

HTC Vive as a Ground-Truth System for Anchor-Based Indoor Localization

Laura Fluera^{toru}*[†], Elena Simona Lohan[†], Jari Nurmi[†] and Dragoș Niculescu^{*}

* Computer Science Department, University Politehnica of Bucharest, Romania

[†] Electrical Engineering Unit, Tampere University, Finland

Abstract—The last decade has seen a surge in the popularity of indoor localization systems. Researchers and companies keep searching for technologies that can locate users on a large scale with low costs and the highest possible accuracy. When evaluating the accuracy of a localization system, there is a trade-off between the cost and labor involved in acquiring ground-truth measurements. The cheapest option is to acquire measurements in fixed spots and manually compute their true location in a local coordinate system using distance measuring tools. However, this method is prone to human errors and has a high setup overhead. In contrast, high-end motion capture systems are easy to set up but have prohibitive prices. A middle-of-the-road solution is to use a consumer-grade motion capture system such as the HTC Vive which, although designed for virtual reality video games, can be adapted for scientific applications. We propose a ground-truth system for anchor-based indoor localization systems which builds on the HTC Vive and we demonstrate its use on ultra-wideband (UWB) localization. We apply Procrustes Analysis to bring location data sets into the coordinate system of a room, in order to easily overlap, visualize, and analyze measurements. We use the HTC Vive to acquire the locations of UWB anchors, which allows users to quickly test which hardware placement yields the lowest localization error. The resulting ground-truth system costs under \$1000, has an average accuracy of more than 5 mm, is easy to set up, and can be used for both static and dynamic measurements.

Index Terms—Localization system, Ultra-Wideband (UWB), Virtual Reality (VR), motion capture system.

I. INTRODUCTION

Indoor localization systems (LSs) are increasing in popularity on the consumer wearable market, being used in smart home, indoor navigation, and assisted living applications. Most of the popular wireless communication technologies such as Wi-Fi, Bluetooth, RFID, and Ultra-Wideband (UWB) double as viable localization technologies with accuracy ranging from several centimeters to a couple of meters [1].

One way to compute the location of a tracked device (called *tag*) is based on messages exchanged between the tag and several devices called *anchors* with fixed, known locations. The location of the tag can be computed using the distance between the tag and the anchors via trilateration, where distances can be estimated using the time of flight (TOF) [2] or received signal strength (RSS) [3] of a signal. The location of the tag can also be computed using the angle of

arrival (AOA) [4] or the time difference of arrival (TDOA) [5] of signals sent by the anchors to the tag (or the other way around). We classify these techniques as *anchor-based*, since the coordinates of anchors are directly used in the tracking algorithm to compute the location of the tag.

When reporting the localization accuracy of these technologies, there is a trade-off between the cost and the labor involved in obtaining ground truth measurements. The cheapest method is acquiring measurements only in fixed spots, whose locations are manually computed using high-precision distance measuring tools. However, this method is prone to human errors when measuring distances, is very laborious, needs to be redone in every room in which measurements are acquired, and cannot be used for objects in motion. In contrast, high-end motion capture systems with millimeter-level accuracy and precision such as OptiTrack and Meta Motion are easy to set up and use but cost tens of thousands of euros per testbed.

The HTC Vive¹ is a consumer-grade motion capture system designed for VR video games, with mm- to cm-level tracking accuracy [6], [7]. Compared to high-end motion capture systems, the HTC Vive is significantly cheaper and covers a larger area, which makes it an affordable candidate for obtaining ground truth measurements.

In this paper, we propose a framework to use the HTC Vive as a ground-truth system for anchor-based indoor localization systems and we demonstrate its use for UWB localization. The proposed system satisfies the following criteria: (1) the user does not need to manually compute the locations of fixed points inside a room, (2) location data sets acquired in the same room share the same coordinate system so that they can be easily compared, and (3) the coordinate system of a location data set is independent of the hardware placement. In this way, users can test multiple hardware configurations inside the same room and easily compare results from one day to the next.

We realize the proposed goals by using the HTC Vive to acquire (1) the locations of a set of reference points, to make all recordings independent of the hardware placement inside the room and (2) the location of UWB anchors, to place UWB measurements in the Vive coordinate system. We apply Procrustes analysis to align data sets obtained in the same room to a reference coordinate system, so that they can be easily compared.

¹We will alternatively refer to the HTC Vive simply as “Vive.”

The authors gratefully acknowledge funding from European Union’s Horizon 2020 Research and Innovation programme under the Marie Skłodowska Curie grant agreement No. 813278 (A-WEAR: A network for dynamic wearable applications with privacy constraints, <http://www.a-wear.eu/>).

In [6], [7], the authors analyzed the accuracy, precision, and latency of the HTC Vive and its potential use as a ground truth system. However, they did not provide a framework for using it as a ground truth system in applications, which is the focus of our work. The authors found that the accuracy of the HTC Vive varies from several mm to several cm and therefore concluded that it is unsuitable as a ground truth system in robotics². However, it can still be useful for other applications. It is commonly accepted that the average localization error of a ground truth system should be at least an order of magnitude lower than the one of the target system. Given that Wi-Fi and Bluetooth have dm- or even m-level accuracy [8]–[10] and UWB has an average 2D accuracy of 10–20 cm [11], [12], the HTC Vive can be a viable ground-truth system for these technologies. In addition, our calibration method can check whether localization errors remain within an acceptable threshold when compared to a reference data set acquired in ideal conditions.

Another affordable ground-truth systems for robotics using the Microsoft Kinect was proposed in [13] but it has an average error of around 10 cm (thus unsuitable for UWB localization). Most of the indoor localization technologies are evaluated using the manual approach [14] or using expensive laser scanners³ [15] or robotics test beds [16]. Therefore, an easy-to-use, accurate, and affordable ground-truth system for indoor localization applications is still needed, which is the scope of our work.

The contributions of this paper are:

- We propose a relatively cheap and accurate ground truth system for anchor-based indoor LSs.
- We apply Procrustes analysis to align location data sets to a coordinate system characteristic to a room, so that we can easily overlap, analyze, and visualize the data.
- We describe how to use the HTC Vive to obtain the location of UWB anchors for fast prototyping of anchor arrangements.
- We analyze the accuracy of the HTC Vive, how to choose the reference points needed by the Procrustes algorithm, and the accuracy of the UWB LS as given by the ground-truth system. In addition, the ground-truth system can also record the orientation of the target device, which can be used to obtain more insights into the target LS.

The rest of the paper is organized as follows. We briefly describe the system components in Section II and the algorithms used by the ground-truth system in Section III. We present the system setup and how to use it in Section IV. Finally, in Section V we evaluate the accuracy of the HTC Vive and the UWB system.

II. SYSTEM OVERVIEW

A. Ultra-Wideband signals

Owing to their large bandwidth, Ultra-Wideband signals have a very high time resolution which can be leveraged

²In rare cases, the HTC Vive returned errors of a couple of meters [6].

³Such a system was used in the Microsoft Indoor Localization Competition 2018.

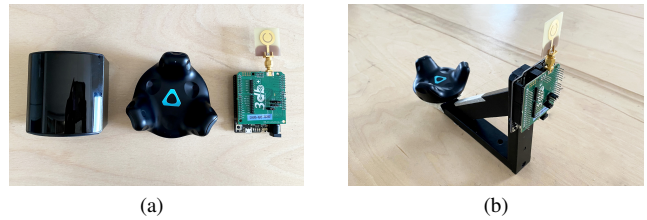


Figure 1: (a) The main hardware components of the proposed system (left to right): HTC Vive base station, HTC Vive tracker, and 3db Access UWB device. (b) Stand holding the tracker and the UWB tag.

to measure the time of flight (TOF) of a signal with sub-nanosecond accuracy. We use TOF measurements to compute the distance between three or more reference devices with known locations (called *anchors*) and the tracked device (called *tag*) and then compute the location of the tag using trilateration algorithms. An UWB system using this method has a typical 2D accuracy of 10–20 cm [1].

In this study, we used UWB devices developed by the Swiss company 3db Access, which perform centimeter-level accuracy ranging and have ultra-low power consumption, suitable for the wearables market. Our setup consists of four UWB anchors and one tag.

B. Virtual Reality (HTC Vive)

HTC Vive is a laser-based tracking system intended for VR video games developed by Valve and HTC. It consists of at least one base station (BS), a headset, a tracker, and other accessories. In this paper, we used two BSs and one tracker (version 2.0).

The BS sweeps laser beams horizontally and vertically across the room with 1900 m s^{-1} at 5 m distance and has a vertical field of view of 110° and a horizontal one of 150° . When the modulated beams hit photodiodes placed on the tracker, the tracker uses information modulated on the beams to compute the AOA of the beam [17]. The tracker combines angle measurements with output from an inertial measurement unit (IMU) to compute its location.

The high accuracy and precision of the HTC Vive system make it an excellent candidate to acquire baseline measurements for the UWB system, whose accuracy and precision are in the range of centimeters. The HTC Vive is especially designed for human motion at high speed, as often required by video games, which suits our application.

Figure 1a shows the main hardware components of the integrated UWB-Vive localization system, in this order: a BS, the tracker, and the 3db Access UWB device which can work as either an anchor or a tag.

III. ALGORITHMS

Using the HTC Vive as a ground-truth system for wireless localization applications poses some challenges. First, we want to make the location coordinate systems of measurements acquired in the same room independent of the locations of

Table I: Anchor locations measured by the HTC Vive in two trials, when the tracker loses connection between the trials.

Location	Trial	x [m]	y [m]	z [m]
Location 1	1	-3.27	1.80	2.79
	2	-3.25	1.77	2.72
Location 2	1	3.95	1.58	2.73
	2	3.91	1.54	2.71
Location 3	1	-1.78	-2.58	2.78
	2	-1.79	-2.58	2.75

anchors and BSs. Therefore, it is useful to translate measurements taken inside the same room but with different hardware placements into the same coordinate system, which is explained in Section III-A. Second, we describe how to translate UWB locations into the Vive coordinate system in Section III-B. Finally, Section III-C proposes an offline method to evaluate the localization error of the target system.

A. Aligning Measurements to a Reference Coordinate System

The 3D coordinates and orientation returned by the Vive tracker are (partly) relative to the location of the BSs. However, we might want to change their location in order to see if a particular arrangement yields a higher localization accuracy. It is useful to have measurements acquired in the same room in the same 3D coordinate system, in order to easily compare the results. Therefore, we need to align the measurements to a chosen reference coordinate system.

In addition, the coordinates and orientation returned by the HTC Vive can change from one day to another, even if the BS and the tracker remain in the same location. Also, the Vive coordinate system changes whenever the tracking is lost, i.e., there is no BS in line of sight (LOS) with the tracker, which is seen as if the tracker were disconnected. This is more problematic in experiments because it means that if the tracker loses its LOS with all BSs during a recording, the reported location of the same physical point will be different before and after the disconnection. To demonstrate this behavior, we measured the coordinates of the same locations in two trials, when the tracker disconnects in between the trials. Table I shows that there can be a difference of even 7 cm between the measured coordinates of the same physical point before and after a disconnection.

To align measurements to a reference coordinate system we use ordinary Procrustes Analysis (PA) which finds the optimal linear transformation between two sets of points acquired in different coordinate systems. Assume that we have two sets of measurements, $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n \in \mathbb{R}^3$ and $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n \in \mathbb{R}^3$, stacked as rows in two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times 3}$. PA first computes the translation and (uniform) scaling components for each dataset: $(\boldsymbol{\mu}_A, s_A)$ and $(\boldsymbol{\mu}_B, s_B)$, respectively. Then, it finds the rotational component [18],

$$\mathbf{T}^* = \arg \min_{\mathbf{T}^T \mathbf{T} = \mathbf{I}_d} \|\tilde{\mathbf{A}}\mathbf{T} - \tilde{\mathbf{B}}\|_F^2, \quad (1)$$

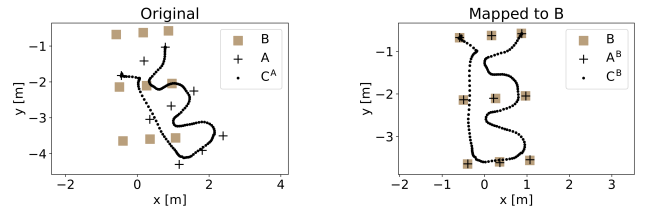


Figure 2: The matrices \mathbf{A} and \mathbf{C}^A before and after being transformed in the coordinate system of matrix \mathbf{B} .

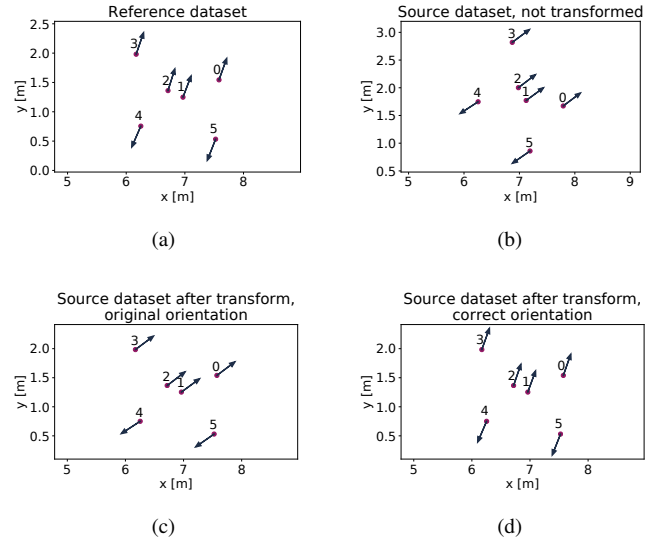


Figure 3: When aligning points from a source data set (Figure 3b) to the coordinate system of a reference data set (Figure 3a), the orientation of the object also needs to be transformed, otherwise the angles will be preserved with respect to the *new* rotated axes (Figure 3c). The correct poses (Figure 3d) can be obtained by converting the rotation matrix \mathbf{T} given by the Procrustes algorithm into the equivalent Tait-Bryan angles, and subtracting them from the object's pose.

where $\|\cdot\|_F$ is the Frobenius norm and $\tilde{\mathbf{A}}$ is the translated and uniformly scaled version of \mathbf{A} , i.e., $\tilde{\mathbf{a}}_i = (\mathbf{a}_i - \boldsymbol{\mu}_A)/s_A$ (analogously for $\tilde{\mathbf{B}}$). To change the coordinate system from \mathbf{A} to \mathbf{B} for a set of points we (i) represent them in the centered and scale-free coordinate system associated with \mathbf{A} by removing the translation and scaling components $(\boldsymbol{\mu}_A, s_A)$, (ii) rotate them using \mathbf{T}^* , and (iii) add back the translation and scaling components of \mathbf{B} , i.e., $(\boldsymbol{\mu}_B, s_B)$.

Figure 2 shows points acquired in the two coordinate systems before and after alignment using Procrustes analysis. Besides the tracker's location, the VR system also outputs its orientation. When the Procrustes algorithm is applied on locations in a data set, the orientations also need to be transformed. Figure 3 illustrates this need. Figures 3a and 3b show the locations and yaw angles of poses acquired in two data sets, which we will call the reference and source data set,

respectively⁴. If we align the source data set to the reference one and leave the orientations unchanged, the orientation angles are preserved with respect to the *new* coordinate system, as shown in Figure 3c. However, this means that the orientation of the tracker relative to other objects (or points) inside the room is lost. To obtain the correct new orientations, one option is to work with Tait-Bryan angles. We can convert the rotation matrix T^* given by the Procrustes algorithm into roll, pitch, and yaw angles, which we add to (or subtract from, depending on the rotation convention) the corresponding orientation angles of the object. This transformation yields the correct poses, as shown in Figure 3d.

B. Transforming Local Coordinates

The Vive tracker and the UWB tag will never be at exactly the same location because of the physical size of the devices. We mount the devices on a stand (Figure 1b) which maintains a fixed relative location and orientation between the two devices. Then, we derive a transformation matrix to convert the 3D coordinates of the VR tracker into the coordinates of the UWB tag, which depends on both the 3D coordinates and the orientation of the tracker.

We use the quaternion representation of the tracker's pose to compute the location of the tag, since it is more stable than Euler angle rotations. We denote the pose of the tracker by the quaternion:

$$\mathbf{q}_t = w + xi + yj + zk. \quad (2)$$

The relative location of the tag with respect to the tracker is given by the vector:

$$\mathbf{d} = d_x \mathbf{i} + d_y \mathbf{j} + d_z \mathbf{k}. \quad (3)$$

If the location of the tracker is $\mathbf{v}_t = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$, the location of the UWB tag can be computed as:

$$\mathbf{v}_u = \mathbf{v}_t + \mathbf{q}_t \mathbf{d} \mathbf{q}_t^{-1} \quad (4)$$

C. Evaluating Localization Errors

The end goal of a ground truth system is to compute the accuracy of the target localization system (in our case, the UWB one). In an ideal scenario, the UWB and Vive systems would have the same update rate and we could compute the localization error using samples acquired at exactly the same time. However, not only we cannot rely on the two systems to have the same update rate, but also this rate will vary in time because they are transmitted over a network. If we compute the error *online*, then we can simply compare the two latest locations given by the UWB and Vive systems.

However, if we analyze the data *offline*, we can use more information to find the ground truth location closest in time to a UWB location. One option is to find the Vive and UWB locations whose timestamps are closest to each other. However, if we assume a target with a speed of 1 m s^{-1} , a time difference of 10 ms between the samples can introduce an error of 1 cm

⁴The reference and source data sets correspond, respectively, to matrices B and A .

Algorithm 1 Finding the location corresponding to a timestamp t_{new} such that $|t_{ref} - t_{new}| < \Delta_T$, where t_{ref} is the reference timestamp and Δ_T is a chosen threshold.

```

1: procedure CLOSEST_LOC( $t_1, l_1, t_2, l_2, t_{ref}$ )
2:    $d_1 = |t_{ref} - t_1|$ 
3:    $d_2 = |t_2 - t_{ref}|$ 
4:   if  $d_1 < \Delta_T$  or  $d_2 < \Delta_T$  then
5:     if  $d_1 < d_2$  then Return  $l_1$  else Return  $l_2$ 
6:   else
7:      $t_{new} = t_1 + (t_2 - t_1)/2$ 
8:      $l_{new} = l_1 + (l_2 - l_1)/2$ 
9:     if  $t_{ref} < t_{new}$  then
10:      Return CLOSEST_LOC( $t_1, l_1, t_{new}, l_{new}, t_{ref}$ )
11:   else
12:     Return CLOSEST_LOC( $t_{new}, l_{new}, t_2, l_2, t_{ref}$ )

```

between the two locations. We can instead generate locations in-between two timestamps by approximating the trajectory between them with a line.

In other words, we want to generate the ground truth location between two locations l_1, l_2 acquired at timestamps t_1, t_2 ($t_1 < t_2$), respectively, such that the time difference between a reference timestamp t_{ref} and the timestamp of the ground truth location is smaller than a chosen threshold Δ_T . If we assume that the target is moving with constant velocity between l_1 and l_2 , we can generate the intermediate locations between t_1 and t_2 by recursively dividing the time period and the location span in half until the condition is met, as described in Algorithm 1.

IV. SYSTEM SETUP

The system is set up such that the Vive and UWB components can run independently and send their results (tracker location and anchor-tag distances, respectively) over a network through the MQTT network protocol [19]. A Python script collects these measurements, processes them, saves the results into files, and shows the locations of the tracker, tag, and anchors in a graphical interface in real-time.

Since the HTC Vive BSs have a range of 6.9 m^5 and the Vive system needs to see all the UWB anchors, the anchors should be placed over an area of approximately $6.9 \text{ m} \times 6.9 \text{ m}$. It is preferable to place them uniformly around the tracked perimeter [20]. The Vive BSs must be placed so that (1) they have direct sight to all anchors and (2) the maximum distance between each anchor and the BS does not exceed the maximum range of 6.9 m. In addition, the BSs should either face each other or otherwise be connected with a synchronization cable [21]; in our work, we chose the first option. The BSs should also be placed at different heights to minimize the geometric dilution of precision (GDOP) on the z axis (height). Figure 5b shows a suggested placement of the anchors and BSs.

⁵The tracking area of the HTC Vive can be increased up to $10 \times 10 \text{ m}$ by using four BSs.

To align measurements to a chosen reference system as described in Section III-A, the user needs to choose the location of at least three reference points, which must be kept constant for all future recordings. Therefore, they should be placed on walls or other furniture which will not be moved. The outline of the tracker can be drawn on the chosen surface, which will help the user place the tracker on the exact same spot each time. For each run (or recording), the user must perform the following steps:

- 1) Place the tracker on each of the reference spots; the software records a measurement each time a key is pressed.
- 2) Center the tracker on the antenna of each UWB anchor to acquire their location. Similarly, the software records a measurement when a key is pressed.
- 3) Track the devices. The software aligns the current locations to a chosen reference coordinate system (if given) and, once the tracking begins, will save: the location and orientation of the Vive tracker, the ground truth location and orientation of the UWB tag, and the location of the tag as measured by the UWB system.

Although it is slightly inconvenient that the system must be calibrated (steps 1 and 2) for each recording, these steps take less than 5 minutes. If the user is certain that the tracker has not disconnected between two recordings, the calibration can be omitted, case in which the reference points and the anchor locations can be read from an indicated directory (e.g. the directory in which the previous recordings were saved).

The UWB system computes the location of the tag using the regularized iterative least-squares (or Gauss-Newton) method presented in [22], which is suitable for online localization. The trilateration algorithm can yield large errors (more than 100 m) if the circles determined by the distance between anchors and the tag, which are centered at the anchors, do not intersect. This is a known issue of the algorithm and we handle it by restricting “valid” locations to the perimeter determined by the anchors (optionally, with a certain tolerance) and discarding locations outside of it.

UWB distance measurement errors acquired with 3db Access devices are inversely proportional to the signal-to-noise-ratio (SNR) up to a certain path loss. To estimate this relationship, we calibrated pairs of devices (the tag with each anchor) by measuring the distance between them at known, fixed points. Then, we fit a line to the distance errors (which increase linearly with the measured distance) and correct raw measurements using the coefficients of the fitted line. We used the Vive system to measure the true distances, since in static scenarios it is known to have mm-level accuracy [6].

V. EVALUATION

In this section, we evaluate the performance of the HTC Vive as a ground truth system for UWB localization. First, we analyze the accuracy of the HTC Vive in Section V-A. In Section V-B we investigate how to choose a set of reference points that minimizes alignment errors. Finally, in Section V-C

Table II: Mean and maximum localization errors of the HTC Vive.

	Location	Axis		
		x	y	z
Mean error [cm]	Wall (Left, Right)	0.74	0.84	3.31
	Table (Center)	0.29	0.24	0.81
	All	0.36	0.34	1.23
Maximum error [cm]	Wall (Left, Right)	2.89	2.69	13.14
	Table (Center)	2.28	0.99	5.48
	All	2.89	2.69	13.14

we evaluate UWB localization error using the proposed ground truth system.

A. HTC Vive Accuracy

First, we evaluate the accuracy of the HTC Vive. We mark 48 points inside a room shown in Figure 4b, out of which 40 points are on a table (*center* area) and 8 points are on the wall (*left* and *right* regions). Although the map shows only four points in the left and right areas, each marker corresponds to two test points at different heights with the same $x - y$ coordinates. All the test points on the table are at the same height of approximately 0.854 m and the points on the wall are at heights between 0.763–1.861 m. We measure the true locations and distances of the marked points, which we will use to compute the average accuracy of the HTC Vive. Figure 4a shows the three regions and the placement of BSs inside the room.

We acquired measurements using three different placements of the two Vive BSs, denoted by S1, S2, and S3 in Figure 4b. BS1 and BS2 are always placed at a height of 2 m and 1.3 m, respectively. We placed the tracker on each of the test points, acquired five location samples, averaged them, and recorded the average location at each test point. We recorded one series of measurements in setup S1 and three series of measurements in each of the setups S2 and S3, where one series of measurements contains all the 48 test points. Measurements in setup S1 were recorded during the evening (so in artificial lighting) and all recordings using setups S2 and S3 were acquired in daylight.

The locations given by the HTC Vive are placed in a local coordinate system. In order to compute the errors between Vive locations and the true locations of the test points, we use the Procrustes algorithm to find the optimal linear transformation between the two data sets. The Procrustes algorithm yields zero errors if the locations perfectly overlap, so any excess disparity between the transformed and the reference data set will be due to HTC Vive localization errors or systematic errors in the true locations⁶.

The goal is to compute the average localization error of the HTC Vive and evaluate whether measurements acquired in different areas of the room have different average errors, since there is a chance that errors increase near the edge of the

⁶We measured the true locations of the points as accurately as possible, so systematic errors in the true dataset should not exceed 0.5 cm.

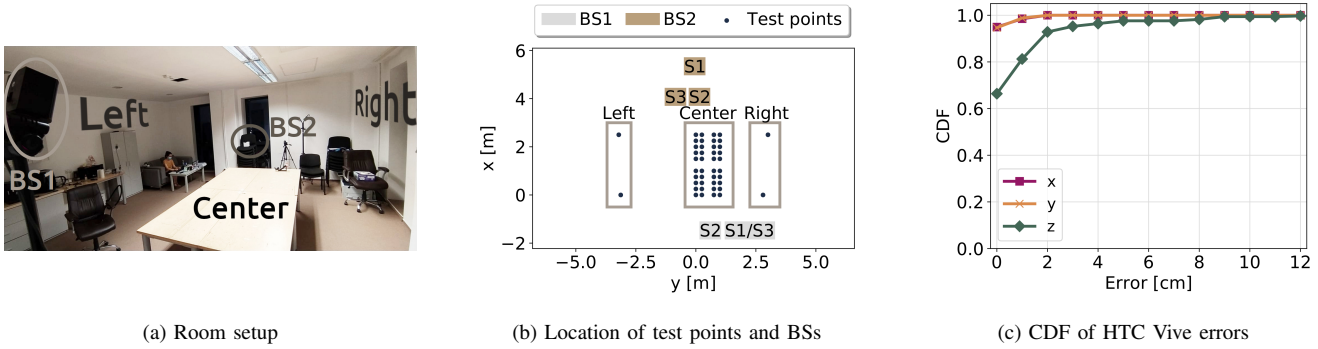


Figure 4: (a) Placement of devices inside the room corresponding to Figure 4b. (b) Layout in the x - y plane of test points, split into three regions. The points in the left and right regions are placed on the wall, while the points in the center are placed on a table. Each recording uses two BSs, placed above and below the tracking area, marked as rectangles. We tried three sets of BS placements, denoted by S1, S2, and S3. (c) CDF of HTC Vive errors on the x , y , and z axes.

field of view (FOV) of each BS (i.e. on the walls). Evaluating the accuracy of locations measured on the walls is especially important because this is where UWB anchors are typically placed. Since we use the HTC Vive to record the location of UWB anchors, Vive localization errors will propagate through the UWB tracking algorithm and it is critical for them to be as low as possible.

Since errors on the z axis (height) tend to be larger than on the x and y axis because of the higher GDOP, we present the localization errors on each axis. Figure 4c shows the CDF of HTC Vive localization errors on the x , y , and z axes and Table II shows the mean and maximum errors of measurements taken on the walls (left and right regions) and on the table (in the center of the room). On average, localization errors in the x - y plane are smaller than 1 cm and slightly higher in the z plane. It is notable that, although all measurements from the set S2 were acquired in strong lighting (sun shining on most of the table and some points on the wall), they did not seem to have significantly higher errors than the others *except for* two measurements on the wall which had the largest z -axis error of 13.14 cm.

B. Choosing the Reference Points

In this section, we want to investigate how to choose the reference points and their number in order to introduce the smallest errors when aligning data sets acquired in the same room. From the recordings described in the previous selection, we select two data sets acquired with different BS placements (we selected one with setup S2 and one with setup S3) and we will align one to the other using the Procrustes algorithm. We first split the 48 points from each data set into a training and a test set. We use the training set of points to compute the transformation between the reference and target coordinate system and the test set to evaluate the errors after the transformation.

To generate the training set, we split the points into three regions (left, center, and right) as shown in Figure 4b. For n training points, we choose an equal number of $\lfloor n/3 \rfloor$ points

Table III: Mean and maximum alignment errors for different numbers of reference points.

Nr. ref. points	Mean error [cm]			Maximum error [cm]		
	x	y	z	x	y	z
3	32.02	10.05	31.53	499.60	262.45	362.48
4	0.76	0.39	0.32	495.03	87.94	218.11
5	0.32	0.29	0.17	1.96	2.14	2.81
6	0.30	0.27	0.15	1.65	2.03	4.16

from each region and the remainder of points at random from all regions. This reduces the chance of obtaining combinations of training points that lead to high alignment errors (i.e. all training points on one side of the room). The test set comprises all the points not in the training set.

Table III shows the mean and maximum error on each axis when using between 3 and 6 training points. Using the minimum number of training points, while convenient, leads to large errors because the Procrustes algorithm cannot generalize the transformation matrix for the entire covered area. While using only one extra training point significantly reduces the average errors, bad configurations can also lead to large maximum errors. For 5 or 6 training points, the average errors on all axes are under 4 mm and none of them exceeds 5 cm. The configurations that yield the smallest errors contain at least one point on each wall, preferably in different corners of the tracking area, and at least one point in the center of the room. It is important to have at least two training points at different heights to obtain the correct transformation matrix.

C. Localization

We now evaluate the UWB localization error using ground truth measurements given by the HTC Vive. We use four UWB anchors, one UWB tag, one Vive tracker, and two Vive BSs placed as in configuration S3 from Figure 4b. We choose a combination of six training points similar to the one which minimized average errors in the previous experiment. Ideally, the UWB anchors should be placed as high as possible to

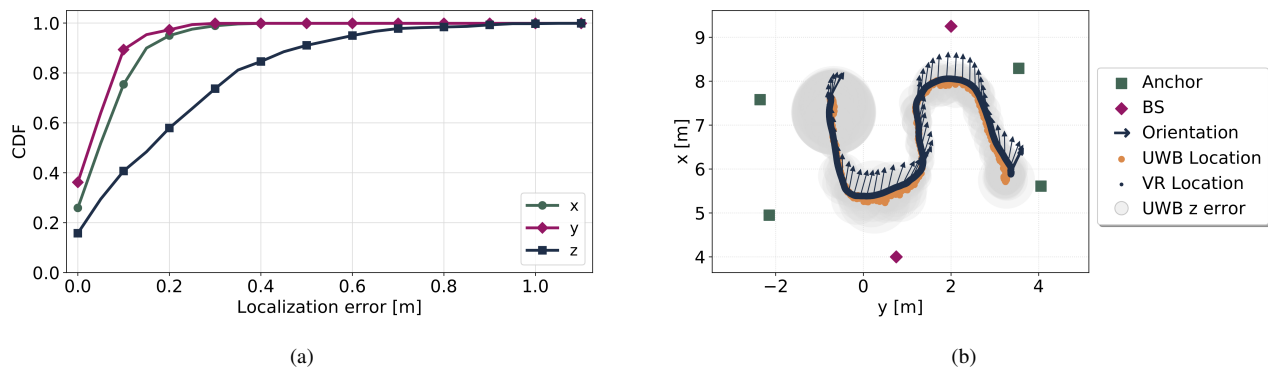


Figure 5: (a) Error on the x , y and z axes between UWB locations and ground truth Vive locations. (b) Map which shows the ground-truth location and orientation of the UWB tag and its location as computed by the UWB system.

improve the visibility to the tag. On the other hand, some anchors should be placed at lower heights to minimize the GDOP on the z axis. Because we need to measure the anchor locations with the Vive tracker, we have yet another constraint: UWB anchors should be reachable by the person who performs the calibration phase. To best satisfy these constraints, we placed the UWB anchors at heights between 1.5–1.8 m.

We performed two rounds of measurements with different BS placement in the $x - y$ plane, similar to configurations S2 and S3 from before, and maintained the same anchor locations and training points. We moved the tracker and tag at walking speed inside the perimeter delimited by the UWB anchors. In addition, we ensured that the tracker is in LOS with at least one BS at all times and tried to keep the tag in LOS with all the anchors. We aligned the second set of measurements to the first one. We compute the closest ground truth location to a given UWB location using the algorithm from Section III-C.

Figure 5a shows the CDF of the absolute value of localization errors on the x , y and z axes. The z axis has significantly larger errors than the other axes because of the high GDOP (there is a height difference of only 30 cm between the anchors). The median error on the x and y axis is 9.6 and 7.5 cm, respectively. On both of these axes over 90% of errors are under 20 cm. Figure 5b shows the measured and ground truth tracks of one of the two trials.

The Vive tracker also records its orientation, which coincides with the orientation of the UWB tag. This information is useful because UWB measurements can be affected by irregular antenna radiation patterns. Knowing the orientation of the tag, we can identify which antenna alignments cause larger errors in distance measurements and we can use this information to optimize the placement of devices in the localization system. The orientation of the tracker or tag in the $x - y$ plane is also shown in Figure 5b.

Because the average error of the UWB system in the $x - y$ plane is about 10 cm and on the height axis approximately 25 cm, we consider the HTC Vive acceptable as a ground truth system since its average errors on the same axes are at least 10 times lower than the UWB ones.

D. Discussion

Using the HTC Vive as a ground truth system has several disadvantages. First and foremost, the HTC Vive is not a measurement tool and therefore does not guarantee an upper bound on its localization errors. While, on average, its static measurements have mm accuracy, in some cases we noticed higher errors of even 13.14 cm. This can be problematic if we need sub-cm accuracy for *all* measurements, especially since we acquire the location of UWB anchors using the Vive system. For the best performance, HTC recommends using the Vive system in low lighting, connecting the BSs with a synchronization cable, and has additional advice for placing the BSs [21]. One way to minimize the chances of large ground truth errors is to acquire the reference data set in these ideal conditions. When subsequent data sets are aligned to the reference one, if the disparity between the aligned training points is larger than a threshold, we can assume that the current setting is not ideal and reconsider the conditions of the experiment.

Another inconvenience is that, although the calibration step takes less than five minutes, it is still inconvenient to perform it at the beginning of each round of measurements. In the future, we could eliminate the need to manually acquire the locations of UWB anchors and instead compute them using the location returned by the tracker and the distances between each pair of UWB devices in use albeit with possibly higher errors.

Even with its shortcomings, the HTC Vive remains an attractive option for acquiring ground truth measurements for wireless LSs given its low price (at least when compared with high-end motion capture systems), high accuracy, and ability to acquire thousands of static and dynamic measurements with a small setup overhead.

Although we tested the ground truth system on UWB localization, the same framework can be applied to any other anchor-based wireless LS. The location of the anchors can be acquired with the procedure described in Section IV and then sent to the target LS through the MQTT network protocol. The HTC Vive and the target LS can then operate independently of

each other and only need to send the estimated locations via MQTT to a computer that centralizes the data and computes the localization error of the target system.

This framework can also be used as a starting point for a joint UWB-HTC Vive LS, since the two components have characteristics that complement each other. The HTC Vive offers localization with mm-level accuracy and precision, but can work only when the BSs are in LOS with the tracker. The UWB LS, on the other hand, works also in non-line-of-sight (NLOS) scenarios and could provide locations with decimeter-level accuracy when the VR system is unreachable. Many tourism applications could benefit from such a joint LS; for instance, museums could offer visitors wearables which use the UWB system to guide them through the museum and the VR component to closely interact with certain exhibits.

VI. CONCLUSIONS

We presented a framework for using the HTC Vive as a ground truth system for anchor-based wireless localization systems and tested it on UWB localization. We introduced a method to transform recordings acquired in the same room in the same coordinate system, in order to easily compare results with different hardware setups over time. We use the HTC Vive to acquire the locations of anchors, which allows us to test different hardware arrangements with little overhead.

Future research directions include simplifying or eliminating the calibration process, finding more setup guidelines which minimize HTC Vive errors, applying the framework to other localization systems, and integrating the UWB and HTC Vive in a joint localization system.

REFERENCES

- [1] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.
- [2] S. Lanzisera, D. T. Lin, and K. S. Pister, "RF time of flight ranging for wireless sensor network localization," in *2006 international workshop on intelligent solutions in embedded systems*. IEEE, 2006, pp. 1–12.
- [3] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, pp. 1–32, 2013.
- [4] A. Pages-Zamora, J. Vidal, and D. H. Brooks, "Closed-form solution for positioning based on angle of arrival measurements," in *The 13th IEEE international symposium on personal, indoor and mobile radio communications*, vol. 4. IEEE, 2002, pp. 1522–1526.
- [5] F. Gustafsson and F. Gunnarsson, "Positioning using time-difference of arrival measurements," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings (ICASSP'03)*, vol. 6. IEEE, 2003, pp. VI–553.
- [6] M. Borges, A. Symington, B. Coltin, T. Smith, and R. Ventura, "HTC Vive: Analysis and accuracy improvement," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2610–2615.
- [7] D. C. Niehorster, L. Li, and M. Lappe, "The accuracy and precision of position and orientation tracking in the HTC Vive virtual reality system for scientific research," *i-Perception*, vol. 8, no. 3, p. 2041669517708205, 2017.
- [8] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, 2016, pp. 165–178.
- [9] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 269–282.
- [10] M. Cominelli, P. Patras, and F. Gringoli, "Dead on arrival: An empirical study of the Bluetooth 5.1 positioning system," in *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, 2019, pp. 13–20.
- [11] B. Großwindhager, M. Stocker, M. Rath, C. A. Boano, and K. Römer, "SnapLoc: An ultra-fast UWB-based indoor localization system for an unlimited number of tags," in *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Apr. 2019, pp. 61–72, ISSN: null.
- [12] D. Neuhold, C. Bettstetter, and A. F. Molisch, "HiPR: High-precision UWB ranging for sensor networks," in *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems - MSWIM '19*. Miami Beach, FL, USA: ACM Press, 2019, pp. 103–107.
- [13] P. Khandelwal and P. Stone, "A low cost ground truth detection system for RoboCup using the Kinect," in *Robot Soccer World Cup*. Springer, 2011, pp. 515–527.
- [14] D. Lymberopoulos and J. Liu, "The Microsoft Indoor Localization Competition: Experiences and lessons learned," *IEEE Signal Processing Magazine*, vol. 34, no. 5, pp. 125–140, 2017.
- [15] C. Sánchez, P. Taddei, S. Ceriani, E. Wolfart, and V. Sequeira, "Localization and tracking in known large environments using portable real-time 3D sensors," *Computer Vision and Image Understanding*, vol. 149, pp. 197–208, 2016.
- [16] A. Ledergerber and R. D'andrea, "Calibrating away inaccuracies in ultra wideband range measurements: A maximum likelihood approach," *IEEE Access*, vol. 6, pp. 78 719–78 730, 2018.
- [17] C. Honnet and G. Lopes, "HiveTracker: 3D positioning for ubiquitous embedded systems," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, 2019, pp. 288–291.
- [18] P. H. Schönemann, "A generalized solution of the orthogonal Procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.
- [19] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S—A publish/subscribe protocol for wireless sensor networks," in *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*. IEEE, 2008, pp. 791–798.
- [20] N. Salman, H. K. Maheshwari, A. H. Kemp, and M. Ghogho, "Effects of anchor placement on mean-CRB for localization," in *2011 The 10th IFIP Annual Mediterranean Ad Hoc Networking Workshop*. IEEE, 2011, pp. 115–118.
- [21] "Vive Pro HMD user guide," accessed on 17-09-2020.
- [22] N. Sirola, "Closed-form algorithms in mobile positioning: Myths and misconceptions," in *2010 7th Workshop on Positioning, Navigation and Communication*. IEEE, 2010, pp. 38–44.