ViPER: **V**ehicle **P**ose **E**stimation using Ultra-WideBand **R**adios

Alireza Ansaripour, Milad Heydariaan, Omprakash Gnawali
Department of Computer Science
University of Houston

Kyungki Kim
Durham School of Architectural Engr. & Construction
University of Nebraska-Lincoln

***Abstract*—Pose estimation is a building block for many location-based applications, such as safety applications in a construction site. Ultra-WideBand (UWB) Radios have been widely used for localization and can be used in pose (location and orientation angle of the object) estimation primarily because of the accuracy with which these radios can estimate the arrival time of radio signals. Current UWB pose estimation solutions do not perform adequately in Non-Line of Sight (NLoS) conditions. Some of these existing solutions in pose estimation rely on two or more types of sensors to tackle the NLoS challenge. These methods suffer from data fusion complexity, making the system not generalizable and limited to some specific simple environments, such as labs. In this paper, we propose ViPER, a UWB-based pose estimating system using only UWB radios. Our goal is to reduce the effects of the NLoS without the inclusion of any auxiliary sensors. ViPER uses low-pass filter, anchor and reference selection method to reduce the effect of NLoS in the measurements. It also estimates the pose of the entities using an optimization problem. We have evaluated ViPER in real-world highway construction and parking lot setting. We find that it improves the average packet reception ratio by 117% and decreases the error rate by 70% over the state of the art in Non-Line of Sight situation.**

***Index Terms*—Pose estimation, Indoor localization, Ultra-wideband, Vehicle tracking, Non-Line of sight, Reference selection, Anchor selection**

I. Introduction

The improvements in wireless technology and embedded systems have resulted in accurate and robust localization solutions. Among all these wireless technologies, ultra-wideband (UWB) localization is one of the technologies gaining popularity in the deployment of indoor localization systems due to its specific characteristics, including the ability to perform accurate timestamping of arriving signals [1]. The low-cost commercial UWB chips, such as DW1000, enable accurate localization in indoor and outdoor settings. In the industrial environment, indoor localization has brought many applications to improve productivity [2].

Pose estimation is the method of estimating the location, orientation of an object, and has many applications. Fig. 1 displays these parameters for a vehicle. The system estimates the location and orientation from data gathered by sensors and calculates the boundary based on the shape of the object. Pose estimation can be useful in safety applications in a construction site to track the location and boundary of working equipment inside of a working area. It can increase the construction safety for both the workers and the machinery, which is critical inside construction sites [3]. Pose estimators are able to track the location and orientation angle of entities inside a construction zone in real-time. By doing this, the system is able to monitor the policies (e.g. proximity detection of equipment) applied for ensuring construction safety. Thus, they can reduce the number of casualties inside a construction zone. Moreover, these systems can be used for indoor and outdoor vehicle tracking.

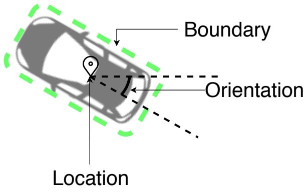


Fig. 1: Pose of the vehicle that is consist of location, orientation, and its boundary

Estimating the location and orientation of the equipment is one of the main challenges in implementing these systems. Despite the reported centimeter-level accuracy of the UWB radios in estimating the range between the sender and the receiver, their accuracy can drastically fall in realistic scenarios if the first path of a signal was to be obstructed. These conditions, referred to as Non-Line of Sight (NLoS) conditions, and factors, such as time synchronization of nodes, temperature, or environmental noise, can also affect the accuracy of the localization [4]. Such errors can make it challenging to accurately determine the location and orientation of the equipment. Moreover, in many applications of pose estimation, the data must be available for analysis in real-time to adapt to the changing activities of the workers and the equipment in a construction field and to raise alerts in case of potential safety violations. Therefore, providing an acceptable pose update rate is also an important issue in these systems.

Existing UWB pose estimation solution mostly disregarded the problem of NLoS situation in their works because they evaluated the feasibility of their solution. However, NLoS solutions are unavoidable in real-world environments, especially in construction sites. Therefore, their solutions are not applicable to these environments. Multi-sensor pose estimators use other auxiliary sensors to mitigate the effect of NLoS errors in UWB radios. The problem with these solutions is that the process of combining data from multiple sensors, know as data fusion, is a very complicated task. The complexity of this task leads to assumptions and limitations for the environment and objects, making the solution not generalizable for all environments.

In this work, we propose ViPER, a real-time equipment pose estimation system that tackles the pose inaccuracy problem in UWB pose estimation methods. In our design, we first remove the noise of the data gathered from the anchors to reduce errors before the localization process. Later we use our pose estimator to estimate the location and orientation based on the points generated by the localization engine.

We implemented ViPER on radinoL4 DW1000 platform and evaluated it in a real construction field with equipment and workers and an outdoor campus setting with a passenger vehicle. Our system reduced the error rate by 70% and improved the average update rate by 117% in NLoS situation compared to the state of the art. Our contributions are:

- Design of a reference anchor selection method to reduce the NLoS error in TDoA localization.
- Design of an optimization based pose estimator method for calculating the pose of the objects.
- Evaluation of our solution against a state-of-the-art alternative in both NLoS and LoS conditions both nlos and los conditions in a real-world highway construction environment.

II. Related Work

We describe the three main areas of localization-related research that are related to our work.

A. Pose Estimation

Pose estimation is an established area of research with a rich body of literature briefly outlined in Table I. No location or IMU sensor is perfect. Hence, this research tries to compute pose within an acceptable error margin despite errors in the sensor and input data.

Zhang et al. designed a system to track the boom of a crane by aggregating, averaging, and interpolating data from multiple UWB tags installed on different parts of the boom [3]. Their technique uses the UWB tags that are always in the line of sight (LoS) while ViPER is designed for scenarios in which there is no fixed subset of UWB tags that are always in LoS with the anchors. Formulating the pose estimation problem as an optimization problem has been shown to be better than the averaging-based approaches. Optimization-based excavator pose estimation [5] has a specific formulation for finding the center of rotation that does not apply to general vehicles. ViPER provides a formulation that is applicable to general vehicles and has specific mechanisms to address NLoS issues in the field.

The second group of solutions deploy two or more types of sensors (e.g. some combination of UWB, GPS, and IMU) for pose estimation [6], [7]. These solutions require sophisticated data fusion techniques increasing the design and operational complexity over ViPER's UWB-only approach.

B. Non-Line of Sight Mitigation

For both ranging and localization to be accurate, the first-path of the signal has to travel from the sender to the receiver in a straight line, without any object blocking the way. In NLoS condition, the first-path is somehow omitted, delayed, or diffracted so that the receiver fails to estimate the true arrival time of the signal. This situation can lead to error in estimating the range or location.

There are two general methods in mitigating the NLoS error [8]. The first method is the first-path detection which tries to detect the first-path [9]. The second type of NLoS mitigation approaches are statistics-based methods. Residual Weighting [10] works reasonably well but computationally expensive to run in real-time systems. Some researchers mitigate the errors caused by NLoS scenario by applying Kalman Filter to the ranging information gathered by the anchors [11]. ViPER uses filters for localization input to improve the accuracy but introduces additional mechanisms because in the TDoA algorithm, anchors do not report ranging information.

C. Reference and Anchor Selection in TDoA

Anchor Selection can be used to select a subset of anchors with less errors so we do not use data from anchors with large errors in TDoA. Reference Selection is used to select one of the anchors to provide global reference time. TDoA performance can degrade if timekeeping in the reference anchor is unreliable (too high variance, jitters, drift, etc.): the reference time impacts how time is computed throughout the network in TDoA [12].

Guvenc et al. used signal-to-noise ratio to find the nearest anchor to the tag and choose the nearest anchor as the reference anchor [13]. However, the nearest anchor is not necessarily in LoS or with the most accurate and reliable clock. Xu et al. calculated the position for every anchor as the reference and chooses the one with minimum residual value as the reference [14]. Residual value method for finding the best reference anchor is more general and accurate but has computational overhead making it challenging for real-time localization systems. Our proposed method for anchor and reference selection is less computationally expensive, making it suitable for real-time localization systems.

III. System Design

We design ViPER to decrease the error caused by NLoS conditions. We use low-pass filtering, anchor and reference selection, and an optimization method to improve the estimation of the location and orientation of equipment. Our design strives for simplicity by not using data fusion with information from sensors/radios other than UWB.

The system is composed of three subsystems: sensing infrastructure, localization engine, and pose estimator as shown in Fig. 2. Sensing infrastructure consists of the tags that are mounted on the equipment we want to track. For tracking location, only one tag is suitable; however, more than a single tag is needed for pose estimation. Mounted tags send radio signals that are received by the anchors. These anchors are

TABLE I: Proposed methods for pose estimation

Publication	Sensors	Update Rate (locations / second)	Experiment Environment	NLoS Situation	Data Fusion Technique
[3]	Multiple UWB	4	Construction Site	Body of the boom	Averaging
[5]	Multiple UWB	0.2	Lab	None	Optimization
[6]	IMU + single UWB	5	Lab	None	Kalman Filter
[7]	GPS + IMU	75	Outdoor	None	Kalman Filter
ViPER	Multiple UWB	5	Outdoor Construction Site	Body of the Vehicle Loader blocking some anchors	Optimization

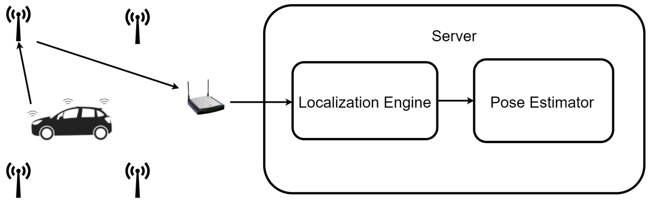


Fig. 2: Design overview of ViPER. The output of the pose estimator is the pose of the vehicle.

connected to the server via WiFi infrastructure. Anchors report the timestamp of the received signals along with some other data to the server.

ViPER has a server on the cloud running localization and pose estimation engines. The localization engine calculates the location of tags using the TDoA algorithm. TDoA, among ToA algorithms, is known to be the most robust in NLoS situations [12]. In order for TDoA to calculate the location of the tag, at least four anchors have to report the timestamp of the received signal from that tag to the localization engine. The system then chooses one of the anchors to be the reference anchor. The location is then calculated by solving an optimization problem. The optimization problem tries to find the (X_{min}, Y_{min}) that minimize the value of $f(x, y)$ in Eq. 1.

$$f(x,y) = \sum_{i}^{N} (\sqrt{(x-x_i)^2 + (y-y_i)^2} - \Delta d_i)^2 \qquad (1)$$

$$d_i = C * (t_i - t_{ref}) \qquad (2)$$

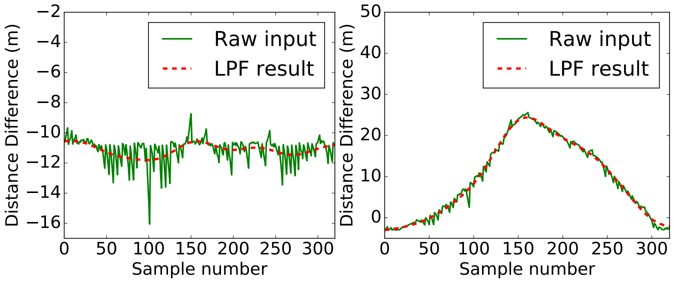
where N is the number of reported timestamps, d_i is the distance difference for anchor i and the reference anchor, t_i is the received timestamp for anchor i, t_{ref} is the timestamp for reference anchor, C is the speed of light, and (X_{min}, Y_{min}) is the location of the tag.

Finally, when the locations of the tags are calculated, they are passed to the pose estimator for pose determination. The pose estimation calculates the location and the orientation of the object based on the location of the tags.

A. *Low-pass filtering*

We apply the low-pass filter to the raw ranging data before passing them to the anchor selection and reference selection methods. Thus, those selection methods have reasonably clean input data and are able to achieve better results.

The TDoA inputs, which are distance differences, are calculated using Eq. 2. The value of d_i is derived from t_i and t_{ref} that are the received timestamp of the signals reported by anchors. Our experience suggests that such filtering is critical to achieving good results.



(a) Static tag

(b) Mobile tag

Fig. 3: The recorded distance differences (raw input) and its filtered result (LPF result) for different tags. The LPF result contains fewer fluctuations that indicate less noise compared to the raw input.

To understand the extent and type of error the raw ranging data has, we do a simple experiment. We put a static tag in one location and two anchors with one as a reference in the vicinity and recorded the distance difference. Fig. 3(a) displays the distance difference data from the anchors for a static tag using a solid line (Raw input). When the tag is not moving, we expect the values to be constant because the signal travels the same distance in every measurement. However, the results show frequent changes in the input in some measurements. In order to enhance the accuracy, these frequent changes, considered as noise, are required to be removed from the input.

We also had the same observation for a mobile tag moving in different directions. The result is shown in Fig. 3(b) with different Y-scale compared to Fig. 3(a). Even for the mobile tags, TDoA inputs follow a smooth changing pattern. Therefore removing the frequent changes can also enhance the accuracy of the localization in these tags.

The advantage of using a low-pass filter over the Kalman Filter is that it requires fewer environmental parameters. The only environmental parameter we had to determine was the frequency cutoff for the filter. This value can be different for every tag depending on how quickly the target changes its location. High-frequency cut-off reduces the effect of low-pass filter on smoothing the data. Meanwhile, low-frequency cut-off can cause errors when the object is moving quickly. We empirically determined 5 to be a good threshold and applied the low-pass filter to the input. The output results are shown in Fig. 3(a) and Fig. 3(b) with a dashed line (LPF result).

B. Anchor Selection

Despite the correction done by the low-pass filtering, our observation suggests that low-pass filtering does not work well in cases where the error in one or more anchors are too high. In these cases, where we failed to adequately correct the input error, the data from that anchor has to be removed. Therefore, we designed an anchor selection method to detect and remove the TDoA inputs with high errors.

In our design, TDoA inputs having a high difference between the raw and the filtered value are good candidates for removal because of the error they introduce in localization. Fig. 4(a) and 4(b) show all TDoA inputs of all anchors along with their filtered results for the static tag with anchor #1 being bolder than other anchors. We have also included the output of our anchor selection method on the top of the figure making the reference anchor bolder than other anchors. As shown in the figure, in sample number 245, the difference between the raw input and the filtered result is approximately 5 m for the anchor #1, causing the filtered result to curve toward the error. In situations like this, the proposed anchor selection solution does not feed the data for this anchor to the next step in the pipeline. Even though we removed the timestamp of that anchor, there are still five other anchors that are sufficient for localization. A new round of anchor selection happens each time the location of the tag is being calculated by the localization engine.

C. Reference Selection

According to Eq. 2, the timestamp of the reference anchor is used to calculate all the TDoA inputs for the localization process. Therefore, error in the received timestamp of the reference anchor can lead to miscalculation of TDoA input for all anchors. It is important for reference selection to choose the anchor with the least amount of error as a reference.

The goal of reference selection is to choose the best anchor to be the reference, avoiding the propagation of error to all TDoA inputs. For example, in sample number 101 in Fig. 4(b), all anchors except the reference are removed by the anchor selection method due to exceeding the difference limit. In situations like this, there is a probability that the miscalculation in received timestamp of the reference anchor caused the error to propagate to other TDoA inputs. In Fig. 4(c), we have generated the TDoA input time series for anchor four as reference. With the new calculation, only anchor zero, the previous reference, is removed by the anchor selection method.

Therefore, in our reference and anchor selection method, first, the distance differences are calculated for each anchor as the reference. Then, for each reference, the anchor selection removes the incorrect inputs. Finally, the reference selection method chooses the anchor with the smallest number of removed anchors as reference. This process occurs every time the location of the tag is being estimated because the correctness of the timestamp is different for anchors in each calculation.

D. Pose Estimator

The pose estimator computes the vehicle position and its orientation based on the provided locations in a certain time slot. The locations generated by the localization engine are passed to the pose estimator at the end of each time slot. The pose estimator determines the location and the orientation of the vehicle based on the locations provided. Depending on the precision of the locations, the pose estimator may not be able to estimate the position of the vehicle. In this situation, it will not report the pose for that time slot.

In our method for pose estimation, we calculate the location and orientation of the vehicle by solving two optimization problems. The first optimization problem calculates the location of the center of the vehicle and the second one determines the orientation, based on the generated locations of tags installed on the specific places of the vehicle.

One of the steps in reducing error is removing the points that might be erroneous. We use residual value as an indicator for the accuracy of the calculated location as typically done for this type of problem [10]. Therefore, before passing the locations to the pose estimator, we remove the locations with residual values higher than a threshold. We empirically determined 5 as the best threshold value for this step.

After removing the inaccurate locations, the next step in determining the pose of the vehicle using optimization problems to estimate its location and orientation. Our optimization method for pose estimation tries to find the (x, y, θ) that minimizes the objective function in Eq. 3:

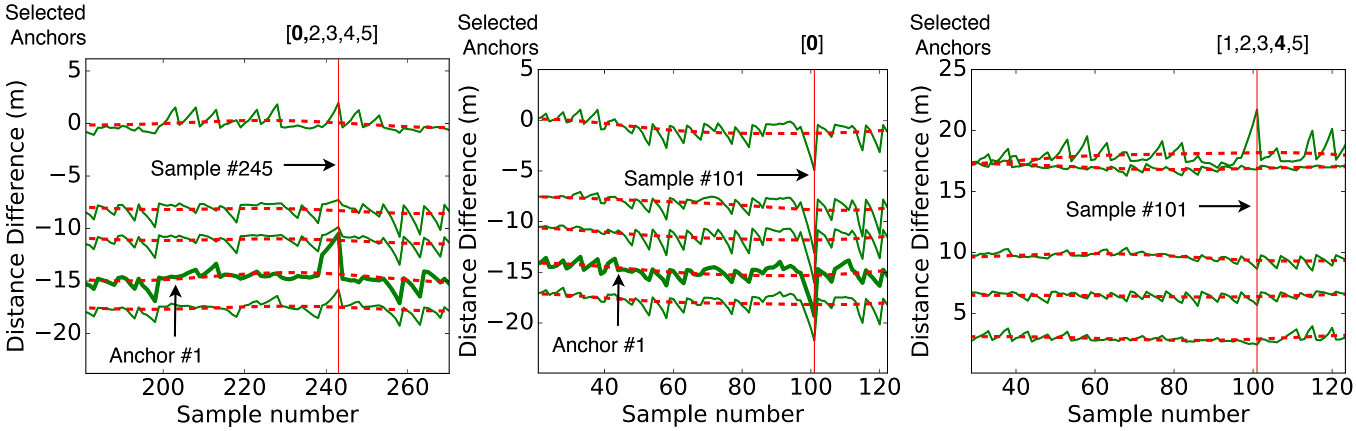
$$f(x,y,\theta) = \sum_{i=1}^{T}\sum_{j=1}^{size_i}\left(\sqrt{(X_i - x_{i,j})^2 + (Y_i - y_{i,j})^2}\right) \quad (3)$$

$$\begin{bmatrix} X_i \\ Y_i \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} * \begin{bmatrix} p_{x,i} \\ p_{y,i} \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}; \quad (4)$$

where

(x, y)	Position of the center of the vehicle
θ	Orientation of the vehicle
T	Number of tags on the vehicle
$size_i$	Number of locations from tag i
$(x_{i,j}, y_{i,j})$	jth location of tag i
$(p_{x,i}, p_{y,i})$	Position of the ith tag relative to the center of the vehicle
(X_i, Y_i)	$(p_{x,i}, p_{y,i})$ with θ rotation

Solving this non-convex optimization is computationally difficult when there are three parameters in the objective function. We also had the local-optima problem in some points where the optimizer could not find the correct location and the orientation of the vehicle. To solve these two challenges, we made the problem easier by breaking it into two separate optimization problems to reduce the optimization parameters



(a) Sample #245

(b) Sample #101 with anchor 0 as reference

(c) Sample #101 with anchor 4 as reference

Fig. 4: Our proposed method for anchor selection in (a) removes the anchor #1 for sample #245. In (b) all anchors are removed by the anchor selection method in sample #101. The reference selection in (c) reduces the number of removed anchors in sample #101 by choosing the best anchor as reference.

required for each objective function. In our observations, we figured out that if we had a rough estimation of our orientation, we can fix the orientation and try to find the location of the vehicle by solving the problem in Eq. 3. In our design, we can assume that the orientation of the vehicle does not change dramatically between time slots. We used the previous orientation as our rough estimate. Therefore, first, we solve the problem described in Eq. 5, with only two parameters (x and y) to find the location of the center of the vehicle.

$$f_1(x, y) = f(x, y, \hat{\theta}) \tag{5}$$

where $\hat{\theta}$ is the previous orientation of the vehicle.

The second optimization problem calculates the orientation of the vehicle based on the location generated by the first steps. Eq. 6 calculates the new orientation of the vehicle.

$$f_2(\theta) = f(X_{min}, Y_{min}, \theta) \tag{6}$$

where (X_{min}, Y_{min}) is the location of the vehicle which is the output of the first step. In this function also we had only one parameter, θ, that needs to be optimized.

E. Error threshold for pose estimation

Existing works in proximity detection systems, suggest using 8 m as the safe boundary for vehicles in the construction zone [15]. They also stated that, in their design, they considered 1 meter as the error threshold. We considered 15° as an error threshold for the orientation angle, as they cause under 1 m error in distance. The maximum speed limit is 4.47 m/s (10 mph), in order to have under 1 m of displacement between two updates, the pose of the entities is updated every 0.2 s.

According to our calculation, 0.6 s is required for the system to notify. with our 0.2 s update interval, three measurements are taken. With the error rate of less than 37% and pose reception rate (PRR) of more than 70%, the probability of missing an alert will fall below 5%, which is acceptable for our system.

IV. Evaluation

We evaluated ViPER in two real-world outdoor environments. The first experiment was done in the University of Houston campus parking lot in a near-perfect setting. The goal was to evaluate the feasibility of the pose estimator solutions. The environment for this experiment was chosen so that the effect of the NLoS condition was minimized. We also deployed all of our tags on the vehicle to make sure the pose estimators have sufficient data to determine the pose of the vehicle.

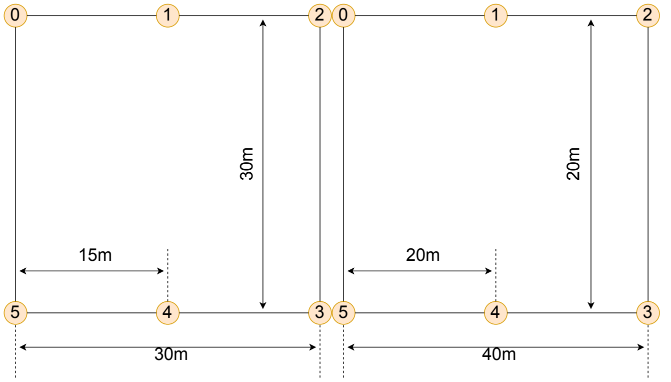
The second experiment was held at a highway construction site with vehicles and trucks creating NLoS condition for the system. This experiment was designed to evaluate ViPER in a harsh NLoS environment. In this experiment, the number of tags installed on the vehicle was also reduced so the settings could be closer to the real environmental settings of the system.

A. Experiment Setup

In both experiments, we dedicated a field known as the tracking zone. The placement of anchors along with their IDs is shown in Fig. 5(a) for the first experiment that was done in the and Fig. 5(b) for the second experiment.

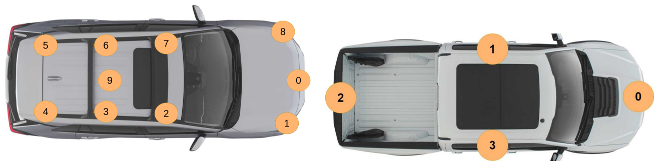
We evaluate ViPER with two different types of vehicles. For each vehicle, we selected a different number of tags with different placements to evaluate ViPER in different settings. The placement of tags and their IDs is shown in Fig. 6(a) for vehicle 1 and Fig. 6(b) for vehicle 2. Vehicle 1 was used for the first environment and vehicle 2 was used in the second one. Vehicle 1 is a 4-door sedan. Vehicle 2 is a pickup truck used in the construction site.

Three different sets of scenarios were tested in these environments for the evaluation of ViPER. In the first one,



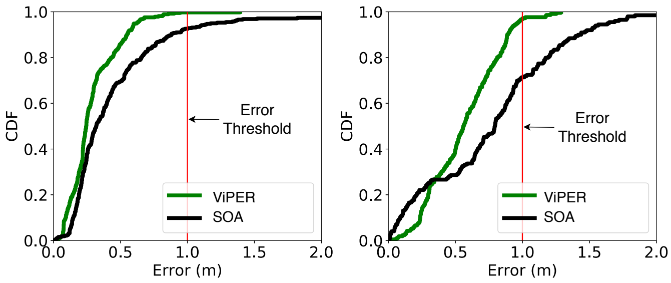
(a) First experiment (b) Second experiment

Fig. 5: Tracking zone and anchor placement for each experiment in (a) campus parking lot (b) road construction site



(a) First experiment (b) Second experiment

Fig. 6: Tag placement on vehicles for each experiment (a) campus parking lot (b) road construction site



(a) static tag (b) mobile tag

Fig. 7: Comparison of error rate CDF between ViPER and the state-of-the-art (SOA) in single tag localization for (a) a static tag with 370 locations and (b) a moving tag with 260 locations. Our anchor and reference selection method enhances the localization output compared to the SOA.

the vehicle was stationary in the center of the tracking zone with 0° orientation with the x-axis. In the second scenario, the car drove around a circle centered in the middle of the tracking zone at a constant speed. These two scenarios were conducted in the first benign environment. In the final scenario, the vehicle traversed a line parallel to the x-axis through the tracking zone and came back with the reverse gear. This last scenario took place in the second environment, i.e., construction site.

B. *Platform/Implementation*

We implemented a substantial part of ViPER on radinoL4 DW1000 platform for all our tags and anchors. We connected each anchor to a Raspberry Pi device for data collection and communication with the localization engine.

For time synchronization between all anchors, we placed a time reference anchor that broadcasts SYNC messages to all other nodes in the network every 500 ms. Upon reception of SYNC message, anchors reply in a TDMA-based scheme. Finally, tags send multiple replies in a TDMA-based scheme with a round-robin protocol. Anchors collect all these messages and forward them to the server for further processing.

We implemented our server using Python language. For solving the optimization problems, we used Scipy library designed for solving optimization problems.

C. *Single tag localization*

We evaluate the performance of ViPER's anchor and reference selection technique with the minimum distance reference selection, mentioned in the related work for the single anchor localization scenario. The error results are shown in Fig. 7(a) and Fig. 7(b) for the static tag with approximately 370 location points and the mobile tag with roughly 260 locations. The vertical line indicates the acceptable error margin for our application domain. It can be concluded that our selection method achieved approximately 10% improvement in static tag localization and 30% in moving tag localization over the SOA method.

D. *Multi-tag pose estimation*

In this part, we evaluate four cases of pose estimation in each scenario. Case 1 (SOA) is the state-of-art pose estimation mentioned that uses the averaging method [3] for pose estimation. Case 2 (OPT) uses the optimization method for pose estimation instead of the averaging method. This case evaluates the effect of our optimization method as a replacement for the averaging method. Case 3 (AR) uses averaging pose estimation but applies the proposed anchor and reference method. The goal of this case is to evaluate our anchor and reference selection method. Finally, Case 4 (ViPER) uses both the proposed techniques, anchor and reference selection and our method for pose estimation to consider the total improvement of our methods. In all cases, we remove the points with high residual values to improve the performance of all the techniques we compared.

For each case, we have evaluated the Pose Reception Rate and the error rate of the cases in tables II and III. The Pose Reception Rate refers to the ratio of the time slots that the pose estimator was successful in estimating the location and orientation. Error rate refers to the ratio of the calculated locations or orientations that exceed the error margin. We also provided the CDF of the location and orientation error in each case to evaluate the performance of the pose estimation in calculating each parameter individually.

1) Static Vehicle in near-perfect environment : For this scenario, we gathered data from 132 time slots. The results of this experiment are shown in Table II. Despite the near-

TABLE II: Evaluation results for static vehicle in near-perfect environment

	SOA	OPT	AR	ViPER
Pose Reception Rate	0.47	0.47	0.77	0.77
Error Rate	0.15	0.02	0.58	0.01
Location Error Rate	0.08	0.02	0.13	0.00
Orientation Error Rate	0.07	0.0	0.48	0.01

TABLE III: Evaluation results for moving vehicle in near-perfect environment

	SOA	OPT	AR	ViPER
Pose Reception Rate	0.94	0.96	0.97	0.98
Error Rate	0.66	0.08	0.58	0.06
Location Error Rate	0.02	0.00	0.03	0.00
Orientation Error Rate	0.66	0.08	0.57	0.06

perfect environment, where NLoS situations were avoided as much as possible, SoA results indicate 47% PRR. This low PRR occurred due to malfunctions as some of the UWB nodes reported incorrect Time-of- Arrivals.

Meanwhile, the results for ViPER demonstrate that our anchor and reference selection method improved PRR by approximately 63% over the SoA method by detecting and removing these incorrect inputs.

2) Moving Vehicle in near-perfect environment: In this scenario, 262 time slots were reported. The results in Table III show a near-perfect Pose Reception Rate because of the LoS condition of the environment. Meanwhile, other sources of error led to the miscalculation of orientation.

In this experiment, we had nearly 100% pose reception rate because all of our sensors were working correctly. Thus, the AR method did not make much improvement in the results.The results also suggest that the accuracy of the averaging method in calculating the orientation was lower by nearly 90%, suggesting that our proposed optimization method has better performance in determining the orientation.

3) Moving Vehicle in construction environment: Our proposed method for anchor and reference selection achieved a 117% improvement in the Pose Reception Rate as shown in Table IV. Thus, we can conclude that the anchor and reference selection method outperformed the state-of-the-art reference selection method in both LoS and NLoS environments. In terms of error rate, our proposed solution reduced the error by 70% by reducing the orientation error.

E. Robustness

One challenge in a real-world deployment is the loss of tags or damage to the tags during construction work. It is critical for localization or pose estimation systems to be resilient to such tag damages or losses. In this section, we evaluate the robustness of ViPER against those adverse events. To emulate tag loss or damage, we randomly remove the tags from the vehicles. We then measure the impact of tag removal on ViPER's performance. Table V reports the effect of removed tags on the Pose Reception Rate and error rate for the moving LoS scenario.

TABLE IV: Evaluation results for moving vehicle in the construction environment

	SOA	OPT	AR	ViPER
Pose Reception Rate	0.46	0.46	1.00	1.00
Error Rate	0.88	0.27	0.51	0.24
Location Error Rate	0.19	0.23	0.23	0.11
Orientation Error Rate	0.76	0.12	0.44	0.19

TABLE V: Evaluation results related to the robustness of ViPER in near-perfect condition. The tag numbers correspond to the tag IDs in the tag placement map of Vehicle 1 shown in Fig. 6(a)

Case #	Removed Tags	Pose Reception Rate	Error Rate
1	[0,2,5]	0.93	0.12
2	[0,2,4,5,8,9]	0.93	0.37
3	[0,1,3,6,8,9]	0.93	0.45
4	[0,1,2,4,5,6,8,9]	0.92	0.37

The results show that the number and the location of removed tags can affect the error rate. Fig. 8(a) and Fig. 8(b) display the CDF of the location and orientation error. The results of these figures denote that the tag removal has more effect on the orientation than the location.

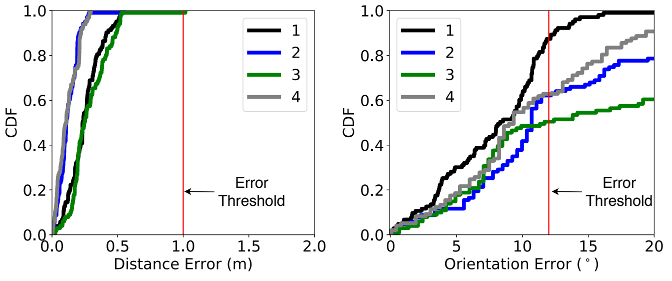
V. DISCUSSION

Although we did our experiments in two environments and with two different vehicles, they presented enough diversity in scenarios that there is some confidence that ViPER can be used in a broad range of scenarios. ViPER can also be utilized in applications other than tracking vehicles. In our design, the only assumption needed for pose estimation is the location of tags relative to a certain location of the object we want to track.

Our evaluation indicates that our proposed anchor and reference selection methods improved the average pose reception rate by reducing the number of time slots that the pose estimation failed to calculate the pose of the vehicle. By applying the anchor and reference selection methods before the running the TDoA estimator, we can reduce the localization error and decrease the residual values for points generated by the localization engine. Therefore, we remove less number of locations, making the pose estimation process converge to a result.

The observed location and orientation error indicate that simple pose estimation methods such as averaging are not effective in determining the orientation. In the averaging method, the error for orientation estimated using tags that are close to each other is comparatively high, leading to inaccurate orientation estimate. However, ViPER utilizes our knowledge of the relative location of the tags on the object we want to track. ViPER estimates the location and orientation by fitting the shape and orientation of the object according to the location of the tags.

One of the key factors in designing ViPER is removing a number of assumptions and limitations in pose estimation. We also aimed to reduce the number of environmental variables



(a)

(b)

Fig. 8: CDF of error rate for (a) location (b) orientation estimate for each case mentioned in TABLE V. Location estimation is more robust to tag removal compared to orientation estimation

required for pose estimation to just anchor placements and tag placements. Nevertheless, in our design, we defined a few variables regarding our observations. We have tested these values in two different environments for different types of tags such as static tags, tags mounted on the vehicle, and tags carried by construction workers. The results indicate that our proposed method is helpful in improving accuracy without changing the values of the threshold. Therefore, our solution is still deployable in most indoor/outdoor environments with a maximum speed of movement for objects.

One limitation of ViPER is the scalability of the number of tags in the system. ViPER adopts time-division multiple access (TDMA) to avoid packet collisions. In our implementation, we define 160 time- slots per second for tags to send their signals individually. This approach limits the maximum number of tags in our system. The limitation in the maximum number of tags affects the number of objects that can be tracked in our system. The maximum number of tracking objects depends on the number of tags installed per object. With the update rate of 1 Hz and 2 tags per object that is the minimum required for pose estimation, ViPER should be able to track 80 objects in the best case.

For our future work, we would like to consider the effect of tag placement on the pose estimation error. In ViPER, we used two different tag placements for our experiments. The number of tags mounted on the vehicle and their positions can be critical parameters in tuning the performance of our pose estimation system.

VI. Conclusions

In this paper, we have designed and implemented ViPER, a pose estimation system with UWB radios. Our main goal in design was to increase the average update rate and reducing the error rate in NLoS situations. In order to reach our goal, we developed an optimization method for pose estimation. We have also designed anchor and reference selection techniques to reduce the error in the localization process for better results in pose estimation. Our results indicate 117% increase in pose reception ratio and 70% decrease in the error rate.

Our proposed solution has been evaluated in the highway construction site for safety applications. With our proposed contributions, the probability of the false alarm false below 5%, which is acceptable for this application.

Acknowledgment

We thank Pulice company for providing experiment environment at one of their highway construction sites. We also thank Hosein Neeli, Nour Smaoui, and Taha Eghtesad for their help with the experiments. Research reported in this publication was supported by National Cooperative Highway Research Program (NCHRP) Idea of the National Academy of Sciences under award NCHRP-206.

References

[1] B. Groβwindhager, 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,” 2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Montreal, QC, Canada, 2019, pp. 61-72.

[2] V. Barral, P. Suárez-Casal, C. J. Escudero, and J. A. García-Naya, “Multi-sensor accurate forklift location and tracking simulation in industrial indoor environments,” Electron., vol. 8, no. 10, 2019.

[3] Zhang, C., A. Hammad, and S. Rodriguez. “Crane pose estimation using UWB real-time location system.” Journal of Computing in Civil Engineering 26, no. 5, 2012

[4] M. Heydariaan, H. Mohammadmoradi, O. Gnawali, “Toward Standard Non-Line-of-Sight Benchmarking of Ultra-Wideband Radio-Based Localization”. Proceedings - 2018 1st Workshop on Benchmarking Cyber-Physical Networks and Systems, CPSBench 2018.

[5] Vahdatikhaki, Faridaddin, Amin Hammad, and Hassaan Siddiqui. “Optimization-based excavator pose estimation using real-time location systems.” Automation in Construction 56 2015

[6] Strohmeier, Michael, et al. “Ultra-wideband based pose estimation for small unmanned aerial vehicles.” IEEE Access 6 ,2018.

[7] Weinstein, Alejandro J., and Kevin L. Moore. “Pose estimation of Ackerman steering vehicles for outdoors autonomous navigation.” 2010 IEEE International Conference on Industrial Technology. IEEE, 2010.

[8] J. Khodjaev, Y. Park, and A. Saeed Malik, “Survey of NLOS identification and error mitigation problems in UWB-based positioning algorithms for dense environments,” Ann. des Telecommun. Telecommun., vol. 65, no. 5–6, pp. 301–311, 2010.

[9] M. Heidari, K. Pahlavan “Identification of the absence of direct path in ToA-based indoor localization systems.” International Journal of Wireless Information Networks, 117–127.

[10] P. C. Chen “A non-line-of-sight error mitigation algorithm in location estimation”. IEEE Wireless Communications and Networking Conference, WCNC, 1, 316–320.

[11] C. Der Wann, C. S. Hsueh “NLOS mitigation with biased Kalman filters for range estimation in UWB systems”. IEEE Region 10 Annual International Conference, Proceedings/TENCON.

[12] S. Hara, D. Anzai, T. Yabu, K. Lee, T. Derham, and R. Zemek, “A perturbation analysis on the performance of TOA and TDOA localization in mixed LOS/NLOS environments,” IEEE Trans. Commun., vol. 61, no. 2, pp. 679–689, 2013.

[13] J. E. Rene, D. Ortiz, P. Venegas, J. Vidal “Selection of the reference anchor node by using SNR in TDOA-based positioning.” 2016 IEEE Ecuador Technical Chapters Meeting, ETCM 2016, 0–3.

[14] Q. Xu, Y. Lei, J. Cao, and H. Wei, “An improved algorithm based on reference selection for time difference of arrival location,” Proc. - 2014 7th Int. Congr. Image Signal Process. CISP 2014, no. 3, pp. 953–957, 2014.

[15] K. Kim, H. Kim, H. Kim “Image-based construction hazard avoidance system using augmented reality in wearable device.” Automation in Construction, 2017, vol. 83, 390–403.