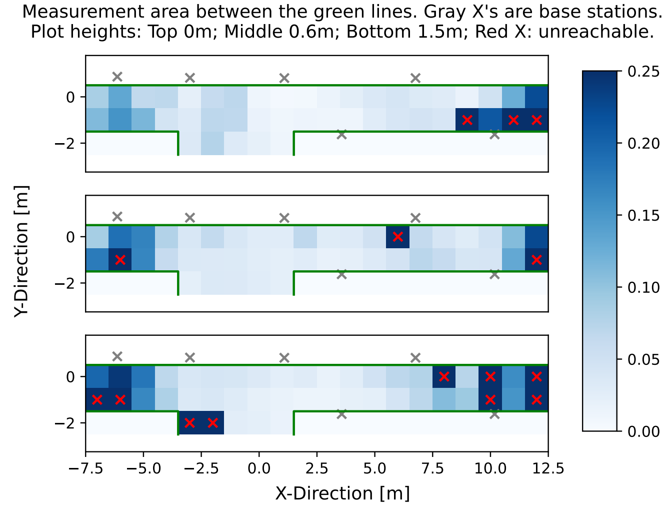
Fig. 19: Setup with six base stations within testbed. The blue squares represent the positions of the base stations within the existing testbed.

architecture. To avoid this problem, we adjust all the outer base stations so every point in the working area receives at most four base stations. Furthermore, we initially tried to use the top left point in Figure 20 as the origin for the configuration. After several tries to configure the system, it turned out that this is impossible, because it seems that the crazyflie cannot calculate base station positions that need more than one base station hop of a station that is not visible at the origin configuration point. Therefore, we use a point somewhere in the middle as the origin. These two facts make this system hard to set up for more than four base stations. For the measurement, we record 1000 values and use the mean value for the evaluation.

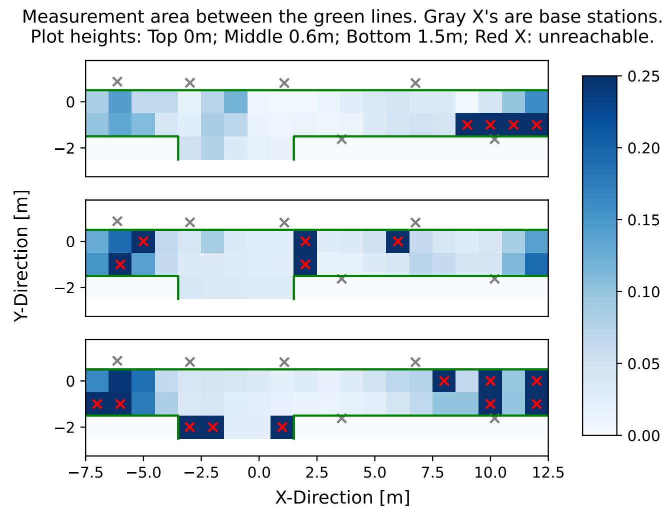
We present our results in Figure 20. In Figure 20a, we show the sweeping evaluation. It shows that the system performs better in the inner than the outer areas. That applies to all three heights. The main reason for this should be the side-tilting of the outer stations to ensure positioning updates with at most four base stations. Due to this, the outer stations do not cover all the nearest measurement points. The crossing method behaves similarly to the sweep. However, crossing suffers more in the outer areas, which should come from tilting the base stations, as crossing needs at least two stations to perform an update. The points at the x coordinate of 2m and a height of 0.6m are points where no proper beams were received. We do not know the reason for this appearance. To summarize, a system with six stations is possible, but there are some restrictions due to the tilting of the stations. The inner section achieves an accuracy of at least 5cm, which is good, but the outer section is not usable. The main limiting factor for a bigger system is that the positioning data gets lost if more than four stations are received in one update cycle. Therefore only 4 stations can be used without problems that lead to side-tilting the outer stations.

IV. Conclusion and Future Work

In this work, we evaluate LighthouseV2 to verify if it offers the requirements for the usage as a reference system for the IoT. We base our evaluation on dedicated experiments, which all cover different aspects that are important for the testbed. The system's latency lies within 40ms for 90% of all time. It has some outliers, but they rarely occur and are within 100ms. In terms of accuracy, a system with only one base station is within a range of 15-21cm. With more than one station, the system provides positioning values with an accuracy error that is mainly in a range of 5cm. Furthermore, an accuracy of up to 3cm is possible with optimal setups. The precision of the localization values on good visible points is within a range of 2mm. Configuring the system can introduce an additional error of 2cm if not done carefully. The yaw of the drone does not have an impact on the accuracy. Regarding disturbances, infrared light can harm the system's accuracy. However, this disturbance source must be either very strong or close to the receiving board. Furthermore, when avoiding the mentioned flaws of the six-station experiment, it is possible to scale up the system with additional base stations. These points lead to the following evaluation. The LighthouseV2 with only one base station is not usable for a mobility-enhanced testbed, as the accuracy is too bad with a range of over 10cm. However, with more base stations, it is possible to use this system as a reference system in a testbed for the IoT. The crossing method should be used for this, as it outperforms sweeping in most of our experiments. Additionally, this ensures that at least two stations are received, which positively affects the accuracy.



(a) Large scale deployment with the sweeping method.



(b) Large scale deployment with crossing.

Fig. 20: Big setup with six base stations. Each square contains one measurement point of the measurement grid. The inner area performs well, whereas the outer area does not. The limit from which the points are marked as unreachable is 25cm.

A. Future Work

Dynamic tests. We base all of our tests on a static placement with the help of tripods. It would be nice to know if the system performs as accurately while moving or if that impacts the accuracy.

Blind spots. Another point that we discovered while measuring is blind spots. These are points where the receiver cannot get any localization updates. However, when moving the drone about one millimeter to the side, the system works again. We do not know the reason for this behavior. It would be interesting to inspect if this comes from reflection, interference, or similar. A starting point for this would be to check if the lighthouse's case limits the radiance of the emitted infrared beam. This could be done with a precise and fast enough infrared camera or a self-built infrared receiver with a photodiode. This hardware could then also be used to record any received infrared beams at specific blind spots. The beam's intensity could then give information if the beam is a reflection or something similar.

Robustness adaptions. One significant bottleneck of the system is that it cannot receive more than four beams from different stations during the same update cycle. Therefore, it is hard to set up more than four stations in a bigger setup within one room. We only scale the system in one direction and observe problems with it. Therefore, the setup should be even harder, especially when simultaneously scaling into the x and y directions. Adapting the system to be able to receive more than four base stations at the same time should offer better scalability with better accuracy. Therefore the FPGA and the software from the crazyflie need to be extended. One solution would be to process the first four received beams only so that a positioning update is still performed if there are more than four stations. The second solution for this would require the FPGA and the software of the crazyflie to be extended to calculate positions with up to 16 base stations. This would probably require a bigger FPGA and a more powerful CPU.
Another flaw of the system is that the encoded laser beam has to be received by all four photodiodes. If not, the system no longer performs positioning updates, and the extended Kalman estimator gets unstable. The system could improve in robustness if it gets adapted so it does not need all photodiodes for localization updates. This would require a software modification from the crazyflie. The sweeping method should still be possible with three sensors receiving, whereas crossing should also work with only two sensors, which could be stripped down to one sensor if the heading is unimportant.

Configuration methods. Configuring the system with more base stations can get complicated, as the configuration software cannot calculate the base stations' positions in a bigger system. That especially occurs when the calculation has to be done via hopping a station that does not see the origin point of the configuration. Therefore, a configuration with multiple premeasured points on the ground would be a nice feature, making configuring bigger systems easier. In addition to that, accuracy might also improve, as there is no error propagation when hopping stations. That should lead to more accurate configuration positions, especially for base stations far from the reference point.

Different hardware architecture. The last thing we want to mention is to redesign the receiving hardware. As the photodiodes of the current version sit on a flat board, it is physically impossible to receive beams when the board is tilted away from a base station. Therefore, a trigonometric sphere with multiple photodiodes on its surface could solve the problem. With this setup, it would be possible to receive base station beams all the time, no matter how the hardware is tilted or placed. Additionally, it is not required to receive beams with all photodiodes. It will be enough if a threshold value of beams is received so that the position gets calculated from them.

References

[1] M. Schuss, C. A. Boano, and K. Römer, "Moving Beyond Competitions: Extending D-Cube to Seamlessly Benchmark Low-Power Wireless Systems," in *Proceedings of the IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*, 2018, pp. 30–35.

[2] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel, "FlockLab: A Testbed for Distributed, Synchronized Tracing and Profiling of Wireless Embedded Systems," in *Proceedings of the 12th International Conference on Information Processing in Sensor Networks, (IPSN)*. Association for Computing Machinery, 2013, p. 153–166. [Online]. Available: https://doi.org/10.1145/2461381.2461402

[3] M. Doddavenkatappa, M. C. Chan, and A. L. Ananda, "Indriya: A Low-Cost, 3D Wireless Sensor Network Testbed," in *Testbeds and Research Infrastructure. Development of Networks and Communities*, T. Korakis, H. Li, P. Tran-Gia, and H.-S. Park, Eds., 2012, pp. 302–316.

[4] M. Schuh, H. Brunner, M. Stocker, M. Schuß, C. A. Boano, and K. Römer, "First Steps in Benchmarking the Performance of Heterogeneous Ultra-Wideband Platforms," in *Proceedings of the Workshop on Benchmarking Cyber-Physical Systems and Internet of Things (CPS-IoTBench)*, 2022, pp. 34–39.

[5] D. Molteni, G. P. Picco, M. Trobinger, and D. Vecchia, "Cloves: A Large-Scale Ultra-Wideband Testbed," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 808–809.

[6] M. R. Mahfouz, C. Zhang, B. C. Merkl, M. J. Kuhn, and A. E. Fathy, "Investigation of High-Accuracy Indoor 3-D Positioning Using UWB Technology," *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, no. 6, pp. 1316–1330, 2008.

[7] Inc. DBA OptiTrack, "OptiTrack - Motion Capture Systems," 2023, [Online] https://optitrack.com/ – Last access: 2023-03-14.

[8] ——, "OptiTrack - Motion Capture for Movement Sciences," 2023, [Online] https://optitrack.com/applications/movement-sciences/ – Last access: 2023-03-24.

[9] Bitcraze AB, "Lighthouse Positioning System," 2023, [Online] https://www.bitcraze.io/documentation/system/positioning/ligthouse-positioning-system/ – Last access: 2023-03-13.

[10] ——.

[11] ——, "Crazyflie 2.1," 2023, [Online] https://store.bitcraze.io/products/lighthouse-v2-base-station – Last access: 2023-03-15.

[12] ——, "Lighthouse Positioning Deck," 2023, [Online] https://www.bitcraze.io/products/lighthouse-positioning-deck/ – Last access: 2023-03-15.

[13] M. Schnegg, "Système de positionnement autonome basé sur la technologie de HTC Vive," Bachelor's Thesis, HEIG-VD, 2019-2020.

[14] Bitcraze AB, "Crazyflie 2 - lighthouse-4 deck," 2018, [Online] https://www.bitcraze.io/documentation/hardware/lighthouse_deck/lighthouse_deck-revd.pdf – Last access: 2023-03-13.

[15] Siemens Semiconductor Group, "BPW34S Datasheet," 2008, [Online] https://www.alldatasheet.com/datasheet-pdf/pdf/44655/SIEMENS/BPW34S.html – Last access: 2023-03-13.

[16] C. Lohr et al., "Let's try figuring out the data stream." 2018, [Online] https://github.com/cnlohr/esptracker/issues/1 – Last access: 2023-03-13.

[17] A. Taffanel, B. Rousselot, J. Danielsson, K. McGuire, K. Richardsson, M. Eliasson, T. Antonsson, and W. Hönig, "Lighthouse Positioning System: Dataset, Accuracy, and Precision for UAV Research," *arXiv preprint arXiv:2104.11523*, 2021.

[18] A. Yates, "Alan Yates on the Impossible Task of Making Valve's VR Work," 2017, [Online] https://www.youtube.com/watch?v=75ZytcYANTA&ab_channel=HACKADAY – Last access: 2023-03-17.

[19] Bitcraze AB, "Userguide Crazyflie Client GUI," 2023, [Online] https://www.bitcraze.io/documentation/repository/crazyflie-clients-python/master/userguides/userguide_client/ – Last access: 2023-03-17.